

IBM Informix



Verze 11.1



Příručka administrátora serveru IBM Informix Dynamic Server

IBM Informix



Verze 11.1



Příručka administrátora serveru IBM Informix Dynamic Server

Poznámka:

Než použijete tyto informace a popisovaný produkt, přečtěte si informace v části “Upozornění” na stránce B-1.

Tento dokument obsahuje patentované údaje IBM. Poskytuje se na základě licenční smlouvy a je chráněn zákonem o autorských právech. Informace uvedené v této příručce nezahrnují záruky na produkt a jakékoliv prohlášení v tomto dokumentu by se mělo interpretovat jako takové.

Pokud odesíláte informace IBM, udělujete tím IBM nevýluční právo tyto informace používat a distribuovat způsobem, který považuje za odpovídající, aniž by tím IBM vznikl jakýkoliv závazek vůči vám.

© Copyright International Business Machines Corporation 1996, 2007. Všechna práva vyhrazena.

Obsah

Úvod	xvii
Obsah úvodní kapitoly	xvii
Obsah příručky	xvii
Druhy uživatelů	xvii
Závislosti na softwaru	xviii
Předpoklady pro národní prostředí	xviii
Demonstrační databáze	xviii
Nové vlastnosti serveru Dynamic Server, verze 11.10.	xviii
Konvence používané v dokumentaci	xix
Typografické konvence	xix
Značky vlastností, produktů a platforem	xx
Konvence použité v ukázkovém kódu	xx
Další dokumentace	xxi
Kompatibilita s oborovými standardy	xxi
IBM ocení veškeré připomínky	xxi

Část 1. Databázový server

Kapitola 1. Instalace a konfigurace databázového serveru	1-1
Obsah kapitoly	1-2
Plán pro databázový server	1-3
Posouzení priorit	1-3
Posouzení prostředí	1-3
Konfigurace operačního systému	1-4
Konfigurace paměti v systému Windows	1-4
Modifikace parametrů jádra systému UNIX	1-4
Konfigurace diskového prostoru	1-5
Použití velkých bloků	1-5
Vytvoření souborů bloků v systému UNIX	1-5
Vytvoření oddílů NTFS ve Windows	1-6
Nastavení oprávnění, vlastnictví a skupiny	1-6
Vytvoření standardních názvů zařízení (UNIX)	1-6
Nastavení proměnných prostředí	1-7
Nastavení proměnných prostředí GLS	1-8
Nastavení proměnných prostředí v systému UNIX	1-8
Nastavení proměnných prostředí v systému Windows	1-9
Konfigurace propojitelnosti	1-9
Soubor sqlhosts v systému UNIX	1-10
Registr sqlhosts v systému Windows	1-10
Konfigurace propojitelnosti pomocí programu ISA	1-10
Konfigurace databázového serveru	1-11
Příprava konfiguračního souboru ONCONFIG	1-11
Použití funkce Nastavení serveru obslužného programu ISA k vlastnímu nastavení konfigurace	1-12
Použití obslužného programu IBM Informix Server Administrator k aktualizaci souboru ONCONFIG	1-12
Použití programu Instance Manager k vytvoření nové instance databázového serveru (Windows)	1-12
Použití programu Instance Manager k přejmenování instance databázového serveru (Windows)	1-13
Konfigurace podpory jazyka Java	1-13
Spuštění a administrace databázového serveru	1-13
Spuštění databázového serveru a inicializace diskového prostoru	1-14
Příprava pro automatické spuštění	1-14
Příprava připojení k aplikacím	1-16
Vytvoření paměťových prostorů a bloků	1-16
Podpora velkých bloků	1-17
Nastavení systému zálohování a ukládání	1-17
Konfigurace vlastností relace	1-17

Konfigurace vlastností relace	1-18
Provádění rutinních úloh administrace	1-19
Změna režimů databázového serveru.	1-19
Zálohování dat a souborů logického protokolu	1-19
Monitorování aktivity	1-19
Kontrola konzistence	1-19
Provádění dalších úloh administrace	1-20
Použití zrcadlení.	1-20
Správa stavu protokolování databáze.	1-20
Správa logického protokolu	1-20
Správa fyzického protokolu	1-20
Správa sdílené paměti	1-21
Správa virtuálních procesorů	1-21
Správa paralelních databázových dotazů	1-21
Použití replikace dat	1-21
Použití auditu	1-22
Použití distribuovaných dotazů	1-22
Monitorování aktivity databázového serveru	1-23
Alarmy událostí	1-23
Program IBM Informix Server Administrator (ISA)	1-23
Protokol zpráv	1-23
Obslužný program ON-Monitor (UNIX)	1-24
Obslužný program oncheck	1-24
Nástroj onperf (UNIX)	1-24
Obslužný program onstat	1-25
Tabulky SMI.	1-25
Systémová konzole	1-25
Nástroje operačního systému UNIX	1-25
Prohlížeč událostí v systému Windows	1-26
Sledování výkonu v systému Windows	1-26
Obslužné programy pro systém Windows	1-26
Nástroj OpenAdmin pro server IDS	1-27

Kapitola 2. Konfigurační parametry 2-1

Obsah kapitoly	2-1
Parametry identifikace databázového serveru	2-1
Parametry diskového prostoru	2-2
Kořenový prostor dbspace	2-2
Zrcadlení kořenového prostoru dbspace	2-2
Další parametry správy prostorů	2-3
Parametry protokolování	2-3
Logický protokol	2-3
Parametry fyzického protokolu	2-3
Parametry odvolávání transakcí a obnovy	2-4
Parametry zálohování a obnovy	2-4
Parametry protokolu zpráv	2-4
Parametry sdílené paměti	2-5
Velikost přidělené sdílené paměti	2-5
Přidělený prostor sdílené paměti	2-5
Řízení vyrovnávací sdílené paměti.	2-6
Využití mezipaměti příkazů SQL	2-6
Parametry podpory rozhodování	2-6
Parametry procesu databázového serveru	2-7
Parametry virtuálních procesorů	2-7
Časové intervaly	2-8
Parametry obnovení	2-8
Parametry replikace High-Availability Data-Replication	2-8
Parametry alarmů událostí	2-8
Parametry vypisování jádra (UNIX)	2-9
Parametry direktiv	2-9
Parametry připojení	2-9

Parametry související se zabezpečením	2-10
Zvláštní parametry	2-10
Parametry prověřování (UNIX)	2-10
Parametry optických médií	2-11
Parametry systému UNIX	2-11
Monitorování informací o konfiguraci	2-11

Kapitola 3. Komunikace mezi klientem a serverem 3-1

Obsah kapitoly	3-2
Architektura klient-server	3-2
Síťový protokol	3-2
Rozhraní pro síťové programování	3-3
Doména sítě systému Windows	3-3
Připojení k databázovému serveru	3-4
Multiplexní připojení	3-4
Připojení podporovaná databázovým serverem	3-5
Místní připojení	3-6
Připojení prostřednictvím sdílené paměti (systém UNIX)	3-6
Proudové propojení procesů (systém UNIX)	3-7
Proudové propojení procesů (systém Linux)	3-8
Připojení prostřednictvím pojmenovaného propojení procesů (systém Windows)	3-8
Připojení prostřednictvím místní zpětné smyčky	3-8
Služby pro podporu komunikace	3-8
Propojovací soubory	3-9
Soubory konfigurace sítě	3-9
Soubory zabezpečení sítě	3-12
Soubor sqlhosts a klíč registru SQLHOSTS	3-13
Informace sqlhosts	3-15
Standardní názvy služeb a čísla portů IANA v souboru sqlhosts.std	3-16
Informace o propojitelnosti	3-17
Informace o skupinách	3-27
Alternativy k připojení TCP/IP	3-27
Podpora adres IPv6 v serveru Dynamic Server	3-30
Parametry souboru ONCONFIG týkající se propojitelnosti	3-31
Konfigurační parametr DBSERVERNAME	3-31
Konfigurační parametr DBSERVERALIASES	3-32
Konfigurační parametr NETTYPE	3-32
Proměnné prostředí týkající se síťových připojení	3-33
Komunikační prostředky DRDA (Distributed Relational Database Architecture)	3-33
Přehledné informace o protokolu DRDA a předpoklady	3-33
Konfigurace serveru Dynamic Server pro připojení ke klientům datových serverů IBM	3-34
Přidělení jednotkových procesů dotazování pro kombinaci rozhraní a protokolu s konfiguračním parametrem NETTYPE	3-35
Určení velikosti vyrovnávací paměti komunikace DRDA pomocí konfiguračního parametru DRDA_COMMBUFSIZE	3-35
Jednotkový proces DRDAEXEC a dotazy klientů	3-36
Jazyk SQL a podporované a nepodporované datové typy	3-36
Zobrazení některých informací o připojení DRDA	3-37
Příklady konfigurací klientu a serveru	3-37
Použití připojení prostřednictvím sdílené paměti (systém UNIX)	3-37
Připojení prostřednictvím místní zpětné smyčky	3-38
Použití síťového připojení	3-39
Použití více typů připojení	3-40
Přístup k více databázovým serverům	3-41
Použití programu IBM Informix MaxConnect	3-42

Kapitola 4. Inicializace databázového serveru 4-1

Obsah kapitoly	4-1
Typy inicializace	4-1
Inicializace diskového prostoru	4-2
Jednotlivé kroky inicializace	4-3
Zpracování konfiguračního souboru	4-3

Vytvoření segmentů sdílené paměti	4-4
Inicializace a restartování sdílené paměti.	4-5
Inicializace diskového prostoru.	4-5
Spuštění všech požadovaných virtuálních procesorů	4-5
Provedení potřebných konverzí.	4-5
Zahájení rychlé obnovy	4-5
Zahájení kontrolního bodu	4-5
Zaznamenání změn konfigurace	4-6
Vytvoření souboru oncfg_servername.servernum	4-6
Vypuštění dočasných prostorů tblspace	4-6
Nastavení vynucené rezidence, je-li určena	4-6
Návrat řízení uživateli	4-6
Vytvoření databáze sysmaster a příprava tabulek SMI	4-7
Vytvoření databáze sysutils	4-7
Monitorování maximálního počtu uživatelských připojení	4-7
Provozní režimy databázového serveru	4-7
Změna provozních režimů databázového serveru	4-9
Uživatelé oprávnění měnit režimy	4-9
Volby programu ISA pro změnu provozních režimů	4-10
Volby programu ON-Monitor pro změnu režimů (UNIX)	4-10
Volby příkazového řádku pro změnu režimů	4-11
Určení uživatelů režimu administrace pomocí konfiguračního parametru ADMIN_MODE_USERS	4-15

Část 2. Správa disků, paměti a procesů

Kapitola 5. Virtuální procesory a jednotkové procesy 5-1

Obsah kapitoly	5-2
Virtuální procesory	5-2
Jednotkové procesy	5-2
Typy virtuálních procesorů	5-3
Výhody virtuálních procesorů	5-5
Obsluha jednotkových procesů virtuálními procesory	5-8
Řídící struktury	5-8
Přepínání kontextu	5-8
Zásobníky	5-9
Fronty.	5-10
Objekty mutex	5-12
Třídy virtuálních procesorů	5-12
Virtuální procesory CPU	5-12
Uživatelem definované třídy virtuálních procesorů	5-15
Virtuální procesory Java	5-17
Virtuální procesory diskového vstupu - výstupu	5-17
Síťové virtuální procesory	5-22
Virtuální procesor Communications Support Module	5-27
Šifrovací virtuální procesory	5-27
Optický virtuální procesor	5-28
Virtuální procesor typu audit	5-28
Virtuální procesor typu miscellaneous	5-28

Kapitola 6. Správa virtuálních procesorů 6-1

Obsah kapitoly	6-1
Nastavení konfiguračních parametrů virtuálních procesorů	6-1
Nastavení parametrů virtuálního procesoru pomocí textového editoru	6-1
Nastavení parametrů virtuálního procesoru pomocí programu ISA	6-2
Nastavení parametrů virtuálního procesoru pomocí obslužného programu ON-Monitor	6-3
Spuštění a zastavení virtuálních procesorů	6-3
Přidání virtuálních procesorů v režimu online	6-3
Vypuštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů	6-4
Monitorování virtuálních procesorů	6-5
Monitorování virtuálních procesorů pomocí obslužných programů příkazového řádku	6-5

Monitorování virtuálních procesorů pomocí tabulek SMI	6-6
Kapitola 7. Sdílená paměť	7-1
Obsah kapitoly	7-2
Sdílená paměť	7-2
Použití sdílené paměti	7-2
Přidělování sdílené paměti	7-3
Velikost sdílené paměti	7-4
Jaké akce provést, pokud je překročena hodnota parametru SHMTOTAL	7-5
Procesy, které se připojují ke sdílené paměti	7-5
Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)	7-6
Jak se ke sdílené paměti připojují obslužné programy	7-6
Jak se ke sdílené paměti připojují virtuální procesory	7-6
Rezidentní segmenty sdílené paměti	7-9
Rezidentní část sdílené paměti	7-9
Záhlaví sdílené paměti	7-10
Společná oblast vyrovnávacích paměti ve sdílené paměti.	7-10
Vyrovnávací paměť logického protokolu	7-11
Vyrovnávací paměť fyzického protokolu	7-12
Vyrovnávací paměť replikace High-Availability Data Replication	7-13
Tabulka zámků	7-13
Virtuální část sdílené paměti	7-14
Správa virtuální části sdílené paměti	7-14
Součásti virtuální části sdílené paměti	7-15
Mezipaměť distribuce dat	7-19
Komunikační část sdílené paměti (systém UNIX)	7-20
Část virtuálních rozšíření sdílené paměti	7-21
Řízení souběžného přístupu	7-21
Objekty mutex sdílené paměti.	7-21
Zámky vyrovnávacích paměti ve sdílené paměti	7-21
Přístup jednotkových procesů databázového serveru k sdíleným vyrovnávacím pamětem	7-22
Fronty FIFO/LRU	7-22
Konfigurace dopředného čtení databázového serveru	7-26
Přístup jednotkových procesů databázového serveru ke stránkám ve vyrovnávací paměti	7-26
Vyprazdňování dat na disk.	7-26
Vyprazdňování vyrovnávacích paměti ve společné oblasti	7-27
Přednostní vyprazdňování předobrazů	7-27
Vyprazdňování vyrovnávací paměti fyzického protokolu.	7-27
Synchronizace vyprazdňování vyrovnávacích paměti	7-28
Popis aktivity vyprazdňování	7-28
Vyprazdňování vyrovnávací paměti logického protokolu.	7-29
Použití vyrovnávací paměti pro velké objekty	7-30
Zápis jednoduchých velkých objektů.	7-30
Přístup k inteligentním velkým objektům	7-32
Využití paměti na 64bitových platformách	7-33
Kapitola 8. Správa sdílené paměti	8-1
Obsah kapitoly	8-1
Nastavení konfiguračních parametrů sdílené paměti operačního systému	8-2
Maximální velikost segmentu sdílené paměti	8-2
Semafore (systém UNIX)	8-3
Nastavení konfiguračních parametrů sdílené paměti databázového serveru	8-3
Nastavení parametrů pro rezidentní sdílenou paměť.	8-3
Nastavení parametrů pro virtuální sdílenou paměť	8-4
Nastavení parametrů pro výkon sdílené paměti	8-4
Nastavení parametrů sdílené paměti pomocí textového editoru	8-5
Nastavení parametrů sdílené paměti pomocí programu ISA	8-5
Nastavení parametrů sdílené paměti pomocí programu ON-Monitor (v systému UNIX)	8-5
Nastavení parametrů mezipaměti příkazů jazyka SQL	8-6
Nastavení sdílené paměti	8-6

Vypnutí nebo zapnutí rezidence rezidentní sdílené paměti	8-6
Zapínání a vypínání rezidence v režimu online	8-7
Zapnutí a vypnutí rezidence při restartování databázového serveru	8-7
Přidání segmentu k virtuální části sdílené paměti	8-7
Monitorování sdílené paměti	8-7
Monitorování segmentů sdílené paměti	8-7
Monitorování profilu sdílené paměti a zámek latch	8-8
Monitorování vyrovnávacích pamětí	8-8
Monitorování aktivity společné oblasti vyrovnávací paměti	8-10
Kapitola 9. Uložení dat	9-1
Obsah kapitoly	9-2
Fyzické a logické paměťové jednotky	9-2
Bloky	9-3
Přidělování disků blokům	9-3
Posuny	9-5
Stránky	9-5
Stránky blobpage	9-6
Stránky sbpage	9-7
Oblasti	9-8
Prostory dbspace	9-9
Řízení umístění ukládaných dat	9-9
Kořenový prostor dbspace	9-11
Dočasné prostory dbspace	9-11
Prostory blobspace	9-12
Prostory sbspace	9-12
Výhody používání prostorů sbspace	9-13
Prostory sbspace a replikace Enterprise Replication	9-13
Metadata, uživatelská data a rezervovaná oblast	9-13
Řízení umístění ukládaných dat	9-14
Paměťové charakteristiky prostorů sbspace	9-15
Úroveň dědičnosti charakteristik prostorů sbspace	9-17
Další informace o prostorech sbspace	9-18
Dočasné prostory sbspace	9-19
Porovnání dočasných a standardních prostorů sbspace	9-20
Dočasné inteligentní velké objekty	9-20
Prostory extspace	9-21
Databáze	9-21
Tabulky	9-22
Poškozené tabulky	9-23
Typy tabulek serveru Dynamic Server	9-24
Standardní trvalé tabulky	9-24
Tabulky typu RAW	9-24
Tabulky typu TEMP	9-25
Vlastnosti typů tabulek	9-25
Dočasné tabulky	9-26
Prostory tblspace	9-28
Maximální počet prostorů tblspace v tabulce	9-29
Prostory tblspace tabulek a indexů	9-29
Prokládání oblastí	9-30
Fragmentace tabulek a uložení dat	9-30
Objem diskového prostoru potřebný k uložení dat	9-31
Velikost kořenového prostoru dbspace	9-32
Objem prostoru požadovaného databázemi	9-33
Pravidla pro rozvržení disků	9-33
Pravidla pro prostory dbspace a bloky	9-34
Pravidla umísťování tabulek	9-34
Ukázková rozvržení disku	9-35
Ukázkové rozvržení upřednostňující výkon	9-36
Ukázkové rozvržení upřednostňující dostupnost	9-37
Správce logických disků	9-38

Kapitola 10. Správa diskového prostoru	10-1
Obsah kapitoly	10-2
Přidělení diskového prostoru	10-3
Určení posunu	10-4
Přidělení prostorů předpřipravených souborů v systému UNIX	10-4
Přidělení diskového prostoru s přímým přístupem v systému UNIX	10-5
Vytvoření symbolických propojení k přímým zařízením (UNIX)	10-5
Přidělení prostoru souborům systému NTFS v systému Windows	10-6
Přidělení diskového prostoru s přímým přístupem v systému Windows	10-6
Určení názvů paměťových prostorů a bloků	10-7
Určení maximální velikosti bloků	10-8
Určení maximálního počtu bloků a paměťových prostorů	10-8
Zálohování po změně fyzického schématu	10-8
Správa prostorů dbspace	10-9
Vytvoření prostoru dbspace, který používá výchozí velikost stránky	10-9
Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky	10-12
Zvýšení výkonu prostorů dbspace pro předpřipravené soubory pomocí přímého vstupu - výstupu	10-16
Vytvoření dočasného prostoru dbspace	10-17
Co dělat, pokud dojde k vyčerpání diskového prostoru	10-18
Přidání bloku do prostoru dbspace nebo blobspace	10-18
Přejmenování prostorů dbspace	10-19
Správa oddílů prostorů dbspace	10-20
Správa prostorů blobspace	10-21
Vytvoření prostoru blobspace	10-21
Příprava prostorů blobspace k ukládání dat typu TEXT a BYTE	10-23
Určení velikosti stránky blobpage	10-23
Správa prostorů sbospace	10-24
Vytvoření prostoru sbospace	10-24
Určení velikosti metadat prostoru sbospace	10-25
Přidání bloku do prostoru sbospace	10-25
Změna paměťových charakteristik inteligentních velkých objektů	10-26
Vytvoření dočasného prostoru dbspace	10-26
Vypuštění bloku	10-27
Ověření, zda je blok prázdný	10-27
Vypuštění bloku z prostoru dbspace pomocí obslužného prostoru onspaces	10-28
Vypuštění bloku z prostoru blobspace	10-28
Vypuštění bloku z prostoru sbospace příkazem onspaces	10-28
Vypuštění paměťového prostoru	10-29
Příprava na vypuštění paměťového prostoru	10-29
Vypuštění zrcadlového paměťového prostoru	10-29
Vypuštění paměťového prostoru s obslužným programem onspaces	10-29
Vypuštění prostorů dbspace a blobspace pomocí programu ON-Monitor (UNIX)	10-30
Zálohování po vypuštění paměťového prostoru	10-30
Správa prostorů extspace	10-30
Vytvoření prostoru extspace	10-30
Vypuštění prostoru extspace	10-31
Přeskakování nepřístupných fragmentů	10-31
Použití konfiguračního parametru DATASKIP	10-31
Použití funkce Dataskip obslužného programu onspaces	10-32
Použití obslužného programu onstat ke kontrole stavu funkce dataskip	10-32
Použití příkazu SET DATASKIP jazyka SQL	10-32
Vliv funkce dataskip na transakce	10-32
Jak určit, kdy používat parametr dataskip	10-33
Monitorování využití fragmentace	10-33
Zobrazení databází	10-34
Použití tabulek SMI	10-34
Použití programu ISA	10-34
Použití programu ON-Monitor (systém UNIX)	10-34
Monitorování využití disku	10-34
Monitorování bloků	10-34
Monitorování prostorů tblspace a oblastí	10-38

Monitorování jednoduchých velkých objektů v prostoru blobspace	10-38
Monitorování prostorů sbpspace	10-41
Zavedení dat do tabulky	10-46

Část 3. Protokolování a administrace protokolů

Kapitola 11. Protokolování 11-1

Obsah kapitoly	11-1
Procesy databázového serveru, které vyžadují protokolování	11-1
Protokolování transakcí	11-3
Protokolování příkazů jazyka SQL a aktivity databázového serveru	11-3
Aktivita, která se vždy protokoluje	11-3
Aktivity protokolované u databázi s protokolováním transakcí	11-4
Aktivita, která se neprotokoluje	11-4
Stav protokolování databáze	11-5
Protokolování transakcí bez vyrovnávací paměti	11-5
Protokolování transakcí s vyrovnávací pamětí	11-6
Protokolování transakcí kompatibilní se standardem ANSI	11-6
Databáze bez protokolování	11-6
Databáze s různými stavy používání vyrovnávací paměti pro protokolování	11-7
Protokolování databáze v prostředí X/Open DTP	11-7
Nastavení nebo změny stavu nebo režimu protokolování	11-7

Kapitola 12. Správa režimu protokolování databáze 12-1

Obsah kapitoly	12-1
Změna režimu protokolování databáze	12-2
Změna režimu protokolování databáze pomocí obslužného programu ondblog	12-3
Změna režimu protokolování pomocí obslužného programu ondblog	12-3
Zrušení změny režimu protokolování pomocí obslužného programu ondblog	12-3
Ukončení protokolování pomocí obslužného příkazu ondblog	12-3
Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ondblog	12-3
Změna režimu protokolování databáze kompatibilní se standardem ANSI	12-3
Změna režimu protokolování databáze pomocí obslužného programu ontape	12-4
Zapnutí protokolování transakcí pomocí obslužného programu ontape	12-4
Ukončení protokolování pomocí obslužného programu ontape	12-4
Změna režimu protokolování pomocí obslužného programu ontape	12-4
Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ontape	12-5
Změna režimu protokolování databáze pomocí programu ISA	12-5
Změna režimu protokolování pomocí programu ON-Monitor (UNIX)	12-5
Změna režimu protokolování tabulek	12-5
Změna tabulky za účelem vypnutí protokolování	12-5
Změna tabulky za účelem zapnutí protokolování	12-6
Zakázání protokolování v dočasných tabulkách	12-6
Monitorování transakcí	12-6
Monitorování režimu protokolování databáze	12-6
Monitorování režimu protokolování u tabulek typu SMI	12-6
Monitorování režimu protokolování pomocí programu ON-Monitor (UNIX)	12-6
Monitorování režimu protokolování pomocí programu ISA	12-7

Kapitola 13. Logický protokol 13-1

Obsah kapitoly	13-1
Co je logický protokol?	13-1
Umístění souborů logického protokolu	13-2
Identifikace souborů logického protokolu	13-2
Příznaky stavu souborů logického protokolu	13-3
Velikost logického protokolu	13-3
Počet souborů logického protokolu	13-4
Pokyny týkající se výkonu	13-4
Dynamické přidělování protokolů	13-5
Uvolnění souborů logického protokolu	13-5

Akce v případě, že další soubor logického protokolu není volný	13-5
Akce v případě, že následující logický soubor obsahuje poslední kontrolní bod	13-6
Protokolování prostorů blobspace a jednoduché velké objekty	13-6
Aktivace prostorů blobspace přepnutím souborů protokolu	13-7
Zálohování souborů protokolů za účelem uvolnění stránek blobpage	13-7
Zálohování prostorů blobspace po vložení nebo odstranění dat typu TEXT a BYTE	13-7
Protokolování prostorů sbospace a inteligentních velkých objektů	13-7
Používání protokolování prostoru sbospace	13-8
Používání záznamů protokolu inteligentních velkých objektů	13-9
Zabránění dlouhým transakcím při protokolování dat inteligentních velkých objektů	13-9
Proces protokolování	13-10
Protokolování prostorů dbspace	13-10
Protokolování prostorů blobspace	13-10

Kapitola 14. Správa souborů logických protokolů. 14-1

Obsah kapitoly	14-1
Odhad velikosti a počtu souborů protokolu.	14-2
Odhad velikosti protokolu při protokolování inteligentních velkých objektů	14-4
Odhad počtu souborů logického protokolu	14-4
Zálohování logických souborů	14-4
Zálohování prostorů blobspace	14-5
Zálohování prostorů sbospace	14-5
Přepnutí na následující soubor logického protokolu	14-5
Uvolnění souboru logického protokolu	14-6
Odstranění souboru protokolu se stavem D	14-6
Uvolnění souboru protokolu se stavem U	14-6
Uvolnění souboru protokolu se stavem U-B nebo F	14-6
Uvolnění souboru protokolu se stavem U-C nebo U-C-L	14-7
Uvolnění souboru protokolu se stavem U-B-L	14-7
Monitorování aktivity protokolování.	14-7
Monitorování logického protokolu za účelem zjištění jeho zaplnění	14-8
Monitorování dočasných logických protokolů	14-8
Použití tabulek SMI.	14-9
Použití programu ON-Monitor (systém UNIX).	14-9
Monitorování stavu zálohy protokolů	14-9
Přidělení souborů protokolu	14-9
Dynamické přidávání protokolů	14-9
Ruční přidání souborů logického protokolu	14-11
Vypuštění souborů logického protokolu	14-12
Změna velikosti souborů logického protokolu	14-13
Přesun souboru logického protokolu do jiného prostoru dbspace	14-14
Změna konfiguračních parametrů protokolování	14-14
Použití programu ON-Monitor ke změně parametru LOGFILES (systém UNIX)	14-15
Zobrazení záznamů logického protokolu	14-16
Monitorování událostí dynamického přidávání protokolů	14-16
Nastavení horních mezí pro odvolání dlouhých transakcí	14-17
Horní mez dlouhé transakce (LTXHWM)	14-17
Horní mez dlouhé transakce s výlučným přístupem (LTXEHW)	14-18
Úprava velikosti souborů protokolu za účelem zabránění dlouhých transakcí	14-18
Obnova zablokované dlouhé transakce.	14-18

Kapitola 15. Fyzické protokolování, kontrolní body a rychlá obnova 15-1

Obsah kapitoly	15-1
Kritická část	15-2
Fyzické protokolování	15-2
Jak rychlá obnova používá fyzicky protokolované stránky	15-2
Jak zálohování používá fyzicky protokolované stránky	15-2
Aktivity databázového serveru, které jsou fyzicky protokolovány	15-2
Velikost a umístění fyzického protokolu.	15-3
Určení umístění fyzického protokolu.	15-3

Postup k určení velikosti fyzického protokolu	15-3
Přetečení fyzického protokolu po vypnutí protokolování transakcí	15-4
Kontrolní body	15-5
Hodnoty LRU pro vyprázdňování společné oblasti vyrovnávací paměti mezi kontrolními body	15-6
Kontrolní body během zálohování	15-6
Rychlá obnova	15-7
Potřeba rychlé obnovy	15-7
Situace, při kterých se spouští rychlá obnova	15-7
Rychlá obnova po kontrolním bodu	15-8

Kapitola 16. Správa fyzického protokolu. 16-1

Obsah kapitoly	16-1
Změna umístění a velikosti fyzického protokolu	16-1
Kontrola odpovídajícího souvislého prostoru	16-2
Použití obslužného programu onparams ke změně umístění nebo velikosti fyzického protokolu	16-2
Použití programu ON-Monitor ke změně umístění a velikosti fyzického protokolu (systém UNIX)	16-2
Monitorování aktivity fyzického a logického protokolování.	16-2
Monitorování informací o kontrolních bodech.	16-3
Zapnutí a vypnutí kontrolních bodů	16-4
Vynucení kontrolního bodu	16-4
Použití statistických údajů o kontrolních bodech poskytovaných serverem	16-5
Použití tabulek SMI.	16-5
Zapnutí a vypnutí automatického ladění LRU	16-5

Část 4. Odolnost vůči selhání

Kapitola 17. Zrcadlení 17-1

Obsah kapitoly	17-1
Zrcadlení	17-1
Výhody zrcadlení	17-2
Požadavky na zrcadlení	17-2
Důsledky nepoužívání zrcadlení	17-2
Data vhodná k zrcadlení	17-2
Alternativy k zrcadlení	17-3
Průběh zrcadlení.	17-3
Vytvoření zrcadleného bloku	17-4
Stavové příznaky zrcadlení	17-4
Obnova	17-4
Činnosti v průběhu zpracování	17-5
Výsledek zastavení zrcadlení	17-6
Struktura zrcadleného bloku	17-6

Kapitola 18. Použití zrcadlení 18-1

Obsah kapitoly	18-1
Příprava k zrcadlení dat	18-1
Povolení konfiguračního parametru MIRROR	18-2
Změna parametru MIRROR pomocí programu ON-Monitor (systém UNIX)	18-2
Přidělení diskového prostoru pro zrcadlená data	18-2
Propojení bloků (UNIX)	18-3
Opětovné propojení bloku se zařízením po poruše disku	18-3
Použití zrcadlení.	18-3
Zrcadlení kořenového prostoru dbspace během spuštění	18-4
Změna stavu zrcadlení	18-4
Správa zrcadlení.	18-4
Spuštění zrcadlení u nezrcadlených paměťových prostorů	18-4
Spuštění zrcadlení nových paměťových prostorů	18-5
Přidání zrcadlených bloků	18-5
Vypnutí zrcadleného bloku	18-6
Obnovení zrcadleného bloku	18-6
Ukončení zrcadlení	18-7

Kapitola 19. Přehled replikace dat (vydání Enterprise/Workgroup).	19-1
Obsah kapitoly	19-2
Replikace dat	19-2
Režimy serveru	19-3
Typy replikovaných dat	19-3
Primární a sekundární databázové servery	19-3
Výhody replikace dat	19-4
Způsob činnosti replikace dat	19-6
Průběh prvotní replikace dat	19-6
Reprodukce aktualizací primárního databázového serveru	19-7
Jednotkové procesy, které obsluhují replikaci dat.	19-10
Kontrolní body mezi databázovými servery	19-11
Jak je zaznamenávána synchronizace dat	19-11
Příklady konfigurace replikace dat	19-12
Příklady konfigurace vzdálených samostatných serverů.	19-12
Příklady konfigurace sekundárních sdílených disků	19-14
Použití replikace Enterprise Replication jako části obnovitelné skupiny	19-18
Příklad konfigurace vysoce dostupných klastrů s replikací Enterprise Replication	19-18
Příklad komplexní strategie obnovy pro případ poruchy.	19-19
Přesměrování a propojitelnost pro klienty replikace dat.	19-23
Návrh klientů pro přesměrování.	19-24
Automatické přesměrování klientů pomocí proměnné prostředí DBPATH	19-24
Směrování klientů pomocí informací o propojitelnosti	19-25
Směrování klientů pomocí proměnné prostředí INFORMIXSERVER	19-28
Přesměrování z aplikace	19-29
Porovnání různých mechanismů přesměrování	19-30
Návrh klientů skupiny replikace dat	19-31
Nastavení režimu uzamykání při přístupu k primárnímu databázovému serveru na čekání	19-31
Návrh klientů k použití sekundárního databázového serveru	19-32
Kapitola 20. Použití replikace HDR (verze pro podniky/pracovní skupiny).	20-1
Obsah kapitoly	20-2
Plánování replikace HDR	20-2
Konfigurace systému pro replikaci HDR	20-2
Splnění požadavků na hardware a operační systém	20-3
Splnění požadavků na databázi a data	20-3
Splnění požadavků na konfiguraci databázového serveru.	20-3
Konfigurace propojitelnosti replikace HDR.	20-6
První spuštění replikace HDR.	20-6
Zkrácení doby nastavení alternativní metodou zálohování	20-9
Provádění základních úloh administrace.	20-9
Změna konfiguračních parametrů databázového serveru	20-9
Zálohování paměťových prostorů a souborů logického protokolu	20-10
Změna režimu protokolování databázi	20-10
Přidání a vypuštění bloků a paměťových prostorů	20-10
Přejmenování bloků	20-10
Uložení stavu bloku na sekundární databázový server	20-11
Použití a změna zrcadlení bloků	20-11
Správa fyzického protokolu	20-12
Správa logického protokolu	20-12
Správa virtuálních procesorů.	20-12
Správa sdílené paměti.	20-12
Replikace indexu sekundárního databázového serveru	20-13
Šifrování přenosu dat mezi databázovými servery HDR.	20-14
Úprava vyprázdnění LRU a automatické ladění v serverových párech HDR	20-15
Změna režimu databázového serveru	20-15
Změna typu databázového serveru	20-16
Monitorování stavu replikace HDR.	20-17
Použití obslužných programů příkazového řádku.	20-17
Použití tabulek SMI	20-18
Použití programu ON-Monitor (systém UNIX)	20-18

Selhání replikace HDR	20-18
Definice selhání replikace HDR	20-18
Zjišťování selhání replikace HDR	20-19
Akce, které je třeba provést při selhání replikace HDR	20-19
Co provést v případě selhání replikace HDR	20-19
Obnovení dat po selhání prostředku	20-22
Obnovení po selhání prostředku na primárním databázovém serveru	20-22
Obnovení po selhání prostředku na sekundárním databázovém serveru	20-23
Replikace indexu na sekundární server	20-23
Restartování replikace HDR po selhání	20-24
Restartování po poškození kritických dat	20-24
Restartování bez poškození kritických dat	20-25

Kapitola 21. Použití serverů RS (remote standalone) (vydání Enterprise) 21-1

Obsah kapitoly	21-1
Sekundární server RS	21-1
Rozdíly mezi sekundárním serverem RS a replikací HDR	21-2
Kdy používat sekundární Server RS	21-2
Charakteristika a omezení sekundárních serverů RS	21-3
Protokolování indexových stránek	21-3
Připojení rozhraní SMX (Server Multiplexer Group)	21-4
Konfigurace sekundárního serveru RS	21-4
Konfigurace sekundárního serveru RS	21-7
Obnovení při havárii v prostředí sekundárního serveru RS	21-8
Zabezpečení sekundárního serveru RS	21-8
Návrh klientů sekundárních serverů RS	21-8
Monitorování alarmů událostí sekundárního serveru RS	21-8

Kapitola 22. Použití sekundárních serverů se sdíleným diskem (vydání Enterprise) 22-1

Obsah kapitoly	22-1
Sekundární server SD	22-1
Kdy používat sekundární Server SD	22-2
Charakteristika a omezení prostředí sekundárního serveru SD	22-2
Požadavky na konfiguraci sekundárního serveru SD	22-2
Obnovení při havárii v prostředích sekundárního serveru SD	22-3
Získání statistiky sekundárního serveru SD	22-4
Konfigurace sekundárního serveru SD	22-4
Obnovení při havárii v prostředí sekundárního serveru SD	22-4
Zabezpečení sekundárního serveru SD	22-5
Monitorování alarmů událostí sekundárního serveru SD	22-6

Kapitola 23. Kontrola konzistence 23-1

Obsah kapitoly	23-1
Provádění periodické kontroly konzistence	23-2
Ověření konzistence	23-2
Monitorování nekonzistence dat	23-3
Zachování konzistentních záloh úrovně 0	23-5
Co provést v případě poškození	23-5
Zjištění příznaků poškození	23-5
Oprava poškození indexu	23-5
Oprava chyb vstupu - výstupu bloku	23-5
Shromažďování diagnostických dat	23-6
Chyby vstupu - výstupu bránící v provozu	23-7
Monitorování chyb vstupu - výstupu bránících v provozu v databázovém serveru	23-8
Používání protokolu zpráv ke monitorování chyb vstupu - výstupu bránících v provozu	23-8
Používání alarmů událostí k monitorování chyb vstupu - výstupu bránících v provozu	23-8
Používání vypnutého mapování chybných sektorů	23-9

Část 5. Distribuovaná data

Kapitola 24. Protokoly vícefázového potvrzování	24-1
Obsah kapitoly	24-2
Správci transakcí	24-2
Použití knihovny TP/XA se správcem transakcí	24-2
Použití serveru MTS/XA (Microsoft Transaction Server).	24-2
Použití podpory transakcí dynamického serveru pro externí zdroje dat vyhovující specifikaci XA.	24-3
Použití volně vázaného režimu a provázaného režimu	24-4
Protokol dvoufázového potvrzování	24-5
Kdy se používá protokol dvoufázového potvrzování	24-5
Koncepte dvoufázového potvrzování	24-5
Fáze protokolu dvoufázového potvrzování	24-6
Jak protokol dvoufázového potvrzování ošetřuje selhání	24-7
Optimalizace předpokládaného konce	24-8
Nezávislé akce	24-8
Situace, které vyvolávají nezávislé akce.	24-8
Možné důsledky nezávislých akcí	24-8
Scénář heuristického odvolání transakce	24-10
Scénář heuristického ukončení transakce	24-12
Monitorování globální transakce	24-13
Chyby protokolu dvoufázového potvrzování	24-14
Dvoufázové potvrzování a záznamy logického protokolu	24-14
Záznamy logického protokolu při potvrzení transakce	24-15
Záznamy logického protokolu zapsané během heuristického odvolání transakce	24-16
Záznamy logického protokolu zapsané během heuristického ukončení transakce	24-17
Konfigurační parametry používané při dvoufázovém potvrzování	24-18
Funkce parametru DEADLOCK_TIMEOUT	24-18
Funkce parametru TXTIMEOUT	24-19
Protokol dvoufázového potvrzování	24-19
Brány, které se mohou účastnit heterogenního potvrzování transakcí	24-20
Povolení a zakázání heterogenního potvrzování	24-20
Jak funguje heterogenní potvrzování	24-21
Důsledky selhání heterogenního potvrzování	24-22
Kapitola 25. Ruční obnova při selhání dvoufázového potvrzování	25-1
Obsah kapitoly	25-1
Určení, zda je požadována ruční obnova	25-1
Určení, zda byla transakce implementována nekonzistentně	25-1
Určení, zda distribuovaná databáze obsahuje nekonzistentní data	25-2
Rozhodnutí, zda je nezbytné k nápravě situace provést akci.	25-4
Příklad ruční obnovy	25-5
Část 6. Automatické monitorování a nápravné akce	
Kapitola 26. Automatické monitorování a nápravné akce	26-1
Přehledné informace o automatickém monitorování a o nápravných akcích.	26-1
Zvýšené požadavky na paměťový prostor	26-2
Databáze sysadmin	26-2
Tabulky databáze sysadmin	26-3
Funkce jazyka SQL rozhraní API administrace.	26-3
Tabulka command_history.	26-3
Přesun databáze sysadmin do nového databázového prostoru	26-4
Plánovač	26-5
Vytváření úloh	26-6
Úprava úloh	26-10
Vzdálená správa pomocí příkazů SQL	26-10
Použití syntaxe onmode v příkazech administračního rozhraní API	26-11
Procházení dotazů	26-11
Určení spouštěcích informací o trasování jazyka SQL pomocí konfiguračního parametru SQLTRACE	26-14
Zakázání trasování historie jazyka SQL globálně nebo pro relaci	26-15
Povolení trasování jazyka SQL u konkrétního uživatele	26-15

Povolení trasování jazyka SQL na nízké úrovni u všech uživatelů	26-16
---	-------

Část 7. Dodatky a přílohy

Dodatek. Usnadnění	A-1
Funkce usnadňující přístup v rámci produktu IBM Informix Dynamic Server	A-1
Funkce usnadnění přístupu	A-1
Navigace pomocí klávesnice	A-1
Informace související s usnadněním přístupu	A-1
IBM a usnadnění přístupu	A-1
Upozornění	B-1
Ochranné známky	B-3
Rejstřík	X-1

Úvod

Obsah úvodní kapitoly	xvii
Obsah příručky	xvii
Druhy uživatelů	xvii
Závislosti na softwaru	xviii
Předpoklady pro národní prostředí	xviii
Demonstrační databáze.	xviii
Nové vlastnosti serveru Dynamic Server, verze 11.10.	xviii
Konvence používané v dokumentaci	xix
Typografické konvence	xix
Značky vlastností, produktů a platforem	xx
Konvence použité v ukázkovém kódu	xx
Další dokumentace	xxi
Kompatibilita s oborovými standardy	xxi
IBM ocení veškeré připomínky.	xxi

Obsah úvodní kapitoly

Tato úvodní kapitola shrnuje obsah této příručky a popisuje konvence, které v ní jsou používány.

Obsah příručky

Tato příručka popisuje koncepty a postupy konfigurace, administrace a používání serverů IBM Informix Dynamic Server a IBM Informix Dynamic Server with J/Foundation.

Doplňková příručka *IBM Informix Dynamic Server Administrator's Reference* obsahuje referenční materiály k používání databázových serverů Informix. Pokud potřebujete ladit výkon databázového serveru a příkazů jazyka SQL, naleznete informace v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Druhy uživatelů

Tato příručka je určena pro následující uživatele:

- uživatele databáze
- administrátory databáze
- administrátory databázového serveru
- techniky pro záležitosti výkonu
- programátory v následujících oblastech:
 - vývojáře aplikací
 - vývojáře modulů DataBlade
 - uživatele vytvářející uživatelské rutiny

Tato příručka je určena především uživatelům s těmito znalostmi:

- Práce s počítačem, operačním systémem a jeho obslužnými programy.
- Částečná znalost práce s relačními databázemi nebo obecná znalost problematiky databází.
- Částečná znalost programování.
- Částečná znalost administrace databázového serveru, operačního systému či sítě.

Podrobnější informace o relačních databázích, jazyku SQL a použitém operačním systému naleznete v příručce *Úvodní příručka serveru IBM Informix Dynamic Server* databázového serveru.

Závislosti na softwaru

Tato příručka je určena uživatelům serveru IBM Informix Dynamic Server verze 11.10.

Předpoklady pro národní prostředí

Produkty IBM Informix podporují mnoho jazyků, národností a znakových sad. Veškeré informace o znakových sadách, třídění a reprezentaci číselných dat, měn, data a času jsou obsaženy v jediném prostředí nazvaném národní prostředí Global Language Support (GLS).

Příklady v této příručce předpokládají použití výchozího národního prostředí **en_us.8859-1**. Toto národní prostředí podporuje formát data, času a měn podle konvencí americké angličtiny. Navíc toto národní prostředí podporuje znakovou sadu ISO 8859-1 zahrnující znakovou sadu ASCII a mnoho dalších 8b znaků, například é, è a ñ.

Použití znaků, které nejsou obsaženy ve výchozí znakové sadě, v datech či identifikátorech jazyka SQL nebo použití jiných pravidel pro třídění je možné až po nastavení příslušného národního prostředí.

Informace o nastavení jiného národního prostředí, příslušné syntaxi a dalších záležitostech týkajících se národního prostředí GLS naleznete v příručce *IBM Informix GLS User's Guide*.

Demonstrační databáze

Obslužný program DB–Access dodávaný s databázovým serverem Informix obsahuje jednu nebo více následujících demonstračních databází:

- Databáze **stores_demo** ilustruje použití relačního schématu obsahujícího údaje o fiktivním velkoobchodu se sportovním zbožím. Mnoho příkladů obsažených v příručkách IBM Informix je založeno na databázi **stores_demo**.
- Databáze **superstores_demo** ilustruje použití objektového relačního schématu. Databáze **superstores_demo** obsahuje příklady rozšířených datových typů, dědičnosti typů a tabulek a uživatelských rutin.

Informace o vytváření a naplnění demonstračních databází naleznete v příručce *IBM Informix DB–Access User's Guide*. Popis těchto databází a jejich obsahu naleznete v příručce *IBM Informix Guide to SQL: Reference*.

Skripty, které slouží k instalaci ukázkových databází, jsou uloženy v adresáři **\$INFORMIXDIR/bin** v systému UNIX a v adresáři **%INFORMIXDIR%\bin** v systému Windows.

Nové vlastnosti serveru Dynamic Server, verze 11.10

Úplný seznam nových vlastností této verze naleznete v příručce *Úvodní příručka serveru IBM Informix Dynamic Server*. Toto téma uvádí nové vlastnosti, které odpovídají této příručce.

Verze 11.10 databázového serveru obsahuje nové vlastnosti, které vylepšují funkčnost databázového serveru a usnadňují jeho používání.

Nové vlastnosti	Odkaz
Šifrování přenosu dat mezi servery ve dvojici replikací HDR	Šifrování přenosu dat mezi databázovými servery HDR

Nové vlastnosti	Odkaz
Konfigurační parametr pro nastavení času (v sekundách), který má dynamický server na obnovu po restartování serveru a přechodu do režimu online nebo do klidového režimu.	“Kontrolní body” na stránce 15-5
Podpora blokování omezených transakcí prostřednictvím neblokujících kontrolních bodů. Umožňuje aplikaci zpracovat transakce, zatímco je zpracováván kontrolní bod. Tato vlastnost eliminuje fuzzy kontrolní body. V předchozích verzích dynamického serveru bylo vyprázdnění LRU nastaveno jako velmi agresivní, aby snížilo blokování času kontrolního bodu. Nyní může být vyprázdnění LRU méně agresivní.	“Kontrolní body” na stránce 15-5
Rozhraní SQL Administration API, které umožňuje vzdálené provádění úloh administrace, vydáváním příkazů SQL.	Kapitola 26, “Automatické monitorování a nápravné akce”, na stránce 26-1
Plánovač, který umožňuje spravovat a spouštět úlohy údržby, monitorování a administrace v předem definovaném čase nebo podle času interně určeného serverem.	Kapitola 26, “Automatické monitorování a nápravné akce”, na stránce 26-1
Funkčnost rozbalení podrobností dotazu, která umožňuje sledovat výkon naposledy provedených příkazů SQL.	Kapitola 26, “Automatické monitorování a nápravné akce”, na stránce 26-1
Konfigurační parametr, který umožňuje určit, zda databázový server používá naposledy potvrzenou verzi dat.	“Tabulka zámeků” na stránce 7-13
Dvě vestavěné uložené procedury, které dovolují měnit vlastnosti relace při připojení nebo v době přístupu, aniž byste změnili aplikaci, kterou tato relace spouští.	“Konfigurace vlastností relace” na stránce 1-18
Funkčnost pro povolení komunikace mezi klienty serveru Dynamic Server a serveru IBM Data Server pomocí protokolu Distributed Relational Database Architecture (DRDA).	“Komunikační prostředky DRDA (Distributed Relational Database Architecture)” na stránce 3-33
Funkčnost umožňující uživateli informix nebo správci databázového serveru povolit několika specifickým uživatelům připojení k databázovému serveru v režimu administrace, novém operačním režimu databázového serveru.	“Provozní režimy databázového serveru” na stránce 4-7
Nový konfigurační parametr, který řídí použití přímého vstupu - výstupu pro předpřipravené soubory používané pro bloky prostorů dbspace, a umožňuje tak zvýšit výkon prostorů dbspace předpřipravených souborů.	“Zvýšení výkonu prostorů dbspace pro předpřipravené soubory pomocí přímého vstupu - výstupu” na stránce 10-16

Konvence používané v dokumentaci

Části popisují konvence použité v produktové dokumentaci k serveru IBM Informix Dynamic Server:

- typografické konvence
- konvence týkající se funkcí, produktů a platforem
- diagramy syntaxe
- konvence příkazového řádku
- konvence kódu příkladů

Typografické konvence

Při zavádění nových termínů, znázorňování obsahu obrazovky, popisu syntaxe příkazů apod. používá tato příručka následující konvence.

Konvence	Význam
KLÍČOVÉ_SLOVO	Klíčová slova jazyků SQL, SPL a některých dalších programovacích jazyků jsou sázena velkými písmeny v patkovém písmu.
<i>kurzíva</i>	V rámci textu jsou nové termíny sázeny kurzívou. V příkladech syntaxe či kódu jsou kurzívou sázeny hodnoty proměnných, které mají být zadány uživatelem.
tučné písmo	Tučným písmem jsou sázeny součásti programů (například třídy, události či tabulky), proměnné prostředí, názvy souborů, cest a prvky rozhraní (například ikony, položky v nabídce či tlačítka).
bezpatkové písmo	Bezpatkovým písmem je sázen text zobrazený daným produktem a text zadáný uživatelem.
KLÁVESKA	Velkými písmeny písma sans serif jsou sázeny klávesy, které by měly být stisknuty uživatelem.
>	Tento symbol označuje položku v nabídce. Například zápis “Zvolte Nástroje> Možnosti ” znamená: Zvolte položku Možnosti v nabídce Nástroje .

Značky vlastností, produktů a platforem

Značky vlastností, produktů a platforem označují odstavce obsahující informace, které se vztahují pouze k danému objektu. Příklady těchto značek:

Dynamic Server

Označuje informace týkající se pouze serveru IBM Informix Dynamic Server

Konec Dynamic Server

Jen pro Windows

Označuje informace týkající se pouze prostředí OS Windows

Konec Jen pro Windows

Toto označení se může vztahovat k jednomu či více odstavcům v rámci jedné části. Pokud se k určitému produktu či platformě váže celá část, je to příslušným textem vyznačeno v jejím záhlaví. Například:

Řazení tabulky (Windows)

Konvence použité v ukázkovém kódu

V celé příručce se vyskytují příklady kódu SQL. Pokud není uvedeno jinak, není daný kód specifický pro žádný konkrétní aplikační vývojový nástroj prostředí IBM Informix.

Příkazy SQL nejsou oddělovány středníky pouze v příkladech. Setkáte se například s následujícím příkladem:

```
CONNECT TO stores_demo
...

DELETE FROM customer
  WHERE customer_num = 121
...

COMMIT WORK
DISCONNECT CURRENT
```

Tento kód SQL je nutné pro každý produkt upravit podle příslušných pravidel syntaxe. Například při použití produktu DB–Access je nutné oddělit příkazy středníky. Při použití rozhraní SQL API je nutné před každý příkaz předřadit EXEC SQL a navíc příkaz zakončit středníkem (či jiným příslušným oddělovačem).

Rada: Tři tečky v kódu příkladu znamenají, že text příkladu není kompletní a při použití je nutné jej doplnit. Všechny části důležité pro objasnění daného tématu jsou však v příkladu obsaženy.

Podrobné pokyny pro používání příkazů SQL v jednotlivých aplikačních vývojových nástrojích nebo rozhraních SQL API naleznete v příručce pro příslušný produkt.

Další dokumentace

Produktovou dokumentaci si můžete prohlížet, prohledávat a tisknout z Informačního centra IBM Informix Dynamic Server na webových stránkách <http://publib.boulder.ibm.com/infocenter/idshelp/v111/index.jsp>.

Další dokumentaci k serveru IBM Informix Dynamic Server a s ním souvisejících produktů (včetně poznámek k vydání, stroji a dokumentaci - release notes, machine notes, documentation notes - naleznete v online knihovně k produktu na webových stránkách <http://www.ibm.com/software/data/informix/pubs/library/>. Produktovou dokumentaci můžete rovněž najít na disku CD Quick Start, které je součástí dodávky produktu (produktovou dokumentaci možno i instalovat).

Kompatibilita s oborovými standardy

Organizace American National Standards Institute (ANSI) a International Organization of Standardization (ISO) společně ustanovily sadu oborových standardů jazyka Structured Query Language (SQL). Produkty IBM Informix založené na jazyku SQL jsou plně kompatibilní se standardem SQL-92 Entry Level (vydáno jako ANSI X3.135-1992). Tento standard je shodný se standardem ISO 9075:1992. Navíc je mnoho vlastností databázových serverů IBM Informix kompatibilních se standardy SQL-92 Intermediate Level a Full Level a se standardy X/Open SQL Common Applications Environment (CAE).

IBM ocení veškeré připomínky

Velice si vážíme vašich připomínek, každé opravy nebo objasnění, které považujete v našich příručkách za užitečné a které nám pomohou vylepšit budoucí verze. Tyto připomínky by měly obsahovat:

- název a verzi příručky, kterou používáte
- číslo části a strany
- vaše návrhy týkající se obsahu příručky

Své připomínky nám zašlete na následující e-mailovou adresu:

docinf@us.ibm.com

Tato e-mailová adresa je rezervována pro informování o chybách a opomenutích v dokumentaci. S technickými problémy se obraťte na oddělení technické podpory IBM. Pokyny naleznete na webu technické podpory IBM Informix Technical Support na webové stránce <http://www.ibm.com/planetwide/>.

Veškeré návrhy jsou vítány.

Část 1. Databázový server

Kapitola 1. Instalace a konfigurace databázového serveru

Obsah kapitoly	1-2
Plán pro databázový server	1-3
Posouzení priorit	1-3
Posouzení prostředí	1-3
Konfigurace operačního systému	1-4
Konfigurace paměti v systému Windows	1-4
Maximální adresový prostor	1-4
Modifikace parametrů jádra systému UNIX	1-4
Konfigurace diskového prostoru	1-5
Použití velkých bloků	1-5
Vytvoření souborů bloků v systému UNIX	1-5
Přímý přístup k disku neboli přístup bez vyrovnávací paměti	1-5
Předpřipravené soubory	1-6
Vytvoření oddílů NTFS ve Windows	1-6
Nastavení oprávnění, vlastnictví a skupiny	1-6
Vytvoření standardních názvů zařízení (UNIX)	1-6
Nastavení proměnných prostředí	1-7
Nastavení proměnných prostředí GLS	1-8
Nastavení proměnných prostředí v systému UNIX	1-8
Nastavení proměnných prostředí v systému Windows	1-9
Konfigurace propojitelnosti	1-9
Soubor sqlhosts v systému UNIX	1-10
Soubory konfigurace sítě	1-10
Soubory zabezpečení sítě	1-10
Registr sqlhosts v systému Windows	1-10
Konfigurace propojitelnosti pomocí programu ISA	1-10
Konfigurace databázového serveru	1-11
Příprava konfiguračního souboru ONCONFIG	1-11
Vytvoření souboru ONCONFIG v systému UNIX	1-11
Vytvoření souboru ONCONFIG v systému Windows	1-11
Použití funkce Nastavení serveru obslužného programu ISA k vlastnímu nastavení konfigurace	1-12
Použití obslužného programu IBM Informix Server Administrator k aktualizaci souboru ONCONFIG	1-12
Použití programu Instance Manager k vytvoření nové instance databázového serveru (Windows)	1-12
Použití programu Instance Manager k přejmenování instance databázového serveru (Windows)	1-13
Konfigurace podpory jazyka Java	1-13
Spuštění a administrace databázového serveru	1-13
Spuštění databázového serveru a inicializace diskového prostoru	1-14
Inicializace pouze při spuštění nového databázového serveru	1-14
Příprava pro automatické spuštění	1-14
Příprava automatického spuštění v systému Windows	1-14
Příprava spouštěcích a ukončovacích skriptů pro systém UNIX	1-15
Příprava připojení k aplikacím	1-16
Vytvoření paměťových prostorů a bloků	1-16
Podpora velkých bloků	1-17
Nastavení systému zálohování a ukládání	1-17
Nastavení obslužného programu ontape	1-17
Nastavení správce paměti a programu ON-Bar	1-17
Konfigurace vlastností relace	1-17
Konfigurace vlastností relace	1-18
Provádění rutinních úloh administrace	1-19
Změna režimů databázového serveru	1-19
Zálohování dat a souborů logického protokolu	1-19
Monitorování aktivity	1-19
Kontrola konzistence	1-19
Provádění dalších úloh administrace	1-20

Použití zrcadlení	1-20
Správa stavu protokolování databáze.	1-20
Správa logického protokolu	1-20
Správa fyzického protokolu	1-20
Správa sdílené paměti	1-21
Správa virtuálních procesorů	1-21
Správa paralelních databázových dotazů	1-21
Použití replikace dat	1-21
Prostředí replikace dat	1-22
Replikace ER (Enterprise Replication)	1-22
Použití auditu	1-22
Použití distribuovaných dotazů	1-22
Použití globálních transakcí	1-22
Použití správce transakcí	1-23
Monitorování aktivity databázového serveru	1-23
Alarmy událostí	1-23
Program IBM Informix Server Administrator (ISA)	1-23
Protokol zpráv	1-23
Určení cíle pro zprávy protokolu zpráv	1-24
Monitorování protokolu zpráv	1-24
Obslužný program ON-Monitor (UNIX)	1-24
Obslužný program oncheck	1-24
Nástroj onperf (UNIX)	1-24
Obslužný program onstat	1-25
Tabulky SMI.	1-25
Systémová konzole	1-25
Nástroje operačního systému UNIX	1-25
Prohlížeč událostí v systému Windows	1-26
Sledování výkonu v systému Windows	1-26
Obslužné programy pro systém Windows	1-26
Nástroj OpenAdmin pro server IDS	1-27

Obsah kapitoly

Při instalaci verze 10.0 systému IBM Informix Dynamic Server byste měli dodržovat pokyny k instalaci, aby byla oprávnění všech klíčových souborů a adresářů nastavena správně. Pokyny k instalaci naleznete v částech *IBM Informix Dynamic Server Installation Guide for UNIX and Linux* a *IBM Informix Dynamic Server Installation Guide for Microsoft Windows*.

Po instalaci nové verze databázového serveru je nutné ho konfigurovat.

Konfigurace je souhrnným označením pro nastavení specifických parametrů, které upraví databázový server pro specifické prostředí zpracování dat: množství a typ dat, počet tabulek, hardware, počet uživatelů a potřeby zabezpečení.

Následující části této kapitoly obsahují informace o základní konfiguraci a úlohách prováděných při spuštění serveru:

- “Plán pro databázový server” na stránce 1-3
- “Konfigurace operačního systému” na stránce 1-4
- “Konfigurace diskového prostoru” na stránce 1-5
- “Nastavení proměnných prostředí” na stránce 1-7
- “Konfigurace propojitelnosti” na stránce 1-9
- “Konfigurace databázového serveru” na stránce 1-11
- “Spuštění a administrace databázového serveru” na stránce 1-13
- “Provádění rutinních úloh administrace” na stránce 1-19
- “Provádění dalších úloh administrace” na stránce 1-20

- “Monitorování aktivity databázového serveru” na stránce 1-23
- “Konfigurace vlastností relace” na stránce 1-17

Tato kapitola obsahuje také informace o nástroji OpenAdmin pro server IDS, který poskytuje možnost spravovat několik instancí databázových serverů z jednoho umístění. Více informací naleznete v části “Nástroj OpenAdmin pro server IDS” na stránce 1-27.

Pokud databázový server spouštíte na počítači se systémem UNIX nebo Linux, obslužné programy serveru kontrolují, zda je prostředí zabezpečené. Další informace naleznete v příručce *IBM Informix Security Guide*

Plán pro databázový server

Konfigurace systému pro správu databáze vyžaduje řadu rozhodnutí, například kam ukládat data, jak k datům přistupovat a jak je chránit. Způsob instalace a konfigurace serveru Dynamic Server významně ovlivní výkon databázových operací.

Databázový server lze individuálně nastavit tak, aby pracoval optimálně v konkrétním prostředí zpracování dat. Aplikace databázového serveru k obsluze 1000 uživatelů provádějících časté rychlé transakce se kupříkladu velmi liší od použití databázového serveru k obsluze jen několika uživatelů provádějících zdlouhavá a složitá hledání.

Při plánování databázového serveru vezměte do úvahy své priority a prostředí.

Posouzení priorit

Při přípravě počáteční konfigurace a plánování strategie zálohování mějte na paměti charakteristické vlastnosti databázového serveru:

- Budou tento databázový server využívat aplikace instalované na jiných počítačích?
- Jaký je maximální předpokládaný počet uživatelů?
- Do jaké míry hodláte řídit prostředí uživatelů?
- Jste omezeni velikostí diskového prostoru, výkonem jednotky CPU nebo dostupností operátorů?
- Má databázový server pracovat bez dozoru administrátora?
- Zpracovává obvykle databázový server velký počet krátkých transakcí nebo malý počet dlouhých transakcí?
- Které nové vlastnosti databázového serveru nebo souvisejících produktů chcete využívat?

Posouzení prostředí

Před zahájením počáteční konfigurace databázového serveru shromážděte následující údaje. O některé z těchto údajů budete muset požádat administrátora systému:

- Názvy hostitelů a adresy IP dalších počítačů v síti.
- Podporuje používaná platforma systému UNIX službu NIS (Network Information Service)?
- Konfigurace řadiče disku.

Kolik diskových jednotek je k dispozici? Jsou některé z diskových jednotek rychlejší než ostatní? Kolik řadičů disku je k dispozici? Jaká je konfigurace řadiče disku?

- Jaké jsou požadavky, vlastnosti a omezení správce paměti a zálohovacích zařízení?

Další informace naleznete v příručce *IBM Informix Storage Manager Administrator's Guide* nebo v dokumentaci správce paměti.

- Bude nutná aktualizace hardwaru a operačního systému? Používáte 32bitový nebo 64bitový operační systém?

- Sdílená paměť operačního systému a další zdroje.
Kolik sdílené paměti je k dispozici? Jak velkou její část lze použít pro databázový server?

Pouze pro systém UNIX: Soubor Poznámky k počítači obsahuje informace o tom, které parametry lze použít pro jednotlivé platformy systému UNIX.

Konfigurace operačního systému

Před zahájením konfigurace databázového serveru je nutné řádně konfigurovat operační systém. Tato úloha bude možná vyžadovat pomoc administrátora systému.

32bitovou verzi databázového serveru Dynamic Server lze spustit v 64bitovém nebo 32bitovém operačním systému. 64bitovou verzi databázového serveru Dynamic Server lze spustit pouze v 64bitovém operačním systému. Více informací naleznete v části "Využití paměti na 64bitových platformách" na stránce 7-33.

Konfigurace paměti v systému Windows

V prostředí systému Windows musíte vytvořit oddíly NTFS a nakonfigurovat paměť. Další informace naleznete také v části "Vytvoření oddílů NTFS ve Windows" na stránce 1-6.

Nedostatek paměti pro databázový server může mít za následek nadměrnou aktivitu při správě vyrovnávací paměti. Při nastavování hodnot virtuální paměti v ovládacím panelu Systém se přesvědčte, že máte dostatek stránkovacího prostoru pro celkový objem fyzické paměti.

Maximální adresový prostor

Maximální adresový prostor na počítačový systém v systému Windows je:

- 1,7 GB, pokud není soubor **boot.ini** změněn na 3 GB.
- 2,7 GB, pokud je soubor **boot.ini** změněn na 3 GB.

Chcete-li velikost prostoru pro virtuální paměť uživatele a jádra určit přesněji, můžete použít přepínač **/userva=xxxx**. Použijte tento soubor s 3gigabajtovým přepínačem v souboru **boot.ini**, chcete-li optimalizovat velikost prostoru režimu uživatele na hodnotu, která je mezi 2 a 3 gigabajty; rozdíl (3072 minus xxxx) bude vrácen do režimu jádra. Pověšměte si, že xxxx je hodnota vyjádřená v megabajtech.

Následující ukázkový soubor **boot.ini** předvádí, jak lze pomocí tohoto nového přepínače nastavit počítač tak, aby přidělil 2 900 MB virtuální paměti režimu uživatele a 1 196 MB virtuální paměti režimu jádra. Tím se zvýší dostupný prostor jádra o 172 MB.

```
[Boot Loader]
Timeout=30
Default=multi(0)disk(0)rdisk(0)partition(2)\WINNT
[Operating Systems]
multi(0)disk(0)rdisk(0)partition(2)\WINNT="Microsoft Windows Server 2003"
/fastdetect /3GB /Userava=2900
```

Modifikace parametrů jádra systému UNIX

Soubor Poznámky k počítači obsahuje doporučené hodnoty konfigurace zdrojů operačního systému. Tyto doporučené hodnoty použijte při konfiguraci operačního systému.

Pokud se doporučené hodnoty pro databázový server výrazně liší od stávajícího prostředí, zvažte změnu konfigurace operačního systému. Další informace naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

V některých operačních systémech lze určit množství sdílené paměti přidělené databázovému serveru. Celkovém množství dostupné paměti ovlivňuje hodnoty parametrů sdílené paměti,

kteří můžete nastavit v konfiguračním souboru. Obecně lze říci, že se zvětšením prostoru dostupného pro sdílenou paměť roste i výkon. Možná bude nezbytné určit počet zámků a semaforů.

Základní informace o roli parametrů jádra systému UNIX naleznete v části Kapitola 8, “Správa sdílené paměti”, na stránce 8-1.

Konfigurace diskového prostoru

Optimální výkon datových tržič a datových skladů můžete dosáhnout správnou konfigurací disků. Diskový vstup - výstup se na době odezvy operací jazyka SQL podílí největší měrou. Databázový server umožňuje paralelní přístup k více diskům počítače. Než přidělíte diskový prostor, přečtěte si informace o diskovém prostoru uvedené v administrační příručce používaného operačního systému a nahlédněte do části “Přidělování disků bloků” na stránce 9-3.

Použití velkých bloků

Velikost bloků pro prostory dbspace je 4 terabajty pro 2kilobajtovou stránku. Bloky mohou být uloženy kdekoli v 64bitovém adresovém prostoru.

Chcete-li aktivovat vytváření velkých bloků, musíte použít obslužný program **onmode**. Obslužný program **onmode** používá příznak **-BC** k řízení dostupnosti velkých bloků, tedy bloků větších než 2 gigabajty.

Při první migraci databázového serveru na novou verzi serveru Dynamic Server nejsou velké bloky povoleny. Chcete-li povolit vytvoření velkých bloků, spusťte příkaz **onmode -BC 1**.

Pokud je nutné převést starší verze serveru Dynamic Server na formát velkých bloků, spusťte příkaz **onmode -BC 2**.

Pokud je při spuštění serveru kořenový blok rozsáhlý, je krok **onmode -BC 1** vynechán a režim **-BC 2** se spustí automaticky.

Vytvoření souborů bloků v systému UNIX

V systému UNIX můžete ukládat data v blocích, které používají buď disky bez vyrovnávací paměti (*přímé*), nebo soubory operačního systému, které jsou známy jako soubory *s vyrovnávací pamětí* nebo *předpřipravené* soubory.

Přímý přístup k disku neboli přístup bez vyrovnávací paměti

Systém UNIX umožňuje přístup k diskům bez vyrovnávací paměti prostřednictvím znakových zařízení (jsou také nazývána *přímá* disková zařízení). Chcete-li v systému UNIX vytvořit přímá disková zařízení, postupujte podle pokynů uvedených v dokumentaci operačního systému.

Databázový server využívá *přímý* přístup na disk k zlepšení rychlosti a spolehlivosti operací diskového vstupu - výstupu. Přímý přístup k disku obchází mechanismus ukládání souborů do vyrovnávací paměti poskytovaný operačním systémem. Databázový server sám řídí přesuny dat mezi diskem a pamětí. Databázový server optimalizuje přístup k tabulkám tím, že zajišťuje souvislé ukládání řádků.

Důležité: I když byste pomocí přímých diskových zařízení v systému UNIX mohli dosahovat vyššího výkonu, nejnovější pokrok v ukládání vstupu - výstupu do vyrovnávací paměti pro předpřipravené zápisy může poskytovat podobný, možná i vyšší výkon. Chcete-li zjistit, jak dosáhnout nejlepšího výkonu zařízení, změřte výkon systému s daným rozvržením prostorů dbspace a tabulek při použití obou typů zařízení.

Přidělení disků pro databázový server:

1. Nakonfigurujte přímé diskové zařízení pro každý disk.
2. Vytvořte standardní názvy zařízení nebo názvy souborů.
3. Nastavte oprávnění, vlastnictví a skupiny pro každé přímé diskové zařízení.

Předpřipravené soubory

Nezáleží-li na optimalizaci výkonu, lze databázový server konfigurovat tak, aby ukládal data do *předpřipravených* souborů. Nastavení předpřipravených souborů je jednodušší než nastavení přímých diskových zařízení.

Vytvoření oddílů NTFS ve Windows

V systému Windows instalujte databázový server do oddílu typu NTFS (New Technology File System) nebo typu FAT (File Allocation Table). Všechny prostory dbspace, blobspace a sbpace však musíte uložit do souborů NTFS nebo na fyzickou jednotku nebo logický diskový oddíl. Měli byste používat soubory NTFS, chcete-li zjednodušit administraci disků. Více informací o souborech NTFS naleznete v dokumentaci systému Windows a v části "Přístup k disku v systému Windows" na stránce 9-3.

Jsou-li všechny používané oddíly typu FAT, bude nutné alespoň jeden z nich převést na oddíl typu NTFS. Můžete použít obslužný program **convert** systému Windows, jak uvádí následující příklad:

```
convert /fs:ntfs
```

Nastavení oprávnění, vlastnictví a skupiny

Soubory nebo přímá disková zařízení, které využívá databázový server, musí mít vhodná vlastnictví a oprávnění.

Jen pro UNIX

V systému UNIX musí být vlastník i skupina nastaveni na hodnotu **informix** a uživateli i skupině (ne však ostatním uživatelům) musí být přidělena oprávnění ke čtení a k zápisu.

Chcete-li, aby uživatelé jiní než **informix** nebo **root** mohli spouštět příkazy ON-Bar, vytvořte skupinu **bargroup**. Pouze členové skupiny **bargroup** mohou spouštět příkazy ON-Bar. Skupina **bargroup** není vytvořena automaticky v rámci instalace databázového serveru. Postup při vytváření skupiny naleznete v příručce *IBM Informix Dynamic Server Installation Guide for UNIX and Linux* nebo v dokumentaci systému UNIX.

Konec Jen pro UNIX

Jen pro Windows

V systému Windows musí být soubory vlastnictvím člena skupiny **Informix-Admin**. Skupina **Informix-Admin** se vytvoří automaticky v rámci instalace databázového serveru.

Konec Jen pro Windows

Vytvoření standardních názvů zařízení (UNIX)

Při přiřazování zkrácených standardních názvů jednotlivým přímým diskovým zařízením byste měli používat symbolické odkazy. Použijete-li symbolické odkazy, můžete porušený disk nahradit novým diskem pouhým přiřazením symbolického názvu novému disku.

Chcete-li vytvořit propojení mezi názvem znakově orientovaného zařízení a názvem jiného souboru, použijte příkaz `link` systému UNIX (obvykle se nazývá **ln**).

Provedením příkazu **ls -l** systému UNIX v adresáři zařízení ověřte, zda obě zařízení a propojení existují. Následující příklad ilustruje propojení s přímými diskovými zařízeními. V případě, že operační systém nepodporuje symbolické odkazy, můžete použít tvrdé odkazy.

```
ls -l
crw-rw--- /dev/rxy0h
crw-rw--- /dev/rxy0a
lrwxrwxrwx /dev/my_root@->/dev/rxy0h
lrwxrwxrwx /dev/raw_dev2@->/dev/rxy0a
```

Nastavení proměnných prostředí

Každý uživatel, který chcete databázový server spustit nebo zastavit, nebo chce k němu získat přístup, musí mít požadovaná oprávnění pro přístup k databázi a musí nastavit příslušné proměnné prostředí. Některé proměnné prostředí jsou povinné, jiné nepovinné.

Nastavení požadovaných proměnných prostředí:

1. Proměnnou **INFORMIXDIR** nastavte na název adresáře, do kterého jste instalovali produkty IBM Informix.
2. Nastavte proměnnou prostředí **PATH** tak, aby obsahovala adresář **\$INFORMIXDIR/bin** (v systému UNIX) nebo adresář **%INFORMIXDIR%\bin** (v systému Windows).
3. Proměnnou **INFORMIXSERVER** nastavte na název databázového serveru.

Rada: Nastavte proměnné prostředí v příslušném spouštěcím souboru pro soubor se skriptem interpretu příkazů nebo pro systém Windows.

Proměnnou prostředí “**\$INFORMIXDIR**” můžete zahrnout do souboru **ONCONFIG**. Měla by být první hodnotou názvu cesty ve specifikacích názvů cest.

Příručka *IBM Informix Guide to SQL: Reference* obsahuje úplný seznam proměnných prostředí. Informace o vlivu nastavení proměnných prostředí na výkon databázového serveru naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Obrázek Tabulka 1-1 znázorňuje proměnné prostředí, které musí být nastaveny před přístupem k databázovému serveru nebo před provedením většiny úloh administrace databázového serveru.

Tabulka 1-1. Povinné proměnné prostředí

Proměnná prostředí	Popis
CLASSPATH	Používáte-li třídy J/Foundation, určete umístění souboru jvphome/krakatoa.jar tak, aby mohla sada Java Development Kit (JDK) kompilovat zdrojové soubory jazyka Java.
INFORMIXDIR	Určuje adresář, do kterého jste instalovali databázový server IBM Informix.
INFORMIXSERVER	Určuje název výchozího databázového serveru. Obsahuje hodnotu určenou v konfiguračním parametru DBSERVERNAME nebo DBSERVERALIASES .
JVPHOME	Používáte-li třídy J/Foundation, určuje adresář, do kterého jste instalovali ovladač IBM Informix JDBC Driver.
ONCONFIG	<p>Určuje název aktivního souboru ONCONFIG. Všichni uživatelé, kteří používají obslužné programy databázového serveru, například onstat, musí nastavit proměnnou prostředí ONCONFIG. Uživatelé, kteří spouštějí klientské aplikace, proměnnou prostředí ONCONFIG nastavovat nemusí.</p> <p>Není-li k dispozici proměnná prostředí ONCONFIG, použije databázový server konfigurační hodnoty ze souboru onconfig:</p> <p>V systému UNIX: \$INFORMIXDIR/etc/onconfig</p> <p>V systému Windows: %INFORMIXDIR%\etc\onconfig.std</p>

Tabulka 1-1. Povinné proměnné prostředí (pokračování)

Proměnná prostředí	Popis
PATH	Určuje umístění spustitelných souborů. V systému UNIX: \$INFORMIXDIR/bin V systému Windows: %INFORMIXDIR%\bin
TERM	Umožňuje příkazu DB–Access rozpoznat terminál, který používáte a komunikovat s ním. Tato proměnná prostředí není při inicializaci ani při spuštění povinná, ale musí být nastavena, než spustíte aplikace.
TERMCAP TERMINFO INFORMIXTERM	Určuje, zda má příkaz DB–Access použít údaje ze souboru termcap nebo z adresáře terminfo . Je-li nastavení těchto proměnných povinné, požádejte o pomoc administrátora systému UNIX, protože hodnoty těchto proměnných závisí na používaném systému.

Nastavení proměnných prostředí GLS

Následující proměnné prostředí umožňují využívat funkci GLS (Global Language Support). Chcete-li používat jiný jazyk než americkou angličtinu, nastavte tyto proměnné prostředí:

- **CLIENT_LOCALE**
- **DB_LOCALE**
- **SERVER_LOCALE**
- **DBLANG**
- **C8BITLEVEL**
- **ESQLMF**
- **GLS8BITFSYS**
- **GL_DATE**
- **GL_DATETIME**

Další informace naleznete v příručce *IBM Informix GLS User's Guide*.

Nastavení proměnných prostředí v systému UNIX

Nastavte proměnné prostředí systému UNIX jedním z následujících způsobů:

- V příkazovém řádku s výzvou operačního systému.
Pokud nastavíte proměnnou prostředí v příkazovém řádku s výzvou operačního systému, budete ji muset při dalším přihlášení do systému znovu přiřadit.
- V konfiguračním souboru prostředí, například v souboru **\$INFORMIXDIR/etc/informix.rc** nebo **.informix**.
Konfigurační soubor prostředí je společný nebo soukromý soubor, v němž lze nastavit proměnné prostředí pro každého uživatele databázového serveru. Použitím konfiguračního souboru prostředí snížíte počet proměnných prostředí, které byste jinak museli zadat v příkazovém řádku nebo do souboru pro interpret příkazů.
- V souboru **.profile** nebo **.login**.
Nastavíte-li proměnnou prostředí v souboru **.login**, **.cshrc** nebo **.profile**, je tato proměnná automaticky přiřazena při každém přihlášení do systému. Více informací o těchto souborech naleznete v příručce operačního systému.

Chcete-li potlačit proměnné prostředí, které byly automaticky nastaveny, použijte soukromý soubor proměnných prostředí, **~/.informix** nebo ručně přiřaďte proměnným prostředím nové hodnoty. Více informací o souborech **.informix** a **informix.rc** naleznete v části *IBM Informix Dynamic Server Administrator's Reference*.

Chcete-li zkontrolovat platnost proměnných prostředí, použijte obslužný program **chkenv**.

Obrázek 1-1 zobrazuje příklad souboru, který obsahuje proměnné prostředí databázového serveru **miami**. Proměnná **LD_LIBRARY_PATH** obsahuje údaj o umístění databázového serveru a souborů knihovny ESQL/C.

```
setenv INFORMIXDIR /ix/informix93
setenv INFORMIXSQLHOSTS /ix/sqlhosts.unified
setenv ONCONFIG s.miami
setenv INFORMIXSERVER miami

# Nastavení tříd J/Foundation
setenv JVPHOME /ix/informix93/extend/krakatoa
setenv CLASSPATH $JVPHOME/krakatoa.jar:$JVPHOME/jdbc.jar:/usr/java/lib/classes.zip

# Přidání cest k souborům JAR jazyka Java, přidání adresáře /usr/ccs/bin
# pro kompilátor jazyka C:
setenv PATH $INFORMIXDIR/bin:$INFORMIXDIR/extend/krakatoa/krakatoa.jar:
    $INFORMIXDIR/extend/krakatoa/jdbc.jar:/usr/ccs/bin:$PATH

setenv LD_LIBRARY_PATH $INFORMIXDIR/lib:$INFORMIXDIR/lib/esql:/usr/lib
```

Obrázek 1-1. Ukázka souboru nastavení

Nastavení proměnných prostředí v systému Windows

V systému Windows připraví instalační procedura soubor **setenv.cmd**, který nastaví proměnné prostředí na jejich správné hodnoty. Soubor **setenv.cmd** je uložen v adresáři **%INFORMIXDIR%**. Než použijete kterýkoliv z obslužných programů v příkazovém řádku, musíte spustit soubor **setenv.cmd**.

Můžete nastavit proměnné prostředí nebo potlačit proměnné prostředí, které byly automaticky nastaveny, a to na následujících místech:

- V systému Windows v nastavení ovládacího panelu **Systém > Prostředí > Uživatelské proměnné**.
- V relaci příkazového řádku.
- V dávkovém souboru **%INFORMIXDIR%\dbservername.cmd**.

Tento dávkový soubor použijte ke konfiguraci obslužných programů příkazového řádku.

Další informace naleznete v příručce *IBM Informix Guide to SQL: Reference* a v části “Vytvoření souboru ONCONFIG v systému Windows” na stránce 1-11.

Konfigurace propojitelnosti

Údaje o propojitelnosti umožňují klientské aplikaci připojit se ke kterémukoliv databázovému serveru IBM Informix v síti. Data propojitelnosti příslušného databázového serveru zahrnují název databázového serveru, typ připojení, který může klient použít pro připojení k tomuto serveru, název hostitelského počítače nebo uzlu, na kterém je spuštěn databázový server a název služby, pod kterým je znám.

Údaje o propojitelnosti musíte připravit i v případě, že se klientská aplikace a databázový server nacházejí na tomtéž počítači nebo uzlu.

Před spuštěním databázového serveru není nutné zadat všechna možná síťová připojení do souboru **sqlhosts** ani do registru. Chcete-li však zpřístupnit nové připojení, musíte uvést databázový server do režimu offline a pak ho znovu uvést do režimu online.

Podrobné informace o konfiguraci propojitelnosti uvádí Kapitola 3, “Komunikace mezi klientem a serverem”, na stránce 3-1.

Při konfiguraci propojitelnosti v systému UNIX zvažte též nastavení konfiguračních parametrů `LISTEN_TIMEOUT` a `MAX_INCOMPLETE_CONNECTIONS`. Tyto parametry umožňují snížit riziko agresivního útoku typu DOS (denial-of-service) tím, že znesnadňují zahlcení virtuálního procesoru typu listener, který zpracovává připojení. Další informace naleznete v příručce *IBM Informix Security Guide*.

Soubor `sqlhosts` v systému UNIX

V systému UNIX jsou informace o propojitelnosti uloženy v souboru `sqlhosts`. Výchozí umístění tohoto souboru je `$INFORMIXDIR/etc/sqlhosts`. Umístíte-li tyto údaje jinak, musíte nastavit proměnnou prostředí `INFORMIXSQLHOSTS`. Další informace naleznete v části “Soubor `sqlhosts` a klíč registru `SQLHOSTS`” na stránce 3-13.

Nastavíte-li několik databázových serverů tak, aby využívaly distribuované dotazy, použijte jeden z následujících způsobů k uložení dat `sqlhosts` pro *všechny* databáze:

- V jednom souboru `sqlhosts`, na který odkazuje proměnná `INFORMIXSQLHOSTS`.
- V samostatných souborech `sqlhosts` v adresáři každého z databázových serverů.

K úpravě souboru `sqlhosts` použijte textový editor nebo obslužný program ISA. Další informace naleznete v části “Konfigurace propojitelnosti pomocí programu ISA” na stránce 1-10.

Soubory konfigurace sítě

Síťová připojení pomocí protokolu sítě Internet vyžadují kromě souborů `sqlhosts` také položky v souborech `/etc/hosts` a `/etc/services`. U připojení typu IPX/SPX jsou názvy pomocných souborů závislé na dodavateli hardwaru.

Soubory zabezpečení sítě

Databázové servery IBM Informix dodržují při vytváření připojení požadavky na zabezpečení systému UNIX. Administrátor systému UNIX by měl potom upravit soubory `/etc/passwd`, `etc/hosts`, `~/.rhosts` a další související soubory.

Soubory pro konfiguraci a zabezpečení sítě jsou podrobně popsány v příručkách operačních systémů.

Registr `sqlhosts` v systému Windows

V systému Windows obsahuje informace `sqlhosts` registr `HKEY_LOCAL_MACHINE`. Údaje registru připraví instalační procedura databázového serveru. Neměli byste upravovat položku registru `HKEY_LOCAL_MACHINE`.

Údaje `sqlhosts` můžete spravovat pomocí obslužného programu `setnet32`. Další informace o obslužném programu `setnet32` naleznete v dokumentaci klienta, v části věnované instalaci. Obslužný program `setnet32` však neumožňuje přiřadit databázový server ke skupině databázových serverů.

Konfigurace propojitelnosti pomocí programu ISA

Program IBM Informix Server Administrator (ISA) můžete použít ke konfigurování údajů o propojitelnosti pro databázové servery IBM Informix a skupin databázových serverů pro replikaci Enterprise Replication. Program ISA umožňuje upravovat soubor `sqlhosts` systému UNIX a registr `sqlhosts` systému Windows. Další informace naleznete v pokynech uživatelského rozhraní programu ISA nebo v nápovědě online.

Konfigurace databázového serveru

Konfigurační parametry jsou uloženy v souboru **ONCONFIG**. Instalační skripty produktu nastaví výchozí hodnoty většiny konfiguračních parametrů.

Další informace o konfiguračních parametrech a o tom, jak je monitorovat, obsahuje Kapitola 2, “Konfigurační parametry”, na stránce 2-1 a kapitola věnovaná konfiguračním parametrům v příručce *IBM Informix Dynamic Server Administrator's Reference*. Další informace o pořadí, v jakém databázový server kontroluje soubory a v nich obsažené konfigurační hodnoty, naleznete v části “Zpracování konfiguračního souboru” na stránce 4-3.

Příprava konfiguračního souboru ONCONFIG

Soubor šablony **onconfig.std** v podadresáři **etc** adresáře **INFORMIXDIR** obsahuje počáteční hodnoty mnoha konfiguračních parametrů. Zkopírujte tuto šablonu a přizpůsobte ji specifickým potřebám konfigurace.

Soubory šablony obsahují počáteční hodnoty mnoha konfiguračních parametrů. Pomocí textového editoru nebo obslužného programu ISA můžete změnit konfigurační parametry v souboru **ONCONFIG**.

Důležité: Soubor šablony neměňte a neodstraňujte. Databázový server poskytuje tyto soubory jako šablony, ne jako funkční konfigurační soubory.

Poznámka: Vynecháte-li některé parametry v kopii souboru **ONCONFIG**, databázový server při spuštění nahradí chybějící parametry hodnotami uvedenými v souboru **onconfig.std**.

Vytvoření souboru ONCONFIG v systému UNIX

Při první instalaci softwaru IBM Informix se vytvoří a inicializuje nová instance databázového serveru. Instalační skript automaticky vytvoří soubor **onconfig.demo**.

Příprava souboru ONCONFIG pomocí textového editoru:

1. Zkopírujte a přejmenujte soubor **\$INFORMIXDIR/etc/onconfig.std** a uložte ho do podadresáře **etc**. (Přeskočte tento krok, používáte-li obslužný program ISA.)
2. Pomocí textového editoru nebo programu ISA upravte konfigurační soubor **ONCONFIG**.
3. Proměnnou prostředí **ONCONFIG** nahraďte názvem nového souboru **ONCONFIG**.
4. Jedná-li se o novou instanci databázového serveru, inicializujte ho.

V opačném případě vypněte a restartujte databázový server.

Vytvoření souboru ONCONFIG v systému Windows

V systému Windows se při první instalaci softwaru IBM Informix vytvoří a inicializuje nová instance databázového serveru. Novou instanci databázového serveru lze vytvořit též s použitím programu Instance Manager (**instmgr.exe**). Program Instance Manager automaticky vytvoří soubor **ONCONFIG**.

Příprava souboru ONCONFIG s použitím programu Instance Manager:

1. Pomocí programu Instance Manager vytvořte novou instanci databázového serveru.
Další informace naleznete v části “Použití programu Instance Manager k vytvoření nové instance databázového serveru (Windows)” na stránce 1-12.
2. Úpravy konfiguračního souboru **ONCONFIG** provádějte pomocí textového editoru nebo v programu ISA.

3. Do proměnné prostředí **ONCONFIG** vložte název nového souboru **ONCONFIG**.
4. Vypněte a restartujte databázový server, aby změny nabyly účinnosti.
 - a. V okně ovládacího panelu **Služba** vyberte službu serveru Dynamic Server a klepněte na tlačítko **Zastavit**.
 - b. Klepněte na tlačítko **Spustit**.

Použití souboru dbservername.cmd ke změně proměnné prostředí ONCONFIG:

1. Proměnnou prostředí **ONCONFIG** příslušné relace změňte v souboru **%INFORMIXDIR%\dbservername.cmd**.
2. Aby změny nabyly účinnosti, spusťte soubor **dbservername.cmd**.

Použití funkce Nastavení serveru obslužného programu ISA k vlastnímu nastavení konfigurace

Funkci **Nastavení serveru** programu ISA můžete použít ke konfigurování provozního databázového serveru a případně i ke konfiguraci tříd J/Foundation a paměti databáze. Další informace naleznete v nápovědě online tohoto programu a v pokynech zobrazovaných v uživatelském rozhraní programu ISA.

Použití funkce Nastavení serveru k vlastnímu nastavení konfigurace:

1. Spusťte webový prohlížeč a otevřete webovou stránku programu ISA:

```
http://<název_hostitele>.<název_domény>:  
číslo_portu/
```

 Pokud jste nenainstalovali nebo nespustili program ISA, nahlédněte do příručky *IBM Informix Installation Guide*, kde jsou uvedené potřebné pokyny.
2. Přihlašte se pod uživatelským jménem a s heslem, které jste vytvořili během instalace.
3. Na hlavní stránce vyberte název databázového serveru.
4. Na stránce zobrazeného serveru v programu ISA klepněte na položku **Nastavení serveru**.

Použití obslužného programu IBM Informix Server Administrator k aktualizaci souboru ONCONFIG

Program ISA můžete použít k úpravě konfiguračních parametrů a k nastavení proměnných prostředí všech databázových serverů. Obslužný program ISA automaticky vytvoří soubor **ONCONFIG** (není zapotřebí kopírovat soubor **onconfig.std**).

Použití programu IBM Informix Server Administrator k aktualizaci souboru ONCONFIG:

1. V obslužném programu ISA zobrazte soubor nastavení a upravte proměnné prostředí.
2. Vyberte název databázového serveru.
3. V obslužném programu ISA zobrazte soubor **ONCONFIG** a upravte příslušné konfigurační parametry.
 Obslužný program ISA automaticky aktualizuje soubor **ONCONFIG**.
4. Vypněte a restartujte databázový server, aby změny nabyly účinnosti.

Použití programu Instance Manager k vytvoření nové instance databázového serveru (Windows)

K vytvoření nové instance databázového serveru můžete použít program Instance Manager. Program Instance Manager automaticky vytvoří soubor **ONCONFIG**. Více informací o programu Instance Manager a prioritách plánování naleznete v příručce *IBM Informix Dynamic Server Installation Guide for Microsoft Windows*.

Použití programu Instance Manager:

1. Vyberte položku **Server Instance Manager** z nabídky serveru **Dynamic Server**.
2. Chcete-li vytvořit novou instanci databázového serveru, klepněte na příkaz **Vytvořit novou** a postupujte podle pokynů v okně průvodce.
3. Chcete-li zvýšit výkon databázového serveru, vyberte v programu Instance Manager položku plánování priority pro proces **oninit**. Výchozí plánování priority v systému Windows je **Normální**.
4. Chcete-li odstranit instanci databázového serveru, klepněte na příkaz **Odstranit server**.

Použití programu Instance Manager k přejmenování instance databázového serveru (Windows)

Chcete-li změnit název instance databázového serveru, můžete též použít volbu přejmenování integrovanou do programu Instance Manager.

Přejmenování instance databázového serveru:

1. Vypněte instanci databázového serveru, kterou chcete přejmenovat.
2. Vyberte položku **Server Instance Manager** z nabídky serveru **Dynamic Server**. Program Instance Manager zobrazí seznam dostupných instancí.
3. Vyberte instanci, která má být přejmenována a klepněte na příkaz **Přejmenovat server**.
4. Do zobrazeného dialogového okna zadejte heslo a určete nový název databázového serveru.

Po přejmenování instance serveru můžete instanci spustit.

Pokud administrátor změnil název souborů ONCONFIG nebo MSGLOG na vlastní název předtím, než byla přejmenována instance serveru, musíte změnit název ruční úpravou souboru.

Konfigurace podpory jazyka Java

Server IBM Informix Dynamic Server with J/Foundation umožňuje vyvíjet a spouštět uživatelské rutiny UDR v jazyku Java. Databázový server nakonfigurujte bez jazyka Java a pak ho upravte a přidejte podporu jazyka Java. Konfigurace databázového serveru, který bude podporovat jazyk Java, vyžaduje několik dalších kroků:

Konfigurace databázového serveru pro podporu uživatelských rutin v jazyku Java:

1. Vytvořte prostor sbpace k uložení souborů JAR jazyka Java.
2. Vytvořte soubor vlastností JVP.
3. Přidejte (nebo změňte) konfigurační parametry jazyka Java v souboru **ONCONFIG**.
4. Nastavte proměnné prostředí.

Pokyny pro nastavení naleznete v příručce *J/Foundation Developer's Guide*.

Spuštění a administrace databázového serveru

Po instalaci a konfiguraci databázového serveru je nutné provést nejméně jednu z následujících úloh:

- Připravit připojení k aplikacím.
- Spustit databázový server a inicializovat diskový prostor.
- Vytvořit paměťové prostory.
- Nastavit systém zálohování a obnovení.

- Provést úlohy administrace.

Spuštění databázového serveru a inicializace diskového prostoru

Jen pro UNIX

Chcete-li v systému UNIX převést databázový server do režimu online, zadejte příkaz **oninit**.

Spouštíte-li nový databázový server v systému UNIX, inicializujte diskový prostor a převedte databázový server do režimu online příkazem **oninit** s příznakem **-i**.

Konec Jen pro UNIX

Jen pro Windows

V systému Windows pracuje databázový server jako služba. K uvedení databázového serveru do režimu online použijte ovládací panel **Služby**. Chcete-li inicializovat databázový server, klepněte na službu Dynamic Server a zadejte parametr **-iy** do políčka **Parametry spuštění**.

Další způsobem, jak inicializovat databázový server v systému Windows, je použití následujícího příkazu, ve kterém proměnná *dbservername* zastupuje název databázového serveru:

```
starts dbservername -iy
```

Konec Jen pro Windows

Automatické spuštění systému není dostupné prostřednictvím administrativního příkazu jazyka SQL. Databázový server musíte spustit podle informací uvedených v této části. Používáte-li při spuštění databázového serveru příkazem **oninit** rozhraní API pro administraci a plánovač, server automaticky provede všechny uložené procedury naplánované ke spuštění při spuštění serveru. Další informace naleznete v části Kapitola 26, “Automatické monitorování a nápravné akce”, na stránce 26-1.

Inicializace pouze při spuštění nového databázového serveru

Upozornění: Při inicializaci diskového prostoru jsou znehodnocena všechna stávající data na databázovém serveru. Diskový prostor inicializujte jen v případě, že spouštíte nový databázový server.

Podrobný popis druhů inicializace a souvisejících příkazů uvádí Kapitola 4, “Inicializace databázového serveru”, na stránce 4-1.

Příprava pro automatické spuštění

Připravte registr nebo skripty operačního systému pro automatické spuštění a ukončení databázového serveru. Další informace naleznete v částech “Příprava automatického spuštění v systému Windows” a “Příprava spouštěcích a ukončovacích skriptů pro systém UNIX” na stránce 1-15.

Příprava automatického spuštění v systému Windows

Pokud v systému Windows změňte heslo serveru Informix, změňte heslo v ovládacím panelu **Služby**.

Automatické spuštění databázového serveru při spuštění systému Windows:

1. V okně ovládacího panelu **Služby** vyberte službu serveru Dynamic Server a klepněte na tlačítko **Spustit**.
2. V dialogovém políčku **Typ spouštění** vyberte položku **Automaticky**.
3. Na kartě **Přihlášení** vyberte položku **Tento účet** a ověřte, zda je v textovém poli zadán uživatel informix.

Příprava spouštěcích a ukončovacích skriptů pro systém UNIX

Spouštěcí skript systému UNIX můžete upravit a automaticky inicializovat databázový server, jakmile počítač přejde do režimu s více uživateli. Můžete také upravit ukončovací skript systému UNIX a řídit vypínání databázového serveru při každém ukončení systému UNIX.

Chcete-li připravit spouštěcí skript systému UNIX, přidejte příkazy obslužných programů systému UNIX a databázového serveru do spouštěcího skriptu systému UNIX tak, aby skript provedl následující kroky.

Program ISA obsahuje ukázkový skript pro spuštění a ukončení systému UNIX. Tento skript můžete v souboru **\$INFORMIXDIR/etc/ids-example.rc** upravit.

Příprava spouštěcího skriptu systému UNIX:

1. Nastavte proměnnou prostředí **INFORMIXDIR** na úplnou cestu k adresáři, ve kterém je nainstalován databázový server.
2. Do proměnné prostředí **PATH** nastavte název adresáře **\$INFORMIXDIR/bin**.
3. Nastavte proměnnou prostředí **ONCONFIG** na požadovaný konfigurační soubor.
4. Nastavte proměnnou prostředí **INFORMIXSERVER** tak, aby mohla být aktualizována (resp. vytvořena, je-li to nutné) databáze **sysmaster**.
5. Spusťte příkaz **oninit**, který spustí databázový server a uvede ho do režimu online.
Máte-li v úmyslu inicializovat více verzí databázového serveru (vícenásobné uložení), musíte znovu nastavit proměnné **ONCONFIG** a **INFORMIXSERVER** a znovu provést příkaz **oninit** pro každou instanci databázového serveru.
6. Pokud používáte ke správě zálohování databázového serveru program IBM Informix Storage Manager (ISM), musíte server ISM spustit v každém uzlu.
Další informace o tom, jak spustit server ISM naleznete v příručce *IBM Informix Installation Guide*.

Pokud jsou v různých adresářích nainstalovány různé verze databázového serveru, musíte znovu nastavit proměnnou **INFORMIXDIR** a zopakovat předchozí kroky pro každou jednotlivou verzi.

Chcete-li řídit vypnutí databázového serveru, kdykoliv je ukončen systém UNIX, přidejte do ukončovacího skriptu systému UNIX příkazy obslužných programů systému UNIX a databázového serveru tak, aby skript provedl následující kroky.

Příprava ukončovacího skriptu systému UNIX:

1. Nastavte proměnnou prostředí **INFORMIXDIR** na úplnou cestu k adresáři, ve kterém je nainstalován databázový server.
2. Do proměnné prostředí **PATH** nastavte název adresáře **\$INFORMIXDIR/bin**.
3. Nastavte proměnnou prostředí **ONCONFIG** na požadovaný konfigurační soubor.
4. Spusťte příkaz **onmode -ky**, který zahájí okamžité ukončení databázového serveru a jeho uvedení do režimu offline.
Máte-li spuštěných několik verzí databázového serveru (vícenásobné uložení), musíte znovu nastavit proměnnou **ONCONFIG** a pro každou instanci databázového serveru znovu provést příkaz **onmode -ky**.

Pokud jsou v různých adresářích nainstalovány různé verze databázového serveru, musíte znovu nastavit proměnnou **INFORMIXDIR** a zopakovat předchozí kroky pro každou jednotlivou verzi.

V ukončovacím skriptu systému UNIX by měly být příkazy pro vypnutí databázového serveru provedeny až poté, co všechny klientské aplikace dokončí své transakce a jsou ukončeny.

Příprava připojení k aplikacím

Jakmile se databázový server nachází v režimu online, můžete připojit klientské aplikace a začít s vytvářením databází. Než získáte přístup k informacím v databázi, musí se klientská aplikace připojit k prostředí databázového serveru. K připojování klientských aplikací k databázovému serveru a k jejich odpojování můžete používat příkazy jazyka SQL prostřednictvím následujících klientských programů:

- DB–Access
- SQL Editor
- IBM Informix ESQL/C
- IBM Informix ODBC Driver
- IBM Informix JDBC Driver

Další informace o vytváření databází naleznete v příručce *IBM Informix Database Design and Implementation Guide* a v příručce *IBM Informix Guide to SQL: Tutorial*. Informace o tom, jak používat klientské aplikace naleznete v příručkách *IBM Informix DB–Access User's Guide*, *IBM Informix ESQL/C Programmer's Manual*, *IBM Informix ODBC Driver Programmer's Manual* a *IBM Informix JDBC Driver Programmer's Guide*.

Vytvoření paměťových prostorů a bloků

Za naplánování a implementaci konfigurace paměti je zodpovědný administrátor. Způsob, jakým distribuujete data na discích, ovlivňuje výkon databázového serveru. *Blok* je logický disk, logická jednotka nebo obyčejný soubor přiřazený k databázovému serveru. Maximální velikost bloku je 4 TB. Maximální počet povolených bloků je 32766. Logický *paměťový prostor* je tvořen jedním nebo několika bloky.

Rada: Chcete-li využít výhod omezení velikosti bloku na celé 4 TB, přiřaďte každé diskové jednotce jen jeden blok. Tento způsob distribuce dat zvyšuje výkon databázového serveru.

Po inicializaci databázového serveru můžete vytvářet paměťové prostory `dbspace`, `blobspace` a `sbspace`. K vytváření paměťových prostorů a bloků použijte obslužný program **onspaces** nebo program `ISA`.

Používáte-li následující funkce, musíte vytvořit prostor `sbspace`:

- třídy `J/Foundation` (k uložení souborů `JAR` pro aplikace psané v jazyku `Java`)
- replikace `Enterprise Replication` (k ukládání řádkových dat zařazených do fronty)
- inteligentní velké objekty (datové typy `BLOB` a `CLOB`)
- datové typy s vícenásobnou reprezentací (např. datové typy modulů `DataBlade` nebo souborů `HTML`)

Podrobný popis paměťových prostorů a dalších fyzických jednotek, jako jsou prostory `tblspace` a oblasti, uvádí Kapitola 9, “Uložení dat”, na stránce 9-1. Informace související s přidělováním a správou paměťových prostorů uvádí část “Parametry diskového prostoru” na stránce 2-2 a Kapitola 10, “Správa diskového prostoru”, na stránce 10-1.

Další informace naleznete v příručce *IBM Informix Dynamic Server Enterprise Replication Guide* a v části *J/Foundation Developer's Guide*.

Podpora velkých bloků

Pokud jste právě převedli server z verze serveru Dynamic Server staší než je 9.4, nejsou povoleny bloky větší než 2 GB. Chcete-li získat podporu velkých bloků a posunů až do maximální velikosti 4 TB a v počtu větším než 2047 bloků, spusíte příkaz **onmode -BC 1**.

Data můžete prověřit v režimu **onmode -BC 1**. Pokud jste spokojeni se správností převodu dat, můžete spustit příkaz **onmode -BC 2**, a tím uvést server do režimu pouze s velkými bloky.

Po provedení příkazu **onmode -BC 2** není nadále podporováno opětovné vrácení. Povolenou podporu velkých bloků nelze zakázat.

Nastavení systému zálohování a ukládání

K zálohování a obnovení dat použijte obslužný program ON-Bar nebo **ontape**. Další informace o nastavení a použití obslužných programů ON-Bar a **ontape** naleznete v části "Parametry zálohování a obnovy" na stránce 2-4 a v příručce *IBM Informix Backup and Restore Guide*.

Nastavení obslužného programu ontape

Používáte-li jako nástroj k zálohování program **ontape**, musíte nejprve nastavit paměťová zařízení (páskové jednotky), než budete moct zálohovat a obnovovat data. Obslužný program **ontape** nevyžaduje ke svému spuštění správce paměti.

Nastavení správce paměti a programu ON-Bar

Používáte-li jako nástroj k zálohování program ON-Bar, musíte nejprve nastavit správce paměti a paměťová zařízení, než budete moct zálohovat a obnovovat data.

Obslužný program ON-Bar je dodáván v rámci sady programů IBM Informix Storage Manager (ISM). *Správce paměti* je aplikace, která spravuje paměťová zařízení a média, která obsahují zálohy. Správce paměti zpracovává veškerá pojmenování médií, požadavky na připojení a paměťové disky. Program ISM může současně ukládat data až do čtyř paměťových zařízení. Program ISM ukládá data na jednoduché páskové jednotky, optické disky a do souborových systémů. Chcete-li používat modernější paměťová zařízení, zálohovat na více než čtyři paměťová zařízení současně nebo provádět zálohování prostřednictvím sítě, můžete zakoupit správce paměti jiného výrobce.

Při plánování harmonogramu zálohování paměťového prostoru a logických protokolů se ujistěte, že jsou k dispozici paměťová zařízení a operátoři zálohování. Informace o konfiguraci a správě paměťových zařízení a médií naleznete v příručce *IBM Informix Storage Manager Administrator's Guide* nebo v dokumentaci dodané výrobcem spolu se správcem paměti.

Konfigurace vlastností relace

Vlastnosti relace databázového serveru můžete změnit při připojení nebo v době přístupu, aniž byste změnili aplikaci, kterou tato relace spouští. Může to být užitečné v případě, že nemůžete nastavit volby prostředí nebo proměnné prostředí změnou zdrojového kódu aplikace ani zahrnout příkazy SQL týkající se této relace, například proto, že příkazy SQL obsahují kód získaný dodavatelem.

Chcete-li změnit vlastnosti relace, navrhněte pro různé databáze vlastní procedury **sysdbopen()** a **sysdbclose()**, které by podporovaly aplikace určitých uživatelů nebo skupiny PUBLIC. Procedury **sysdbopen()** a **sysdbclose()** mohou obsahovat posloupnost příkazů

SET, SET ENVIRONMENT, SQL nebo SPL, které databázový server provádí pro takového uživatele nebo skupinu PUBLIC při otvírání nebo zavírání databáze.

Můžete například definovat procedury pro uživatele *user1* tak, aby obsahovaly příkazy SET PDQPRIORITY, SET ISOLATION LEVEL, SET LOCK MODE, SET ROLE nebo SET EXPLAIN ON, které budou prováděny vždy, když uživatel *user1* otevře databázi příkazem DATABASE nebo CONNECT TO.

Veškerá nastavení proměnných prostředí PDQPRIORITY a OPTCOMPIND této relace, která jsou zadána příkazy SET ENVIRONMENT v procedurách **sysdbopen()** přetrvávají po celou dobu trvání relace. Příkazy SET PDQPRIORITY a SET ENVIRONMENT OPTCOMPIND, které v běžných procedurách nejsou trvalé, jsou trvalé, pokud jsou obsaženy v procedurách **sysdbopen()**.

Procedura *user.sysdbclose()* běží, odpojí-li se uživatel, který je vlastníkem této procedury, od databáze (anebo procedura **PUBLIC.sysdbclose()** běží, pokud existuje a pokud aktuálním uživatelem není vlastníkem žádné procedury **sysdbclose()**).

Ve vlastních procedurách **sysdbopen()** a **sysdbclose()** neignoruje dynamický server jméno vlastníka uživatelské rutiny, je-li v databázi vyvolána rutina, která není kompatibilní se standardem ANSI.

Další informace naleznete v části “Konfigurace vlastností relace” a v příručce *IBM Informix Guide to SQL: Syntax*.

Konfigurace vlastností relace

Procedury **sysdbopen()**, které mění vlastnosti relace při připojení nebo v době přístupu, můžete nastavit tak, aby neměnily aplikaci, kterou tato relace spouští. Může to být užitečné v případě, že nemůžete nastavit volby prostředí nebo proměnné prostředí změnou zdrojového kódu aplikace ani zahrnout příkazy SQL týkající se této relace, například proto, že příkazy SQL obsahují kód získaný dodavatelem.

Nezbytný předpoklad: Procedury **sysdbopen()** a **sysdbclose()** v příkazech ALTER PROCEDURE, ALTER ROUTINE, CREATE PROCEDURE, CREATE PROCEDURE FROM, CREATE ROUTINE FROM, DROP PROCEDURE a DROP ROUTINE jazyka SQL může měnit pouze administrátor databáze nebo uživatel **informix**.

Konfigurace vlastností relace nastavením procedur **sysdbopen()** a **sysdbclose()**

1. Nastavte proměnnou prostředí IFX_NODBPROC na libovolnou hodnotu včetně hodnoty 0, aby databázový server nepoužíval procedury **sysdbopen()** nebo **sysdbclose()** nebo zabránil jejich provedení.
2. Zápisem příkazu CREATE PROCEDURE nebo CREATE PROCEDURE FROM definujte proceduru pro určitého uživatele nebo skupinu PUBLIC.
3. Vyzkoušejte tuto proceduru například pomocí procedury **sysdbclose()** v příkazu EXECUTE PROCEDURE.
4. Zrušením nastavení proměnné prostředí IFX_NODBPROC povolte databázovému serveru, aby spouštěl procedury **sysdbopen()** a **sysdbclose()**.

Příklady

Následující procedura nastavuje roli a prioritu paralelního databázového dotazu pro určitého uživatele:

```
create procedure oltp_user.sysdbopen()  
    set role to oltp;  
    set pdqpriority 5;  
end procedure;
```

Následující procedura nastavuje roli the prioritu paralelního databázového dotazu pro skupinu PUBLIC:

```
create procedure public.sysdbopen()  
    set role to others;  
    set pdqpriority 1;  
end procedure
```

Další informace o procedurách **sysdbopen()** a **sysdbclose()** naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

Provádění rutinních úloh administrace

V závislosti na potřebách organizace může být správce serveru odpovědný za provádění periodických úloh popsaných v následujících odstavcích. Ne každá z těchto úloh je vhodná pro každou instalaci. Pokud je například databázový server k dispozici 24 hodin denně, 7 dní v týdnu, nebude pravděpodobně uváděn do režimu offline, změna provozního režimu databázového serveru tedy nebude rutinní úlohou.

Změna režimů databázového serveru

Administrátor databázového serveru je odpovědný za spouštění a vypínání databázového serveru změnou jeho provozního režimu. Část “Provozní režimy databázového serveru” na stránce 4-7 objasňuje, jak měnit provozní režimy databázového serveru.

Zálohování dat a souborů logického protokolu

Chcete-li zajistit, abyste v případě poruchy mohli databázi obnovit, měli byste často zálohovat paměťové prostory a logické protokoly. Zálohy vytvořené programem ON-Bar můžete rovněž prověřit obslužným programem **archecker**.

Intervaly zálohování paměťových prostorů jsou závislé na četnosti aktualizace dat a na míře jejich důležitosti. Harmonogram zálohování může obsahovat úplnou zálohu (úroveň 0) jednou týdně, přírůstkové zálohy (úroveň 1) jednou denně a zálohy úrovně 2 každou hodinu. Zálohování úrovně 0 musíte provést též po dokončení některých úloh administrace, jako jsou přidání prostoru dbspace, odstranění souboru logického protokolu nebo aktivace zrcadlení.

Soubor logického protokolu zálohujte ihned po jeho zaplnění. Tyto soubory lze zálohovat ručně nebo automaticky. Více informací o použití obslužných programů ON-Bar a **ontape** naleznete v příručce *IBM Informix Backup and Restore Guide*.

Monitorování aktivity

Architektura databázového serveru IBM Informix umožňuje monitorování všech aspektů serveru. Část “Monitorování aktivity databázového serveru” na stránce 1-23 obsahuje popis dostupných informací, pokyny k získání informací a doporučení k jejich používání.

Kontrola konzistence

Měli byste provádět příležitostné kontroly konzistence dat. Další informace o těchto úlohách uvádí Kapitola 23, “Kontrola konzistence”, na stránce 23-1.

Provádění dalších úloh administrace

Tato část pojednává o různých úlohách administrace, které se vyskytují při správě provozního databázového serveru.

Použití zrcadlení

Když používáte zrcadlení, databázový server zapisuje data na dvě různá umístění. Zrcadlení snižuje riziko ztráty dat zapříčiněné selháním paměťového zařízení. Stanou-li se zrcadlená data nedostupná, ať už důvod jakýkoliv, zrcadlo těchto dat je uživatelům okamžitě a transparentně k dispozici. Informace o zrcadlení uvádí Kapitola 17, “Zrcadlení”, na stránce 17-1. Pokyny týkající se úloh souvisejících se zrcadlením dat uvádí Kapitola 18, “Použití zrcadlení”, na stránce 18-1.

Důležité: Měli byste zrcadlit kritické prostory *dbspaces*, které obsahují soubory logického protokolu, fyzický protokol a kořenový prostor *dbspace*.

Správa stavu protokolování databáze

Můžete určit, zda má databáze ve výchozím nastavení používat protokolování transakcí, zda výchozí režim protokolování jednotlivých databází má nebo nemá používat vyrovnávací paměť a zda má režim protokolování být kompatibilní se standardem ANSI.

V databázi s protokolováním můžete vytvořit následující typy tabulek:

- STANDARD
- TEMP
- RAW

Další informace uvádí část “Dočasné tabulky” na stránce 9-26 a Kapitola 11, “Protokolování”, na stránce 11-1. Další informace o změnách voleb protokolování uvádí Kapitola 12, “Správa režimu protokolování databáze”, na stránce 12-1.

Správa logického protokolu

Databázový server obsahuje několik souborů nazývaných *logické protokoly*, které zaznamenávají transakce dat a administrativní údaje, jako jsou záznamy kontrolních bodů a přidání a odstranění bloků.

Mezi typické úlohy administrace logického protokolu patří zálohování souborů logického protokolu, přidávání, uvolňování a změna velikostí souborů logického protokolu a určování horních mezí (high-watermarks).

Databázový server v režimu online dynamicky přiděluje soubory logického protokolu, aby zabránil zahlcení databázového serveru dlouhými transakcemi.

Další informace uvádí část “Parametry protokolování” na stránce 2-3 a Kapitola 13, “Logický protokol”, na stránce 13-1. Pokyny k vytváření a úpravám konfigurace logického protokolu uvádí část Kapitola 14, “Správa souborů logických protokolů”, na stránce 14-1. Informace o zálohování logického protokolu naleznete v příručce *IBM Informix Backup and Restore Guide*.

Správa fyzického protokolu

Můžete měnit velikost a umístění fyzického protokolu. Další informace o fyzickém protokolu uvádí část “Parametry fyzického protokolu” na stránce 2-3, Kapitola 15, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 15-1 a Kapitola 16, “Správa fyzického protokolu”, na stránce 16-1.

Databázový server při spuštění kontroluje, zda je fyzický protokol prázdný, protože to znamená, že vypnutí proběhlo řízeně a tedy řádně. Pokud fyzický protokol *není* prázdný, databázový server automaticky provede akci nazývanou *rychlá obnova*. Rychlá obnova automaticky obnoví databáze do stavu fyzické a logické konzistence po selhání systému, v jehož důsledku mohly zůstat některé transakce nedokončeny.

Správa sdílené paměti

Ke správě sdílené paměti patří následující úlohy:

- Změna velikosti nebo počtu vyrovnávacích pamětí (změnou velikosti vyrovnávací paměti logického nebo fyzického protokolu, nebo změnou počtu vyrovnávacích pamětí ve společné oblasti vyrovnávacích pamětí).
- Změna hodnot parametrů sdílené paměti, je-li to zapotřebí.
- Změna vynucené rezidence (zapnout nebo vypnout, dočasně nebo pro tuto relaci).
- Ladění intervalu kontrolních bodů.
- Přidání segmentů k virtuální části sdílené paměti.
- Použití mezipaměti příkazů jazyka SQL ke snížení využití paměti a doby potřebné k přípravě dotazů.

Informace o tom, jak databázový server využívá sdílenou paměť, uvádí Kapitola 7, “Sdílená paměť”, na stránce 7-1. Další informace uvádí část “Parametry sdílené paměti” na stránce 2-5 a Kapitola 8, “Správa sdílené paměti”, na stránce 8-1.

Správa virtuálních procesorů

Konfigurace a správa virtuálních procesorů (VP) má přímý vliv na výkon databázového serveru. Optimální počet a výběr virtuálních procesorů pro konkrétní databázový server závisí na hardwaru a typech aplikací podporovaných databázovým serverem.

Informace o virtuálních procesorech uvádí Kapitola 5, “Virtuální procesory a jednotkové procesy”, na stránce 5-1. Další informace uvádí část “Parametry virtuálních procesorů” na stránce 2-7 a Kapitola 6, “Správa virtuálních procesorů”, na stránce 6-1.

Správa paralelních databázových dotazů

Můžete řídit zdroje, které databázový server využívá ke zpracování paralelních dotazů pro podporu rozhodování. Musíte vyvážit požadavky dotazů pro podporu rozhodování a požadavky dotazů zpracování transakcí online (OLTP). Mezi zdroje, které byste měli vzít v úvahu, patří sdílená paměť, jednotkové procesy, prostor pro dočasné tabulky a šířku pásma potřebnou pro prohledávání. Další informace o paralelním dotazování databáze (PDQ) a vlivu fragmentace paměťového prostoru na výkon PDQ naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*. Informace o konfiguračních parametrech naleznete v části “Parametry podpory rozhodování” na stránce 2-6.

Použití replikace dat

Pojem *replikace dat* označuje proces reprezentace objektů databáze na více než jednom samostatném místě. Konfigurace replikace dat se skládá z primárního serveru a jednoho nebo více sekundárních serverů, například ze sekundárního serveru replikace HDR, jednoho nebo více sekundárních serverů SD a jednoho nebo více sekundárních serverů RS. Kromě toho server Dynamic Server podporuje replikaci *IBM Informix Enterprise Replication* (ER). Na jednom databázovém serveru můžete kombinovat replikaci dat a replikaci Enterprise Replication. Další informace naleznete v příručce *IBM Informix Dynamic Server Enterprise Replication Guide*.

Prostředí replikace dat

Primární server spolu s replikací HDR podporuje synchronní replikaci celé databáze na sekundární databázový server, čímž poskytuje okamžitou náhradu v případě vážné poruchy počítače. Více informací uvádí část “Parametry replikace High-Availability Data-Replication” na stránce 2-8, Kapitola 19, “Přehled replikace dat (vydání Enterprise/Workgroup)”, na stránce 19-1 a Kapitola 20, “Použití replikace HDR (verze pro podniky/pracovní skupiny)”, na stránce 20-1.

Primární server spolu se sekundárním serverem SD poskytuje přístup jen pro čtení k datům na diskovém poli, která jsou sdílena s primárním serverem. Primární server spolu se sekundárním serverem RS podporuje asynchronní replikaci databáze na primárním serveru. Primární server spolu s libovolnou kombinací serverů replikace HDR, sekundárních serverů RS a sekundárních serverů SD může existovat v prostředí replikace dat. Další informace naleznete v částech Kapitola 21, “Použití serverů RS (remote standalone) (vydání Enterprise)”, na stránce 21-1 a Kapitola 22, “Použití sekundárních serverů se sdíleným diskem (vydání Enterprise)”, na stránce 22-1.

Replikace ER (Enterprise Replication)

Replikace Enterprise Replication podporuje asynchronní replikaci dat na geograficky oddělené databázové servery a umožňuje replikaci jak celé databáze, tak menších částí databáze a tabulek. Replikace Enterprise Replication však poskytuje jen malou podporu uživatelských typů dat. Více informací o replikaci Enterprise Replication naleznete v příručce *IBM Informix Dynamic Server Enterprise Replication Guide*.

Použití auditu

Máte-li v úmyslu provádět auditování databázového serveru, musíte určit, kam budou ukládány záznamy o sledovaných událostech, jak ošetřit chybové stavy a podobně. Můžete také chtít změnit způsob auditu uživatelů, podezíráte-li je, že zneužívají svá přístupová oprávnění. Další informace o úlohách souvisejících s auditem naleznete v části “Parametry prověřování (UNIX)” na stránce 2-10 a v příručce *IBM Informix Security Guide*.

Použití distribuovaných dotazů

Databázový server umožňuje zadávat dotazy na více databázích na několika databázových serverech stejné instance databázového serveru (distribuované dotazy na mnoha databázích) a v několika instancích databázového serveru (distribuované dotazy na mnoha serverech) a umožňuje také tyto databáze aktualizovat. Tento typ dotazování se nazývá *distribuovaný dotaz*. Tento pojem není omezen na příkazy SELECT a často se obecněji týká operace DML nebo provedení rutiny, která vrací objekty mimo lokální databázi nebo na ně odkazuje. V distribuovaných dotazech na mnoha serverech mohou být zúčastněné databázové servery umístěny na jednom hostitelském počítači, ve stejné síti nebo k nim lze přistupovat prostřednictvím jedné brány.

Více informací o distribuovaných dotazech naleznete v příručce *IBM Informix Database Design and Implementation Guide* a v příručce *IBM Informix Guide to SQL: Tutorial*.

Použití globálních transakcí

Globální transakce je transakce, která zahrnuje více než jeden databázový server. Databázové servery IBM Informix podporují dva typy globálních transakcí:

- transakce TP/XA se správcem transakcí
- dvoufázové potvrzování

System Informix využívá *protokol dvoufázového potvrzování*, který má zajistit, aby byly distribuované dotazy shodně potvrzeny nebo odvolány ve více databázových serverech současně. Další informace uvádí Kapitola 24, “Protokoly vícefázového potvrzování”, na stránce 24-1.

Použití správce transakcí

Programy typu *správce transakcí* slouží ke správě terminálů a k obnovení dat. Knihovna TP/XA umožňuje komunikaci mezi správcem transakcí získaným od dodavatele a databází IBM Informix v prostředí X/Open. Další informace naleznete v příručce *IBM Informix TP/XA Programmer's Manual* a v části “Správci transakcí” na stránce 24-2.

Monitorování aktivity databázového serveru

Informace o aktivitě databázového serveru můžete získat z následujících zdrojů.

Zdroj informací	UNIX	Windows
Alarmy událostí	X	X
IBM Informix Server Administrator (ISA)	X	X
protokol zpráv	X	X
ON-Monitor	X	
Obslužný program oncheck	X	X
Obslužný program onperf	X	
Obslužný program onstat	X	X
Tabulky SMI	X	X
Systémová konzole	X	X
Prohlížeč událostí systému Windows		X
Sledování výkonu systému Windows		X

Následující části popisují každý z uvedených zdrojů.

Alarmy událostí

K nahlášení situací vyžadujících okamžitou pozornost používá databázový server funkci alarmu událostí. Chcete-li používat funkci alarm událostí, nastavte konfigurační parametr ALARMPROGRAM na úplný název cesty ke spustitelnému souboru, který provede všechny potřebné administrativní akce.

Více informací naleznete v příloze věnované alarmu událostí a v kapitole věnované konfiguračním parametrům v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Program IBM Informix Server Administrator (ISA)

Program ISA je nástroj založený na prohlížeči, který zajišťuje systémovou administraci celé řady databázových serverů IBM Informix. Program ISA poskytuje přístup téměř ke všem funkcím databázového serveru IBM Informix dostupným z příkazového řádku.

Další informace o programu ISA naleznete v nápovědě online k programu ISA.

Protokol zpráv

Protokol zpráv databázového serveru je soubor operačního systému. Zprávy obsažené v protokolu zpráv databázového serveru obvykle nevyžadují okamžitý zásah. K nahlášení

situaci vyžadujících okamžitou pozornost používá databázový server funkci alarmu událostí. Více informací naleznete v části “Alarmy událostí” na stránce 1-23. Protokol zpráv můžete zobrazit v programu ISA.

Určení cíle pro zprávy protokolu zpráv

K určení názvu cesty protokolu zpráv použijte nastavení konfiguračního parametru MSGPATH. Změny konfiguračního parametru MSGPATH nabývají účinku po vypnutí a restartu databázového serveru. Více informací o konfiguračním parametru MSGPATH naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Monitorování protokolu zpráv

Protokol zpráv byste měli kontrolovat jednou nebo dvakrát denně, aby bylo zajištěno, že zpracování probíhá normálně a že události jsou protokolovány podle očekávání. Pomocí příkazu **onstat -m** získáte název protokolu zpráv a 20 nejaktuálnějších záznamů. K zobrazení celého protokolu zpráv použijte textový editor. Chcete-li zobrazit zprávy tak, jak se objevují, použijte příkaz operačního systému (například příkaz operačního systému UNIX **tail -f**).

Monitorujte velikost protokolu zpráv, protože databázový server přidává k tomuto souboru nové záznamy. Protokol podle potřeby upravte nebo ho zálohujte na pásku a odstraňte.

Dojde-li k poruše databázového serveru, poslouží protokol zpráv jako protokol sledovaných událostí, v němž lze znovu vystopovat události, které později vyústily v nepředvídatelný problém. Databázový server poskytuje v protokolu zpráv mnohdy přesnou povahu problému i doporučení nápravné akce.

Chcete-li zachytit události, které předcházely vzniklým potížím, můžete v protokolu zpráv minutu po minutě číst záznam zpracování databázového serveru. Tento druh monitorování však není nutný.

Více informací naleznete v kapitole věnované zprávám v příručce *IBM Informix Dynamic Server Administrator's Reference*. Další informace naleznete také v příručce *IBM Informix Error Messages*.

Obslužný program ON-Monitor (UNIX)

Obslužný program ON-Monitor poskytuje snadný způsob, jak monitorovat celou řadu aspektů databázového serveru. Většina monitorovacích funkcí je k dispozici v nabídce **Status**. Další informace naleznete v kapitole věnované obslužnému programu ON-Monitor v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Obslužný program oncheck

Obslužný program **oncheck** zobrazuje informace o konfiguraci a využití disků databáze, jako je například počet stránek použitých pro tabulku, obsah rezervovaných stránek a počet oblastí v tabulce. Další informace o obslužném programu **oncheck** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Nástroj onperf (UNIX)

Databázový server obsahuje grafický nástroj k monitorování nazvaný **onperf**. Tento nástroj dokáže monitorovat většinu metrik poskytovaných obslužným programem **onstat**. Oproti obslužnému programu **onstat** poskytuje následující výhody:

- Hodnoty metriky zobrazuje graficky a v reálném čase.
- Umožní zvolit metriku, která má být monitorována.
- Uloží data metriky z nedávné historie do vyrovnávací paměti v paměti. Tato data jsou k dispozici, chcete-li analyzovat aktuální trend.

- Údaje o výkonu může ukládat do souboru.

Více informací o nástroji **onperf** naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Obslužný program onstat

Obslužný program **onstat** představuje způsob, jak monitorovat sdílenou paměť databázového serveru z příkazového řádku. Obslužný program **onstat** čte data ze sdílené paměti a vytváří statistiky relevantní v momentu, kdy je prováděn příslušný příkaz. Znamená to, že obslužný program **onstat** poskytuje informace, které se během zpracování dynamicky mění, včetně změn týkajících se vyrovnávacích pamětí, zámků, indexů a uživatelů.

Tabulky SMI

Tabulky SMI (*rozhraní pro monitorování systému* - system monitoring interface) jsou speciální tabulky spravované databázovým serverem, které obsahují dynamické údaje o stavu databázového serveru. Můžete v nich používat příkazy SELECT a s jejich pomocí určit téměř jakýkoliv údaj o databázovém serveru, který se chcete dozvědět. Popis tabulek SMI naleznete v kapitole věnované databázi **sysmaster** v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Systémová konzole

Databázový server odesílá zprávy, které jsou užitečné pro administrátora databázového serveru prostřednictvím *systémové konzole*. Chcete-li určit cílový název cesty zpráv konzole, nastavte konfigurační parametr CONSOLE. Více informací o konfiguračním parametru CONSOLE naleznete v kapitole věnované konfiguračním parametrům v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Změny konfiguračního parametru CONSOLE nabydou účinku po vypnutí a restartu databázového serveru.

Jen pro Windows

Administrátor databázového serveru se může přihlásit ke konzole z kteréhokoliv uzlu, aby provedl správu systému a monitorování úloh.

Konec Jen pro Windows

Nástroje operačního systému UNIX

Databázový server využívá operačního systému hostitelského počítače, který poskytuje přístup k systémovým zdrojům, jako jsou jednotky CPU, paměť a různá vstupní-výstupní rozhraní a souborů na discích bez vyrovnávací paměti. Každý operační systém má vlastní sadu obslužných programů, které informují o míře a způsobu využití systémových zdrojů. Různé operační systémy mohou mít monitorovací obslužné programy téhož názvu, ale s odlišnými volbami a odlišným zobrazením příslušných údajů.

Následující tabulka zobrazuje typické obslužné programy pro monitorování zdrojů v operačním systému UNIX. Informace o tom, jak monitorovat zdroje operačního systému, naleznete v administrační příručce operačního systému.

Obslužný program pro systém UNIX	Popis
vmstat	Zobrazuje statistiku virtuální paměti.
iostat	Zobrazuje statistiku využití vstupu - výstupu.
sar	Zobrazuje řadu statistických údajů o zdrojích.
ps	Zobrazuje informace o aktivních procesech.
cron	Umožňuje zachycovat stav systémových zdrojů pomocí systémového plánovače, který v pravidelných intervalech spouští příkazy nebo programy. Můžete použít i jiné nástroje k plánování, které jsou k dispozici v rámci operačního systému.

Prohlížeč událostí v systému Windows

Prohlížeč událostí zobrazuje informační, výstražné a chybové zprávy operačního systému, jiných aplikací a databázového serveru.

Zobrazení zpráv databázového serveru v prostředí Windows:

1. Zvolte **Nástroje pro správu > Prohlížeč událostí**.
2. Zvolte položku **Protokol > Zabezpečení**.
3. Poklepáním na kteroukoliv z událostí zobrazíte podrobnější zprávu.

Sledování výkonu v systému Windows

Program Sledování výkonu v systému Windows (**perfmon.exe**) monitoruje zdroje jako jsou procesor, paměť, mezipaměť, jednotkové procesy a procesy. Sledování výkonu poskytuje též grafy, výstrahy, možnost vytvářet sestavy a ukládat informace do souborů protokolu pro pozdější analýzu.

Chcete-li zobrazit nástroj Sledování výkonu v systému Windows, zvolte položky **Nástroje pro správu > Výkon**.

Obslužné programy pro systém Windows

Následující obslužné programy systému Informix zjednodušují administraci databázového serveru v prostředí systému Windows.

Obslužný program pro systém Windows	Popis a použití
ixpasswd.exe	<p>Změní přihlašovací heslo pro všechny služby, které se přihlašují jako uživatel informix. Heslo lze měnit interaktivně nebo v příkazovém řádku s použitím volby -y. Tento obslužný program ušetří práci spojenou s ruční změnou hesla u každé služby pokaždé, když změníte heslo uživatele informix.</p> <p>Pokud jste lokálně přihlášení a spustíte obslužný program ixpasswd, změní tento heslo pro služby, které se přihlašují jako lokální uživatel informix. Pokud jste přihlášení do <i>domény</i> a spustíte obslužný program ixpasswd, změní tento heslo pro služby, které se přihlašují jako uživatel <i>doména</i>\informix.</p> <p>Použití: <code>ixpasswd [-y nové_heslo]</code></p>

Obslužný program pro systém Windows	Popis a použití
ixsu.exe	<p>Zobrazí okno příkazového řádku spuštěné jako určený uživatel. Uživatel je místním uživatelem, pokud ovšem neurčíte název domény. Neurčíte-li uživatelské jméno, je výchozím uživatelem uživatel informix. Nyní se již nemusíte odhlásit od aktuálního uživatele a znovu se přihlásit jako uživatel informix, abyste mohli provádět úlohy administrátora DBA, které musí být spuštěny uživatelem informix.</p> <p>Obslužný program ixsu vyžaduje rozšířená uživatelská oprávnění:</p> <ul style="list-style-type: none"> • Jednat jako část operačního systému. • Zvýšit kvóty. • Nahradit token úrovně procesu. <p>Chcete-li konfigurovat rozšířená uživatelská oprávnění v systému Windows NT, vyberte položku Správce uživatelů > Zásady > Uživatelská oprávnění a zaškrtněte volbu Rozšířená uživatelská oprávnění. Změníte-li rozšířená uživatelská oprávnění pro stávajícího uživatele, musíte se odhlásit a znovu přihlásit, aby nová oprávnění nabyla platnosti.</p> <p>Použití: <code>ixsu [[doména\]uživatelské_jméno]</code></p>
ntchname.exe	<p>Změní záznamy databázového serveru Dynamic Server v registru ze starého názvu hostitele na nový. Obslužný program ntchname spusíte po změně názvu hostitele. Tento obslužný program nemění uživatelské proměnné prostředí.</p> <p>Po provedení příkazu ntchname upravte soubor <code>%INFORMIXDIR%\%INFORMIXSERVER%.cmd</code> a změňte záznam INFORMIXSQLHOSTS na nový název hostitele.</p> <p>Použití: <code>ntchname starý_název nový_název</code></p>

Nástroj OpenAdmin pro server IDS

Nástroj OpenAdmin pro server IDS je administrační nástroj založený na technologii PHP, který poskytuje možnost spravovat několik instancí databázových serverů z jednoho umístění. Mezi úlohy, které můžete s nástrojem OpenAdmin provádět, patří:

- Definování a správa automatizovaných úloh prostřednictvím rozhraní SQL Administration API
- Vytváření a zobrazování histogramů výkonu pro analýzu a ladění
- Monitorování řešení vysoké dostupnosti, které zahrnují replikaci HDR, sekundární servery se sdílenými disky a vzdálené samostatné sekundární servery.

Do nástroje OpenAdmin můžete zapojit vlastní rozšíření a vytvořit tak funkčnost, kterou potřebujete.

Nástroj OpenAdmin je program s otevřeným zdrojem, který je možné stáhnout z tohoto webového serveru: <http://www-306.ibm.com/software/data/informix/downloads.html>.

Kapitola 2. Konfigurační parametry

Obsah kapitoly	2-1
Parametry identifikace databázového serveru	2-1
Parametry diskového prostoru	2-2
Kořenový prostor dbspace	2-2
Zrcadlení kořenového prostoru dbspace	2-2
Další parametry správy prostorů	2-3
Parametry protokolování	2-3
Logický protokol	2-3
Parametry fyzického protokolu	2-3
Parametry odvolávání transakcí a obnovy	2-4
Parametry zálohování a obnovy	2-4
Parametry protokolu zpráv	2-4
Parametry sdílené paměti	2-5
Velikost přidělené sdílené paměti	2-5
Přidělený prostor sdílené paměti	2-5
Řízení vyrovnávací sdílené paměti	2-6
Využití mezipaměti příkazů SQL	2-6
Parametry podpory rozhodování	2-6
Parametry procesu databázového serveru	2-7
Parametry virtuálních procesorů	2-7
Časové intervaly	2-8
Parametry obnovení	2-8
Parametry replikace High-Availability Data-Replication	2-8
Parametry alarmů událostí	2-8
Parametry vypisování jádra (UNIX)	2-9
Parametry direktiv	2-9
Parametry připojení	2-9
Parametry související se zabezpečením	2-10
Zvláštní parametry	2-10
Parametry prověřování (UNIX)	2-10
Parametry optických médií	2-11
Parametry systému UNIX	2-11
Monitorování informací o konfiguraci	2-11

Obsah kapitoly

Tato kapitola obsahuje přehled konfiguračních parametrů souboru ONCONFIG používaných databázovým serverem a popisuje různé způsoby monitorování konfiguračních informací. Kapitola vám pomůže určit, které konfigurační parametry jsou v konkrétním prostředí klíčové a které lze naopak nastavit později při ladění výkonu databázového serveru. Podrobnosti o jednotlivých parametrech naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Parametry identifikace databázového serveru

K jednoznačné identifikaci všech instancí databázového serveru použijte parametry SERVERNUM a DBSERVERNAME.

Konfigurační parametr	Popis
DBSERVERALIASES	Určuje alternativní název nebo názvy instance databázového serveru. Maximální počet aliasů je 32. Informace o vytvoření několika naslouchajících koncových bodů pro klientská připojení pomocí parametru DBSERVERALIASES naleznete v části “Určení jednotkových procesů listen a cyklického dotazování pro připojení klienta k serveru” na stránce 5-23. Upozornění: Neměňte konfigurační parametr DBSERVERNAME bez restartování databázového serveru.
DBSERVERNAME	Určuje jedinečný název instance databázového serveru. Konfigurační parametr DBSERVERNAME použijte pro proměnnou prostředí INFORMIXSERVER a v souboru sqlhosts .
SERVENUM	Určuje jedinečné číslo instance databázového serveru. Databázový server používá parametr SERVENUM k určení adres segmentů sdílené paměti.

Parametry diskového prostoru

Parametry diskového prostoru řídí způsob, jakým databázový server spravuje diskový prostor.

Kořenový prostor dbspace

První přidělený paměťový prostor se nazývá *kořenový databázový prostor* nebo *kořenový prostor dbspace*. V tomto prostoru se ukládají všechny základní údaje, které popisují databázový server. K popisu kořenového prostoru dbspace použijte následující parametry.

Konfigurační parametr	Popis
ROOTNAME	Určuje název kořenového prostoru dbspace. Jako hodnotu parametru ROOTNAME můžete zadat libovolný popisný název, ale obvykle se kořenový prostor nazývá rootdbs . Další informace naleznete v části “Kořenový prostor dbspace” na stránce 9-11.
ROOTOFFSET	Zadává posun. Informace o tom, kdy je třeba zadat parametr ROOTOFFSET naleznete v části “Určení posunu” na stránce 10-4.
ROOTPATH	Určuje název cesty k paměti přidělené kořenovému prostoru dbspace. Informace o způsobu volby a přidělování paměti naleznete v části “Přidělení diskového prostoru” na stránce 10-3.
ROOTSIZE	Určuje velikost prostoru přiděleného kořenovému prostoru dbspace. Informace o výběru vhodné velikosti kořenového prostoru dbspace naleznete v části “Velikost kořenového prostoru dbspace” na stránce 9-32.
TBLTBLFIRST	Určuje velikost první oblasti prostoru tblspace tblspace v kilobajtech.
TBLTBLNEXT	Určuje velikost další oblasti prostoru tblspace tblspace v kilobajtech.

Zrcadlení kořenového prostoru dbspace

Zrcadlení umožňuje rychlou obnovu po selhání disku, kdy server zůstává v režimu online. Pokud je zrcadlení aktivní, jsou data ukládána současně na dva disky. Dojde-li k selhání jednoho disku, data jsou stále dostupná na druhém disku. K popisu způsobu zrcadlení kořenového prostoru dbspace použijte následující parametry.

Konfigurační parametr	Popis
MIRROR	Určuje, zda je zrcadlení zakázáno nebo povoleno. Další informace naleznete v části Kapitola 18, “Použití zrcadlení”, na stránce 18-1.
MIRRORPATH	Určuje úplný název cesty k bloku, který je zrcadlem počátečního bloku kořenového prostoru dbspace.
MIRROROFFSET	Určuje posun v rámci zařízení, které slouží jako zrcadlo počátečního bloku kořenového prostoru dbspace. Další informace naleznete v části “Určení posunu” na stránce 10-4.

Další parametry správy prostorů

Pomocí následujících parametrů určete, jak má databázový server spravovat konkrétní typy diskového prostoru.

Konfigurační parametr	Popis
DBSPACETEMP	Určuje seznam prostorů dbspace, které databázový server může použít jako paměť pro dočasné tabulky. Další informace naleznete v části "Vytvoření dočasného prostoru dbspace" na stránce 10-17.
FILLFACTOR	Určuje míru zaplnění stránek indexů při vytváření indexů. Další informace naleznete v příručce <i>Příručka výkonosti serveru IBM Informix Dynamic Server</i> .
ONDBSPACEDOWN	Určuje, jak má databázový server pracovat ze zakázaným prostorem dbspace, který není kritický pro provoz serveru.
SBSPACENAME	Určuje název výchozího prostoru sbspace. Další informace naleznete v části "Řízení umístění ukládaných dat" na stránce 9-9.
SBSPACETEMP	Určuje název výchozího dočasného prostoru sbspace. Další informace naleznete v části "Dočasné prostory sbspace" na stránce 9-19.
SYSSBSPACENAME	Určuje název prostoru sbspace, do kterého databázový server ukládá statistické údaje shromážděné příkazem UPDATE STATISTICS pro některé uživatelem definované datové typy.

Parametry protokolování

Pomocí parametrů protokolování můžete řídit vytváření logických a fyzických protokolů.

Logický protokol

Logický protokol obsahuje záznamy o změnách provedených v instanci databázového serveru. Záznamy logického protokolu se používají k odvolávání transakcí, k obnově po selhání systému a podobně. Následující parametry ovlivní logické protokolování.

Konfigurační parametr	Popis
DYNAMIC_LOGS	Určuje, zda má databázový server přidělovat nové soubory logického protokolu automaticky. Další informace naleznete v části Kapitola 13, "Logický protokol", na stránce 13-1.
LOGBUFF	Určuje množství sdílené paměti rezervované pro vyrovnávací paměti, které udržují záznamy logického protokolu před vyprázdněním na disk. Informace o ladění vyrovnávací paměti logického protokolu naleznete v části "Vyrovnávací paměť logického protokolu" na stránce 7-11.
LOGFILES	Určuje počet souborů logického protokolu používaných k ukládání záznamů logického protokolu před zálohováním záznamů na disk. Další informace naleznete v části "Odhad velikosti a počtu souborů protokolu" na stránce 14-2.
LOGSIZE	Určuje velikost souborů logického protokolu.
LTXHWM	Určuje procento dostupného prostoru logického protokolu, po jehož zaplnění začne databázový server kontrolovat délku transakcí. Další informace naleznete v části "Nastavení horních mezí pro odvolání dlouhých transakcí" na stránce 14-17.
LTXEHWM	Určuje bod, od kterého odvolávaná dlouhá transakce získá výhradní přístup k logickému protokolu.
parametr TEMPTAB_NOLOG	Zakazuje protokolování dočasných tabulek.

Parametry fyzického protokolu

Fyzický protokol obsahuje obrazy všech stránek (paměťových jednotek), které byly od posledního kontrolního bodu změněny. Kombinace logického protokolu a fyzického protokolu umožňuje rychlou obnovu po selhání systému. K popisu fyzického protokolu použijte následující parametry.

Konfigurační parametr	Popis
PHYSBUFF	Určuje množství sdílené paměti rezervované pro vyrovnávací paměti stránek, které mají být změněny.
PHYSDBS	Určuje název prostoru typu dbspace, do kterého bude uložen fyzický protokol v okamžiku, kdy je poprvé vytvořen databázový server.
PHYSFILE	Určuje velikost počátečního fyzického protokolu při prvním vytvoření databázového serveru. Po uvedení databázového serveru do režimu online a inicializaci diskového prostoru pomocí příkazu oninit -i použijte obslužný program onparams a změňte umístění logického protokolu a jeho velikost.

Parametry odvolávání transakcí a obnovy

Informace o používání následujících parametrů ovlivňujících obnovu a odvolávání transakcí naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*. Informace naleznete také v části Kapitola 15, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 15-1.

Konfigurační parametr	Popis
konfigurační parametr RTO_SERVER_RESTART	Umožňuje použít cílovou dobu obnovy (RTO k nastavení doby, v sekundách, během které musí dynamický server provést obnovu poté, co ho restartujete, a uvede server do režimu online nebo do klidového režimu.

Parametry zálohování a obnovy

Pomocí obslužných programů ON-Bar nebo **ontape** můžete vytvářet zálohy logických protokolů a paměťových prostorů dat databázového serveru. K ověření záloh paměťových prostorů použijte nástroj ON-Bar. Další informace o programech ON-Bar a **ontape** naleznete v příručce *IBM Informix Backup and Restore Guide*.

Pokud používáte obslužný program **ontape**, popište pásková zařízení pomocí následujících parametrů. Chcete-li využít úplnou fyzickou kapacitu pásky, nastavte čtení a zápis do konce média nastavením parametrů TAPESIZE a LTAPESIZE na hodnotu 0.

Konfigurační parametr	Popis
TAPEDEV	Určuje pásková zařízení.
TAPEBLK	Určuje velikost bloku páskového zařízení.
TAPESIZE	Určuje maximální objem dat, který má být zapsán na každou pásku.
LTAPEDEV	Určuje pásková zařízení.
LTAPEBLK	Určuje velikost bloku páskového zařízení.
LTAPESIZE	Určuje maximální objem dat, který má být zapsán na každou pásku.

Parametry protokolu zpráv

Soubory zpráv poskytují informace o provozu databázového serveru.

Konfigurační parametr	Popis
CONSOLE (systém UNIX)	Určuje název cesty pro zprávy vypisované do konzoly. Další informace naleznete v části “Systémová konzole” na stránce 1-25.
MSGPATH	Určuje název cesty k souboru protokolu zpráv databázového serveru. Další informace naleznete v části “Protokol zpráv” na stránce 1-23.

Parametry sdílené paměti

Parametry sdílené paměti ovlivňují výkon databázového serveru.

Velikost přidělené sdílené paměti

Pomocí následujících parametrů zadejte způsob přidělování sdílené paměti databázového serveru a umístění této paměti.

Konfigurační parametr	Popis
EXTSHMADD	Určuje velikost přidaného rozšiřujícího segmentu.
SHMADD	Určuje přírůstek sdílené paměti, přidělovaný v okamžiku, kdy databázový server požádá o další paměť.
SHMBASE	Určuje adresu počátku sdílené paměti. Tento parametr je závislý na počítači. Hodnota parametru závisí na platformě a také na tom, zda je procesor 32bitový nebo 64bitový. Informace o potřebné hodnotě parametru SHMBASE naleznete v poznámkách k počítači.
SHMTOTAL	Určuje maximální množství sdílené paměti, které smí databázový server používat.
SHMVIRTSIZE	Určuje velikost prvního úseku sdílené paměti, ke kterému se databázový server připojí.

Další informace o těchto parametrech naleznete v části Kapitola 7, “Sdílená paměť”, na stránce 7-1.

Informace o těchto parametrech konfigurace sdílené paměti databázového serveru specifické pro danou platformu naleznete v souboru Poznámky k počítači v systému UNIX a v souboru Poznámky k verzi v systému Windows.

Přidělený prostor sdílené paměti

Pomocí těchto parametrů zadejte způsob přidělování prostoru ve sdílené paměti.

Konfigurační parametr	Popis
BUFFERPOOL	Určuje informace o společné oblasti vyrovnávací paměti, které je třeba definovat pro každou jednotlivou velikost stránky používanou prostorem dbSPACE. Další informace naleznete v části “Vytvoření prostoru dbSPACE s jinou než výchozí velikostí stránky” na stránce 10-12. Informace, které byly před verzí 10.0 určovány pomocí konfiguračních parametrů BUFFERS, LRUS, LRU_MAX_DIRTY a LRU_MIN_DIRTY, jsou nyní určovány pomocí konfiguračního parametru BUFFERPOOL.
CKPTINTVL	Určuje maximální časový interval, který smí uplynout před dosažením kontrolního bodu, pokud není nastaven konfigurační parametr RTO_SERVER_RESTART, který automaticky zapne optimalizaci kontrolního bodu. Další informace naleznete v části Kapitola 15, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 15-1.
DD_HASHMAX	Určuje maximální počet položek každého sektoru hashovací tabulky v mezipaměti datového slovníku. Další informace o nastavení parametru DD_HASHMAX naleznete v příručce <i>Řízení výkonu systému IBM Informix</i> .
DD_HASHSIZE	Určuje počet sektorů hashovací tabulky v mezipaměti datového slovníku. Další informace o nastavení parametru DD_HASHSIZE naleznete v příručce <i>Řízení výkonu systému IBM Informix</i> .
DEF_TABLE_LOCKMODE	Nastaví režim uzamykání nových tabulek na stránkový nebo řádkový. Další informace naleznete v příručce <i>IBM Informix Guide to SQL: Tutorial</i> .
LOCKS	Určuje počáteční počet zámků dostupných pro uživatelské procesy databázového serveru při zpracování transakcí. Další informace naleznete v části Kapitola 8, “Správa sdílené paměti”, na stránce 8-1 a v příručce <i>Příručka výkonosti serveru IBM Informix Dynamic Server</i> .

Konfigurační parametr	Popis
PC_POOLSIZE	Určuje počet rutin UDR (rutin jazyka SPL i externích rutin), které lze uložit v mezipaměti rutin URD. Další informace o nastavení parametru PC_POOLSIZE naleznete v příručce <i>Příručka výkonosti serveru IBM Informix Dynamic Server</i> .
PC_HASHSIZE	Určuje počet sektorů hashovací tabulky v mezipaměti rutin UDR. Další informace o nastavení parametru PC_HASHSIZE naleznete v příručce <i>Příručka výkonosti serveru IBM Informix Dynamic Server</i> .
RESIDENT	Určuje, zda má být vynucena rezidence sdílené paměti. Další informace naleznete v části Kapitola 8, “Správa sdílené paměti”, na stránce 8-1 a v příručce <i>Příručka výkonosti serveru IBM Informix Dynamic Server</i> .
STACKSIZE	Určuje velikost zásobníku uživatelských jednotkových procesů databázového serveru. Informace týkající se používání zásobníků naleznete v části “Zásobníky” na stránce 7-18.

Řízení vyrovnávací sdílené paměti

K řízení společné oblasti sdílených vyrovnávacích pamětí použijte následující parametry.

Konfigurační parametr	Popis
CLEANERS	Řídí počet jednotkových procesů, které vyprazdňují stránky na disk a vracejí stránky do společné oblasti sdílené paměti. Další informace naleznete v části “Vyprazdňování dat na disk” na stránce 7-26.
RA_PAGES a RA_THRESHOLD	Řídí počet diskových stránek, které databázový server při sekvenčním prohledávání načítá v předstihu. Další informace naleznete v části “Konfigurace dopředného čtení databázového serveru” na stránce 7-26.

Využití mezipaměti příkazů SQL

Ke konfiguraci mezipaměti příkazů jazyka SQL použijte následující parametry. Další informace naleznete v části “Nastavení parametrů mezipaměti příkazů jazyka SQL” na stránce 8-6.

Konfigurační parametr	Popis
STMT_CACHE	Zapne, povolí nebo zakáže mezipaměť příkazů jazyka SQL v paměti. Pokud je parametr zapnut, určuje, zda má mezipaměť příkazů jazyka SQL uchovávat analyzované a optimalizované příkazy SQL.
STMT_CACHE_SIZE	Určuje velikost mezipaměti příkazů jazyka SQL.
STMT_CACHE_HITS	Určuje počet použití příkazu (odkazu na příkaz), po kterém bude příkaz zcela vložen do mezipaměti příkazů jazyka SQL.
STMT_CACHE_NOLIMIT	Řídí, zda mají být příkazy jazyka SQL vkládány do mezipaměti příkazů i po překročení velikosti určené hodnotou parametru STMT_CACHE_SIZE.
STMT_CACHE_NUMPOOL	Definuje počet společných oblastí mezipaměti příkazů jazyka SQL.

Parametry podpory rozhodování

Pokud v systému konfiguruje virtuální sdílenou paměť, je zapotřebí rozhodnout, jak velká část paměti bude vyhrazena dotazům pro podporu rozhodování. Dotazy pro podporu rozhodování využívají k provádění operací spojení a řazení značný objem virtuální části sdílené paměti.

Pomocí následujících parametrů můžete řídit způsob zpracování dotazů pro podporu rozhodování a množství paměti, které databázový server pro tyto dotazy bude používat. Další

informace o ladění těchto konfiguračních parametrů naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Konfigurační parametr	Popis
DATASKIP	Řídí, zda má databázový server přeskakovat nedostupné fragmenty tabulek.
DS_MAX_QUERIES	Určuje maximální počet dotazů, které mohou být spuštěny souběžně.
DS_MAX_SCANS	Omezuje počet jednotkových procesů prohledávání pro paralelní databázové dotazy (PDQ), které může databázový server provádět souběžně.
DS_TOTAL_MEMORY	Určuje velikost paměti, která je dostupná pro dotazy PDQ. Nastavte konfigurační parametr DS_TOTAL_MEMORY na jakoukoli hodnotu nepřekračující (SHMVRTSIZE - 10 MB).
DS_NONPDQ_QUERY_MEM	Umožňuje zvýšit velikost paměti dostupné jiným dotazům, než jsou dotazy PDQ.
parametr MAX_PDQPRIORITY	Omezuje množství zdrojů, které může dotaz využívat.
OPTCOMPIND	Doporučuje optimalizátoru strategii spojování, která je vhodná pro konkrétní aplikaci.

Parametry procesu databázového serveru

Konfigurační parametry procesů databázového serveru popisují typy procesorů v počítači a určují chování virtuálních procesorů.

Parametry virtuálních procesorů

Pomocí následujících parametrů určete typ procesorů ve vašem prostředí a přiřadíte virtuální procesory.

Podle počtu procesorů použité platformy je třeba nastavit následující parametry na konkrétní hodnoty:

- AUTO_AIOVPS
- MULTIPROCESSOR
- NETTYPE
- SINGLE_CPU_VP
- VPCLASS

Pravidla nastavování těchto parametrů naleznete v části “Nastavení konfiguračních parametrů virtuálních procesorů” na stránce 6-1.

Konfigurační parametr	Popis
parametr AUTO_AIOVPS	Povoluje nebo zakazuje schopnost databázového serveru automaticky zvýšit počet AIO VPS a podprocesů vyprazdňovače v případě, kdy server zjistí, že AIO VPs nestačí pracovnímu zatížení vstupu - výstupu.
MULTIPROCESSOR	Určuje vhodný typ zamykání.
NETTYPE	Umožňuje ladit jednotlivé komunikační protokoly.
SINGLE_CPU_VP	Určuje, že databázový server používá jediný procesor a umožňuje tak databázovému serveru optimalizovat svou činnost v tomto prostředí.
VPCLASS	Určuje třídu virtuálních procesorů, počet virtuálních procesorů, které má databázový server spouštět, maximální povolený počet procesorů, afinitu procesorů a stárnutí priority. Místo parametrů NOAGE, NUMCPUVPS, NUMAIOVPS, AFF_NPROCS a AFF_SPROCS byste měli používat parametr VPCLASS. Další informace naleznete v části “Určení parametru VPCLASS” na stránce 6-2.

Přidáváte-li nové ukládací prostory typu dbspace, blobspace a sbspace, databázový server automaticky upraví počet AIO VP a podprocesů vyčištění, aby splnil požadavky nového uložště.

Časové intervaly

Pomocí následujících parametrů můžete řídit časové intervaly, které databázový server používá při zpracování transakcí.

Konfigurační parametr	Popis
DEADLOCK_TIMEOUT	Určuje čas, po který bude databázový server čekat na zdroje sdílené paměti při distribuované transakci.
HETERO_COMMIT	Určuje, zda má databázový server používat transakce s heterogenním potvrzováním.
TXTIMEOUT	Určuje, jak dlouho má při dvoufázovém potvrzování účastník čekat na příjem instrukce <i>commit</i> .
USEOSTIME	Řídí granularitu časových údajů ve zprávách databázového serveru.

Parametry obnovení

Pomocí následujících parametrů můžete řídit počet jednotkových procesů, které databázový server přiděluje logickému obnovení online a offline. Další informace naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Konfigurační parametr	Popis
OFF_RECOVERY_THREADS	Určuje počet jednotkových procesů obnovení, které server používá při studeném obnovení.
ON_RECOVERY_THREADS	Určuje počet jednotkových procesů obnovení, které server používá při rychlé obnově a při teplém obnovení.

Parametry replikace High-Availability Data-Replication

Pomocí parametrů replikace High-Availability Data-Replication (HDR) můžete řídit chování dvojice serverů HDR. Další informace naleznete v části Kapitola 19, “Přehled replikace dat (vydání Enterprise/Workgroup)”, na stránce 19-1.

Konfigurační parametr	Popis
DRIDXAUTO	Určuje, zda má primární server replikace High-Availability Data Replication (HDR) automaticky spustit replikaci indexu, pokud sekundární server HDR zjistí poškození indexu.
DRAUTO	Určuje, jak má sekundární databázový server reagovat na selhání replikace HDR.
DRINTERVAL	Určuje maximální interval (v sekundách) mezi vyprázdněními vyrovnávací paměti replikace dat.
DRLOSTFOUND	Určuje název cesty k souboru, který obsahuje transakce potvrzené v primárním databázovém serveru, ale nepotvrzené v sekundárním databázovém serveru po selhání primárního serveru.
DRTIMEOUT	Určuje dobu v sekundách, po kterou databázový server čeká na potvrzení přenosu druhým databázovým serverem v páru.

Parametry alarmů událostí

Databázový server může spustit program, který provede akci, kdykoli dojde ke vzniku některého z významných alarmů událostí nebo kdykoli dojde ke vzniku jakéhokoli alarmu událostí. Mezi významné alarmy událostí patří selhání databáze, tabulky, indexu, převedení bloku nebo prostoru dbspace do režimu offline, selhání interního subsystému, selhání při inicializaci a zjištění dlouhé transakce. Oznámení o vzniku alarmu události může přijmout pomocí e-mailu nebo pomocí služby pager.

Pomocí následujících parametrů určete:

- Zda má program alarmů událostí pracovat se všemi alarmy událostí nebo jen s některými.
- Jaké akce se mají provést, pokud dojde k výskytu alarmu události.

Konfigurační parametr	Popis
ALRM_ALL_EVENTS	Určuje, zda má být program ALARMPROGRAM spuštěn pro všechny události protokolované v souboru určeném parametrem MSGPATH nebo zda má být spuštěn jen pro určité významné události.
ALARMPROGRAM	Určuje umístění souboru, který bude spuštěn, když dojde ke vzniku alarmu události. Parametr ALARMPROGRAM můžete nastavit tak, aby byly automaticky zálohovány zaplněné protokoly. Další informace naleznete v příručce <i>IBM Informix Backup and Restore Guide</i> .

Další informace naleznete v příručce *Administrator's Reference*.

Parametry vypisování jádra (UNIX)

Pomocí následujících parametrů můžete řídit typy a umístění výpisů jádra, ke kterým dochází v případě selhání databázového serveru. Další informace naleznete v části "Monitorování nekonzistence dat" na stránce 23-3.

Konfigurační parametr	Popis
DUMPCNT	Určuje počet chyb prohlášení, po kterém jednotkový proces vypíše do souboru sdílenou paměť.
DUMPCORE	Řídí, zda má virtuální procesor po chybě prohlášení vypsát do souboru jádro paměti.
DUMPDIR	Určuje adresář, do kterého databázový server ukládá výpisy sdílené paměti, soubory gcore a zprávy o chybách prohlášení.
DUMPGCORE	Pokud použitý operační systém podporuje obslužný program gcore , spustí databázový server po chybě prohlášení obslužný program gcore .
DUMPSHMEM	Určuje, zda má po chybě prohlášení být do souboru vypsána sdílená paměť.

Parametry direktiv

Pomocí následujících konfiguračních parametrů můžete zapnout nebo vypnout direktivy, na které může databázový server narazit.

Konfigurační parametr	Popis
DIRECTIVES	Určuje, zda se má optimalizátor řídit všemi direktivami. Výchozí hodnota 1 znamená, že se optimalizátor bude řídit všemi direktivami. Další informace naleznete v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i> .
EXT_DIRECTIVES	Povolí externí direktivy. Další informace naleznete v příručce <i>IBM Informix Dynamic Server Administrator's Reference</i> a v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i> .

Parametry připojení

Pomocí následujících parametrů můžete konfigurovat propojitelnost, omezit délku časového limitu pro nedokončená připojení, omezit počet nedokončených připojení a snížit tak riziko agresivního útoku DoS (Denial of Service). Další informace naleznete v příručce *IBM Informix Security Guide*.

Konfigurační parametr	Popis
LISTEN_TIMEOUT	Nastavuje délku časového limitu pro nedokončená připojení. Výchozí hodnota je 10 s.
MAX_INCOMPLETE_CONNECTIONS	Omezuje počet nedokončených žádostí o připojení. Výchozí hodnota je 1024.

Parametry související se zabezpečením

Následující parametry související se zabezpečením jsou podrobněji popsány v části *IBM Informix Security Guide*.

Konfigurační parametr	Popis
DBCREATE_PERMISSION	Určitému uživateli poskytuje oprávnění vytvářet databáze; tím zabraňuje vytvářet databáze ostatním uživatelům.
DB_LIBRARY_PATH	Umožňuje administrátorovi databázového serveru (DBSA) zadat seznam platných umístění adresářových předpon oddělených čárkami, ze kterých může databázový server načíst externí moduly.
IFX_EXTEND_ROLE	Umožňuje administrátorovi databázového serveru (DBSA) zabránit neoprávněným uživatelům registrovat uživatelské rutiny (UDR) modulů DataBlade.
parametr SECURITY_LOCALCONNECTION	Umožňuje administrátorovi databázového serveru (DBSA) nastavit kontrolu zabezpečení pro místní připojení se stejným hostitelem.
parametr UNSECURE_ONSTAT	Odstraňuje omezení, které dovolují pouze uživatelům DBA provádět příkazy onstat -ses , onstat -stm , onstat -ssc a onstat -sql .

Následující parametry související se zabezpečením jsou podrobněji popsány v části Šifrování přenosu dat mezi databázovými servery HDR a *IBM Informix Dynamic Server Administrator's Reference*.

Konfigurační parametr	Popis
parametr ENCRYPT_HDR	Umožňuje šifrování HDR.
parametry ENCRYPT_CIPHERS, ENCRYPT_MAC, ENCRYPT_MACFILE a ENCRYPT_SWITCH	Určuje informace pro šifrování HDR.

Zvláštní parametry

Některé parametry se v konfiguračním souboru nacházejí pouze tehdy, pokud používáte zvláštní funkce databázového serveru.

Parametry prověřování (UNIX)

Parametry prověřování použijte pouze tehdy, pokud použitý databázový server podporuje funkce prověřování. Další informace o těchto parametrech naleznete v příručce *IBM Informix Security Guide*.

Konfigurační parametr	Popis
ADTERR	Určuje, jak má databázový server postupovat, pokud dojde k chybě při zapisování záznamu o sledované události.
ADTMODE	Řídí, zda audit akcí uživatelů provádí databázový server nebo operační systém.

Konfigurační parametr	Popis
ADTPATH	Určuje adresář, do kterého databázový server může ukládat soubory auditu.
ADTSIZE	Určuje maximální velikost souboru auditu.

Parametry optických médií

Pokud používáte podsystém Optical Subsystem, použijte následující parametry. Další informace o těchto parametrech naleznete v příručce *IBM Informix Optical Subsystem Guide*.

Konfigurační parametr	Popis
OPCACHEMAX	Určuje velikost mezipaměti.
STAGEBLOB	Určuje název prostoru blobspace, do kterého mají být ukládány jednoduché velké objekty určené k zápisu na optický disk.

Program IBM Informix Storage Manager umožňuje zálohovat data na optická média, neumožňuje však databázovému serveru přímý přístup k datům uloženým na discích.

Parametry systému UNIX

Na některých platformách systému UNIX jsou dostupné další konfigurační parametry. Popis těchto specializovaných parametrů a pokyny k jejich použití naleznete v souboru Poznámky k počítači.

Monitorování informací o konfiguraci

Jedním z úkolů administrátora databázového serveru je průběžně zaznamenávat konfiguraci serveru. Tabulka 2-1 popisuje způsoby získávání informací o konfiguraci.

Tabulka 2-1. Monitorování informací o konfiguraci

Příkaz	Popis
onstat -c	Zobrazí kopii souboru ONCONFIG. Další informace naleznete v části "Konfigurace databázového serveru" na stránce 1-11. Změny souboru ONCONFIG se projeví při vypnutí a restartování databázového serveru, také nazývaném <i>opětovná inicializace sdílené paměti</i> . Pokud změníte konfigurační parametr, ale nevypnete a nerestartujete databázový server, bude se platná konfigurace lišit od výpisu příkazu onstat -c . Hodnoty konfiguračních parametrů jsou uloženy v souboru určeném proměnnou prostředí ONCONFIG . Pokud není nastavena proměnná prostředí ONCONFIG , hodnoty konfiguračních parametrů jsou uloženy v souboru \$INFORMIXDIR/etc/onconfig v systému UNIX nebo v souboru %INFORMIXDIR%\etc\onconfig.std v systému Windows.
oncheck -pr	Vypíše rezervovanou stránku. Databázový server ukládá aktuální informace o konfiguraci také ve vyhrazené stránce PAGE_CONFIG. Pokud změníte konfigurační parametry v příkazovém řádku a spustíte příkaz oncheck -pr , aniž byste vypnuli a restartovali databázový server, nebudou se hodnoty zobrazené programem oncheck shodovat s aktuálními hodnotami ve vyhrazených stránkách. Obslužný program oncheck vypíše varovnou zprávu.
ON-Monitor (systém UNIX)	Výběrem příkazu Status > Configuration vytvoříte kopii aktuální konfigurace a uložíte ji do zadaného adresáře a souboru. Pokud zadáte pouze název souboru, uloží databázový server tento soubor do aktuálního pracovního adresáře. Změny konfiguračních parametrů se projeví po vypnutí a restartování databázového serveru.
ISA	Zobrazí nebo aktualizuje konfigurační parametry.

Obrázek 2-1 zobrazuje ukázkou výstupu příkazu **oncheck -pr**.

```
...  
Validating Informix database server reserved pages - PAGE_CONFIG  
ROOTNAME          rootdbs  
ROOTPATH          /home/dyn_srv/root_chunk  
ROOTOFFSET        0  
ROOTSIZE          8000  
MIRROR            0  
MIRRORPATH        0  
MIRROROFFSET      0  
PHYSDBS           rootdbs  
PHYSFILE          1000  
LOGFILES          5  
LOGSIZE           500  
MSGPATH           /home/dyn_srv/online.log  
CONSOLE           /dev/tty5  
...               ...
```

Obrázek 2-1. PAGE_CONFIG Reserved Page

Kapitola 3. Komunikace mezi klientem a serverem

Obsah kapitoly	3-2
Architektura klient-server	3-2
Síťový protokol	3-2
Rozhraní pro síťové programování	3-3
Doména sítě systému Windows	3-3
Připojení k databázovému serveru	3-4
Multiplexní připojení	3-4
Připojení podporovaná databázovým serverem	3-5
Místní připojení	3-6
Připojení prostřednictvím sdílené paměti (systém UNIX)	3-6
Proudové propojení procesů (systém UNIX)	3-7
Proudové propojení procesů (systém Linux)	3-8
Připojení prostřednictvím pojmenovaného propojení procesů (systém Windows)	3-8
Připojení prostřednictvím místní zpětné smyčky	3-8
Služby pro podporu komunikace	3-8
Propojovací soubory	3-9
Soubory konfigurace sítě	3-9
Propojovací soubory sítě TCP/IP	3-9
Více portů TCP/IP	3-11
Propojovací soubory IPX/SPX (systém UNIX)	3-11
Soubory zabezpečení sítě	3-12
Soubor hosts.equiv	3-12
Informace souboru netrc	3-12
Soubor sqlhosts a klíč registru SQLHOSTS	3-13
Soubor sqlhosts (systém UNIX)	3-13
Nástroje k aktualizaci informací SQLHOSTS	3-14
Klíč registru SQLHOSTS (systém Windows)	3-14
Informace sqlhosts	3-15
Standardní názvy služeb a čísla portů IANA v souboru sqlhosts.std	3-16
Informace o propojitelnosti	3-17
Název databázového serveru	3-17
Pole typu připojení	3-17
Pole názvu hostitele	3-19
Pole názvu služby	3-20
Pole options	3-21
Informace o skupinách	3-27
Skupina databázových serverů	3-27
Alternativy k připojení TCP/IP	3-27
Adresy IP při připojení prostřednictvím protokolu TCP/IP	3-27
Určení adresy připojení TCP/IP pomocí zástupných znaků	3-28
Čísla portů při připojení prostřednictvím protokolu TCP/IP	3-30
Podpora adres IPv6 v serveru Dynamic Server	3-30
Parametry souboru ONCONFIG týkající se propojitelnosti	3-31
Konfigurační parametr DBSERVERNAME	3-31
Konfigurační parametr DBSERVERALIASES	3-32
Konfigurační parametr NETTYPE	3-32
Proměnné prostředí týkající se síťových připojení	3-33
Komunikační prostředky DRDA (Distributed Relational Database Architecture)	3-33
Přehledné informace o protokolu DRDA a předpoklady	3-33
Konfigurace serveru Dynamic Server pro připojení ke klientům datových serverů IBM	3-34
Přidělení jednotkových procesů dotazování pro kombinaci rozhraní a protokolu s konfiguračním parametrem NETTYPE	3-35
Určení velikosti vyrovnávací paměti komunikace DRDA pomocí konfiguračního parametru DRDA_COMMBUFSIZE	3-35
Jednotkový proces DRDAEXEC a dotazy klientů	3-36
Jazyk SQL a podporované a nepodporované datové typy	3-36
Zobrazení některých informací o připojení DRDA	3-37

Příklady konfigurací klientu a serveru	3-37
Použití připojení prostřednictvím sdílené paměti (systém UNIX)	3-37
Připojení prostřednictvím místní zpětné smyčky	3-38
Použití síťového připojení	3-39
Položka souboru sqlhosts pro protokol IPX/SPX (systém UNIX)	3-40
Použití více typů připojení.	3-40
Přístup k více databázovým serverům	3-41
Použití programu IBM Informix MaxConnect	3-42

Obsah kapitoly

V této kapitole jsou vysvětleny koncepty a termíny potřebné k pochopení komunikace mezi klientem a serverem a její konfigurace. Kapitola obsahuje následující části:

- Popis architektury klient-server.
- Typy připojení k databázovému serveru.
- Služby komunikace.
- Propojovací soubory.
- Parametry propojitelnosti v souboru ONCONFIG.
- Proměnné prostředí týkající se propojitelnosti.
- Příklady konfigurací klient-server.

Architektura klient-server

Produkty IBM Informix jsou navrženy v souladu s modelem softwarového návrhu zvaným *architektura klient-server*. Model klient-server umožňuje umístit aplikaci nebo *klient* do jednoho počítače a databázový *server* do jiného počítače, ale oba programy mohou být umístěny i v jediném počítači. Klientské aplikace odesílají do databázového serveru požadavky na služby a data databázového serveru. Databázový server jim zpětně odesílá data a poskytuje služby požadované klienty.

K *připojení* klientu k serveru a k přenosu dat mezi klientem a serverem se používá *síťový protokol* a *rozhraní pro síťové programování*. V následujících částech jsou tyto termíny popsány podrobněji.

Síťový protokol

Síťový protokol je sada pravidel, která řídí způsob přenosu dat mezi aplikacemi, v tomto případě mezi klientem a databázovým serverem. Tato pravidla určují mimo jiné formát dat přenášených po síti. Příkladem síťového protokolu je protokol TCP/IP.

Pravidla protokolu implementuje *ovladač síťového protokolu*. Ovladač síťového protokolu obsahuje kód, který formátuje data odesílaná klientem do databázového serveru a data, která klient z databázového serveru přijme.

Klientům a databázovým serverům poskytuje přístup k síťovému ovladači *rozhraní pro síťové programování*. Rozhraní pro síťové programování se skládá ze systémových volání a knihovných rutin, které poskytují přístup k prostředkům síťové komunikace. Příkladem rozhraní pro síťové programování v systému UNIX je rozhraní TLI (Transport Layer Interface). Příkladem rozhraní pro síťové programování v systému Windows je knihovna WINSOCK (rozhraní pro programování soketů).

Důležitou vlastností síťového protokolu je jeho schopnost zajistit komunikaci mezi klientem a serverem i v případech, kdy se klient a server nacházejí na různých počítačích s různými architekturami a operačními systémy.

Databázový server můžete nakonfigurovat tak, aby podporoval více než jeden protokol. Tuto volbu však zvažte pouze tehdy, pokud některé klienty používají protokol TCP/IP a jiné klienty používají protokol IPX/SPX.

Protokoly podporované jednotlivými operačními systémy jsou uvedeny v části “Připojení k databázovému serveru” na stránce 3-4.

Pokud chcete určit, jaký protokol má databázový server používat, nastavte v systému UNIX v souboru **sqlhosts** hodnotu **nettype**. V systému Windows nastavte hodnotu pole **PROTOCOL** v klíči registru **SQLHOSTS**. Další informace naleznete v části “Soubor sqlhosts a klíč registru SQLHOSTS” na stránce 3-13.

Rozhraní pro síťové programování

Rozhraní pro síťové programování je rozhraní pro programování aplikací (API), které obsahuje sadu komunikačních rutin a systémových volání. Aplikace může voláním těchto rutin komunikovat s jinou aplikací spuštěnou v témže nebo v jiném počítači. V kontextu této diskuse jsou klient a databázový server aplikace, které volají rutiny rozhraní TLI nebo soketového rozhraní API. Klienty i databázové servery odesílají a přijímají data pomocí rozhraní pro síťové programování v souladu s komunikačním protokolem.

Aby mohla komunikace klientu a serveru probíhat úspěšně, je třeba, aby byla prostředí klientu a databázového serveru nakonfigurována tak, aby používala společný protokol. Některé síťové protokoly jsou však přístupné pomocí více než jednoho rozhraní pro síťové programování. Například protokol TCP/IP je přístupný pomocí rozhraní TLI i pomocí soketů, v závislosti na tom, jaká programovací rozhraní je na platformě daného operačního systému k dispozici. Proto může klient v jednom počítači používající protokol TCP/IP prostřednictvím rozhraní TLI komunikovat s databázovým serverem v jiném počítači používajícím rozhraní TCP/IP prostřednictvím soketů a naopak. Příklad naleznete v části “Použití síťového připojení” na stránce 3-39.

Doména sítě systému Windows

Síťová technologie systému Windows umožňuje vytvářet síťové *domény*. Doména je skupina propojených počítačů se systémem Windows, které sdílejí informace o uživatelských účtech a zásady zabezpečení. *Řadič domény* spravuje informace uživatelských účtech pro všechny členy domény.

Řadič domény usnadňuje administraci sítě. Řadič domény spravuje jediný seznam účtů platný pro všechny členy domény a administrátor sítě tak nemusí synchronizovat seznamy účtů ve všech počítačích v doméně. Jinými slovy, administrátor sítě při vytváření nebo změně uživatelského účtu aktualizuje pouze seznam účtů v řadiči domény a nemusí aktualizovat seznamy účtů ve všech počítačích domény.

Aby se uživatel počítače se systémem Windows mohl přihlásit k databázovému serveru v jiném počítači se systémem Windows, musí být členem téže domény nebo musí být členem *důvěryhodné domény*. Důvěryhodná doména je doména, která má s jinou doménou vytvořený *vztah důvěryhodnosti*. Pokud je mezi doménami vytvořen vztah důvěryhodnosti, nacházejí se uživatelské účty pouze v důvěryhodné doméně, ale uživatelé se mohou přihlašovat k důvěryhodné doméně.

Uživatel, který se pokouší přihlásit k počítači se systémem Windows, který je členem domény, může použít místní přihlášení a profil nebo doménové přihlášení a profil. Pokud je však uživatel veden jako důvěryhodný uživatel nebo počítač, ze kterého se přihlašuje, je veden jako důvěryhodný počítač, může být uživateli udělen přihlašovací přístup i bez profilu.

Důležité: Klientská aplikace se k databázovému serveru Informix může připojit pouze tehdy, pokud v doméně systému Windows spuštěného databázového serveru existuje účet s příslušným uživatelským ID. Toto pravidlo platí i pro důvěryhodné domény.

Další informace o doménách naleznete v příručkách k operačnímu systému Windows.

Důležité: Mechanismus důvěryhodných klientů serveru Informix nemá žádnou souvislost se vztahem důvěryhodnosti, který může být vytvořen mezi doménami systému Windows. To znamená, že i pokud se klient připojuje z důvěryhodné domény systému Windows, musí v doméně databázového serveru existovat účet tohoto uživatele. Další informace o způsobu ověřování klientů databázovým serverem naleznete v částech “Služby pro podporu komunikace” na stránce 3-8 a “Soubory zabezpečení sítě” na stránce 3-12.

Připojení k databázovému serveru

Připojení je logické přidružení dvou aplikací, v tomto kontextu je to přidružení klientské aplikace a databázového serveru. K připojení klientu k databázovému serveru musí dojít *před* zahájením přenosu dat. Připojení také musí být udržováno po celou dobu přenosu dat.

Rada: Interní komunikační vrstva systému Informix se nazývá Association Services Facility (ASF). Pokud se zobrazují chybové zprávy odkazující na vrstvu ASF, došlo k potížím s připojením.

Klientská aplikace se k databázovému serveru připojuje pomocí příkazu CONNECT nebo DATABASE jazyka SQL. Aby se například aplikace mohla připojit k serveru `muj_server`, může obsahovat příkaz CONNECT v následujícím tvaru:

```
CONNECT TO '@muj_server'
```

Další informace o příkazech CONNECT a DATABASE naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

Multiplexní připojení

Některé aplikace se mohou z pověření uživatele připojit k databázovému serveru prostřednictvím několika současných připojení. *Multiplexní připojení* obsluhuje několik připojení *k databázi* prostřednictvím jediného *síťového* připojení. Klientské aplikace mohou vytvořit několik připojení k databázovému serveru, aby mohly za uživatele přistupovat k více než jedné databázi. Pokud připojení nejsou multiplexní, vytvoří se pro každé databázové připojení samostatné síťové připojení k databázovému serveru. Každé další síťové připojení spotřebuje další část paměti počítače a času procesoru CPU, a to i ta připojení, která nejsou aktivní. Multiplexní připojení umožňují databázovému serveru vytvořit několik databázových připojení bez zvýšených nároků na zdroje počítače potřebné pro vytvoření dalších síťových připojení.

Konfigurace databázového serveru pro podporu multiplexních připojení:

1. Definujte alias pomocí konfiguračního parametru DBSERVERALIAS. Můžete například zadat:
`DBSERVERALIAS ids_mux`
2. Přidejte pro alias položku souboru SQLHOSTS s použitím hodnoty `sqlmux` jako hodnoty **nettype** této položky. Hodnota **nettype** je druhý sloupec souboru SQLHOSTS. Můžete například zadat:
`ids_mux onsqlmux`

Ostatní pole této položky, pole **hostname** a **servicename**, musejí být zadána, jsou však ignorována.

- Multiplexování vybraných typů připojení povolíte zadáním volby `m=1` v souboru **sqlhosts** nebo v registru, který klient používá k připojení k databázovému serveru. Například:

```
menlo ontlitcp valley jfk1 m=1
```
- Na platformě systému Windows je třeba nastavit také proměnnou prostředí **IFX_SESSION_MUX**.

Následující příklad znázorňuje nastavení propojitelnosti s následujícími položkami v konfiguračním souboru:

- DBSERVERNAME web_tli
 - DBSERVERALIASES web_mux
- ```
web_tli ontlitcp node5 svc5 m=1
web_mux onsqlmux - -
```

Není zapotřebí provádět žádné změny souboru **sqlhosts** ani registru databázového serveru. Klientský program nemusí provádět žádná zvláštní volání příkazů SQL, aby mohl používat multiplexní připojení. Multiplexní připojení bude povoleno automaticky, pokud je správně nakonfigurován soubor ONCONFIG a soubor **sqlhosts** nebo klíč registru SQLHOSTS. Další informace o souboru **sqlhosts** a registru naleznete v části “Soubor sqlhosts a klíč registru SQLHOSTS” na stránce 3-13.

Pro multiplexní připojení platí následující omezení:

- Nejsou podporována připojení klientů s vícenásobnými jednotkovými procesy.
- Nejsou podporována připojení prostřednictvím sdílené paměti.
- Připojení k podřízeným databázovým serverům (například při distribuovaných dotazech nebo při replikaci dat) nejsou multiplexní.
- Funkce **sqlbreak()** jazyka ESQL/C není podporována.
- Podporu multiplexních připojení můžete v databázovém serveru aktivovat pouze při spuštění serveru.

Pokud bude některá z těchto podmínek při vytváření připojení porušena, vytvoří databázový server standardní připojení. Databázový server neoznámí chybu jazyka SQL.

## Připojení podporovaná databázovým serverem

Databázový server podporuje následující typy připojení klientských aplikací k serveru.

| Typ připojení                 | Windows | UNIX | Místní | Síťové |
|-------------------------------|---------|------|--------|--------|
| Sokety                        | X       | X    | X      | X      |
| TLI (TCP/IP)                  |         | X    | X      | X      |
| TLI (IPX/SPX)                 |         | X    | X      | X      |
| Sdílená paměť                 |         | X    | X      |        |
| Proudové propojení procesů    |         | X    | X      |        |
| Pojmenované propojení procesů | X       |      | X      |        |

**Poznámka:** Při konfigurování propojitelnosti zvažte nastavení konfiguračních parametrů **IFX\_LISTEN\_TIMEOUT** a **MAX\_INCOMPLETE\_CONNECTION**. Tyto parametry umožňují snížit riziko agresivního útoku typu DOS (denial-of-service) tím, že znesnadňují zahlcení virtuálního procesoru typu listener, který zpracovává připojení. Další informace naleznete v příručce *IBM*

**Jen pro UNIX**

Na mnoha platformách systému UNIX podporuje databázový server několik rozhraní pro síťové programování. Soubor Poznámky k počítači uvádí kombinace rozhraní a protokolu podporované databázovým serverem v konkrétním operačním systému:

Machine Specific Notes:

=====

1. The following interface/protocol combinations(s) are supported for this platform:

Berkeley sockets using TCP/IP

**Konec Jen pro UNIX**

**Nastavení klientského připojení:**

1. Určete v souboru ONCONFIG konfigurační parametry propojitelnosti a připojení.
2. Nastavte příslušné položky v propojovacích souborech použité platformy.
3. Nastavte proměnné prostředí týkající se propojitelnosti v inicializačních skriptech systému UNIX nebo v místním a doménovém registru systému Windows.
4. Definiujte v souboru **sqlhosts** nebo registru skupinu dbserver databázového serveru.

Následující části podrobněji popisují typy připojení k databázovému serveru. Podrobné informace o implementaci dále popisovaných připojení naleznete v následujících částech:

- “Propojovací soubory” na stránce 3-9
- “Informace sqlhosts” na stránce 3-15
- “Parametry souboru ONCONFIG týkající se propojitelnosti” na stránce 3-31
- “Proměnné prostředí týkající se síťových připojení” na stránce 3-33

---

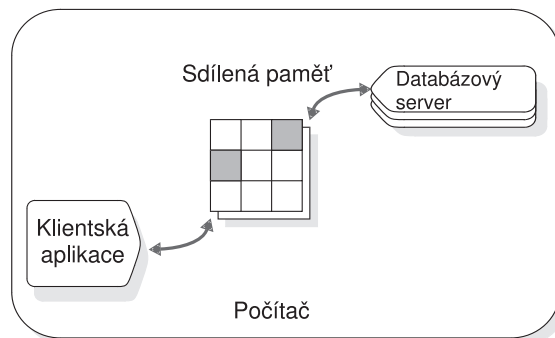
## Místní připojení

*Místní připojení* je připojení klienta k databázovému serveru v témže počítači. Následující části popisují typy těchto místních připojení.

### Připojení prostřednictvím sdílené paměti (systém UNIX)

*Připojení prostřednictvím sdílené paměti* používá oblast sdílené paměti jako *kanál*, pomocí kterého komunikuje klient s databázovým serverem. Obrázek Obrázek 3-1 znázorňuje připojení prostřednictvím sdílené paměti.





Obrázek 3-1. Připojení prostřednictvím sdílené paměti.

Sdílená paměť poskytuje rychlý přístup k databázovému serveru, ale představuje jisté bezpečnostní riziko. Chybná nebo škodlivá aplikace může zničit nebo číst vlastní vyrovnávací paměti zpráv nebo vyrovnávací paměti zpráv jiných místních uživatelů. Komunikace prostřednictvím sdílené paměti je také zranitelná chybami programu, pokud klientská aplikace explicitně adresuje paměť nebo překračuje meze indexů datových polí. Takové chyby neovlivňují provoz databázového serveru, pokud komunikuje prostřednictvím síťové komunikace nebo pomocí proudového propojení procesů. Příklad připojení prostřednictvím sdílené paměti naleznete v části “Použití připojení prostřednictvím sdílené paměti (systém UNIX)” na stránce 3-37.

Klient může s databázovým serverem navázat pouze jediné připojení prostřednictvím sdílené paměti.

Informace o části sdílené paměti používané databázovým serverem ke komunikaci s klienty naleznete v části “Komunikační část sdílené paměti (systém UNIX)” na stránce 7-20. Další informace naleznete také v části “Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)” na stránce 7-6.

## Proudové propojení procesů (systém UNIX)

*Proudové propojení procesů* je funkce meziprocessorové komunikace systému UNIX, která umožňuje procesům ve stejném počítači navzájem komunikovat. Připojení prostřednictvím proudového propojení procesů můžete použít, kdykoli se databázový server a klient nacházejí v téže počítači. Další informace naleznete v příručce “Pole síťového protokolu” na stránce 3-18 a v příručce “Komunikace prostřednictvím sdílené paměti a proudového propojení procesů (systém UNIX)” na stránce 3-20.

Připojení prostřednictvím proudového propojení procesů má následující výhody:

- Na rozdíl od připojení prostřednictvím sdílené paměti nepředstavuje proudové propojení procesů bezpečnostní riziko, protože nemůže dojít ke čtení nebo přepsání dat jinými programy, které by přistupovaly k téže části sdílené paměti.
- Na rozdíl od připojení prostřednictvím sdílené paměti umožňuje připojení prostřednictvím proudového propojení procesů provádět distribuované transakce mezi databázovými servery v téže počítači.

Připojení prostřednictvím proudového propojení procesů má následující nevýhody:

- Připojení prostřednictvím proudového propojení procesů může v být některých počítačích pomalejší než připojení prostřednictvím sdílené paměti.
- Proudové propojení procesů není k dispozici na všech platformách.
- Pokud ke komunikaci klientu se serverem používáte sdílenou paměť nebo proudové propojení procesů, bude položka **hostname** bude ignorována.

## Proudové propojení procesů (systém Linux)

Server Dynamic Server na platformách systému Linux podporuje meziprocesovou komunikaci (IPC) pomocí proudového propojení procesů a pomocí doménových soketů systému UNIX.

Aby se vytvářela proudová propojení procesů, musejí být parametr NETTYPE v souboru `$INFORMIXDIR/etc/SONCONFIG` a pole `nettype` v položce souboru `$INFORIMXDIR/etc/sqlhosts` nastaveny na hodnotu `onipcstr`.

Pokud je například zadána hodnota `onipcstr`, mohou se místní 32bitové aplikace připojit k 64bitovému serveru pomocí protokolů IPC proudového propojení procesů.

## Připojení prostřednictvím pojmenovaného propojení procesů (systém Windows)

*Pojmenovaná propojení procesů* jsou rozhraní pro programování aplikací (API) pro obousměrnou meziprocesovou komunikaci (IPC) v systému Windows. Připojení prostřednictvím pojmenovaného propojení procesů poskytují vysokoúrovňové rozhraní k síťovému softwaru díky transparentnímu provozu přenosové vrstvy. Pojmenovaná propojení procesů ukládají data v paměti a poskytují je na požádání způsobem podobným zápisu a čtení v souborovém systému.

Pojmenovaná propojení procesů jsou podporována při místním připojení k databázovému serveru.

## Připojení prostřednictvím místní zpětné smyčky

Síťové připojení klientské aplikace k databázovému serveru v témže počítači se nazývá připojení prostřednictvím *místní zpětné smyčky*. Toto připojení používá stejné prostředky síťové komunikace, jako kdyby byly klientská aplikace a databázový server spuštěny v jiných počítačích. Připojení prostřednictvím místní zpětné smyčky můžete provést tehdy, pokud je počítač vybaven ke zpracování síťových transakcí. Připojení prostřednictvím místní zpětné smyčky není tak rychlé jako připojení prostřednictvím sdílené paměti, ale na rozdíl od sdílené paměti nepředstavuje bezpečnostní riziko.

Pokud je klient připojen prostřednictvím místní zpětné smyčky, data jsou zdánlivě přenášena z klientské aplikace do sítě a pak zpět do databázového serveru. Přestože však databázový server používá rozhraní pro síťové programování (TLI nebo sokety), přenášejí interní komunikační procesy data mezi klientem a serverem přímo a *nevyužívají* vnější síť.

Příklad připojení prostřednictvím místní zpětné smyčky naleznete v části “Připojení prostřednictvím místní zpětné smyčky” na stránce 3-38.

---

## Služby pro podporu komunikace

Mezi *služby pro podporu komunikace* patří služby týkající se propojitelnosti, například následující služby zabezpečení:

- Ověřování je proces, který ověřuje identitu uživatele nebo aplikace. Nejobvyklejší formou ověřování je požadavek na zadání uživatelského jména nebo hesla, než bude uživateli poskytnut přístup k počítači nebo k aplikaci.
- Integrita zpráv zajišťuje, že všechny zprávy, které jsou předmětem komunikace, dorazí do cíle nezměněné a nepoškozené.
- Utajení zpráv představuje ochranu zpráv při přenosu před neoprávněnými uživateli, zpravidla pomocí šifrování a dešifrování.

Mezi služby pro podporu komunikace patří také další služby, například komprimace dat nebo účtování na základě objemu přenesených dat.

Databázový server poskytuje výchozí metodu ověřování, popsanou v části “Soubory zabezpečení sítě” na stránce 3-12. Databázový server použije výchozí metodu ověřování, pokud neurčíte modul pro podporu komunikace.

Databázový server poskytuje další služby pro podporu komunikace týkající se zabezpečení prostřednictvím modulů plug-in zvaných moduly pro podporu komunikace (CSM). Podrobnější informace naleznete v příručce *IBM Informix Security Guide*.

---

## Propojovací soubory

*Propojovací soubory* obsahují informace, které umožňují komunikaci klientu se serverem. Tyto soubory také umožňují databázovému serveru komunikovat s jiným databázovým serverem. Konfigurační soubory propojitelnosti lze rozdělit do tří skupin:

- soubory konfigurace sítě
- soubory zabezpečení sítě
- soubor **sqlhosts** nebo klíč registru SQLHOSTS

### Soubory konfigurace sítě

Tato část popisuje soubory konfigurace sítí TCP/IP a IPX/SPX a vysvětluje jejich použití.

#### Propojovací soubory sítě TCP/IP

Pokud konfigurujete databázový server k použití síťového protokolu TCP/IP, jsou k přípravě údajů pro soubor **sqlhosts** zapotřebí soubory konfigurace sítě **hosts** a **services**.

Tyto soubory spravuje administrátor sítě. Pokud přidáváte do sítě hostitele nebo síťovou službu, například databázový server, informujte administrátora sítě. Administrátor může zajistit, aby informace v těchto souborech odpovídaly.

V souboru **hosts** musí být zadána jedna položka pro každou kartu síťového rozhraní, která připojuje počítač s klientským nebo serverovým produktem Informix k síti. Každý řádek souboru obsahuje následující informace:

- adresu v síti Internet (nebo adresa IP karty sítě Ethernet),
- název hostitele
- aliasy hostitele (nepovinné)

Přestože délka názvu hostitele není v souboru **hosts** omezena, omezují produkty Informix délku názvu hostitele na 256 bajtů. Tabulka 3-4 na stránce 3-27 představuje příklad souboru **hosts**.

Soubor **services** obsahuje položku pro každou službu dostupnou pomocí protokolu TCP/IP. Každá položka představuje jeden řádek, který obsahuje následující informace:

- Název služby.

Produkty IBM Informix používají tento název k určení protokolu a čísla portu potřebného k propojení serveru s klientem. Název služby může být dlouhý až 128 bajtů.

- Číslo portu a protokol

Číslo portu představuje port počítače a hodnota protokolu pro protokol TCP/IP je **tcp**.

Operační systém klade omezení na čísla portů. Uživatel **informix** musí používat číslo portu větší než nebo rovné 1024. Pouze uživatelé **root** mohou používat číslo portu menší než 1024.

- Aliasy (nepovinné).

Název služby a číslo portu mohou být libovolné. Musejí však být jedinečné v kontextu tohoto souboru a identické ve všech počítačích, ve kterých jsou spuštěny produkty IBM Informix. Pole aliasů je nepovinné. Soubor **services** může například pro databázový server obsahovat následující položku:

```
server2 1526/tcp
```

Tato položka způsobí, že název server2 bude názvem služby portu TCP číslo 1526. Databázový server pak bude moci používat tento port k obsluhování požadavků na připojení. Obrázek 3-4 na stránce 3-20 obsahuje ukázkou souboru **services**.

**Důležité:** V případě databázových serverů, které komunikují s jinými databázovými servery je třeba definovat připojení TCP/IP pomocí parametru DBSERVERNAME nebo DBSERVERALIAS, a to i tehdy, pokud jsou obě instance spuštěny v jediném počítači.

Informace o souborech **hosts** a **services** naleznete v dokumentaci k použitému operačnímu systému.

**Propojovací soubory sítě TCP/IP v systému UNIX:** V systému UNIX se soubory **hosts** a **services** nacházejí v adresáři **/etc**. Tyto soubory musejí být přítomné v každém počítači, ve kterém jsou spuštěny klientské nebo serverové produkty IBM Informix, nebo v serveru NIS, pokud síť používá službu *Network Information Service* (NIS).

**Upozornění:** V systémech používajících službu NIS se soubory **/etc/hosts** a **/etc/services** nacházejí v serveru NIS. Soubory **/etc/hosts** a **/etc/services** v místním počítači nemusejí být používány a také nemusejí být aktuální. Obsah souborů NIS můžete zobrazit zadáním následujících příkazů v příkazovém řádku:

```
ypcat hosts
ypcat services
```

**Propojovací soubory sítě TCP/IP v systému Windows:** Údaje ze souborů **hosts** a **services** se používají k přípravě klíče registru SQLHOSTS pro síťový protokol TCP/IP. Tyto soubory se nacházejí v následujících umístěních:

- %WINDIR%\system32\drivers\etc\hosts
- %WINDIR%\system32\drivers\etc\services

Protokol TCP/IP můžete také nakonfigurovat tak, aby k překladu názvu hostitele používal službu Domain Name Service (DNS). Informace o těchto souborech naleznete v dokumentaci k použitému operačnímu systému.

Protokol Dynamic Host Configuration Protocol (DHCP) dynamicky přiřazuje adresy IP ze společné oblasti adres IP, nepoužívá tedy adresy explicitně přiřazené každé pracovní stanici. Pokud použitý systém používá protokol DHCP, je třeba, aby byla nainstalována služba Windows Internet Name Service (WINS). Protokol DHCP je pro databázový server transparentní.

**Co se děje mezi klientem a serverem při navazování připojení TCP/IP:** Když se navazuje připojení TCP/IP, načítají se na straně klientu následující informace:

- INFORMIXSERVER
- Informace o souboru **hosts** (INFORMIXSQLHOSTS, \$INFORMIXDIR/etc/sqlhosts, položka registru systému Windows NT) a souboru **services**
- další proměnné prostředí
- soubory zdrojů

Na straně serveru se načítají následující informace:

- DBSERVERNAME
- DBSERVERALIAS
- proměnné prostředí a konfigurační parametry serveru včetně všech konfiguračních parametrů NETTYPE, které řídí připojení TCP/IP.

Informace o konfiguračním parametru NETTYPE naleznete v části “Konfigurační parametr NETTYPE” na stránce 3-32 a v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Více portů TCP/IP

Abyste mohli používat několik karet sítě Ethernet současně, proveďte následující akce:

- Vytvořte v souboru **services** položku pro každý port, který bude používán databázovým serverem, jako v následujícím příkladu:

```
soc1 21/tcp
soc2 22/tcp
```

V případě jediné adresy IP musí být každý port jedinečný. Několik samostatných karet sítě Ethernet může používat jedinečná nebo sdílená čísla portů. Pokud je databázový server připojen několika kartami sítě Ethernet, můžete pro tyto karty chtít používat stejné číslo portu. (V tomto případě se používá totožný název služby.)

- Pro každou kartu sítě Ethernet s vlastní adresou IP přidejte do souboru **hosts** jednu položku jako v následujícím příkladu:

```
192.147.104.19 svc8
192.147.104.20 svc81
```

- V konfiguračním souboru ONCONFIG zadejte pro jednu kartu sítě Ethernet parametr DBSERVERNAME a pro ostatní karty parametr DBSERVERALIAS. Zde je uvedena ukázka položek souboru ONCONFIG:

```
DBSERVERNAME chicago1
DBSERVERALIASES chicago2
```

- Vytvořte v souboru **sqlhosts** v systému UNIX nebo v klíči registru SQLHOSTS v systému Windows, jednu položku pro každou kartu sítě Ethernet. Jinými slovy, vytvořte jednu položku pro parametr DBSERVERNAME a další položku pro parametr DBSERVERALIAS.

```
chicago1 onsoctcp svc8 soc1
chicago2 onsoctcp svc81 soc2
```

Po vytvoření této konfigurace komunikuje aplikace prostřednictvím karty sítě Ethernet přiřazené názvu **dbservername**, který poskytuje proměnná prostředí **INFORMIXSERVER**.

## Propojovací soubory IPX/SPX (systém UNIX)

Pokud chcete databázový server nakonfigurovat tak, aby používal protokol IPX/SPQ v síti systému UNIX, zakupte software protokolu IPX/SPX a nainstalujte ho do počítače databázového serveru. Výběr softwaru protokolu IPX/SPX závisí na používaném operačním systému. Pro některé operační systémy je software protokolu IPX/SPX dodáván společně s produkty založenými na softwaru NetWare for UNIX nebo Portable NetWare. Různí dodavatelé systému UNIX distribují software protokolu IPX/SPX, mohou také používat různé sady konfiguračních souborů.

Nápovědu, jak nastavit konfigurační soubory pro tyto softwarové produkty, naleznete v dokumentaci dodávanou s příslušným softwarem protokolu IPX/SPX.

## Soubory zabezpečení sítě

Produkty IBM Informix dodržují standardní postupy zabezpečení předepsané údaji v souborech zabezpečení sítě. Pokud se klientská aplikace připojuje k databázovému serveru ve vzdáleném počítači, musí mít uživatel klientské aplikace ve vzdáleném počítači platné ID uživatele.

### Soubor **hosts.equiv**

V souboru **hosts.equiv** jsou uvedeni vzdálení hostitelé a uživatelé, kterým počítač databázového serveru důvěřuje. Důvěryhodným uživatelům a uživatelům připojícím se ze důvěryhodných serverů je umožněn přístup k počítači bez zadání hesla. Operační systém používá soubor **hosts.equiv**, aby mohl určit, zda má být uživateli umožněn přístup bez zadání hesla. Produkty Informix vyžadují soubor **hosts.equiv** pro výchozí metodu ověřování.

Pokud klientská aplikace poskytne neplatný název účtu a heslo, databázový server odmítne připojení, a to i tehdy, pokud pro klientský počítač existuje položka v souboru **hosts.equiv**. Soubor **hosts.equiv** by měl být používán pouze pro ty klientské aplikace, které neodesílají serveru uživatelský účet a heslo. Soubor **hosts.equiv** se v systému UNIX nachází v adresáři **/etc**. V systému Windows se soubor **hosts.equiv** nachází v adresáři **%WINDIR%\system32\drivers\etc**. Pokud se v počítači soubor **hosts.equiv** nenachází, je třeba ho vytvořit.

V některých sítích se může název hostitele používaný vzdáleným hostitelem k připojení ke konkrétnímu počítači lišit od názvu hostitele, kterým tento počítač odkazuje na sebe. Síťový název hostitele může například obsahovat plně kvalifikovaný název domény (FQDN) jako v následujícím příkladu:

```
plně_kvalifikovaný_název_domény.informix.com
```

Počítač však na sebe může odkazovat pomocí místního názvu hostitele jako v následujícím příkladu:

```
viking
```

Pokud dojde k této situaci, zadejte do souboru **hosts.equiv** oba formáty názvu hostitele.

Abyste určili, zda je klient důvěryhodný, spusíte v klientském počítači následující příkaz:

```
rlogin název_hostitele
```

Pokud se vám podaří přihlásit a nezobrazí se výzva k zadání hesla, je klient důvěryhodný počítač.

Alternativně může jednotlivý uživatel uvést hostitele, ze kterých se může přihlásit jako důvěryhodný uživatel, v souboru **.rhosts**. Tento soubor se nachází v domovském adresáři uživatele v počítači, ve kterém je spuštěn databázový server.

#### Jen pro Windows

V systému Windows se domovský adresář uživatele nepřirazuje automaticky v okamžiku vytvoření identity uživatele administrátorem systému Windows. Administrátor může domovský adresář přidat do uživatelského profilu pomocí aplikace Správce uživatelů.

#### Konec Jen pro Windows

### Informace souboru **netrc**

Údaje v souboru **netrc** představují nepovinné informace, které určují data o identitě. Uživatel, který není oprávněný přistupovat k databázovému serveru nebo nepoužívá počítač, který je pro databázový server důvěryhodný, může pomocí tohoto souboru poskytnout

důvěryhodné uživatelské jméno a heslo. Tyto informace může poskytnout také uživatel, který ve vzdáleném počítači používá odlišný uživatelský účet a heslo.

V systému UNIX se informace souboru **netrc** nacházejí v souboru **.netrc** v domovském adresáři uživatele. K vytvoření souboru **.netrc** použijte jakýkoli standardní textový editor. Systém Windows spravuje informace souboru **netrc** v klíčích registru. K úpravě informací **netrc** použijte kartu Informace o hostiteli obslužného programu **setnet32**.

Pokud v aplikaci neposkytnete uživatelské heslo vzdáleného serveru explicitně (pomocí klauzule USER příkazu CONNECT nebo pomocí výzev k zadání uživatelského jména a hesla v programu DB–Access), vyhledá klientská aplikace uživatelské jméno a heslo v informacích **netrc**. Pokud uživatel zadal heslo explicitně v aplikaci nebo se připojuje k místnímu databázovému serveru, nebudou informace **netrc** použity.

Databázový server použije informace **netrc** bez ohledu na to, zda používá výchozí metodu ověřování nebo modul pro podporu komunikace.

Informace o specifickém obsahu tohoto souboru naleznete v dokumentaci k operačnímu systému.

#### Jen pro Windows

V systému Windows se domovský adresář uživatele nepřirazuje automaticky v okamžiku vytvoření identity uživatele administrátorem systému Windows. Administrátor může domovský adresář přidat do uživatelského profilu pomocí aplikace Správce uživatelů.

#### Konec Jen pro Windows

**Převzetí totožnosti uživatele:** V případě některých klientských dotazů nebo operací musí databázový server převzít totožnost klientu, aby mohl spustit proces nebo program z pověření klientu. Aby mohl převzít totožnost klientu, musí databázový server při každém klientském připojení přijmout heslo. Klienty mohou poskytnout ID uživatele a heslo prostřednictvím příkazu CONNECT nebo informací **netrc**.

Následující příklady představují způsob poskytnutí hesla potřebného k převzetí totožnosti klientu.

| Soubor nebo příkaz     | Příklad                                           |
|------------------------|---------------------------------------------------|
| informace <b>netrc</b> | machine trngpc3 login bruce password im4golf      |
| příkaz CONNECT         | CONNECT TO o_l_trngpc3 USER bruce USING "im4golf" |

## Soubor **sqlhosts** a klíč registru **SQLHOSTS**

Informace o propojitelnosti klientů a serverů Informix (informace *sqlhosts*) obsahují údaje, které klientské aplikaci umožňují nalézt v síti libovolný databázový server Informix a připojit se k němu.

Podrobný popis informací **sqlhosts** naleznete v části "Informace **sqlhosts**" na stránce 3-15.

### Soubor **sqlhosts** (systém UNIX)

V systému UNIX se soubor **sqlhosts** ve výchozím nastavení nachází v adresáři **\$INFORMIXDIR/etc**. Alternativně můžete nastavit proměnnou prostředí **INFORMIXSQLHOSTS** tak, aby obsahovala úplný název cesty a název souboru, který obsahuje informace souboru **sqlhosts**. Soubor **sqlhosts** musí být přítomný v každém počítači, který je hostitelem databázového serveru nebo klientu.

Každá položka (každý řádek) souboru **sqlhosts** obsahuje informace **sqlhosts** jednoho databázového serveru. Pole souboru oddělujte pomocí *prázdných znaků* (mezer, tabulátorů nebo obou těchto znaků). *Uvnitř* polí žádné mezery ani tabulátory nepište. Pokud chcete do souboru **sqlhosts** vložit poznámku, zahajte řádek znakem poznámky (#). Pro zlepšení čitelnosti můžete také vynechávat celé prázdné řádky. Další pravidla syntaxe všech polí jsou popsána v následujících částech, které popisují položky souboru **sqlhosts**. K zadání informací do souboru **sqlhosts** použijte libovolný standardní textový editor.

Tabulka 3-1 představuje ukázkou souboru **sqlhosts**.

Tabulka 3-1. Ukázka souboru **sqlhosts**

| dbservername | nettype  | hostname | servicename | options    |
|--------------|----------|----------|-------------|------------|
| menlo        | onipcshh | valley   | menlo       |            |
| newyork      | ontlitcp | hill     | dynsrvr2    | s=2,b=5120 |
| sales        | ontlisp  | knight   | sales       | k=0,r=0    |
| payroll      | onsoctcp | dewar    | py1         |            |
| asia         | group    | —        | —           | e=asia.3   |
| asia.1       | ontlitcp | node6    | svc8        | g=asia     |
| asia.2       | onsoctcp | node0    | svc1        | g=asia     |

## Nástroje k aktualizaci informací SQLHOSTS

Ke správě informací SQLHOSTS použijte jeden z následujících nástrojů:

- textový editor
- IBM Informix Server Administrator (ISA)
- obslužný program **setnet32**

**Rada:** Ke správě informací o propojitelnosti SQLHOSTS použijte program ISA. Přestože obslužný program **setnet32** umožňuje nastavit databázové servery (pole nettype, hostname, servicename a options), neumožňuje nastavit skupiny databázových serverů.

## Klíč registru SQLHOSTS (systém Windows)

Pokud instalujete databázový server, vytvoří **instalační program** v registru systému Windows následující klíč:

HKEY\_LOCAL\_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS

V této větvi podstromu HKEY\_LOCAL\_MACHINE jsou uloženy informace **sqlhosts**. Každý klíč větve SQLHOSTS představuje název databázového serveru. Pokud klepnete na název databázového serveru, zobrazí se v registru hodnoty polí HOST, OPTIONS, PROTOCOL a SERVICE konkrétního databázového serveru.

Informace o propojitelnosti musejí být přítomny v klíči registru sqlhosts nebo v ústředním registru v každém počítači, který je hostitelem databázového serveru nebo klientu. Pokud se klientská aplikace připojuje k databázovému serveru spuštěnému v témže počítači, sdílí oba programy jediný klíč registru sqlhosts.

**Umístění klíče registru SQLHOSTS:** Pokud instalujete databázový server, zobrazí instalační program dotaz, kam má být uložen klíč registru SQLHOSTS. Můžete určit jednu ze dvou následujících voleb:

- Místní počítač, do kterého instalujete databázový server.
- Jiný počítač v síti, který slouží jako sdílené centrální úložiště informací **sqlhosts** pro několik databázových serverů v síti.

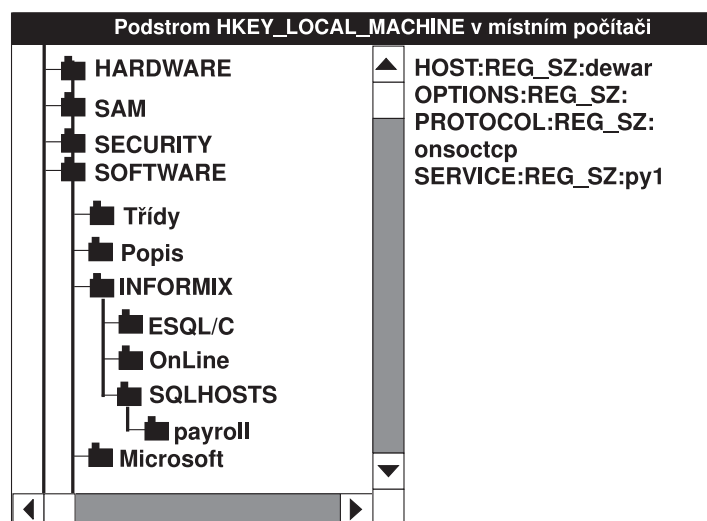


Pokud použijete sdílený klíč registru SQLHOSTS, nebudete muset udržovat totožné informace **sqlhosts** ve více počítačích. Soubory **hosts** a **services** v *každém* počítači však musejí obsahovat informace o všech počítačích, ve kterých jsou spuštěny databázové servery.

Pokud určíte sdílení klíče registru sqlhosts, nastavte proměnnou prostředí **INFORMIXSQLHOSTS** v místním počítači na název počítače se systémem Windows, ve kterém se nachází tento registr. Databázový server bude nejprve v počítači INFORMIXSQLHOSTS hledat klíč registru sqlhosts. Pokud databázový server v počítači INFORMIXSQLHOSTS klíč registru sqlhosts nenalezne nebo pokud proměnná **INFORMIXSQLHOSTS** není nastavena, bude databázový server hledat klíč registru sqlhosts v místním počítači.

Aby místní počítač mohl získat přístup ke sdílenému klíči registru sqlhosts, musejí být dodrženy konvence přístupu k síti systému Windows a přístupová oprávnění souborů. Informace o konvencích přístupu k síti a oprávněních souborů naleznete v dokumentaci k systému Windows.

Obrázek Obrázek 3-2 znázorňuje umístění a obsah klíče registru SQLHOSTS pro databázový server **payroll**.



Obrázek 3-2. Informace sqlhosts v registru systému Windows

## Informace sqlhosts

Informace **sqlhosts** v souboru **sqlhosts** v systému UNIX nebo v klíči registru SQLHOSTS registry v systému Windows obsahují informace o propojitelnosti všech databázových serverů. Informace **sqlhosts** obsahují také definice skupin. Databázový server hledá informace o propojitelnosti tehdy, pokud se databázový server spouští, pokud se klientská aplikace připojuje k databázovému serveru a pokud se databázový server připojuje k jinému databázovému serveru.

Informace o propojitelnosti každého databázového serveru se skládají ze čtyř polí požadovaných informací a jednoho nepovinného pole. Informace o skupinách obsahují pouze tři pole.

Pět polí informací o propojitelnosti tvoří v systému UNIX jeden řádek souboru **sqlhosts**. V systému Windows je název databázového serveru přiřazen podklíči klíče registru SQLHOSTS a ostatní pole tvoří hodnoty tohoto klíče. Následující tabulka shrnuje pole, která tvoří informace **sqlhosts**.

| UNIX<br>Název pole | Windows<br>Název pole                                                         | Popis<br>informací<br>o propojitelnosti      | Popis<br>informací<br>o skupině                                                 |
|--------------------|-------------------------------------------------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------|
| dbservername       | Klíč názvu databázového serveru <i>nebo</i> klíč skupiny databázového serveru | Název databázového serveru                   | Název skupinu databázových serverů                                              |
| nettype            | PROTOCOL                                                                      | Typ připojení                                | Slovo <i>group</i>                                                              |
| hostname           | HOST                                                                          | Hostitelský počítač databázového serveru     | <i>Žádné informace.</i><br>Jako zástupný symbol tohoto pole použijte spojovník. |
| servicename        | SERVICE                                                                       | Alias čísla portu                            | <i>Žádné informace.</i><br>Jako zástupný symbol tohoto pole použijte spojovník. |
| options            | OPTIONS                                                                       | Volby, které popisují nebo omezují připojení | Volby skupiny                                                                   |

#### Jen pro UNIX

Pokud instalujete produkt IBM Informix Enterprise Gateway with DRDA do téhož adresáře jako databázový server, obsahuje soubor **sqlhosts** také položky pro bránu Gateway a databázové servery jiné než Informix. V této příručce jsou však popsány pouze ty položky, které se týkají databázového serveru. Informace o dalších položkách souboru **sqlhosts** naleznete v příručce *IBM Informix Enterprise Gateway with DRDA User Manual*.

Konec Jen pro UNIX

## Standardní názvy služeb a čísla portů IANA v souboru **sqlhosts.std**

Organizace IANA (Internet Assigned Numbers Authority) přidělila databázovým serverům IBM Informix následující názvy služeb a čísla portů:

| Port/slужba | Kód IANA | Popis                                         |
|-------------|----------|-----------------------------------------------|
| sqlexec     | 9088/tcp | Rozhraní SQL serveru IBM Informix             |
| sqlexec-ssl | 9089/tcp | Rozhraní SQL serveru IBM Informix (šifrované) |

Tyto názvy služeb se nyní nacházejí v souboru **sqlhosts.std** serveru Dynamic Server. Nainstalované systémy serveru Informix Dynamic Server nemusíte měnit, protože budou i nadále pracovat s existujícími čísly portů a názvy služeb. (Také není zaručeno, že jiný systém již nepoužívá čísla portů a názvy služeb přidělené serveru Informix Dynamic Server.)

Organizace, které mají předpisy pro dodržování standardů, mohou tyto názvy služeb a čísla portů používat, pokud chtějí, aby databázový server dodržoval standard IANA. Pokud je na téže počítači nainstalována jiná aplikace, která již používá některý z těchto názvů služeb nebo čísel portů, můžete vydavatele této nekompatibilní aplikace požádat, aby požádal organizaci IANA o registraci čísel portů, aby se předešlo konfliktům. Pamatujte, že dokud provozujete nekompatibilní aplikaci, můžete server Dynamic Server spouštět na nestandardních portech.

Další informace naleznete na webu organizace IANA.

## Informace o propojitelnosti

V následující části jsou popsány informace o propojitelnosti, které se nacházejí v každém poli souboru `sqlhosts` nebo klíče registru `SQLHOSTS`.

### Název databázového serveru

Pole názvu databázového serveru (**dbservername**) obsahuje název databázového serveru, jehož informace o propojitelnosti jsou určovány. Název každého databázového serveru musí být ve všech přidružených sítích jedinečný. Pole **dbservername** musí odpovídat síťovému názvu databázového serveru určenému konfiguračními parametry `DBSERVERNAME` a `DBSERVERALIASES` v konfiguračním souboru `ONCONFIG`. Další informace o těchto konfiguračních parametrech naleznete v části “Parametry souboru `ONCONFIG` týkající se propojitelnosti” na stránce 3-31.

Pole **dbservername** smí obsahovat jakýkoli tisknutelný znak s výjimkou velkého písmene, oddělovače polí, znaku nového řádku a znaku poznámky. Délka pole je omezena na 128 bajtů.

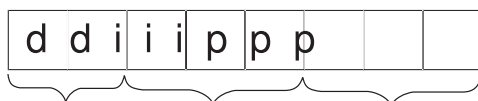
#### Jen pro UNIX

Pokud soubor `sqlhosts` obsahuje několik položek se shodným názvem `dbservername`, bude použita pouze první položka.

#### Konec Jen pro UNIX

### Pole typu připojení

Pole typu připojení (**nettype** v systému UNIX nebo `PROTOCOL` v systému Windows) popisuje typ připojení, které má být vytvářeno mezi databázovým serverem a klientskou aplikací nebo mezi dvěma databázovými servery. Toto pole je posloupností osmi písmen a skládá se ze tří podskupin, jak znázorňuje Obrázek 3-3.



Produkt databázového serveru    Typ rozhraní    Síťový protokol nebo mechanismus IPC

Obrázek 3-3. Formát pole typu připojení

V následujících částech jsou popsány jednotlivé části pole typu připojení.

**Serverový databázový produkt:** První dvě písmena pole typu připojení představují serverový databázový produkt.

#### Část pole určující produkt

##### Produkt

on *nebo* ol      Databázový server

dr (systém UNIX)

IBM Informix Enterprise Gateway with DRDA

Informace o příkazu `DRDA` naleznete v příručce *IBM Informix Enterprise Gateway with DRDA User Manual*.

Informace o používání protokolu DRDA ke komunikaci mezi serverem Dynamic Server a klienty datových serverů IBM naleznete v části “Komunikační prostředky DRDA (Distributed Relational Database Architecture)” na stránce 3-33.

**Typ rozhraní:** Průstřední tři písmena pole typu připojení představují rozhraní pro síťové programování, které umožňuje serveru komunikovat. Další informace naleznete v části “Rozhraní pro síťové programování” na stránce 3-3.

**Část pole určující rozhraní**

|     | <b>Typ rozhraní</b>                      |
|-----|------------------------------------------|
| ipc | IPC (meziprocesová komunikace)           |
| soc | Sokety                                   |
| tli | TLI (rozhraní Transport Layer Interface) |

Meziprocesová komunikace (IPC) se používá pouze ke komunikaci mezi dvěma procesy spuštěnými v téže počítači.

**Pole síťového protokolu:** Poslední tři písmena pole typu připojení představují síťový protokol nebo konkrétní mechanismus komunikace IPC.

**Část pole určující protokol**

|     | <b>Typ protokolu</b>                                                  |
|-----|-----------------------------------------------------------------------|
| imc | Síťový protokol TCP/IP, který používá program IBM Informix MaxConnect |
| nmp | Komunikace prostřednictvím pojmenovaného propojení procesů            |
| shm | Komunikace prostřednictvím sdílené paměti                             |
| spx | Síťový protokol IPX/SPX                                               |
| str | Komunikace prostřednictvím proudového propojení procesů               |
| tcp | Síťový protokol TCP/IP                                                |

Připojení prostřednictvím komunikace IPC používají sdílenou paměť nebo proudové propojení procesů. Databázový server podporuje dva síťové protokoly: TCP/IP a IPX/SPX.

Tabulka 3-2 uvádí přehled možných hodnot pole typu připojení k databázovému serveru. Další informace o protokolech programu MaxConnect naleznete v části “Použití programu IBM Informix MaxConnect” na stránce 3-42.

Tabulka 3-2. Přehled hodnot pole *nettype*

| Hodnota pole <i>nettype</i> (systém UNIX) | Hodnota klíče PROTOCOL (systém Windows) | Popis                                                                 | Typ připojení |
|-------------------------------------------|-----------------------------------------|-----------------------------------------------------------------------|---------------|
| onipcshm                                  |                                         | Komunikace prostřednictvím sdílené paměti                             | IPC           |
| onipcstr                                  |                                         | Komunikace prostřednictvím proudového propojení procesů               | IPC           |
|                                           | onipcnmp                                | Komunikace prostřednictvím pojmenovaného propojení procesů            | IPC           |
| ontlitcp                                  |                                         | Rozhraní TLI s protokolem TCP/IP                                      | Síťové        |
| onsoctcp                                  | onsoctcp                                | Sokety s protokolem TCP/IP                                            | Síťové        |
| ontlispix                                 |                                         | Rozhraní TLI s protokolem IPX/SPX                                     | Síťové        |
| onsocime                                  |                                         | Sokety s protokolem TCP/IP ke komunikaci s programem MaxConnect       | Síťové        |
| ontliimc                                  |                                         | Rozhraní TLI s protokolem TCP/IP ke komunikaci s programem MaxConnect | Síťové        |
| onsqlmux                                  | onsqlmux                                | Jediné síťové připojení s několika databázovými připojeními.          | Síťové        |

Pokud chcete povolit připojení SQLMUX, použijte jako položku **nettype** hodnotu **sqlmux**. Můžete například zadat:

```
ids_mux onsqlmux
```

Informace o typech připojení pro použitou platformu naleznete v části “Připojení podporovaná databázovým serverem” na stránce 3-5.

Pokud používáte protokol DRDA (Distributed Relational Database Architecture) k obsluze požadavků klientů datových serverů IBM, musíte v souboru SQLHOSTS registru systému Windows nakonfigurovat nový alias serveru, který jako protokol připojení používá protokol **drtlitcp** nebo protokol **drsoctcp**.

### Pole názvu hostitele

Pole názvu hostitele (pole **hostname** v systému UNIX nebo pole **host** v systému Windows) obsahuje název počítače, ve kterém je spuštěn databázový server. Pole názvu hostitele smí obsahovat jakýkoli tisknutelný znak s výjimkou velkého písmene, oddělovače polí, znaku nového řádku a znaku poznámky. Délka pole názvu hostitele je omezena na 256 bajtů.

Pole **hostname** musí být přítomné, pokud je však typ připojení nastaven na hodnotu **onsqlmux**, je toto pole ignorováno.

V následujících částech je popsán způsob, jakým klientské aplikace odvozují hodnoty použité v poli názvu hostitele.

**Síťová komunikace prostřednictvím protokolu TCP/IP:** Pokud používáte síťový protokol TCP/IP, bude pole názvu hostitele použito jako klíč souboru **hosts**, ze kterého bude získána síťová adresa počítače. Název použitý v poli názvu hostitele musí odpovídat *názvu* v souboru **hosts**. Ve většině případů se název hostitele v souboru **hosts** shoduje s názvem počítače. Další informace o souboru **hosts** naleznete v části “Propojovací soubory sítě TCP/IP” na stránce 3-9.

V některých případech můžete chtít použít v poli názvu hostitele skutečnou adresu IP sítě Internet. Informace o používání adres IP naleznete v části “Adresy IP při připojení prostřednictvím protokolu TCP/IP” na stránce 3-27.

**Komunikace prostřednictvím sdílené paměti a proudového propojení procesů (systém UNIX):** Pokud ke komunikaci klientu se serverem používáte sdílenou paměť nebo proudové propojení procesů, musí pole **hostname** obsahovat skutečný název hostitele, ve kterém je databázový server spuštěn.

**Síťová komunikace prostřednictvím protokolu IPX/SPX (systém UNIX):** Pokud ke komunikaci používáte síťový protokol IPX/SPX, musí pole **hostname** obsahovat název souborového serveru NetWare. Název souborového serveru NetWare je většinou *název hostitele* tohoto počítače v systému UNIX. Nemusí tomu ale tak být vždy. V případě potřeby se obraťte na administrátora serveru NetWare, který vám sdělí správné názvy souborových serverů NetWare.

**Rada:** Instalační a administrační obslužné programy systému NetWare mohou zobrazovat název souborového serveru NetWare velkými písmeny, například VALLEY. V souboru **sqlhosts** můžete zadat název malými i velkými písmeny.

### Pole názvu služby

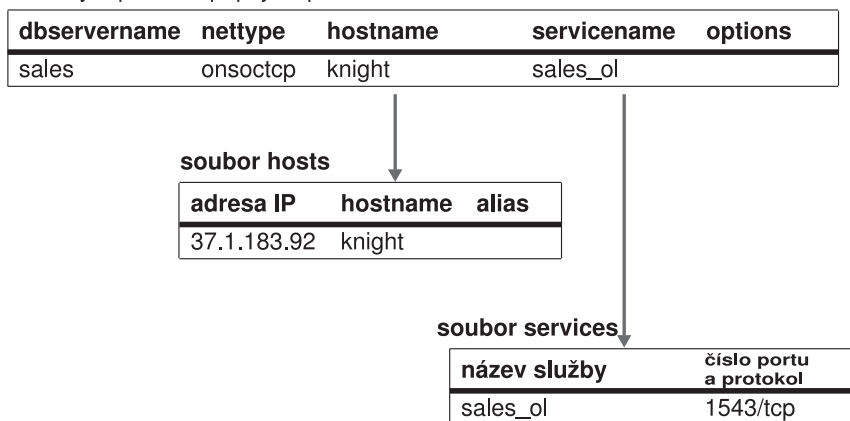
Interpretace hodnoty v poli názvu služby (**servicename** nebo **service**) závisí na typu připojení určeném v poli typu připojení (**nettype** nebo **PROTOCOL**). Pole názvu služby smí obsahovat jakýkoli tisknutelný znak s výjimkou velkého písmene, oddělovače polí, znaku nového řádku a znaku poznámky. Délka pole názvu služby je omezena na 128 bajtů.

Pole **servicename** musí být přítomné, pokud je však typ připojení nastaven na hodnotu **onsqlmux**, je toto pole ignorováno.

**Síťová komunikace prostřednictvím protokolu TCP/IP:** Pokud k připojení používáte protokol TCP/IP, musí pole názvu služby odpovídat položce názvu služby v souboru **services**. Číslo portu v souboru **services** sděluje síťovému softwaru, jak nalézt databázový server v určeném hostiteli. Na názvu služby nezáleží, je však třeba se shodnout na společném názvu s administrátorem sítě.

Obrázek 3-4 znázorňuje vztahy mezi souborem **sqlhosts** nebo příslušným klíčem registru a souborem **hosts** a také vztahy mezi souborem **sqlhosts** a souborem **services**.

Položky sqlhosts k připojení protokolem TCP/IP



Obrázek 3-4. Vztahy mezi souborem **sqlhosts** nebo příslušným klíčem registru a soubory **hosts** a **services**

V některých případech můžete jako pole názvu služby použít skutečné číslo portu TCP, na kterém server očekává připojení. Informace o používání čísla portu naleznete v části “Číslo portů při připojení prostřednictvím protokolu TCP/IP” na stránce 3-30.

**Komunikace prostřednictvím pojmenovaného propojení procesů (systém Windows):**

Pokud pole PROTOCOL určuje připojení prostřednictvím pojmenovaného propojení procesů (**onipcnp**), může položku SERVICE tvořit libovolná krátká skupina písmen, která je jedinečná v prostředí hostitelského počítače, ve kterém je spuštěn databázový server.

**Komunikace prostřednictvím sdílené paměti a proudového propojení procesů (systém UNIX):**

Pokud pole **nettype** určuje připojení prostřednictvím sdílené paměti (**onipcshm**) nebo připojení prostřednictvím proudového propojení procesů (**onipcstr**), použijte databázový server interně hodnotu položky **servicename** k vytvoření souboru, pomocí kterého uskuteční připojení. V případě připojení typu **onipcshm** i **onipcstr** může položku **servicename** tvořit libovolná krátká skupina písmen, která je jedinečná v prostředí hostitelského počítače, ve kterém je spuštěn databázový server.

V případě připojení prostřednictvím proudového propojení procesů je doporučeno jako název **servicename** použít název **dbservername**.

**Síťová komunikace prostřednictvím protokolu IPX/SPX (systém UNIX):**

*Službou se v síti IPX/SPX rozumí program, který je připraven vykonávat na požádání práci, například databázový server. Při použití připojení IPX/SPX může být hodnota pole **servicename** libovolný řetězec, musí však být mezi službami dostupnými v síti IPX/SPX jedinečná. Je vhodné použít jako hodnotu pole **servicename** pole **dbservername**.*

## Pole options

Pole **options** obsahuje položky pro následující vlastnosti.

| Název volby                  | Písmeno volby                       | Odkaz                              |
|------------------------------|-------------------------------------|------------------------------------|
| Velikost vyrovnávací paměti  | b                                   | Strana 3-22                        |
| Přesměrování připojení       | c                                   | Strana 3-22                        |
| Konec skupiny                | e                                   | Strana 3-23                        |
| Skupina                      | g                                   | Strana 3-23                        |
| Identifikátor                | i                                   | Strana 3-26                        |
| Multiplexní připojení        | m                                   | Strana 3-4                         |
| Volba keep-alive             | k                                   | Strana 3-26                        |
| Zabezpečení                  | s (databázový server)<br>r (klient) | Strana 3-26                        |
| Modul pro podporu komunikace | csm                                 | <i>IBM Informix Security Guide</i> |

Pokud změníte hodnoty v poli **options**, ovlivní tyto změny následující připojení klientské aplikace. Nemusíte ukončit a restartovat klientskou aplikaci, aby se změny projevíly. Databázový server však načítá vlastní informace o propojitelnosti *pouze* v průběhu inicializace nebo restartu databázového serveru. Pokud změníte volby databázového serveru, je zapotřebí restartovat databázový server, aby se změny projevíly.

**Pravidla syntaxe pole voleb:** Všechny položky pole **options** mají následující formát:

písmeno=hodnota

V poli **options** můžete kombinovat několik položek a můžete je zapsat v libovolném pořadí. Maximální délka pole **options** je 256 bajtů.

Jako oddělovač voleb můžete použít čárku nebo prázdný znak. Prázdné znaky nelze používat uvnitř jednotlivých voleb.

Databázový server vyhodnocuje pole **options** jako sérii sloupců. Čárka nebo prázdný znak v poli **options** představuje konec sloupce. Klientské aplikace a databázové servery ověřují každý sloupec, aby zjistily, zda jsou jednotlivé volby podporované. Pokud některá volba není podporovaná, uživatel nebude upozorněn. Volba bude pouze ignorována.

Následující příklady představují ukázkou platné i neplatné syntaxe.

| Syntaxe        | Platná syntaxe | Poznámka                                                                              |
|----------------|----------------|---------------------------------------------------------------------------------------|
| k=0,s=3,b=5120 | Ano            | Syntaxe je platná.                                                                    |
| s=3,k=0 b=5120 | Ano            | Syntaxe je ekvivalentní předchozímu příkladu. (Namísto čárky je použit prázdný znak.) |
| k=s=0          | Ne             | Položky nelze kombinovat.                                                             |

**Volba velikosti vyrovnávací paměti:** Pomocí volby velikosti vyrovnávací paměti ( $b=hodnota$ ) můžete zadat velikost komunikační vyrovnávací paměti v bajtech. Volba velikosti vyrovnávací paměti platí pouze pro připojení prostřednictvím síťového protokolu TCP/IP. Ostatní typy připojení ignorují nastavení velikosti vyrovnávací paměti. Tuto volbu můžete použít tehdy, pokud výchozí velikost není pro konkrétní aplikaci dostatečně účinná. Výchozí velikost vyrovnávací paměti při použití protokolu TCP/IP je 4096 bajtů.

Změna velikosti vyrovnávací paměti umožňuje účinněji využít systémové a síťové zdroje, avšak pokud bude velikost vyrovnávací paměti nastavena na příliš velkou hodnotu, bude připojení uživatele odmítnuto, protože nebude možné přidělit paměť. Pokud například v systému, který používá 1000 uživatelů nastavíte hodnotu  $b=64000$ , bude systém požadovat pro komunikační vyrovnávací paměti 64 MB paměti. Toto nastavení může způsobit vyčerpání paměťových zdrojů počítače.

V mnoha operačních systémech je maximální velikost vyrovnávací paměti protokolu TCP/IP omezena na 16 kB. Maximální povolenou velikost vyrovnávací paměti naleznete v dokumentaci k operačnímu systému, nebo se obraťte na služby technické podpory, které poskytuje dodavatel použité platformy.

Pokud se síť skládá z několika různých typů počítačů, buďte při nastavování velikosti komunikační vyrovnávací paměti zvláště opatrní.

**Rada:** Používejte výchozí hodnotu velikosti komunikační vyrovnávací paměti. Pokud se rozhodnete nastavit odlišnou hodnotu velikosti vyrovnávací paměti, nastavte shodnou velikost vyrovnávací paměti na straně klientu i databázového serveru.

**Volba přesměrování připojení:** V souboru **sqlhosts** klientu můžete definovat skupinu databázových serverů. Při připojování ke skupině serverů se klient bude pokoušet připojit k jednotlivým serverům s využitím zabezpečení pro případ poruchy. Volba *přesměrování připojení* ( $c=hodnota$ ) označuje pořadí, ve kterém bude připojující se software vybírat databázové servery, aliasy nebo souběžné servery v rámci skupiny.

Volbu přesměrování připojení použijte v následujících případech:

- Chcete rozložit zatížení několika instancí databázového serveru.
- Chcete pomocí replikace High-Availability Data Replication (HDR) přesměrovat připojení na záložní databázový server, pokud dojde k selhání primárního serveru.



Následující tabulka popisuje možná nastavení volby přesměrování připojení.

| Nastavení | Výsledek                                                                                                                                                                                                                                                    |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| c=0       | Klientská aplikace se ve výchozím nastavení připojí k první instanci databázového serveru uvedené v souboru <b>sqlhosts</b> ve skupině databázových serverů. Pokud se klientu nepodaří připojit k první instanci, pokusí se připojit ke druhé instanci atd. |
| c=1       | Klientská aplikace vybere náhodný počáteční bod, ze kterého se připojí ke členům skupiny databázových serverů uvedeným v seznamu.                                                                                                                           |

**Důležité:** Volba přesměrování připojení je platná pouze ve skupině databázových serverů. Další informace o použití volby *g* k určení skupin databázových serverů naleznete v části “Volba skupiny” na stránce 3-23.

**Volba modulu pro podporu komunikace:** Pomocí volby modulu pro podporu komunikace (CSM) popište modul CSM každého databázového serveru, který používá modul CSM. Pokud volbu CSM neurčíte, databázový server použije výchozí metodu ověřování. Stejně nastavení volby CSM můžete určit pro všechny databázové servery popsané v souboru **sqlhosts**, také můžete pro každou položku **sqlhosts** určit jinou volbu CSM nebo můžete volbu CSM u položek vynechat.

Následující příklad představuje formát volby CSM:

```
csm=(název_csm,volby_připojení_csm)
```

Hodnota položky *název\_csm* se musí shodovat s položkou *název\_csm* v souboru **concsm.cfg**. Parametr *volby\_připojení\_csm* potlačí výchozí volby *připojení\_csm* určené v souboru **concsm.cfg**.

V následujícím příkladu je určeno, že k připojení bude použit modul pro podporu komunikace ENCCSM:

```
csm=(ENCCSM)
```

Informace o modulech pro podporu komunikace naleznete v příručce *IBM Informix Security Guide*.

**Volba konce skupiny:** Pomocí volby konce skupiny (*e=název\_dbservername*) určete název posledního databázového serveru ve skupině. Pokud tuto volbu určíte v jiné položce než v položce skupiny databázových serverů, bude ignorována.

Pokud nebude pro skupinu určena volba konce skupiny, bude software předpokládat, že členy skupiny tvoří sousedící položky. Konec skupiny bude určen prvním výskytem položky, která nepatří ke skupině nebo koncem souboru. Příklad volby konce skupiny znázorňuje Tabulka 3-3 na stránce 3-25.

**Volba skupiny:** Pokud v registru nebo v souboru **sqlhosts** definujete skupiny databázových serverů, můžete několik souvisejících položek používat jako jedinou logickou entitu určenou k vytváření nebo ke změně připojení klientů k serveru.

**Vytvoření skupiny databázových serverů v souboru sqlhosts (systém UNIX):**

1. Určete v poli DBSERVERNAME název skupiny databázových serverů, do které patří položka **sqlhosts** (maximálně 18 bajtů).

Název skupiny databázových serverů se může shodovat s počáteční hodnotou parametru DBSERVERNAME databázového serveru.

2. Do pole typu připojení zadejte klíčové slovo **group**.

3. Pole názvu hostitele a služby nebudou použita. Označte pole jako nepoužitá pomocí pomlček (-), které slouží k označování prázdných polí. Pokud nepoužíváte volby, můžete označení prázdných polí vynechat.

Položka skupiny databázových serverů umožňuje používat následující volby:

- c = přesměrování připojení
- e = konec skupiny
- g = volba skupiny
- i = volba identifikátoru

### **Vytvoření skupiny databázových serverů v klíči registru SQLHOSTS (systém Windows):**

1. Po výběru klíče SQLHOSTS zvolte možnost **Úpravy > Přidat klíč**.
2. V dialogovém okně Přidat klíč zadejte do pole **Název klíče** název skupiny databázových serverů.

Tato hodnota musí odpovídat hodnotě Možnosti u názvu klíče databázového serveru.

Dialogové okno Třída ponechejte prázdné. Klepněte na tlačítko **OK**.

3. Vyberte nový klíč, který jste právě vytvořili (klíč s názvem skupiny databázových serverů).

4. Zvolte možnost **Úpravy > Přidat hodnotu**.

5. V dialogovém okně Přidat hodnotu zadejte do pole **Název hodnoty** jednu z hodnot informací **sqlhosts** (HOST, OPTIONS, PROTOCOL, SERVICE).

Neměňte dialogové okno Typ dat. Klepněte na tlačítko **OK**.

6. V dialogovém okně Editor řetězců zadejte hodnotu u zvoleného pole a klepněte na tlačítko **OK**.

Pro skupinu databázových serverů zadejte následující hodnoty:

|          |                                        |
|----------|----------------------------------------|
| HOST     | -                                      |
| OPTIONS  | <i>i=jedinečná celočíselná hodnota</i> |
| PROTOCOL | skupina                                |
| SERVICE  | -                                      |

Každá skupina databázových serverů musí mít přidruženu hodnotu identifikátoru (**i=**), která je mezi všemi databázovými servery ve vašem prostředí jedinečná. Pokud chcete označit, že k těmto polím nepřidružujete žádné specifické hodnoty, zadejte u polí HOST a SERVICE znaménko mínus (-).

7. Po výběru klíče skupiny databázových serverů, zvolte možnost **Úpravy > Přidat klíč**.
8. V dialogovém okně Přidat klíč zadejte do pole **Název klíče** název databázového serveru.

Tato hodnota musí odpovídat klíči databázového serveru, jehož hodnota OPTIONS byla nastavena u klíče skupiny databázových serverů, vybraného v kroku 7.

Vytvořte klíče pro primární a sekundární servery replikace HDR ve stejné skupině databázových serverů.

9. Opakujte kroky 1 až 8 pro každé pole informací **sqlhosts**.

### **Použití skupin databázových serverů k replikaci Enterprise Replication:**

K replikaci Enterprise Replication použijte volby **i** a **g**. Všechny databázové servery, které se účastní replikace, musejí být členy skupiny databázových serverů. Každý podnikový databázový server musí být označen jedinečným identifikátorem, kterým je skupina serverů. Přesvědčte se, zda je soubor **sqlhosts** správně nastaven ve všech počítačích, které jsou účastníky replikace.

Další informace naleznete v části o přípravě prostředí replikace v příručce *IBM Informix Dynamic Server Enterprise Replication Guide*.

### Použití skupin databázových serverů k replikaci High-Availability Replication:

K replikaci High-Availability Data Replication (HDR) použijte volby **c**, **e** a **g**. Replikace HDR vyžaduje dva identické systémy. Další informace naleznete v části “Směrování klientů pomocí informací o propojitelnosti” na stránce 19-25.

**Důležité:** Skupiny databázových serverů nelze vnořovat do jiných skupin databázových serverů a člen skupiny databázových serverů smí být členem pouze jediné skupiny.

Tabulka 3-3 představuje příklad se dvěma skupinami - skupinou **asia** a skupinou **peru**. Skupina **asia** obsahuje následující členy:

- **asia.1**,
- **asia.2**,
- **asia.3**.

Protože skupina **asia** používá volbu konce skupiny (**e=asia.4**), hledá databázový server členy skupiny, dokud nenalezne položku **e=asia.4**, takže do skupiny bude zahrnut i server **usa.2**. Členy skupiny nemusejí být seřazeny ani nemusejí být uvedeny přímo ve skupině, do které patří. Proto je položka **asia.3** platná i přesto, že se nachází za definicí skupiny **peru**. Protože skupina **peru** nepoužívá volbu konce skupiny, bude do ní databázový server přidávat všechny členy, dokud nenalezne další definici skupiny nebo dokud nenalezne další definici člena skupiny s položkou **g=server** určující jinou skupinu. V tomto příkladu skupina **peru** group zahrnuje servery **peru.1**, **peru.2**, **peru.3** a **usa.1**.

Tabulka 3-3 představuje příklad skupin databázových serverů v souboru **sqlhosts**.

Tabulka 3-3. Skupiny databázových serverů v souboru **sqlhosts** nebo v registru

| dbservername | nettype  | hostname    | servicename | options  |
|--------------|----------|-------------|-------------|----------|
| asia         | group    | —           | —           | e=asia.4 |
| asia.1       | ontlitcp | node6       | svc8        | g=asia   |
| asia.2       | onsoctcp | node0       | svc1        | g=asia   |
| usa.2        | ontlisp  | node9       | sv2         |          |
| asia.4       | onsoctcp | node1       | svc9        | g=asia   |
| peru         | group    | —           | —           |          |
| peru.1       | ontlitcp | node4       | svc4        |          |
| peru.2       | ontlitcp | node5       | svc5        | g=peru   |
| peru.3       | ontlitcp | node7       | svc6        |          |
| usa.1        | onsoctcp | 37.1.183.92 | sales_ol    | k=1, s=0 |
| asia.3       | onsoctcp | node10      | svc10       | g=asia   |

Název skupiny databázových serverů můžete použít místo názvu databázového serveru v příkazu SQL CONNECT a v následujících proměnných prostředí:

- **INFORMIXSERVER.**

Hodnotou proměnné **INFORMIXSERVER** klientské aplikace může být název skupiny databázových serverů. Název skupiny databázových serverů však nelze použít jako hodnotu proměnné **INFORMIXSERVER** databázového serveru nebo obslužných programů databázového serveru.

- **DBPATH.**

Proměnná prostředí **DBPATH** může jako názvy databázových serverů obsahovat názvy skupin databázových serverů.

**Volba identifikátoru:** Volba identifikátoru (*i=číslo*) přiřazuje skupině databázových serverů číselný identifikátor. Identifikátor musí být kladné celé číslo a musí být v prostředí sítě jedinečný.

Další informace o použití volby identifikátoru naleznete v příručce *IBM Informix Dynamic Server Enterprise Replication Guide*.

**Volba keep-alive:** Volba keep-alive je volba sítě týkající se protokolu TCP/IP. Jiné typy připojení tato volba neovlivňuje. Pokud v poli **options** volbu keep-alive neuvedete, bude tato volba ve výchozím nastavení povolena. Volbu keep-alive nastavte na straně klientu, databázového serveru nebo na obou stranách určením parametru **k=1** v pátém sloupci souboru **sqlhosts**. Ve většině případů byste měli volbu keep-alive povolit.

Písmeno *k* identifikuje v poli **options** položky keep-alive následujícím způsobem:

k=0      Zakáže funkci keep-alive.  
k=1      Povolí funkci keep-alive.

Pokud si server a připojený klient nevyměňují data a volba keep-alive je povolena, kontroluje síťová služba v pravidelných intervalech spojení. Pokud přijímající strana neodpoví v čase určeném parametry operačního systému, zjistí síťová služba přerušené spojení a uvolní používané zdroje.

Pokud je volba keep-alive zakázána, nebude síťová služba pravidelně kontrolovat, zda je připojení stále aktivní. Pokud protistrana neočekávaně a bez upozornění přeruší připojení, například pokud počítač PC restartuje, nemusí síťová služba zjistit, že připojení bylo přerušeno.

**Volba multiplexního připojení:** Další informace o multiplexním připojení naleznete v části "Multiplexní připojení" na stránce 3-4.

**Volby zabezpečení:** Volby zabezpečení umožňují řídit vyhledávání v souborech zabezpečení operačního systému. Písmeno *s* označuje nastavení na straně databázového serveru, písmeno *r* označuje nastavení na straně klientu. V poli **options** lze určit obě tyto volby. Klient ignoruje nastavení *s* a databázový server ignoruje nastavení *r*.

V následující tabulce jsou uvedena možná nastavení volby zabezpečení.

| Nastavení | Výsledek                                                                                                                                                     |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| r=0       | Zakáže vyhledávání informací <b>netrc</b> na straně klientu (nebude možné poskytnout heslo).                                                                 |
| r=1       | Povolí vyhledávání informací <b>netrc</b> na straně klientu (výchozí nastavení na straně klientu).                                                           |
| s=0       | Zakáže na straně databázového serveru vyhledávání v souborech <b>hosts.equiv</b> a <b>rhosts</b> (budou povolena pouze příchozí připojení s použitím hesla). |
| s=1       | Povolí na straně databázového serveru pouze vyhledávání v souboru <b>hosts.equiv</b> .                                                                       |
| s=2       | Povolí na straně databázového serveru pouze vyhledávání v souboru <b>rhosts</b> .                                                                            |
| s=3       | Povolí na straně databázového serveru vyhledávání v souborech <b>hosts.equiv</b> i <b>rhosts</b> (výchozí nastavení na straně databázového serveru).         |

Volby zabezpečení umožňují řídit způsob, jakým může klient (uživatel) získat přístup k databázovému serveru. Ve výchozím nastavení použije databázový server Informix následující informace, aby určil, zda je hostitelský počítač klienta důvěryhodný:

- soubor **hosts.equiv**
- informace **rhosts**

Pomocí voleb zabezpečení můžete zakázat nebo povolit použití kteréhokoli z těchto souborů.

Pokud například chcete zamezit tomu, aby koncoví uživatelé mohli určit důvěryhodné hostitele v souboru **rhosts**, nastavte v poli **options** souboru **sqlhosts** nebo v klíči registru **SQLHOSTS** databázového serveru volbu **s=1**, která zakáže vyhledávání informací **rhosts**.

**Důležité:** Nezakazujte vyhledávání v souboru **hosts.equiv** v databázových serverech, které jsou používány k distribuovaným databázovým operacím. Nenastavujte tedy volby **s=0** a **s=2**, pokud předpokládáte, že bude server používán k distribuovanému zpracování dat.

## Informace o skupinách

Následující část popisuje pole souboru **sqlhosts** nebo registru týkající se skupin.

### Skupina databázových serverů

Skupiny databázových serverů umožňují zacházet s několika souvisejícími položkami databázových serverů jako s jedinou logickou entitou určenou k vytváření nebo ke změně připojení klientů k serveru. Pomocí skupin databázových serverů můžete také zjednodušit přesměrování připojení k databázovým serverům. Další informace o skupinách databázových serverů naleznete v části "Volba skupiny" na stránce 3-23.

## Alternativy k připojení TCP/IP

Následující části popisují některé způsoby, jak obejít vyhledávání portů a adres IP při připojení prostřednictvím protokolu TCP/IP.

### Adresy IP při připojení prostřednictvím protokolu TCP/IP

Pokud používáte připojení prostřednictvím protokolu TCP/IP (pomocí rozhraní TLI nebo soketů), můžete v poli **hostname** místo názvu hostitele nebo aliasu uvedených v souboru **hosts** použít skutečnou adresu IP hostitele. Adresa IP se skládá ze čtyř celých čísel v rozsahu 0 až 255, oddělených tečkami. Obrázek Tabulka 3-4 na stránce 3-27 znázorňuje ukázkové adresy IP a hostitele v souboru **hosts**.

Tabulka 3-4. Ukázkový soubor **hosts**

| Adresa IP    | Název hostitele | Alias hostitele |
|--------------|-----------------|-----------------|
| 555.12.12.12 | smoke           |                 |
| 98.765.43.21 | odyssey         |                 |
| 12.34.56.789 | knight          | sales           |

Při použití adresy IP počítače **knight** z tabulky Tabulka 3-4 jsou následující dvě položky **sqlhosts** ekvivalentní:

```
sales ontlitcp 12.34.56.789 sales_01
sales ontlitcp knight sales_01
```

Použití adresy IP může v některých případech urychlit připojení. Protože jsou však počítače obvykle známy prostřednictvím názvu hostitele, může použití adresy IP v poli názvu hostitele

ztížit identifikaci počítače, ke kterému je položka přidružena.

#### Jen pro UNIX

Adresu IP můžete nalézt v poli síťové adresy v souboru **hosts** nebo můžete použít příkazy **arp** a **ypmatch** systému UNIX.

#### Konec Jen pro UNIX

#### Jen pro Windows

System Windows můžete nakonfigurovat tak, aby k překladu doménových názvů sítě Internet (*mujpocitac.informix.com*) na adresy protokolu IP (149.8.73.14) používal jeden z následujících mechanismů:

- služba Windows Internet Name Service
- služba Domain Name Server

#### Konec Jen pro Windows

### Určení adresy připojení TCP/IP pomocí zástupných znaků

Pokud jsou splněny *obě* následující podmínky, můžete určit adresu v poli názvu hostitele pomocí zástupných znaků:

- Používáte připojení prostřednictvím protokolu TCP/IP.
- Počítač, ve kterém je spuštěn databázový server, je vybaven několika kartami síťového rozhraní (například třemi kartami sítě Ethernet).

Pokud jsou splněny uvedené podmínky, můžete v poli názvu hostitele používaného databázovým serverem použít znak hvězdičky (\*) jako *zástupný znak*. Pokud zadáte do pole názvu hostitele zástupný znak, bude databázový server přijímat připojení na všech platných adresách IP hostitelského počítače.

Ke každé adrese IP je přidružen jedinečný název hostitele. Pokud je počítač vybaven několika kartami síťového rozhraní (NIC), jak znázorňuje Obrázek 3-5 na stránce 3-29, musí v souboru **hosts** být položka pro každou kartu rozhraní. Soubor **hosts** počítače **texas** může například obsahovat následující položky.

| Karta NIC | Adresa IP    | Název hostitele |
|-----------|--------------|-----------------|
| Karta 1   | 123.45.67.81 | texas1          |
| Karta 2   | 123.45.67.82 | texas2          |

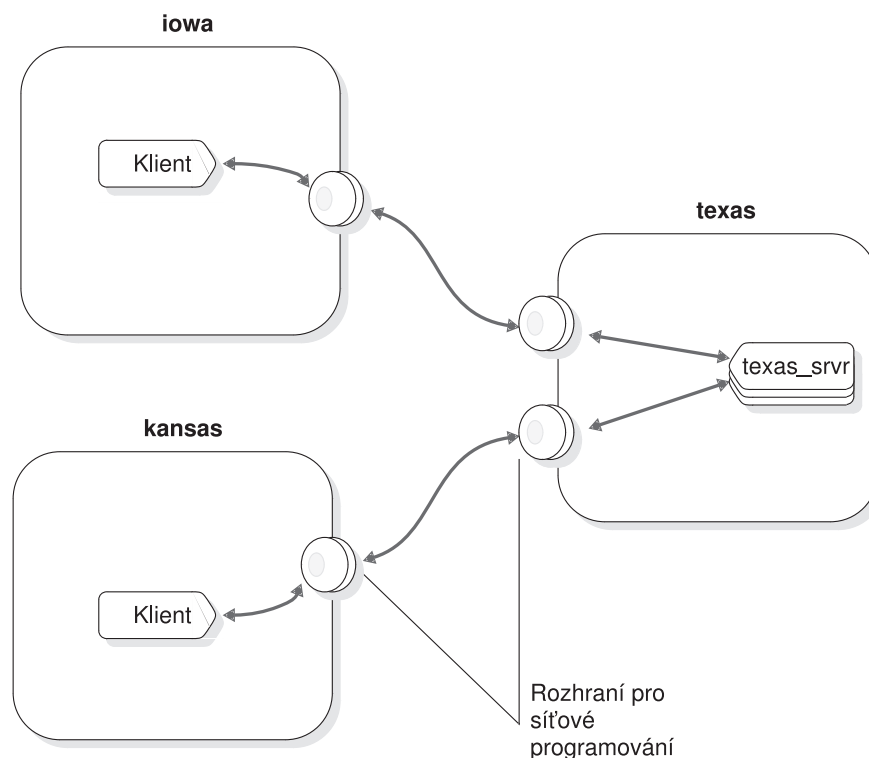
Zástupný znak (\*) můžete použít samostatně nebo jako předponu názvu hostitele nebo adresy IP, jak znázorňuje Tabulka 3-5 na stránce 3-29.

Pokud klientská aplikace a databázový server sdílejí informace o propojitelnosti (soubor **sqlhosts** nebo klíč registru SQLHOSTS), můžete v poli **názvu hostitele** určit zástupný znak i název hostitele nebo adresu IP (například \*texas1 nebo \*123.45.67.81). Klientská aplikace bude zástupný znak ignorovat a použije k připojení název hostitele (nebo adresu IP) a databázový server použije zástupný znak a bude přijímat připojení z jakékoli adresy IP.

Formát se zástupným znakem umožňuje jednotkovému procesu typu listener databázového serveru očekávat klientská připojení s použitím shodného čísla portu služby na všech platných kartách síťového rozhraní. Pokud však server očekává připojení na více adresách IP,

může vyžadovat větší množství času jednotky CPU, než kdyby očekával připojení ke konkrétnímu názvu hostitele nebo ke konkrétní adrese IP.

Obrázek Obrázek 3-5 znázorňuje databázový server v počítači (**texas**), který je vybaven dvěma kartami síťového rozhraní. Dva klientské počítače používají ke komunikaci s databázovým serverem dvě různé síťové karty.



Obrázek 3-5. Použití více karet síťového rozhraní

Obrázek Tabulka 3-5 znázorňuje informace o propojitelnosti databázového serveru **texas\_srvr**.

Tabulka 3-5. Možné položky propojitelnosti databázového serveru

| Název databázového serveru | Typ připojení | Název hostitele | Název služby |
|----------------------------|---------------|-----------------|--------------|
| texas_srvr                 | ontlitcp      | *texas1         | pd1_on       |
| texas_srvr                 | ontlitcp      | *123.45.67.81   | pd1_on       |
| texas_srvr                 | ontlitcp      | *texas2         | pd1_on       |
| texas_srvr                 | ontlitcp      | *123.45.67.82   | pd1_on       |
| texas_srvr                 | ontlitcp      | *               | pd1_on       |

**Důležité:** Můžete zadat pouze jednu z těchto položek.

Pokud informace o propojitelnosti odpovídá některému z uvedených řádků, bude databázový server **texas\_srvr** moci přijímat klientská připojení prostřednictvím obou síťových karet. Databázový server nalezne zástupný znak v poli **názvu hostitele** a bude ignorovat explicitně zadaný název hostitele.

**Rada:** V zájmu přehlednosti a snadné správy byste měli při použití zástupného znaku v poli názvu hostitele zadat i název hostitele (tedy použít formát \*hostitel místo pouhého znaku \*).

Informace o propojitelnosti používané klientskou aplikací musejí obsahovat explicitně zadaný název hostitele nebo adresu IP. Klientské aplikace spouštěné v počítači **iowa** mohou používat kterýkoli z následujících názvů hostitele: `texas1`, `*texas1`, `123.45.67.81` nebo `*123.45.67.81`. Pokud pole **názvu hostitele** obsahuje zástupný znak (\*), bude ho klientská aplikace ignorovat.

Klientské aplikace spouštěné v počítači **kansas** mohou používat kterýkoli z následujících názvů hostitele: `texas2`, `*texas2`, `123.45.67.82` nebo `*123.45.67.82`.

### Čísla portů při připojení prostřednictvím protokolu TCP/IP

Pokud používáte síťový protokol TCP/IP, můžete v poli názvu služby použít skutečné číslo portu TCP. Číslo portu TCP se nachází v poli **číslo portu** v souboru **services**.

Pokud je například v souboru **services** určeno číslo portu databázového serveru **sales** jako `1543/tcp`, můžete položku souboru **sqlhosts** zapsat následovně:

| servername | nettype  | hostname | servicename |
|------------|----------|----------|-------------|
| sales      | ontlitcp | knight   | 1543        |

Použití skutečného čísla portu může v některých případech zkrátit dobu potřebnou k připojení. Avšak stejně jako v případě použití adresy IP v poli **názvu hostitele** může být administrace informací o propojitelnosti složitější, pokud budete používat skutečné číslo portu.

---

## Podpora adres IPv6 v serveru Dynamic Server

Server Dynamic Server umí používat adresy IPv6 (Internet Protocol Version 6) v délce 128 bitů na mnoha platformách.

Server Dynamic Server podporuje na všech platformách adresy IPv4 (Internet Protocol Version 4) v délce 32 bitů.

Server IBM Informix Dynamic Server verze 10.00.UC4 a sada CSDK 2.90.UC4 nejprve kontrolují, zda je v použitém operačním systému protokol IPv6 podporovaný. Pokud je protokol IPv6 podporovaný, je použita adresa IPv6. Pokud použitý operační systém protokol IPv6 nepodporuje, je použita adresa IPv4. Server Dynamic Server a sada CSDK získávají adresu IP pomocí služby pro překlad názvů.

Server Dynamic Server spuštěný v hostiteli s nakonfigurovanou adresou IPv4 i IPv6 je ekvivalentní serveru spuštěnému v počítači s více kartami síťového rozhraní. Server Dynamic Server můžete v počítači s adresami IPv4 i IPv6 nakonfigurovat jedním z následujících způsobů:

- Vytvořte aliasy (pomocí konfiguračního parametru `DBSERVERALIASES`) a přiřaďte jednomu aliasu adresu IPv6 a druhému aliasu adresu IPv4.
- Pomocí zástupného znaku použitého jako název hostitele v souboru `SQLHOSTS` nakonfigurujte server Dynamic Server tak, aby očekával připojení na všech adresách IP nakonfigurovaných v hostiteli.

Například:

```
olserver1 oltlitcp *myhost olservice1
```



Povšimněte si hvězdičky před názvem hostitele. Toto schéma pojmenování hostitele se shoduje se schématem používaným pro servery spouštěné v počítači s více kartami síťového rozhraní. Počínaje serverem Dynamic Server verze 10.0 je název hostitele v souboru SQLHOSTS mapován na adresu IPv6, pokud má hostitel nakonfigurovanou adresu IPv6. Pokud hostitel nemá nakonfigurovanou adresu IPv6, bude položka názvu hostitele mapována na adresu IPv4. Název hostitele, kterému předchází hvězdička, bude mapován na adresu se zástupným znakem.

---

## Parametry souboru ONCONFIG týkající se propojitelnosti

Pokud restartujete databázový server, budou při restartu použity hodnoty parametrů konfiguračního souboru ONCONFIG. Následující parametry souboru ONCONFIG se týkají propojitelnosti:

- DBSERVERNAME
- DBSERVERALIASES
- NETTYPE

Následující části popisují tyto konfigurační parametry.

### Konfigurační parametr DBSERVERNAME

Konfigurační parametr DBSERVERNAME určuje název databázového serveru, nazývaný také název *dbservername*. Pokud například chcete jako název *dbservername* přiřadit hodnotu *nyc\_research*, použijte v konfiguračním souboru ONCONFIG následující řádek:

```
DBSERVERNAME nyc_research
```

Pokud se klientská aplikace připojuje k databázovému serveru, musí určit název *dbservername*. Informace **sqlhosts** přidružené k určenému názvu *dbservername* popisují typ připojení, které má být vytvořeno.

Klientské aplikace určují název databázového serveru jedním z následujících způsobů:

- Pomocí proměnné prostředí **INFORMIXSERVER**.
- V příkazech jazyka SQL, například v příkazech **CONNECT**, **DATABASE**, **CREATE TABLE** a **ALTER TABLE**, které umožňují určit prostředí databáze.
- Pomocí proměnné prostředí **DBPATH**.

---

#### Jen pro Windows

V systému Windows nelze parametr DBSERVERNAME změnit prostřednictvím konfiguračního souboru, protože informace o instanci databázového serveru ukládá v klíči DBSERVERNAME registr.

---

#### Konec Jen pro Windows

Parametr DBSERVERNAME musí určovat název *dbservername* nebo jeden z názvů *dbserveraliases*. Název musí začínat malým písmenem a může obsahovat další malá písmena, číslice a podtržítka. Název nesmí obsahovat velká písmena, oddělovač polí (mezeru nebo tabulátor) ani znak nového řádku. Jiné znaky ze základní sady ASCII nemusejí pracovat spolehlivě. Pomlčka nebo znak mínus například mohou působit problémy a dvojtečka nemusí pracovat spolehlivě. Znak **@** je vyhrazen k oddělování názvu databáze od názvu serveru (formát je *databáze@server*).

## Konfigurační parametr DBSERVERALIASES

Parametr DBSERVERALIASES umožňuje přiřadit témuž databázovému serveru další názvy dbservername. Maximální počet aliasů je 32. Obrázek 3-6 znázorňuje položky konfiguračního souboru ONCONFIG, které přiřazují jediné instanci databázového serveru tři názvy dbservername.

```
DBSERVERNAME sockets_srvr
DBSERVERALIASES ipx_srvr,shm_srvr
```

Obrázek 3-6. Příklad parametrů DBSERVERNAME a DBSERVERALIASES

Položky **sqlhosts** přidružené k názvům dbservername, jak je znázorňuje Obrázek 3-6, mohou obsahovat i položky, které znázorňuje Obrázek 3-7. Protože ke každému názvu dbservername je přidružena odpovídající položka v souboru **sqlhosts** nebo klíči registru SQLHOSTS, můžete k jednomu databázovému serveru přiřadit několik typů připojení.

```
shm_srvr onipcshm my_host my_shm
sockets_srvr onsoctcp my_host port1
ipx_srvr ontlisp nw_file_server ipx_srvr
```

Obrázek 3-7. Tři položky jediného databázového serveru v souboru sqlhosts ve formátu systému UNIX.

Klientská aplikace používající soubor **sqlhosts**, jak ho znázorňuje Obrázek 3-7, se může připojit k databázovému serveru prostřednictvím sdílené paměti pomocí následujícího příkazu:

```
CONNECT TO '@shm_srvr'
```

Klientská aplikace se může k *témuž* databázovému serveru připojit prostřednictvím soketu TCP/IP pomocí následujícího příkazu:

```
CONNECT TO '@sockets_srvr'
```

Parametr DBSERVERALIASES musí začínat malým písmenem a může obsahovat další malá písmena, číslice a podtržítka. Parametr DBSERVERALIASES nesmí obsahovat velká písmena, oddělovač polí (mezeru nebo tabulátor) ani znak nového řádku. Jiné znaky ze základní sady ASCII nemusejí pracovat spolehlivě. Pomlčka nebo znak mínus například mohou působit problémy a dvojtečka nemusí pracovat spolehlivě. Znak @ je vyhrazen k oddělování názvu databáze od názvu serveru (formát je databáze@server).

V uvedených příkladech se příkazem @shm\_srvr klient připojí k neidentifikované databázi v uvedeném serveru. Alternativně se můžete připojit k databázi dbase1@shm\_srvr.

## Konfigurační parametr NETTYPE

Konfigurační parametr NETTYPE umožňuje upravit počet a typ virtuálních procesorů, které databázový server používá ke komunikaci. V konfiguračním souboru může být pro každý typ síťového připojení (ipcshm, ipestr, ipcnp, soctcp, tlitcp a tlisp) zadána samostatná položka NETTYPE.

Doporučení: Přestože parametr NETTYPE není vyžadovaný, měli byste parametr NETTYPE nastavit, pokud používáte dva nebo více typů připojení. Pokud je databázový server již nějakou dobu spuštěn, můžete pomocí konfiguračního parametru NETTYPE vyladit databázový server k vyššímu výkonu.

Další informace o parametru NETTYPE naleznete v části "Síťové virtuální procesory" na stránce 5-22. Informace o konfiguračním parametru NETTYPE naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

---

## Proměnné prostředí týkající se síťových připojení

Proměnné **INFORMIXCONTIME** (doba trvání připojení) a **INFORMIXCONRETRY** (opakování připojení) jsou proměnné prostředí *klientu*, které ovlivňují chování klientu při připojování k databázovému serveru. Pomocí těchto proměnných prostředí můžete minimalizovat vznik chyb připojení v důsledku vysokého provozu v síti.

Pokud se klientská aplikace explicitně připojuje k segmentům sdílené paměti, je možné, že bude zapotřebí nastavit proměnnou **INFORMIXSHMBASE** (počátek sdílené paměti). Další informace naleznete v části “Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)” na stránce 7-6.

Proměnná prostředí **INFORMIXSERVER** umožňují určit název dbservername výchozího serveru, ke kterému se budou klientské aplikace připojovat.

Další informace o proměnných prostředí naleznete v příručce *IBM Informix Guide to SQL: Reference*.

---

## Komunikační prostředky DRDA (Distributed Relational Database Architecture)

Tato část obsahuje informace o konfiguraci serveru Dynamic Server tak, aby používal protokol DRDA (Distributed Relational Database Architecture), což je sada protokolů, které dovolují více databázovým systémům a aplikačním programům pracovat společně.

Tato část obsahuje následující témata týkající se protokolu DRDA:

- “Přehledné informace o protokolu DRDA a předpoklady”
- “Konfigurace serveru Dynamic Server pro připojení ke klientům datových serverů IBM” na stránce 3-34
- “Přidělení jednotkových procesů dotazování pro kombinaci rozhraní a protokolu s konfiguračním parametrem NETTYPE” na stránce 3-35
- “Určení velikosti vyrovnávací paměti komunikace DRDA pomocí konfiguračního parametru DRDA\_COMMBUFFSIZE” na stránce 3-35
- “Jednotkový proces DRDAEXEC a dotazy klientů” na stránce 3-36
- “Jazyk SQL a podporované a nepodporované datové typy” na stránce 3-36
- “Zobrazení některých informací o připojení DRDA” na stránce 3-37

### Přehledné informace o protokolu DRDA a předpoklady

Architektura DRDA (Distributed Relational Database Architecture) je sada protokolů, které umožňují komunikaci mezi aplikacemi a databázovými systémy na různých platformách a které umožňují distribuovat data mezi několika platformami. Je možné připojit libovolnou kombinaci produktů pro správu relačních databází, které využívají protokol DRDA, a vytvořit tak distribuovaný systém pro správu relačních databází. Protokol DRDA koordinuje komunikaci mezi systémy definováním položek, které musí být vyměněny, a způsobu, kterým mají být tyto položky vyměněny.

Server IBM Informix Dynamic Server můžete nakonfigurovat tak, aby odpovídal na požadavky obecného rozhraní API (například rozhraní ovladače IBM Data Server JDBC a poskytovatele IBM Data Server .NET Provider) při komunikaci se serverem Dynamic Server.

Pokud chcete používat obecné rozhraní API a komunikovat prostřednictvím protokolu DRDA, musíte nainstalovat klient datového serveru IBM IBM a příslušný ovladač.

Replikace ER (Enterprise Replication), replikace dat a obslužné programy serveru Dynamic Server, například program DB-Access, vyžadují připojení SQLI. Přestože replikace ER, HDR a obslužné programy připojení DRDA nepoužívají, mohou replikace ER a replikace dat a připojení DRDA pracovat vedle sebe.

**Důležité:** Zatímco ID uživatele a hesla používaná ke komunikaci mezi obecným API a serverem Dynamic Server je možné šifrovat, komunikaci DRDA mezi obecným rozhraním API a serverem Dynamic Server šifrovat nelze.

Další informace naleznete v následujících částech:

- “Konfigurace serveru Dynamic Server pro připojení ke klientům datových serverů IBM”
- “Přidělení jednotkových procesů dotazování pro kombinaci rozhraní a protokolu s konfiguračním parametrem NETTYPE” na stránce 3-35
- “Určení velikosti vyrovnávací paměti komunikace DRDA pomocí konfiguračního parametru DRDA\_COMMBUFFSIZE” na stránce 3-35
- “Jednotkový proces DRDAEXEC a dotazy klientů” na stránce 3-36
- “Jazyk SQL a podporované a nepodporované datové typy” na stránce 3-36
- “Zobrazení některých informací o připojení DRDA” na stránce 3-37

## Konfigurace serveru Dynamic Server pro připojení ke klientům datových serverů IBM

Toto téma popisuje, jak nakonfigurovat připojení serveru Dynamic Server ke klientům datových serverů IBM.

**Předpoklady:** Musí být nainstalován klient datového serveru IBM Data Server Client a příslušný ovladač. Další informace naleznete v části “Přehledné informace o protokolu DRDA a předpoklady” na stránce 3-33.

Konfigurace serveru Dynamic Server pro připojení ke klientům datových serverů IBM:

1. Nakonfigurujte v souboru SQLHOSTS nebo v registru systému Windows, který jako protokol připojení používá protokol **drtlitcp** nebo protokol **drsoctcp**.

**Důležité:** Každá položka souboru SQLHOSTS nebo klíč registru SQLHOSTS musejí obsahovat jedinečný název služby, který je ekvivalentem čísla portu protokolu TCP/IP. Klienty DRDA musejí používat jiné porty než klienty SQLI, protože jednotkový proces naslouchání protokolu DRDA nerozumí pokusům klientů SQLI o připojení a jednotkový proces naslouchání protokolu SQLI nerozumí pokusům klientů DRDA o připojení.

V souboru SQLHOSTS nebo v registru systému Windows můžete určit informace v následujícím formátu:

*název serveru drtlitcp název počítače název služby nebo číslo portu*

*název služby drsoctcp název počítače název služby nebo číslo portu*

Název serveru by měl být název nebo alias serveru IBM Informix Dynamic Server.

Můžete například zadat:

valley01\_dr drtlitcp paulw 9502

Další informace o těchto souborech naleznete v části “Soubor sqlhosts a klíč registru SQLHOSTS” na stránce 3-13.

2. Ujistěte se, že je v souboru ONCONFIG připojení DRDA uvedeno jako jeden z aliasů serveru. Tento alias by neměl být název DBSERVERNAME.
3. K připojení k serveru použijte protokol DRDA.

Pokud při přístupu k serveru prostřednictvím protokolu DRDA obdržíte chybu 23104, je možné, že se klientská aplikace pokouší vázat hodnotu, která má jiné kódování než národní prostředí databáze. Před spuštěním serveru nastavte proměnnou prostředí `GL_USEGLU` na hodnotu 1. To umožní serveru inicializovat odpovídající převaděče pro převody sad kódů.

Viz také:

- “Přidělení jednotkových procesů dotazování pro kombinaci rozhraní a protokolu s konfiguračním parametrem `NETTYPE`”
- “Určení velikosti vyrovnávací paměti komunikace DRDA pomocí konfiguračního parametru `DRDA_COMMBUFFSIZE`”
- “Jednotkový proces `DRDAEXEC` a dotazy klientů” na stránce 3-36
- “Jazyk SQL a podporované a nepodporované datové typy” na stránce 3-36
- “Zobrazení některých informací o připojení DRDA” na stránce 3-37

## Přidělení jednotkových procesů dotazování pro kombinaci rozhraní a protokolu s konfiguračním parametrem `NETTYPE`

Konfigurační parametr `NETTYPE` konfiguruje jednotkové procesy dotazování každého typu připojení podporovaného používanou instancí databázového serveru. Pomocí tohoto konfiguračního parametru můžete jediné kombinaci rozhraní a protokolu přidělit více než jeden jednotkový proces dotazování.

Konfigurační parametr `NETTYPE` nastavte následovně:

1. Jako protokol připojení zadejte hodnotu **SQLI**, **drtlitcp** nebo **drsoctcp**.
2. Přidejte informace o počtu jednotkových procesů dotazování, o počtu připojení a o třídě virtuálního procesoru.

Můžete například zadat:

```
NETTYPE drtlitcp,3,2,CPU
```

Položka `NETTYPE` dokáže pracovat s více aliasy databázového serveru s totožným typem protokolu. Pokud tedy používáte protokol DRDA, bude mít jednotkový proces naslouchání v síti (`NETTYPE drtlitcp` nebo `drsoctcp`) obvykle otevřené alespoň dva sokety a bude naslouchat připojením. Jeden soket je otevřený pro připojení SQLI a druhý pro připojení DRDA. Další sokety mohou být otevřené tehdy, pokud jste nakonfigurovali více samostatných aliasů serverů.

Další informace o konfiguračním parametru `NETTYPE` naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Určení velikosti vyrovnávací paměti komunikace DRDA pomocí konfiguračního parametru `DRDA_COMMBUFFSIZE`

Pomocí konfiguračního parametru `DRDA_COMMBUFFSIZE` určete velikost vyrovnávací paměti komunikace DRDA. Minimální velikost je 4 kB, maximální velikost je 2 MB a výchozí velikost je 32 kB.

Vyrovnávací paměť o velikosti 1 MB můžete určit následovně: 1M, 1m, 1024K, 1024k nebo 1024. Server Dynamic Server hodnoty menší než 4 kB automaticky změní na 32 kB.

Po vytvoření relace DRDA tato relace přidělí vyrovnávací paměť komunikace o velikosti aktuální vyrovnávací paměti.

Pomocí funkce `isgetdrdacommbuffersize()` můžete získat aktuální velikost parametru `DRDA_COMMBUFFSIZE`.

Pomocí příkazu `onmode -wm` nelze toto nastavení změnit, zatímco je databázový server spuštěný.

## Jednotkový proces DRDAEXEC a dotazy klientů

Pro každý klient DRDA vytváří server Dynamic Server ke zpracování a spuštění dotazů relaci a jednotkový proces DRDAEXEC, který je ekvivalentem jednotkového procesu SQLEXEC. Tento jednotkový proces také formátuje výsledky dotazů podle protokolu DRDA a odesílá výsledky zpět do klientského počítače.

Dotazy odesílané z klientu DRDA jsou spouštěny paralelně, pokud je nastaven parametr `PDQPRIORITY` a příslušný dotaz může být spouštěn paralelně. Dotazy spouštěné z jednotkových procesů DRDAEXEC mohou být také spouštěny paralelně.

## Jazyk SQL a podporované a nepodporované datové typy

Při použití protokolu DRDA je syntaxe serveru Dynamic Server podporována v obecném rozhraní API.

V obecném rozhraní API jsou podporovány následující datové typy:

- BYTE
- TEXT
- BLOB
- CLOB
- CHAR(32 k)
- VARCHAR(255)
- NCHAR(32 k)
- NVARCHAR(255)
- LVARCHAR(32 k)
- DECIMAL
- MONEY
- DATE
- DATETIME
- INTERVAL
- SMALLINT
- INT
- INT8
- SERIAL
- SERIAL8
- SMALLFLOAT
- FLOAT
- BOOLEAN

Při použití připojení DRDA zaokrouhluje server Dynamic Server dekadické hodnoty a hodnoty měna s 32bitovou přesností pro všechny operace vyhledávání dat s dekadickými nebo datovými typy měna.

Hodnoty datového typu DATETIME serveru Dynamic Server jsou mapovány na hodnoty datových typů DATE, TIME a TIMESTAMP.

Následující datové typy jsou podporovány pro hostitelské proměnné databázového serveru:

- SMALLINT
- INT
- CHAR
- VCHAR
- DATE

## Zobrazení některých informací o připojení DRDA

Pomocí následujících příkazů **onstat** a **onmode** můžete zobrazit informace, které obsahují název jednotkového procesu DRDA a indikátor rozlišující relace SQLI DRDA:

- **onstat -g ses**
- **onstat -g sql**
- **onstat -g ath**
- **onstat -g stk**
- **onstat -u**
- **onstat -x**
- **onstat -G**
- **onstat -g ddr**
- **onstat -g env**
- **onstat -g stm**
- **onstat -g ssc**
- **onmode -D**
- **onmode -Z**

Výstup příkazu `onstat`, například jako název jednotkového procesu může zobrazovat název "drdaexec".

---

## Příklady konfigurací klientu a serveru

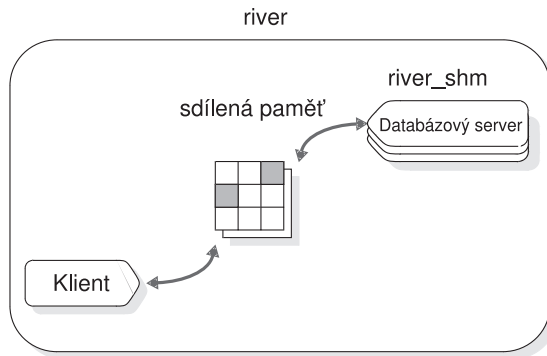
V následujících částech této kapitoly jsou uvedeny správné položky souboru `sqlhosts` a registru `SQLHOSTS` pro několik připojení klientů k serverům. Jednotlivé příklady představují:

- připojení prostřednictvím sdílené paměti,
- připojení prostřednictvím místní zpětné smyčky,
- připojení prostřednictvím sítě,
- použití více typů připojení,
- přístup k několika databázovým serverům.

**Důležité:** Následující příklady předpokládají, že soubory konfigurace sítě `hosts` a `services` byly správně připraveny, přestože tyto soubory nejsou v textu výslovně zmíněny.

## Použití připojení prostřednictvím sdílené paměti (systém UNIX)

Obrázek Obrázek 3-8 představuje připojení prostřednictvím sdílené paměti k počítači s názvem **river**.



Obrázek 3-8. Připojení prostřednictvím sdílené paměti

Konfigurační soubor ONCONFIG této instalace obsahuje následující řádek:

```
DBSERVERNAME river_shm
```

Následující tabulka představuje soubor **sqlhosts** nebo klíč registru SQLHOSTS pro připojení, které znázorňuje obrázek Obrázek 3-8.

Tabulka 3-6. Položka sqlhosts

| dbservername | nettype  | hostname | servicename |
|--------------|----------|----------|-------------|
| river_shm    | onipcshm | river    | rivershm    |

Klientská aplikace se k tomuto databázovému serveru připojuje prostřednictvím následujícího příkazu:

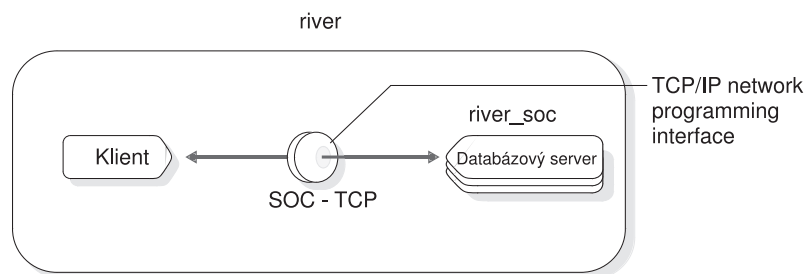
```
CONNECT TO '@river_shm'
```

Protože je použito připojení prostřednictvím sdílené paměti, nejsou zapotřebí žádné položky v souborech konfigurace sítě. V případě připojení prostřednictvím sdílené paměti můžete jako hodnoty polí **hostname** a **servicename** souboru **sqlhosts** nebo klíče registru SQLHOSTS použít libovolné hodnoty.

Další informace o připojení prostřednictvím sdílené paměti naleznete v části “Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)” na stránce 7-6.

## Připojení prostřednictvím místní zpětné smyčky

Obrázek Obrázek 3-9 znázorňuje připojení prostřednictvím místní zpětné smyčky, které používá sokety a protokol TCP/IP. Název hostitelského počítače je **river**.



Obrázek 3-9. Připojení prostřednictvím místní zpětné smyčky.

Následující tabulka znázorňuje soubor **sqlhosts** nebo klíč registru SQLHOSTS pro připojení, které zobrazuje obrázek Obrázek 3-9.



Tabulka 3-7. Položka `sqlhosts`

| <code>dbservername</code> | <code>nettype</code>  | <code>hostname</code> | <code>servicename</code> |
|---------------------------|-----------------------|-----------------------|--------------------------|
| <code>river_soc</code>    | <code>onsoctcp</code> | <code>river</code>    | <code>riverol</code>     |

Pokud síťové připojení bude místo soketů používat rozhraní TLI, změní se v tomto příkladu pouze položka **nettype**. V takovém případě bude položka **nettype** mít hodnotu `ontlitcp`, nikoli `onsoctcp`.

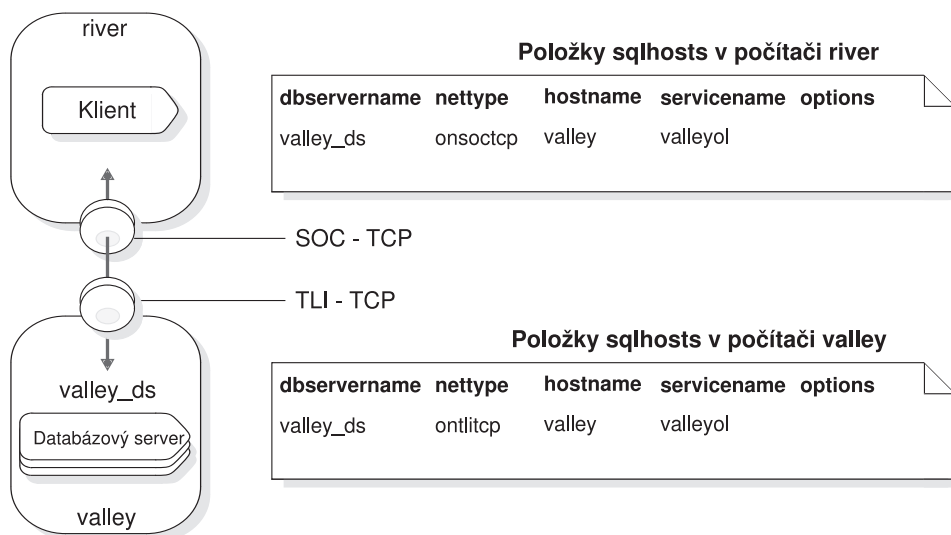
Soubor **ONCONFIG** obsahuje následující řádek:

```
DBSERVERNAME river_soc
```

Tento příklad předpokládá, že se v souboru **hosts** nachází položka **river** a v souboru **services** položka **riverol**.

## Použití síťového připojení

Obrázek Obrázek 3-10 znázorňuje konfiguraci, ve které se klientská aplikace nachází v hostiteli **river** a databázový server je spuštěn v hostiteli **valley**.



Obrázek 3-10. Konfigurace sítě

Položka databázového serveru **valley\_ds** se nachází v souborech **sqlhosts** nebo registrech obou počítačů. Ke každé položce souboru **sqlhosts** nebo klíče registru **SQLHOSTS** počítače, ve kterém je spuštěn databázový server, existuje odpovídající položka v počítači, ve kterém je spuštěna klientská aplikace.

### Jen pro UNIX

Oba počítače se nacházejí v téže síti TCP/IP, ale hostitel **river** používá jako rozhraní pro síťové programování sokety, zatímco hostitel **valley** používá jako rozhraní pro síťové programování rozhraní TLI. Pole **nettype** se řídí typem rozhraní pro síťové programování, které je používáno v počítači, ve kterém jsou uloženy informace **sqlhosts**. V tomto příkladu je hodnota pole **nettype** databázového serveru **valley\_ds** v hostiteli **river** rovna `onsoctcp` a hodnota pole **nettype** databázového serveru **valley\_ds** v hostiteli **valley** je rovna `ontlitcp`.

### Konec Jen pro UNIX

## Položka souboru sqlhosts pro protokol IPX/SPX (systém UNIX)

Software IPX/SPX často poskytuje rozhraní TLI. Obrázek Tabulka 3-8 znázorňuje položky souboru **sqlhosts** v obou počítačích platné pro případ, kdy konfigurace, kterou znázorňuje obrázek Obrázek 3-10 na stránce 3-39, používá rozhraní IPX/SPX namísto rozhraní TCP/IP.

Tabulka 3-8. Položka *sqlhosts*

| <b>dbservername</b> | <b>nettype</b> | <b>hostname</b> | <b>servicename</b> |
|---------------------|----------------|-----------------|--------------------|
| valley_us           | ontlisp        | valley_nw       | valley_us          |

V tomto případě obsahuje pole **hostname** název souborového serveru NetWare. Pole **servicename** obsahuje název, který je v síti IPX/SPX jedinečný a shoduje se s názvem dbservername.

## Použití více typů připojení

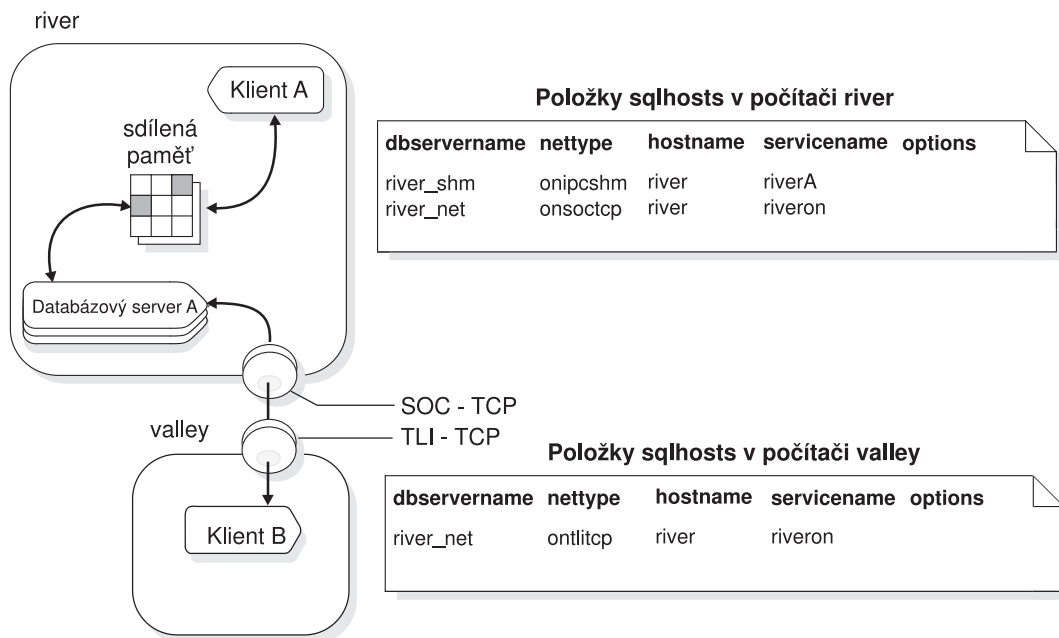
Instance databázového serveru může podporovat více typů připojení. Obrázek Obrázek 3-11 znázorňuje takovou konfiguraci. Databázový server se nachází v hostiteli **river**. Klient A se k databázovému serveru připojuje prostřednictvím sdílené paměti, protože sdílená paměť je rychlá. Klient B musí použít síťové připojení, protože klient a server se nacházejí v různých počítačích.

Pokud chcete, aby databázový server přijímal připojení několika typů, proveďte následující akce:

- Zadejte do konfiguračního souboru ONCONFIG položky DBSERVERNAME a DBSERVERALIASES.
- Do souboru **sqlhosts** nebo klíče registru SQLHOSTS zadejte jednu položku pro každou kombinaci databázového serveru a typu připojení.

V případě konfigurace, kterou znázorňuje obrázek Obrázek 3-11, má databázový server dva názvy dbservername: **river\_net** a **river\_shm**. Konfigurační soubor ONCONFIG obsahuje následující položky:

```
DBSERVERNAME river_net
DBSERVERALIASES river_shm
```



Obrázek 3-11. Konfigurace v systému UNIX používající více typů připojení

O typu připojení, které bude použito, rozhoduje název **dbservername** použitý klientskou aplikací. Obrázek Obrázek 3-11 znázorňuje situaci, ve které klient A použije k připojení k databázovému serveru následující příkaz:

```
CONNECT TO '@river_shm'
```

Parametr **nettype** přidružený v souboru **sqlhosts** nebo v klíči registru **SQLHOSTS** k názvu **river\_shm** určuje připojení prostřednictvím sdílené paměti, proto klient použije připojení prostřednictvím sdílené paměti.

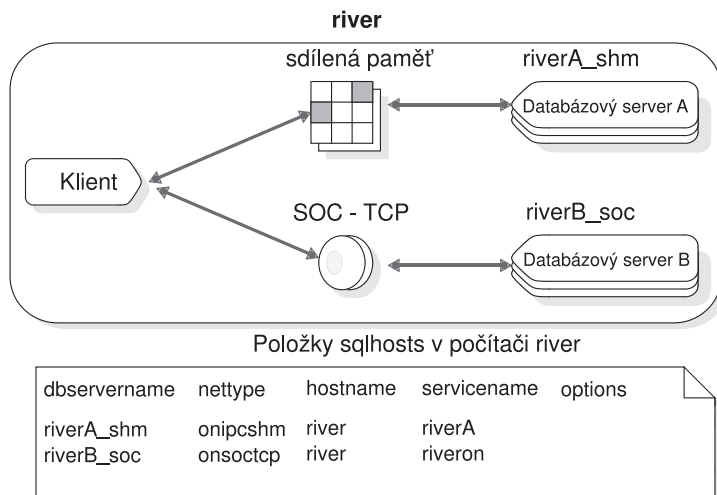
Klient B použije k připojení k databázovému serveru následující příkaz:

```
CONNECT TO '@river_net'
```

Hodnota parametru **nettype** přidružená v souboru **sqlhosts** nebo v registru k názvu **river\_net** určuje síťové připojení (prostřednictvím protokolu TCP/IP), proto klient B použije síťové připojení.

## Přístup k více databázovým serverům

Obrázek Obrázek 3-12 znázorňuje konfiguraci se dvěma databázovými servery spuštěnými v hostiteli **river**. Pokud je v jediném počítači aktivní více než jeden databázový server, jedná se o *vícenásobné uložení*. (Další informace o vícenásobném uložení naleznete v příručce *IBM Informix Installation Guide*.)



Obrázek 3-12. Několik databázových serverů v systému UNIX

Konfigurace, kterou znázorňuje obrázek Obrázek 3-12, vyžaduje dva konfigurační soubory ONCONFIG, jeden pro databázový server **A** a druhý pro databázový server **B**. Soubor **sqlhosts** a klíč registru SQLHOSTS obsahují informace o propojitelnosti obou databázových serverů.

Konfigurační soubor **ONCONFIG** databázového serveru **A** obsahuje následující řádek:  
 DBSERVERNAME                    riverA\_shm

Konfigurační soubor **ONCONFIG** databázového serveru **B** obsahuje následující řádek:  
 DBSERVERNAME riverB\_soc

## Použití programu IBM Informix MaxConnect

Program IBM Informix MaxConnect je síťový produkt pro prostředí databázových serverů Informix v systému UNIX. Program MaxConnect může spravovat velké množství (od několika set do desítek tisíc) spojení klient-server. Program MaxConnect multiplexuje připojení tak, že poměr počtu klientských připojení k počtu databázových připojení může být až 200:1 nebo větší. Program MaxConnect zvyšuje přizpůsobitelnost systému na mnoho tisíc připojení, šetří systémové zdroje, zkracuje dobu odezvy a snižuje požadavky na jednotku CPU. Program MaxConnect je nejvhodnější pro přenosy dat OLTP a není doporučeno ho používat ho k přenosům velkých objemů multimediálních dat.

Program MaxConnect nainstalujte nezávisle na serverových a klientských aplikacích Informix. Abyste dosáhli zvýšení výkonu, nainstalujte program MaxConnect do samostatného počítače, ke kterému se budou připojovat klienty systému Informix nebo do klientského aplikačního serveru. Program MaxConnect lze instalovat v následujících konfiguracích:

- Do vyhrazeného serveru, ke kterému se připojují klienty systému Informix.
- Do klientského aplikačního serveru.
- Do počítače databázového serveru.

Pro uživatele MaxConnect jsou pro multiplexní spojení dostupné protokoly **ontliime** a **onsocime**. Protokoly **ontliime** a **onsocime** lze používat v následujících dvou konfiguracích:

- K připojení programu MaxConnect k databázovému serveru.  
 V této konfiguraci jsou klientská připojení multiplexní a používají agregaci paketů.

- K přímému připojení klientských aplikací k databázovému serveru, které obchází program MaxConnect.

V této konfiguraci klient nevyužívá výhod multiplexního připojení ani agregace paketů. Tuto konfiguraci vyberte tehdy, pokud klientská aplikace přenáší data jednoduchých nebo inteligentních velkých objektů, protože v takovém případě je přímé připojení k databázovému serveru nejlepší alternativa.

Informace o konfiguraci programu MaxConnect a jeho monitorování pomocí příkazů **onstat -g imc** a **imcadmin** naleznete v příručce *IBM Informix MaxConnect User's Guide*.

**Důležité:** Program MaxConnect a příručka *IBM Informix MaxConnect User's Guide* se dodávají samostatně, nikoli s databázovým serverem Informix.



---

## Kapitola 4. Inicializace databázového serveru

|                                                                                       |      |
|---------------------------------------------------------------------------------------|------|
| Obsah kapitoly                                                                        | 4-1  |
| Typy inicializace                                                                     | 4-1  |
| Inicializace diskového prostoru                                                       | 4-2  |
| Jednotlivé kroky inicializace                                                         | 4-3  |
| Zpracování konfiguračního souboru                                                     | 4-3  |
| Vytvoření segmentů sdílené paměti                                                     | 4-4  |
| Inicializace a restartování sdílené paměti                                            | 4-5  |
| Inicializace diskového prostoru                                                       | 4-5  |
| Spuštění všech požadovaných virtuálních procesorů                                     | 4-5  |
| Provedení potřebných konverzí                                                         | 4-5  |
| Zahájení rychlé obnovy                                                                | 4-5  |
| Zahájení kontrolního bodu                                                             | 4-5  |
| Zaznamenání změn konfigurace                                                          | 4-6  |
| Vytvoření souboru oncfg_servername.servnum                                            | 4-6  |
| Vypuštění dočasných prostorů tbspace                                                  | 4-6  |
| Nastavení vynucené rezidence, je-li určena                                            | 4-6  |
| Návrat řízení uživateli                                                               | 4-6  |
| Vytvoření databáze sysmaster a příprava tabulek SMI                                   | 4-7  |
| Vytvoření databáze sysutils                                                           | 4-7  |
| Monitorování maximálního počtu uživatelských připojení                                | 4-7  |
| Provozní režimy databázového serveru                                                  | 4-7  |
| Změna provozních režimů databázového serveru                                          | 4-9  |
| Uživatelé oprávnění měnit režimy                                                      | 4-9  |
| Volby programu ISA pro změnu provozních režimů                                        | 4-10 |
| Volby programu ON-Monitor pro změnu režimů (UNIX)                                     | 4-10 |
| Volby příkazového řádku pro změnu režimů                                              | 4-11 |
| Změna z režimu offline do klidového režimu                                            | 4-11 |
| Změna z režimu offline do režimu online                                               | 4-11 |
| Změna z režimu offline do režimu administrace                                         | 4-12 |
| Změna z klidového režimu do režimu online                                             | 4-12 |
| Nenásilná změna z režimu online do klidového režimu                                   | 4-12 |
| Okamžitá změna z režimu online do klidového režimu                                    | 4-13 |
| Změna z klidového režimu nebo režimu online do režimu administrace                    | 4-13 |
| Změna z režimu administrace do režimu online                                          | 4-14 |
| Změna z režimu administrace do klidového režimu                                       | 4-14 |
| Okamžitá změna z jakéhokoli režimu do režimu offline                                  | 4-14 |
| Určení uživatelů režimu administrace pomocí konfiguračního parametru ADMIN_MODE_USERS | 4-15 |

---

### Obsah kapitoly

Tato kapitola pojednává o tom, jak inicializovat databázový server, popisuje jednotlivé kroky instalace a operační režimy databázového serveru a obsahuje informace o změně operačních režimů.

---

### Typy inicializace

Inicializace databázového serveru je souhrnným označením dvou navzájem souvisejících kroků: inicializace sdílené paměti a inicializace disku.

*Inicializace sdílené paměti* neboli spuštění serveru vytvoří následující obsah sdílené paměti databázového serveru: interní tabulky, vyrovnávací paměti a komunikační oblast sdílené paměti. Sdílená paměť je inicializována při každém spuštění databázového serveru.

K inicializaci sdílené paměti databázového serveru a uvedení databázového serveru do režimu online použijte obslužný program **oninit** z příkazového řádku.

K inicializaci sdílené paměti dochází také při restartování databázového serveru.

Inicializace sdílené paměti se liší od inicializace disku v jednom zásadním rysu:

- Inicializace sdílené paměti nemá vliv na přidělení nebo rozvržení diskového prostoru. Nedochozí při ní ke zničení dat.

*Inicializace diskového prostoru* použije hodnoty uložené v konfiguračním souboru k vytvoření počátečního bloku na kořenovém prostoru dbspace na disku. V rámci inicializace diskového prostoru inicializuje databázový server automaticky i sdílenou paměť. Diskový prostor je inicializován při prvním spuštění databázového serveru. Jeho další inicializace se pak provádějí jen během studeného obnovení nebo na požadavek správce databázového serveru.

**Upozornění:** Při inicializaci diskového prostoru dochází k přepisu veškerých dat původně uložených v tomto prostoru. Pokud znovu inicializujete diskový prostor existujícího databázového serveru, budou všechna původní data databázového serveru nedostupná a v podstatě zničená.

---

## Inicializace diskového prostoru

Než zahájíte inicializaci, musí být databázový server v režimu offline.

Spouštíte-li databázový server poprvé nebo chcete-li odstranit všechny prostory dbspace a k nim přidružená data, použijte k inicializaci diskového prostoru a k uvedení databázového serveru do režimu online následující metody.

| Operační systém | Kroky k uvedení databázového serveru do režimu online                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UNIX            | Chcete-li inicializovat databázový server, musíte být přihlášení jako uživatel <b>informix</b> nebo <b>root</b> . Spusťte příkaz <b>oninit -iy</b> .                                                                                                                                                                                                                                                                                               |
| Windows         | Chcete-li inicializovat databázový server, musíte být členem skupiny <b>Administrators</b> nebo <b>Power Users</b> . <ul style="list-style-type: none"><li>• Databázový server pracuje jako služba. V ovládacím panelu <b>Služby</b> vyberte v poli Parametry pro spuštění službu databázového serveru a jeho typ <b>-iy</b>. Pak klepněte na <b>Start</b>.</li><li>• Do příkazového řádku vložte příkaz <b>starts dbservername -iy</b>.</li></ul> |

**Upozornění:** Provedením těchto příkazů znehodnotíte všechna stávající data v diskovém prostoru databázového serveru. Spouštíte-li novou instanci databázového serveru, použijte jen příznak **-i**.

Chcete-li inicializovat sdílenou paměť a pak ponechat databázový server v klidovém režimu, můžete použít volbu **oninit -s**.

### Jen pro Windows

Když instalujete databázový server a rozhodnete se inicializovat novou instanci databázového serveru nebo když k vytvoření nové instance databázového serveru použijete program Server Instance Manager, bude databázový server inicializován.

Doporučení: K inicializaci databázového serveru nepoužívejte příkaz **oninit -iy**, s výjimkou případů, kdy diagnostikujete případný problém.

### Konec Jen pro Windows



Automatické spuštění systému není dostupné prostřednictvím administrace SQL. Databázový server musíte spustit podle informací uvedených v této části. Používáte-li při spuštění databázového serveru příkazem **oninit** rozhraní API pro administraci a plánovač, server automaticky provede všechny uložené procedury naplánované ke spuštění při spuštění serveru. Další informace naleznete v části Kapitola 26, “Automatické monitorování a nápravné akce”, na stránce 26-1.

Více informací o obslužném programu **oninit** naleznete v části *IBM Informix Dynamic Server Administrator's Reference*.

Více informací o užití těchto obslužných programů při inicializaci databázového serveru a změně režimu databázového serveru naleznete v části “Spuštění databázového serveru a inicializace diskového prostoru” na stránce 1-14.

## Jednotlivé kroky inicializace

Inicializace diskového prostoru vždy zahrnuje i inicializaci sdílené paměti. Avšak některé kroky, ke kterým dochází během inicializace sdílené paměti (například nahrání změn konfigurace), nejsou během inicializace disku nutné, protože tyto aktivity nesouvisí s novou inicializací disku.

Obrázek Tabulka 4-1 znázorňuje hlavní kroky obou druhů inicializace. Následující text obsahuje popisy jednotlivých kroků.

Tabulka 4-1. Kroky inicializace

| Inicializace sdílené paměti                                                      | Inicializace disku                                                               |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Zpracování konfiguračního souboru.                                               | Zpracování konfiguračního souboru.                                               |
| Vytvoření segmentů sdílené paměti.                                               | Vytvoření segmentů sdílené paměti.                                               |
| Inicializace struktur sdílené paměti.                                            | Inicializace struktur sdílené paměti.                                            |
|                                                                                  | Inicializace diskového prostoru.                                                 |
| Spuštění všech požadovaných virtuálních procesorů.                               | Spuštění všech požadovaných virtuálních procesorů.                               |
| Provedení potřebných převodů.                                                    |                                                                                  |
| Zahájení rychlé obnovy.                                                          |                                                                                  |
| Vyvolání kontrolního bodu.                                                       | Vyvolání kontrolního bodu.                                                       |
| Dokumentace změn konfigurace.                                                    |                                                                                  |
| Aktualizace souboru <b>oncfg_servername.servernum</b> .                          | Aktualizace souboru <b>oncfg_servername.servernum</b> .                          |
| Změna na klidový režim.                                                          | Změna na klidový režim.                                                          |
| Vypuštění dočasných prostorů tblspaces (nepovinné).                              |                                                                                  |
| Nastavení vynucené rezidence, je-li požadováno.                                  | Nastavení vynucené rezidence, je-li určena.                                      |
| Změna na režim online a vrácení řízení uživateli.                                | Změna na režim online a vrácení řízení uživateli.                                |
| Nejsou-li tabulky SMI aktuální, aktualizace tabulek.                             | Vytvoření databáze <b>sysmaster</b> obsahující tabulky SMI.                      |
|                                                                                  | Vytvoření databáze <b>sysutils</b> .                                             |
| Monitorování maximálního počtu uživatelských připojení v každém kontrolním bodu. | Monitorování maximálního počtu uživatelských připojení v každém kontrolním bodu. |

## Zpracování konfiguračního souboru

Databázový server použije konfigurační parametry při přidělování segmentů sdílené paměti v průběhu inicializace a restartu. Změníte-li konfigurační parametry sdílené paměti, musíte databázový server ukončit a restartovat, aby změny nabyly účinnosti.

Soubory **ONCONFIG**, **onconfig** a **onconfig.std** jsou uloženy v adresáři **\$INFORMIXDIR/etc** na platformě UNIX a v adresáři **%INFORMIXDIR%\etc** v systému Windows. Během inicializace vyhledá databázový server v následujících souborech hodnoty konfigurace, a to v tomto pořadí:

1. Pokud je nastavena proměnná prostředí **ONCONFIG**, načte databázový server hodnoty ze souboru **ONCONFIG**.  
Pokud je nastavena proměnná prostředí **ONCONFIG**, ale databázový server nemůže přistoupit k danému souboru, vrátí chybovou zprávu.
2. Není-li nastavena proměnná prostředí **ONCONFIG**, načte databázový server hodnoty konfigurace ze souboru **ONCONFIG**.
3. Vynecháte-li konfigurační parametry v souboru **onconfig**, databázový server načte hodnoty konfigurace ze souboru **\$INFORMIXDIR/etc/onconfig.std**.

Před každou inicializací nebo restartováním databázového serveru byste měli nastavit proměnnou prostředí **ONCONFIG**. Výchozí konfigurační soubory slouží jako šablony, ne jako funkční konfigurace. Pokud ale chybí soubor **onconfig\_std**, server nebude inicializován. Více informací o konfiguračním souboru naleznete v části “Konfigurace databázového serveru” na stránce 1-11.

Proces restartování porovná hodnoty v aktuálním konfiguračním souboru s předchozími hodnotami, uloženými na vyhrazené stránce **PAGE\_CONFIG** kořenového prostoru **dbspace**, jsou-li takové hodnoty k dispozici. Vyskytnou-li se rozdíly, použije databázový server při restartování hodnoty z aktuálního konfiguračního souboru **ONCONFIG**.

## Vytvoření segmentů sdílené paměti

Databázový server používá konfigurační hodnoty k vypočtení požadované velikosti rezidentní sdílené paměti databázového serveru. Databázový server kromě toho vypočítá z interních hodnot další požadavky konfigurace. Jsou vypočteny a uloženy režijní požadavky na prostor.

Aby mohl databázový server vytvořit sdílenou paměť, přidělí mu operační systém prostor pro tři různé typy paměti:

- Rezidentní část využívanou pro vyrovnávací paměť dat a interní tabulky.
- Virtuální část využívanou pro většinu systémových paměťových požadavků a požadavků uživatelských relací.

### Jen pro UNIX

- Komunikační část využívanou pro komunikaci typu IPC.

Databázový server přidělí tuto část sdílené paměti jen v případě, že konfiguruje připojení typu IPC prostřednictvím sdílené paměti.

### Konec Jen pro UNIX

Pak databázový server připojí segmenty sdílené paměti k jeho virtuálnímu prostoru adres a inicializuje struktury sdílené paměti. Více informací o strukturách sdílené paměti naleznete v části “Virtuální část sdílené paměti” na stránce 7-14.

Po dokončení inicializace a spuštění databázového serveru, může server podle potřeby vytvořit další segmenty sdílené paměti. Databázový server vytvoří segmenty v přírůstcích velikosti stránky.

## Inicializace a restartování sdílené paměti

Poté, co se databázový server připojí ke sdílené paměti, odstraní z prostoru sdílené paměti neinicializovaná data. Poté databázový server rozvrhne data záhlaví sdílené paměti a inicializuje data v strukturách sdílené paměti. Databázový server rozvrhne prostor potřebný pro vyrovnávací paměť logického protokolu, inicializuje struktury a propojí navzájem tři samostatné vyrovnávací paměti, které tvoří vyrovnávací paměť logického protokolu. Více informací o těchto strukturách naleznete v části věnované obslužnému programu **onstat** v části *IBM Informix Dynamic Server Administrator's Reference*.

Poté, co databázový server znovu zmapuje prostor sdílené paměti, zaregistruje nově spuštěné adresy a velikosti každé struktury v novém záhlaví sdílené paměti.

Inicializace sdílené paměti neovlivní struktury a rozvržení disku. Databázový server načte z disku potřebné adresní údaje, např. údaje o umístění logického a fyzického protokolu, a tyto informace pak použije k aktualizaci ukazatelů ve sdílené paměti.

## Inicializace diskového prostoru

Tento postup se provádí jen během inicializace diskového prostoru, nikoli při restartování databázového serveru. Po inicializaci struktur sdílené paměti zahájí databázový server inicializaci disku. Databázový server inicializuje všechny rezervované stránky, které uchovává v kořenovém prostoru dbspace na disku a zapíše na disk řídicí údaje stránky PAGE\_PZERO.

## Spuštění všech požadovaných virtuálních procesorů

Databázový server spustí všechny virtuální procesory, které potřebuje. Parametry v souboru ONCONFIG ovlivňují, které procesory budou spuštěny. Parametr NETTYPE může například ovlivnit počet a typ procesorů, které vytvářejí připojení. Více informací o virtuálních procesorech naleznete v části "Virtuální procesory" na stránce 5-2.

## Provedení potřebných konverzí

Databázový server zkontroluje své interní soubory. Pocházejí-li soubory z dřívější verze databázového serveru, aktualizuje tyto soubory na aktuální formát. Více informací o převodu databáze naleznete v části *IBM Informix Migration Guide*.

## Zahájení rychlé obnovy

Databázový server zkontroluje, je-li potřebná rychlá obnova a pokud ano, zahájí ji. Více informací o rychlé obnově naleznete v části "Rychlá obnova" na stránce 15-7.

Rychlá obnova není provedena během inicializace diskového prostoru, protože inicializovaný disk neobsahuje žádná data k obnovení.

## Zahájení kontrolního bodu

Databázový server po dokončení rychlé obnovy provede kontrolní bod, aby zajistil, že všechny obnovené transakce budou vyprázdněny na disk, a aby je nebylo nutné opakovat.

V rámci procedury kontrolního bodu zapíše databázový server zprávu o dokončení kontrolního bodu do protokolu zpráv. Více informací o kontrolních bodech naleznete "Kontrolní body" na stránce 15-5.

Databázový server se nyní uvede do klidového režimu nebo do režimu online, v závislosti na tom, jak jste spustili proces inicializace nebo restartování.

## Zaznamenání změn konfigurace

Databázový server porovná aktuální hodnoty uložené v konfiguračním souboru s hodnotami dříve uloženými na rezervované stránce kořenového prostoru dbspace PAGE\_CONFIG. Zjistí-li rozdíly, zaznamená databázový server obě hodnoty (novou i starou) do zprávy v protokolu zpráv.

Tato úloha se neprovádí během inicializace diskového prostoru nebo restartování.

## Vytvoření souboru oncfg\_servername.servernum

Databázový server vytvoří soubor **oncfg\_servername.servernum** a aktualizuje ho pokaždé, když přidáte nebo odstraníte prostory dbspace, blobspace, soubor logického protokolu nebo blok. S tímto souborem není nutno žádným způsobem manipulovat, ale můžete ho vidět v seznamu adresáře \$INFORMIXDIR/etc na platformě UNIX nebo v adresáři %INFORMIXDIR%\etc v systému Windows. Databázový server používá soubor **oncfg\_nazevserveru.cisloserveru** při úplném obnovení systému k záchraně logického protokolu.

Více informací o souboru **oncfg\_servername.servernum** naleznete v části věnované souborům, které používá databázový server, v části *IBM Informix Dynamic Server Administrator's Reference*.

## Vypuštění dočasných prostorů tblspace

Databázový server prohledá všechny prostory dbspace, zda neobsahují dočasné prostory tblspace. (Používáte-li k inicializaci databázového serveru volbu obslužného programu **-p oninit**, databázový server tento krok přeskočí.) Tyto dočasné prostory tblspace, najdete-li nějaké, jsou prostory tblspace zbylé po uživatelských procesech, které skončily předčasně a nemohly provést řádné vyčištění prostorů. Databázový server odstraní všechny dočasné prostory tblspace a uvolní prostor na disku. Více informací o dočasných prostorech tblspaces naleznete v části "Dočasné tabulky" na stránce 9-26.

Tento úkol se provádí při restartování databázového serveru, nikoli při inicializaci diskového prostoru.

## Nastavení vynucené rezidence, je-li určena

Pokud je hodnota konfiguračního parametru RESIDENT rovna -1 nebo je větší než 0, zkusí databázový server vynutit rezidenci sdílené paměti. Pokud systém hostitelského počítače nepodporuje vynucenou rezidenci, inicializační procedura pokračuje. Není vynucena rezidence a databázový server vyše chybovou zprávu do protokolu zpráv. Více informací o konfiguračním parametru RESIDENT naleznete v části *IBM Informix Dynamic Server Administrator's Reference*.

## Návrat řízení uživateli

Databázový server zapíše do protokolu zpráv zprávu IBM Informix Dynamic Server inicializován - inicializace disku dokončena. pouze při inicializaci, nikoli při restartování databázového serveru. Databázový server též dynamicky přidělí segment virtuální sdílené paměti.

V této chvíli je řízení vráceno uživateli. Všechny chybové zprávy generované během inicializační procedury se zobrazí na následujících místech:

- Příkazový řádek
- Soubor protokolu zpráv databázového serveru určený konfiguračním parametrem MSGPATH

Více informací o parametru MSGPATH naleznete v části *IBM Informix Dynamic Server Administrator's Reference*.

- Část **Souhrn** v programu IBM Informix Server Administrator (ISA)

## Vytvoření databáze sysmaster a příprava tabulek SMI

Ačkoliv databázový server vrátí řízení uživateli, jeho práce není u konce. Databázový server nyní zkontroluje *tabulky* (SMI) rozhraní monitorování systému. Pokud tabulky SMI nejsou aktuální, databázový server provede jejich aktualizaci. Pokud tabulky SMI neexistují, například po inicializaci disku, databázový server vytvoří tabulky. Po vytvoření tabulek SMI, databázový server vloží do protokolu zpráv zprávu **Databáze sysmaster úspěšně vytvořena**. Databázový server znovu vytvoří databázi **sysmaster** i během převodu a opětovného vrácení. Více informací o tabulkách SMI naleznete v kapitole o databázi **sysmaster** v části *IBM Informix Dynamic Server Administrator's Reference*.

Pokud databázový server vypnete předtím, než dokončí vytváření tabulek SMI, proces vytváření tabulek se zastaví. Tato skutečnost nezpůsobí poškození databázového serveru. Databázový server vytvoří tabulky SMI po příštím uvedení do režimu online. Pokud ale neumožníte dokončení vytváření tabulek SMI, nemůžete v těchto tabulkách spustit žádné dotazy a k zálohování nemůžete používat obslužný program ON-Bar.

Po vytvoření tabulek SMI je databázový server připraven k použití. Databázový server je spuštěn, dokud ho nezastavíte nebo dokud nevykáže poruchu.

Doporučení: *Nepokoušejte se* zastavovat databázový server odstraněním virtuálního procesoru nebo jiného procesu databázového serveru. Více informací naleznete v části "Spuštění a zastavení virtuálních procesorů" na stránce 6-3.

## Vytvoření databáze sysutils

Během inicializace disku, převodu nebo opětovného vrácení databázový server vypustí a znovu vytvoří databázi **sysutils**. Obslužný program ON-Bar uloží zálohovací a obnovovací data do databáze **sysutils**. Vyčkejte, dokud se v protokolu zpráv nezobrazí zpráva **databáze sysutils úspěšně vytvořena**. Další informace naleznete v příručce *IBM Informix Backup and Restore Guide*.

## Monitorování maximálního počtu uživatelských připojení

V každém kontrolním bodu zobrazí databázový server v protokolu zpráv maximální počet uživatelských připojení: **max. připojení k serveru číslo**. Máte možnost monitorovat počet uživatelů, kteří se přihlásili k databázovému serveru od posledního restartu nebo inicializace disku.

Zobrazené číslo je při opětovné inicializaci databázového serveru resetováno.

---

## Provozní režimy databázového serveru

Aktuální režim databázového serveru lze určit pomocí obslužného programu **onstat** z příkazového řádku. Záhlaví **onstat** zobrazí příslušný režim.

Obrázek Tabulka 4-2 znázorňuje hlavní provozní režimy databázového serveru.

Tabulka 4-2. Provozní režimy

| Provozní režim     | Popis                                                                                                                                                                                                                                                                                                            | Uživatelé s oprávněním k přístupu                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| režim offline      | Databázový server nepracuje. Není přidělena sdílená paměť.                                                                                                                                                                                                                                                       | Pouze administrátor (uživatel <b>informix</b> ) může přepnout z tohoto režimu na jiný.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| klidový režim      | Jsou spuštěny procesy databázového serveru a přidělené zdroje sdílené paměti.<br><br>Administrátoři využívají tento režim k provedení udržovacích funkcí, které nevyžadují spuštění příkazů jazyka SQL a DDL.                                                                                                    | Na databázový server může přistupovat jen administrátor (uživatel <b>informix</b> ).<br><br>Jiní uživatelé mohou zobrazit údaje o stavu databázového serveru, ale nemohou na něj přistupovat.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| režim administrace | Tento režim je přechodným režimem mezi klidovým režimem a režimem online.<br><br>Administrátoři používají tento režim k provedení veškerých udržovacích úloh včetně těch, které vyžadují spuštění příkazů v jazyku SQL a DDL. Administrátoři mohou též provádět všechny ostatní funkce dostupné v režimu online. | V režimu administrace se k databázovému serveru mohou připojovat následující uživatelé: <ul style="list-style-type: none"> <li>• Uživatel <b>informix</b>.</li> <li>• Uživatelé s rolí DBSA.</li> </ul> Pokud chcete, aby se k databázovému serveru mohli v režimu administrace připojovat uživatelé s rolí DBSA (namísto uživatele <b>informix</b> ), nastavte na hodnotu 1 konfigurační parametr <code>ADMIN_USER_MODE_WITH_DBSA</code> . <ul style="list-style-type: none"> <li>• Jeden nebo více uživatelů, kteří mají přístup k režimu administrace.</li> </ul> Uživatel <b>informix</b> a administrátor DBSA mohou dynamicky udělovat jednomu nebo více konkrétním uživatelům oprávnění k připojení k databázovému serveru v režimu administrace pomocí příkazu <b>onmode -j</b> , pomocí příkazu <b>oninit -U</b> nebo pomocí konfiguračního parametru <code>ADMIN_MODE_USERS</code> .<br><br>Jiní uživatelé mohou zobrazit údaje o stavu databázového serveru, ale nemohou na něj přistupovat. |
| režim online       | Je běžným provozním režimem databázového serveru.                                                                                                                                                                                                                                                                | Kterýkoliv oprávněný uživatel se může připojit k databázovému serveru a provést kteroukoliv z databázových aktivit.<br><br>Uživatelé <b>informix</b> nebo <b>root</b> mohou využít obslužných programů příkazového řádku a změnit celou řadu hodnot parametrů databázového serveru <code>ONCONFIG</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

Kromě toho se může databázový server nacházet v jednom z následujících režimů:

- *Režim pouze pro čtení* je využíván sekundárním databázovým serverem v prostředí replikace dat. Aplikace mohou odesílat dotazy na sekundární databázový server v režimu pouze pro čtení, ale nemohou zapisovat do databáze určené pouze ke čtení.
- *Režim obnovy* je přechodným režimem. Vyskytuje se v případech, kdy databázový server provádí rychlou obnovu nebo se obnovuje ze systémového archivu či obnovení systému. K obnově dochází během přepnutí z režimu offline do klidového režimu.
- *Režim vypnutí* je rovněž přechodným režimem. Databázový server se v něm nachází při přechodu z režimu online do klidového režimu nebo při přechodu z režimu online (či klidového) do režimu offline. Stávající uživatelé mohou přistupovat k systému, novým uživatelům však není povolen přístup.

Jakmile je iniciován režim vypnutí, nelze ho zrušit.

---

## Změna provozních režimů databázového serveru

Tato část popisuje, jak přepínat mezi jednotlivými režimy databázového serveru pomocí obslužných programů **oninit**, **onmode** a **ISA**. Obsahuje také informace o používání konfiguračního parametru **ADMIN\_MODE\_USER**, který určuje, kteří uživatelé se mohou k serveru připojovat v režimu administrace.

### Jen pro Windows

V systému Windows pracuje databázový server jako služba. Systém Windows obsahuje aplikaci k řízení služeb (nazývanou též nástroj Služby), která slouží ke spuštění, zastavení a přerušení služby. Aplikace k řízení služeb se nachází v programové skupině Ovládací panely. Název služby databázového serveru obsahuje název databázového serveru (hodnota parametru **DBSERVERNAME** v souboru **ONCONFIG**). Služba Dynamic Server databázového serveru **newyork** je například:

IBM Informix Database Server - newyork

Chcete-li změnit režim pomocí nástroje Služby, spusíte nástroj a vyberte službu databázového serveru. Pak vyberte vhodné tlačítko v okně Služby. Tabulky uvedené v další části této kapitoly vysvětlují, které tlačítko zvolit pro jednotlivé režimy.

Ke spuštění a zastavení databázového serveru můžete použít i jiné nástroje systému Windows, například příkaz **NET** a nástroj **Server Manager**. Více informací o těchto metodách naleznete v dokumentaci operačního systému Windows.

### Konec Jen pro Windows

**Rada:** Poté, co změníte režim databázového serveru, proveďte příkaz **onstat** a prověřte aktuální stav serveru.

## Uživatelé oprávnění měnit režimy

### Jen pro UNIX

Provozní režim databázového serveru mohou měnit uživatelé přihlášení jako uživatel **root** nebo **informix**. Pokud je nastavena proměnná prostředí **INF\_ROLE\_SEP**, může administrátor databázového serveru měnit také provozní režim serveru. Proměnná prostředí **INF\_ROLE\_SEP** vynucuje rozdělení úloh administrace mezi uživatele, které provozují databázový server a provádějí audit provozu databázového serveru.

Pokud chcete, aby se k databázovému serveru mohli v režimu administrace připojovat uživatelé s rolí **DBSA**, nastavte konfigurační parametr **ADMIN\_USER\_MODE\_WITH\_DBSA** na hodnotu 1. Pokud tento parametr nastavíte na hodnotu nula, bude přístup omezen pouze na uživatele **informix**.

Uživatel **informix** a administrátor **DBSA** mohou dynamicky udělovat jednomu nebo více konkrétním uživatelům oprávnění k připojení k databázovému serveru v režimu administrace pomocí obslužného programu **onmode**, pomocí obslužného programu **oninit** nebo pomocí konfiguračního parametru **ADMIN\_MODE\_USERS**.

### Jen pro Windows

Tabulka Tabulka 4-2 na stránce 4-8 znázorňuje, kteří uživatelé mohou měnit provozní režim databázového serveru v systému Windows. Používáte-li obslužný program ISA, můžete se přihlásit do operačního systému jako kterýkoliv uživatel, ale k serveru Apache se musíte přihlásit jako uživatel **informix**. Server Apache je spuštěn jako člen skupiny **Informix-Admin**.

Tabulka 4-3. Změna provozních režimů v systému Windows.

| Změna provozního režimu                                   | Skupina administrátorů | skupina Informix-Admin |
|-----------------------------------------------------------|------------------------|------------------------|
| Obslužné programy příkazového řádku jako je <b>starts</b> |                        | X                      |
| ISA                                                       |                        | X                      |
| Ovládací panel <b>Služby</b>                              | X                      |                        |

### Konec Jen pro Windows

## Volby programu ISA pro změnu provozních režimů

Ke změně režimu databázového serveru můžete použít program ISA. Další informace naleznete v nápovědě online k programu ISA.

| Akce                                                                 | Volba programu ISA                                                                        |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Inicializace databázového serveru.                                   | <b>Režim &gt; Online (inicializace disků)</b><br>nebo <b>Klidový (inicializace disků)</b> |
| Změna z režimu offline nebo z režimu administrace na klidový režim.  | <b>Režim &gt; Klidový</b>                                                                 |
| Změna z kteréhokoliv režimu na režim administrace.                   | <b>Režim &gt; Jednouzivatelský</b>                                                        |
| Změna z režimů offline, klidového nebo administrace na režim online. | <b>Režim &gt; Online</b>                                                                  |
| Nenásilná změna z režimu online do klidového režimu.                 | <b>Režim &gt; Klidový (nenásilný)</b>                                                     |
| Okamžitá změna z režimu online do klidového režimu.                  | <b>Režim &gt; Klidový (okamžitý)</b>                                                      |
| Vypnutí databázového serveru.                                        | <b>Režim &gt; Offline</b>                                                                 |

## Volby programu ON-Monitor pro změnu režimů (UNIX)

Ke změně režimu databázového serveru můžete v systému UNIX použít program ON-Monitor. Více informací naleznete v části věnované obslužnému programu ON-Monitor v publikaci *IBM Informix Dynamic Server Administrator's Reference*.



| Akce                                                          | Volba nabídky               |
|---------------------------------------------------------------|-----------------------------|
| Inicializace databázového serveru.                            | Parametry > Inicializace    |
| Změna z režimu offline do klidového režimu.                   | Režim > Spuštění            |
| Změna z režimu offline nebo klidového režimu na režim online. | Režim > Online              |
| Nenásilná změna z režimu online do klidového režimu.          | Režim > Nenásilné vypnutí   |
| Okamžitá změna z režimu online do klidového režimu.           | Režim > Okamžité vypnutí    |
| Změna na režim administrace.                                  | Režim > Jednouživatelský    |
| Vypnutí databázového serveru.                                 | Režim > Přepnout na offline |

## Volby příkazového řádku pro změnu režimů

Obrázek Tabulka 4-2 na stránce 4-8 obsahuje popisy všech režimů a znázorňuje, kteří uživatelé mohou přistupovat na databázový server v jednotlivých jeho provozních režimech. Tato část, která obsahuje příkazy ke změně provozních režimů a informace o tom, jak změny režimů ovlivňují relace uživatelů, se skládá z následujících podčástí:

- “Změna z režimu offline do klidového režimu”
- “Změna z režimu offline do režimu online”
- “Změna z režimu offline do režimu administrace” na stránce 4-12
- “Změna z klidového režimu do režimu online” na stránce 4-12
- “Nenásilná změna z režimu online do klidového režimu” na stránce 4-12
- “Okamžitá změna z režimu online do klidového režimu” na stránce 4-13
- “Změna z klidového režimu nebo režimu online do režimu administrace” na stránce 4-13
- “Změna z režimu administrace do režimu online” na stránce 4-14
- “Změna z režimu administrace do klidového režimu” na stránce 4-14
- “Okamžitá změna z jakéhokoli režimu do režimu offline” na stránce 4-14

Další informace naleznete také v části “Určení uživatelů režimu administrace pomocí konfiguračního parametru ADMIN\_MODE\_USERS” na stránce 4-15.

### Změna z režimu offline do klidového režimu

Když databázový server přepne z režimu offline na klidový režim, databázový server provede inicializaci sdílené paměti. K databázovému serveru mají přístup pouze administrátoři, kteří mohou provádět udržovací funkce nezahrnující provedení příkazů v jazyku SQL a DDL.

#### *Operační systém*

UNIX

#### *Akce*

- Proveďte příkaz **oninit -s**.

Windows

- V příkazovém řádku použijte příkaz **starts dbservername -s**.

### Změna z režimu offline do režimu online

Přepnete-li databázový server z režimu offline do režimu online, databázový server inicializuje sdílenou paměť a je dostupný pro všechny uživatelské relace.

#### *Operační systém*

UNIX

#### *Akce*

- Proveďte příkaz **oninit**.

Windows

- V nástroji Služby vyberte službu databázového serveru a klepněte na příkaz **Start**.
- V příkazovém řádku použijte příkaz **starts dbservername**.

## Změna z režimu offline do režimu administrace

Když přepínáte databázový server z režimu offline do režimu administrace, přepínáte ho do režimu, který mohou využívat jen administrátoři, chtějí-li provést funkce databázového serveru a funkce údržby včetně těch, které zahrnují provedení příkazů jazyka SQL a DDL.

*Operační systém*

*Akce*

UNIX nebo Windows

- Proveďte příkaz **oninit -j**.

Uživatel **informix** nebo administrátor DBSA může pomocí příkazu **oninit -U** určit seznam uživatelů s přístupem k režimu administrace, jak znázorňuje následující příklad:

```
oninit -U mark,ajay,carol
```

Uživatelé určení pomocí příkazu **oninit -U** se mohou připojit po dobu trvání instance serveru nebo do spuštění příkazu **onmode -j -U**, který tento seznam uživatelů změní. Pokud chcete odebrat všechny uživatele ze seznamu, spusíte příkaz **onmode -j -U** s mezerou místo jmen, jak znázorňuje následující příklad:

```
oninit -U " "
```

Další informace naleznete také v části “Určení uživatelů režimu administrace pomocí konfiguračního parametru ADMIN\_MODE\_USERS” na stránce 4-15.

## Změna z klidového režimu do režimu online

Když databázový server přepnete z klidového režimu do režimu online, všechny relace získají přístup.

V případě, že jste databázový server již přepnuli z režimu online do klidového a nyní ho uvádíte zpět do režimu online, budou muset všichni uživatelé, jejichž zpracování bylo dříve přerušeno, znovu vybrat příslušnou databázi a znovu deklarovat kurzor.

*Operační systém*

*Akce*

UNIX nebo Windows

- Proveďte příkaz **onmode -m**.

Pouze systém Windows

- Pomocí nástroje Služby vyberte službu databázového serveru a klepněte na příkaz **Pokračovat**.

## Nenásilná změna z režimu online do klidového režimu

Databázový server převedte nenásilně z režimu online do klidového režimu, chcete-li omezit přístup uživatelů bez přerušování aktuálně probíhajícího zpracování.

Poté, co provedete tuto úlohu, nastaví databázový server příznak, který zabrání novým relacím v přístupu na databázový server. Aktuálním relacím je umožněno dokončit zpracování.

Jakmile zahájíte změnu režimu, nelze změnu již zrušit. Během změny režimu z režimu online do klidového se databázový server nachází v režimu vypnutí.

*Operační systém*

*Akce*

UNIX nebo Windows

- Proveďte příkazy **onmode -s** nebo **onmode -sy**.

pouze systém *Windows*

- Pomocí nástroje Služby vyberte službu databázového serveru a klepněte na **Pozastavit**.

### Okamžitá změna z režimu online do klidového režimu

Databázový server uveďte okamžitě z režimu online do klidového režimu, chcete-li zamezit přístup k databázovému serveru co nejrychleji. Nedokončené procesy budou ztraceny.

Výzva požádá o potvrzení okamžitého vypnutí. Potvrdíte-li ji, databázový server vyšle všem relacím připojeným ke sdílené paměti signál "odpojit". Pokud relace neodbrzí signál k odpojení nebo není schopna reagovat automaticky během 10 s, databázový server relaci ukončí.

Uživatelé databázového serveru obdrží buď chybovou zprávu -459, která oznamuje, že databázový server byl vypnut, nebo chybovou zprávu -457 oznamující, že jejich relace byla neočekávaně ukončena.

Databázový server odebere všechny relace, který ukončil. Aktivní transakce jsou odvolány.

#### *Operační systém*

#### *Akce*

UNIX nebo Windows

- Proveďte příkazy **onmode -u** nebo **onmode -uy**.  
Volba **-y** eliminuje potřebu potvrdit výzvu.

### Změna z klidového režimu nebo režimu online do režimu administrace

Přepnete-li databázový server z klidového režimu nebo režimu online na režim administrace, uvedete server do režimu, který mohou používat pouze administrátoři.

Začnete-li v režimu online, databázový server automaticky odhlásí všechny uživatele, kteří jsou přihlášení pod ID jiným, než je uživatel **informix** a uživatelé obdrží chybovou zprávu. Pokud je spojení ukončeno během transakce, databázový server odvolá transakci.

Na režim administrace přepněte tehdy, pokud chcete provést příkazy jazyka SQL a DDL a zároveň nechcete, aby byli připojeni další uživatelé.

#### *Operační systém*

#### *Akce*

UNIX nebo Windows

- Proveďte příkaz **onmode -j**.

Uživatel **informix** nebo administrátor DBSA mohou pomocí volby **onmode -j -U** udělit jednotlivým uživatelům oprávnění k přístupu k databázovému serveru v režimu administrace.

Spuštěním následujícího příkazu například umožníte třem jednotlivým uživatelům připojení a přístup k databázovému serveru, dokud databázový server nezmění režim na režim offline, klidový režim nebo režim online:

```
onmode -j -U mark,ajay,carol
```

Po připojení mohou tito jednotliví uživatelé spouštět jakékoli příkazy jazyka SQL nebo DDL. Když server přejde do režimu administrace, jsou těm uživatelům, kteří nebyli identifikováni příkazem **onmode -j -U**, přerušena všechna připojení uživatelských relací.

Po počátečním spuštění příkazu **onmode -j -U** můžete jednotlivé uživatele odstranit spuštěním příkazu **onmode -j -U** s uživatelskými jmény odstraněnými z nového seznamu, například spuštěním příkazu:

```
onmode -j -U mark,carol
```

Spuštěním příkazu **onmode -j -U** s mezerou místo uživatelského jména odstraníte ze seznamu všechny uživatele, jak znázorňuje následující příklad:

```
oninit -U " "
```

Další informace naleznete také v části “Určení uživatelů režimu administrace pomocí konfiguračního parametru ADMIN\_MODE\_USERS” na stránce 4-15.

## Změna z režimu administrace do režimu online

Když databázový server přepnete z režimu administrace na režim online, mohou na databázový server přistupovat všichni uživatelé.

### *Operační systém*

### *Akce*

UNIX nebo Windows

- Proveďte příkaz **onmode -m**.

## Změna z režimu administrace do klidového režimu

Když přepínáte databázový server z režimu administrace do klidového režimu, přepínáte ho do režimu, který mohou využívat jen administrátoři, chtějí-li provést funkce údržby nezahrnující příkazy jazyka SQL a DDL.

### *Operační systém*

### *Akce*

UNIX nebo Windows

- Proveďte příkaz **onmode -s**.

## Okamžitá změna z jakéhokoli režimu do režimu offline

Databázový server lze z kteréhokoliv režimu okamžitě přepnout na režim offline. Po uvedení databázového serveru do režimu offline můžete server restartovat v klidovém režimu, v režimu administrace nebo v režimu online. Při restartu provede databázový server rychlou obnovu, aby byla zajištěna logická konzistence dat.

Výzva požádá o potvrzení uvedení do režimu offline. Potvrdíte-li ji, databázový server iniciuje požadavek kontrolního bodu a všem relacím připojeným ke sdílené paměti vyšle signál "odpojit". Pokud relace neobdrží signál k odpojení nebo není schopna reagovat automaticky během 10 s, databázový server relaci ukončí.

Uživatelé databázového serveru obdrží buď chybovou zprávu -459, která oznamuje, že databázový server byl vypnut, nebo chybovou zprávu -457 oznamující, že jejich relace byla neočekávaně ukončena.

Po uvedení databázového serveru do režimu offline můžete server restartovat v klidovém režimu, v režimu administrace nebo v režimu online. Při restartu provede databázový server rychlou obnovu, aby byla zajištěna logická konzistence dat.

Databázový server odebere všechny relace, které ukončil. Aktivní transakce jsou odvolány.

### *Operační systém*

### *Akce*

UNIX nebo Windows

- Proveďte příkazy **onmode -k** nebo **onmode -ky**. Volba **-y** eliminuje automatickou výzvu k potvrzení okamžitého vypnutí.

pouze systém *Windows*

- V nástroji Služby vyberte službu databázového serveru a klepněte na příkaz **Stop**.

## Určení uživatelů režimu administrace pomocí konfiguračního parametru **ADMIN\_MODE\_USERS**

Konfigurační parametr **ADMIN\_MODE\_USERS** umožňuje určit, kteří uživatelé se mohou k databázovému serveru připojit v režimu administrace. Na rozdíl od příkazů **oninit** a **onmode**, které umožňují určit uživatele režimu administrace pouze do změny režimu serveru na režim offline, klidový režim nebo režim online, konfigurační parametr **ADMIN\_MODE\_USERS** uchovává seznam uživatelů režimu administrace po neurčitou dobu.

Chcete-li vytvořit seznam uživatelů režimu administrace, který bude uchováván v souboru **ONCONFIG**, určete jako hodnotu konfiguračního parametru **ADMIN\_MODE\_USERS** seznam uživatelů oddělený čárkami, například **mark,ajay,carol**.

K potlačení parametru **ADMIN\_MODE\_USERS** v průběhu relace použijte příkaz **onmode -wf**, jak znázorňuje následující příklad:

```
onmode -wf ADMIN_MODE_USERS=sharon,ka|pana
```

Konfigurační parametr **ADMIN\_MODE\_USERS** rozšiřuje seznam osob, které mají povolen přístup k databázovému serveru v režimu administrace. Osoby uvedené v příkazu **onmode** potlačují osoby uvedené v souboru **ONCONFIG**.



---

## **Část 2. Správa disků, paměti a procesů**





---

## Kapitola 5. Virtuální procesory a jednotkové procesy

|                                                                                                                           |      |
|---------------------------------------------------------------------------------------------------------------------------|------|
| Obsah kapitoly                                                                                                            | 5-2  |
| Virtuální procesory                                                                                                       | 5-2  |
| Jednotkové procesy                                                                                                        | 5-2  |
| Typy virtuálních procesorů                                                                                                | 5-3  |
| Výhody virtuálních procesorů                                                                                              | 5-5  |
| Sdílení zpracování                                                                                                        | 5-6  |
| Úspory paměti a zdrojů                                                                                                    | 5-6  |
| Paralelní zpracování                                                                                                      | 5-6  |
| Přidávání a vypouštění virtuálních procesorů v režimu online                                                              | 5-7  |
| Vázání virtuálních procesorů k jednotce CPU                                                                               | 5-7  |
| Obsluha jednotkových procesů virtuálními procesory                                                                        | 5-8  |
| Řídící struktury                                                                                                          | 5-8  |
| Přepínání kontextu                                                                                                        | 5-8  |
| Zásobníky                                                                                                                 | 5-9  |
| Fronty                                                                                                                    | 5-10 |
| Fronty připravených procesů                                                                                               | 5-10 |
| Fronty spících procesů                                                                                                    | 5-11 |
| Fronty čekajících procesů                                                                                                 | 5-11 |
| Objekty mutex                                                                                                             | 5-12 |
| Třídy virtuálních procesorů                                                                                               | 5-12 |
| Virtuální procesory CPU                                                                                                   | 5-12 |
| Stanovení počtu potřebných virtuálních procesorů CPU                                                                      | 5-12 |
| Spuštění na víceprocesorovém počítači                                                                                     | 5-13 |
| Spuštění na jednoprocessorovém počítači                                                                                   | 5-13 |
| Přidávání a vypouštění virtuálních procesorů CPU v režimu online                                                          | 5-14 |
| Jak zabránit stárnutí priority                                                                                            | 5-14 |
| Užití procesorové afinity                                                                                                 | 5-14 |
| Uživatелеm definované třídy virtuálních procesorů                                                                         | 5-15 |
| Stanovení potřebného počtu uživatelem definovaných virtuálních procesorů                                                  | 5-16 |
| Užití uživatelem definovaných virtuálních procesorů                                                                       | 5-16 |
| Určení uživatelem definovaného virtuálního procesoru                                                                      | 5-16 |
| Přirazení UDR k uživatelem definované třídě virtuálních procesorů                                                         | 5-16 |
| Přidávání a vypouštění uživatelem definovaných virtuálních procesorů v režimu online                                      | 5-17 |
| Virtuální procesory Java                                                                                                  | 5-17 |
| Virtuální procesory diskového vstupu - výstupu                                                                            | 5-17 |
| Priority vstupu - výstupu                                                                                                 | 5-18 |
| Vstup - výstup logického protokolu                                                                                        | 5-18 |
| Vstup - výstup fyzického protokolu                                                                                        | 5-19 |
| Asynchronní vstup - výstup                                                                                                | 5-19 |
| Síťové virtuální procesory                                                                                                | 5-22 |
| Určení síťových připojení                                                                                                 | 5-22 |
| Spouštění jednotkových procesů cyklického dotazování na virtuálních procesorech CPU nebo síťových virtuálních procesorech | 5-23 |
| Stanovení počtu síťových virtuálních procesorů                                                                            | 5-23 |
| Určení jednotkových procesů listen a cyklického dotazování pro připojení klienta k serveru                                | 5-23 |
| Použití rychlého cyklického dotazování                                                                                    | 5-25 |
| Spuštění několika jednotkových procesů typu listener                                                                      | 5-26 |
| Virtuální procesor Communications Support Module                                                                          | 5-27 |
| Šifrovací virtuální procesory                                                                                             | 5-27 |
| Optický virtuální procesor                                                                                                | 5-28 |
| Virtuální procesor typu audit                                                                                             | 5-28 |

## Obsah kapitoly

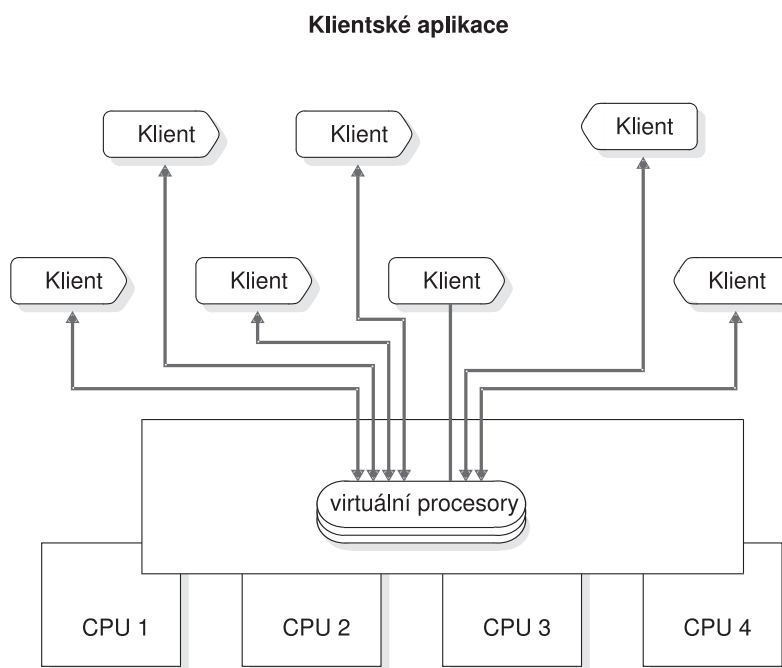
Tato kapitola popisuje virtuální procesory, vysvětluje, jak se ve virtuálních procesorech spouštějí jednotkové procesy a vysvětluje, jak databázový server využívá virtuální procesory a jednotkové procesy pro zlepšení výkonu.

## Virtuální procesory

Procesy databázových serverů jsou nazývány *virtuální procesory*, protože způsob jejich funkce je podobný způsobu, jakým v počítači pracuje jednotka CPU. Podobně jako jednotka CPU provádí řadu procesů operačního systému a obsluhuje tak řadu uživatelů, virtuální procesor databázového serveru provádí řadu *jednotkových procesů*, kterými obsluhuje řadu klientských aplikací používajících jazyk SQL .

Virtuální procesor je proces, jehož zpracování plánuje operační systém.

Obrázek 5-1 znázorňuje vztah klientských aplikací a virtuálních procesorů. Malý počet virtuálních procesorů obsluhuje podstatně větší počet klientských aplikací nebo dotazů.



Obrázek 5-1. Virtuální procesory.

## Jednotkové procesy

Jednotkový proces je úlohou virtuálního procesoru ve stejném smyslu, v jakém je virtuální procesor úlohou jednotky CPU. Virtuální procesor je úloha, kterou naplňuje ke zpracování v jednotce CPU operační systém, jednotkový proces databázového serveru je úloha, kterou virtuální procesor naplňuje ke zpracování interně. Jednotkové procesy se někdy nazývají *odlehčené procesy*, protože se sice podobají procesům, ale jejich zpracování je pro operační systém méně náročné.

Virtuální procesory databázového serveru jsou procesy *s vícenásobnými jednotkovými procesy*, protože spouštějí několik jednotkových procesů souběžně.

Následující text popisuje povahu jednotkových procesů.

| Operační systém | Popis jednotkového procesu                                                                                                                                                                                                                                                      |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UNIX            | Jednotkový proces je úloha, kterou virtuální procesor plánuje ke zpracování interně.                                                                                                                                                                                            |
| Windows         | Jednotkový proces je úloha, kterou virtuální procesor plánuje ke zpracování interně. Protože je virtuální procesor v systému Windows implementován jako jednotkový proces, jsou jednotkové procesy databázového serveru spouštěné v rámci jednotkových procesů systému Windows. |

**Důležité:** V celé této kapitole odkazují veškeré zmínky o “jednotkových procesech” na jednotkové procesy, které vytváří, plánuje a odstraňuje databázový server. Veškeré zmínky o jednotkových procesech systému “Windows” odkazují na jednotkové procesy, které vytváří, plánuje a odstraňuje systém Windows.

Virtuální procesor spouští jednotkové procesy pro klientské aplikace jazyka SQL (jednotkové procesy *relací*) a také jednotkové procesy ke splnění interních požadavků (*interní* jednotkové procesy). Ve většině případů spouští databázový server jeden jednotkový proces relace pro každé připojení klientské aplikace. Databázový server spouští interní jednotkové procesy pro vykonání vlastních úloh, jako jsou vstupy - výstupy databáze, vstupy - výstupy protokolování, čištění stránek a úlohy administrace. Případy, ve kterých databázový server spouští vícenásobné jednotkové procesy pro jediného klienta naleznete v části “Paralelní zpracování” na stránce 5-6.

*Uživatelský jednotkový proces* je jednotkový proces databázového serveru, který obsluhuje požadavky klientských aplikací. Uživatelské jednotkové procesy zahrnují jednotkové procesy relací nazývané jednotkové procesy **sqlexec**, které jsou primárními jednotkovými procesy spuštěnými databázovým serverem k obsluze klientských aplikací.

Uživatelské jednotkové procesy zahrnují rovněž jednotkové procesy, které obsluhují požadavky obslužného programu **onmode**, jednotkové procesy pro obnovu, jednotkové procesy prohledávání B-stromu a jednotkové procesy čištění stránek.

K zobrazení aktivních uživatelských jednotkových procesů použijte obslužný program **onstat -u**. Další informace o monitorování relací a jednotkových procesů naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

## Typy virtuálních procesorů

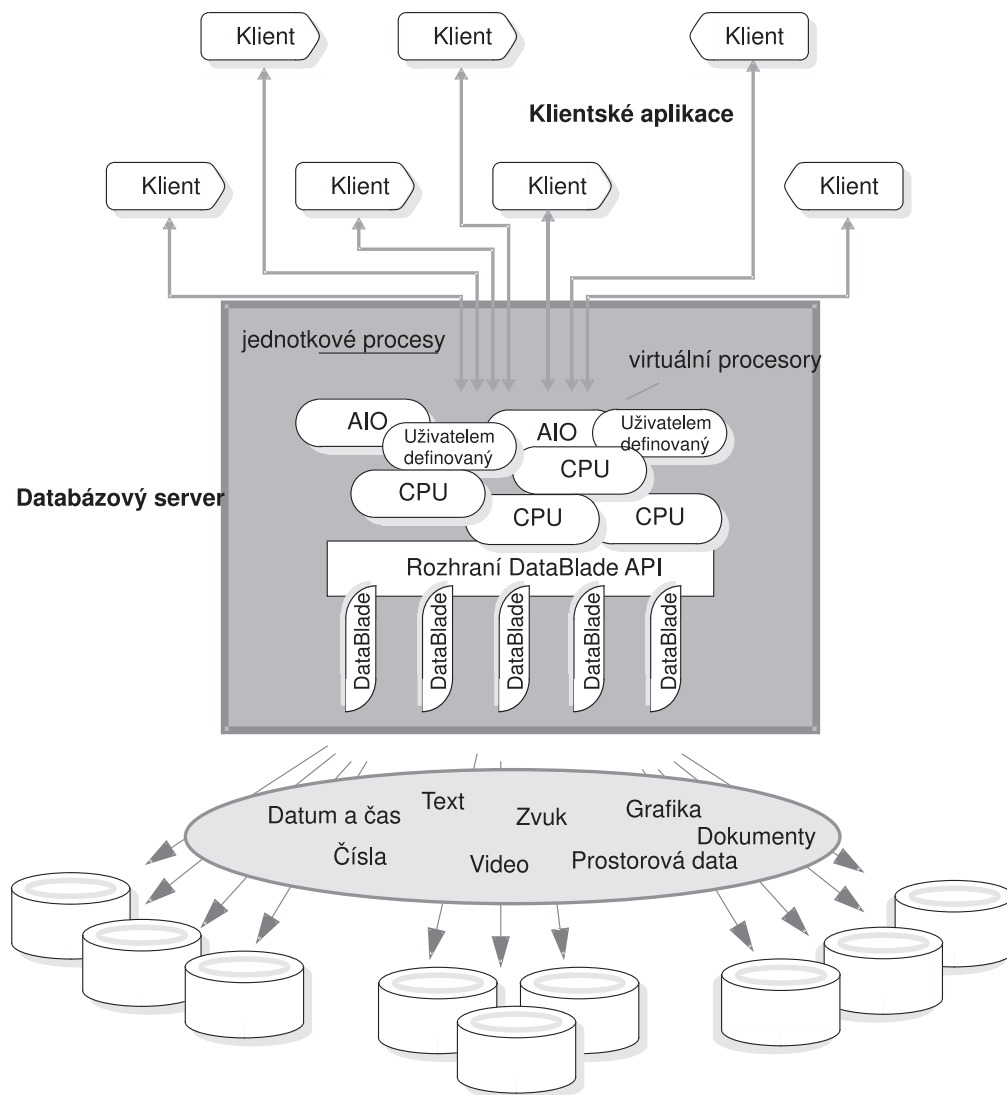
Obrázek Tabulka 5-1 znázorňuje *třídy* virtuálních procesorů a druhy zpracování, které provádějí. Každá třída virtuálního procesoru je vyhrazena pro zpracování určitého typu jednotkových procesů.

Počet virtuálních procesorů konfigurovaných pro kteroukoli třídu je závislý na počtu dostupných fyzických procesorů (jednotek CPU), velikosti fyzické paměti a používaných databázových aplikacích.

Tabulka 5-1. Třídy virtuálních procesorů

| <b>Třída virtuálního procesoru</b> | <b>Kategorie</b>             | <b>Účel</b>                                                                                                                                                                                                                                                             |
|------------------------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU                                | Centrální zpracování         | Zpracovává všechny jednotkové procesy relací a některé systémové jednotkové procesy. Zpracovává jednotkový proces pro asynchronní vstup - výstup jádra (KAIO), je-li dostupný. V závislosti na konfiguraci může spouštět jeden jednotkový proces cyklického dotazování. |
| PIO                                | Diskový vstup - výstup       | Zapíše do souboru fyzického protokolu (interní třída), nachází-li se tento v předpřipraveném diskovém prostoru.                                                                                                                                                         |
| LIO                                | Diskový vstup - výstup       | Zapíše do souborů logického protokolu (interní třída), nacházejí-li se tyto v předpřipraveném diskovém prostoru.                                                                                                                                                        |
| AIO                                | Diskový vstup - výstup       | Zajišťuje neprotokolující diskový vstup - výstup. Pokud je využíván asynchronní vstup - výstup jádra, zajišťují virtuální procesory třídy AIO vstup - výstup předpřipravených diskových prostorů.                                                                       |
| SHM                                | Síť                          | Zajišťuje komunikaci prostřednictvím sdílené paměti.                                                                                                                                                                                                                    |
| TLI                                | Síť                          | Využívá rozhraní TLI (Transport Layer Interface) k zajištění síťové komunikace.                                                                                                                                                                                         |
| SOC                                | Síť                          | Využívá sokety k zajištění síťové komunikace.                                                                                                                                                                                                                           |
| OPT (systém UNIX)                  | Optický podsystém            | Zajišťuje vstup - výstup optického disku.                                                                                                                                                                                                                               |
| ADM                                | Administrativní              | Zajišťuje administrativní funkce.                                                                                                                                                                                                                                       |
| ADT                                | Prověřování                  | Zajišťuje funkce auditu.                                                                                                                                                                                                                                                |
| MSC                                | Různé                        | Obsluhuje požadavky na systémová volání vyžadující velmi velký zásobník.                                                                                                                                                                                                |
| CSM                                | Modul pro podporu komunikace | Zajišťuje obslužné operace pro podporu komunikace.                                                                                                                                                                                                                      |
| Šifrovací                          | Šifrování                    | Databázový server ho používá při volání šifrovacích a dešifrovacích funkcí.                                                                                                                                                                                             |
| <i>Název třídy</i>                 | Uživatелеm definovaný        | Spouští uživatelské rutiny způsobem, který je bezpečný při použití více jednotkových procesů, takže když se rutina nezdaří, databázový server tím nebude ovlivněn. Je určen konfiguračním parametrem VPCLASS. Je třeba zadat <i>název třídy</i> .                       |
| Java VP (JVP)                      | Rutiny UDR jazyka Java       | Provádí rutiny UDR jazyka Java. Obsahuje software Java Virtual Machine (JVM).                                                                                                                                                                                           |

Obrázek 5-2 znázorňuje hlavní komponenty a rozšiřitelnost databázového serveru.



Obrázek 5-2. Databázový server

## Výhody virtuálních procesorů

Ve srovnání s procesy databázových serverů, které obsluhují jedinou klientskou aplikaci, poskytují dynamické virtuální procesory databázového serveru s podporou vícenásobných jednotkových procesů následující výhody:

- Virtuální procesory mohou sdílet zpracování.
- Virtuální procesory šetří paměť a zdroje.
- Virtuální procesory mohou provádět paralelní zpracování.
- Můžete spouštět další virtuální procesory a ukončovat aktivní virtuální procesory CPU, zatímco je databázový server spuštěný.
- Virtuální procesory můžete navázat na konkrétní jednotky CPU.

Tyto a další výhody jsou popsány v následujícím textu.

## Sdílení zpracování

Virtuální procesory jedné třídy mají identický kód a sdílejí přístup k datům i ke frontám zpracování v paměti. Kterýkoli z virtuálních procesorů určité třídy může spustit jakýkoli jednotkový proces, který náleží k této třídě.

Databázový server se obvykle snaží udržet jednotkový proces v jediném virtuálním procesoru, protože jeho přesun na jiný virtuální procesor může vyžadovat přenos některých dat z paměti procesoru na sběrnici. Pokud ale jednotkový proces čeká na přidělení procesoru, může databázový server přesunout jednotkový proces na jiný virtuální procesor, protože přínos vyrovnání pracovní zátěže vyváží režii spojenou s přenosem dat.

Sdílené zpracování v rámci jedné třídy virtuálních procesorů probíhá automaticky a je pro uživatele databáze transparentní.

## Úspory paměti a zdrojů

Databázový server je schopen obsluhovat velký počet klientů pomocí malého počtu serverových procesů v porovnání s architekturami, které pro každého klienta spouštějí samostatný serverový proces. Dosahuje toho tak, že pro každého klienta spouští jednotkový proces namísto běžného procesu operačního systému.

Provoz s vícenásobnými jednotkovými procesy umožňuje efektivnější využití zdrojů operačního systému, protože jednotkové procesy sdílejí zdroje přidělené virtuálnímu procesoru. Veškeré jednotkové procesy spouštěné virtuálním procesorem mají stejný přístup k paměti virtuálního procesoru, komunikačním portům a souborům. Virtuální procesor koordinuje přístup jednotkových procesů ke zdrojům. Naproti tomu individuální procesy mají každý odlišnou sadu zdrojů a v případě, že několik procesů žádá přístup ke stejným zdrojům, musí jejich přístup koordinovat operační systém.

Virtuální procesor obvykle dokáže přepínat mezi jednotlivými jednotkovými procesy rychleji, než operační systém přepíná mezi jednotlivými systémovými procesy. Když operační systém přepíná mezi procesy, musí nejprve zastavit proces právě zpracovávaný procesorem, uložit jeho aktuální stav zpracování (neboli kontext) a spustit jiný proces. Zpracování obou procesů musí vstoupit do jádra operačního systému a opět ho opustit a také může být zapotřebí nahradit obsah části fyzické paměti. Naproti tomu jednotkové procesy sdílejí stejnou virtuální paměť a souborové deskriptory. Když virtuální procesor přepíná mezi jednotkovými procesy, dochází pouze k přepnutí z jednoho toku instrukcí na druhý. Virtuální procesor, který je vlastně procesem, dále používá přidělenou jednotku CPU bez přerušení. Popis způsobu, jakým virtuální procesor přepíná mezi jednotlivými jednotkovými procesy naleznete v části "Přepínání kontextu" na stránce 5-8.

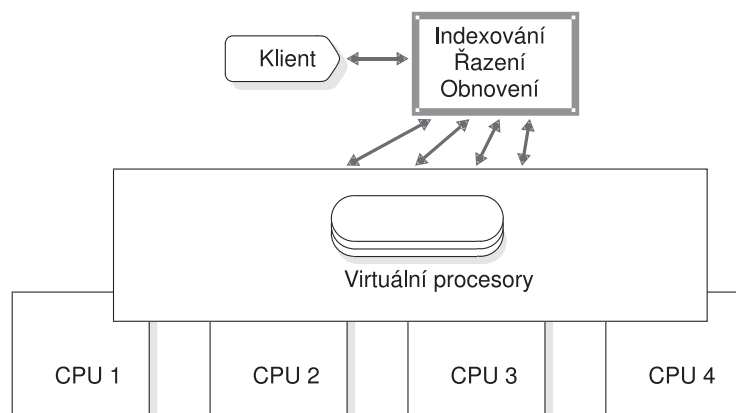
## Paralelní zpracování

V následujících případech mohou virtuální procesory třídy CPU spouštět několik jednotkových procesů relací pro jednoho klienta současně:

- vytváření indexu
- řazení
- obnova
- prohledávání
- spojování
- Při agregaci.
- Při seskupování.
- Při provádění uživatelské rutiny (UDR)

Další informace o paralelním provádění rutin UDR naleznete v příručce *IBM Informix User-Defined Routines and Data Types Developer's Guide*.

Obrázek 5-3 znázorňuje paralelní zpracování. Pokud klient zahájí indexování, řazení nebo logická obnova, databázový server spustí několik jednotkových procesů, které zpracovávají úlohu současně a s maximálním možným využitím zdrojů počítače. Zatímco jeden jednotkový proces čeká na vstup - výstup, jiný může pracovat.



Obrázek 5-3. Paralelní zpracování

### Přidávání a vypouštění virtuálních procesorů v režimu online

Virtuální procesory můžete přidávat v závislosti na zvyšujících se požadavcích na obsluhu, zatímco je databázový server spuštěn. Pokud například dojde k tomu, že virtuální procesory určité třídy budou omezeny výkonem jednotky CPU nebo výkonem vstupu - výstupu (to znamená, že požadavky na jednotku CPU nebo na vstup - výstup se budou hromadit rychleji, než jak je aktuální počet virtuálních procesorů dokáže zpracovávat), můžete spustit další virtuální procesory, aby se zlepšila distribuce pracovního zatížení.

Zatímco je databázový server spuštěný, můžete přidávat virtuální procesory jakékoli třídy. Další informace naleznete v části “Přidání virtuálních procesorů v režimu online” na stránce 6-3.

#### Jen pro Windows

V systému Windows nelze vypustit virtuální procesor žádné třídy.

#### Konec Jen pro Windows

Zatímco je databázový server spuštěný, můžete vypouštět virtuální procesory třídy CPU nebo uživatelem definované třídy. Další informace naleznete v části “Nastavení konfiguračních parametrů virtuálních procesorů” na stránce 6-1.

### Vázání virtuálních procesorů k jednotce CPU

Některé víceprocesorové systémy umožňují navázat proces k určité jednotce CPU. Tato vlastnost se nazývá *afinita k procesoru*.

Na víceprocesorových počítačích, ve kterých databázový server podporuje procesorovou afinitu, lze navázat virtuální procesory CPU ke konkrétním jednotkám CPU v počítači. Pokud je virtuální procesor třídy CPU navázán k jednotce CPU, bude virtuální procesor spuštěn výhradně v této jednotce CPU. Tato operace zvýší výkon virtuálního procesoru, protože sníží počet přepnutí mezi procesy, které operační systém musí provádět. Navázání virtuálních procesorů CPU ke konkrétním jednotkám CPU umožňuje rovněž izolovat práci databáze na určitých procesorech počítače a ponechat zbývající procesory volné pro jiné operace. Pouze virtuální procesory CPU lze navázat ke konkrétní jednotce CPU.

Další informace o tom, jak přiřadit virtuální procesory CPU k fyzickým procesorům naleznete v části “Užití procesorové afinity” na stránce 5-14.

---

## Obsluha jednotkových procesů virtuálními procesory

V určitý okamžik může virtuální procesor provádět jen jeden jednotkový proces. Virtuální procesor obsluhuje několik jednotkových procesů souběžně tím způsobem, že mezi nimi přepíná. Virtuální procesor provádí jednotkový proces, dokud jednotkový proces nepředá řízení. Jakmile jednotkový proces předá řízení, přepne virtuální procesor na následující jednotkový proces, který je připraven ke spuštění. Virtuální procesor pokračuje v tomto procesu, až se nakonec vrátí k původně zpracovávanému jednotkovému procesu, jakmile je tento připraven pokračovat. Některé jednotkové procesy dokončí svou práci a virtuální procesor spustí nové jednotkové procesy, aby zpracovaly nové úlohy. Protože virtuální procesor neustále přepíná mezi jednotkovými procesy, umožňuje jednotce CPU neustále pracovat. Rychlost zpracování vytváří dojem, že virtuální procesor zpracovává několik úloh současně, což ve výsledku skutečně činí.

Spuštění několika souběžných jednotkových procesů vyžaduje plánování a synchronizaci, aby nedocházelo ke vzájemnému ovlivňování práce jednotlivých jednotkových procesů. Virtuální procesory používají ke koordinování souběžného zpracování několika jednotkovými procesy následující struktury a metody:

- řídicí struktury
- přepínání kontextu
- zásobníky
- fronty
- objekty mutex

Tato část popisuje, jak virtuální procesory využívají těchto struktur a metod.

### Řídicí struktury

Když se klient připojí k databázovému serveru, databázový server vytvoří strukturu *relace*, která se nazývá *řídicí blok relace* a která obsahuje informace o připojení a o uživateli. Relace začíná, když se klient připojí k databázovému serveru a končí s ukončením připojení.

Databázový server také vytvoří strukturu jednotkového procesu, která se nazývá *řídicí blok jednotkového procesu* (TCB) relace a spustí primární jednotkový proces (**sqlxec**), který bude zpracovávat požadavky klienta. Když jednotkový proces *předává řízení*, tedy když se pozastaví a umožní spuštění jiného jednotkového procesu, uloží virtuální procesor informace o stavu jednotkového procesu do řídicího bloku jednotkového procesu. Tato informace zahrnuje obsah systémových registrů procesu, čítač instrukcí (adresu následující instrukce k provedení) a ukazatel zásobníku. Tyto informace tvoří *kontext* jednotkového procesu.

Ve většině případů spouští databázový server jeden primární jednotkový proces pro každou relaci. V případech, kdy provádí paralelní zpracování, ovšem vytváří několik jednotkových procesů relace pro jediného klienta a také vytváří několik odpovídajících řídicích bloků jednotkových procesů.

### Přepínání kontextu

Virtuální procesor přepíná z jednoho jednotkového procesu na druhý pomocí *přepínání kontextu*. Databázový server nepřerušuje probíhající jednotkový proces nuceně, jak to činí operační systém s procesem po uplynutí pevně stanoveného časového intervalu (časového kvanta). Jednotkový proces místo toho přepne řízení v jednom z následujících okamžiků:

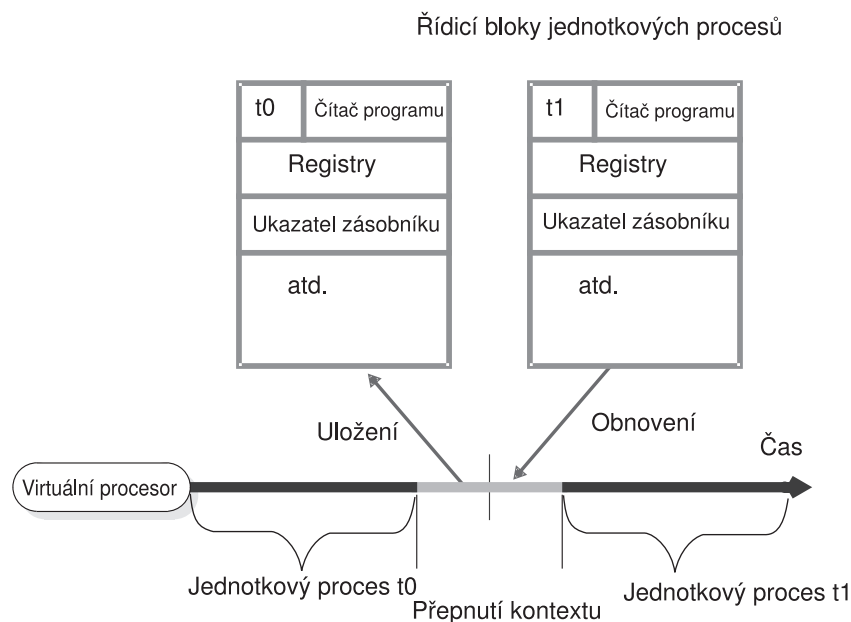


- V předem určeném bodu v kódu.
- Pokud jednotkový proces nelze nadále provádět, dokud nebude splněna určitá podmínka.

Pokud by objem zpracování potřebný k provedení úlohy způsobil čekání jiných jednotkových procesů po příliš dlouhou dobu, předá jednotkový proces v předem stanoveném okamžiku řízení. Kód pro takto dlouhodobé úkoly obsahuje volání funkce, pomocí které jednotkový proces ve strategických okamžicích zpracování předává řízení. Když jednotkový proces provádí jednu z těchto úloh, předá řízení, jakmile dojde k volání funkce předání řízení. Poté mohou být spuštěny ostatní jednotkové procesy čekající ve frontě připravených procesů. Jakmile je znovu na řadě původní jednotkový proces, začne tento proces opět provádět svůj kód od bodu, který následoval za voláním funkce předání řízení. Předem stanovená volání funkce předání řízení umožňují databázovému serveru přerušit jednotkové procesy v bodech, které jsou z hlediska výkonu nejvýhodnější.

Jednotkový proces předá řízení i tehdy, pokud nebude moci pokračovat ve zpracování úlohy, dokud nebude splněna určitá podmínka. Jednotkový proces například předá řízení, pokud čeká na diskový vstup - výstup, aby mohl dokončit svou úlohu, když čeká na příjem dat od klienta nebo když čeká na zámek či jiný zdroj.

Když jednotkový proces předává řízení, virtuální procesor uloží jeho kontext do řídicího bloku jednotkového procesu. Poté virtuální procesor vybere z fronty připravených jednotkových procesů nový jednotkový proces k provedení, zavede kontext nového jednotkového procesu z jeho řídicího bloku a začne provádět instrukce na nové adrese v čítači programu. Obrázek Obrázek 5-4 znázorňuje, jak virtuální procesor provede přepnutí kontextu.



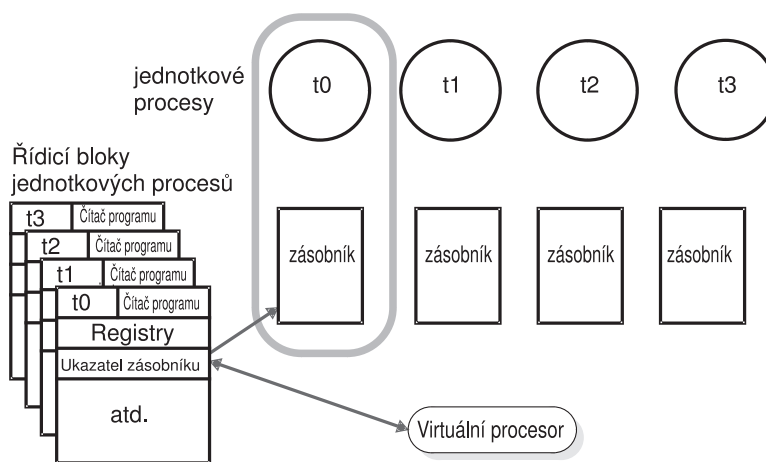
Obrázek 5-4. Přepnutí kontextu: Jak virtuální procesor přepíná z jednoho jednotkového procesu na jiný.

## Zásobníky

Databázový server přidělí ve virtuální části sdílené paměti oblast k uložení nesdílených dat pro funkce prováděné jednotkovým procesem. Tato oblast se nazývá *zásobník*. Informace o tom, jak nastavit velikost zásobníku, naleznete v části "Zásobníky" na stránce 7-18.

Zásobník umožňuje virtuálnímu procesoru chránit nesdílená data jednotkového procesu před přepsáním jinými jednotkovými procesy, které současně provádějí tentýž kód. Pokud například několik klientských aplikací současně provádí příkazy SELECT, jednotkové procesy relací každého klienta provádějí v kódu řadu stejných funkcí. Pokud by jednotkový proces neměl soukromý zásobník, mohl by jeden jednotkový proces přepsat místní data, která patří jinému jednotkovému procesu v rámci stejné funkce.

Když virtuální procesor přepíná na nový jednotkový proces, zavede z pole v řídicím bloku jednotkového procesu ukazatel zásobníku pro tento jednotkový proces. V ukazateli zásobníku je uložena počáteční adresa zásobníku. Virtuální procesor pak může přistupovat k datům v zásobníku pomocí posunů vůči této počáteční adrese. Obrázek Obrázek 5-5 znázorňuje, jak virtuální procesor využívá zásobník k oddělení nesdílených dat jednotkových procesů relací.



Obrázek 5-5. Virtuální procesory oddělují nesdílená data každého uživatele.

## Fronty

Databázový server využívá tři typy front k plánování zpracování mnoha souběžně prováděných jednotkových procesů:

- fronty připravených procesů
- fronty spících procesů
- fronty čekajících procesů

Virtuální procesory stejné třídy sdílejí fronty. Tato skutečnost zčásti umožňuje jednotkovým procesům migrovat z jednoho virtuálního procesoru na jiný virtuální procesor stejné třídy, je-li to nutné.

### Fronty připravených procesů

Fronty připravených procesů obsahují jednotkové procesy, které jsou připravené ke spuštění, jakmile aktuální (prováděný) jednotkový proces předá řízení. Jakmile jednotkový proces předá řízení, virtuální procesor vybere další jednotkový proces z fronty připravených procesů s odpovídající prioritou. V rámci fronty zpracovává virtuální procesor jednotkové procesy téže priority v pořadí FIFO (first-in-first-out).

Pokud ve víceprocesorovém počítači zpozorujete, že se ve frontě připravených procesů pro konkrétní třídu virtuálních procesorů hromadí jednotkové procesy (což naznačuje, že se práce hromadí rychleji, než ji dokáže virtuální procesor zpracovávat), můžete spustit další virtuální procesory této třídy a rozdělit tak zátěž při zpracování. Další informace o monitorování front připravených procesů naleznete v části “Monitorování virtuálních procesorů” na stránce 6-5.

Další informace o přidávání virtuálních procesorů, zatímco je databázový server v režimu online naleznete v části “Přidání virtuálních procesorů v režimu online” na stránce 6-3.

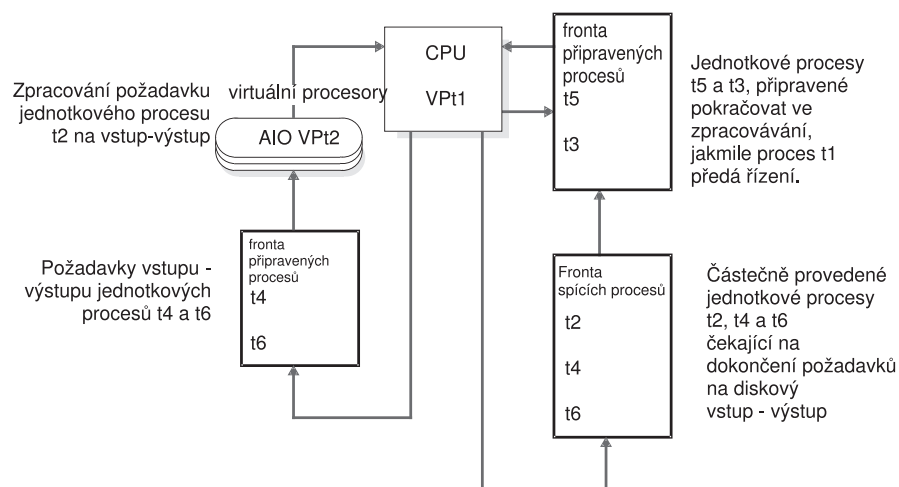
## Fronty spících procesů

Fronty spících procesů obsahují kontexty jednotkových procesů, které v daném čase nemají žádné úlohy. Jednotkový proces je uveden do režimu spánku buď po určitý časový interval, nebo *navždy*.

Administrační třída (ADM) virtuálních procesorů spouští systémový časovač a speciální pomocné jednotkové procesy. Virtuální procesory v této třídě jsou vytvářeny a spouštěny automaticky. Tuto třídu virtuálních procesorů neovlivňuje žádný z konfiguračních parametrů.

Virtuální procesor třídy ADM probouzí jednotkové procesy, které byly usnány na určitou dobu. Jednotkový proces, který je spuštěn ve virtuálním procesoru ADM kontroluje spící jednotkové procesy v sekundových intervalech. Pokud již vypršela doba, po kterou byl jednotkový proces usnán, virtuální procesor ADM ho přesune do příslušné fronty připravených procesů. Jednotkový proces, který spí po určitou dobu, může být rovněž explicitně probuzen jiným jednotkovým procesem.

Jednotkový proces, který byl usnán *navždy*, je probuzen ve chvíli, kdy má provést další úlohu. Když například jednotkový proces spuštěný ve virtuálním procesoru CPU potřebuje přístup k disku, vydá požadavek vstupu - výstupu, umístí se do fronty spících procesů virtuálního procesoru CPU a předá řízení. Jakmile jednotkový proces vstupu - výstupu vyrozumí virtuální procesor CPU, že byla operace vstupu - výstupu dokončena, naplánuje virtuální procesor CPU další zpracování původního jednotkového procesu tím, že ho přesune z fronty spících procesů do fronty připravených procesů. Obrázek Obrázek 5-6 znázorňuje řazení jednotkových procesů databázového serveru do fronty k provádění operací vstupu - výstupu databáze.



Obrázek 5-6. Řazení jednotkových procesů databázového serveru do fronty k provedení databázového vstupu - výstupu.

## Fronty čekajících procesů

Fronty čekajících procesů obsahují jednotkové procesy, které musí čekat na určitou událost, než budou moci pokračovat. Fronty čekajících procesů koordinují například přístup jednotkových procesů ke sdíleným datům. Pokud se uživatel pokouší získat zámek logického protokolu a zjistí, že je zámek využíván jiným uživatelem, umístí se jednotkový proces, kterému byl zamítnut přístup, do fronty čekajících procesů logického protokolu. Pokud je

jednotkový proces držící zámek připraven zámek uvolnit, zkontroluje případné čekající jednotkové procesy a nalezne-li je, probudí následující jednotkový proces ve frontě čekajících procesů.

## Objekty mutex

Objekt typu mutex (z anglického *mutually* (vzájemně) *exclusive* (výlučný)), označovaný též jako *zámek* (*latch*), je uzamykací mechanismus využívaný databázovým serverem k synchronizaci přístupu více jednotkových procesů ke sdíleným zdrojům. Objekty mutex lze přirovnat k semaforům používaným některými operačními systémy k řízení přístupu více *procesů* ke sdíleným datům. Objekty mutex ovšem umožňují vyšší míru souběžného přístupu než semaforey.

Objekt mutex je proměnná, která je přidružená ke sdílenému zdroji, jakým je například vyrovnávací paměť. Jednotkový proces musí před přístupem k samotnému zdroji získat objekt mutex pro tento zdroj. Ostatním jednotkovým procesům je znemožněn přístup ke zdroji, dokud ho jeho vlastník neuvolní. Jednotkový proces získá objekt mutex poté, co se objekt mutex stane dostupným nastavením do stavu "používá se". Synchronizace, kterou objekty mutex poskytují je zárukou, že do konkrétní oblasti sdílené paměti zapisuje vždy pouze jeden jednotkový proces.

Další informace o monitorování objektů mutex naleznete v části "Monitorování profilu sdílené paměti a zámků latch" na stránce 8-8.

---

## Třídy virtuálních procesorů

Virtuální procesor určité třídy může spouštět pouze jednotkové procesy této třídy. Tato část popisuje typy jednotkových procesů (typy zpracování) prováděné jednotlivými třídami virtuálních procesorů. Vysvětluje také, jak určit počet virtuálních procesorů, který je zapotřebí spouštět v každé třídě.

## Virtuální procesory CPU

Ve virtuálních procesorech CPU jsou spouštěny všechny jednotkové procesy relací (tedy jednotkové procesy, které zpracovávají požadavky klientských aplikací používajících jazyk SQL a některé interní jednotkové procesy. Interní jednotkové procesy zajišťují interní služby databázového serveru. Například jednotkový proces, který od klientských aplikací přijímá požadavky na připojení, je interní jednotkový proces.

Každý virtuální procesor CPU může mít přidruženou vlastní vyrovnávací paměť. Každý blok vlastní vyrovnávací paměti obsahuje 1 až 32 stránek paměti; každá stránka paměti je velká 4096 bajtů. Databázový server používá vlastní vyrovnávací paměť, aby zkrátil přístupovou dobu k paměťovým blokům. Chcete-li povolit použití vlastní vyrovnávací paměti a zadat o vyrovnávací paměti informace, použijte konfigurační parametr `VP_MEMORY_CACHE_KB`. Další informace naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference* a v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

## Stanovení počtu potřebných virtuálních procesorů CPU

Správný počet virtuálních procesorů CPU je takový, při kterém jsou všechny virtuální procesory zaneprázdněné, ne však natolik, aby nezvládaly přijímat příchozí požadavky. Neměli byste přidělit více virtuálních procesorů CPU, než je počet fyzických (hardwarových) procesorů v počítači.

Pokud chcete posoudit výkon virtuálních procesorů CPU za provozu databázového serveru, opakujte v pravidelných intervalech po určitou dobu následující příkaz:

```
onstat -g glo
```

Pokud součet časových intervalů *usercpu* a *syscpu* dosáhne 100 procent času uplynulého v jednom testovacím intervalu, přidejte další virtuální procesor třídy CPU, máte-li k dispozici jednotku CPU, na které by ho bylo možné spustit.

Další informace týkající se rozhodnutí o tom, kolik virtuálních procesorů CPU potřebujete, naleznete v části “Spouštění jednotkových procesů cyklického dotazování na virtuálních procesorech CPU nebo síťových virtuálních procesorech” na stránce 5-23.

Parametr VPCLASS umožňuje určit následující údaje:

- Počet virtuálních procesorů, které mají být primárně spuštěny pro tuto třídu.
- Maximální počet virtuálních procesorů spuštěných pro danou třídu.
- Procesorovou afinitu pro virtuální procesory třídy CPU.
- Zakázání stárnutí priority, je-li to vhodné.

**Důležité:** Namísto parametrů AFF\_SPROC, AFFNPROCS, NOAGE, NUMCPUVPS a NUMAIOVPS byste měli použít parametr VPCLASS. Parametr VPCLASS nelze používat v kombinaci s těmito parametry. Pokud soubor ONCONFIG například obsahuje parametr NUMCPUVPS, obdržíte chybovou zprávu, obsahuje-li soubor rovněž parametr VPCLASS *cpu*. Další informace o konfiguraci parametru VPCLASS naleznete v příručce *IBM Informix Administrator's Reference*.

Kromě toho, že posoudíte počet jednotek CPU v počítači a počet uživatelů, kteří se budou připojovat k databázovému serveru, zvažte též skutečnost, že uživatelské rutiny a moduly DataBlade (které jsou kolekcemi uživatelských rutin) jsou prováděny ve virtuálních procesorech CPU nebo v uživatelem definovaných virtuálních procesorech. Další informace o tom, jak k virtuálnímu procesoru přiřadit uživatelskou rutinu naleznete v části “Přiřazení UDR k uživatelem definované třídě virtuálních procesorů” na stránce 5-16.

## Spuštění na víceprocesorovém počítači

Spouštíte-li několik virtuálních procesorů CPU na počítači s více procesory, nastavte v souboru ONCONFIG parametr MULTIPROCESSOR na hodnotu 1. Když nastavíte hodnotu parametru MULTIPROCESSOR na 1, databázový server provede uzamčení způsobem, který je vhodný pro počítač s více procesory. Další informace o nastavení víceprocesorového režimu naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Spuštění na jednoprocessorovém počítači

Pokud databázový server spouštíte na jednoprocessorovém počítači, nastavte konfigurační parametr MULTIPROCESSOR na hodnotu 0. Chcete-li databázový server spustit pouze s jedním virtuálním procesorem CPU, nastavte parametr SINGLE\_CPU\_VP na hodnotu 1.

Nastavení parametru MULTIPROCESSOR na hodnotu 0 umožní databázovému serveru obcházet zamykání, které je nezbytné ke spuštění více procesů na víceprocesorovém počítači. Další informace o konfiguračním parametru MULTIPROCESSOR naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Nastavení parametru SINGLE\_CPU\_VP hodnotu 1 umožní databázovému serveru obejít některé z volání mutex, která obvykle provádí, spouští-li několik virtuálních procesorů CPU. Další informace o nastavení parametru SINGLE\_CPU\_VP naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

**Důležité:** Nastavení parametru VPCLASS *num* na hodnotu 1 a parametru SINGLE\_CPU\_VP na hodnotu 0 nesníží počet volání objektů mutex, a to i přesto, že databázový server spouští pouze jeden virtuální procesor CPU. Parametr SINGLE\_CPU\_VP

musíte nastavit na hodnotu 1, chcete-li snížit četnost zamykání, ke kterému dochází, spustíte-li jediný virtuální procesor CPU.

Nastavení parametru SINGLE\_CPU\_VP na hodnotu 1 zavede v databázovém serveru následující dvě důležitá omezení:

- Povolen je pouze jeden virtuální procesor CPU.  
Virtuální procesory CPU nelze přidávat, zatímco je databázový server v režimu online.
- Nejsou povoleny uživatelem definované třídy. (Uživatelé však mohou definovat rutiny, které jsou spouštěny přímo ve virtuálním procesoru CPU VP.)

Další informace naleznete v části “Přidání virtuálních procesorů v režimu online” na stránce 6-3.

## **Přidávání a vypouštění virtuálních procesorů CPU v režimu online**

Virtuální procesory třídy CPU lze přidávat a vypouštět, i když je databázový server online. Pokyny, jak to provést, naleznete v částech “Přidání virtuálních procesorů v režimu online” na stránce 6-3 a “Vypouštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů” na stránce 6-4.

## **Jak zabránit stárnutí priority**

Některé operační systémy postupně snižují prioritu dlouhodobě probíhajících procesů s narůstající dobou jejich zpracování. Tato funkce operačního systému se nazývá *stárnutí priority*. Stárnutí priority může způsobit klesání výkonu procesů databázového serveru s přibývajícím časem. Přesto operační systém v některých případech umožňuje tuto vlastnost vypnout a zachovat vysokou prioritu dlouhodobě spuštěných procesů.

Chcete-li zjistit, zda je stárnutí priority k dispozici ve vašem počítači, naleznete informace v souboru Poznámky k počítači, který je součástí instalační sady a je popsán v úvodu k této příručce.

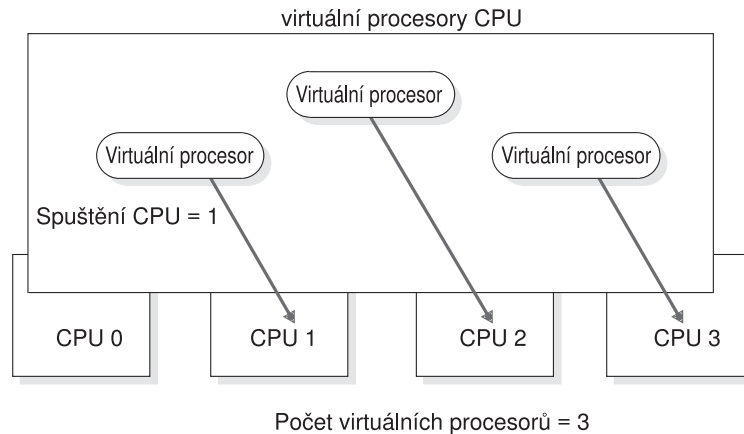
Pokud operační systém umožňuje vypnout stárnutí priority, můžete ho vypnout stanovením hodnoty `noage` pro položku priority v konfiguračním parametru VPCLASS. Další informace naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## **Užití procesorové afinity**

Databázový server podporuje automatické navázání virtuálních procesorů CPU k procesorům na víceprocesorových počítačích, které podporují *procesorovou afinitu*. Instalační médium databázového serveru obsahuje soubor Poznámky k počítači s informací, zda vaše verze databázového serveru podporuje tuto vlastnost. Když přiřadíte virtuální procesor CPU k příslušné jednotce CPU, virtuální procesor bude spuštěn jen na této jednotce CPU, avšak na této jednotce CPU mohou být spuštěny i jiné jednotkové procesy.

Použijte konfigurační parametr VPCLASS s volbou *aff* k implementování procesorové afinity na víceprocesorových počítačích, které to podporují.

Obrázek 5-7 ilustruje koncept procesorové afinity.



Obrázek 5-7. Procesorová afinita

### Jen pro UNIX

Informace o tom, zda je procesorová afinita podporovaná na vaší platformě UNIX, naleznete v souboru Poznámky k počítači.

### Konec Jen pro UNIX

**Nastavení procesorové afinity pomocí parametru VPCLASS:** Pro nastavení procesorové afinity pomocí parametru VPCLASS, určete rozsah CPU, ke kterému přiřadíte virtuální procesory. Databázový server přiřadí virtuální procesory CPU k jednotkám CPU metodou cyklické obsluhy, počínaje prvním číslem procesoru určeným parametrem *aff*. Určíte-li více virtuálních procesorů CPU než je k dispozici, databázový server začne znovu od začátku. V následujícím příkladě položka afinity přiřadí virtuální procesor k CPU 1, 2 a 3. Jsou dostupné tři virtuální procesory (určené položkou *num*):

```
VPCLASS CPU,num=3,aff=1-3
```

V následujícím příkladu přiřadí databázový server dva virtuální procesory k jednotce CPU 5 a 6 a jeden virtuální procesor k jednotce CPU 7:

```
VPCLASS CPU,num=3,aff=5-7
```

Další informace naleznete pod heslem VPCLASS v kapitole o konfiguračních parametrech v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Uživatелеm definované třídy virtuálních procesorů

Můžete definovat speciální třídy virtuálních procesorů ke spuštění uživatelských rutin nebo modulu DataBlade. Uživatelské rutiny bývají obvykle napsány tak, aby podporovaly uživatelem definované datové typy. Nechcete-li, aby byla uživatelská rutina spuštěna v třídě CPU, což je výchozí nastavení, můžete ji přiřadit k uživatelem definované třídě virtuálních procesorů (VP). Uživatелеm definované třídy virtuálních procesorů se též nazývají *rozšiřující virtuální procesory*.

Tato část obsahuje následující informace o uživatelem definovaných virtuálních procesorech:

- Kdy spouštět rutiny UDR psané v jazyku C v uživatelem definovaném VP místo ve virtuálním procesoru CPU VP.
- Jak přiřadit rutiny UDR psané v jazyku C ke konkrétní, uživatelem definované třídě VP.
- Jak přidávat a vypouštět uživatelem definované virtuální procesory VP, zatímco je databázový server v režimu online.

## Stanovení potřebného počtu uživatelem definovaných virtuálních procesorů

Můžete určit tolik uživatelem definovaných virtuálních procesorů, kolik umožňuje použitý operační systém. Pokud spouštíte mnoho rutin UDR nebo paralelních dotazů PDQ používajících rutiny UDR, měli byste nakonfigurovat více uživatelem definovaných virtuálních procesorů.

### Užití uživatelem definovaných virtuálních procesorů

Uživatelem definované třídy virtuálních procesorů chrání databázový server před *nesprávně pracujícími* uživatelskými rutinami. Nesprávně pracující uživatelská rutina se vyznačuje alespoň jednou z následujících charakteristik:

- Nepředává řízení jiným jednotkovým procesům.
- Provádí blokující volání operačního systému.
- Mění globální stav procesoru VP.

Správně pracující rutina UDR psaná v jazyku C nemá žádnou z těchto vlastností. Ve virtuálním procesoru CPU spouštějte pouze správně pracující rutiny UDR v jazyku C.

**Upozornění:** Spuštění nesprávně pracující rutiny ve virtuálním procesoru CPU může způsobit vážné narušení práce databázového serveru a s nejvyšší pravděpodobností může způsobit jeho selhání nebo chybnou funkci. Kromě toho ani rutina samotná nemusí vyprodukovat správné výsledky.

Abyste zajistili správné provádění, přiřaďte veškeré nesprávně pracující uživatelské rutiny k uživatelem definované třídě virtuálních procesorů. Uživatelem definované virtuální procesory VP odstraní následující omezení programátora, která se týkají třídy CPU VP:

- Nutnost pravidelného předávání řízení procesoru.
- Potřebu eliminování blokujících volání vstupu - výstupu.

Funkce, které jsou spouštěny v uživatelem definované třídě virtuálních procesorů, nemusejí předávat řízení a mohou provádět přímá volání systému souborů, která blokují další zpracování virtuálním procesorem, dokud není vstup - výstup dokončen.

Normální zpracování uživatelských dotazů není nesprávně pracujícími rutinami UDR v jazyku C ovlivněno, protože tyto rutiny UDR nejsou spouštěny ve virtuálních procesorech CPU. Podrobnější diskusi na téma nesprávně pracujících rutin naleznete v příručce *IBM Informix DataBlade API Programmer's Guide*.

### Určení uživatelem definovaného virtuálního procesoru

Parametr VPCLASS s volbou *vpclass* definuje uživatelem definovanou třídu virtuálních procesorů. Můžete též určit uživatelem definovaný virtuální procesor, který nepředává řízení. Další informace naleznete v části "Určení parametru VPCLASS" na stránce 6-2 a v kapitole věnované konfiguračním parametrům v příručce *IBM Informix Dynamic Server Administrator's Reference*.

### Přiřazení UDR k uživatelem definované třídě virtuálních procesorů

Příkaz SQL CREATE FUNCTION slouží k registrování uživatelské rutiny. Například následující příkaz CREATE FUNCTION zaregistruje uživatelskou rutinu **GreaterThanEqual()** a určuje, že volání směřovaná na tuto rutinu mají být provedena uživatelem definovanou třídou VP nazvanou UDR:



```
CREATE FUNCTION GreaterThanEqual(ScottishName, ScottishName)
 RETURNS boolean
 WITH (CLASS = UDR)
 EXTERNAL NAME '/usr/lib/objects/udrs.so'
 LANGUAGE C
```

K provedení této funkce musí soubor ONCONFIG obsahovat parametr VPCLASS, který definuje třídu UDR. Není-li tomu tak, volání funkce **GreaterThanEqual** se nezdaří.

**Rada:** Modifikátor rutiny CLASS může určit jakýkoliv název třídy VP. Tento název třídy nemusí existovat již při registraci rutiny UDR. Pokud však chcete spustit rutinu UDR, která ke svému provedení určuje uživatelem definovanou třídu VP, musí tato třída existovat a mít k sobě přiřazené virtuální procesory.

Pro konfiguraci třídy UDR přidejte do souboru ONCONFIG řádek podobný následujícímu. Tento řádek konfiguruje třídu UDR se dvěma virtuálními procesory a bez funkce stárnutí priority.

```
VPCLASS UDR ,num=2,noage
```

Předchozí řádek definuje třídu virtuálního procesoru UDR VP jako třídu VP předávající řízení; jinými slovy, tato třída VP umožňuje UDR v jazyku C přepínat řízení na jiné jednotkové procesy, které potřebují přístup do třídy virtuálních procesorů UDR. Další informace o tom, jak používat konfigurační parametr VPCLASS naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Další informace o příkazu CREATE FUNCTION naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

## Přidávání a vypouštění uživatelem definovaných virtuálních procesorů v režimu online

Virtuální procesory uživatelem definované třídy lze přidávat a vypouštět, zatímco je databázový server online. Pokyny, jak to provést, naleznete v částech “Přidání virtuálních procesorů v režimu online” na stránce 6-3 a “Vypouštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů” na stránce 6-4.

## Virtuální procesory Java

Uživatelské rutiny Java UDR a aplikace Java jsou prováděny na specializovaných virtuálních procesorech nazývaných *virtuální procesory Java* (JVP). Virtuální procesor JVP obsahuje ve svém kódu software JVM (Java virtual machine). JVP má stejné možnosti jako virtuální procesor CPU v tom smyslu, že dokáže zpracovávat úplné dotazy jazyka SQL.

Můžete specifikovat tolik procesorů JVP, kolik umožňuje použitý operační systém. Spouštíte-li mnoho uživatelských rutin UDR v jazyku Java nebo paralelních dotazů PDQ používajících rutiny UDR v jazyku Java, měli byste nakonfigurovat více procesorů JVP. Další informace o rutinách UDR napsaných v jazyku Java naleznete v příručce *J/Foundation Developer's Guide*.

Ke konfiguraci procesoru JVP použijte konfigurační parametr VPCLASS s klíčovým slovem `jvp`. Další informace naleznete v kapitole věnované konfiguračním parametrům v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Virtuální procesory diskového vstupu - výstupu

Následující třídy virtuálních procesorů provádějí diskový vstup - výstup:

- PIO (vstup - výstup fyzického protokolu)
- LIO (vstup - výstup logického protokolu)

- AIO (asynchronní vstup - výstup)
- CPU (asynchronní vstup - výstup jádra)

Třída PIO zajišťuje veškerý vstup - výstup souborů fyzického protokolu a třída LIO zajišťuje veškerý vstup - výstup souborů logického protokolu, *vyjma případů*, kdy jsou tyto soubory uloženy v diskovém prostoru s přímým přístupem a databázový server implementuje vstup - výstup KAIO.

V operačních systémech, které nepodporují vstup - výstup KAIO, použijte databázový server třídu virtuálních procesorů AIO k zajištění databázového vstupu - výstupu, který nevyužívá fyzické nebo logické protokolování.

Databázový server využívá třídu CPU k zajištění vstupu - výstupu KAIO, je-li tento na dané platformě dostupný. Pokud databázový server implementuje vstup - výstup KAIO, obstarává veškerý vstup - výstup diskového prostoru s přímým přístupem jednotkový proces KAIO, včetně vstupu - výstupu fyzických a logických protokolů.

#### Jen pro UNIX

Informace o tom, zda vaše platforma UNIX podporuje vstup - výstup KAIO, naleznete v souboru Poznámky k počítači.

**Konec Jen pro UNIX**

#### Jen pro Windows

Operační systém Windows vstup - výstup KAIO podporuje.

**Konec Jen pro Windows**

Další informace o neprotokolujícím vstupu - výstupu naleznete v části “Asynchronní vstup - výstup” na stránce 5-19.

### Priority vstupu - výstupu

Databázový server obecně stanovuje priority diskového vstupu - výstupu přiřazením různých typů vstupu - výstupu různým třídám virtuálních procesorů a přiřazením priorit frontám neprotokolujícího vstupu - výstupu. Stanovení priorit například zajišťuje, aby operace vstupu - výstupu protokolu s vysokou prioritou nemohla být zařazena do fronty za zápis do dočasného souboru, jehož priorita je nízká. Databázový server rozděluje podle priorit různé typy diskového vstupu - výstupu, jak znázorňuje Tabulka 5-2.

*Tabulka 5-2. Jak databázový server stanovuje priority diskového vstupu - výstupu*

| Priorita | Typy vstupu - výstupu              | Třída VP     |
|----------|------------------------------------|--------------|
| 1.       | Vstup - výstup logického protokolu | CPU nebo LIO |
| 2.       | Vstup - výstup fyzického protokolu | CPU nebo PIO |
| 3.       | Vstup - výstup databáze            | CPU nebo AIO |
| 3.       | Vstup - výstup čištění stránek     | CPU nebo AIO |
| 3.       | Vstup - výstup čtení napřed        | CPU nebo AIO |

### Vstup - výstup logického protokolu

Třída virtuálních procesorů LIO zajišťuje vstup - výstup souborů logického protokolu v následujících případech:

- Vstup - výstup KAIO není implementován.

- Soubory logického protokolu se nacházejí v předpřipraveném diskovém prostoru.

Jen v případě, že je implementován vstup - výstup KAIO a soubory logického protokolu se nacházejí v diskovém prostoru s přímým přístupem, využívá databázový server jednotkový proces KAIO ve virtuálním procesoru CPU k zajištění vstupu - výstupu logického protokolu.

Soubory logického protokolu ukládají data, která umožňují databázovému serveru odvolávat transakce a obnovit data po selhání systému. Vstup - výstup souborů logického protokolu je diskový vstup - výstup s nejvyšší prioritou, který databázový server provádí.

Nacházejí-li se soubory logického protokolu v prostoru dbspace, který *není zrcadlen*, spustí databázový server pouze jeden virtuální procesor LIO. Nacházejí-li se soubory logického protokolu v prostoru dbspace, který *je zrcadlen*, spustí databázový server dva virtuální procesory LIO. Tato třída virtuálních procesorů nemá žádné přidružené parametry.

### **Vstup - výstup fyzického protokolu**

Třída virtuálních procesorů PIO zajišťuje vstup - výstup souboru fyzického protokolu v následujících případech:

- Vstup - výstup KAIO není implementován.
- Soubor fyzického protokolu je uložen v blocích souborů s použitím vyrovnávací paměti.

Jen v případě, kdy je implementován vstup - výstup KAIO a soubor fyzického protokolu se nachází v diskovém prostoru s přímým přístupem, použije databázový server jednotkový proces KAIO ve virtuálním procesoru CPU k zajištění vstupu - výstupu fyzického protokolu. V souboru fyzického protokolu jsou uloženy *předobrazy* stránek dbspace, které se změnilly od posledního *kontrolního bodu*. (Další informace o kontrolních bodech naleznete v části “Kontrolní body” na stránce 15-5.) Na počátku obnovení, ještě před zpracováním transakcí z logického protokolu, použije databázový server fyzický protokol k obnovení předobrazů stránek prostoru dbspace, které se změnilly od posledního kontrolního bodu. Vstup - výstup souborů fyzického protokolu je vstup - výstup s druhou nejvyšší prioritou po vstupu - výstupu souborů logického protokolu.

Nachází-li se soubor fyzického protokolu v prostoru dbspace, který *není zrcadlen*, spustí databázový server pouze jeden virtuální procesor PIO. Nachází-li se soubor fyzického protokolu v prostoru dbspace, který *je zrcadlen*, spustí databázový server dva virtuální procesory PIO. Tato třída virtuálních procesorů nemá žádné přidružené parametry.

### **Asynchronní vstup - výstup**

Databázový server provádí databázový vstup - výstup asynchronně, to znamená, že operace vstupu - výstupu jsou zařazeny do fronty a provedeny nezávisle na jednotkovém procesoru, který o vstup - výstup žádá. Asynchronní provedení vstupu - výstupu umožňuje jednotkovému procesu, který vyslal požadavek, pokračovat v práci, zatímco operace vstupu - výstupu probíhá.

Databázový server provádí veškerý databázový vstup - výstup asynchronně pomocí jednoho z následujících mechanismů:

- Virtuální procesory AIO.
- Vstup - výstup KAIO na platformách, které ho podporují.

Databázový vstup - výstup zahrnuje vstup - výstup pro příkazy SQL, čtení napřed, čištění stránek a kontrolní body, spolu s dalšími vstupy - výstupy.

**Asynchronní vstup - výstup jádra:** Databázový server využívá vstup - výstup KAIO, pokud jsou splněny následující podmínky:

- Je podporován počítačem i operačním systémem.

- Dojde ke zvýšení výkonu.
- Vstup - výstup se týká diskového prostoru s přímým přístupem.

Databázový server implementuje KAIO spuštěním jednotkového procesu KAIO na virtuálním procesoru CPU. Jednotkový proces KAIO zajišťuje vstup - výstup prostřednictvím systémových volání operačního systému, což vede k zajištění vstupu - výstupu nezávisle na virtuálním procesoru. Jednotkový proces KAIO může zajistit lepší výkon diskového vstupu - výstupu než virtuální procesor AIO, protože nevyžaduje přepínání mezi virtuálními procesory CPU a AIO.

#### Jen pro UNIX

Dynamický server implementuje KAIO v případě, kdy je tato vlastnost podporována platformou, do které vedou porty dynamického serveru. Administrátor databázového serveru vstup - výstup KAIO nekonfiguruje. Chcete-li zjistit, zda použitá platforma podporuje KAIO, nahlédněte do souboru poznámek k počítači.

#### Konec Jen pro UNIX

#### Jen pro Linux

Asynchronní vstup - výstup jádra (KAIO) je ve výchozím nastavení povolený. Zadáte-li v prostředí procesu, které zobrazí (vyvolá) server, že KAIOFF=1, můžete tuto možnost zakázat.

Systém Linux má celosystémové omezení maximálního počtu paralelních požadavků KAIO. Tuto hodnotu obsahuje soubor **/proc/sys/fs/aio-max-nr**. Administrátor systému Linux může hodnotu zvýšit, například pomocí příkazu:

```
echo new_value > /proc/sys/fs/aio-max-nr
```

Aktuální počet přidělených příkazů všech procesů operačního systému je k vidění v souboru **/proc/sys/fs/aio-nr**.

Ve výchozím nastavení přiděluje dynamická verze polovinu z maximálního počtu požadavků a přiřazuje je rovnoměrně podle počtu konfigurovaných virtuálních procesorů CPU. Chcete-li řídit počet požadavků, které se přidělí jednotlivým virtuálním procesorům CPU, můžete použít proměnnou prostředí KAIOON. Provedete to tak, že nastavíte proměnnou KAIOON na požadovanou hodnotu předtím, než spustíte dynamický server.

Minimální hodnota proměnné prostředí KAIOON je 100. Pokud v systému Linux docházejí zdroje pro vstup - výstup KAIO, například když se dynamicky přidává mnoho virtuálních procesorů CPU, do souboru **online.log** jsou vypsána varování. V případě, že se tak stane, měl by administrátor systému Linux přidat zdroje vstupu - výstupu KAIO pomocí postupu popsaného výše.

#### Konec Jen pro Linux

**Virtuální procesory AIO:** V případě, že platforma nepodporuje vstup - výstup KAIO nebo pokud server používá vstup - výstup bloků souborů s použitím vyrovnávací paměti, bude databázový server provádět vstup - výstup prostřednictvím virtuálních procesorů třídy AIO. Všechny virtuální procesory AIO obsluhují veškeré požadavky vstupu - výstupu v rámci své třídy rovnoměrně.

Databázový server přiřadí každému bloku disku frontu, někdy označovanou též jako *fronta gfd*, v závislosti na názvu souboru bloku. Databázový server řadí požadavky vstupu - výstupu

ve frontě pomocí algoritmu, který minimalizuje pohyb diskové hlavy. Virtuální procesory AIO obsluhují fronty, které obsahují nevyřízené úkoly metodou cyklické obsluhy.

Veškeré další neblokované operace vstupu - výstupu jsou zařazeny do fronty AIO.

Použijte parametr VPCLASS s klíčovým slovem *aio* k určení počtu virtuálních procesorů AIO, které databázový server spustí jako první. Další informace o parametru VPCLASS naleznete v kapitole věnované konfiguračním parametrům v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Pokud je databázový server v režimu online, můžete spustit další virtuální procesory AIO. Další informace naleznete v části "Přidání virtuálních procesorů v režimu online" na stránce 6-3.

Virtuální procesory AIO nelze vypouštět, když je databázový server v režimu online.

**Automatické zvyšování a snižování počtu virtuálních procesorů AIO:** Konfigurační parametr AUTO\_AIOVPS povoluje nebo zakazuje databázovému serveru automaticky zvyšovat počet AIO VPS a jednotkových procesů vyprazdňovače v případě, kdy server zjistí, že AIO VPs nestačí pracovnímu zatížení vstupu - výstupu. Zakažte tento parametr, pokud chcete počet ručně upravit. Další informace o tom, jak tento parametr nastavit, naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

**Stanovení počtu potřebných virtuálních procesorů AIO:** Cílem při přidělování virtuálních procesorů AIO je přidělit takové množství VP, aby byly fronty požadavků vstupu - výstupu krátké, tedy aby fronty obsahovaly co nejmenší počet požadavků vstupu - výstupu. Pokud jsou fronty *gfd* po dlouhou dobu krátké, znamená to, že je vstup - výstup diskových zařízení zpracováván se stejnou rychlostí, s jakou přicházejí další požadavky.

Příkaz **onstat -g ioq** zobrazí délku a další statistické údaje ohledně front vstupu - výstupu. Tento příkaz vám umožní sledovat délku front *gfd* virtuálních procesorů AIO. Další informace naleznete v části "Monitorování virtuálních procesorů" na stránce 6-5 a informace ohledně sledování virtuálních procesorů naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Jeden virtuální procesor AIO by mohl být dostatečný:

- Pokud databázový server na vaší platformě implementuje jádro asynchronního vstupu - výstupu (KAIO) a pokud se všechny prostory typu dbspace skládají z diskového prostoru s přímým přístupem
- Pokud váš systém souborů podporuje přímý vstup - výstup pro velikost stránky použité pro blok prostoru typu dbspace a pokud přímý vstup - výstup používáte

Aktivnímu prostoru typu dbspace, který se skládá z bloků souborů s vyrovnávací pamětí, přidělte dva virtuální procesory AIO.

- Pokud databázový server implementuje výstup - výstup KAIO, ale pro bloky používáte několik souborů s vyrovnávací pamětí
- Pokud není vstup - výstup KAIO podporován systémem pro bloky.

Pokud na použité platformě není vstup - výstup KAIO implementován, přidělte dva virtuální procesory AIO každému disku, ke kterému databázový server často přistupuje.

V případě, že použijete předpřipravené soubory a povolíte přímý vstup - výstup pomocí konfiguračního parametru DIRECT\_IO, možná budete schopni snížit počet virtuálních procesorů AIO.

Pokud databázový server implementuje vstup - výstup KAIO a povolíte přímý vstup - výstup pomocí konfiguračního parametru `DIRECT_IO`, dynamický server se pokusí použít vstup - výstup KAIO, takže pravděpodobně nebudete potřebovat více než jeden virtuální procesor AIO. I v případě, že je povolený přímý vstup - výstup, ale systém souborů přímý vstup - výstup ani vstup - výstup KAIO nepodporuje, budete stále muset každému aktivnímu prostoru typu `dbspace`, který se skládá z bloků souborů s vyrovnávací pamětí, nebo nepoužívá vstup - výstup KAIO, přiřadit dva virtuální procesory AIO.

Dočasné prostory typu `dbspace` přímý vstup - výstup nepoužívají. V případě, že máte dočasné prostory typu `dbspace`, budete pravděpodobně potřebovat více než jeden virtuální procesor AIO.

Přidělte dostatečný počet virtuálních procesorů AIO, aby server dokázal zpracovávat maximální počet požadavků vstupu - výstupu. Obecně není přidělení příliš mnoha virtuálních procesorů AIO na závadu.

Konfigurační parametr `AUTO_AIOVPS` můžete použít k povolení databázovému serveru automaticky zvyšovat počet virtuálních procesorů AIO a jednotkových procesů čištění stránky, pokud servery zjistí, že virtuální procesory AIO nestačí zátěži na vstupu - výstupu.

## Síťové virtuální procesory

Jak popisuje část Kapitola 3, “Komunikace mezi klientem a serverem”, na stránce 3-1, klient se může k databázovému serveru připojit následujícími způsoby:

- Prostřednictvím síťového připojení.
- Prostřednictvím propojení procesů.

Jen pro UNIX

- Prostřednictvím sdílené paměti.

Konec Jen pro UNIX

Síťové připojení může být vytvořeno klientem na vzdáleném počítači nebo klientem na lokálním počítači napodobujícím připojení ze vzdáleného počítače (nazývané *připojení prostřednictvím místní zpětné smyčky*).

### Určení síťových připojení

Parametry `DBSERVERNAME` a `DBSERVERALIASES` obecně definují názvy `dbservername`, pro které existují odpovídající záznamy v souboru nebo registru `sqlhosts`. Každý parametr `dbservername` v souboru `sqlhosts` má záznam `nettype`, který určuje kombinaci rozhraní a protokolu. Databázový server spouští jeden nebo více *jednotkových procesů cyklického dotazování* pro každý jedinečný záznam `nettype`. Podrobný popis pole `nettype` naleznete v části “Pole typu připojení” na stránce 3-17.

Konfigurační parametr `NETTYPE` poskytuje nepovinné konfigurační údaje o kombinaci rozhraní a protokolu. Umožňuje přidělit kombinaci rozhraní a protokolu více než jeden jednotkový proces cyklického dotazování a též určit třídu virtuálního procesoru (CPU nebo NET), v níž budou jednotkové procesory cyklického dotazování spouštěny.

Úplný popis konfiguračního parametru `NETTYPE` naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Spouštění jednotkových procesů cyklického dotazování na virtuálních procesorech CPU nebo síťových virtuálních procesorech

Jednotkové procesy cyklického dotazování lze spustit na virtuálních procesorech CPU nebo síťových virtuálních procesorech. Jednotkové procesy cyklického dotazování obecně (zvláště pak na jednoprocessorových počítačích) pracují efektivněji na virtuálních procesorech CPU. To ovšem nemusí platit u víceprocesorových počítačů s velkým počtem vzdálených klientů.

Parametr NETTYPE obsahuje nepovinný záznam, nazývaný **vp class**, který umožňuje určit hodnotu CPU pro třídu virtuálního procesoru CPU nebo hodnotu NET pro síťové virtuální procesory.

Neurčíte-li třídu virtuálního procesoru pro kombinaci rozhraní a protokolu (jednotkové procesy cyklického dotazování) přidruženou k proměnné DBSERVERNAME, bude třída nastavena na výchozí třídu CPU. Databázový server předpokládá, že kombinace rozhraní a protokolu přidružená k proměnné DBSERVERNAME je primární kombinací rozhraní a protokolu a tedy by měla být nejvíce efektivní.

U ostatních kombinací rozhraní a protokolu je v případě, že není určena proměnná **vp class**, jako výchozí nastavena třída NET.

Pokud se databázový server nachází v režimu online, nelze vypustit virtuální procesor CPU, na kterém je spuštěn jednotkový proces cyklického dotazování.

**Poznámka:** Měli byste pečlivě rozlišovat mezi jednotkovými procesy cyklického dotazování pro síťová připojení a jednotkovými procesy cyklického dotazování pro připojení prostřednictvím sdílené paměti, které by měly být spuštěny vždy jeden v každém virtuálním procesoru CPU. Připojení TCP by měla být vytvářena pouze v síťových virtuálních procesorech, a to v minimálním počtu potřebném k dosažení požadované odezvy. Připojení prostřednictvím sdílené paměti by měla být vytvářena pouze ve virtuálních procesorech CPU a měla by být spuštěna v každém virtuálním procesoru CPU.

## Stanovení počtu síťových virtuálních procesorů

Každý jednotkový proces cyklického dotazování vyžaduje samostatný virtuální procesor, takže určením počtu jednotkových procesů cyklického dotazování pro kombinaci rozhraní a protokolu nepřímo určíte i počet síťových virtuálních procesorů, určíte-li zároveň, že budou spouštěny třídou virtuálních procesorů NET. Určíte-li hodnotu CPU jako proměnnou **vp class**, musíte přidělit dostatečný počet virtuálních procesorů CPU pro spuštění jednotkových procesů cyklického dotazování. Nemá-li databázový server virtuální procesor CPU ke spuštění jednotkového procesu cyklického dotazování CPU, spustí pro něj síťový virtuální procesor stanovené třídy.

U většiny systémů postačuje jeden jednotkový proces cyklického dotazování a tedy i jeden virtuální procesor na jednu kombinaci síťového rozhraní a protokolu. U systémů s 200 a více uživateli sítě může spuštění dalších síťových virtuálních procesorů zlepšit výkon. V takovém případě budete muset zjistit optimální počet virtuálních procesorů pro každou kombinaci rozhraní a protokolu experimentálně.

## Určení jednotkových procesů listen a cyklického dotazování pro připojení klienta k serveru

Když spouštíte databázový server, proces **oninit** spustí interní jednotkový proces nazývaný *jednotkový proces typu listener* pro každý název dbservername, který určíte pomocí parametru DBSERVERNAME a DBSERVERALIASES v souboru ONCONFIG. Chcete-li určit port, na kterém databázový server očekává připojení pro každý z těchto záznamů dbservername, přiřaďte mu jedinečnou kombinaci záznamů **hostname** a **service name** v souboru **sqlhosts**.

Záznam v souboru nebo registru **sqlhosts**, který uvádí Tabulka 5-3, například způsobí, že databázový server **soc\_011** spustí jednotkový proces typu listener pro **port1** u hostitele nebo na síťové adrese **myhost**.

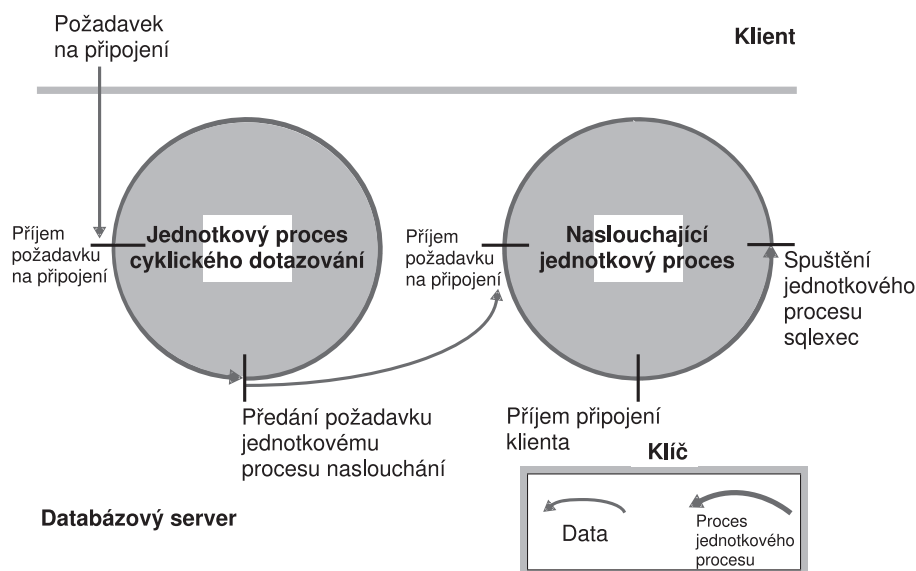
Tabulka 5-3. Jednotkový proces typu listener pro každý port, na kterém databázový server očekává připojení

| dbservername | nettype  | hostname | název služby |
|--------------|----------|----------|--------------|
| soc_011      | onsoctcp | myhost   | port1        |

Jednotkový proces typu listener otevře port a vyžádá si jeden z jednotkových procesů cyklického dotazování pro příslušnou kombinaci rozhraní a protokolu k monitorování klientských požadavků na tomto portu. Jednotkový proces cyklického dotazování může být spuštěn ve virtuálním procesoru CPU nebo v síťovém virtuálním procesoru použitého připojení. Další informace o počtu jednotkových procesů cyklického dotazování naleznete v části “Stanovení počtu síťových virtuálních procesorů” na stránce 5-23.

Informace o tom, jak určit, zda budou jednotkové procesy cyklického dotazování pro určitou kombinaci rozhraní a protokolu pracovat ve virtuálním procesoru CPU nebo v síťovém virtuálním procesoru, naleznete v části “Spouštění jednotkových procesů cyklického dotazování na virtuálních procesorech CPU nebo síťových virtuálních procesorech” na stránce 5-23 a informace o konfiguračním parametru NETTYPE naleznete v příručce *IBM Informix Administrator's Reference*.

Pokud jednotkový proces cyklického dotazování obdrží od klienta požadavek na připojení, předá tento požadavek jednotkovému procesu typu listener pro tento port. Jednotkový proces typu listener ověří uživatele, vytvoří připojení k databázovému serveru a spustí jednotkový proces **sqlexec**, jednotkový proces relace, který provádí primární zpracování pro klienta. Obrázek Obrázek 5-8 znázorňuje roli jednotkových procesů typu listener a jednotkových procesů cyklického dotazování při vytváření spojení s klientskou aplikací.



Obrázek 5-8. “Role jednotkových procesorů cyklického dotazování a jednotkových procesů typu listener při připojení ke klientovi

Jednotkový proces cyklického zpracování čeká na požadavky klienta a umístí je do sdílené paměti, aby mohly být zpracovány jednotkovým procesem **sqlexec**. V případě síťových připojení umístí jednotkový proces cyklického dotazování zprávu do fronty v globální



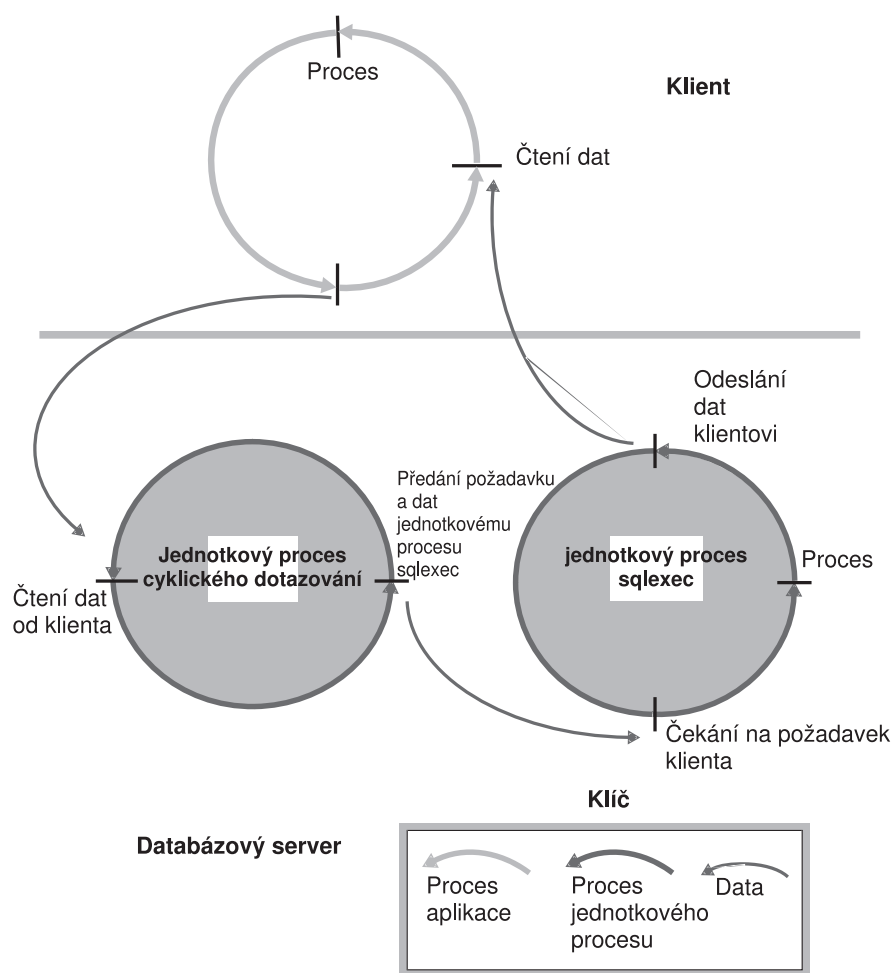
společné oblasti sdílené paměti. Jednotkový proces cyklického dotazování poté probudí jednotkový proces **sqlexec** klienta, aby zpracoval požadavek. Kdykoli je to možné, zapisuje jednotkový proces **sqlexec** zpět přímo do klienta bez pomoci jednotkového procesu cyklického dotazování. Jednotkový proces cyklického dotazování obecně načítá data z klienta a jednotkový proces **sqlexec** odesílá tato data ke klientovi.

**Jen pro UNIX**

V případě připojení prostřednictvím sdílené paměti umístí jednotkový proces cyklického dotazování zprávu do komunikační části sdílené paměti.

**Konec Jen pro UNIX**

Obrázek Obrázek 5-9 znázorňuje základní úlohy, které provádějí jednotkový proces cyklického dotazování a jednotkový proces **sqlexec** při komunikaci s klientskou aplikací.



Obrázek 5-9. Role jednotkových procesů cyklického dotazování a jednotkových procesů typu **sqlexec** při komunikaci s klientskou aplikací

### Použití rychlého cyklického dotazování

Konfigurační parametr FASTPOLL můžete použít k povolení nebo zakázání rychlého cyklického dotazování ve vaší síti, pokud platforma vašeho operačního systému rychle cyklické dotazování podporuje. Rychlé cyklické dotazování je užitečné, pokud máte velké

množství připojení. Pokud například máte více než 300 souběžných připojení s databázovým serverem, můžete pro zlepšení výkonu povolit konfigurační parametr FASTPOLL.

#### Chcete-li povolit rychlé cyklické dotazování:

1. Nastavte konfigurační parametr FASTPOLL na hodnotu 1.

V případě, že operační systém nepodporuje rychlé cyklické dotazování, dynamický server konfigurační parametr FASTPOLL ignoruje.

### Spuštění několika jednotkových procesů typu listener

Pokud databázový server nedokáže uspokojivě obsloužit požadavky na připojení pro danou kombinaci rozhraní a protokolu pomocí jediného portu a příslušného jednotkového procesu typu listener, můžete obsluhu požadavků na připojení zlepšit dvěma následujícími způsoby:

- Přidáním jednotkových procesů typu listener pro další porty.
- Přidáním další karty síťového rozhraní.

**Přidání jednotkových procesů typu listener:** Jak již bylo zmíněno dříve, databázový server spustí jednotkový proces pro každý název dbservername, který určíte pomocí konfiguračních parametrů DBSERVERNAME a DBSERVERALIASES.

Chcete-li přidat jednotkové procesy typu listener k dalším portům, musíte nejprve použít parametr DBSERVERALIASES k určení názvu dbservername pro každý z portů. Parametr DBSERVERALIASES, který znázorňuje Obrázek 5-10, například definuje dva další názvy dbservername **soc\_o12** a **soc\_o13** pro instanci databázového serveru označenou jako **soc\_o11**.

```
DBSERVERNAME soc_o11
DBSERVERALIASES soc_o12,soc_o13
```

*Obrázek 5-10. Definování několika názvů dbservername pro vícenásobná připojení stejného typu*

Poté, co definujete další názvy dbservername pro databázový server, je potřeba určit kombinaci rozhraní/protokol a port každého z nich v souboru **sqlhosts** nebo v registru. Každý port je určen jedinečnou kombinací záznamů **hostname** a **servicename**. Záznamy **sqlhosts**, které zobrazuje Tabulka 5-4, například způsobí, že databázový server spustí tři jednotkové procesy typu listener pro kombinaci rozhraní a protokolu **onsoctcp**, jednu pro každý z definovaných portů.

*Tabulka 5-4. Záznamy sqlhosts k naslouchání na několika portech pro jedinou kombinaci rozhraní a protokolu.*

| dbservername | nettype  | hostname | název služby |
|--------------|----------|----------|--------------|
| soc_o11      | onsoctcp | myhost   | port1        |
| soc_o12      | onsoctcp | myhost   | port2        |
| soc_o13      | onsoctcp | myhost   | port3        |

Pokud vložíte parametr NETTYPE pro kombinaci rozhraní a protokolu, bude tento parametr použit pro všechna připojení této kombinace rozhraní a protokolu. Jinými slovy, existuje-li parametr NETTYPE pro **onsoctcp**, jak znázorňuje Tabulka 5-4, je tento parametr použit pro všechna zobrazená připojení. V tomto příkladu spouští databázový server pouze jeden jednotkový proces *cyklického dotazování* pro kombinaci rozhraní a protokolu **onsoctcp**, neurčuje-li parametr NETTYPE více. Další informace o záznamech v souboru nebo registru **sqlhosts** naleznete v části “Propojovací soubory” na stránce 3-9.

**Přidání karty síťového rozhraní:** V případě, že karta síťového rozhraní hostitelského počítače nemůže uspokojivě obsloužit požadavky připojení, nebo chcete-li připojit databázový server k více sítím, můžete přidat další kartu síťového rozhraní.

Pro podporu více karet síťového rozhraní musíte každé z karet přiřadit jedinečný název **hostname** (síťovou adresu) v souboru nebo registru **sqlhosts**. Použijete-li například stejné názvy dbservername, jaké uvádí Obrázek 5-10, záznamy v souboru nebo registru **sqlhosts**, které uvádí Tabulka 5-5 na stránce 5-27, způsobí, že databázový server spustí tři jednotkové procesy typu listener pro tutéž kombinaci rozhraní a protokolu (podobně, jak to učinily záznamy, které uvádí Tabulka 5-4). V tomto případě ale dva z jednotkových procesů naslouchají na portech na jedné kartě rozhraní (**myhost1**) a třetí jednotkový proces naslouchá na portu na druhé kartě rozhraní (**myhost2**).

*Tabulka 5-5. Příklad záznamů sqlhosts pro podporu dvou karet síťového rozhraní pro kombinaci rozhraní a protokolu onsoctcp*

| dbservername | nettype  | hostname | název služby |
|--------------|----------|----------|--------------|
| soc_ol1      | onsoctcp | myhost1  | port1        |
| soc_ol2      | onsoctcp | myhost1  | port2        |
| soc_ol3      | onsoctcp | myhost2  | port1        |

## Virtuální procesor Communications Support Module

Třída virtuálních procesorů CSM (communications support module) zajišťuje služby podpory komunikačních prostředků a funkce modulu pro podporu komunikačních prostředků.

Databázový server spustí stejný počet virtuálních procesorů CSM jako virtuálních procesorů CPU.

Další informace o službě podpory komunikačních prostředků naleznete v části Kapitola 3, “Komunikace mezi klientem a serverem”, na stránce 3-1.

## Šifrovací virtuální procesory

Není-li v konfiguračním souboru ONCONFIG definovaná volba parametru VPCLASS **encrypt**, spustí databázový server při prvním volání šifrovací nebo dešifrovací funkce definované pro šifrování na úrovni sloupců jeden šifrovací virtuální procesor ENCRYPT VP. V případě potřeby můžete definovat několik šifrovacích virtuálních procesorů a snížit tak čas potřebný ke spuštění databázového serveru.

Ke konfiguraci šifrovacích virtuálních procesorů použijte konfigurační parametr VPCLASS spolu s klíčovým slovem **encrypt**. Chcete-li například přidat pět šifrovacích virtuálních procesorů, zadejte do souboru ONCONFIG následující údaje:

```
VPCLASS encrypt,num=5
```

Tytéž údaje lze upravovat pomocí obslužného programu **onmode**, jak to uvádí následující příklad:

```
onmode -p 5 encrypt
```

Další informace naleznete v kapitole věnované konfiguračním parametrům a obslužnému programu **onmode** v příručce *IBM Informix Dynamic Server Administrator's Reference*. Další informace o šifrování na úrovni sloupců naleznete v příručce *IBM Informix Security Guide*.

## Optický virtuální procesor

Třída optických virtuálních procesorů (OPT) se využívá jen společně s podsystémem Optical Subsystem. Podsystém Optical Subsystem spouští jeden virtuální procesor optické třídy, je-li přítomen konfigurační parametr STAGEBLOB. Další informace o podsystému Optical Subsystem naleznete v příručce *IBM Informix Optical Subsystem Guide*.

## Virtuální procesor typu audit

Zapnete-li režim auditování nastavením hodnoty parametru ADTMODE v souboru ONCONFIG na 1, databázový server spustí jeden virtuální procesor v třídě auditování (ADT). Další informace o auditování databázového serveru naleznete v příručce *IBM Informix Security Guide*.

## Virtuální procesor typu miscellaneous

Virtuální procesor typu miscellaneous obsluhuje požadavky na systémová volání, které mohou vyžadovat velmi velký zásobník, například načítání údajů o aktuálním uživateli nebo názvu hostitelského systému. V tomto virtuálním procesoru se spouští pouze jeden jednotkový proces, procesor ho provádí se zásobníkem o velikosti 128 kB.

---

## Kapitola 6. Správa virtuálních procesorů

|                                                                                                   |     |
|---------------------------------------------------------------------------------------------------|-----|
| Obsah kapitoly                                                                                    | 6-1 |
| Nastavení konfiguračních parametrů virtuálních procesorů                                          | 6-1 |
| Nastavení parametrů virtuálního procesoru pomocí textového editoru                                | 6-1 |
| Určení parametru VPCLASS                                                                          | 6-2 |
| Zakázání stárnutí priority (UNIX)                                                                 | 6-2 |
| Nastavení parametrů virtuálního procesoru pomocí programu ISA                                     | 6-2 |
| Nastavení parametrů virtuálního procesoru pomocí obslužného programu ON-Monitor                   | 6-3 |
| Spuštění a zastavení virtuálních procesorů                                                        | 6-3 |
| Přidání virtuálních procesorů v režimu online                                                     | 6-3 |
| Přidání virtuálních procesorů v režimu online pomocí obslužného programu onmode                   | 6-3 |
| Přidání virtuálních procesorů v režimu online pomocí obslužného programu ON-Monitor (systém UNIX) | 6-4 |
| Přidání síťových virtuálních procesorů                                                            | 6-4 |
| Vypuštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů         | 6-4 |
| Monitorování virtuálních procesorů                                                                | 6-5 |
| Monitorování virtuálních procesorů pomocí obslužných programů příkazového řádku                   | 6-5 |
| onstat -g ath                                                                                     | 6-5 |
| onstat -g glo                                                                                     | 6-5 |
| onstat -g ioq                                                                                     | 6-6 |
| onstat -g rea                                                                                     | 6-6 |
| Monitorování virtuálních procesorů pomocí tabulek SMI                                             | 6-6 |

---

### Obsah kapitoly

Tato kapitola popisuje, jak nastavit konfigurační parametry, které ovlivní virtuální procesory databázového serveru, a jak spustit a zastavit virtuální procesory.

Popis tříd virtuálních procesorů a informace o množství virtuálních procesorů, které lze určit pro každou z tříd, uvádí Kapitola 5, “Virtuální procesory a jednotkové procesy”, na stránce 5-1.

---

### Nastavení konfiguračních parametrů virtuálních procesorů

Jako uživatel **root** nebo **informix** použijte k nastavení konfiguračních parametrů pro virtuální procesory databázového serveru následující nástroje:

- textový editor
- IBM Informix Server Administrator (ISA)

**Jen pro UNIX**

- ON-Monitor

**Konec Jen pro UNIX**

K zavedení jakýchkoliv změn konfiguračních parametrů je nutné ukončit a restartovat databázový server. Další informace naleznete v části “Nastavení sdílené paměti” na stránce 8-6.

### Nastavení parametrů virtuálního procesoru pomocí textového editoru

K nastavení parametrů ONCONFIG můžete využít textový editor. Pomocí editoru vyhledejte parametr, který chcete změnit, zadejte novou hodnotu a uložte soubor na disk.

Tabulka 6-1 obsahuje seznam parametrů ONCONFIG, používaných ke konfigurování virtuálních procesorů. Více informací o tom, jak tyto parametry ovlivňují virtuální procesory, naleznete v příručce “Třídy virtuálních procesorů” na stránce 5-12.

Tabulka 6-1. Parametry pro konfigurování virtuálních procesorů

| Parametr           | Podparametry                                        | Účel                                                                                     |
|--------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------|
| MULTIPROCESSOR     |                                                     | Určuje, že používáte víceprocesorový počítač.                                            |
| NETTYPE            |                                                     | Určuje parametry jednotkových procesů síťových protokolů (a virtuálních procesorů).      |
| SINGLE_CPU_VP      |                                                     | Určuje, že používáte jediný virtuální procesor typu CPU.                                 |
| VPCLASS            | adm adt cpu kio lio msc opt lio shm soc str tli     | Určuje třídu virtuálních procesorů.                                                      |
| VPCLASS            | num=num_VPs                                         | Určuje počet virtuálních procesorů určité třídy, které by měl databázový server spustit. |
| VPCLASS            | max=max_VPs                                         | Určuje maximální počet virtuálních procesorů povolených pro jednu třídu.                 |
| VPCLASS            | noage                                               | Určuje, že byla zakázána funkce stárnutí priority.                                       |
| VPCLASS            | aff=processor_number<br>aff=start_range - end_range | Určuje přiřazení k jednotce CPU, je-li k dispozici funkce procesorové afinity.           |
| VPCLASS            | user_defined                                        | Určuje uživatelem definovaný (uživatelský) virtuální procesor.                           |
| VPCLASS            | jvp                                                 | Určuje virtuální procesor Java virtual processor (JVP).                                  |
| VPCLASS            | noyield                                             | Určuje virtuální procesor nepředávající řízení.                                          |
| VP_MEMORY_CACHE_KB |                                                     | Urychluje přístup k paměťovým blokům.                                                    |

## Určení parametru VPCLASS

Měli byste místo parametrů NUMCPUVPS, NUMAIOVPS, NOAGE, AFF\_SPROC a AFF\_NPROCS použít parametr VPCLASS. Můžete zadat název parametru VPCLASS o délce až 128 bajtů. Název parametru VPCLASS musí začínat písmenem nebo podtržítkem a může obsahovat písmena, číslice, podtržítka nebo znaky \$.

Uživatelem definované třídy virtuálních procesorů a virtuální procesory Java určíte v rámci konfiguračního parametru VPCLASS. Více informací o konfiguračním parametru VPCLASS včetně výchozích nastavení a rozsahů hodnot naleznete v částech *IBM Informix Dynamic Server Administrator's Reference*, “Uživatelem definované třídy virtuálních procesorů” na stránce 5-15 a “Virtuální procesory Java” na stránce 5-17.

## Zakázání stárnutí priority (UNIX)

Chcete-li zakázat proces stárnutí priority na platformách umožňujících tuto funkci, použijte parametr VPCLASS s volbou *noage*.

Doporučené hodnoty pro tyto parametry databázového serveru v systému UNIX naleznete v souboru Poznámky k počítači.

## Nastavení parametrů virtuálního procesoru pomocí programu ISA

Program ISA můžete použít k zobrazení informací o třídách virtuálního procesoru a k monitorování, přidávání nebo odebrání tříd virtuálních procesorů. Další informace naleznete v nápovědě online k programu ISA.

## Nastavení parametrů virtuálního procesoru pomocí obslužného programu ON-Monitor

Pro nastavení konfiguračních parametrů virtuálního procesoru pomocí obslužného programu ON-Monitor vyberte volbu **Parameters > perFormance**.

K určení síťových virtuálních procesorů zadejte počet virtuálních procesorů a pak jednu z následujících kombinací rozhraní/protokolu: **ipeshm**, **ipestr**, **tlitcp**, **tlispx** nebo **soctcp**.

---

## Spuštění a zastavení virtuálních procesorů

Když spustíte databázový server, obslužný program **oninit** spustí celou řadu různých virtuálních procesorů, které jste přímo nebo nepřímo určili. Virtuální procesory konfiguruje primárně prostřednictvím parametrů ONCONFIG a v případě síťových virtuálních procesorů prostřednictvím parametrů v souboru nebo registru **sqlhosts**. Více informací o třídách virtuálních procesorů naleznete v části “Třídy virtuálních procesorů” na stránce 5-12.

Databázový server umožňuje spustit nanejvýš 1000 virtuálních procesorů.

Až bude databázový server v režimu online, můžete spustit další virtuální procesory, a v případě potřeby tak zvýšit výkon. Další informace naleznete v následující části Přidání virtuálních procesorů v režimu online.

Pokud je databázový server v režimu online, můžete vypouštět virtuální procesory třídy CPU a uživatelem definované třídy. Více informací naleznete v části “Vypuštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů” na stránce 6-4.

K ukončení databázového serveru a tudíž i ukončení všech virtuálních procesorů použijte příkaz **onmode -k**. Více informací o použití příkazu **onmode -k** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Přidání virtuálních procesorů v režimu online

Když je databázový server v režimu online, můžete spustit další virtuální procesory těchto tříd: CPU, AIO, PIO, LIO, SHM, STR, TLI, SOC, JVP a uživatelem definované třídy. Databázový server automaticky spustí po jednom virtuálním procesoru tříd LIO a PIO s výjimkou případu, kdy je využíváno zrcadlení - v takovém případě spustí dva virtuální procesory pro každou z těchto tříd.

Tyto virtuální procesory můžete spustit některou z následujících metod:

- volba **-p** obslužného programu **onmode**
- ISA

Můžete též spustit další virtuální procesory z uživatelem definovaných tříd pro spuštění uživatelských rutin. Více informací o uživatelem definovaných virtuálních procesorech naleznete v části “Přiřazení UDR k uživatelem definované třídě virtuálních procesorů” na stránce 5-16.

## Přidání virtuálních procesorů v režimu online pomocí obslužného programu onmode

Pokud je databázový server v režimu online, použijte pro přidání virtuálních procesorů volbu **-p** příkazu **onmode**. Počet virtuálních procesorů, které chcete přidat, určete kladným celým číslem. Před číslem udávající počet virtuálních procesorů můžete případně vložit znaménko plus (+). Za číslem uveďte malými písmeny třídu virtuálního procesoru. Oba následující příkazy například spouštějí čtyři další virtuální procesory třídy AIO:

```
onmode -p 4 aio
```

```
onmode -p +4 aio
```

Obslužný program **onmode** okamžitě spustí další virtuální procesory.

Přidávat však lze vždy jen virtuální procesory jedné třídy. Chcete-li přidat virtuální procesory jiné třídy, musíte znovu spustit obslužný program **onmode**.

### **Přidání virtuálních procesorů v režimu online pomocí obslužného programu ON-Monitor (systém UNIX)**

Chcete-li přidat k databázovému serveru v režimu online virtuální procesory pomocí programu ON-Monitor, vyberte volbu **Režimy> Přidat-proc**. Můžete přidat virtuální procesory následujících tříd: CPU, AIO, LIO, PIO a NET.

K určení síťových virtuálních procesorů zadejte nejprve počet virtuálních procesorů a pak jednu z následujících kombinací rozhraní/protokolu: **ipcshm**, **ipcstr**, **tlitcp**, **tlispx** nebo **soctcp**.

Obslužný program ON-Monitor nelze použít ke spuštění dalších virtuálních procesorů uživatelem definované třídy. Více informací naleznete v části "Přidání virtuálních procesorů v režimu online pomocí obslužného programu onmode" na stránce 6-3.

### **Přidání síťových virtuálních procesorů**

Když přidáváte síťové virtuální procesory, přidáváte jednotkové procesy cyklického dotazování, z nichž každý vyžaduje spuštění vlastního virtuálního procesoru. Pokud se pro daný protokol pokoušíte přidat jednotkové procesy cyklického dotazování, zatímco je databázový server v režimu online, a určili jste v konfiguračním parametru NETTYPE, že jsou jednotkové procesy cyklického dotazování spouštěny ve třídě CPU, nespustí databázový server nové jednotkové procesy cyklického dotazování, nejsou-li k dispozici žádné virtuální procesory CPU, které by je mohly spustit.

V následujícím příkladu zpracovávají jednotkové procesy cyklického dotazování celkem 240 připojení:

```
NETTYPE ipcshm,4,60,CPU # Konfigurace jednotkových podprocesů cyklického dotazování pro typ for nettype
```

Počet jednotkových procesů cyklického dotazování pro protokol ipcshm odpovídá počtu paměťových segmentů. Je-li například konfigurační parametr NETTYPE nastaven na hodnotu 3100 a chcete-li jeden vyzvaný podproces, nastavte ho na hodnotu 1300.

## **Vypuštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů**

Když je databázový server v režimu online, můžete použít volbu **-p** obslužného programu **onmode** k vypuštění nebo ukončení virtuálních procesorů třídy CPU a uživatelem definované třídy.

### **Vypuštění virtuálních procesorů třídy CPU:**

Za příkaz **onmode** vložte záporné číslo vyjadřující počet virtuálních procesorů, které chcete vypustit a pak malými písmeny určete třídu CPU. Následující příkaz například vypustí dva virtuální procesory třídy CPU:

```
% onmode -p -2 cpu
```

Pokusíte-li se vypustit virtuální procesor typu CPU, který spouští jednotkový proces cyklického dotazování, zobrazí se následující zpráva:



onmode: failed when trying to change the number of cpu virtual processor by *-number*.

Další informace naleznete v části “Spouštění jednotkových procesů cyklického dotazování na virtuálních procesorech CPU nebo síťových virtuálních procesorech” na stránce 5-23.

#### Vypuštění uživatelem definovaných virtuálních procesorů:

Za příkaz **onmode** vložte záporné číslo vyjadřující počet virtuálních procesorů, které chcete vypustit a pak malými písmeny určete uživatelem definovanou třídu. Následující příkaz například vypustí dva virtuální procesory třídy *usr*:

```
onmode -p -2 usr
```

#### Jen pro Windows

V systému Windows můžete mít vždy jen jednu uživatelem definovanou třídu virtuálních procesorů. Vynechejte parametr *číslo* v příkazu **onmode -p třída virtuálního procesoru**.

#### Konec Jen pro Windows

Informace o tom, jak vytvářet uživatelem definované třídy virtuálních procesorů a jak k nim přiřazovat uživatelské rutiny, naleznete v části “Uživatelé definované třídy virtuálních procesorů” na stránce 5-15.

---

## Monitorování virtuálních procesorů

Monitorování virtuálních procesorů slouží ke zjištění, zda je počet virtuálních procesorů konfigurovaných pro databázový server optimální pro aktuální úroveň aktivity. Více informací o těchto volbách obslužného programu **onstat -g** naleznete v kapitole věnované vlivu konfigurace na využití jednotky CPU v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Příklady výstupu příkazů **onstat -g** naleznete v části týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

## Monitorování virtuálních procesorů pomocí obslužných programů příkazového řádku

K monitorování virtuálních procesorů můžete použít následující volby obslužného programu **onstat -g** :

- **ath**
- **glo**
- **ioq**
- **rea**

### **onstat -g ath**

Příkaz **onstat -g ath** zobrazí informace o systémových jednotkových procesech a třídách virtuálních procesorů.

### **onstat -g glo**

Příkaz **onstat -g glo** slouží k zobrazení informací o každém právě běžícím virtuálním procesoru a též kumulativní statistiky pro každou z tříd virtuálních procesorů. Příklad výstupu příkazu **onstat -g glo** naleznete v části týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

## **onstat -g ioq**

Pomocí volby **onstat -g ioq** lze zjistit, zda je nutné přidělit další virtuální procesory. Příkaz **onstat -g ioq** zobrazí délku a další statistické údaje o frontách vstupu - výstupu.

Pokud délka fronty vstupu - výstupu narůstá, hromadí se požadavky vstupu - výstupu rychleji, než mohou být zpracovány virtuálními procesory typu AIO. Naznačuje-li délka fronty vstupu - výstupu, že dochází k hromadění požadavků vstupu - výstupu, zvažte možnost přidání virtuálních procesorů typu AIO.

Příklad výstupu příkazu **onstat -g ioq** naleznete v příručce *IBM Informix Administrator's Reference*.

## **onstat -g rea**

K monitorování počtu jednotkových procesů ve frontě připravených procesů použijte volbu **onstat -g rea**. Pokud počet jednotkových procesů jedné třídy virtuálních procesorů (například třídy CPU) ve frontě připravených procesů roste, bude pravděpodobně nutné rozšířit konfiguraci o další virtuální procesory.

Příklad výstupu příkazu **onstat -g rea** naleznete v příručce *IBM Informix Administrator's Reference*.

## **Monitorování virtuálních procesorů pomocí tabulek SMI**

Informace o právě spuštěných virtuálních procesorech vyhledáte v tabulce **sysvpprof**. Tato tabulka obsahuje následující sloupce.

| <b>sloupec</b> | <b>Popis</b>                                       |
|----------------|----------------------------------------------------|
| <b>vpid</b>    | Číslo ID virtuálního procesoru.                    |
| <b>class</b>   | Třída virtuálního procesoru.                       |
| <b>usercpu</b> | Spotřebované minuty času uživatelské jednotky CPU. |
| <b>syscpu</b>  | Spotřebované minuty času systémové jednotky CPU.   |

---

## Kapitola 7. Sdílená paměť

|                                                                                               |      |
|-----------------------------------------------------------------------------------------------|------|
| Obsah kapitoly . . . . .                                                                      | 7-2  |
| Sdílená paměť . . . . .                                                                       | 7-2  |
| Použití sdílené paměti . . . . .                                                              | 7-2  |
| Přidělování sdílené paměti . . . . .                                                          | 7-3  |
| Velikost sdílené paměti . . . . .                                                             | 7-4  |
| Jaké akce provést, pokud je překročena hodnota parametru SHMTOTAL . . . . .                   | 7-5  |
| Procesy, které se připojují ke sdílené paměti . . . . .                                       | 7-5  |
| Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX) . . . . .           | 7-6  |
| Jak se ke sdílené paměti připojují obslužné programy . . . . .                                | 7-6  |
| Jak se ke sdílené paměti připojují virtuální procesory . . . . .                              | 7-6  |
| Získání hodnot klíčů segmentů sdílené paměti . . . . .                                        | 7-7  |
| Určení místa připojení prvního segmentu sdílené paměti . . . . .                              | 7-7  |
| Připojení dalších segmentů sdílené paměti . . . . .                                           | 7-8  |
| Definování adresy dolní hranice sdílené paměti . . . . .                                      | 7-8  |
| Rezidentní segmenty sdílené paměti . . . . .                                                  | 7-9  |
| Rezidentní část sdílené paměti . . . . .                                                      | 7-9  |
| Záhlaví sdílené paměti . . . . .                                                              | 7-10 |
| Společná oblast vyrovnávacích paměti ve sdílené paměti . . . . .                              | 7-10 |
| Přetečení vyrovnávacích paměti do virtuální části . . . . .                                   | 7-11 |
| Velikost vyrovnávací paměti . . . . .                                                         | 7-11 |
| Vyrovnávací paměť logického protokolu . . . . .                                               | 7-11 |
| Vyrovnávací paměť fyzického protokolu . . . . .                                               | 7-12 |
| Vyrovnávací paměť replikace High-Availability Data Replication . . . . .                      | 7-13 |
| Tabulka zámek . . . . .                                                                       | 7-13 |
| Virtuální část sdílené paměti . . . . .                                                       | 7-14 |
| Správa virtuální části sdílené paměti . . . . .                                               | 7-14 |
| Velikost virtuální části sdílené paměti . . . . .                                             | 7-14 |
| Součásti virtuální části sdílené paměti . . . . .                                             | 7-15 |
| Interní tabulky sdílené paměti . . . . .                                                      | 7-15 |
| Velké vyrovnávací paměti . . . . .                                                            | 7-18 |
| Data relací . . . . .                                                                         | 7-18 |
| Data jednotkových procesů . . . . .                                                           | 7-18 |
| Mezipaměť distribuce dat . . . . .                                                            | 7-19 |
| Mezipaměť slovníku . . . . .                                                                  | 7-19 |
| Mezipaměť příkazů SQL . . . . .                                                               | 7-19 |
| Řadicí paměť . . . . .                                                                        | 7-20 |
| Rutiny SPL a mezipaměť rutin UDR . . . . .                                                    | 7-20 |
| Globální společná oblast . . . . .                                                            | 7-20 |
| Komunikační část sdílené paměti (systém UNIX) . . . . .                                       | 7-20 |
| Část virtuálních rozšíření sdílené paměti . . . . .                                           | 7-21 |
| Řízení souběžného přístupu . . . . .                                                          | 7-21 |
| Objekty mutex sdílené paměti . . . . .                                                        | 7-21 |
| Zámky vyrovnávacích paměti ve sdílené paměti . . . . .                                        | 7-21 |
| Typy zámek vyrovnávacích paměti . . . . .                                                     | 7-21 |
| Přístup jednotkových procesů databázového serveru k sdíleným vyrovnávacím pamětem . . . . .   | 7-22 |
| Fronty FIFO/LRU . . . . .                                                                     | 7-22 |
| Komponenty front LRU . . . . .                                                                | 7-22 |
| Stránky seřazené podle nejdelsí doby nepoužití . . . . .                                      | 7-23 |
| Správa společné oblasti vyrovnávací paměti a front LRU . . . . .                              | 7-23 |
| Počet front LRU, které je třeba nakonfigurovat . . . . .                                      | 7-23 |
| Počet jednotkových procesů čištění, které je třeba přidělit . . . . .                         | 7-24 |
| Počet stránek přidávaných do front MLRU . . . . .                                             | 7-24 |
| Konec čištění fronty MLRU . . . . .                                                           | 7-25 |
| Konfigurace dopředného čtení databázového serveru . . . . .                                   | 7-26 |
| Přístup jednotkových procesů databázového serveru ke stránkám ve vyrovnávací paměti . . . . . | 7-26 |

|                                                                                                                  |      |
|------------------------------------------------------------------------------------------------------------------|------|
| Vyprazdňování dat na disk. . . . .                                                                               | 7-26 |
| Vyprazdňování vyrovnávacích pamětí ve společné oblasti . . . . .                                                 | 7-27 |
| Přednostní vyprazdňování předobrazů . . . . .                                                                    | 7-27 |
| Vyprazdňování vyrovnávací paměti fyzického protokolu. . . . .                                                    | 7-27 |
| Synchronizace vyprazdňování vyrovnávacích pamětí . . . . .                                                       | 7-28 |
| Popis aktivity vyprazdňování . . . . .                                                                           | 7-28 |
| Zápis na popředí. . . . .                                                                                        | 7-28 |
| Zápisy front LRU . . . . .                                                                                       | 7-28 |
| Zápisy bloků . . . . .                                                                                           | 7-29 |
| Vyprazdňování vyrovnávací paměti logického protokolu. . . . .                                                    | 7-29 |
| Po přípravě nebo dokončení transakce v databázi s protokolováním bez vyrovnávací paměti . . . . .                | 7-30 |
| Při ukončení relace, která používá databáze bez protokolování nebo protokolování bez vyrovnávací paměti. . . . . | 7-30 |
| Pokud nastane kontrolní bod . . . . .                                                                            | 7-30 |
| Pokud byla změněna stránka, která nevyžaduje předobraz v souboru fyzického protokolu . . . . .                   | 7-30 |
| Použití vyrovnávací paměti pro velké objekty . . . . .                                                           | 7-30 |
| Zápis jednoduchých velkých objektů. . . . .                                                                      | 7-30 |
| Stránky blobpage a sdílená paměť . . . . .                                                                       | 7-30 |
| Vytváření jednoduchých velkých objektů . . . . .                                                                 | 7-31 |
| Vytvoření vyrovnávacích pamětí stránek blobpage . . . . .                                                        | 7-31 |
| Přístup k inteligentním velkým objektům . . . . .                                                                | 7-32 |
| Využití paměti na 64bitových platformách . . . . .                                                               | 7-33 |

---

## Obsah kapitoly

Tato kapitola popisuje obsah sdílené paměti databázového serveru, faktory, které určují velikost oblastí sdílené paměti a přesuny dat do sdílené paměti a ze sdílené paměti. Postup při změně konfiguračních parametrů databázového serveru, které určují přidělování sdílené paměti popisuje Kapitola 8, “Správa sdílené paměti”, na stránce 8-1.

---

## Sdílená paměť

Sdílená paměť je funkce operačního systému, která umožňuje procesům a jednotkovým procesům databázového serveru sdílet data prostřednictvím sdíleného přístupu ke společným oblastem paměti. Databázový server používá sdílenou paměť k těmto účelům:

- Lepší využití paměti a snížení počtu diskových vstupně-výstupních operací.
- Vysokorychlostní komunikace mezi procesy.

Sdílená paměť umožňuje databázovému serveru snížit celkový objem využitě paměti, protože účastníci se procesy, v tomto případě virtuální procesory, nemusejí udržovat soukromé kopie dat uložených ve sdílené paměti.

Sdílená paměť omezuje počet diskových vstupně-výstupních operací, protože vyrovnávací paměti, které jsou spravované jako společná oblast paměti, jsou vyprazdňovány společně v rámci celého databázového serveru, nikoli na úrovni jednotlivých procesů. Dále virtuální procesor také často nemusí načítat data z disku, protože data jsou již uložena ve sdílené paměti v důsledku dřívější operace čtení. Zmenšením počtu diskových vstupně-výstupních operací se zmenšuje také celková doba provádění požadavků.

Sdílená paměť představuje nejrychlejší metodu meziprocesové komunikace, protože zapisování a čtení zpráv probíhá rychlostí paměťových přenosů.

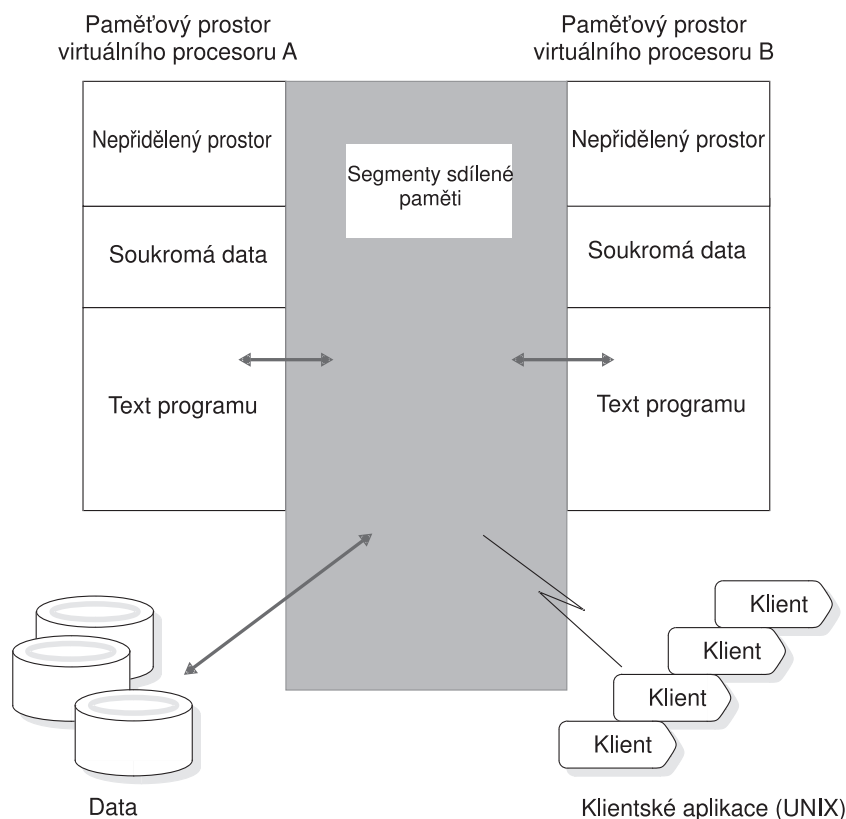
---

## Použití sdílené paměti

Databázový server používá sdílenou paměť k těmto účelům:

- Sdílení dat mezi virtuálními procesory a obslužnými programy.
- Rychlá komunikace s místními klientskými aplikacemi, které používají komunikaci IPC.

Obrázek Obrázek 7-1 znázorňuje schéma sdílené paměti.



Obrázek 7-1. Jak databázový server používá sdílenou paměť

## Přidělování sdílené paměti

Databázový server vytváří následující bloky sdílené paměti:

- *rezidentní část*
- *virtuální část*

### Jen pro UNIX

- Část vyhrazenou pro zprávy a komunikaci IPC.  
Pokud soubor **sqlhosts** určuje komunikaci prostřednictvím sdílené paměti, přidělí databázový server paměť komunikační části sdílené paměti.

### Konec Jen pro UNIX

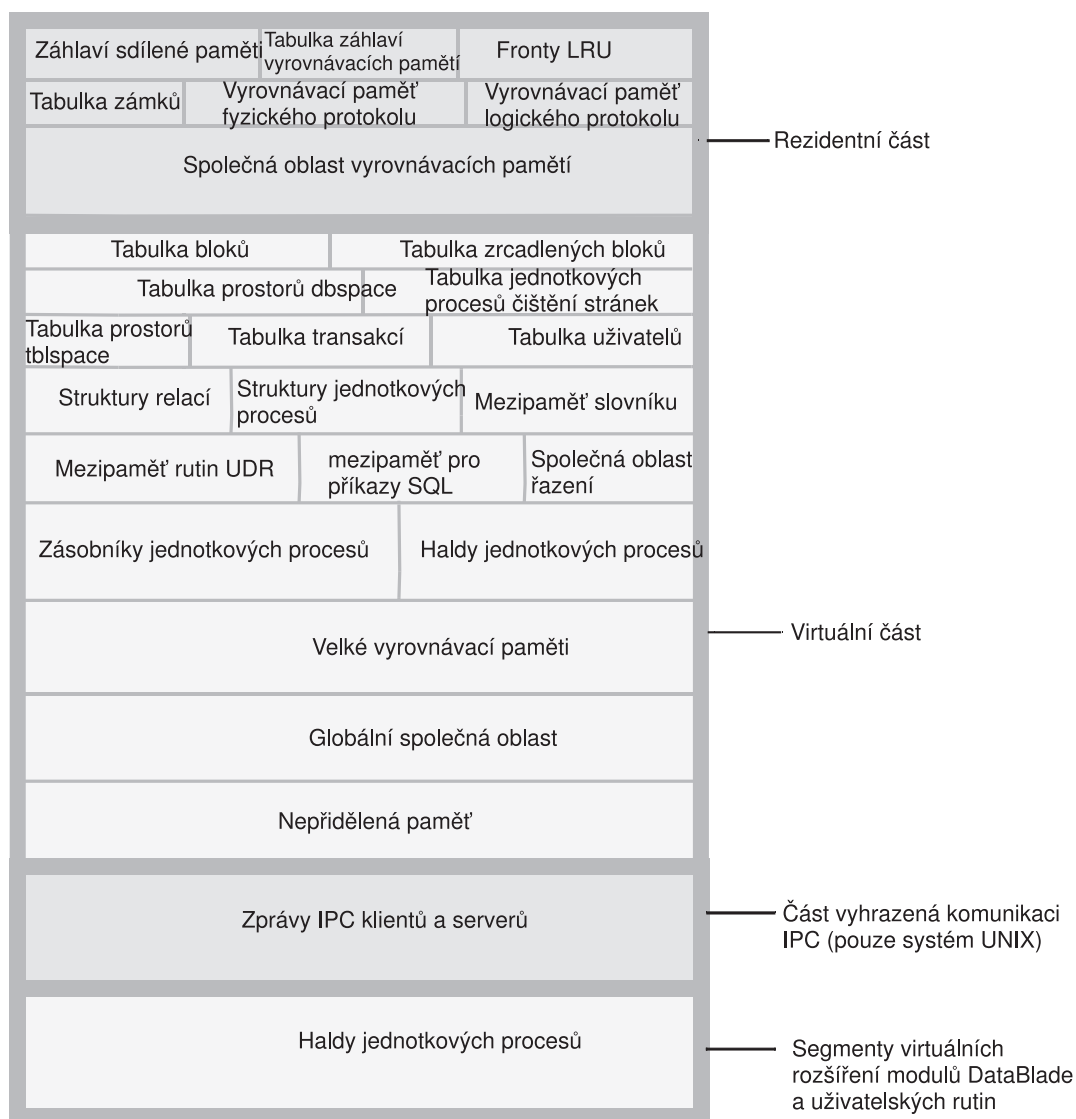
- Část *virtuálních rozšíření*.

Databázový server podle potřeby přidává segmenty operačního systému k virtuálním částem a k částem virtuálních rozšíření sdílené paměti.

Další informace o nastaveních sdílené paměti pro konkrétní platformu naleznete v souboru Poznámky k počítači. Obrázek Obrázek 7-2 na stránce 7-4 znázorňuje obsah všech částí sdílené paměti.

Všechny virtuální procesory databázového serveru mají zpřístupněné tytéž segmenty sdílené paměti. Každý virtuální procesor si udržuje přehled o své činnosti tím, že spravuje vlastní sadu ukazatelů, zámek a zámek latch. Virtuální procesory se připojí ke sdílené paměti, když

nastavíte databázový server z režimu offline na režim klidový, administrativní nebo online. Databázový server používá zámky k řízení souběžného přístupu několika jednotkových procesů ke zdrojům sdílené paměti.



Obrázek 7-2. Obsah sdílené paměti databázového serveru

## Velikost sdílené paměti

Každá část sdílené paměti databázového serveru se skládá z jednoho nebo více segmentů paměti operačního systému, které se skládají se série bloků o velikosti 4 kB spravovaných pomocí bitové mapy.

Ve výpisu obslužného programu **onstat** je v řádku záhlaví uvedena velikost sdílené paměti v kilobajtech. Také můžete pomocí příkazu **onstat -g seg** sledovat, kolik paměti databázový server přiděluje jednotlivým částem sdílené paměti. Další informace týkající se použití obslužného programu **onstat** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Nastavením parametru **SHMTOTAL** v konfiguračním souboru **ONCONFIG** můžete omezit režijní požadavky na paměť, které databázový server klade na počítač nebo uzel. Parametr

SHMTOTAL určuje celkovou velikost sdílené paměti, kterou databázový server může přidělit. Některé operace se však nemusejí zdařit, pokud databázový server potřebuje více paměti, než kolik určuje parametr SHMTOTAL. Pokud se takový stav vyskytne, zobrazí databázový server následující zprávu v protokolu zpráv:

```
size of resident + virtual segments x + y > z
total allowed by configuration parameter SHMTOTAL
```

Dále databázový server vrátí chybovou zprávu aplikaci zodpovědné za spuštění operace, která tento stav způsobila. Pokud databázový server například potřebuje k operaci jako je vytvoření indexu nebo spojení typu hash více paměti, než určuje parametr SHMTOTAL vrátí aplikaci chybovou zprávu podobnou jedné z následujících:

```
-567 Cannot write sorted rows.
-116 ISAM error: cannot allocate memory.
```

Jakmile databázový server odešle tyto zprávy, odvolá všechny částečné změny způsobené dotazem.

Interní operace, například čištění stránek nebo kontrolní body, mohou také způsobit, že databázový server překročí hranici stanovenou parametrem SHMTOTAL. Pokud k této situaci dojde, odešle databázový server zprávu do protokolu zpráv. Předpokládejme například, že se databázový server pokusí přidělit další paměť procesům čištění stránek a přidělení se nezdaří. V důsledku toho odešle databázový server zprávu do protokolu zpráv.

```
17:19:13 Assert Failed: WARNING! No memory available for page cleaners
17:19:13 Who: Thread(11, flush_sub(0), 9a8444, 1)
17:19:13 Results: Database server may be unable to complete a checkpoint
17:19:13 Action: Make more virtual memory available to database server
17:19:13 See Also: /tmp/af.c4
```

Poté, co databázový server informuje o nezdařeném požadavku na přidělení další paměti, odvolá transakce, které způsobily překročení limitu SHMTOTAL. Po odvolání transakcí již nebudou operace selhávat z důvodu nedostatku paměti a databázový server bude zpracovávat další transakce obvyklým způsobem.

## Jaké akce provést, pokud je překročena hodnota parametru SHMTOTAL

Pokud databázový server potřebuje více paměti, než povoluje hodnota parametru SHMTOTAL, bude vzniklý stav dočasný. Jeho příčinou může být například špičkový vzrůst aktivity, která tak překročí běžné zatížení zpracováním transakcí. Selhat by měly pouze operace, které způsobily dočasný nedostatek paměti databázového serveru. Další operace budou zpracovány obvyklým způsobem.

Pokud zprávy oznamují, že databázový server pravidelně požaduje více paměti, než kolik povoluje parametr SHMTOTAL, nebyl databázový server správně nakonfigurován. Možné řešení je snížení hodnoty parametru DS\_TOTAL\_MEMORY nebo volby **buffers** v konfiguračním parametru BUFFERPOOL, dále můžete také zvýšit hodnotu parametru SHMTOTAL.

---

## Procesy, které se připojují ke sdílené paměti

Ke sdílené paměti databázového serveru se připojují následující procesy:

Jen pro UNIX

- Procesy klientských aplikací, které komunikují s databázovým serverem prostřednictvím komunikační části sdílené paměti (**ipcshm**).

### Konec Jen pro UNIX

- Virtuální procesory databázového serveru.
- Obslužné programy databázového serveru.

Následující části popisují, jak se jednotlivé typy procesů připojují ke sdílené paměti databázového serveru.

## Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)

Procesy klientských aplikací, které komunikují s databázovým serverem prostřednictvím sdílené paměti (parametr `nettype ipcshm`) se transparentně připojují ke komunikační části sdílené paměti. Funkce systémových knihoven, které jsou automaticky kompilovány do aplikace, umožňují aplikaci připojit se ke komunikační části sdílené paměti. Informace o způsobu určení připojení prostřednictvím sdílené paměti naleznete v částech Kapitola 3, “Komunikace mezi klientem a serverem”, na stránce 3-1 a “Síťové virtuální procesory” na stránce 5-22.

Pokud není nastavena proměnná prostředí **INFORMIXSHMBASE**, připojuje se aplikace ke komunikační části paměti na adrese specifické pro platformu. Pokud se klientská aplikace připojuje k ostatním segmentům sdílené paměti (ne ke sdílené paměti databázového serveru), uživatel může nastavit proměnnou prostředí **INFORMIXSHMBASE** k určení adresy, na které se má klientská aplikace připojovat k segmentům komunikačního prostředí sdílené paměti databázového serveru. Pokud určíte adresu, od které mají být adresovány komunikační segmenty sdílené paměti, můžete tím zabránit kolizím databázového serveru s jinými segmenty sdílené paměti, které aplikace používá. Informace o způsobu nastavení proměnné prostředí **INFORMIXSHMBASE** naleznete v příručce *IBM Informix Guide to SQL: Reference*.

## Jak se ke sdílené paměti připojují obslužné programy

Obslužné programy databázového serveru, například **onstat**, **onmode** a **ontape**, se ke sdílené paměti připojují prostřednictvím jednoho z následujících souborů.

| Operační systém | Soubor                                             |
|-----------------|----------------------------------------------------|
| UNIX            | <code>\$INFORMIXDIR/etc/infos.název_serveru</code> |
| Windows         | <code>%INFORMIXDIR%\etc\infos.název_serveru</code> |

Proměnná **název\_serveru** je hodnota konfiguračního parametru **DBSERVERNAME** v souboru **ONCONFIG**. Obslužné programy získávají část **název\_serveru** názvu souboru z proměnné prostředí **INFORMIXSERVER**.

Proces **oninit** při spouštění databázového serveru načte soubor **ONCONFIG** a vytvoří soubor **.infos.název\_serveru**. Soubor je odstraněn při ukončení databázového serveru.

## Jak se ke sdílené paměti připojují virtuální procesory

Virtuální procesory databázového serveru se během nastavení připojují ke sdílené paměti. V průběhu tohoto procesu musí databázový server zajistit splnění následujících dvou požadavků:

- Server musí zajistit, aby všechny virtuální procesory mohly nalézt tytéž segmenty sdílené paměti a mohly k nim přistupovat.



- Server musí zajistit, aby segmenty sdílené paměti byly ve fyzické paměti uloženy na jiném místě, než segmenty sdílené paměti přiřazené jiným instancím databázového serveru, pokud takové instance v počítači existují.

Databázový server používá ke splnění těchto požadavků následující dva konfigurační parametry: SERVERNUM a SHMBASE.

Když se virtuální procesor připojuje ke sdílené paměti, provádí následující důležité kroky:

- Načte parametr SERVERNUM ze souboru ONCONFIG.
- Pomocí parametru SERVERNUM vypočte hodnotu klíče sdílené paměti.
- Požádá o segment sdílené paměti s použitím klíče sdílené paměti.  
Operační systém vrátí identifikátor sdílené paměti přiřazený prvnímu segmentu sdílené paměti.
- Požádá operační systém, aby připojil první segment sdílené paměti k adresovému prostoru procesu na adrese SHMBASE.
- Podle potřeby připojí další segmenty sdílené paměti tak, aby navazovaly na první segment.

Následující části popisují, jak databázový server používá hodnoty konfiguračních parametrů SERVERNUM a SHMBASE při připojování segmentů sdílené paměti.

### Získání hodnot klíčů segmentů sdílené paměti

Hodnoty konfiguračního parametru SERVERNUM a interně vypočítaného čísla *shmkey* určují hodnotu jedinečného klíče každého segmentu sdílené paměti.

Pokud chcete zobrazit hodnoty klíčů segmentů sdílené paměti, spusťte příkaz **onstat -g seg**. Další informace naleznete v částech týkajících se parametru SHMADD a společné oblasti vyrovnávací paměti v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Když virtuální procesor žádá operační systém o připojení prvního segmentu sdílené paměti, poskytne systému hodnotu jedinečného klíče, který slouží jako identifikátor segmentu. Operační systém vrátí *identifikátor segmentu sdílené paměti* přidružený k hodnotě klíče. Virtuální procesor pomocí tohoto identifikátoru požádá operační systém o připojení segmentu sdílené paměti k adresovému prostoru virtuálního procesoru.

### Určení místa připojení prvního segmentu sdílené paměti

Parametr SHMBASE souboru ONCONFIG určuje virtuální adresu, ke které každý virtuální procesor připojí první (nebo základní) segment sdílené paměti. Všechny virtuální procesory připojují první segment sdílené paměti k téže virtuální adrese. To umožňuje všem virtuálním procesorům v rámci jediné instance databázového serveru odkazovat na totožná umístění ve sdílené paměti, aniž by při tom musely přepočítávat adresy sdílené paměti. Všechny adresy sdílené paměti se v rámci instance databázového serveru počítají relativně se základem SHMBASE.

**Upozornění:** Neměňte hodnotu parametru SHMBASE.

Databázový server je citlivý na změnu hodnoty parametru SHMBASE, a to z následujících důvodů:

- Konkrétní hodnota parametru SHMBASE je závislá na platformě a na tom, zda je procesor 32bitový nebo 64bitový. Hodnota parametru nemůže být SHMBASE libovolné číslo a je volena tak, aby segmenty sdílené paměti byly chráněny, pokud virtuální procesor dynamicky získává od operačního systému další paměťový prostor.
- Různé operační systémy přidělují další paměť na různých adresách. Některé architektury rozšiřují nejvyšší virtuální adresu datového segmentu virtuálního procesoru, aby mohly poskytnout prostor pro další segment. V takovém případě může datový segment přetéct do

segmentu sdílené paměti.

#### Jen pro UNIX

- Některé verze systému UNIX vyžadují, aby uživatel nastavil parametr SHMBASE virtuální adresy na hodnotu nula. Adresa s hodnotou nula informuje jádro systému UNIX, že by mělo vybrat nejlepší adresu, na které se mají připojit segmenty sdílené paměti. Ne všechny architektury systému UNIX však tuto volbu podporují. V některých systémech navíc volba adresy prováděná jádrem nemusí být optimální.

#### Konec Jen pro UNIX

Informace o parametru SHMBASE naleznete v poznámkách k počítači dodávaných se serverem Dynamic Server.

### Připojení dalších segmentů sdílené paměti

Každý virtuální procesor musí připojit všechnu sdílenou paměť, kterou databázový server získal od operačního systému. Virtuální procesor po připojení každého segmentu sdílené paměti vypočítá, jak velkou část sdílené paměti již připojil a jak velká část ještě zbývá. Databázový server usnadňuje tento proces tím, že do prvního segmentu sdílené paměti zapisuje záhlaví sdílené paměti. Z šestnácti bajtů záhlaví může virtuální procesor získat následující data:

- Celkovou velikost sdílené paměti databázového serveru.
- Velikost každého segmentu sdílené paměti.

Virtuální procesor připojuje další segmenty sdílené paměti tak, že si je od operačního systému vyžádá podobným způsobem jako v případě prvního segmentu. V případě dalších segmentů však virtuální procesor přičítá číslo 1 k předchozí hodnotě klíče *shmkey*. Virtuální procesor požádá operační systém o připojení segmentu k adrese vypočítané podle následujícího vzorce:  $SHMBASE + (velikost\_segmentu \times počet\ připojených\ segmentů)$

Virtuální procesor opakuje tento proces, dokud nepřipojí celou sdílenou paměť.

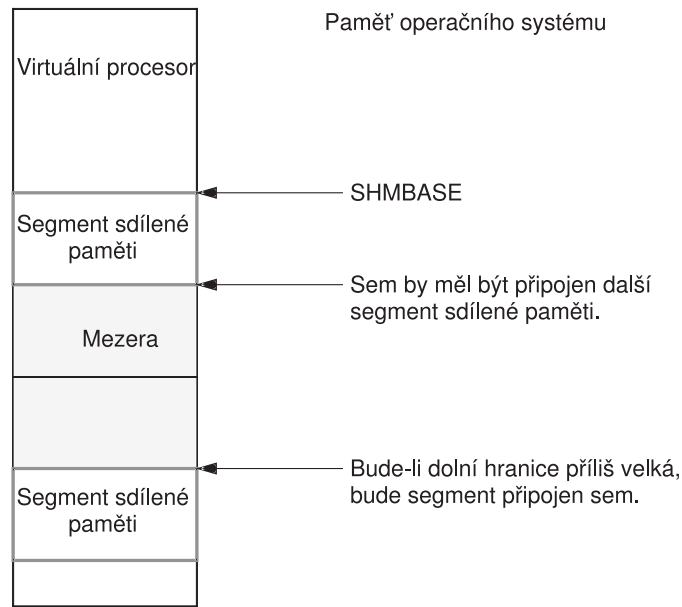
Protože pro počáteční hodnotu klíče platí vztah  $(SERVERNUM * 65536) + shmkey$ , může databázový server požádat až o 65536 segmentů sdílené paměti, než požádá o hodnotu klíče sdílené paměti, kterou už používá jiná instance databázového serveru v téže počítači.

### Definování adresy dolní hranice sdílené paměti

Pokud operační systém používá parametr, který definuje adresu dolní hranice sdílené paměti, a tento parametr je nastaven nesprávně, může se stát, že segmenty sdílené paměti nebude možno připojit souvisle.

Obrázek Obrázek 7-3 znázorňuje tento problém. Pokud je adresa dolní hranice menší než koncová adresa předchozího segmentu plus velikost aktuálního segmentu, připojí operační systém aktuální segment za konec předchozího segmentu. Tím dojde ke vzniku mezery mezi dvěma segmenty. Protože sdílená paměť musí být připojena k virtuálnímu procesoru tak, aby byla pro aplikaci souvislá, představuje tato mezera potíže. Pokud tato situace nastane, databázový server přijme chybové zprávy.

Abyste napravili tento problém, zkontrolujte parametr operačního systému, který určuje adresu dolní hranice, nebo změňte konfiguraci jádra systému tak, aby umožňovalo používat větší segmenty sdílené paměti.



Obrázek 7-3. Přehled adresy dolní hranice sdílené paměti

## Rezidentní segmenty sdílené paměti

Operační systém za obvyklých okolností při přepínání procesů odkládá části paměti na disk. Pokud je však část paměti označena jako *rezidentní*, není odkládána na disk. Pokud jsou často používaná data rezidentní v paměti, dojde ke zlepšení výkonu, protože se sníží počet vstupně-výstupních operací disku, které by byly jinak zapotřebí ke zpřístupnění dat.

Databázový server požádá operační systém o zachování virtuálních částí ve fyzické paměti, pokud nastane jedna z následujících podmínek:

- Operační systém podporuje rezidenci sdílené paměti.
- Parametr RESIDENT v souboru ONCONFIG je nastaven na hodnotu -1 nebo na hodnotu větší než 0.

**Upozornění:** Pokud posuzujete, zda nastavit parametr RESIDENT na hodnotu -1, zvažte využití sdílené paměti. Pokud zamknete veškerou sdílenou paměť, aby ji mohl používat databázový server Informix, může to nepříznivě ovlivnit výkon ostatních aplikací spuštěných v počítači, pokud takové aplikace existují.

Další informace o konfiguračním parametru RESIDENT naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Rezidentní část sdílené paměti

V rezidentní části sdílené paměti databázového serveru jsou uloženy následující datové struktury, které za provozu databázového serveru nemění velikost:

- záhlaví sdílené paměti
- společná oblast vyrovnávací paměti
- vyrovnávací paměť logického protokolu
- vyrovnávací paměť fyzického protokolu
- tabulka zámeků

## Záhlaví sdílené paměti

Záhlaví sdílené paměti obsahuje popis všech dalších struktur uložených ve sdílené paměti, včetně interních tabulek a společné oblasti vyrovnávací paměti.

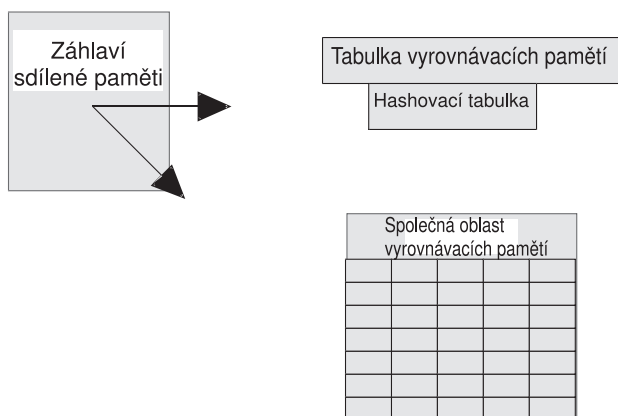
Záhlaví sdílené paměti obsahuje také ukazatele na umístění těchto struktur. Pokud se virtuální procesor poprvé připojuje ke sdílené paměti, načte informace záhlaví sdílené paměti, které ho nasměrují ke všem dalším strukturám.

Velikost záhlaví sdílené paměti je přibližně 200 kB. Přesná velikost je však závislá na použité počítačové platformě. Velikost záhlaví nelze ladit.

## Společná oblast vyrovnávacích pamětí ve sdílené paměti

Společná oblast vyrovnávacích pamětí v rezidentní části sdílené paměti obsahuje vyrovnávací paměti, ve kterých jsou uloženy stránky prostorů dbspace načtené z disku. Společná oblast vyrovnávacích pamětí představuje největší přidělenou oblast v rezidentní části sdílené paměti.

Obrázek Obrázek 7-4 znázorňuje záhlaví sdílené paměti a společnou oblast vyrovnávacích pamětí.



Obrázek 7-4. Společná oblast vyrovnávacích pamětí ve sdílené paměti

K určení informací o společné oblasti vyrovnávacích pamětí, včetně počtu vyrovnávacích pamětí, se používá konfigurační parametr BUFFERPOOL. Aby byl přidělen vhodný počet vyrovnávacích pamětí, přidejte alespoň čtyři vyrovnávací paměti na každého uživatele. Pro více než 500 uživatelů je požadováno nejméně 2000 vyrovnávacích pamětí. Příliš malý počet vyrovnávacích pamětí může mít závažný dopad na výkon, monitorujte proto databázový server a laděním volby buffers určete přijatelnou hodnotu. Další informace o ladění počtu vyrovnávacích pamětí naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Pokud neexistuje společná oblast vyrovnávacích pamětí pro stránky jiné než výchozí velikosti, vytvoří databázový server vyrovnávací paměť velkých stránek automaticky.

Pokud vytváříte prostor dbspace s velikostí stránky jinou než výchozí, musí pro prostor dbspace existovat odpovídající společná oblast vyrovnávacích pamětí. Pokud například vytvoříte prostor dbspace s velikostí stránky 6 kB, je třeba vytvořit také společnou oblast vyrovnávacích pamětí s velikostí stránky 6 kB.

Automatické ladění fronty LRU (nejdéle nepoužívaná) ovlivňuje všechny společné oblasti vyrovnávací paměti a upravuje hodnoty `lru_min_dirty` a `lru_max_dirty` v konfiguračním parametru BUFFERPOOL.

Další informace o nastavení konfiguračního parametru BUFFERPOOL naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Stav vyrovnávacích pamětí je zaznamenáván v tabulce vyrovnávacích pamětí. V rámci sdílené paměti jsou vyrovnávací paměti organizovány do front FIFO/LRU vyrovnávací paměti. Získávání vyrovnávacích pamětí je řízeno prostřednictvím zámek latch nazývaných *objekty mutex* a informací o přístupu k zámek.

Popis činnosti front LRU naleznete v části “Fronty FIFO/LRU” na stránce 7-22. Popis objektů mutex naleznete v části “Objekty mutex” na stránce 5-12.

## Přetečení vyrovnávacích pamětí do virtuální části

Protože maximální počet vyrovnávacích pamětí je při 64bitovém adresování roven hodnotě  $2^{31}-1$ , nemusí být rezidentní část sdílené paměti schopna pojmout všechny vyrovnávací paměti tvořící velkou společnou oblast vyrovnávacích pamětí. V takovém případě může některé vyrovnávací paměti obsahovat virtuální část sdílené paměti databázového serveru.

## Velikost vyrovnávací paměti

Každá vyrovnávací paměť má velikost jedné stránky databázového serveru.

Databázový server obecně provádí vstupně-výstupní operace v jednotkách celých stránek, tedy o velikosti vyrovnávací paměti. Výjimkou jsou operace vstupu - výstupu prováděné s obsahem velkých vyrovnávacích pamětí, vyrovnávacích pamětí prostorů blobspace a vyrovnávacích pamětí odlehčeného vstupu - výstupu. (Další informace naleznete v částech “Velké vyrovnávací paměti” na stránce 7-18 a “Vytvoření vyrovnávacích pamětí stránek blobpage” na stránce 7-31.) Informace o tom, kdy použít soukromé vyrovnávací paměti, naleznete v částech příručky *Příručka výkonosti serveru IBM Informix Dynamic Server* věnovaných operacím odlehčeného vstupu - výstupu.

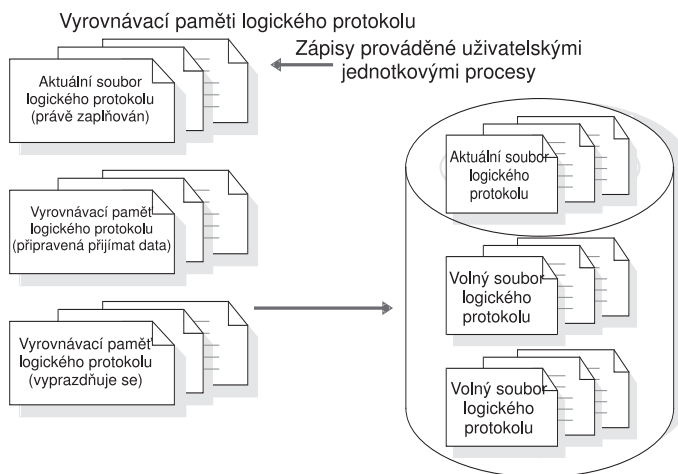
Informace o vyrovnávacích pamětech zobrazuje příkaz **onstat -b**. Informace o příkazu **onstat** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Vyrovnávací paměť logického protokolu

Databázový server používá logický protokol k ukládání záznamů o změnách dat databázového serveru od poslední zálohy prostorů dbspace. Logický protokol ukládá záznamy, které představují logické jednotky práce databázového serveru. Logický protokol ukládá mimo jiné následujících pět typů záznamů protokolu:

- Příkazy jazyka SQL pro definici dat spouštěné ve všech databázích.
- Příkazy jazyka SQL pro manipulaci s daty spouštěné v databázích, které byly vytvořeny jako protokolované.
- Záznam změny stavu protokolování databáze.
- Záznam kontrolního bodu
- Záznam o změně konfigurace.

Databázový server používá vždy pouze jedinou vyrovnávací paměť logického protokolu. Tato vyrovnávací paměť je aktuální vyrovnávací paměť logického protokolu. Než databázový server vyprázdní aktuální vyrovnávací paměť logického protokolu na disk, nastaví další vyrovnávací paměť logického protokolu jako aktuální, aby do ní mohl zapisovat v průběhu vyprazdňování předchozí vyrovnávací paměti. Pokud bude druhá vyrovnávací paměť logického protokolu zaplněna před dokončením vyprazdňování první vyrovnávací paměti, bude jako aktuální nastavena třetí vyrovnávací paměť logického protokolu. Tento proces znázorňuje Obrázek 7-5.



Obrázek 7-5. Vyrovnávací paměť logického protokolu a její vztah k souborům logického protokolu na disku

Popis způsobu, jakým databázový server vyprazdňuje vyrovnávací paměť logického protokolu, naleznete v části “Vyprazdňování vyrovnávací paměti logického protokolu” na stránce 7-29.

Konfigurační parametr LOGBUFF určuje velikost vyrovnávacích pamětí logických protokolů. Malé vyrovnávací paměti mohou působit problémy, pokud ukládáte záznamy větší, než je velikost vyrovnávacích pamětí (například data typu TEXT nebo BYTE v prostorech dbspace). Doporučená hodnota velikosti vyrovnávací paměti logického protokolu je 64 kB. Kdykoli je nastavena menší než doporučená hodnota, databázový server navrhne hodnotu při spuštění. Možné hodnoty, které můžete přiřadit k tomuto konfiguračnímu parametru, naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Další informace o vlivu dat typu TEXT a BYTE na vyrovnávací paměti ve sdílené paměti naleznete v části “Použití vyrovnávací paměti pro velké objekty” na stránce 7-30.

## Vyrovnávací paměť fyzického protokolu

Databázový server používá vyrovnávací paměť fyzického protokolu k ukládání předobrazu některých změněných stránek prostorů dbspace. Předobrazy ve fyzickém protokolu a záznamy logického protokolu umožňují databázovému serveru rychle obnovit konzistenci dat po selhání systému.

Vyrovnávací paměť fyzického protokolu se ve skutečnosti skládá ze dvou vyrovnávacích pamětí. Zdvojená vyrovnávací paměť umožňuje procesu databázového serveru zpracovávat zápisy do aktivní vyrovnávací paměti fyzického protokolu, zatímco je druhá vyrovnávací paměť fyzického protokolu vyprazdňována na disk. Popis způsobu, jakým databázový server vyprazdňuje vyrovnávací paměť fyzického protokolu na disk, naleznete v části “Vyprazdňování vyrovnávací paměti fyzického protokolu” na stránce 7-27. Informace o monitorování souboru fyzického protokolu naleznete v části “Monitorování aktivity fyzického a logického protokolování” na stránce 16-2.

Velikost vyrovnávacích pamětí fyzického protokolu určuje parametr PHYSBUFF v souboru ONCONFIG. Zápis do vyrovnávací paměti fyzického protokolu zapisuje přesně jednu stránku. Pokud není určena velikost vyrovnávací paměti fyzického protokolu beze zbytku dělitelná velikostí stránky, zaokrouhlí databázový server velikost směrem dolů na nejbližší hodnotu beze zbytku dělitelnou velikostí stránky. Přestože některé operace vyžadují předčasné vyprázdnění vyrovnávací paměti, vyprazdňuje obecně databázový server vyrovnávací paměť

do souboru fyzického protokolu, jakmile je vyrovnávací paměť zaplněna. Velikost vyrovnávací paměti proto určuje, jak často musí databázový server vyprazdňovat vyrovnávací paměť na disk.

Výchozí hodnota velikosti vyrovnávací paměti fyzického protokolu je 512 kilobajtů. Pokud se rozhodnete použít menší hodnotu, databázový server zobrazí zprávu, která oznámí, že nemusí být dosažen optimální výkon. Použití fyzického protokolu vyrovnávací paměti menšího než 512 kilobajtů má vliv pouze na výkon, ne na integritu transakcí.

Další informace o tomto konfiguračním parametru naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

## Vyrovňovací paměť replikace High-Availability Data Replication

Replikace dat vyžaduje dvě instance databázového serveru (primární instanci a sekundární instanci), spuštěné ve dvou počítačích. Pokud implementujete v databázovém serveru replikaci dat, ukládá databázový server záznamy logického protokolu do vyrovnávacích pamětí replikace dat, než je odešle sekundárnímu databázového serveru. Velikost vyrovnávací paměti replikace dat se vždy shoduje s velikostí vyrovnávací paměti logického protokolu. Informace o velikosti vyrovnávací paměti logického protokolu naleznete v předcházející části "Vyrovňovací paměť logického protokolu" na stránce 7-11. Další informace o použití vyrovnávací paměti logického protokolu naleznete v části "Způsob činnosti replikace dat" na stránce 19-6.

## Tabulka zámků

K vytvoření zámku dojde tehdy, pokud uživatelský jednotkový proces запиše položku do tabulky zámků. Tabulka zámků představuje společnou oblast dostupných zámků. Jediná transakce může spravovat několik zámků. Vysvětlení zamykání a příkazů jazyka SQL souvisejících se zamykáním naleznete v příručce *IBM Informix Guide to SQL: Tutorial*.

Každý zámek je popsán následujícími informacemi uloženými v tabulce zámků:

- Adresa transakce, které zámek patří.
- Typ zámku (výlučný, aktualizací, sdílený, bajtový nebo pokus o zamknutí).
- Stránka nebo identifikátor rowid jsou zamknuté.
- Prostor tabulky, kde je umístěn zámek.
- Informace o uzamčených bajtech (zámky rozsahu bajtů inteligentních velkých objektů):
  - ID inteligentního velkého objektu.
  - Posun vůči počátku inteligentního velkého objektu, od kterého zámek začíná.
  - Počet uzamčených bajtů počínaje počátkem.

Počáteční velikost tabulky zámků určete pomocí konfiguračního parametru LOCKS. Informace o použití konfiguračního parametru LOCKS k určení počtu zámků relace naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference* a kapitole týkající se efektů konfigurace na využití paměti v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Pokud počet zámků přidělených relacemi přesáhne hodnotu určenou v konfiguračním parametru LOCKS, databázový server může až patnáctkrát zdvojnásobit velikost tabulky zámků. Maximální hodnota konfiguračního parametru LOCKS na 32bitovém serveru je 8 000 000. Maximální hodnota konfiguračního parametru LOCKS na 64bitovém serveru je 500 000 000.

Pomocí parametru DEF\_TABLE\_LOCKMODE nastavte režim zamykání nových tabulek na stránkový nebo řádkový.

Zámky mohou zabránit před čtením dat relací, dokud není souběžná transakce potvrzena nebo odvolána. U databází vytvořených protokolováním transakcí je možné pomocí konfiguračního parametru USELASTCOMMITTED v souboru ONCONFIG určit, zda databázový server používá poslední potvrzenou verzi dat. Poslední potvrzená verze dat je ta verze dat, která existovala předtím, než došlo k jakýmkoli aktualizacím. Hodnota, kterou nastavíte pomocí konfiguračního parametru USELASTCOMMITTED, potlačí úroveň izolace určenou příkazem SET ISOLATION TO COMMITTED READ jazyka SQL. Další informace o použití konfiguračního parametru USELASTCOMMITTED naleznete v kapitole týkající se konfiguračních parametrů v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Další informace o používání a monitorování zámků naleznete v kapitole o zamykání v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server* a v příručce *IBM Informix Guide to SQL: Tutorial*.

---

## Virtuální část sdílené paměti

Virtuální část sdílené paměti může být databázovým serverem zvětšována a také může být operačním systémem odložena na disk. Když je databázový server spuštěný, automaticky připojuje podle potřeby k virtuální části sdílené paměti další segmenty poskytnuté operačním systémem.

### Správa virtuální části sdílené paměti

Databázový server používá *společné oblasti* paměti k zaznamenávání přidělených částí paměti podobného typu a velikosti. Udržování souvisejících přidělených částí paměti ve společné oblasti omezuje fragmentaci paměti. Také umožňuje databázovému serveru uvolnit velkou část přidělené paměti v jediném kroku, namísto toho, aby jednotlivé části tvořící společnou oblast musel uvolňovat postupně.

Všechny relace mají přiděleny jednu nebo více společných oblastí paměti. Pokud databázový server potřebuje paměť, hledá nejprve v určené společné oblasti. Pokud množství paměti ve společné oblasti nedostačuje ke splnění požadavku, přidá databázový server paměť ze společné oblasti systému. Pokud databázový server nenalezne dostatečné množství paměti ve společné oblasti systému, dynamicky přidělí další segmenty k virtuální části sdílené paměti.

Databázový server přiděluje virtuální sdílenou paměť každému ze svých podsystémů (společné oblasti relací, zásobníky, haldy, řídicí bloky, systémový katalog, mezipaměti rutin SPL, mezipaměti příkazů jazyka SQL, společné oblasti řazení a vyrovnávací paměti zpráv) ze společných oblastí, které zaznamenávají volný prostor ve spojovém seznamu. Pokud databázový server přiděluje část paměti, hledá nejprve v seznamu volných oblastí společné paměti *fragment* dostatečné velikosti. Pokud žádný nenalezne, přidá ke společné oblasti nové bloky virtuální části sdílené paměti. Po uvolnění bude paměť navracena zpět do společné oblasti jako volný fragment a zůstane ve společné oblasti, dokud nebude společná oblast odstraněna. Pokud například databázový server spouští relaci klientské aplikace, přidělí paměť pro společnou oblast relace. Po ukončení relace vrátí databázový server přidělenou paměť jako volné fragmenty.

### Velikost virtuální části sdílené paměti

Počáteční velikost virtuální části sdílené paměti určete pomocí konfiguračního parametru SHMVIRTSIZE v souboru ONCONFIG. Pokud chcete určit velikost segmentů, které jsou k virtuální paměti přidávány později, nastavte konfigurační parametr SHMADD nebo EXTSHMADD.

Množství paměti dostupné pro dotazy PDQ nastavte pomocí parametru DS\_TOTAL\_MEMORY.



Pokud chcete zvětšit množství paměti dostupné pro dotazy jiné než PDQ a priorita PDQ je nastavena na hodnotu 0 (nula), můžete ke změně množství paměti použít kteroukoli z následujících voleb:

- konfigurační parametr DS\_NONPDQ\_QUERY\_MEM
- příkaz **onmode -wm** nebo **onmode -wf**
- volbu **Non PDQ Query Memory** v nabídce **pdQ** programu ON-Monitor

Pokud například používáte obslužný program **onmode**, určete hodnotu způsobem znázorněným v následujícím příkladu:

```
onmode -wf DS_NONPDQ_QUERY_MEM=500
```

Minimální hodnota parametru DS\_NONPDQ\_QUERY\_MEM je 128 kB. Maximální podporovaná hodnota je 25 % hodnoty parametru DS\_TOTAL\_MEMORY.

Další informace o konfiguračních parametrech SHMVIRTSIZE, SHMADD, EXTSHMADD, DS\_TOTAL\_MEMORY, DS\_TOTAL\_SIZE a DS\_NONPDQ\_QUERY\_MEM naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server* a *IBM Informix Dynamic Server Administrator's Reference*. Další informace naleznete také v části “Přidání segmentu k virtuální části sdílené paměti” na stránce 8-7.

## Součásti virtuální části sdílené paměti

Ve virtuální části sdílené paměti jsou uložena následující data:

- interní tabulky
- velké vyrovnávací paměti
- data relací
- data jednotkových procesů (zásobníky a haldy)
- mezipaměť distribuce dat
- mezipaměť slovníku
- mezipaměť rutin SPL
- mezipaměť pro příkazy SQL
- společná oblast řazení
- globální společná oblast

### Interní tabulky sdílené paměti

Sdílená paměť databázového serveru obsahuje sedm interních tabulek, které zaznamenávají informace o zdrojích sdílené paměti. Interní tabulky sdílené paměti jsou následující tabulky:

- tabulka vyrovnávacích pamětí
- tabulka bloků
- tabulka prostorů dbspace
- tabulka jednotkových procesů čištění stránek
- tabulka prostorů tbspace
- tabulka transakcí
- tabulka uživatelů

**Tabulka vyrovnávacích pamětí:** Tabulka vyrovnávacích pamětí zaznamenává adresy a stavy jednotlivých vyrovnávacích pamětí ve společné oblasti sdílené paměti. Pokud je vyrovnávací paměť použita, obsahuje obraz stránky indexu nebo dat načtené z disku. Další informace o účelu a obsahu diskových stránek naleznete v části “Stránky” na stránce 9-5.

Každá vyrovnávací paměť v tabulce vyrovnávacích pamětí obsahuje následující řídicí informace potřebné ke správě vyrovnávacích pamětí:

- Stav vyrovnávací paměti.  
Stav vyrovnávací paměti může mít hodnotu prázdná, změněno a nezměněno. Nezměněná vyrovnávací paměť obsahuje data, je však možné je přepsat. Změněná (neaktualizovaná) paměť obsahuje data, která je před přepsáním zapotřebí zapsat na disk.
- Aktuální úroveň přístupu zámku.  
Vyrovnávacím pamětem jsou přidělovány úrovně přístupu zámku v závislosti na typu operace, kterou provádí uživatelský jednotkový proces. Databázový server podporuje dvě úrovně přístupu zámku: sdílenou a výlučnou.
- Jednotkové procesy čekající na vyrovnávací paměť.  
Záhlaví každé vyrovnávací paměti udržuje seznam jednotkových procesů, které čekají na vyrovnávací paměť a úroveň přístupu zámku požadovanou každým z čekajících jednotkových procesů.

Pro každou vyrovnávací paměť databázového serveru existuje v tabulce vyrovnávacích pamětí jedna položka.

Informace o vyrovnávacích pamětech databázového serveru naleznete v části “Rezidentní část sdílené paměti” na stránce 7-9. Informace o tom, jak monitorovat vyrovnávací paměti, naleznete v části “Monitorování vyrovnávacích pamětí” na stránce 8-8.

Databázový server určuje počet položek hashovací tabulky vyrovnávacích pamětí v závislosti na počtu přidělených vyrovnávacích pamětí. Maximální počet hodnot hashovací tabulky je roven největší mocnině čísla 2, která je menší než hodnota volby **buffers** určené jedním z polí konfiguračního parametru BUFFERPOOL.

**Tabulka bloků:** Tabulka bloků zaznamenává všechny bloky v databázovém serveru. Pokud bylo povoleno zrcadlení, je při vytvoření sdílené paměti vytvořena také odpovídající tabulka zrcadlených bloků. Tabulka zrcadlených bloků zaznamenává všechny zrcadlené bloky.

Tabulka bloků ve sdílené paměti obsahuje informace, které umožňují databázovému serveru nalézt bloky na disku. Mezi tyto informace patří číslo počátečního bloku a číslo následujícího bloku v prostoru dbspace. Dále stav bloků popisují příznaky: zrcadlený nebo primární, režim offline, režim online nebo režim obnovy a příznak, zda je blok částí prostoru blobspace. Informace o monitorování bloků naleznete v části “Monitorování bloků” na stránce 10-34.

Maximální počet položek v tabulce bloků může být omezen maximálním počtem souborových deskriptorů, které operační systém může přidělit jedinému procesu. Maximální počet souborových deskriptorů přidělených jedinému procesu můžete obvykle určit pomocí konfiguračního parametru jádra operačního systému. Podrobnosti naleznete v příručkách k použitému operačnímu systému.

**Tabulka prostorů dbspace:** Tabulka prostorů dbspace zaznamenává paměťové prostory databázového serveru. Tabulka prostorů dbspace obsahuje následující informace o každém prostoru dbspace:

- Číslo prostoru dbspace.
- Název a vlastník prostoru dbspace.
- Stav zrcadlení prostoru dbspace (zrcadlený nebo nezrcadlený).
- Datum a čas vytvoření prostoru dbspace.

Pokud se jedná o paměťový prostor blobspace, označují příznaky typ média, na kterém se prostor blobspace nachází (magnetické, vyměnitelné nebo optické). Pokud se jedná o

paměťový prostor sbspace, obsahuje interní tabulky, které zaznamenávají metadata inteligentních velkých objektů a velkých souvislých bloků stránek obsahujících uživatelská data.

Informace o monitorování prostorů dbspace naleznete v části “Monitorování využití disku” na stránce 10-34.

**Tabulka jednotkových procesů čištění stránek:** Tabulka jednotkových procesů čištění stránek zaznamenává stav a umístění všech jednotkových procesů čištění stránek. Počet jednotkových procesů čištění stránek je určen konfiguračním parametrem CLEANERS v souboru ONCONFIG. Pokyny k určení správného počtu jednotkových procesů čištění stránek naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Tabulka jednotkových procesů čištění stránek obsahuje vždy 128 položek, nezávisle na počtu jednotkových procesů čištění stránek určených konfiguračním parametrem CLEANERS v souboru ONCONFIG.

Informace o sledování aktivity jednotkových procesů čištění stránky naleznete u volby **onstat -F** v příručce *IBM Informix Dynamic Server Administrator's Reference*.

**Tabulka prostorů tblspace:** Tabulka prostorů tblspace zaznamenává všechny aktivní prostory tblspace v instanci databázového serveru. Aktivní prostor tblspace je prostor tblspace aktuálně používaný databázovou relací. Každá aktivní tabulka představuje jednu položku v tabulce prostorů tblspace. Mezi aktivní prostory tblspace patří databázové tabulky, dočasné tabulky a interní řídicí tabulky, například tabulky systémového katalogu. Každá položka tabulky prostorů tblspace obsahuje záhlaví informací o prostoru tblspace, název prostoru tblspace a ukazatele na prostor tblspace **tblspace** v prostorech dbspace na disku. (Tabulka aktivních prostorů tblspace ve sdílené paměti není totožná s prostorem tblspace **tblspace**.) Informace o monitorování prostorů tblspace naleznete v části “Monitorování prostorů tblspace a oblastí” na stránce 10-38.

Databázový server spravuje pro každý prostor dbspace jednu tabulku prostorů tblspace.

**Tabulka transakcí:** Tabulka transakcí zaznamenává všechny transakce zpracovávané databázovým serverem.

Zaznamenávané informace odvozené od tabulky transakcí se zobrazují ve výstupu příkazu **onstat -x**. Příklad výstupu zobrazovaného příkazem **onstat -x** naleznete v části příručky *Příručka výkonnosti serveru IBM Informix Dynamic Server* věnované monitorování transakcí.

Databázový server v závislosti na počtu aktuálních transakcí automaticky zvětšuje počet položek tabulky transakcí, a to až do maximální hodnoty 32767.

Další informace o transakcích a příkazech jazyka SQL, které se používají s transakcemi, naleznete v příručkách *IBM Informix Guide to SQL: Tutorial*, *IBM Informix Guide to SQL: Reference* a *IBM Informix Guide to SQL: Syntax*.

#### Jen pro UNIX

Tabulka transakcí také určitým způsobem podporuje prostředí X/Open. Podpora prostředí X/Open vyžaduje produkt TP/XA. Popis transakcí prováděných v tomto prostředí naleznete v příručce *IBM Informix TP/XA Programmer's Manual*.

#### Konec Jen pro UNIX

**Tabulka uživatelů:** Tabulka uživatelů zaznamenává všechny uživatelské a systémové jednotkové procesy. Každé klientské relaci je přidružen jeden primární jednotkový proces a nula nebo více sekundárních jednotkových procesů v závislosti na určené míře paralelního zpracování. Mezi systémové jednotkové procesy patří jednotkový proces, který monitoruje a řídí kontrolní body, jednotkový proces zpracovávající příkazy **onmode**, jednotkové procesy prohledávání B-stromu a jednotkové procesy čištění stránek.

Databázový server zvětšuje počet položek tabulky uživatelů podle potřeby. Uživatelské jednotkové procesy můžete monitorovat pomocí příkazu **onstat -u**.

## Velké vyrovnávací paměti

Velká vyrovnávací paměť je vyrovnávací paměť, která se skládá z několika stránek. Skutečný počet stránek závisí na platformě. Databázový server přiděluje velké vyrovnávací paměti, aby dosáhl vyššího výkonu při zapisování a čtení velkých objemů dat.

Databázový server používá velkou vyrovnávací paměť, kdykoli zapisuje na disk několik fyzicky sousedících stránek. Databázový server se například pokouší používat velké vyrovnávací paměti k provádění sekvenčních operací čtení (odlehčeného prohledávání) nebo k načítání jednoduchých velkých objektů uložených v prostorech dbspace.

Uživatelé nemohou používání velkých vyrovnávacích pamětí ovlivnit. Pokud databázový server použije lehká prohledávání, přidělí velké vyrovnávací paměti ve sdílené paměti.

Informace o monitorování velkých vyrovnávacích pamětí pomocí příkazu **onstat** naleznete v kapitole o vlivech konfigurace serveru na aktivitu vstupně-výstupní operací v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

## Data relací

Když klientská aplikace požádá o připojení k databázovému serveru, zahájí databázový server *relaci* s klientem a vytvoří ve sdílené paměti datovou strukturu zvanou *řídící blok relace*. Řídící blok relace obsahuje ID relace, ID uživatele, ID procesu klienta, název hostitelského počítače a různé stavové příznaky.

Databázový server přiděluje paměť pro data relace podle potřeby.

## Data jednotkových procesů

Když se klient připojuje k databázovému serveru, spustí databázový server kromě nové relace také primární jednotkový proces relace a vytvoří pro tento jednotkový proces *řídící blok jednotkového procesu*.

Databázový server také spouští vlastní interní jednotkové procesy a vytváří pro ně řídicí bloky jednotkových procesů. Když databázový server přepíná z jednoho spuštěného jednotkového procesu na jiný jednotkový proces (přepíná kontexty), ukládá informace o jednotkovém procesu - například obsahy registrů, čítač programu (adresu příští instrukce) a globální ukazatele - do řídicího bloku jednotkového procesu. Další informace o řídicím bloku jednotkového procesu a jeho použití naleznete v části "Přepínání kontextu" na stránce 5-8.

Databázový server přiděluje paměť řídicím blokům jednotkových procesů podle potřeby.

**Zásobníky:** Každý jednotkový proces databázového serveru má vyhrazenou vlastní oblast zásobníku ve virtuální části sdílené paměti. Popis způsobu, jakým jednotkové procesy využívají zásobníky, naleznete v části "Zásobníky" na stránce 5-9. Další informace o způsobu monitorování velikosti zásobníku relace naleznete v části týkající se monitorování relací a jednotkových procesů v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Velikost zásobníku uživatelského jednotkového procesu určuje parametr STACKSIZE konfiguračního souboru ONCONFIG. Výchozí velikost zásobníku je 32 kB. V případě potřeby můžete změnit velikost zásobníku všech jednotkových procesů změnou hodnoty parametru STACKSIZE. Informace a upozornění týkající se nastavení velikosti zásobníku naleznete v části věnované parametru STACKSIZE v kapitole o konfiguračních parametrech příručky *IBM Informix Dynamic Server Administrator's Reference*.

Pokud chcete změnit velikost zásobníku primárního jednotkového procesu konkrétní relace, nastavte proměnnou prostředí **INFORMIXSTACKSIZE**. Hodnota proměnné **INFORMIXSTACKSIZE** potlačí pro konkrétního uživatele hodnotu parametru STACKSIZE. Informace o tom, jak potlačit velikost zásobníku pro konkrétního uživatele jinou hodnotou, naleznete v popisu proměnné prostředí **INFORMIXSTACKSIZE** v příručce *IBM Informix Guide to SQL: Reference*.

Pokud chcete změnit velikost zásobníku bezpečněji, použijte raději namísto změny konfiguračního parametru STACKSIZE proměnnou prostředí **INFORMIXSTACKSIZE**. Proměnná prostředí **INFORMIXSTACKSIZE** ovlivňuje prostor zásobníku pouze u jednoho uživatele a má menší vliv na nové klientské aplikace, které původně nebyly změřeny.

**Haldy:** Každý jednotkový proces má přidělenou haldu. Do haldy ukládá datové struktury, které vytváří za svého provozu. Halda je dynamicky přidělena v okamžiku vytvoření jednotkového procesu. Velikost haldy jednotkového procesu nelze konfigurovat.

## Mezipaměť distribuce dat

Databázový server používá statistické údaje o distribuci dat generované příkazem UPDATE STATISTICS v režimu MEDIUM nebo HIGH k určení plánu dotazu s nejmenší nákladovostí. Pokud databázový server poprvé přistupuje ke statistickým údajům distribuce dat daného sloupce, načte statistické údaje distribuce dat z tabulky **sysdistrib** systémového katalogu na disk a uloží statistické údaje do mezipaměti distribuce dat. Tyto statistické údaje pak může číst a používat k optimalizaci dotazů, které přistupují k tomuto sloupci.

Pokud jsou tyto statistické údaje účinně uloženy a zpřístupněny v mezipaměti distribuce dat, výkon databázového serveru se zvýší. Velikost mezipaměti distribuce dat můžete konfigurovat pomocí konfiguračních parametrů DS\_HASHSIZE a DS\_POOLSIZE. Informace o změně výchozí velikosti mezipaměti distribuce dat naleznete v kapitole o dotazech a optimalizátoru dotazů v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

## Mezipaměť slovníku

Když relace provádí příkaz jazyka SQL, který vyžaduje přístup k systémové tabulce katalogu, databázový server přečte data z tabulek systémového katalogu. Databázový server ukládá data katalogu pro každou dotazovanou tabulku ve struktuře, ke které může efektivněji přistupovat při následných dotazech na tabulku. Tyto struktury se vytvářejí ve virtuální části sdílené paměti a jsou sdílené všemi relacemi. Tyto struktury tvoří mezipaměť slovníku.

Velikost mezipaměti slovníku můžete konfigurovat pomocí konfiguračních parametrů DD\_HASHSIZE a DD\_HASHMAX. Další informace o těchto parametrech naleznete v kapitole o vlivech konfigurace na využití paměti v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

## Mezipaměť příkazů SQL

Mezipaměť příkazů jazyka SQL omezuje využití paměti a dobu přípravy dotazů. Databázový server používá mezipaměť příkazů jazyka SQL k ukládání analyzovaných a optimalizovaných dotazů jazyka SQL spouštěných uživatelem. Pokud uživatel spustí příkaz, který byl již uložen do mezipaměti příkazů jazyka SQL, nebude databázový server příkaz analyzovat a optimalizovat znovu, čímž dojde ke zvýšení výkonu.

Další informace naleznete v části “Nastavení parametrů mezipaměti příkazů jazyka SQL” na stránce 8-6. Informace o vlivech těchto parametrů na výkon mezipaměti příkazů SQL naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

## Řadící paměť

Následující databázové operace mohou využívat velké množství virtuální části sdílené paměti k řazení dat:

- Dotazy pro podporu rozhodování, jejichž součástí jsou operace spojování, seskupování, agregace a řazení výsledků.
- Vytváření indexů.
- Příkaz UPDATE STATISTICS jazyka SQL.

Množství sdílené virtuální paměti přidělené operaci řazení databázovým serverem závisí mimo jiné na počtu řazených řádků a na velikosti řádku.

Informace o paralelním řazení naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

## Rutiny SPL a mezipaměť rutin UDR

Databázový server převádí rutiny v jazyku SPL do spustitelného formátu a ukládá je v mezipaměti rutin UDR, ke které mají přístup všechny relace.

Pokud některá relace poprvé vyžaduje přístup k určité rutině jazyka SPL nebo jiné uživatelské rutině, načte databázový server definici z tabulek systémového katalogu a uloží ji do mezipaměti rutin UDR.

Velikost mezipaměti rutin UDR můžete konfigurovat pomocí konfiguračních parametrů PC\_HASHSIZE a PC\_POOLSIZE. Další informace o změně výchozí velikosti mezipaměti rutin UDR naleznete v kapitole o dotazech a optimalizátoru dotazů v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

## Globální společná oblast

V globální společné oblasti jsou uloženy globální struktury databázového serveru. Globální společná oblast obsahuje například fronty zpráv, do kterých jednotkové procesy cyklického dotazování ukládají zprávy přijaté od klientů. Jednotkové procesy **sqlexec** vyzvedávají zprávy z globální společné oblasti a zpracovávají je.

Další informace naleznete v částech o společných oblastech síťových vyrovnávacích pamětí a o virtuální části sdílené paměti v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

---

## Komunikační část sdílené paměti (systém UNIX)

Pokud nakonfigurujete alespoň jedno připojení jako připojení prostřednictvím sdílené paměti, přidělí databázový server část sdílené paměti jako společnou oblast vyhrazenou pro komunikaci IPC. Databázový server provede toto přiřazení při vytvoření sdílené paměti. Komunikační část obsahuje vyrovnávací paměti pro zprávy místních klientských aplikací, které používají sdílenou paměť ke komunikaci s databázovým serverem.

Velikost komunikační části sdílené paměti se v kilobajtech rovná přibližně dvanáctinásobku očekávaného počtu potřebných připojení prostřednictvím sdílené paměti (**nettype ipcshm**). Pokud parametr **nettype ipcshm** není uveden, očekávaný počet připojení bude mít výchozí hodnotu 50. Informace o tom, jak se klient připojuje ke komunikační části sdílené paměti, naleznete v části “Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)” na stránce 7-6.

---

## Část virtuálních rozšíření sdílené paměti

Část virtuálních rozšíření sdílené paměti obsahuje další virtuální segmenty a segmenty virtuálních rozšíření. Segmenty virtuálních rozšíření obsahují haldy jednotkových procesů modulů DataBlade a uživatelských rutin spouštěných v uživatelských virtuálních procesorech.

Konfigurační parametry SHMADD, EXTSHMADD a SHMTOTAL se používají na část virtuálního rozšíření sdílené paměti, stejně jako se používají na ostatní části sdílené paměti.

---

## Řízení souběžného přístupu

Jednotkové procesy databázového serveru spuštěné v jednom virtuálním procesoru i v různých virtuálních procesorech sdílejí zdroje ve sdílené paměti. Pokud jednotkový proces zapisuje do sdílené paměti, používá mechanismy zvané *objekty mutex* a *zámky lock*, aby do téže oblasti nezapisovaly současně jiné jednotkové procesy. Objekt mutex uděluje procesu oprávnění k přístupu ke sdílenému prostředku. Zámek zabraňuje ostatním jednotkovým procesům zapisovat do vyrovnávací paměti, dokud jednotkový proces, který paměť uzamknul, nedokončí operaci prováděnou ve vyrovnávací paměti a zámek neuvolní.

### Objekty mutex sdílené paměti

Databázový server používá *objekty mutex* ke koordinování jednotkových procesů, které se pokoušejí měnit data ve sdílené paměti. Ke každému prostředku sdílené paměti, který lze měnit, je přidružen objekt mutex. Než jednotkový proces může změnit zdroj ve sdílené paměti, musí nejprve získat objekt mutex přidružený k tomuto zdroji. Jakmile jednotkový proces získá objekt mutex, může měnit zdroj v paměti. Po dokončení změn jednotkový proces objekt mutex opět uvolní.

Pokud se jednotkový proces pokusí získat objekt mutex a zjistí, že objekt mutex je již uzamčen jiným jednotkovým procesem, musí počkat na uvolnění objektu mutex.

Pokud se například dva jednotkové procesy pokusí přistupovat k téže položce tabulky bloků, získá objekt mutex přidružený k tabulce bloků pouze jeden z nich. Pouze jednotkový proces, který získal objekt mutex, smí zapisovat položky do tabulky bloků. Druhý jednotkový proces musí počkat na uvolnění objektu mutex a pak se ho může pokusit získat.

Informace o monitorování objektů mutex (také označovaných jako zámky latch) naleznete v části "Monitorování profilu sdílené paměti a zámků latch" na stránce 8-8.

### Zámky vyrovnávacích pamětí ve sdílené paměti

Základní výhodou sdílené paměti je možnost sdílení přístupu k diskovým stránkám uloženým ve společné oblasti vyrovnávacích pamětí ve sdílené paměti jednotlivými jednotkovými procesy databázového serveru. Databázový server prostřednictvím určité strategie zamykání datových vyrovnávacích pamětí dosahuje vyšší míry souběžného přístupu a současně stále zajišťuje izolaci jednotkových procesů.

#### Typy zámků vyrovnávacích pamětí

Databázový server používá k řízení přístupu k vyrovnávacím pamětem ve sdílené paměti dva typy zámků:

- sdílené zámky
- výlučné zámky

Každý z těchto typů zámků v průběhu provádění vynucuje požadovanou úroveň izolace jednotkových procesů.

**Sdílený zámek:** Vyrovnávací paměť se nachází ve sdíleném režimu (nebo je uzamčena sdíleným zámekem), pokud několik jednotkových procesů přistupuje k vyrovnávací paměti a čte uložená data, ale žádný jednotkový proces nezamýšlí data měnit.

**Výlučný zámek:** Vyrovnávací paměť se nachází ve výlučném režimu (nebo je uzamčena výlučným zámekem), pokud některý jednotkový proces požádá o výlučný přístup ke sdílené paměti. Všechny ostatní požadavky jednotkových procesů na přístup k vyrovnávací paměti jsou umístěny do fronty čekajících procesů. Jakmile je prováděný jednotkový proces připraven uvolnit výlučný zámek, probudí tím další jednotkový proces ve frontě čekajících procesů.

---

## Přístup jednotkových procesů databázového serveru k sdíleným vyrovnávacím pamětem

Jednotkové procesy databázového serveru přistupují ke sdíleným vyrovnávacím pamětem prostřednictvím systému front s použitím synchronizace přístupu a ochrany dat pomocí objektů mutex a zámků.

### Fronty FIFO/LRU

Ve vyrovnávací paměti jsou uložena data, aby k nim server mohl mít rychlý přístup. Databázový server používá k nahrazování dat uložených v mezipaměti fronty nejdéle nepoužitých položek (fronty LRU). Server Dynamic Server také používá frontu FIFO (First-in First-out). Pokud nastavujete počet front LRU, nastavujete ve skutečnosti počet front FIFO/LRU.

Konfigurační parametr BUFFERPOOL slouží k určení údajů o společné oblasti vyrovnávací paměti, včetně údajů o počtu front LRU, které se mají vytvořit při vytvoření sdílené paměti databázového serveru, a hodnot voleb `lru_min_dirty` a `lru_max_dirty`, které ovládají, jak často je sdílená paměť vyprazdňována na disk.

Lepší propustnosti transakcí dosáhnete zvýšením hodnot `lru_min_dirty` a `lru_max_dirty`. Neměňte však rozdíl mezi hodnotou `lru_min_dirty` a `lru_max_dirty`.

**Poznámka:** Informace, které byly před verzí 10.0 určovány pomocí konfiguračních parametrů `BUFFERS`, `LRUS`, `LRU_MAX_DIRTY` a `LRU_MIN_DIRTY`, jsou nyní určovány pomocí konfiguračního parametru `BUFFERPOOL`.

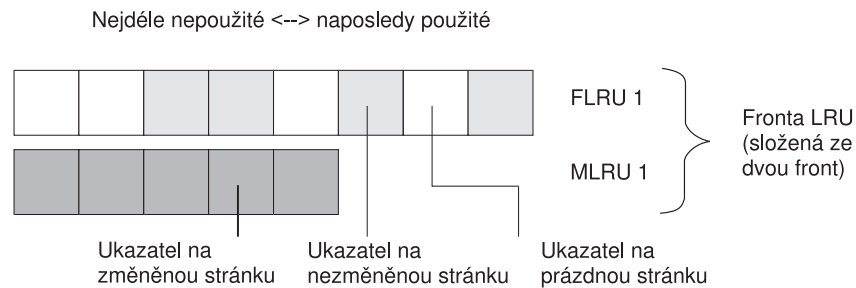
### Komponenty front LRU

Každá fronta LRU se skládá z dvojice následujících spojových seznamů:

- Seznam FLRU (free least-recently used), který zaznamenává volné a nezměněné stránky ve frontě.
- Seznam MLRU (modified least-recently used), který zaznamenává změněné stránky ve frontě.

Seznam volných a nezměněných stránek je označován jako fronta FLRU páru front, seznam změněných stránek je označován jako fronta MLRU. Díky dvěma samostatným seznamům není zapotřebí hledat volné a nezměněné stránky prohledáváním fronty. Obrázek Obrázek 7-6 znázorňuje strukturu front LRU.





Obrázek 7-6. Fronta LRU

### Stránky seřazené podle nejděší doby nepoužití

Pokud databázový server zpracovává požadavky na načtení stránky z disku, musí rozhodnout, kterou stránku v paměti nahradí načítanou stránkou. Databázový server nevybírání stránky náhodně, ale předpokládá, že nedávno použité stránky budou opět použity s větší pravděpodobností, než stránky, které dlouho použity nebyly. Server tedy nenahrazuje stránky, ke kterým jednotkové procesy nedávno přistupovaly, ale nahradí stránku, která byla po nejděší dobu nepoužita. Databázový server může snadno nalézt nejděle nepoužité stránky, protože udržuje seznam stránek seřazený sestupně podle doby nepoužívání.

### Správa společné oblasti vyrovnávací paměti a front LRU

Před zahájením zpracování jsou vyrovnávací paměti stránek prázdné a každá vyrovnávací paměť je reprezentována položkou v jedné z front FLRU. Rozložení vyrovnávacích pamětí ve frontách FLRU je rovnoměrné. Počet vyrovnávacích pamětí v každé frontě lze vypočítat vydělením celkového počtu vyrovnávacích pamětí počtem front LRU. Počet vyrovnávacích pamětí a front LRU je určen konfiguračním parametrem BUFFERPOOL .

Pokud uživatelský jednotkový proces potřebuje získat vyrovnávací paměť, vybere databázový server náhodnou frontu FLRU a použije nejděle nepoužitou položku v seznamu. Pokud je možné nejděle nepoužitou položku zamknout pomocí zámku latch, bude stránka z fronty odstraněna.

Pokud je fronta FLRU uzamčena a není možné uzamknout její koncovou stránku pomocí zámku latch, vybere databázový server náhodně jinou frontu FLRU.

Pokud uživatelský jednotkový proces hledá konkrétní stránku ve sdílené paměti, získá umístění stránky ve frontě LRU pomocí řídicích informací uložených v tabulce vyrovnávacích pamětí.

Jakmile prováděný jednotkový proces dokončí prováděnou operaci, opět vyrovnávací paměť uvolní. Pokud byla stránka změněna, je umístěna na konec fronty MLRU, na kterém se nacházejí naposledy použité položky. Pokud stránka byla čtena, ale nebyla změněna, je umístěna zpět do fronty FLRU na konec, na kterém se nacházejí naposledy použité položky. Informace o tom, jak monitorovat fronty LRU, naleznete v části "Monitorování aktivity společné oblasti vyrovnávací paměti" na stránce 8-10.

### Počet front LRU, které je třeba nakonfigurovat

Účel většího počtu front LRU je následující:

- Omezují soupeření uživatelských jednotkových procesů o fronty.
- Umožňují vyprazdňovat stránky z front LRU pomocí více jednotkových procesů čištění stránek a udržují tak procento neaktualizovaných stránek na přijatelné úrovni.

Doporučené počáteční hodnoty parametru LRUS závisí na počtu jednotek CPU dostupných v počítači. Pokud má váš počítač jeden procesor, začněte nastavením hodnoty **lrus** v konfiguračním parametru BUFFERPOOL na hodnotu 4. Pokud má váš počítač více procesorů, použijte následující zápis:

```
LRUS = max(4, (NUMCPUVPS))
```

Po zadání počáteční hodnoty volby **lrus** v konfiguračním parametru BUFFERPOOL monitorujte fronty LRU pomocí příkazu **onstat -R**. Pokud zjistíte, že procento neaktualizovaných front LRU soustavně překračuje hodnotu určenou pomocí volby **lru\_max\_dirty**, přidejte fronty LRU zvýšením hodnoty volby **lrus**

Nastavili jste například volbu **lru\_max\_dirty** na hodnotu 70 a následně jste zjistili, že fronty LRU jsou trvale neaktualizované ze 75 procent. Zvažte možnost zvýšit hodnotu volby **lrus**. Pokud zvýšíte počet front LRU, zkrátíte tím tyto fronty a omezíte tak práci jednotkových procesů čištění stránek. Je však také zapotřebí přidělit dostatečný počet jednotkových procesů čištění pomocí konfiguračního parametru CLEANERS popsaného v následující části. Zachovejte stejný rozdíl mezi hodnotou **lru\_max\_dirty** a **lru\_min\_dirty**.

**Poznámka:** Informace, které byly před verzí 10.0 určovány pomocí konfiguračních parametrů BUFFERS, LRUS, LRU\_MAX\_DIRTY a LRU\_MIN\_DIRTY, jsou nyní určovány pomocí konfiguračního parametru BUFFERPOOL.

## Počet jednotkových procesů čištění, které je třeba přidělit

Obecně byste měli nakonfigurovat jeden proces čištění pro každý disk, který aplikace často aktualizují. Měli byste však také posoudit délku front LRU a frekvenci kontrolních bodů. Tyto ukazatele jsou vysvětleny v následujících odstavcích.

Kromě nedostatečného počtu front LRU může schopnost jednotkových procesů čištění dostatečně rychle zpracovávat stránky k vyčištění ovlivnit i počet přidělených jednotkových procesů čištění. Procento neaktualizovaných stránek může v některých frontách překročit hodnotu volby **lru\_max\_dirty** parametru BUFFERPOOL, protože k vyčištění front nejsou dostupné žádné jednotkové procesy čištění. Po určité době mohou jednotkové procesy čištění zaostávat za potřebou čištění stránek do takové míry, že procento neaktualizovaných stránek ve společné oblasti vyrovnávacích pamětí překročí hodnotu zadanou pomocí volby **lru\_max\_dirty**.

Předpokládejme například, že parametr CLEANERS je nastaven na hodnotu 8 a počet front LRU byl zvýšen z hodnoty 8 na hodnotu 12. V takovém případě lze očekávat pouze malý nárůst výkonu, protože nyní musí osm existujících jednotkových procesů čištění čistit navíc další čtyři fronty. Pokud zvýšíte počet jednotkových procesů čištění změnou parametru CLEANERS na hodnotu 12, bude moci být každá ze zkrácených front účinněji čištěna vlastním jednotkovým procesem čištění.

Pokud parametr CLEANERS nastavíte na příliš nízkou hodnotu, může v každém kontrolním bodě dojít k poklesu výkonu, protože v kontrolním bodě musejí jednotkové procesy čištění vyprázdnit na disk všechny změněné stránky. Pokud nenakonfigurujete dostatečný počet jednotkových procesů čištění, budou kontrolní body trvat déle a celkový výkon serveru poklesne.

Další informace naleznete v části “Vyprazdňování vyrovnávacích pamětí ve společné oblasti” na stránce 7-27.

## Počet stránek přidávaných do front MLRU

Jednotkové procesy pravidelně vyprazdňují změněné vyrovnávací paměti z fronty MLRU na disk. Bod zahájení čištění určete pomocí volby **lru\_max\_dirty** konfiguračního parametru BUFFERPOOL.

Hodnota volby **lru\_max\_dirty** určuje, kdy bude spuštěno čištění stránek, a omezuje tak počet vyrovnávacích pamětí stránek, které mohou být připojeny k frontě MLRU. Počáteční hodnota volby **lru\_max\_dirty** je 60.00, to znamená, že čištění bude spuštěno, jakmile bude změněno šedesát procent vyrovnávacích pamětí spravovaných frontou.

V praxi je čištění spouštěno, pokud je splněna jedna z několika možných podmínek. Dosažení hodnoty **lru\_max\_dirty** frontou MLRU je pouze jedna z těchto podmínek. Další informace o tom, jak databázový server vyprazdňuje společnou oblast vyrovnávacích pamětí, naleznete v části “Vyprazdňování dat na disk” na stránce 7-26.

Obrázek Obrázek 7-7 znázorňuje, jakým způsobem hodnota volby **lru\_max\_dirty** určuje okamžik spuštění čištění stránek fronty LRU, čímž omezuje počet vyrovnávacích pamětí ve frontě MLRU.

```
Buffers specified as 8000
lrus specified as 8
lru_max_dirty specified as 60 percent
```

```
Page cleaning begins when the number of buffers in the MLRU
queue is equal to lru_max_dirty.
```

```
Buffers per lru queue = (8000/8) = 1000
```

```
Max buffers in MLRU queue and point at which page cleaning
begins: 1000 x 0.60 = 600.
```

*Obrázek 7-7. Jak hodnota lru\_max\_dirty spouští čištění stránek a omezuje velikost fronty MLRU*

## Konec čištění fronty MLRU

Můžete také určit bod, ve kterém bude čištění fronty MLRU ukončeno. Hodnota **lru\_min\_dirty** konfiguračního parametru BUFFERPOOL určuje přijatelné procento vyrovnávacích pamětí ve frontě MLRU. Pokud je například volba **lru\_min\_dirty** nastavena na hodnotu 50.00, nebude čištění stránek požadováno, pokud bude změněno padesát procent vyrovnávacích pamětí fronty LRU. V praxi může čištění stránek řízené jednotkovými procesy čištění pokračovat i po překročení tohoto bodu.

Obrázek Obrázek 7-8 znázorňuje, jakým způsobem hodnota **lru\_min\_dirty** určuje přijatelné procento vyrovnávacích pamětí ve frontě MLRU, po jehož dosažení je čištění stránek fronty LRU ukončeno.

```
Buffers specified as 8000
lrus specified as 8
lru_min_dirty specified as 50 percent
```

```
The acceptable number of buffers in the MLRU queue and
the point at which page cleaning can end is equal
to lru_min_dirty.
```

```
Buffers per LRU queue = (8000/8) = 1000
```

```
Acceptable number of buffers in MLRU queue and the point
at which page cleaning can end: 1000 x .50 = 500
```

*Obrázek 7-8. Jak hodnota volby lru\_min\_dirty určuje bod, ve kterém může být čištění ukončeno.*

Jako hodnoty voleb **lru\_max\_dirty** a **lru\_min\_dirty** můžete použít desetinná čísla. Pokud například nastavíte volbu **lru\_max\_dirty** na hodnotu 1.0333 a volbu **lru\_min\_dirty** na hodnotu 1.0, bude zápis vyrovnávacích pamětí fronty LRU spuštěn, pokud počet neaktualizovaných vyrovnávacích pamětí dosáhne hodnoty 3100 a poté bude opět zastaven, pokud klesne na hodnotu 3000.

Další informace o tom, jak databázový server vyprazdňuje společnou oblast vyrovnávacích pamětí, naleznete v části “Vyprazdňování dat na disk” na stránce 7-26.

## Konfigurace dopředného čtení databázového serveru

Databázový server můžete pro případy sekvenčního prohledávání tabulek a indexů nakonfigurovat tak, aby načítal několik stránek v předstihu, zatímco aktuální zpracovává. Čtení napřed umožňuje urychlit běh aplikací, protože omezuje čekání na dokončení operací diskového vstupu - výstupu.

Databázový server provádí čtení napřed, kdykoli zjistí, že je zapotřebí při provádění sekvenčního čtení tabulek nebo indexů.

Konfigurační parametr `RA_PAGES` v souboru `ONCONFIG` určuje počet stránek, které databázový server načítá z disku nebo indexu při čtení napřed.

Parametr `RA_THRESHOLD` určuje počet nezpracovaných stránek v paměti, po jehož dosažení provede databázový server další čtení napřed. Pokud je například parametr `RA_PAGES` nastaven na hodnotu 10 a parametr `RA_THRESHOLD` na hodnotu 4, načte databázový server v předstihu dalších deset stránek, jakmile budou ve vyrovnávací paměti zbývat ke zpracování poslední čtyři stránky.

Další informace o použití příkazu `onstat -p` ke sledování používání čtení napřed databázovým serverem naleznete v části “Monitorování profilu sdílené paměti a zámků latch” na stránce 8-8. Příklad výstupu příkazu `onstat -p` naleznete v příručce *IBM Informix Administrator's Reference*.

## Přístup jednotkových procesů databázového serveru ke stránkám ve vyrovnávací paměti

Databázový server ukládá stránky do vyrovnávací paměti s použitím sdílených zámků, aby umožnil přistupovat k téže vyrovnávací paměti několika jednotkovým procesům současně. Databázový server používá dvě kategorie zámků vyrovnávacích pamětí, aby mohl poskytnout tento souběžný přístup a nenarušil přitom izolaci jednotkových procesů. Tyto dvě kategorie přístupu zámků jsou *sdílený zámek* a *vylučný zámek*. (Další informace naleznete v části “Typy zámků vyrovnávacích pamětí” na stránce 7-21.)

---

## Vyprazdňování dat na disk

Zapisování vyrovnávací paměti na disk se nazývá *vyprazdňování vyrovnávací paměti*. Když uživatelský jednotkový proces mění data ve vyrovnávací paměti, označí vyrovnávací paměť jako *neaktualizovanou*. Když databázový server vyprazdňuje vyrovnávací paměť na disk, *zruší* označení vyrovnávací paměti jako *neaktualizované* a umožní přepsat data ve vyrovnávací paměti.

Databázový server vyprazdňuje následující vyrovnávací paměti:

- Společnou oblast vyrovnávacích pamětí (popisovanou v této části).
- Vyrovnávací paměť fyzického protokolu.

Další informace naleznete v části “Vyprazdňování vyrovnávací paměti fyzického protokolu” na stránce 7-27.

- Vyrovnávací paměť logického protokolu.

Další informace naleznete v části “Vyprazdňování vyrovnávací paměti logického protokolu” na stránce 7-29.

Vyprazdňování vyrovnávacích pamětí spravují jednotkové procesy čištění stránek. V databázovém serveru je vždy spuštěn alespoň jeden jednotkový proces čištění stránek. Pokud je databázový server nakonfigurován tak, aby spouštěl více než jeden jednotkový proces čištění stránek, jsou fronty LRU rozděleny mezi jednotkové procesy čištění stránek tak, aby vyprazdňování probíhalo účinněji. Informace o tom, jak určit počet jednotkových procesů čištění stránek, které má databázový server spouštět, naleznete v části o konfiguračním parametru CLEANERS v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Vyprazdňování vyrovnávacích pamětí fyzického protokolu, změněných stránek vyrovnávacích pamětí ve sdílené paměti a vyrovnávacích pamětí logického protokolu musí být synchronizováno s aktivitou jednotkových procesů čištění stránek podle určitých pravidel, která jsou navržena tak, aby zajistila konzistenci dat.

## Vyprazdňování vyrovnávacích pamětí ve společné oblasti

Vyprazdňování vyrovnávacích pamětí vyvolává jedna z následujících podmínek:

- Počet neaktualizovaných vyrovnávacích pamětí ve frontě MLRU dosáhne hodnoty určené volbou **lru\_max\_dirty** konfiguračního parametru BUFFERPOOL.
- Jednotkové procesy čištění stránek zaostávají za potřebou čištění stránek. Jinými slovy, uživatelský jednotkový proces potřebuje získat vyrovnávací paměť, ale žádná nezměněná vyrovnávací paměť není dostupná.
- Databázový server musí provést kontrolní bod. (Viz “Kontrolní body” na stránce 15-5.)

Automatické ladění fronty LRU ovlivňuje všechny všechny společné oblasti vyrovnávací paměti a upravuje hodnoty **lru\_min\_dirty** a **lru\_max\_dirty** v konfiguračním parametru BUFFERPOOL.

## Přednostní vyprazdňování předobrazů

Předobrazy změněných stránek jsou vyprázdněny na disk před samotnými změněnými stránkami.

V praxi je nejprve vyprazdňována vyrovnávací paměť fyzického protokolu a pak vyrovnávací paměti obsahující změněné stránky. Pokud tedy musí být na disk zapsána stránka vyrovnávací paměti ve sdílené paměti, protože některý jednotkový proces potřebuje vyrovnávací paměť, ale žádná není dostupná (zápis na popředí), nelze vyprázdnit stránky ve vyrovnávacích pamětech, dokud nebude na disk zapsán předobraz stránky.

## Vyprazdňování vyrovnávací paměti fyzického protokolu

Databázový server ve vyrovnávací paměti fyzického protokolu dočasně ukládá předobrazy některých změněných diskových stránek. Pokud je předobraz zapsán do vyrovnávací paměti fyzického protokolu, avšak ne do fyzického protokolu na disk, server vyprázdní vyrovnávací paměť fyzického protokolu na disk před vyprázdněním změněné stránky na disk.

Databázový server vždy před načtením dat do vyrovnávací paměti vyprázdní obsah vyrovnávací paměti fyzického protokolu na disk.

Následující události způsobí vyprázdnění aktivní vyrovnávací paměti fyzického protokolu:

- Aktivní fyzický protokol se naplní.
- Změněná stránka ve sdílené paměti musí být vyprázdněna, ale předobraz je stále v aktivní vyrovnávací paměti fyzického protokolu.
- Vyskytne se kontrolní bod.

Databázový server používá vždy pouze jednu ze dvou vyrovnávacích pamětí fyzického protokolu. Tato vyrovnávací paměť je aktivní (nebo aktuální) vyrovnávací paměť fyzického

protokolu. Než databázový server vyprázdní aktivní vyrovnávací paměť logického protokolu na disk, nastaví další vyrovnávací paměť jako aktivní vyrovnávací paměť fyzického protokolu, aby do ní mohl zapisovat v průběhu vyprazdňování první předchozí vyrovnávací paměti.

Vyrovňovací paměť fyzického protokolu a fyzický protokol pomáhají udržovat fyzickou a logickou konzistenci dat. Informace o fyzickém protokolování, kontrolních bodech a rychlé obnově naleznete v části Kapitola 15, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 15-1.

## Synchronizace vyprazdňování vyrovnávacích pamětí

Při prvním vytvoření sdílené paměti jsou všechny vyrovnávací paměti prázdné. V průběhu zpracování dat jsou datové stránky načítány z disku do vyrovnávacích pamětí a uživatelské jednotkové procesy tyto stránky začnou měnit.

## Popis aktivity vyprazdňování

Aby mohl poskytovat informace o konkrétních stavech, které aktivovaly vyprazdňování vyrovnávacích pamětí, rozlišuje databázový server tři typy zápisů a počítá, jak často k jednotlivým typům zápisů došlo:

- zápis na popředí
- zápis front LRU
- zápis bloku.

Ke zobrazení počtu zápisů, které databázový server spravuje, použijte příkaz `onstat -F`, jak je popsáno v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Pokud implementujete zrcadlení databázového serveru, budou data zapsána vždy nejprve do primárního bloku. Pak bude zápis zopakován v zrcadleném bloku. Zápisy do zrcadlených bloků jsou započítávány do počtu zápisů. Další informace o monitorování typů zápisů prováděných databázovým serverem naleznete v části “Monitorování aktivity společné oblasti vyrovnávací paměti” na stránce 8-10.

### Zápis na popředí

Kdykoli zapíše vyrovnávací paměť na disk jednotkový proces **sqlexec**, je tento zápis označován jako *zápis na popředí*. K zápisu na popředí dojde, pokud jednotkový proces **sqlexec** prohledává z pověření uživatele fronty LRU, ale nemůže najít žádnou prázdnou nebo nezměněnou stránku. Aby získal prostor, vyprazdňuje jednotkový proces **sqlexec** jednotlivé stránky na disk, aby mohl uložit data načtená z disku. (Další informace naleznete v části “Fronty FIFO/LRU” na stránce 7-22.)

Pokud procesy **sqlexec** musejí vyprazdňovat stránky, aby získaly vyrovnávací paměti ve sdílené paměti, může dojít ke snížení výkonu serveru. Zápisy na popředí nejsou žádoucí. Počet zápisů na popředí zobrazte pomocí příkazu **onstat -F**. Pokud zjistíte, že k zápisům na popředí dochází často, vyladte parametry jednotkových procesů čištění stránek. Zvyšte počet jednotkových procesů čištění nebo v parametru `BUFFERPOOL` snižte hodnotu volby **lru\_max\_dirty**.

### Zápisy front LRU

Na rozdíl od zápisů na popředí jsou zápisy front LRU prováděny jednotkovými procesy čištění stránek, nikoli jednotkovými procesy **sqlexec**. Databázový server provádí zápisy front LRU jako zápisy na pozadí, k čemuž obvykle dochází, když procentní část neaktualizovaných vyrovnávacích pamětí přesáhne procento určené u volby **lru\_max\_dirty** v konfiguračním parametru `BUFFERPOOL`.

Dále může být zápisem na popředí spuštěn zápis fronty LRU. Pokud dojde zápisu na popředí, bude jednotkový proces čištění stránky upozorněn a probuzen zapisujícím jednotkovým procesem **sqlexec** a následně vyčistí frontu LRU, pro kterou byl zápis proveden.

Ve správně vyladěném systému se jednotkové procesy čištění starají o to, aby byl vždy dostupný dostatečný počet nezměněných vyrovnávacích pamětí k ukládání stránek načtených z disku. Proto jednotkové procesy **sqlexec**, které provádějí dotazy, nemusejí před načtením požadovaných diskových stránek vyprazdňovat stránky na disk. V takovém případě může dojít k výraznému zvýšení výkonu dotazů, které nepoužívají zápisy na popředí.

Zápisy front LRU jsou upřednostňovány před zápisy na popředí, protože jednotkové procesy čištění stránek zapisují vyrovnávací paměti mnohem efektivněji než jednotkové procesy **sqlexec**. K monitorování obou typů zápisů použijte příkaz **onstat -F**.

## Zápisy bloků

*Zápisy bloků* provádějí jednotkové procesy čištění stránek obvykle v průběhu kontrolního bodu, případně tehdy, pokud byly změněny všechny stránky ve společné oblasti vyrovnávacích pamětí ve sdílené paměti. Zápisy bloků, které jsou prováděny jako zápisy seřazených stránek, jsou neúčinnějším typem zápisů, které databázový server může provádět.

Při zapisování bloků je každý jednotkový proces čištění stránek přiřazen jednomu nebo více blokům. Každý jednotkový proces čištění stránek čte záhlaví vyrovnávacích pamětí a vytváří pole ukazatelů na stránky přidružené k bloku, který mu byl určen. (Jednotkové procesy čištění stránek mají k těmto informacím přístup, protože číslo bloku je součástí číselné adresy fyzické stránky, která je uložena v záhlaví stránky.) Toto řazení umožňuje minimalizovat pohyb diskových hlav (vyhledávací dobu disku) a také umožňuje jednotkovým procesům čištění stránek při zápisu používat velké vyrovnávací paměti, pokud je to možné.

Protože uživatelské jednotkové procesy musejí čekat na dokončení kontrolního bodu, nemusejí jednotkové procesy čištění stránek soupeřit s velkým počtem ostatních jednotkových procesů o čas jednotky CPU. Díky tomu mohou jednotkové procesy čištění stránek dokončit svou práci s menším počtem přepnutí kontextů.

## Vyprazdňování vyrovnávací paměti logického protokolu

Databázový server používá vyrovnávací paměť logického protokolu ve sdílené paměti jako dočasnou paměť záznamů, které popisují změny provedené v stránkách databázového serveru. Tyto záznamy změn jsou následně přepsány z vyrovnávací paměti logického protokolu do aktuálního souboru logického protokolu na disku a nakonec na zálohovací médium logického protokolu. Popis logického protokolování naleznete v části Kapitola 13, "Logický protokol", na stránce 13-1.

Vyprázdnění aktuální vyrovnávací paměti logického protokolu může způsobit pět různých událostí:

- Dojde k zaplnění aktuální vyrovnávací paměti logického protokolu.
- V databázi s protokolováním bez vyrovnávací paměti je připravena nebo potvrzena transakce.
- Dojde k ukončení relace databáze bez protokolování.
- Vyskytne se kontrolní bod.
- Dojde ke změně stránky, která nevyžaduje předobraz ve fyzickém protokolu.

Následující části podrobně popisují každou z těchto událostí.

## Po přípravě nebo dokončení transakce v databázi s protokolováním bez vyrovnávací paměti

Následující záznamy protokolu způsobí v databázi s protokolováním bez vyrovnávací paměti vyprázdnění vyrovnávacích pamětí logického protokolu:

- COMMIT
- PREPARE
- XPREPARE
- ENDTRANS

Porovnání protokolování s vyrovnávací paměti a protokolování bez vyrovnávací paměti naleznete v části týkající se příkazu SET LOG v příručce *IBM Informix Guide to SQL: Syntax*.

## Při ukončení relace, která používá databáze bez protokolování nebo protokolování bez vyrovnávací paměti

Databázový server protokoluje určité aktivity měnící schéma databáze, například vytváření tabulek nebo oblastí i v případě databází bez protokolování. Pokud databázový server ukončí relaci, která používá protokolování bez vyrovnávací paměti nebo databáze bez protokolování, bude vyrovnávací paměť logického protokolu vyprázdněna, aby server zajistil, že všechny aktivity protokolování budou zaznamenány.

## Pokud nastane kontrolní bod

Podrobný popis událostí, ke kterým dochází v kontrolním bodu naleznete v části “Kontrolní body” na stránce 15-5.

## Pokud byla změněna stránka, která nevyžaduje předobraz v souboru fyzického protokolu

Pokud byla změněna stránka, která nevyžaduje předobraz ve fyzickém protokolu, musí být před vyprázdněním stránky na disk vyprázdněna vyrovnávací paměť logického protokolu.

---

## Použití vyrovnávací paměti pro velké objekty

Jednoduché velké objekty (data typu TEXT a BYTE) mohou být uloženy v prostorech dbspace nebo v prostorech blobspace. Inteligentní velké objekty (data typu CLOB a BLOB) jsou ukládány pouze do prostorů sbspace. Databázový server přistupuje k různým typům paměťových prostorů pomocí různých metod. Následující části popisují způsob použití vyrovnávací paměti pro každý z těchto typů paměťových prostorů.

### Zápis jednoduchých velkých objektů

Databázový server zapisuje jednoduché velké objekty do diskových stránek prostoru dbspace stejným způsobem, jakým zapisuje ostatní data. Další informace naleznete v části “Vyprazdňování dat na disk” na stránce 7-26.

Jednoduché velké objekty můžete také přiřadit prostorům blobspace. Databázový server zapisuje jednoduché velké objekty do prostorů blobspace jiným způsobem, než jakým zapisuje ostatní data do vyrovnávacích pamětí ve sdílené paměti a pak je vyprazdňuje na disk. Popis prostorů blobspace naleznete v kapitole o diskových strukturách a paměti v příručce *IBM Informix Dynamic Server Administrator's Reference*.

### Stránky blobpage a sdílená paměť

Do stránek blobpage v prostorech blobspace jsou ukládány velké objemy dat. Proto databázový server nevytváří ani nepřístupňuje stránky blobpage prostřednictvím společné oblasti vyrovnávacích pamětí ve sdílené paměti ani stránky blobpage v prostoru blobspace nezapisuje do fyzických nebo logických protokolů.



Pokud data prostoru blobpage procházejí společnou oblastí ve sdílené paměti, může dojít ke snížení účinnosti společné oblasti z důvodu vytěsnění datových stránek a stránek indexů. Namísto toho jsou data stránek blobpage zapisována přímo na disk v okamžiku vytvoření.

Aby omezil provoz logického a fyzického protokolu, zapisuje databázový server stránky blobpage z magnetického média na zálohovací pásky prostorů dbspace jiným způsobem, než jakým zapisuje stránky prostorů dbspace. Popis způsobu, jakým jsou protokolovány prostory blobpage naleznete v části “Protokolování prostorů blobpage a jednoduché velké objekty” na stránce 13-6.

Stránky blobpage ukládané na optická média nejsou zapisovány na zálohovací pásky prostorů dbspace a logického protokolu, protože optická média jsou vysoce spolehlivá.

### **Vytváření jednoduchých velkých objektů**

Řádek, do kterého data jednoduchého velkého objektu patří, nemusí v okamžiku zápisu těchto dat na disk ještě existovat. Jednoduchý velký objekt může být například v průběhu operace INSERT přenesen před ostatními daty. Po vytvoření jednoduchého velkého objektu je vytvořen datový řádek s deskriptorem o velikosti 56 B ukazujícím na umístění objektu. Popis způsobu fyzického uložení jednoduchých velkých objektů naleznete v části týkající se struktury stránky blobpage prostoru dbspace v kapitole o diskové paměti a strukturách v příručce *IBM Informix Dynamic Server Administrator's Reference*.

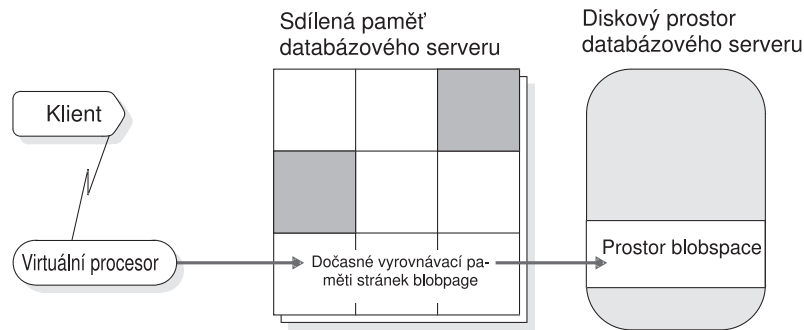
### **Vytvoření vyrovnávacích pamětí stránek blobpage**

Aby mohl databázový server přijímat od procesu aplikace data jednoduchých velkých objektů, vytváří dvojici vyrovnávacích pamětí prostoru blobpage, jednu ke čtení a jednu k zápisu. Obě paměti mají velikost jedné stránky blobpage prostoru blobpage. Každý uživatel má k dispozici pouze jednu sadu vyrovnávacích pamětí prostoru blobpage, a proto může v každém okamžiku přistupovat pouze k jedinému jednoduchému velkému objektu.

Data jednoduchého velkého objektu jsou z klientské aplikace přenášena do databázového serveru v segmentech o velikosti 1 kB. Databázový server začne zaplňovat vyrovnávací paměti prostoru blobpage částmi o velikosti 1 kB a bude se pokoušet ukládat do vyrovnávací paměti dvě stránky blobpage současně. Databázový server ukládá do vyrovnávací paměti dvě stránky blobpage současně, aby mohl určit, kdy má do první stránky přidat ukazatel na druhou stránku. Pokud databázový server zaplní první vyrovnávací paměť a zjistí, že na přenos čekají další data, přidá ke stránce před zápisem na disk ukazatel na následující stránku. Pokud již nezbyvají žádná další data k přenosu, zapíše databázový server na disk poslední stránku bez ukazatele na další stránku.

Jakmile začne jednotkový proces zapisovat první vyrovnávací paměť prostoru blobpage na disk, pokusí se provést operaci vstupu - výstupu na základě uživatelem definované velikosti stránky blobpage. Pokud je například velikost stránky blobpage 32 kB, bude se databázový server pokoušet zapisovat a číst data v blocích po 32768 B. Pokud použitý hardware (například řadič disku) nedokáže přenášet takový objem dat v jediné operaci, bude jádro operačního systému interně (v režimu jádra) provádět operace ve smyčce, dokud nebude přenos dat dokončen.

Vyrovňovací paměti prostoru blobpage budou existovat, dokud nebude dokončen jednotkový proces, který je vytvořil. Jakmile bude jednoduchý velký objekt zapsán na disk, zruší databázový server přidělenou dvojici vyrovnávacích pamětí prostoru blobpage. Obrázek Obrázek 7-9 znázorňuje proces zápisu jednoduchého velkého objektu do prostoru blobpage.



Obrázek 7-9. Zápis jednoduchého velkého objektu do prostoru blobpace

Stránky blobpage jsou v prostorech blobpace přidělovány a zaznamenávány prostřednictvím stránky s mapou volných stránek. Propojení, která spojují stránky blobpage a ukazatele na následující segmenty stránek blobpage, jsou vytvářeny podle potřeby.

Záznam o operacích insert (vlození), update (aktualizace) a delete (odstranění) je zapisován do vyrovnávací paměti logického protokolu.

## Přístup k inteligentním velkým objektům

Databázový server přistupuje k inteligentním velkým objektům prostřednictvím vyrovnávacích pamětí ve sdílené paměti, podobným způsobem, jakým přistupuje k datům uloženým v prostoru dbspace. Část inteligentního velkého objektu obsahující uživatelská data je však do vyrovnávací paměti ukládána s nižší prioritou než běžné stránky vyrovnávacích pamětí, aby nedošlo k vytěsnění cenných dat z vyrovnávací paměti. Použití vyrovnávací paměti umožňuje rychlejší přístup k často používaným inteligentním velkým objektům.

Inteligentní velké objekty jsou ukládány do prostorů sbspace. Do prostorů sbspace nelze ukládat jednoduché velké objekty a inteligentní velké objekty nelze ukládat do prostorů blobpace. Prostor sbspace se skládá z oblasti uživatelských dat a z oblasti metadat. Oblast uživatelských dat obsahuje data inteligentního velkého objektu. Oblast metadat obsahuje informace o obsahu prostoru sbspace. Další informace o prostorech sbspace naleznete v části "Prostory sbspace" na stránce 9-12.

Protože inteligentní velké objekty procházejí společnou oblastí vyrovnávacích pamětí ve sdílené paměti a mohou být protokolovány, je třeba s nimi počítat při přidělování vyrovnávacích pamětí. K přidělení vyrovnávacích pamětí ve sdílené paměti použijte konfigurační parametr BUFFERPOOL. Obecné pravidlo určuje, že by měl být dostupný dostatek vyrovnávacích pamětí k uložení dvou stránek každého otevřeného inteligentního velkého objektu. (Druhá stránka je dostupná proto, aby bylo možné provádět čtení napřed.) Další informace o ladění vyrovnávacích pamětí pro inteligentní velké objekty naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Velikost vyrovnávací paměti logického protokolu nastavte pomocí konfiguračního parametru LOGBUFF. Informace o nastavení následujících konfiguračních parametrů naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*:

- BUFFERPOOL
- LOGBUFF

Oblast uživatelských dat protokolovaných inteligentních velkých objektů neprochází fyzickým protokolem, proto není zapotřebí nastavovat k používání inteligentních velkých objektů parametr PHYSBUFF.

Další informace o struktuře prostorů sbspace naleznete v části týkající se struktury prostorů sbspace v kapitole o diskových strukturách a paměti v příručce *IBM Informix Dynamic Server Administrator's Reference*. Další informace o vytváření prostorů sbspace naleznete v informacích o obslužném programu **onspaces** v příručce *IBM Informix Dynamic Server Administrator's Reference*.

---

## Využití paměti na 64bitových platformách

Při použití 64bitového adresování můžete prostřednictvím velkých společných oblastí vyrovnávacích pamětí omezit počet operací vstupu - výstupu potřebných k získání dat z disku. Protože 64bitové platformy umožňují používat větší paměťový adresový prostor, mají následující konfigurační parametry na 64bitových platformách větší maximální hodnoty:

- BUFFERPOOL
- CLEANERS
- DS\_MAX\_QUERIES
- DS\_TOTAL\_MEMORY
- LOCKS
- LRUS
- SHMADD
- SHMVIRTSIZE

Maximální hodnoty těchto konfiguračních parametrů a parametrů specifických pro platformu (například parametru SHMMAX) uvádí soubor Poznámky k počítači každé 64bitové platformy. Další informace o konfiguračních parametrech naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference* a v kapitole o sdílené paměti v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.



---

## Kapitola 8. Správa sdílené paměti

|                                                                                |      |
|--------------------------------------------------------------------------------|------|
| Obsah kapitoly                                                                 | 8-1  |
| Nastavení konfiguračních parametrů sdílené paměti operačního systému           | 8-2  |
| Maximální velikost segmentu sdílené paměti                                     | 8-2  |
| Použití více než 2 GB Paměti (systém Windows)                                  | 8-2  |
| Maximální počet identifikátorů sdílené paměti (systém UNIX)                    | 8-3  |
| Semafore (systém UNIX)                                                         | 8-3  |
| Nastavení konfiguračních parametrů sdílené paměti databázového serveru         | 8-3  |
| Nastavení parametrů pro rezidentní sdílenou paměť                              | 8-3  |
| Nastavení parametrů pro virtuální sdílenou paměť                               | 8-4  |
| Nastavení parametrů pro výkon sdílené paměti                                   | 8-4  |
| Nastavení parametrů sdílené paměti pomocí textového editoru                    | 8-5  |
| Nastavení parametrů sdílené paměti pomocí programu ISA                         | 8-5  |
| Nastavení parametrů sdílené paměti pomocí programu ON-Monitor (v systému UNIX) | 8-5  |
| Nastavení parametrů mezipaměti příkazů jazyka SQL                              | 8-6  |
| Nastavení sdílené paměti                                                       | 8-6  |
| Vypnutí nebo zapnutí rezidence rezidentní sdílené paměti                       | 8-6  |
| Zapínání a vypínání rezidence v režimu online                                  | 8-7  |
| Zapnutí a vypnutí rezidence při restartování databázového serveru              | 8-7  |
| Přidání segmentu k virtuální části sdílené paměti                              | 8-7  |
| Monitorování sdílené paměti                                                    | 8-7  |
| Monitorování segmentů sdílené paměti                                           | 8-7  |
| Monitorování profilu sdílené paměti a zámků latch                              | 8-8  |
| Použití obslužných programů příkazového řádku                                  | 8-8  |
| Použití programu IBM Informix Server Administrator                             | 8-8  |
| Použití programu ON-Monitor (systém UNIX)                                      | 8-8  |
| Použití tabulek SMI                                                            | 8-8  |
| Monitorování vyrovnávacích pamětí                                              | 8-8  |
| Použití obslužných programů příkazového řádku                                  | 8-9  |
| Použití programu ON-Monitor (systém UNIX)                                      | 8-10 |
| Použití tabulek SMI                                                            | 8-10 |
| Monitorování aktivity společné oblasti vyrovnávací paměti                      | 8-10 |
| Použití obslužných programů příkazového řádku                                  | 8-10 |
| Použití tabulek SMI                                                            | 8-11 |

---

### Obsah kapitoly

Tato kapitola informuje o tom, jak provádět následující úlohy, které se týkají správy sdílené paměti:

- Nastavení konfiguračních parametrů sdílené paměti.
- Nastavení sdílené paměti
- Vypnutí nebo zapnutí rezidence rezidentní části sdílené paměti databázového serveru.
- Přidání segmentu k virtuální části sdílené paměti.
- Monitorování sdílené paměti.

Tato kapitola se netýká konfiguračního parametru DS\_TOTAL\_MEMORY. Tento parametr nastavuje horní hranici přidělování paměti pro dotazy pro podporu rozhodování. Informace o tomto parametru najdete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

---

## Nastavení konfiguračních parametrů sdílené paměti operačního systému

Některé konfigurační parametry operačního systému mohou mít vliv na to, jak databázový server používá sdílenou paměť. Názvy parametrů nejsou uvedeny, protože názvy se liší u různých platform, a ne všechny parametry existují ve všech platformách. Následující seznam popisuje parametry podle jejich funkce:

- Maximální velikost segmentu sdílené paměti operačního systému vyjádřená v bajtech nebo kilobajtech.
- Minimální velikost segmentu sdílené paměti vyjádřená v bajtech.
- Maximální počet identifikátorů sdílené paměti.
- Adresa spodní hranice pro sdílenou paměť.
- Maximální počet segmentů sdílené paměti, které lze připojit k jedinému procesu.
- Maximální množství sdílené paměti na úrovni operačního systému.

### Jen pro UNIX

- Maximální počet identifikátorů semaforů.
- Maximální počet semaforů.
- Maximální počet semaforů na jeden identifikátor.

Soubor Poznámky k počítači v systému UNIX obsahuje doporučené hodnoty, které slouží ke konfiguraci zdrojů operačního systému. Tyto doporučené hodnoty použijte při konfiguraci operačního systému. Další informace o tom, jak nastavit parametry operačního systému najdete v dokumentaci k operačnímu systému.

Konkrétní informace o prostředí daného operačního systému najdete v souboru Poznámky k počítači, který je dodáván společně s databázovým serverem.

### Konec Jen pro UNIX

## Maximální velikost segmentu sdílené paměti

Databázový server se při vytváření požadovaných segmentů sdílené paměti pokouší získat od operačního systému segment největší možné velikosti. Velikost prvního segmentu, který se databázový server pokouší získat, je rovna velikosti části, kterou přiděluje (rezidentní, virtuální nebo komunikační) zaokrouhlené na nejbližší násobek 8 kB.

Databázový server obdrží chybu od operačního systému, jestliže požadovaný segment překročí maximální povolenou velikost. Jestliže databázový server obdrží chybu, dělí požadovanou velikost dvěma a udělá nový pokus. Pokusy o získání pokračují, dokud nebude vytvořen největší možný segment, jehož velikost je násobkem 8 kilobajtů. Databázový server potom vytvoří požadovaný počet dalších segmentů.

### Použití více než 2 GB Paměti (systém Windows)

V systému Windows může databázový server přistupovat k segmentům sdílené paměti větším než 2 GB. Tuto vlastnost je však nutné povolit pomocí položky v zaváděcím souboru systému Windows.

Tuto položku je možné přidat editováním zaváděcího souboru boot.ini (je umístěn v nejvyšším nebo v kořenovém adresáři). Je možné buď přidat novou zaváděcí volbu, nebo použít aktuálně existující zaváděcí volbu. Pro povolení podpory více než dvou gigabajtů je zapotřebí přidat na konec zaváděcího řádku následující text:

/3GB

V následujícím příkladu je povolena podpora více 2 GB:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operační systémy]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows NT
Workstation Version 4.00"
/3GB
```

Maximální velikost segmentu sdílené paměti závisí na operačním systému, ale v operačním systému Windows jsou to přibližně 3 GB, bez dalších ovladačů.

### Maximální počet identifikátorů sdílené paměti (systém UNIX)

Identifikátory sdílené paměti ovlivňují operace databázového serveru, když se virtuální procesor pokouší připojit ke sdílené paměti. Operační systém identifikuje každý segment sdílené paměti pomocí identifikátoru sdílené paměti. U většiny operačních systémů získávají virtuální procesory identifikátory na základě pravidla obslužení prvního požadavku až do limitu, který je definován pro operační systém jako celek. Další informace o identifikátorech sdílené paměti najdete v části “Jak se ke sdílené paměti připojují virtuální procesory” na stránce 7-6.

Maximální množství sdílené paměti, kterou operační systém může přidělit, se vypočítá jako součin počtu identifikátorů sdílené paměti a maximální velikosti segmentu sdílené paměti.

## Semafore (systém UNIX)

Činnost databázového serveru vyžaduje jeden semafor systému UNIX pro každý virtuální procesor, jeden pro každého uživatele, který se k databázovému serveru připojuje prostřednictvím sdílené paměti (protokol **ipcshm**), šest pro obslužné programy databázového serveru a šestnáct pro ostatní účely.

---

## Nastavení konfiguračních parametrů sdílené paměti databázového serveru

Konfigurační parametry sdílené paměti spadají do následujících kategorií podle svého účelu:

- Parametry, které ovlivňují rezidentní část sdílené paměti.
- Parametry, které ovlivňují virtuální část sdílené paměti.
- Parametry, které ovlivňují výkon.

K nastavení konfiguračních parametrů sdílené paměti je možné použít následující prostředky:

- textový editor
- IBM Informix Server Administrator (ISA)

---

### Jen pro UNIX

- ON-Monitor

### Konec Jen pro UNIX

---

V systému UNIX, může tuto metodu použít pouze uživatel **root** nebo uživatel **informix**. V systému Windows je nutné být členem skupiny Informix Admin.

## Nastavení parametrů pro rezidentní sdílenou paměť

Tabulka Tabulka 8-1 obsahuje seznam parametrů v souboru ONCONFIG, které určují konfiguraci společné oblasti vyrovnávací paměti a vnitřních tabulek v rezidentní části

sdílené paměti. Změny konfiguračních parametrů se projeví až po vypnutí a opětovném spuštění databázového serveru. Popis konfiguračních parametrů najdete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Tabulka 8-1. Konfigurace rezidentní části sdílené paměti

| Konfigurační parametr | Účel                                                                                                                                                   |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| BUFFERPOOL            | Určuje informace o společné oblasti vyrovnávací paměti, které je třeba definovat pro každou jednotlivou velikost stránky používanou prostorem dbspace. |
| LOCKS                 | Určuje počáteční počet zámek databázových objektů; například řádků, klíčových hodnot, stránek a tabulek.                                               |
| LOGBUFF               | Určuje velikost vyrovnávacích pamětí logických protokolů.                                                                                              |
| PHYSBUFF              | Určuje velikost vyrovnávacích pamětí fyzických protokolů.                                                                                              |
| RESIDENT              | Určuje rezidence rezidentní části sdílené paměti databázového serveru.                                                                                 |
| SERVERNUM             | Určuje jedinečné identifikační číslo databázového serveru v místním hostitelském počítači.                                                             |
| SHMTOTAL              | Určuje celkovou paměť, která bude využita databázovým serverem.                                                                                        |

## Nastavení parametrů pro virtuální sdílenou paměť

Tabulka Tabulka 8-2 obsahuje seznam parametrů souboru ONCONFIG, které slouží ke konfiguraci virtuální části sdílené paměti. Další informace naleznete v kapitole o vlivu konfigurace na paměť, v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Tabulka 8-2. Konfigurace virtuální části sdílené paměti

| Konfigurační parametr | Účel                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DS_HASHSIZE           | Počet sektorů hashovací tabulky pro seznamy v mezipaměti distribuce dat.                                                                                                                                                                                                                                                                                                                           |
| DS_POOLSIZE           | Maximální počet položek v mezipaměti distribuce dat.                                                                                                                                                                                                                                                                                                                                               |
| PC_HASHSIZE           | Určuje počet segmentů hashovací tabulky pro mezipaměť rutin UDR a další mezipaměti používané databázovým serverem. Další informace o nastavení parametru PC_HASHSIZE naleznete v příručce <i>Příručka výkonosti serveru IBM Informix Dynamic Server</i> .                                                                                                                                          |
| PC_POOLSIZE           | Určuje počet uživatelských rutin (rutin SPL a externích rutin), které je možno uložit v mezipaměti uživatelských rutin. Tento parametr určuje také velikost dalších mezipamětí databázového serveru, jako je mezipaměť názvů typů a mezipaměť tříd operátorů. Další informace o nastavení parametru PC_POOLSIZE najdete v příručce <i>Příručka výkonosti serveru IBM Informix Dynamic Server</i> . |
| SHMADD                | Určuje velikost dynamicky přidávaných segmentů sdílené paměti.                                                                                                                                                                                                                                                                                                                                     |
| EXTSHMADD             | Určuje velikost přidaného rozšiřujícího segmentu.                                                                                                                                                                                                                                                                                                                                                  |
| SHMTOTAL              | Určuje celkovou paměť, která bude využita databázovým serverem.                                                                                                                                                                                                                                                                                                                                    |
| SHMVIRTSIZE           | Určuje počáteční velikost virtuální části sdílené paměti.                                                                                                                                                                                                                                                                                                                                          |
| STACKSIZE             | Určuje velikost zásobníku uživatelských jednotkových procesů databázového serveru.                                                                                                                                                                                                                                                                                                                 |

## Nastavení parametrů pro výkon sdílené paměti

Tabulka Tabulka 8-3 obsahuje parametry souboru ONCONFIG, které nastavují volby výkonu sdílené paměti. Další informace najdete v kapitole o konfiguračních parametrech v příručce *IBM Informix Dynamic Server Administrator's Reference*.



Tabulka 8-3. Nastavení voleb výkonu sdílené paměti

| Konfigurační parametr | Účel                                                                                                                                                                                                                             |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CKPTINTVL             | Určuje maximální počet sekund, které mohou uplynout, než databázový server zkontroluje, zda je nutný kontrolní bod a zda není konfigurační parametr RTO_SERVER_RESTART nastaven, aby zapnul automatické ladění kontrolních bodů. |
| CLEANERS              | Určuje počet jednotkových procesů čištění stránky, které jsou v databázovém serveru spuštěny.                                                                                                                                    |
| parametr RA_PAGES     | Určuje počet diskových stránek, které by se měl databázový server pokusit číst napřed při provádění sekvenčního prohledávání dat nebo indexovaných záznamů.                                                                      |
| parametr RA_THRESHOLD | Určuje počet nezpracovaných paměťových stránek, které po přečtení způsobují, že databázový server čte napřed na disk.                                                                                                            |

## Nastavení parametrů sdílené paměti pomocí textového editoru

Pomocí textového editoru lze nastavit konfigurační parametry pro rezidentní a virtuální sdílenou paměť a pro výkon sdílené paměti. Vyhleďte parametr v souboru ONCONFIG, zadejte nové hodnoty a zapište soubor znovu na disk. Změny se projeví až po vypnutí a novém spuštění databázového serveru.

## Nastavení parametrů sdílené paměti pomocí programu ISA

Pomocí programu ISA můžete monitorovat a nastavovat následující parametry sdílené paměti. Další informace naleznete v nápovědě online programu ISA

- Provádění příkazů obslužných programů, jako například **onmode** a **onstat**.
- Úpravy parametrů v souboru ONCONFIG.
- Monitorování segmentů.
- Monitorování společných oblastí paměti.
- Monitorování rezidentní paměti.
- Monitorování nerezidentní paměti.
- Monitorování mezipaměti datového slovníku.

## Nastavení parametrů sdílené paměti pomocí programu ON-Monitor (v systému UNIX)

Chcete-li nastavit konfigurační parametry pro rezidentní a virtuální část sdílené paměti pomocí programu ON-Monitor, vyberte volbu **Parameters > Shared-Memory**.

Chcete-li určit velikost stránky ve svém systému, vyberte v programu ON-Monitor volbu **Parameters> Shared-memory**. Velikost stránky databázového serveru je poslední položkou na stránce.

**Důležité:** Konfigurační parametry SHMADD, EXTSHMADD a SHMTOTAL ovlivňují rezidentní i virtuální části sdílené paměti.

Pokud chcete nastavit konfigurační parametry následujících voleb výkonu sdílené paměti programu ON-Monitor, vyberte volbu **Parameters > performance**:

- CKPTINTVL
- parametr RA\_PAGES
- parametr RA\_THRESHOLD

---

## Nastavení parametrů mezipaměti příkazů jazyka SQL

Tabulka 8-4 zobrazuje různé způsoby konfigurace mezipaměti příkazů jazyka SQL.

Tabulka 8-4. Konfigurace mezipaměti příkazů jazyka SQL

| Konfigurační parametr | Účel                                                                                                                                                                                       | Příkaz onmode                           |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| STMT_CACHE            | Zapne, povolí nebo zakáže mezipaměť příkazů jazyka SQL v paměti. Pokud je parametr zapnut, určuje, zda má mezipaměť příkazů jazyka SQL uchovávat analyzované a optimalizované příkazy SQL. | <b>onmode -e režim</b>                  |
| STMT_CACHE_HITS       | Určuje počet použití příkazu (odkazů na příkaz), po kterém bude příkaz zcela vložen do mezipaměti příkazů jazyka SQL.                                                                      | <b>onmode -W<br/>STMT_CACHE_HITS</b>    |
| STMT_CACHE_NOLIMIT    | Řídí, zda mají být příkazy jazyka SQL vkládány do mezipaměti příkazů i po překročení velikosti určené hodnotou parametru STMT_CACHE_SIZE.                                                  | <b>onmode -W<br/>STMT_CACHE_NOLIMIT</b> |
| STMT_CACHE_NUMPOOL    | Definuje počet společných oblastí mezipaměti příkazů jazyka SQL.                                                                                                                           | žádný                                   |
| STMT_CACHE_SIZE       | Určuje velikost mezipaměti příkazů jazyka SQL.                                                                                                                                             | žádný                                   |

Pomocí následujících voleb programu **onstat** lze monitorovat mezipaměť příkazů jazyka SQL:

- **onstat -g ssc** (stejně jako **onstat -g cac stmt**)
- **onstat -g ssc all**
- **onstat -g ssc pool**

Více informací o těchto konfiguračních parametrech, volbách **onstat -g** a příkazech **onmode** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Další informace o použití mezipaměti příkazů jazyka SQL, o jejím monitorování pomocí voleb **onstat -g** a o ladění konfiguračních parametrů naleznete v části týkající se zlepšení výkonu dotazů v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*. Podrobnosti o kvalifikačních a identických příkazech najdete v příručce *IBM Informix Guide to SQL: Syntax*.

---

## Nastavení sdílené paměti

Pokud chcete nastavit sdílenou paměť, převedte databázový server do režimu offline a pak do režimu online. Informace o tom, jak převést databázový server z režimu online do režimu offline, najdete v příručce “Okamžitá změna z jakéhokoli režimu do režimu offline” na stránce 4-14.

---

## Vypnutí nebo zapnutí rezidence rezidentní sdílené paměti

Rezidenci rezidentní části sdílené paměti můžete zapnout nebo vypnout následujícími dvěma způsoby:

- Pomocí obslužného programu **onmode** můžete změnit stav rezidence sdílené paměti, pokud se databázový server nachází v režimu online.
- Změnou parametru **RESIDENT** v souboru **ONCONFIG** můžete zapnout nebo vypnout rezidenci pro příští nastavení sdílené paměti databázového serveru.

Popis rezidentní části sdílené paměti najdete v části “Rezidentní část sdílené paměti” na stránce 7-9.

## Zapínání a vypínání rezidence v režimu online

Chcete-li rezidenci vypnout nebo zapnout, je-li databázový server v režimu online, použijte obslužný program **onmode**.

Chcete-li ihned zapnout rezidenci rezidentní části sdílené paměti, proveďte následující příkaz:

```
% onmode -r
```

Chcete-li rezidenci rezidentní části sdílené paměti ihned vypnout, proveďte následující příkaz:

```
% onmode -n
```

Tyto příkazy nemění hodnotu parametru **RESIDENT** v souboru **ONCONFIG**. To znamená, že tato změna není trvalá a rezidence se při příštím nastavení sdílené paměti vrátí do stavu určeného parametrem **RESIDENT**. V systému **UNIX** může rezidenci zapnout nebo vypnout pouze uživatel **root** nebo uživatel **informix**. V systému **Windows** může rezidenci zapnout nebo vypnout pouze uživatel, který je členem skupiny **Informix Admin**.

## Zapnutí a vypnutí rezidence při restartování databázového serveru

Rezidenci můžete zapnout nebo vypnout pomocí textového editoru. Chcete-li změnit aktuální stav rezidence, vyhledejte parametr **RESIDENT** pomocí textového editoru. Chcete-li rezidenci zapnout, nastavte parametr **RESIDENT** na hodnotu **1**, chcete-li rezidenci vypnout, nastavte tento parametr na hodnotu **0** a potom soubor znovu zapište na disk. Změny se projeví až po vypnutí a novém spuštění databázového serveru.

---

## Přidání segmentu k virtuální části sdílené paměti

Volba **-a** obslužného programu **onmode** umožňuje přidat k virtuální sdílené paměti segment určené velikosti.

Obvykle není nutné přidávat segmenty k virtuální sdílené paměti, protože databázový server automaticky přidává segmenty podle potřeby.

Možnost přidat segment pomocí obslužného programu **onmode** je užitečná, je-li počet segmentů operačního systému omezený a je-li velikost počátečního segmentu tak malá vzhledem k požadované velikosti, že je téměř překročeno omezení operačního systému pro segmenty sdílené paměti.

---

## Monitorování sdílené paměti

Tato část popisuje monitorování segmentů sdílené paměti, profil sdílené paměti a použití určitých zdrojů sdílené paměti (vyrovnávacích pamětí, zámků latch a zámků).

Pomocí obslužného programu **onstat -o** lze získat statický snímek sdílené paměti databázového serveru pro pozdější analýzu a porovnání.

## Monitorování segmentů sdílené paměti

Monitorování segmentů sdílené paměti umožňuje určit počet a velikost segmentů, které databázový server vytváří. Databázový server přiděluje segmenty sdílené paměti dynamicky, takže se jejich počet může měnit. Jestliže databázový server přiřazuje příliš mnoho segmentů sdílené paměti, můžete zvýšit konfigurační parametr **SHMVIRTSIZE**. Další informace najdete v kapitole o konfiguračních parametrech v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Příkaz **onstat -g seg** zobrazí informace o každém segmentu sdílené paměti, včetně adresy a velikosti segmentu a množství volné a používané paměti. Příklad výstupu příkazu **onstat -g seg** naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Administrator's Reference*.

## Monitorování profilu sdílené paměti a zámků latch

Monitorování profilu databázového serveru umožňuje analyzovat výkon a použití zdrojů sdílené paměti. Obrazovka profilu udržuje kumulativní statistiky o používání sdílené paměti. Chcete-li tyto statistiky vynulovat, použijte příkaz **onstat -z**. Popis ostatních polí, která obslužný program **onstat** zobrazuje, naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Můžete získat statistické údaje o použití zámků latch a informace o určitých zámcích latch. Tyto statistické údaje udávají měřítko aktivity systému.

### Použití obslužných programů příkazového řádku

Monitorování sdílené paměti a zámků latch můžete provádět pomocí následujících obslužných programů příkazového řádku:

- **onstat -s**
- **onstat -p**

**onstat -s:** Pomocí příkazu **onstat -s** lze získat informace o zámcích latch.

**onstat -p:** Provedením příkazu **onstat -p** zobrazíte statistické údaje aktivity databázového serveru a čekajících zámků latch (v poli **lchwaits**). Příklad výstupu příkazu **onstat -p** naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Administrator's Reference*.

### Použití programu IBM Informix Server Administrator

Pomocí programu ISA lze získat informace o zámcích latch, zámcích spin a profilech.

### Použití programu ON-Monitor (systém UNIX)

Vyberte volbu **Status > Profile**. Na obrazovce se zobrazí statistika sdílené paměti, aktuální provozní režim, čas zavedení, aktuální čas a zámky latch.

### Použití tabulek SMI

Dotazováním na tabulku **sysprofile** získáte statistiky sdílené paměti. Tato tabulka obsahuje všechny statistiky dostupné ve výstupu příkazu **onstat -p**, s výjimkou statistik **ovbuff**, **usercpu** a **syscpu**.

## Monitorování vyrovnávacích pamětí

Můžete získat statistiky o použití vyrovnávacích pamětí i informace o určitých vyrovnávacích pamětech. Statistické informace obsahují údaje o tom, kolik procent datových zápisů prochází mezipamětí a kolikrát musely jednotkové procesy čekat, než získaly vyrovnávací paměť. Procento zápisů v mezipaměti je důležitým měřítkem výkonu. (Informace o tom, jak využít tuto statistiku k vyladění databázového serveru, naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.)

Počet čekání na vyrovnávací paměti je měřítkem souběžného zpracování v systému.

Údaje o určitých vyrovnávacích pamětech obsahují seznam všech vyrovnávacích pamětí ve sdílené paměti, které jsou používány jednotkovými procesy. Tyto údaje umožňují sledovat stav jednotlivých vyrovnávacích pamětí. Lze například zjistit, zda na danou vyrovnávací paměť čeká jiný jednotkový proces.

## Použití obslužných programů příkazového řádku

Monitorování vyrovnávacích pamětí můžete provádět pomocí následujících obslužných programů příkazového řádku:

- **onstat -p**
- **onstat -B**
- **onstat -b**
- **onstat -X**
- **onstat -R**

**onstat -p:** Pomocí příkazu **onstat -p** můžete získat statistiku o zápisech a čteních ukládaných do mezipaměti. Následující statistika ukládání do mezipaměti se objeví ve čtyřech polích v horním řádku výstupní obrazovky:

- Počet čtení z mezipaměti ve sdílené paměti (**bufreads**).
- Procento čtení z mezipaměti (**%cached**).
- Počet zápisů do sdílené paměti (**bufwrits**).
- Procento zápisů do mezipaměti (**%cached**).
- Informace o obecných stránkách (tj. nestandardních stránkách ve společné oblasti vyrovnávací paměti).

Počet čtení a zápisů ve výstupu se může zobrazovat jako záporné číslo, pokud počet výskytů přesáhne hodnotu  $2^{32}$  (závisí na platformě).

Volba **onstat -p** zobrazí také statistiku (**bufwaits**), která ukazuje, kolikrát musely tyto relace čekat na vyrovnávací paměť.

Příklad výstupu příkazu **onstat -p** naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Administrator's Reference*.

**onstat -B:** Provedením příkazu **onstat -B** získáte informace o všech aktuálně používaných vyrovnávacích pamětech včetně:

- Adresy každé běžné vyrovnávací paměti sdílené paměti.
- Adresy jednotkového procesu, který vyrovnávací paměť právě využívá.
- Adresy prvního jednotkového procesu, který čeká na každou vyrovnávací paměť.
- Informace o společných oblastech vyrovnávací paměti.

Příklad výstupu příkazu **onstat -B** naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Administrator's Reference*.

**onstat -b:** Provedením příkazu **onstat -b** jsou pro každou vyrovnávací paměť zobrazeny následující informace:

- Adresa každé vyrovnávací paměti, kterou je právě používána jednotkovým procesem.
- Číslo stránek pro stránku, která je právě uložena ve vyrovnávací paměti.
- Typ stránky, která je právě uložena ve vyrovnávací paměti (například datová stránka, stránka prostoru tblspace atd.).
- Typ zámku, kterým je vyrovnávací paměť uzamčena (výhradní nebo sdílený).
- Adresa jednotkového procesu, který vyrovnávací paměť právě používá.
- Adresa prvního jednotkového procesu, který čeká na každou vyrovnávací paměť.
- Informace o společných oblastech vyrovnávací paměti.

Porovnáním adres uživatelských jednotkových procesů s adresami, které se zobrazí po provedení příkazu **onstat -u**, můžete získat číslo ID relace.

Další informace o polích, která obslužný program **onstat** zobrazuje, naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Dynamic Server Administrator's Reference*

**onstat -X:** Provedením příkazu **onstat -X** můžete získat stejné údaje jako provedením příkazu **onstat -b** a také *úplný* seznam všech jednotkových procesů, které čekají na vyrovnávací paměti, ne pouze první čekající jednotkový proces.

**onstat -R:** Pomocí příkazu **onstat -R** můžete zobrazit informace o společné oblasti vyrovnávací paměti včetně informací o vyrovnávacích pamětech.

## Použití programu ON-Monitor (systém UNIX)

Pokud chcete přistupovat k polím zmiňovaným na stránce 8-9 u příkazu **onstat -p (bufreads, %cached, bufwrits %cached)**, vyberte volbu **Status > Profile**.

Následující příklad ukazuje statistiku čtení a zápisů uložených v mezipaměti, které jsou zobrazeny pomocí volby **Profile** v nabídce **Status** programu ON-Monitor.

```
...
Disk Reads Buff. Reads %Cached Disk Writes Buff. Writes %Cached
 177 330 46.36 4 0 0.00
...
```

## Použití tabulek SMI

Dotazováním na tabulku **sysprofile** získáte statistiku čtení a zápisů v mezipaměti a celkový počet čekání na mezipaměť. Důležité jsou následující řádky.

| Řádek            | Popis                                                         |
|------------------|---------------------------------------------------------------|
| <b>dskreads</b>  | Počet čtení z disku.                                          |
| <b>bufreads</b>  | Počet čtení z vyrovnávací paměti.                             |
| <b>dskwrites</b> | Počet zápisů na disk.                                         |
| <b>bufwrites</b> | Počet zápisů do vyrovnávacích pamětí.                         |
| <b>buffwts</b>   | Počet čekání všech jednotkových procesů na vyrovnávací paměť. |

## Monitorování aktivity společné oblasti vyrovnávací paměti

Můžete získat statistiku, která se vztahuje k dostupnosti vyrovnávací paměti, a také informace o vyrovnávacích pamětech v každé frontě LRU.

Statistické údaje obsahují počet pokusů databázového serveru o překročení maximálního počtu vyrovnávacích pamětí a počet zápisů na disk (seřazených podle události, která způsobila vyprázdnění vyrovnávacích pamětí). Tyto statistické údaje pomáhají zjistit, zda je počet vyrovnávacích pamětí vhodný. Informace o ladění vyrovnávacích pamětí databázového serveru najdete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Informace o vyrovnávacích pamětech v každé frontě LRU se skládá z délky fronty a procenta změněných vyrovnávacích pamětí ve frontě.

## Použití obslužných programů příkazového řádku

Informace o aktivitě společné oblasti vyrovnávací paměti můžete získat pomocí obslužného programu **onstat**. Volby obslužného programu **onstat** lze také používat z programu ISA.

Další informace o volbách obslužného programu **onstat** naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Dynamic Server Administrator's Reference*.

**onstat -p:** Výstup příkazu **onstat -p** obsahuje statistiku (**ovbuff**), která ukazuje, kolikrát se databázový server pokoušel překročit maximální počet sdílených vyrovnávacích pamětí, který je určen hodnotou **buffers** konfiguračního parametru **BUFFERPOOL**.

**onstat -F:** Provedením příkazu **onstat -F** můžete získat počet provedených zápisů podle typu zápisu. (Vysvětlení různých typů zápisů naleznete v části “Popis aktivity vyprazdňování” na stránce 7-28.)

Příkaz **onstat -F** zobrazí celkové počty následujících typů zápisu:

- zápis na popředí
- zápis front LRU
- zápis bloku.

Pomocí příkazu **onstat -F** jsou také zobrazeny následující informace o jednotkových procesech čištění stránek:

- číslo jednotkového procesu čištění stránek
- adresa sdílené paměti jednotkového procesu čištění stránek
- aktuální stav jednotkového procesu čištění stránek
- fronta LRU, k níž byl jednotkový proces čištění stránek přiřazen

Příklad výstupu příkazu **onstat -F** naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Administrator's Reference*.

**onstat -R:** Provedením příkazu **onstat -R** můžete získat informace o počtu vyrovnávacích pamětí v každé frontě LRU a počtu a procentu změněných nebo uvolněných vyrovnávacích pamětí.

Příklad výstupu příkazu **onstat -R** naleznete v informacích o obslužném programu **onstat** v příručce *IBM Informix Administrator's Reference*

## Použití tabulek SMI

Dotazováním na tabulku **sysprofile** můžete získat statistiku typů zápisu, které se nacházejí v následujících řádcích.

| <b>Řádek</b>       | <b>Popis</b>             |
|--------------------|--------------------------|
| <b>fgwrites</b>    | Počet zápisů na popředí. |
| <b>lruwrites</b>   | Počet zápisů front LRU.  |
| <b>chunkwrites</b> | Počet zápisů bloků.      |





---

## Kapitola 9. Uložení dat

|                                                                                        |      |
|----------------------------------------------------------------------------------------|------|
| Obsah kapitoly . . . . .                                                               | 9-2  |
| Fyzické a logické paměťové jednotky . . . . .                                          | 9-2  |
| Bloky . . . . .                                                                        | 9-3  |
| Přidělování disků blokům . . . . .                                                     | 9-3  |
| Přístup k disku v systému Windows . . . . .                                            | 9-3  |
| Přístup k disku s vyrovnávací pamětí a bez vyrovnávací paměti v systému UNIX . . . . . | 9-4  |
| Posuny . . . . .                                                                       | 9-5  |
| Stránky . . . . .                                                                      | 9-5  |
| Stránky blobpage . . . . .                                                             | 9-6  |
| Stránky sbpage . . . . .                                                               | 9-7  |
| Oblasti . . . . .                                                                      | 9-8  |
| Prostory dbspace . . . . .                                                             | 9-9  |
| Řízení umístění ukládaných dat . . . . .                                               | 9-9  |
| Kofenový prostor dbspace . . . . .                                                     | 9-11 |
| Dočasné prostory dbspace . . . . .                                                     | 9-11 |
| Prostory blobspace . . . . .                                                           | 9-12 |
| Prostory sbpace . . . . .                                                              | 9-12 |
| Výhody používání prostorů sbpace . . . . .                                             | 9-13 |
| Prostory sbpace a replikace Enterprise Replication . . . . .                           | 9-13 |
| Metadata, uživatelská data a rezervovaná oblast . . . . .                              | 9-13 |
| Řízení umístění ukládaných dat . . . . .                                               | 9-14 |
| Paměťové charakteristiky prostorů sbpace . . . . .                                     | 9-15 |
| Velikosti oblastí v prostorech sbpace . . . . .                                        | 9-15 |
| Průměrná velikost inteligentních velkých objektů . . . . .                             | 9-16 |
| Režim použití vyrovnávací paměti . . . . .                                             | 9-16 |
| Čas posledního přístupu . . . . .                                                      | 9-16 |
| Režim uzamčení . . . . .                                                               | 9-17 |
| Protokolování . . . . .                                                                | 9-17 |
| Úroveň dědičnosti charakteristik prostorů sbpace . . . . .                             | 9-17 |
| Další informace o prostorech sbpace . . . . .                                          | 9-18 |
| Dočasné prostory sbpace . . . . .                                                      | 9-19 |
| Porovnání dočasných a standardních prostorů sbpace . . . . .                           | 9-20 |
| Dočasné inteligentní velké objekty . . . . .                                           | 9-20 |
| Prostory extspace . . . . .                                                            | 9-21 |
| Databáze . . . . .                                                                     | 9-21 |
| Tabulky . . . . .                                                                      | 9-22 |
| Poškozené tabulky . . . . .                                                            | 9-23 |
| Typy tabulek serveru Dynamic Server . . . . .                                          | 9-24 |
| Standardní trvalé tabulky . . . . .                                                    | 9-24 |
| Tabulky typu RAW . . . . .                                                             | 9-24 |
| Tabulky typu TEMP . . . . .                                                            | 9-25 |
| Vlastnosti typů tabulek . . . . .                                                      | 9-25 |
| Zavedení dat do tabulky . . . . .                                                      | 9-25 |
| Rychlá obnova jednotlivých typů tabulek . . . . .                                      | 9-25 |
| Zálohování a obnovení tabulek typu RAW . . . . .                                       | 9-26 |
| Dočasné tabulky . . . . .                                                              | 9-26 |
| Uživatelské dočasné tabulky . . . . .                                                  | 9-27 |
| Dočasné tabulky vytvářené databázovým serverem . . . . .                               | 9-28 |
| Prostory tblspace . . . . .                                                            | 9-28 |
| Maximální počet prostorů tblspace v tabulce . . . . .                                  | 9-29 |
| Prostory tblspace tabulek a indexů . . . . .                                           | 9-29 |
| Prokládání oblastí . . . . .                                                           | 9-30 |
| Fragmentace tabulek a uložení dat . . . . .                                            | 9-30 |
| Objem diskového prostoru potřebný k uložení dat . . . . .                              | 9-31 |
| Velikost kofenového prostoru dbspace . . . . .                                         | 9-32 |

|                                                         |      |
|---------------------------------------------------------|------|
| Fyzické a logické protokoly . . . . .                   | 9-32 |
| Dočasné tabulky. . . . .                                | 9-33 |
| Kritická data . . . . .                                 | 9-33 |
| Další prostor . . . . .                                 | 9-33 |
| Objem prostoru požadovaného databázemi . . . . .        | 9-33 |
| Pravidla pro rozvržení disků . . . . .                  | 9-33 |
| Pravidla pro prostory dbspace a bloky . . . . .         | 9-34 |
| Pravidla umístování tabulek . . . . .                   | 9-34 |
| Ukázková rozvržení disku . . . . .                      | 9-35 |
| Ukázkové rozvržení upřednostňující výkon. . . . .       | 9-36 |
| Ukázkové rozvržení upřednostňující dostupnost . . . . . | 9-37 |
| Správce logických disků . . . . .                       | 9-38 |

---

## Obsah kapitoly

Tato kapitola definuje termíny a vysvětluje koncepty, jejichž znalost je nezbytná k provádění úloh, které popisuje Kapitola 10, “Správa diskového prostoru”, na stránce 10-1. V této kapitole jsou popsána následující témata:

- Definice logických a fyzických jednotek, které bude databázový server používat k ukládání dat.
- Pokyny k výpočtu diskového prostoru potřebného k uložení dat.
- Obecné pokyny k určení rozvržení diskového prostoru a rozmístění databází a tabulek.

Jakékoli doplňující informace týkající se maximálních hodnot platných pro paměťové jednotky popisované v této kapitole naleznete v poznámkách k verzi serveru Dynamic Server.

---

## Fyzické a logické paměťové jednotky

Databázový server používá k přidělování diskového prostoru fyzické jednotky paměti. Na rozdíl od logických paměťových jednotek, jejichž velikost je proměnná, má každá fyzická paměťová jednotka pevnou nebo přiřazenou velikost určenou architekturou disku. Databázový server používá ke správě diskového prostoru následující fyzické jednotky:

- blok
- stránka
- oblast
- stránka blobpage
- stránka sbpage

Databázový server ukládá data do následujících logických jednotek:

- prostor dbspace
- dočasný prostor dbspace
- prostor blobspace
- prostory sbpace
- dočasný prostor sbpace
- prostor extspace
- databáze
- tabulka
- prostor tblspace
- oddíl

K zachování logické a fyzické konzistence dat udržuje databázový server následující struktury:

- logický protokol
- fyzický protokol
- rezervované stránky

V následujících částech jsou popsány různé jednotky uložení dat podporované databázovým serverem a vztahy mezi těmito jednotkami. Informace o rezervovaných stránkách naleznete v kapitole o diskových strukturách a diskové paměti v příručce *IBM Informix Dynamic Server Administrator's Reference*.

---

## Bloky

*Blok* je největší jednotka fyzického disku vyhrazená k uložení dat databázového serveru. Bloky umožňují administrátorům přidělovat diskový prostor v jednotkách značné velikosti. Maximální velikost bloku je 4 TB. Maximální počet povolených bloků je 32766. Maximální povolenou velikost a počet bloků je nejprve zapotřebí povolit pomocí příkazu `onmode -BC`. Pokud nebude spuštěn příkaz `-BC`, bude maximální velikost bloku omezena na 2 GB.

Pokud se zaplnění následujících paměťových prostorů bude blížit jejich úplné kapacitě, přidá administrátor databázového serveru k těmto prostorům další bloky:

- Prostory `dbspace` (viz strana 9-9).
- Prostory `blobspace` (viz strana 9-12).
- Prostory `sbspace` (viz strana 9-12).

Informace o názvech, velikostech a počtech bloků naleznete v části “Určení názvů paměťových prostorů a bloků” na stránce 10-7 a v části “Určení maximální velikosti bloků” na stránce 10-8.

Databázový server používá bloky také k zrcadlení. *Primární blok* je blok, ze kterého databázový server kopíruje data do *zrcadleného bloku*. Pokud dojde k selhání primárního bloku, převede databázový server zrcadlený blok automaticky do režimu online. Další informace naleznete v části kapitola 17, “Zrcadlení”, na stránce 17-1.

## Přidělování disků blokům

Databázový server může k ukládání dat používat běžné soubory operačního systému nebo *přímá disková zařízení*. Pokud je výkon důležitým faktorem, je v systému UNIX UNIX, doporučeno k ukládání dat používat přímá disková zařízení. V systému Windows je z důvodu snazší administrace doporučeno k ukládání používat soubory souborového systému NTFS.

Paměťový prostor Informix se může nacházet v systému souborů NFS za použití běžného operačního systému.

### Přístup k disku v systému Windows

V systému Windows používá databázový server při práci s přímými disky i soubory NTFS rozhraní jádra pro asynchronní operace vstupu - výstupu (KAIO). Správa souborů systému Windows zatěžuje operace diskového vstupu - výstupu jistou režii, proto přináší použití přímých disků mírné zvýšení výkonu. Protože soubory NTFS představují standardnější způsob ukládání dat, je doporučeno místo přímých disků používat k ukládání dat soubory systému NTFS. Použití přímých disků zvažte tedy, pokud databázový server vyžaduje velký počet přístupů na disk.

**Diskový prostor s přímým přístupem v systému Windows:** V systému Windows může *diskový prostor s přímým přístupem* být fyzická jednotka bez přiřazeného písmene jednotky nebo logický diskový oddíl, kterému bylo pomocí nástroje **Správce disků** přiřazeno písmeno

jednotky. Prostor může být formátovaný nebo neformátovaný. Pokud obsahuje data, budou tato data po přidělení prostoru databázovému serveru přepsána. Další informace naleznete v části “Přidělení diskového prostoru s přímým přístupem v systému Windows” na stránce 10-6.

**Soubory systému NTFS:** V systému Windows musejí být pro diskový prostor použity soubory systému NTFS, nikoli soubory systému FAT. Další informace naleznete v části “Přidělení prostoru souborům systému NTFS v systému Windows” na stránce 10-6.

## **Přístup k disku s vyrovnávací pamětí a bez vyrovnávací paměti v systému UNIX**

Diskový prostor můžete přidělit dvěma způsoby:

- Pomocí souborů, které operační systém při přístupu ukládá do vyrovnávací paměti. Tyto soubory se také nazývají *předpřipravené soubory*.
- Pomocí přístupu k disku bez vyrovnávací paměti a diskového prostoru s *přímým* přístupem.

Pokud jsou prostory dbspace umístěny na *diskových zařízeních s přímým přístupem* (také nazývaných *znakově orientovaná zařízení*), použijte databázový server přístup k disku bez vyrovnávací paměti.

Pokud chcete vytvořit přímé zařízení, nakonfigurujte *blokové zařízení* (pevný disk) s použitím přímého rozhraní. Paměťový prostor poskytovaný tímto zařízením se nazývá *diskový prostor s přímým přístupem*. Blok diskového prostoru s přímým přístupem je fyzicky souvislý.

Název bloku je název znakově orientovaného souboru v adresáři **/dev**. V mnoha operačních systémech lze rozlišit znakově orientované soubory a blokově orientované soubory pomocí prvního písmene názvu souboru (typicky *r*). Například soubor **/dev/rsd0f** představuje znakově orientované zařízení, kterému odpovídá blokově orientované zařízení **/dev/sd0f**.

Další informace naleznete v části “Přidělení diskového prostoru s přímým přístupem v systému UNIX” na stránce 10-5.

*Předpřipravený soubor* je běžný soubor spravovaný operačním systémem. Spolehlivost souborových bloků a bloků přímých diskových zařízení se neliší. Na rozdíl od diskového prostoru s přímým přístupem nemusejí logicky souvislé bloky předpřipraveného souboru být souvislé i fyzicky.

Předpřipravené soubory lze přidělit snáze než diskový prostor s přímým přístupem. K přidělení předpřipraveného souboru je zapotřebí pouze vytvořit soubor na některém z existujících oddílů. Název bloku je úplný název cesty k souboru. Tyto kroky jsou podrobněji popsány v části “Přidělení prostorů předpřipravených souborů v systému UNIX” na stránce 10-4.

Ve výukovém prostředí, kde není výkon důležitý, nebo k uložení statických dat mohou být předpřipravené soubory snáze použitelné. Pokud musíte použít předpřipravené UNIX soubory, uložte do těchto souborů data, ke kterým se přistupuje nejméně. Tyto soubory uložte do souborového systému s minimální aktivitou.

V případě bloků v předpřipravených souborech zpracovává operační systém všechny operace *vstupu - výstupu* s využitím vlastní společné oblasti vyrovnávacích pamětí a zaručuje, že všechny zápisy do bloků budou fyzicky zapsány na disk.

**Důležité:** Přestože doporučujeme v rámci dosažení vyššího výkonu používat v systému UNIX přímá disková zařízení, pokud povolíte konfigurační parametr **DIRECT\_IO**, výkon předpřipravených souborů se může blížit výkonu přímých zařízení používaných u bloků prostoru dbspace. K tomu dochází, protože přímý vstup -

výstup obchází použití vyrovnávacích pamětí systému souborů. Další informace naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Další informace naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Chcete-li zjistit, jak dosáhnout nejlepšího výkonu zařízení, změřte výkon systému s daným rozvržením prostorů dbspace a tabulek při použití obou typů zařízení.

Pokud používáte přímé disky, není zapotřebí při vytváření bloků a souborů větších než 2 GB provádět jakékoli zvláštní akce. Pokud chcete vytvářet velké bloky v předpřipravených souborech nebo pokud chcete společně s velkými soubory používat obslužné programy k importu a exportu, přesvědčte se, zda jsou správně nakonfigurovány souborové systémy, ve kterých budou velké soubory uloženy.

## Posuny

Je možné, že systémový administrátor rozdělil fyzický disk na *oddíly*, což jsou různé části disku s vlastními názvy cest. Přestože je doporučeno k přidělení bloku na přímém diskovém zařízení použít celý diskový oddíl, můžete oddíly nebo předpřipravené soubory rozdělít na menší bloky pomocí *posunů*. Další informace naleznete v části “Pravidla pro rozvržení disků” na stránce 9-33.

**Rada:** Protože je velikost bloku omezena na 4 TB, můžete se vyhnout dělení disku na oddíly a přidělit celou diskovou jednotku jedinému bloku.

Posun umožňuje určit umístění daného bloku v diskovém oddílu, souboru nebo zařízení. Předpokládejme například, že vytváříte blok o velikosti 1000 kB, který chcete rozdělit do dvou bloků po 500 kB. Počátek prvního bloku můžete označit pomocí posunu s hodnotou 0 kB a počátek druhého bloku pomocí posunu s hodnotou 500 kB.

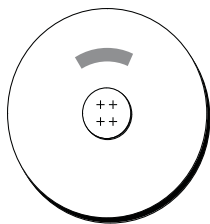
Posun můžete určit, kdykoli vytváříte, přidáváte nebo vypouštíte blok prostoru dbspace, blobspace nebo sbspace.

Je možné, že posun bude zapotřebí zadat také proto, aby databázový server nepřepsal informace o oddílech. “Přidělení diskového prostoru s přímým přístupem v systému UNIX” na stránce 10-5 popisuje, kdy a jak je třeba určit posun.

---

## Stránky

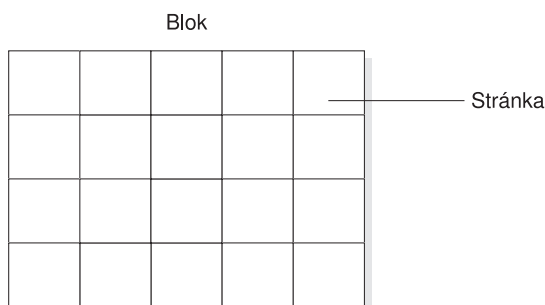
*Stránka* je fyzická jednotka diskové paměti, kterou databázový server používá ke čtení a zápisu do databázi Informix. Obrázek Obrázek 9-1 znázorňuje koncept stránky, zde reprezentované tmavým sektorem na disku.



Obrázek 9-1. Stránka na disku

Ve většině systémů UNIX je velikost stránky 2 kB. V systému Windows je velikost stránky 4 kB. Tuto hodnotu nelze změnit, protože je určena použitým hardwarem.

Blok obsahuje určitý počet stránek, jak znázorňuje Obrázek 9-2. Stránka je vždy celá obsažena v jediném bloku, nemůže tedy překračovat hranici dvou bloků.



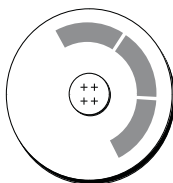
Obrázek 9-2. Blok logicky dělený do série stránek

Informace o způsobu, jakým databázový server strukturuje data v rámci stránky, naleznete v kapitole o diskových strukturách a paměti v příručce *IBM Informix Dynamic Server Administrator's Reference*

## Stránky blobpage

Stránka blobpage je jednotka přidělování diskového prostoru, který databázový server používá k ukládání jednoduchých velkých objektů (dat typu TEXT a BYTE) uvnitř prostorů blobspace. Popis prostorů blobspace naleznete v části "Prostory blobspace" na stránce 9-12.

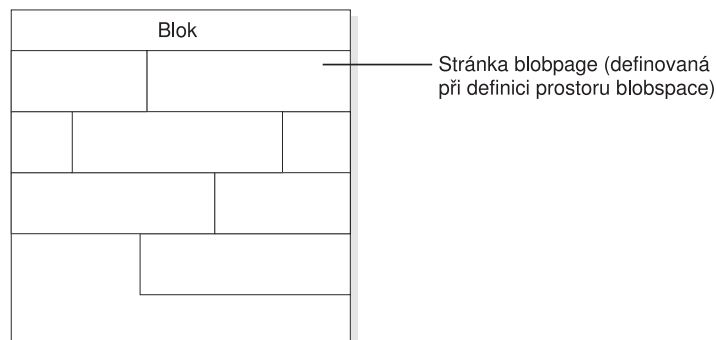
Velikost stránky blobpage se určuje jako násobek velikosti stránky databázového serveru. Protože databázový server přiděluje stránky blobpage jako souvislé prostory, jsou jednoduché velké objekty ukládány efektivněji, pokud velikost stránek blobpage co nejlépe odpovídá velikosti ukládaných dat. Obrázek Obrázek 9-3 znázorňuje koncept stránky blobpage reprezentované několika (třemi) datovými stránkami.



Obrázek 9-3. Stránka blobpage na disku

Informace o způsobu, jakým server Dynamic Server organizuje data ukládaná do stránek blobpage, naleznete v části věnované struktuře stránek blobpage v prostorech blobpage v kapitole o diskových strukturách a paměti v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Podobně jako stránky vytvářejí blok, i určitý počet stránek blobpage tvoří blok v prostoru blobpage, jak znázorňuje Obrázek 9-4. Stránka blobpage je vždy celá obsažena v jediném bloku a nemůže překračovat hranici dvou bloků.



Obrázek 9-4. Blok v prostoru blobpace logicky dělený do série stránek blobpage

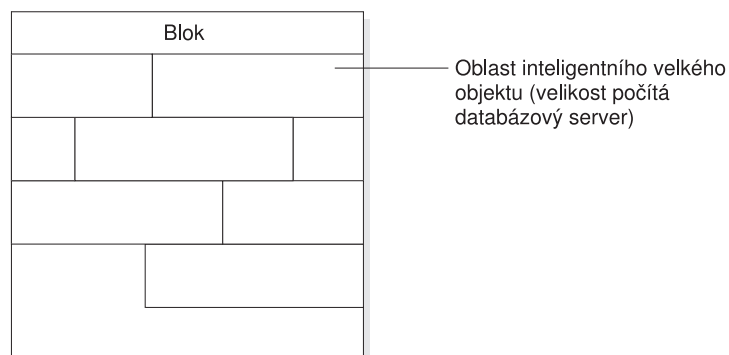
Data jednoduchých velkých objektů nemusíte ukládat v prostoru blobpace, můžete je ukládat i v prostoru dbspace. V případě jednoduchých velkých objektů větších než dvě stránky však bude výkon vyšší, pokud je budete ukládat v prostoru blobpace. Jednoduché velké objekty uložené v prostoru dbspace mohou sdílet jedinou stránku, jednoduché velké objekty uložené v prostoru blobpace stránky nesdílejí.

Informace o tom, jak určit velikost stránky blobpage, naleznete v části “Určení velikosti stránky blobpage” na stránce 10-23.

## Stránky sbpace

Stránka sbpace je typ stránky, do které databázový server ukládá v prostoru sbpace inteligentní velké objekty. Popis prostorů sbpace naleznete v části “Prostory sbpace” na stránce 9-12. Na rozdíl od stránek blobpage nelze stránky sbpace konfigurovat. Stránka sbpace má stejnou velikost jako stránka databázového serveru, což znamená obvykle 2 kB v systému UNIX a 4 kB v systému Windows.

Jednotkou přidělování v prostoru sbpace je oblast, zatímco jednotkou přidělování v prostoru blobpace je stránka blobpage. Podobně jako stránky vytvářejí blok, i určitý počet oblastí inteligentních velkých objektů tvoří blok prostoru sbpace, jak znázorňuje Obrázek 9-5. Oblast je vždy celá obsažena v jediném bloku a nemůže překračovat hranici dvou bloků.



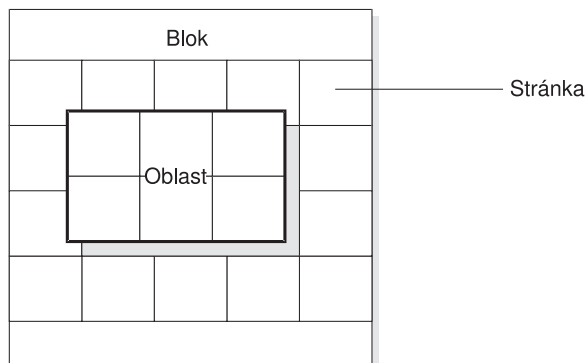
Obrázek 9-5. Blok v prostoru sbpace logicky dělený do série stránek oblastí

Inteligentní velké objekty nelze ukládat do prostorů dbspace a blobpace. Další informace naleznete v části “Prostory sbpace” na stránce 9-12 a v kapitole o diskových strukturách a paměti v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Databázový server vypočítává velikost oblasti inteligentních velkých objektů na základě sady heuristických pravidel, využívá například údaj o počtu bajtů přenášených při operaci zápisu. Další informace naleznete v části “Velikosti oblastí v prostorech sbspace” na stránce 9-15.

## Oblasti

Při vytváření tabulky databázový server přiděluje pevné množství diskového prostoru, do kterého bude ukládat data této tabulky. Pokud se tento prostor zaplní, musí databázový server přidělit další paměťový prostor. Fyzická jednotka paměti, kterou databázový server používá k přidělování počátečních paměťových prostorů i dalších paměťových prostorů, se nazývá *oblast*. Obrázek Obrázek 9-6 znázorňuje koncept oblasti.



Obrázek 9-6. Oblast, která se skládá ze šesti sousedících stránek na přímém diskovém zařízení

Oblast tvoří kolekce sousedících stránek, do kterých jsou ukládána data konkrétní tabulky. (Další informace naleznete v části “Tabulky” na stránce 9-22.) Ke každé trvalé databázové tabulce jsou přidruženy dvě velikosti oblastí. Velikost *počáteční oblasti* je počet kilobajtů přidělených tabulce při jejím počátečním vytvoření. Velikost *dalších oblastí* je počet kilobajtů přidělovaných tabulce tehdy, pokud dojde k zaplnění počáteční oblasti a všech dalších existujících oblastí. V případě trvalých tabulek a uživatelských dočasných tabulek se po přidání šestnácti oblastí k tabulce velikost další oblasti zdvojnásobuje. V případě systémových dočasných tabulek se velikost další oblasti zdvojnásobuje po přidání čtyř oblastí k tabulce.

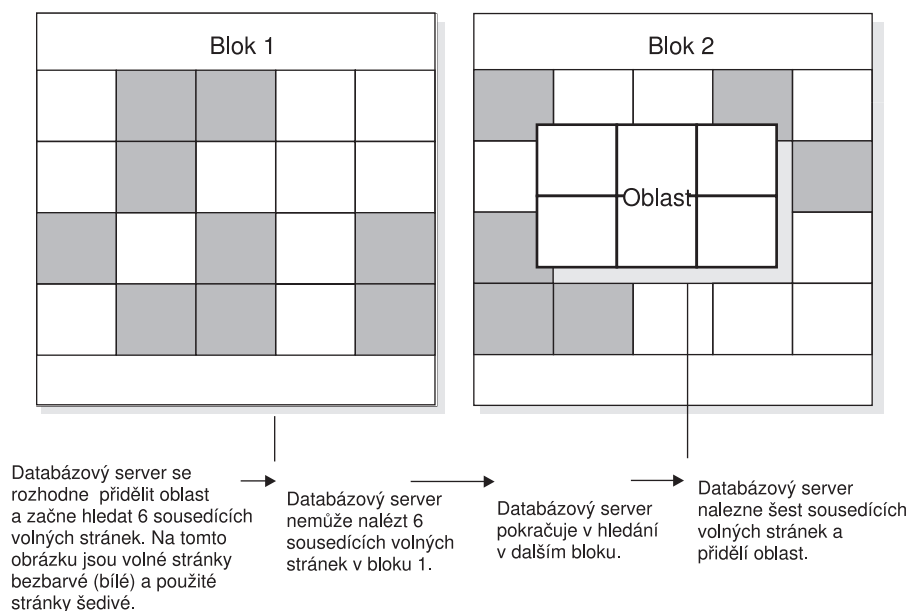
Velikost počáteční oblasti a dalších oblastí určete pomocí příkazů `CREATE TABLE` a `ALTER TABLE`. Další informace naleznete v příručce *IBM Informix Guide to SQL: Syntax* a v části o diskových strukturách v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Pokud vytváříte tabulku se sloupcem datového typu `CLOB` nebo `BLOB`, definujete také oblasti prostoru `sbspace`. Další informace naleznete v příručce “Paměťové charakteristiky prostorů `sbspace`” na stránce 9-15.

Obrázek Obrázek 9-7 znázorňuje, jakým způsobem databázový server přiděluje oblasti šest stránek:

- Oblast je vždy celá obsažena v jediném bloku a nemůže překračovat hranici dvou bloků.
- Pokud databázový server nemůže nalézt souvislý diskový prostor určený velikostí další oblasti, pokusí se nalézt potřebný souvislý diskový prostor v dalším prostoru `dbspace`.





Obrázek 9-7. Proces přidělování oblasti

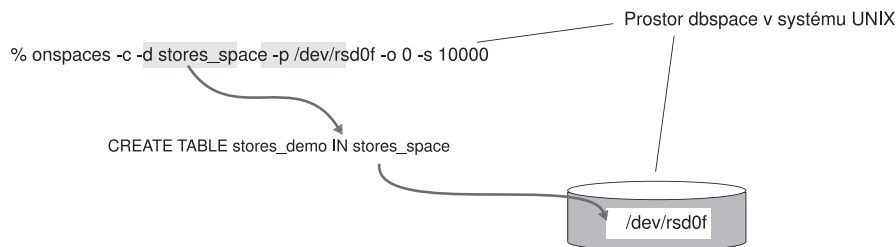
## Prostory dbspace

Prostor *dbspace* je logická jednotka, která může obsahovat 1 až 32767 bloků. Databáze, tabulky, logické protokoly a fyzické protokoly ukládáte do prostorů *dbspace*.

### Řízení umístění ukládaných dat

Mezi klíčové zodpovědnosti administrátora databázového serveru patří řízení umístění dat ukládaných databázovým serverem. Pokud umístíte často používané tabulky nebo *kritické prostory dbspace* (kořenový prostor *dbspace*, fyzický protokol a logický protokol) na nejrychlejší diskovou jednotku, můžete tím zvýšit výkon serveru. Pokud uložíte kritická data na samostatná fyzická zařízení, zajistíte tím, že v případě selhání disků, na kterých jsou uložena nekritická data, bude ovlivněna dostupnost pouze těchto dat.

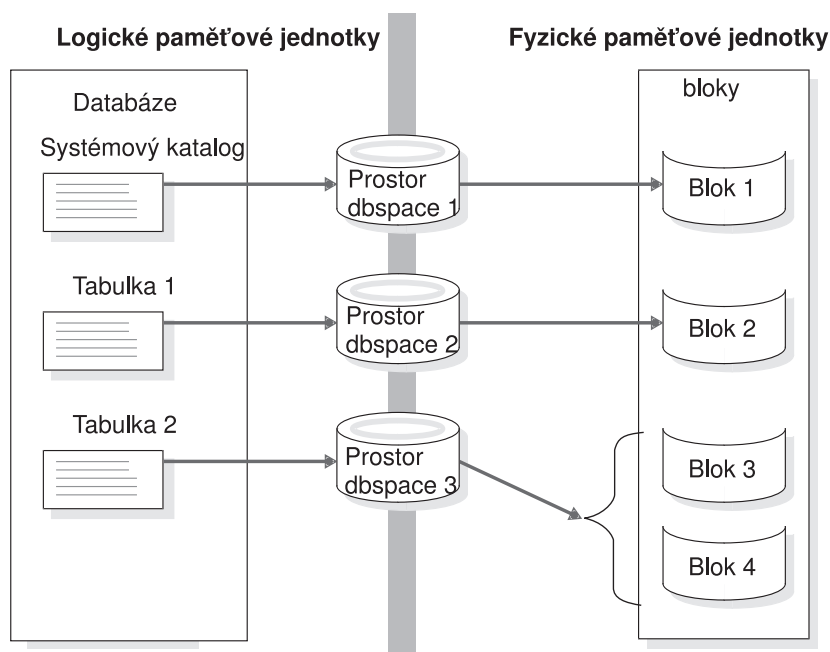
Jak znázorňuje Obrázek 9-8, můžete k řízení umístění databázi a tabulek používat volbu *IN prostor\_dbspace* příkazů `CREATE DATABASE` a `CREATE TABLE`. (Další informace naleznete v části “Tabulky” na stránce 9-22.)



Obrázek 9-8. Řízení umístění tabulek pomocí příkazu `CREATE TABLE... IN`

Než v prostoru *dbspace* vytvoříte databázi nebo tabulku, je zapotřebí prostor *dbspace* nejprve vytvořit. Informace o tom, jak vytvořit prostor *dbspace* naleznete v části “Vytvoření prostoru *dbspace*, který používá výchozí velikost stránky” na stránce 10-9.

Prostor dbspace obsahuje jeden nebo více bloků, jak znázorňuje Obrázek 9-9. Do prostoru dbspace můžete kdykoli přidávat další bloky. Velmi důležitým úkolem administrátora databázového serveru je monitorovat zaplnění bloků v prostorech dbspace a včas předvídat potřebu přidělení dalších bloků prostorům dbspace. (Viz “Monitorování využití disku” na stránce 10-34.) Pokud prostor dbspace obsahuje více než jeden blok, nelze určit blok, do kterého budou data uložena.



Obrázek 9-9. Prostory dbspace, které propojují logické a fyzické paměťové jednotky

Databázový server používá prostory dbspace k ukládání databází a tabulek. (Další informace naleznete v části “Tabulky” na stránce 9-22.)

Pokud vytváříte standardní nebo dočasný prostor dbspace, můžete určit velikost stránky tohoto prostoru. Velikost stránky nemůžete určit prostorům blobspace, prostorům sbpace a externím prostorům. Pokud neurčíte velikost stránky, bude jako výchozí použita velikost stránky kořenového prostoru dbspace. Další informace naleznete v části “Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky” na stránce 10-12.

Pokud vytváříte standardní prostor dbspace, můžete určit velikost první oblasti a velikost dalších oblastí prostoru tblspace **tblspace** tohoto prostoru dbspace. Tento krok proveďte, pokud chcete zmenšit počet oblastí prostoru tblspace **tblspace** a omezit frekvenci situací, ve kterých by bylo zapotřebí umístit další oblasti prostoru tblspace **tblspace** do bloku jiného než primárního. Další informace naleznete v části “Určení velikosti první oblasti a dalších oblastí prostoru tblspace **tblspace**” na stránce 10-10.

Každý blok můžete zrcadlit v zrcadleném prostoru dbspace. Jakmile databázový server přidělí zrcadlený blok, označí veškerý prostor v zrcadleném bloku jako zaplněný. Další informace naleznete v části “Monitorování využití disku” na stránce 10-34.

V části Kapitola 10, “Správa diskového prostoru”, na stránce 10-1 naleznete informace o použití programu ISA a obslužného programu **onspaces** k provádění následujících úloh.

- Vytvoření prostoru dbspace.
- Přidání bloku do prostoru dbspace.

- Přejmenování prostoru dbspace.
- Vypuštění bloku.
- Vypuštění prostoru dbspace, blobspace nebo sbpace.

## Kořenový prostor dbspace

Kořenový prostor dbspace je počáteční prostor dbspace vytvářený databázovým serverem. Kořenový prostor dbspace se od ostatních prostorů dbspace odlišuje tím, že obsahuje rezervované stránky a interní tabulky, které zaznamenávají všechny fyzické a logické paměťové jednotky. (Další informace o těchto tématech naleznete v části “Tabulky” na stránce 9-22 a v kapitole o diskových strukturách a diskové paměti v příručce *IBM Informix Dynamic Server Administrator's Reference*.) Počáteční blok kořenového prostoru dbspace a jeho zrcadlený blok jsou jediné bloky vytvářené při inicializaci diskového prostoru. Další bloky můžete do kořenového prostoru dbspace přidat po inicializaci diskového prostoru.

Následující konfigurační parametry disku v konfiguračním souboru ONCONFIG se vztahují k prvnímu (počátečnímu) bloku kořenového prostoru dbspace:

- ROOTPATH
- ROOTOFFSET
- ROOTNAME
- MIRRORPATH
- MIRROROFFSET
- TBLTBLFIRST
- TBLTBLNEXT

Kořenový prostor dbspace je také výchozím umístěním všech databází vytvářených pomocí příkazu CREATE DATABASE.

Kořenový prostor dbspace je výchozím umístěním všech dočasných tabulek vytvářených databázovým serverem za účelem provádění požadované správy dat.

“Velikost kořenového prostoru dbspace” na stránce 9-32 vysvětluje, kolik prostoru je třeba přidělit kořenovému prostoru dbspace. Po inicializaci diskového prostoru databázového serveru můžete ke kořenovému prostoru dbspace přidat další bloky.

## Dočasné prostory dbspace

Dočasný prostor dbspace je prostor dbspace určený výhradně k ukládání dočasných tabulek. Dočasné prostory dbspace nelze zrcadlit.

Databázový server nikdy nevypouští dočasné prostory dbspace, pokud neobdrží explicitní požadavek na jejich vypuštění. Dočasný prostor dbspace je dočasný pouze v tom smyslu, že databázový server nezachová jeho obsah, pokud databázový server nebude vypnut správným způsobem.

Při inicializaci databázového serveru budou všechny dočasné prostory dbspace opět inicializovány. Databázový server z prostorů odstraní všechny dočasné tabulky, které byly v prostorech uloženy před vypnutím databázového serveru.

Databázový server neprovádí fyzické ani logické protokolování dočasných prostorů dbspace. Protože dočasné prostory dbspace nejsou fyzicky protokolovány, snižuje se frekvence kontrolních bodů a operací vstupu - výstupu, a tím dochází ke zvýšení výkonu.

Databázový server v případě dočasných tabulek ve standardních prostorech dbspace protokoluje vytváření tabulek, přidělování oblastí a vypouštění tabulek. Neprotokoluje však

tabulky uložené v dočasných prostorech dbSPACE. Potlačení logického protokolu pro dočasné prostory dbSPACE se také omezuje počet záznamů protokolu, které je třeba přehrát v průběhu logické obnovy, což zvyšuje výkon serveru v kritickém okamžiku obnovy vypnutého serveru.

Použití dočasných prostorů dbSPACE k uložení dočasných tabulek také zmenšuje velikost záloh paměťových prostorů, protože databázový server dočasné prostory dbSPACE nezalohuje.

Pokud máte vytvořený více než jeden dočasný prostor dbSPACE a spustíte příkaz SELECT s výstupem do dočasné tabulky, budou výsledky dotazu ukládány s použitím cyklické obsluhy jednotlivými dočasnými prostory dbSPACE.

Podrobné informace o tom, jak vytvořit dočasný prostor dbSPACE naleznete v části “Vytvoření dočasného prostoru dbSPACE” na stránce 10-17.

**Důležité:** Pokud je databázový server spuštěn jako sekundární databázový server v páru replikace HDR, vyžaduje server dočasný prostor dbSPACE k ukládání všech interních tabulek generovaných dotazy v režimu pouze pro čtení.

---

## Prostory blobSPACE

Prostor blobSPACE je logická paměťová jednotka obsahující jeden nebo více bloků, do kterých jsou ukládána pouze data typu TEXT a BYTE. Prostor blobSPACE ukládá data typu TEXT a BYTE nejefektivnějším možným způsobem. Do jediného prostoru blobSPACE lze ukládat sloupce typu TEXT a BYTE, které patří do různých tabulek (další informace naleznete v části “Tabulky” na stránce 9-22).

Databázový server zapisuje data ukládaná do prostoru blobSPACE přímo na disk. Tato data neprocházejí rezidentní sdílenou pamětí. Pokud by procházela sdílenou pamětí, způsobil by veliký objem těchto dat při uložení do stránek sdílené paměti vytěsnění jiných dat a indexů ze společné oblasti vyrovnávacích pamětí. Z této důvodu nezapisuje databázový server objekty typu TEXT a BYTE přiřazené prostoru blobSPACE ani do logického a fyzického protokolu. Databázový server protokoluje objekty v prostorech blobSPACE při zálohování logických protokolů přímým zápisem z disku do zálohovacích pásek logického protokolu. Objekty ukládané do prostorů blobSPACE nikdy neprocházejí soubory logického protokolu.

Při vytváření prostoru blobSPACE přiřazujete tomuto prostoru jeden nebo více bloků. Do prostoru dbSPACE můžete kdykoli přidávat další bloky. Jedním z úkolů administrátora databázového serveru je monitorovat zaplnění bloků a včas předvídat potřebu přidělení dalších bloků prostorům blobSPACE. Informace o tom, jak monitorovat zaplnění bloků naleznete v části “Monitorování jednoduchých velkých objektů v prostoru blobSPACE” na stránce 10-38. Informace o tom, jak vytvořit prostor blobSPACE, jak přidat do prostoru blobSPACE další bloky a jak vypustit blok z prostoru blobSPACE naleznete v části Kapitola 10, “Správa diskového prostoru”, na stránce 10-1.

Informace o struktuře prostoru blobSPACE naleznete v kapitole o diskových strukturách a paměti v příručce *IBM Informix Dynamic Server Administrator's Reference*.

---

## Prostory sbSPACE

Prostor sbSPACE je logická paměťová jednotka složená z jednoho nebo více bloků, do kterých jsou ukládány *inteligentní velké objekty*. Mezi inteligentní velké objekty patří objekty CLOB (znakové velké objekty) a objekty BLOB (binární velké objekty). Prostory sbSPACE mohou být používány také k ukládání uživatelských datových typů. Další informace o datových typech naleznete v příručce *IBM Informix Guide to SQL: Reference*.

## Výhody používání prostorů sbpace

Prostory sbpace mají v porovnání s prostory blobspace následující výhody:

- Z pohledu operací čtení, zápisu a vyhledávání mají podobné vlastnosti jako standardní soubory systému UNIX.

K vyhledávání, čtení a zapisování inteligentních velkých objektů mohou programátoři použít funkce podobné funkcím systémů UNIX a Windows. Dynamic Server poskytuje toto rozhraní inteligentních velkých objektů v rozhraní API modulů DataBlade a v ESQL/C programovacím rozhraní.

- Tyto objekty lze obnovit.

Všechny operace zápisu dat do prostorů sbpace můžete protokolovat. Pokud dojde k selhání v průběhu transakce, můžete provedené změny potvrdit nebo odvolat.

- Respektují úroveň izolace transakcí.

Inteligentní velké objekty můžete s různou granularitou zamykat. Doba trvání zámků respektuje pravidla pro úroveň izolace transakcí. Další informace o zamykání a souběžném zpracování naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

- inteligentní velké objekty v řádcích tabulek není zapotřebí získávat celé v rámci jediného příkazu.

Aplikace může inteligentní velké objekty ukládat nebo získávat po částech pomocí programovacích rozhraní DataBlade API nebo ESQL/C. Další informace o funkcích rozhraní DataBlade API naleznete v příručce *IBM Informix DataBlade API Function Reference*. Další informace o funkcích jazyka ESQL/C naleznete v příručce *IBM Informix ESQL/C Programmer's Manual*.

## Prostory sbpace a replikace Enterprise Replication

Před definováním replikačního serveru replikace Enterprise Replication je zapotřebí vytvořit prostor sbpace. Replikace Enterprise Replication zařazuje replikovaná data do front v inteligentních velkých objektech. Název prostoru sbpace určete pomocí konfiguračního parametru CDR\_QDATA\_SBSPACE. Replikace Enterprise Replication používá k zařazování řádkových dat do fronty výchozí režim protokolování, se kterým byl prostor sbpace vytvořen. Konfigurační parametr CDR\_QDATA\_SBSPACE přijímá i více prostorů sbpace. Lze zadat maximálně 32 prostorů. Replikace Enterprise Replication podporuje zařazování řádkových dat do fronty i v případě kombinace protokolujících a neprotokolujících souborů sbpace. Další informace naleznete v příručce *IBM Informix Dynamic Server Enterprise Replication Guide*.

## Metadata, uživatelská data a rezervovaná oblast

Podobně jako při vytváření prostorů blobspace a dbspace je i při vytváření prostoru sbpace přiřazen tomuto prostoru jeden nebo více bloků. První blok prostoru sbpace se však vždy skládá ze tří částí:

- *Oblast metadat.*

Metadata určují klíčové aspekty prostoru sbpace a všech inteligentních velkých objektů uložených v prostoru sbpace a umožňují databázovému serveru manipulovat s inteligentními velkými objekty uloženými v tomto prostoru a provádět jejich obnovu.

- *Oblast uživatelských dat.*

Uživatelská data jsou data inteligentních velkých objektů ukládaná do prostoru sbpace uživatelskými aplikacemi. Blok může obsahovat až dvě oblasti uživatelských dat.

- *Rezervovaná oblast*

Databázový server přiděluje oblasti metadat nebo oblasti uživatelských dat v případě potřeby prostor z rezervované oblasti. Blok může obsahovat až dvě rezervované oblasti.

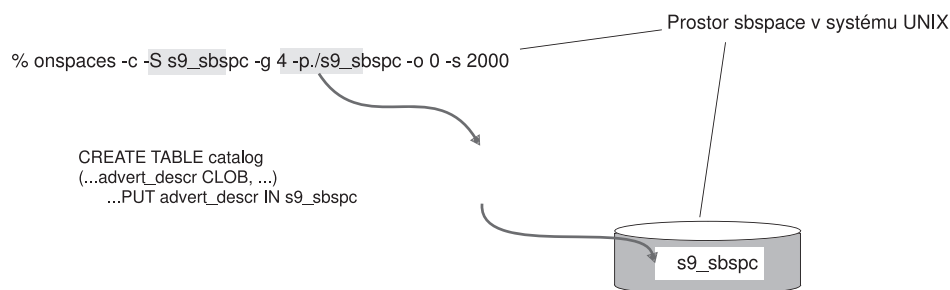
Informace o tom, jak správně přidělit oblasti metadat a uživatelských dat v prostorech sbspace naleznete v části “Určení velikosti metadat prostoru sbspace” na stránce 10-25 a v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Pokud přidáváte blok k prostoru sbspace, můžete určit, zda má obsahovat oblast metadat a uživatelských dat nebo zda má být vyhrazen pouze pro uživatelská data. Do prostoru sbspace můžete kdykoli přidávat další bloky. Pokud aktualizujete inteligentní velké objekty, jsou operace vstupu - výstupu pracující s uživatelskými daty mnohem rychlejší při použití přímých disků než při použití předpřipravených souborů bloků. Informace o tom, jak vytvořit prostor sbspace, jak přidat do prostoru sbspace další bloky a jak vypustit blok z prostoru sbspace naleznete v části Kapitola 10, “Správa diskového prostoru”, na stránce 10-1.

**Důležité:** Metadata prostoru sbspace jsou vždy protokolována, nezávisle na tom, jak je nastaveno protokolování databáze.

## Řízení umístění ukládaných dat

Datový typ sloupce je určen při vytváření tabulky. Použití inteligentních velkých objektů určují datové typy CLOB a BLOB a uživatelské datové typy. Jak znázorňuje obrázek Obrázek 9-10, můžete umístění inteligentních velkých objektů řídit pomocí volby `IN prostor_sbspace` v klauzuli `PUT` příkazu `CREATE TABLE`.



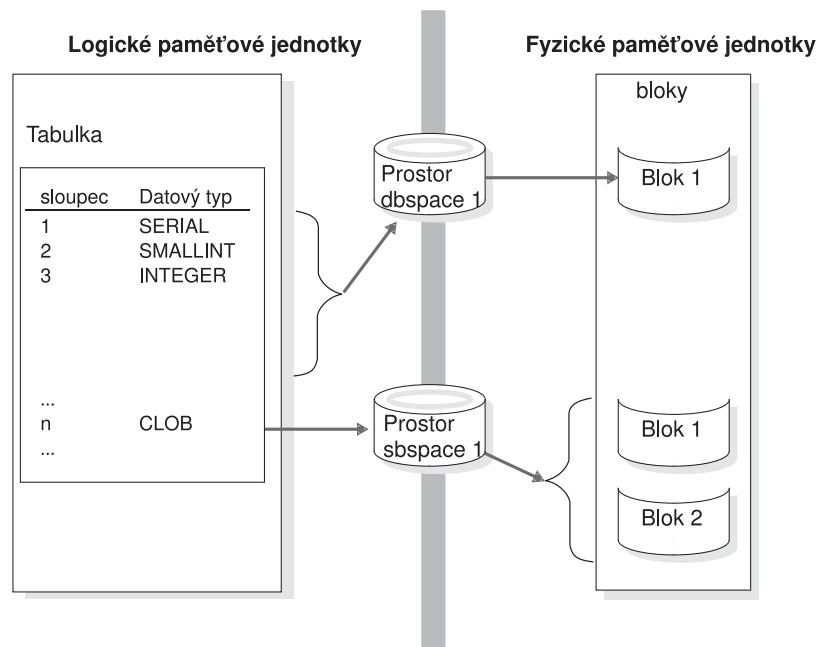
Obrázek 9-10. Řízení umístění inteligentních velkých objektů

Než určíte prostor sbspace v klauzuli `PUT`, je nejprve nutné prostor sbspace vytvořit. Informace o tom, jak vytvořit prostor sbspace pomocí příkazu **onspaces -c -S** naleznete v části “Přidání bloku do prostoru sbspace nebo blobspace” na stránce 10-18. Další informace o tom, jak určit charakteristiky inteligentních velkých objektů v klauzuli `PUT` naleznete v části věnované příkazu `CREATE TABLE` v příručce *IBM Informix Guide to SQL: Syntax*.

Pokud neurčíte klauzuli `PUT`, bude databázový server ukládat inteligentní velké objekty do výchozího prostoru sbspace určeného konfiguračním parametrem `SBSPACENAME`. Další informace o parametru `SBSPACENAME` naleznete v části věnované konfiguračním parametrům v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Prostor sbspace obsahuje jeden nebo více bloků, jak znázorňuje obrázek Obrázek 9-11. Pokud prostor sbspace obsahuje více než jeden blok, nelze určit blok, do kterého budou data uložena.

Do prostoru sbspace můžete kdykoli přidávat další bloky. Velmi důležitým úkolem administrátora databázového serveru je monitorovat zaplnění bloků v prostorech sbspace a včas předvídat potřebu přidělení dalších bloků prostorům sbspace. Další informace o monitorování prostorů sbspace naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.



Obrázek 9-11. Prostory sbspace, které propojují logické a fyzické paměťové jednotky

Databázový server prostory sbspace používá k ukládání sloupců tabulek, které obsahují inteligentní velké objekty. K ukládání ostatních sloupců tabulek používá databázový server prostory dbspace.

Zrcadlením prostoru sbspace můžete urychlit obnovu v případě selhání média. Další informace naleznete v části “Zrcadlení” na stránce 17-1.

Informace o použití příkazu **onspaces** k provádění následujících úloh uvádí Kapitola 10, “Správa diskového prostoru”, na stránce 10-1.

- Vytvoření prostoru sbspace.
- Přidání bloku k prostoru sbspace.
- Změna paměťových charakteristik inteligentních velkých objektů.
- Vytvoření dočasného prostoru sbspace.
- Vypuštění prostoru sbspace.

## Paměťové charakteristiky prostorů sbspace

Jako administrátor databázového serveru můžete jako tyto paměťové charakteristiky použít systémové výchozí hodnoty nebo můžete paměťové charakteristiky určit pomocí voleb **-Df** při vytváření prostoru sbspace pomocí příkazu **onspaces -c**. Později můžete tyto charakteristiky prostorů sbspace změnit pomocí příkazu **onspaces -ch**. Administrátor nebo programátor může tyto výchozí hodnoty paměťových charakteristik a atributů potlačit pro jednotlivé tabulky.

### Velikosti oblastí v prostorech sbspace

Oblast v prostoru sbspace je podobně jako oblasti tabulky tvořena kolekcí sousedících stránek, do kterých jsou ukládána data inteligentních velkých objektů.

Jednotkou přidělování v prostoru sbspace je oblast. Databázový server vypočítává velikost oblasti inteligentních velkých objektů na základě sady heuristických pravidel, využívá například údaj o počtu bajtů přenášených při operaci zápisu. Pokud například operace požaduje zápis 30 kB dat, pokusí se databázový server přidělit oblast o velikosti 30 kB.

**Důležité:** Pro většinu aplikací je doporučeno používat velikost oblasti vypočtenou databázovým serverem.

Pokud znáte velikost inteligentního velkého objektu, můžete k nastavení velikosti oblasti použít jednu z následujících funkcí. Databázový server přiděluje celému inteligentnímu velkému objektu jedinou oblast (pokud je v daném bloku dostupná oblast potřebné velikosti):

- Funkce **mi\_lo\_specset\_estbytes()** rozhraní DataBlade API.  
Další informace o funkcích rozhraní DataBlade API pro práci s inteligentními velkými objekty naleznete v příručce *IBM Informix DataBlade API Function Reference*.
- Funkce **ifx\_lo\_specset\_estbytes** jazyka ESQL/C.  
Další informace o funkcích rozhraní ESQL/C pro práci s inteligentními velkými objekty naleznete v příručce *IBM Informix ESQL/C Programmer's Manual*.

Informace o ladění velikosti oblastí naleznete v části o inteligentních velkých objektech v kapitole o vlivu konfigurace na využití vstupu - výstupu v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

## Průměrná velikost inteligentních velkých objektů

Inteligentní velké objekty mají obvykle různou velikost. K výpočtu velikosti prostoru sbspace můžete poskytnout databázovému serveru průměrnou velikost inteligentních velkých objektů, které budete chtít ukládat. Tuto průměrnou velikost můžete určit pomocí parametru AVG\_LO\_SIZE volby příkazu **onspaces -c -Df**.

Pokud chcete určit velikost a umístění oblasti metadat, určete v příkazu **onspaces** příznaky **-Ms** a **-Mo**. Pokud nepoužijete příznak **-Ms**, odhadne databázový server velikost oblasti metadat, kterou je třeba přidělit, pomocí hodnoty parametru AVG\_LO\_SIZE. Další informace naleznete v části “Určení velikosti metadat prostoru sbspace” na stránce 10-25.

## Režim použití vyrovnávací paměti

Pokud vytvoříte prostor sbspace, bude ve výchozím nastavení režim ukládání do vyrovnávací paměti zapnutý, což znamená, že databázový server bude používat společnou oblast vyrovnávacích pamětí v rezidentní části sdílené paměti.

Jako administrátor databáze můžete určit režim ukládání do vyrovnávací paměti pomocí parametru BUFFERING volby příkazu **onspaces -c -Df**. Výchozí nastavení “buffering=ON” znamená, že server bude používat společnou oblast vyrovnávacích pamětí. Pokud použití vyrovnávací paměti vypnete, použije databázový server soukromé vyrovnávací paměti ve virtuální části sdílené paměti.

**Důležité:** Platí obecné pravidlo, podle kterého při vytváření prostoru sbspace neurčíte režim použití vyrovnávací paměti, pokud operace čtení a zápisu inteligentních velkých objektů nepracují s daty většími než 8 kB. Pokud čtete nebo zapisujete krátké bloky dat, například o velikosti 2 kB nebo 4 kB, ponechte výchozí nastavení “buffering=ON”, kterým dosáhnete zvýšení výkonu.

Informace o tom, kdy je vhodné používat soukromé vyrovnávací paměti, naleznete v části týkající se lehkých operací vstupu - výstupu v kapitole o vlivu konfigurace na využití vstupu - výstupu v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

## Čas posledního přístupu

Při vytváření prostoru sbspace můžete pomocí parametru ACESSTIME volby příkazu **onspaces -c -Df** určit, zda databázový server má či nemá ukládat čas posledního čtení nebo aktualizace inteligentního velkého objektu. Výchozí nastavení parametru je “ACESSTIME=OFF”. Databázový server ukládá čas posledního přístupu do oblasti metadat.



Informace o tom, jak programátoři používají údaj o čase posledního přístupu naleznete v příručce *IBM Informix DataBlade API Programmer's Guide* a v příručce *IBM Informix ESQL/C Programmer's Manual*.

## Režim uzamčení

Při vytváření prostoru sbspace můžete pomocí parametru `LOCK_MODE` volby příkazu **onspaces -c -Df** určit, zda má databázový server zamykat celé inteligentní velké objekty nebo pouze rozsahy bajtů uvnitř inteligentních velkých objektů. Výchozí nastavení "`LOCK_MODE=BLOB`" znamená, že databázový server bude zamykat celé inteligentní velké objekty. Další informace naleznete v kapitole o zámcích v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

## Protokolování

Při vytváření prostoru sbspace můžete určit, zda má být protokolování inteligentních velkých objektů zapnuto nebo vypnuto. Výchozí režim je režim bez protokolování. Další informace naleznete v části "Protokolování prostorů sbspace a inteligentních velkých objektů" na stránce 13-7.

**Důležité:** Pokud používáte protokolující databáze, protokolování prostorů sbspace zapněte. Pokud dojde k selhání vyžadující obnovu protokolu, budou inteligentní velké objekty po obnově konzistentní se zbytkem databáze.

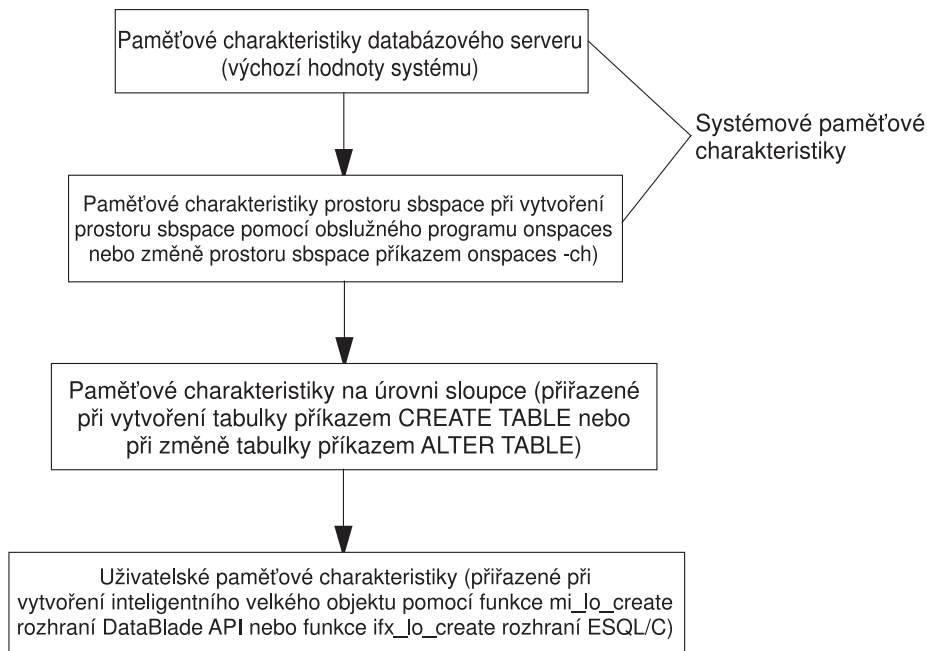
Stav protokolování určete pomocí parametru `LOGGING` volby příkazu **onspaces -c -Df**. Výchozí nastavení je "`LOGGING=off`". Stav protokolování můžete změnit pomocí volby **onspaces -ch -Df**. Stav protokolování můžete také potlačit pomocí klauzule `PUT` příkazů `CREATE TABLE` a `ALTER TABLE` jazyka SQL. Další informace o těchto příkazech jazyka SQL naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

Programátor může potlačit tento režim protokolování pomocí funkcí poskytovaných rozhraními DataBlade API a ESQL/C. Další informace o funkcích rozhraní DataBlade API pro práci s inteligentními velkými objekty naleznete v příručce *IBM Informix DataBlade API Function Reference*. Další informace o funkcích rozhraní ESQL/C pro práci s inteligentními velkými objekty naleznete v příručce *IBM Informix ESQL/C Programmer's Manual*.

Pokud zapnete protokolování prostorů sbspace, budou inteligentní velké objekty procházet rezidentní částí sdílené paměti. Přestože aplikace mohou inteligentní velké objekty získávat po jednotlivých částech, zvažte možnost, že společnou oblastí vyrovnávací paměti a vyrovnávacími paměťmi logického protokolu budou procházet větší objemy dat. Další informace naleznete v části "Přístup k inteligentním velkým objektům" na stránce 7-32.

## Úrovně dědičnosti charakteristik prostorů sbspace

Existují čtyři úrovně dědičnosti charakteristik prostoru sbspace: Úroveň systému, úroveň prostoru sbspace, úroveň sloupce a úroveň inteligentního velkého objektu. Pro atributy prostorů sbspace můžete používat systémové výchozí hodnoty nebo je můžete potlačit pro konkrétní prostory sbspace, konkrétní sloupce tabulky nebo pro konkrétní inteligentní velké objekty. Obrázek Obrázek 9-12 znázorňuje hierarchii charakteristik inteligentního velkého objektu.



Obrázek 9-12. Hierarchie paměťových charakteristik

Obrázek 9-12 znázorňuje, že výchozí systémová nastavení můžete potlačit následujícími způsoby:

- Pomocí parametrů volby **-Df** příkazu **onspaces -c -S** můžete potlačit výchozí nastavení systému pro konkrétní prostor sbspace.  
Později můžete tyto atributy prostoru sbspace změnit pomocí volby příkazu **onspaces -ch**. Další informace o platných rozsazích parametrů volby **-Df** naleznete v části věnované příkazu **onspaces** v příručce *IBM Informix Dynamic Server Administrator's Reference*.
- Výchozí systémová nastavení můžete potlačit pro konkrétní sloupec určením těchto atributů v klauzuli PUT příkazů CREATE TABLE a ALTER TABLE.  
Další informace o těchto příkazech jazyka SQL naleznete v příručce *IBM Informix Guide to SQL: Syntax*.
- Programátor může potlačit výchozí hodnoty atributů prostoru sbspace pro konkrétní inteligentní velké objekty pomocí funkcí poskytovaných rozhraními DataBlade API a ESQL/C.

## Další informace o prostorech sbspace

Tabulka 9-1 uvádí zdroje informací o různých úlohách, které se týkají používání a správy prostorů sbspace.

Tabulka 9-1. Nalezení informací o úlohách týkajících se prostorů sbspace

| Úkol                                                              | Odkaz                                                                                             |
|-------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Nastavení konfiguračních parametrů inteligentních velkých objektů | Kapitola 8, “Správa sdílené paměti”, na stránce 8-1                                               |
| Porozumění stránkám sbpage                                        | “Stránky sbpage” na stránce 9-7                                                                   |
| Určení charakteristik vstupu - výstupu prostoru sbspace           | Volba příkazu <b>onspaces</b> v části “Paměťové charakteristiky prostorů sbspace” na stránce 9-15 |
| Přidělení prostoru prostorům sbspace.                             | “Vytvoření prostoru sbspace” na stránce 10-24                                                     |
| Přidání bloku k prostoru sbspace.                                 | “Přidání bloku do prostoru sbspace” na stránce 10-25                                              |

Tabulka 9-1. Nalezení informací o úlohách týkajících se prostorů sbspace (pokračování)

| Úkol                                                                                                       | Odkaz                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Definování nebo změna paměťových charakteristik inteligentních velkých objektů.                            | “Změna paměťových charakteristik inteligentních velkých objektů” na stránce 10-26<br><br>Klauzule PUT příkazů CREATE TABLE a ALTER TABLE v příručce <i>IBM Informix Guide to SQL: Syntax</i> |
| Monitorování prostorů sbspace.                                                                             | “Monitorování prostorů sbspace” na stránce 10-41<br>Kapitola o posouzení výkonu tabulek v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i>                            |
| Nastavení protokolování prostoru sbspace.                                                                  | “Protokolování prostorů sbspace a inteligentních velkých objektů” na stránce 13-7                                                                                                            |
| Zálohování prostoru sbspace.                                                                               | “Zálohování prostorů sbspace” na stránce 14-5                                                                                                                                                |
| Kontrola konzistence prostoru sbspace.                                                                     | “Ověření metadat” na stránce 23-3                                                                                                                                                            |
| Porozumění struktuře prostoru sbspace.                                                                     | Kapitola o diskových strukturách v příručce <i>IBM Informix Administrator's Reference</i>                                                                                                    |
| Použití <b>onspaces</b> k operacím s prostory.                                                             | Kapitola o obslužných programech v příručce <i>IBM Informix Administrator's Reference</i>                                                                                                    |
| Vytvoření tabulky s datovými typy CLOB a BLOB.                                                             | <i>IBM Informix Guide to SQL: Syntax</i>                                                                                                                                                     |
| Přístup k inteligentním velkým objektům z aplikace                                                         | <i>IBM Informix DataBlade API Programmer's Guide</i><br><i>IBM Informix ESQ/C Programmer's Manual</i>                                                                                        |
| Výpočet velikosti oblasti metadat.<br>zlepšení vstupu - výstupu metadat<br>Změna paměťových charakteristik | Kapitola o výkonu tabulek v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i>                                                                                          |
| Porozumění zámkům inteligentních velkých objektů                                                           | Kapitola o zamykání v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i>                                                                                                |
| Konfigurace prostorů sbspace pro dočasné velké objekty                                                     | Kapitola o vlivu konfigurace na aktivitu vstupu - výstupu v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i> .                                                        |

## Dočasné prostory sbspace

*Dočasné prostory sbspace* používejte k ukládání inteligentních velkých objektů bez protokolování metadat a uživatelských dat. Pokud ukládáte dočasné inteligentní velké objekty do standardního prostoru sbspace, dochází k protokolování metadat. Dočasné prostory sbspace se podobají dočasným prostorům dbspace. K vytvoření dočasného prostoru sbspace použijte příkaz **onspaces -c -S** s volbou **-t**. Další informace naleznete v části “Vytvoření dočasného prostoru dbspace” na stránce 10-26.

Dočasné velké objekty můžete ukládat do standardního prostoru sbspace nebo do dočasného prostoru sbspace.

- Pokud určíte dočasný prostor sbspace v konfiguračním parametru SBSPACETEMP, můžete do něj ukládat dočasné inteligentní velké objekty.
- Pokud určíte standardní prostor sbspace v parametru SBSPACENAME, můžete do něj ukládat trvalé a dočasné inteligentní velké objekty.
- Pokud určíte název dočasného prostoru sbspace v příkazu CREATE TEMP TABLE, můžete do něj ukládat dočasné inteligentní velké objekty.
- Pokud určíte název trvalého prostoru sbspace v příkazu CREATE TABLE, můžete do něj ukládat dočasné inteligentní velké objekty.

- Pokud vynecháte parametry SBSPACETEMP a SBSPACENAME a vytvoříte inteligentní velký objekt, může se zobrazit chybová zpráva -12053.
- Pokud určíte v konfiguračním parametru SBSPACENAME dočasný prostor sbspace, nebudete do tohoto prostoru sbspace moci ukládat *trvalé* inteligentní velké objekty. Bude však možné ukládat do tohoto prostoru dočasné inteligentní velké objekty.

## Porovnání dočasných a standardních prostorů sbspace

Tabulka 9-2 porovnává standardní a dočasné prostory sbspace.

Tabulka 9-2. Dočasné a standardní prostory sbspace

| Charakteristiky                            | Standardní prostor sbspace                                                                                                                                                                                 | Dočasný prostor sbspace                                                                                                                                                                                                                                                                     |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ukládá inteligentní velké objekty?         | Ano                                                                                                                                                                                                        | Ne                                                                                                                                                                                                                                                                                          |
| Ukládá dočasné inteligentní velké objekty? | Ano                                                                                                                                                                                                        | Ano                                                                                                                                                                                                                                                                                         |
| Protokoluje metadata?                      | Metadata jsou vždy protokolována.                                                                                                                                                                          | Metadata nejsou protokolována.                                                                                                                                                                                                                                                              |
| Protokoluje uživatelská data?              | Uživatelská data v případě dočasných inteligentních velkých objektů nejsou protokolována, jsou však protokolována v případě trvalých inteligentních velkých objektů, pokud je nastaven parametr LOGGING=ON | Uživatelská data nejsou protokolována.<br>Vytváření a odstraňování prostoru a přidávání bloků je protokolováno.                                                                                                                                                                             |
| rychlá obnova                              | Ano                                                                                                                                                                                                        | Ne (prostor sbspace je vyprázdněn při restartování databázového serveru) Chcete-li vytvořit sdílenou paměť, aniž byste museli odstraňovat dočasné inteligentní velké objekty, zadejte příkaz <b>oninit -p</b> . Pokud dočasné velké objekty zachováte, budou se nacházet v neurčitěm stavu. |
| Podporuje zálohování a obnovení?           | Ano                                                                                                                                                                                                        | Ne                                                                                                                                                                                                                                                                                          |
| Lze přidávat a vypouštět bloky?            | Ano                                                                                                                                                                                                        | Ano                                                                                                                                                                                                                                                                                         |
| Konfigurační parametr                      | SBSPACENAME                                                                                                                                                                                                | SBSPACETEMP                                                                                                                                                                                                                                                                                 |

## Dočasné inteligentní velké objekty

*Dočasné inteligentní velké objekty* používejte k ukládání obrazových a textových dat (typu CLOB nebo BLOB), která nevyžadují obnovení za zálohy nebo obnovení přehráním protokolu v průběhu rychlé obnovy. Platnost dočasných inteligentních velkých objektů je omezena dobou trvání uživatelské relace a lze je aktualizovat mnohem rychleji než trvalé inteligentní velké objekty.

Dočasné inteligentní velké objekty můžete vytvářet stejným způsobem, jako trvalé inteligentní velké objekty, pouze je zapotřebí nastavit příznak `LO_CREATE_TEMP` funkce **ifx\_lo\_specset\_flags** nebo **mi\_lo\_specset\_flags**. K vytvoření trvalého inteligentního velkého objektu z dočasného inteligentního velkého objektu použijte funkci **mi\_lo\_copy** nebo **ifx\_lo\_copy**. Další informace o vytváření dočasných inteligentních velkých objektů naleznete v příručce *IBM Informix DataBlade API Programmer's Guide*.

**Důležité:** Ukazatele na dočasné velké objekty ukládejte pouze do dočasných tabulek. Pokud je uložíte do standardních tabulek a pak restartujete databázový server, může server zobrazit chybovou zprávu, že požadovaný velký objekt neexistuje.

Tabulka 9-3 porovnává standardní a dočasné inteligentní velké objekty.

Tabulka 9-3. Dočasné a standardní inteligentní velké objekty

| Charakteristiky                             | Inteligentní velký objekt                        | Dočasný inteligentní velký objekt                                   |
|---------------------------------------------|--------------------------------------------------|---------------------------------------------------------------------|
| Příznaky při vytváření                      | LO_CREATE_LOG nebo LO_CREATE_NOLOG               | příznak LO_CREATE_TEMP                                              |
| Odvolání transakce                          | Ano                                              | Ne                                                                  |
| Protokolování                               | Ano, pokud je zapnuto                            | Ne                                                                  |
| Doba trvání                                 | Objekt je trvalý (dokud ho uživatel neodstraní). | Objekt je odstraněn při ukončení uživatelské relace nebo transakce. |
| Typy tabulek, do kterých lze objekt ukládat | Trvalé a dočasné tabulky                         | Dočasné tabulky                                                     |

## Prostory extspace

Prostor extspace je logický název přidružený ke konkrétnímu řetězci představujícímu umístění externích dat. Zdroj odkazovaný prostorem extspace je závislý na uživatelské přístupové metodě, která je potřebná k přístupu k jeho obsahu.

Uživatel databáze může například požadovat přístup k binárním souborům kódovaným v proprietárním formátu. Vývojář nejprve vytvoří *přístupovou metodu*, což je sada funkcí pro přístup k datům. Tyto rutiny jsou zodpovědné za veškerou interakci databázového serveru a externího souboru. Administrátor DBA pak přidá k databázi prostor extspace, ke kterému je tento soubor přiřazen jako cíl. Jakmile administrátor DBA vytvoří tabulku v prostoru extspace, budou uživatelé moci k datům v proprietárních souborech přistupovat pomocí příkazu jazyka SQL. K určení umístění těchto souborů použijte informace o prostoru extspace.

Prostor extspace nemusí představovat název souboru. Prostor extspace může být například síťové umístění. Rutiny přistupující k datům mohou interpretovat informace nalezené v řetězci přidruženému v souboru extspace jakýmkoli způsobem.

Další informace o uživatelských přístupových metodách naleznete v příručce *IBM Informix Virtual-Table Interface Programmer's Guide*. Další informace o vytváření funkcí a primárních přístupových metod naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

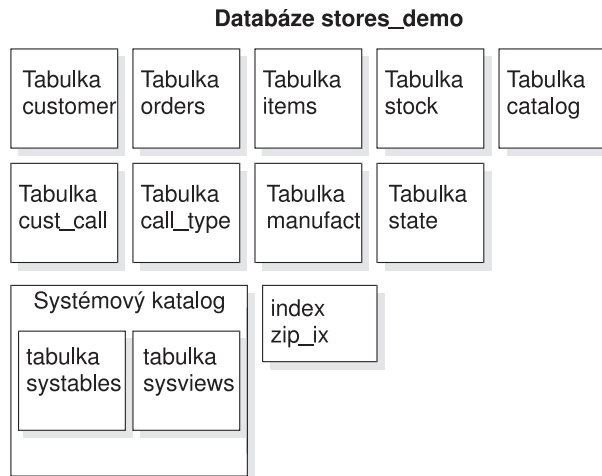
## Databáze

Databáze je logická paměťová jednotka, která obsahuje tabulky a indexy. (Další informace naleznete v části "Tabulky" na stránce 9-22.) Každá databáze obsahuje také systémový katalog, který zaznamenává informace o mnoha prvcích databáze, například o indexech, tabulkách, rutinách SPL a integritních omezeních.

Databáze je uložena v prostoru dbspace určeném v příkazu CREATE DATABASE. Pokud nezadáte název prostoru dbspace explicitně v příkazu CREATE DATABASE, bude databáze uložena do kořenového prostoru dbspace. Pokud v příkazu CREATE DATABASE *určíte* prostor dbspace, budou do tohoto prostoru dbspace umístěny následující tabulky:

- Tabulky systémového katalogu databáze.
- Všechny tabulky patřící do této databáze.

Obrázek Obrázek 9-13 znázorňuje tabulky, které jsou součástí databáze stores\_demo.



Obrázek 9-13. Databáze stores\_demo

Omezení velikosti, která platí pro databáze, souvisejí s jejich umístěním v prostorech dbspace. Pokud chcete zaručit, že všechny tabulky konkrétní databáze budou vytvořeny na konkrétním fyzickém zařízení, přiřadíte tomuto zařízení jediný blok a vytvoříte prostor dbspace, který bude obsahovat pouze tento blok. Databázi umístíte do tohoto prostoru dbspace. Pokud umístíte databázi do bloku přiřazeného konkrétnímu fyzickému zařízení, bude velikost databáze omezena na velikost tohoto bloku.

Pokyny, jak zobrazit vytvořené databáze naleznete v části “Zobrazení databází” na stránce 10-34.

## Tabulky

V relačních databázových systémech se tabulka skládá z řádku záhlaví sloupců a z libovolného počtu řádků s datovými hodnotami. Řádek záhlaví sloupců označuje jeden nebo více sloupců a datové typy všech sloupců.

Databázový server při vytváření tabulky přiděluje tabulce diskový prostor v podobě bloku stránek nazývaného oblast. (Další informace naleznete v části “Oblasti” na stránce 9-8.) Můžete určit velikost první přidělené oblasti i všech následujících oblastí.

Tabulku můžete do určitého prostoru dbspace umístit určením názvu prostoru při vytváření tabulky (obvykle pomocí volby *IN prostor\_dbspace* příkazu CREATE TABLE). Pokud neurčíte prostor dbspace, bude tabulka uložena do prostoru dbspace, ve kterém je uložena databáze.

Dále můžete:

- Rozdělit tabulku do fragmentů ve více prostorech dbspace. Nemůžete však fragmenty umístit do prostorů dbspace s rozdílnými velikostmi stránek. Všechny fragmenty musejí mít shodnou velikost stránky.

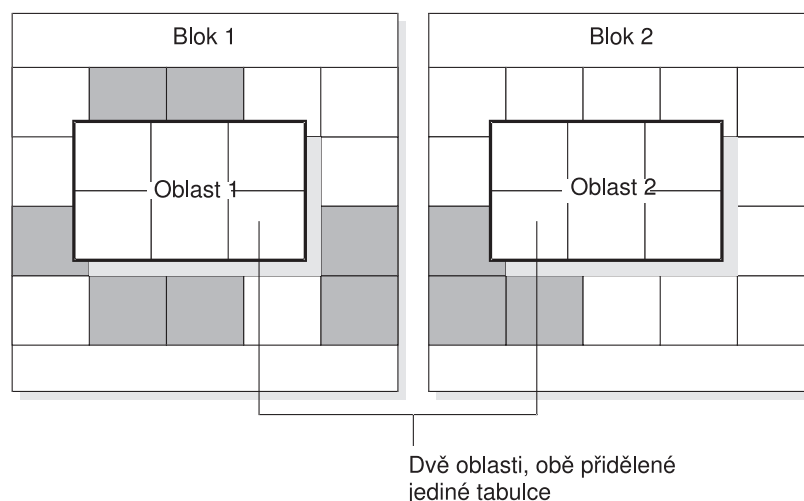
Je zapotřebí definovat schéma distribuce, jenž určuje, které řádky tabulky budou uloženy v jednotlivých prostorech dbspace.

- Vytvořit několik oddílů fragmentované tabulky v rámci jediného prostoru dbspace, pokud fragmentovaná tabulka používá schéma distribuce založené na výrazu nebo typu cyklická obsluha.

Tabulky a fragmenty tabulek jsou celé uloženy v prostoru dbspace, ve kterém byly vytvořeny. Díky tomu může administrátor databázového serveru omezit růst tabulky tím, že umístí tabulku do prostoru dbspace a po jeho zaplnění nepovolí přidání dalších bloků k tomuto prostoru.

Další informace o schématech distribuce naleznete v příručce *IBM Informix Database Design and Implementation Guide*. Informace o zlepšování výkonu, souběžného zpracování, dostupnosti dat a zálohování prostřednictvím fragmentace tabulek a indexů na více disků naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Tabulka složená z oblastí se může rozprostírat přes několik bloků, jak znázorňuje Obrázek 9-14.



Obrázek 9-14. Tabulka, která se rozkládá přes více bloků

Jednoduché velké objekty jsou uloženy ve stránkách blobpage, a to v prostoru dbspace obsahujícím datové stránky nebo v samostatném prostoru blobspace. Pokud používáte podsystém Optical Subsystem, můžete jednoduché velké objekty ukládat také do optického paměťového podsystému.

Informace o tom, kam je vhodné umístit tabulky, naleznete v části “Pravidla pro rozvržení disků” na stránce 9-33 a v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

## Poškozené tabulky

Následující položky mohou poškodit tabulku:

- Chybné vyprázdnění vyrovnávací paměti.
- Uživatelská chyba.
- Připojení systému souborů nebo dalšího bloku na horní část bloku.
- Odstranění nebo aktualizace, není-li oblast změn omezená podle potřeb uživatele.

Poškozené indexy mohou způsobit, že bude tabulka vypadat poškozená, ačkoliv je ve skutečnosti v pořádku.

Pomocí příkazů **oncheck** nelze opravit většinu poškozených tabulek. Pokud je stránka poškozená, pomocí příkazu **oncheck** můžete zjistit a opravit poškozenou stránku, ale nebude možné opravit data v ní obsažená.

## Typy tabulek serveru Dynamic Server

V databázi serveru Dynamic Server s protokolováním můžete vytvářet tabulky s protokolováním i bez protokolování. Tyto dva typy tabulek jsou STANDARD (tabulka s protokolováním) a RAW (tabulka bez protokolování). Výchozí standardní tabulka odpovídá tabulkám, které byly v dřívějších verzích vytvářeny bez určení zvláštního klíčového slova. Můžete vytvořit tabulku typu STANDARD nebo RAW a později změnit její typ na opačný.

V databázi bez protokolování jsou tabulky typu STANDARD i typu RAW tabulkami bez protokolování. V databázi bez protokolování se tabulky typu STANDARD a tabulky typu RAW liší pouze tím, že tabulky typu RAW neumožňují používat indexy a referenční omezení.

Tabulka 9-4 uvádí typy tabulek dostupných v serveru Dynamic Server. Hodnoty příznaku jsou hexadecimální hodnoty každého typu tabulky ve sloupci **flags** tabulky **systables**.

Tabulka 9-4. Typy tabulek serveru Dynamic Server

| Charakteristika           | STANDARD | RAW                           | TEMP  |
|---------------------------|----------|-------------------------------|-------|
| Je tabulka trvalá?        | Ano      | Ano                           | Ne    |
| Je tabulka protokolovaná? | Ano      | Ne                            | Ano   |
| Indexy?                   | Ano      | Ne                            | Ano   |
| Odvolání transakce        | Ano      | Ne                            | Ano   |
| Podporuje obnovu?         | Ano      | Ano, pokud není aktualizovaná | Ne    |
| Lze tabulku obnovit?      | Ano      | Ano, pokud není aktualizovaná | Ne    |
| Lze tabulku zavést?       | Ano      | Ano                           | Ano   |
| Enterprise Replication    | Ano      | Ne                            | Ne    |
| Hodnota příznaků          | Žádné    | 0x10                          | Žádné |

## Standardní trvalé tabulky

Tabulka typu STANDARD je tabulka, kterou databázový server vytváří v databázi s protokolováním. Tabulky typu STANDARD nepoužívají lehká připojení. Všechny operace jsou protokolovány po jednotlivých záznamech, proto tabulky STANDARD podporují obnovu a odvolávání transakcí. Tabulky typu STANDARD můžete zálohovat a obnovovat. Protokolování umožňuje při teplém obnovení a při obnovení do bodu v čase použít aktualizace provedené od poslední fyzické zálohy k aktualizaci tabulky. Tabulky typu STANDARD lze replikovat pomocí replikace Enterprise Replication.

Tabulka typu STANDARD je výchozí typ tabulky v databázích s protokolováním i v databázích bez protokolování. Tabulky typu STANDARD jsou protokolovány, pokud jsou uloženy v databázi s protokolováním, nejsou však protokolovány, pokud jsou uloženy v databázi bez protokolování.

## Tabulky typu RAW

Tabulky typu RAW jsou trvalé tabulky bez protokolování, které se podobají tabulkám v databázi bez protokolování. Tabulky typu RAW používají *lehká připojení*, pomocí kterých lze rychle přidávat řádky na konce jednotlivých fragmentů tabulky. Tabulky typu RAW podporují aktualizaci, vkládání i odstraňování řádků, tyto operace však nejsou protokolovány. Tabulky s přímým přístupem nepodporují referenční a jedinečná omezení a odvolávání transakcí. Tabulku typu RAW můžete obnovit z poslední fyzické zálohy, pokud nebyla od provedení



této zálohy aktualizována. Při rychlé obnově jsou odvolány všechny nedokončené transakce v tabulkách typu STANDARD, ale nikoli v tabulkách typu RAW. Tabulka typu RAW má tytéž atributy bez ohledu na to, zda je uložena v databázi s protokolováním nebo v databázi bez protokolování.

Tabulky typu RAW jsou určeny k počátečnímu zavedení a ověření dat. K zavedení tabulek typu RAW můžete použít jakýkoli zaváděcí obslužný program včetně obslužného programu **dbexport** a programu High-Performance Loader (zavaděč HPL) v expresním režimu. Pokud dojde k selhání při zavádění tabulky typu RAW, budou výsledná data odpovídat datům uloženým na disku v okamžiku selhání.

Není doporučeno používat tabulky s přímým přístupem v rámci transakce. Jakmile zavedete data, změňte pomocí příkazu ALTER TABLE typ tabulky na typ STANDARD a proveďte zálohu úrovně 0, než použijete tabulku v transakci.

**Upozornění:** Nepoužívejte replikaci Enterprise Replication pro tabulky typu RAW nebo TEMP.

## Tabulky typu TEMP

Tabulky typu TEMP jsou dočasné tabulky s protokolováním, které jsou vypouštěny při ukončení uživatelské relace, při vypnutí databázového serveru a při restartování po selhání serveru. Dočasné tabulky podporují indexy, omezení a odvolávání transakcí. Dočasné tabulky nelze obnovit, zálohovat nebo obnovovat. Dočasné tabulky podporují hromadné operace, jako jsou například lehká připojení, pomocí kterých lze rychle přidávat řádky na konce jednotlivých fragmentů tabulky. Další informace o lehkých připojeních naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Další informace naleznete v části “Dočasné tabulky” na stránce 9-26.

## Vlastnosti typů tabulek

Tato část popisuje zavádění tabulek, rychlou obnovu a zálohování a obnovu jednotlivých typů tabulek.

### Zavedení dat do tabulky

Server Dynamic Server ve výchozím nastavení vytváří tabulky typu STANDARD, které používají protokolování. Aplikace datových skladů mohou používat obrovské tabulky, jejichž zavedení může trvat dlouhou dobu. Tabulky bez protokolování se zavádějí rychleji než tabulky s protokolováním. Před zavedením tabulky můžete vytvořit tabulku typu RAW pomocí příkazu CREATE RAW TABLE nebo změnit tabulku typu STANDARD na tabulku typu RAW pomocí příkazu ALTER TABLE. Po zavedení tabulky spusíte pro tuto tabulku příkaz UPDATE STATISTICS.

Další informace o tom, jak zvýšit výkon při zavádění velkých tabulek naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*. Další informace o použití příkazu ALTER TABLE ke změně tabulky bez protokolování na tabulku s protokolováním naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

### Rychlá obnova jednotlivých typů tabulek

Obrázek Tabulka 9-5 znázorňuje scénáře rychlé obnovy pro jednotlivé typy tabulek dostupné v serveru Dynamic Server.

Tabulka 9-5. Rychlá obnova jednotlivých typů tabulek

| Typ tabulky | Chování při rychlé obnově                                                                                                                |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------|
| STANDARD    | Rychlá obnova proběhne úspěšně. Všechny potvrzené záznamy protokolu budou znovu přehrány a všechny nedokončené transakce budou odvolány. |

Tabulka 9-5. Rychlá obnova jednotlivých typů tabulek (pokračování)

| Typ tabulky | Chování při rychlé obnově                                                                                                                                                                                                                                                                                  |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RAW</b>  | <p>Pokud byl od poslední změny tabulky typu RAW dokončen kontrolní bod, bude možné obnovit všechna data.</p> <p>Budou ztracena veškerá vložená data, aktualizace dat a odstranění dat, ke kterým došlo po posledním kontrolním bodu.</p> <p>Nedokončené transakce v tabulce typu RAW nebudou odvolány.</p> |

## Zálohování a obnovení tabulek typu RAW

Tabulka 9-6 popisuje scénáře zálohování pro jednotlivé typy tabulek dostupné v serveru Dynamic Server.

Tabulka 9-6. Zálohování tabulek serveru Dynamic Server

| Typ tabulky     | Je zálohování povoleno?                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>STANDARD</b> | Ano.                                                                                                                                                                          |
| <b>TEMP</b>     | Ne.                                                                                                                                                                           |
| <b>RAW</b>      | Ano. Pokud aktualizujete tabulku typu RAW, je třeba tabulku zálohovat, aby mohla být obnovena všechna data uložená v tabulce. Zálohování logických protokolů není dostatečné. |

**Důležité:** Po zavedení tabulky typu RAW nebo změně typu tabulky z typu RAW na typ STANDARD je třeba provést zálohu úrovně 0.

Obrázek Tabulka 9-7 znázorňuje scénáře obnovení pro tyto typy tabulek.

Tabulka 9-7. Obnovení tabulek serveru Dynamic Server

| Typ tabulky     | Je obnovení povoleno?                                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>STANDARD</b> | Ano. Lze použít teplé obnovení, studené obnovení a obnovení k bodu v čase.                                                                                                                                                     |
| <b>TEMP</b>     | Ne.                                                                                                                                                                                                                            |
| <b>RAW</b>      | Pokud obnovíte tabulku typu RAW, bude obsahovat pouze data uložená v tabulce v okamžiku posledního zálohování. Protože tabulky typu RAW nejsou protokolovány, nebudou obnoveny žádné změny provedené od posledního zálohování. |

## Dočasné tabulky

Databázový server potřebuje poskytovat diskový prostor následujícím dvěma druhům dočasných tabulek:

- Dočasným tabulkám vytvářeným pomocí příkazů jazyka SQL, například příkazů CREATE TEMP TABLE... a SELECT .... INTO SCRATCH.
- Dočasným tabulkám, které databázový server vytváří při zpracování dotazu.

Presvědčte se, zda má databázový server nakonfigurován dostatek dočasného prostoru pro dočasné tabulky vytvářené uživatelem i serverem. Některé způsoby využití databázového serveru mohou vyžadovat dočasný paměťový prostor stejně velký jako trvalý paměťový prostor, nebo i větší.

Databázový server ve výchozím nastavení ukládá dočasné tabulky v kořenovém prostoru dbspace. Pokud se rozhodnete neukládat dočasné tabulky v kořenovém prostoru dbspace, určete pomocí proměnné prostředí **DBSPACETEMP** nebo konfiguračního parametru **DBSPACETEMP** seznam prostorů dbspace pro dočasné tabulky.

## Uživatelské dočasné tabulky

Dočasnou tabulku můžete vytvořit kterýmkoli z následujících příkazů jazyka SQL:

- Volbou TEMP TABLE příkazu CREATE TABLE.
- Pomocí klauzule INTO TEMP příkazu SELECT, například příkazem `SELECT * FROM customer INTO TEMP cust_temp`.

Tuto tabulku lze použít pouze v relaci, ve které byla vytvořena. Po ukončení relace bude tabulka automaticky vypuštěna.

Pokud vytváříte dočasnou tabulku, použije databázový server následující kritéria:

- Pokud dotaz použitý k naplnění tabulky typu TEMP nevrátil žádné řádky, vytvoří databázový server prázdnou, nefragmentovanou tabulku.
- Pokud celková velikost řádků vrácených dotazem nepřekračuje 8 kB, bude dočasná tabulka uložena v jediném prostoru dbspace.
- Pokud celkový objem řádků překračuje 8 kB a pro tabulku nebyla zadána metoda fragmentace a umístění, vytvoří databázový server několik fragmentů a použije schéma fragmentace typu cyklická obsluha.

Pokud použijete příkazy CREATE TEMP a SELECT...INTO TEMP jazyka SQL a byl nastaven parametr DBSPACETEMP:

- Prostory dbspace S PROTOKOLOVÁNÍM uvedené v seznamu budou použity k vytvoření tabulek, u kterých je určena klauzule WITH LOG nebo u kterých by byla použita implicitně.
- Prostory dbspace BEZ PROTOKOLOVÁNÍ uvedené v seznamu budou použity k vytvoření tabulek, u kterých je určena klauzule WITH NO LOG.

Pokud byly použity příkazy CREATE TEMP a SELECT...INTO TEMP jazyka SQL a parametr DBSPACETEMP nebyl nastaven nebo neobsahuje prostory dbspace správného typu, použije server Dynamic Server k uložení dočasných tabulek prostor dbspace databáze. Další informace naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

**Kam jsou ukládány uživatelské dočasné tabulky:** Pokud aplikace umožňuje určit umístění dočasných tabulek, můžete určit buď prostory s protokolováním nebo prostory bez protokolování vytvořené výhradně k ukládání dočasných tabulek. Tabulky typu SCRATCH nejsou protokolovány a musejí být vytvářeny v dočasných prostorech.

Informace o vytváření dočasných prostorů dbspace a dbslice naleznete v části věnované obslužnému programu **onspaces**.

Pokud neurčíte umístění dočasných tabulek, uloží databázový server dočasnou tabulku v jednom z prostorů zadaných jako argument konfiguračního parametru nebo proměnné prostředí DBSPACETEMP. Databázový server zaznamenává název posledního prostoru dbspace použitého k uložení dočasných tabulek. Jakmile databázový server přijme další požadavek na dočasný paměťový prostor, použije další dostupný prostor dbspace, aby rozložil vstup - výstup přes všechny dostupné dočasné paměťové prostory.

Informace o tom, kam jsou v databázi ukládány dočasné tabulky, pokud nejsou uvedeny žádné prostory jako argument konfiguračního parametru DBSPACETEMP naleznete v části týkající se parametru DBSPACETEMP v příručce *IBM Informix Administrator's Reference*.

Pokud k vytvoření dočasných tabulek použijete aplikaci, budete tabulku moci používat, dokud aplikace nebude ukončena nebo dokud neprovede jednu z následujících akcí:

- Zavře databázi, ve které byla tabulka vytvořena a otevře databázi v jiném databázovém serveru.

- Zavře databázi, ve které byla tabulka vytvořena.
- Explicitně vypustí dočasnou tabulku.

### Dočasné tabulky vytvářené databázovým serverem

Databázový server občas vytváří dočasné tabulky při spouštění dotazů v databázi nebo při jejím zálohování. Databázový server může vytvořit dočasnou tabulku, kdykoli jsou použity následující příkazy:

- Příkazy s klauzulemi GROUP BY nebo ORDER BY.
- Příkazy, které používají agregační funkce s klíčovými slovy UNIQUE nebo DISTINCT.
- Příkazy SELECT, které používají automatické indexování nebo spojení typu hash.
- Komplexní příkazy CREATE VIEW.
- Příkazy DECLARE, které vytvářejí posouvateľné kurzory.
- Příkazy obsahující souvztažné poddotazy.
- Příkazy, které obsahují poddotazy v klauzuli IN nebo ANY.
- Příkazy CREATE INDEX.

Databázový server odstraní vytvořené dočasné tabulky po ukončení procesu, který způsobil jejich vytvoření.

Pokud je databázový server vypnut bez odstranění dočasných tabulek, provede vyčištění dočasných tabulek při příští inicializaci sdílené paměti. Pokud chcete inicializovat sdílenou paměť bez vyčištění dočasných tabulek, spusíte příkaz **oninit** s volbou **-p**.

**Důležité:** Kromě dočasných tabulek ukládá databázový server do dočasného diskového prostoru i předobrazy dat, které jsou přepisovány v průběhu zálohování a při zpracování dotazů přetečou z paměti. Přesvědčte se, že je nakonfigurován dostatek dočasného prostoru pro všechna možná použití.

**Kam jsou ukládány dočasné tabulky vytvářené databázovým serverem:** Když databázový server vytváří dočasnou tabulku, ukládá tuto tabulku do jednoho z prostorů dbSPACE určených pomocí konfiguračního parametru DBSPACETEMP nebo pomocí proměnné prostředí DBSPACETEMP. Zde platí, že proměnná prostředí potlačuje odpovídající konfigurační parametr.

Pokud neurčíte pomocí parametru DBSPACETEMP žádný dočasný prostor dbSPACE nebo pokud dočasné prostory dbSPACE nejsou dostatečně velké, vytvoří databázový server tabulku ve standardním prostoru dbSPACE podle následujících pravidel:

- Pokud jste dočasnou tabulku vytvořili pomocí příkazu CREATE TEMP TABLE, vytvoří databázový server tuto tabulku v prostoru dbSPACE obsahujícím databázi, do které tato tabulka patří.
- Pokud jste dočasnou tabulku vytvořili pomocí volby INTO TEMP příkazu SELECT, uloží databázový server tuto tabulku do kořenového prostoru dbSPACE.

Další informace naleznete v části “Vytvoření dočasného prostoru dbSPACE” na stránce 10-17.

---

## Prostory tblspace

Administrátoři databází mohou občas potřebovat sledovat využití diskového prostoru konkrétní tabulkou. *Prostor tblspace* obsahuje veškerý diskový prostor přidělený konkrétní tabulce nebo fragmentu tabulky (pokud je tabulka fragmentovaná). Samostatný prostor *tblspace* obsahuje diskový prostor přidělený přidruženému indexu.

Prostor tblspace neodpovídá například žádné konkrétní části bloku ani žádnému konkrétnímu bloku. Indexy a data tvořící prostor tblspace mohou být náhodně roztroušeny v různých blocích. Prostor tblspace však představuje praktickou účetní jednotku, pokud jde o prostor jednotlivých bloků vyhrazených konkrétní tabulce. (Další informace naleznete v části “Tabulky” na stránce 9-22.)

## Maximální počet prostorů tblspace v tabulce

Pro tabulku můžete určit nejvýše 2\*\*20 (nebo 1048576) prostorů tblspace.

## Prostory tblspace tabulek a indexů

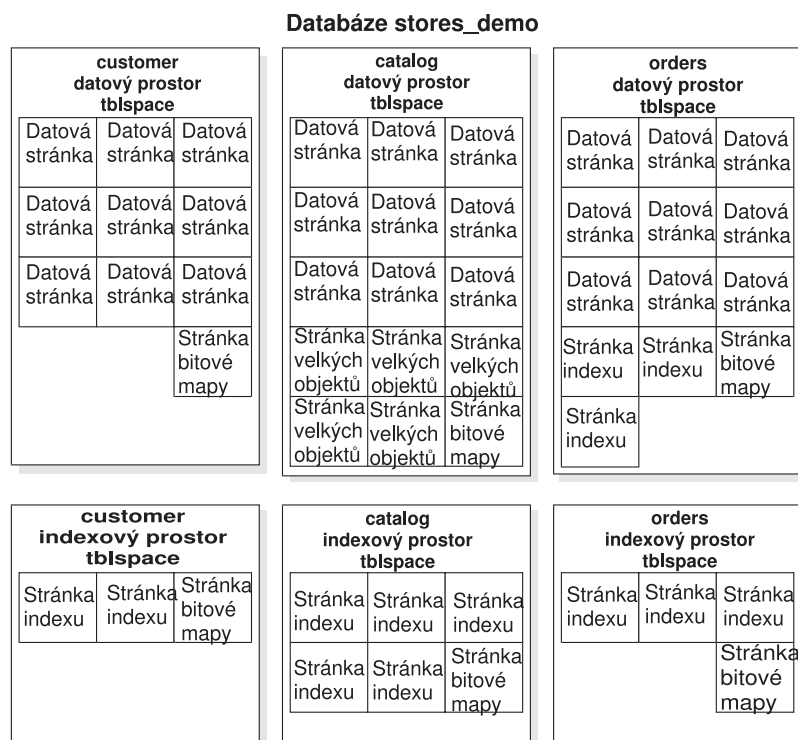
Prostor tblspace tabulky obsahuje následující typy stránek:

- Stránky přidělené datům.
- Stránky přidělené indexům.
- Stránky použité k uložení dat typu TEXT a BYTE v prostoru dbspace. (Neobsahuje však stránky dat typu TEXT a BYTE uložené v prostorech blobspace.)
- Stránky bitových map, které zaznamenávají využití stránek v rámci oblastí tabulky.

Prostor tblspace indexu obsahuje následující typy stránek:

- Stránky přidělené indexům.
- Stránky bitových map, které zaznamenávají využití stránek v rámci oblastí indexu.

Obrázek Obrázek 9-15 znázorňuje prostory tblspace tří tabulek, které tvoří část databáze **stores\_demo**. V každém prostoru tblspace se nachází pouze jedna tabulka (nebo fragment tabulky). Index je uložen v samostatném prostoru tblspace, nikoli v prostoru tblspace přidružené tabulky. Stránky blobpage představují data typu TEXT a BYTE uložená v prostoru dbspace.

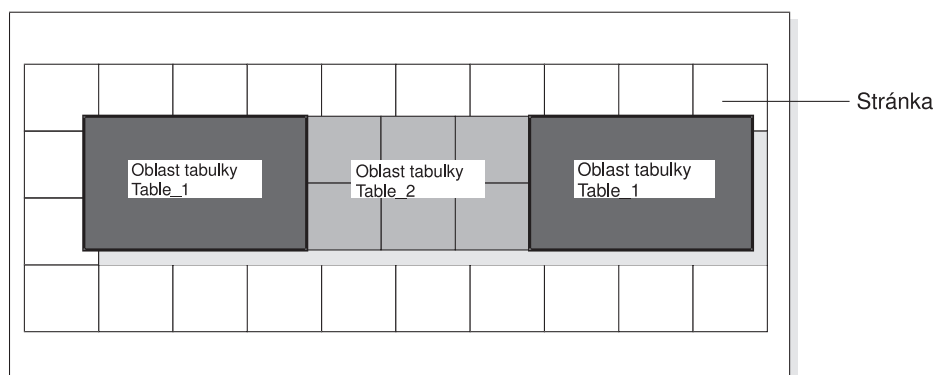


Obrázek 9-15. Ukázkové prostory tblspace v databázi stores\_demo

## Prokládání oblastí

Databázový server přiděluje stránky tvořící prostor tblspace jako oblasti. Přestože stránky tvořící oblast spolu sousedí, mohou oblasti být roztroušené po prostoru dbspace, ve kterém je tabulka uložena (a to i v různých blocích).

Obrázek Obrázek 9-16 znázorňuje tuto situaci v podobě dvou nesousedících oblastí, které patří do prostoru tblspace tabulky **table\_1** a třetí oblasti, která patří do prostoru tblspace tabulky **table\_2**. Oblast tabulky **table\_2** se nachází mezi první a druhou oblastí tabulky **table\_1**. Pokud dojde k této situaci, budou oblasti prokládané. Protože prohledávání tabulky **table\_1** sekvenčním přístupem vyžaduje, aby se diskové hlavy přesouvaly přes oblast tabulky **table\_2**, bude výkon serveru nižší, než kdyby tabulku **table\_1** tvořily sousedící oblasti. Pokyny, jak zabránit prokládání oblastí nebo ho odstranit naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.



Obrázek 9-16. Tři oblasti, které patří do dvou různých prostorů tblspace v jediném prostoru dbspace.

## Fragmentace tabulek a uložení dat

Funkce fragmentace poskytuje větší kontrolu nad umístěním dat databázového serveru. Určit můžete nejen umístění jednotlivých tabulek a indexů. Můžete také určit umístění *fragmentů* tabulek a indexů, což jsou části tabulky nebo indexu, které jsou uloženy v různých paměťových prostorech. Fragmentovat můžete následující paměťové prostory:

- prostory dbspace
- prostory sbspace

Fragmentace tabulky se obvykle provádí při prvotním vytvoření tabulky. Příkaz CREATE TABLE má jednu z následujících forem:

```
CREATE TABLE název_tabulky ... FRAGMENT BY ROUND ROBIN IN dbspace1,
dbspace2, dbspace3;
```

```
CREATE TABLE název_tabulky ...FRAGMENT BY EXPRESSION
<Výraz 1> in dbspace1,
<Výraz 2> in dbspace2,
<Výraz 3> in dbspace3;
```

Klíčová slova FRAGMENT BY ROUND ROBIN a FRAGMENT BY EXPRESSION odkazují na dvě různá schémata distribuce. Oba příkazy přiřazují fragmenty prostorům dbspace.

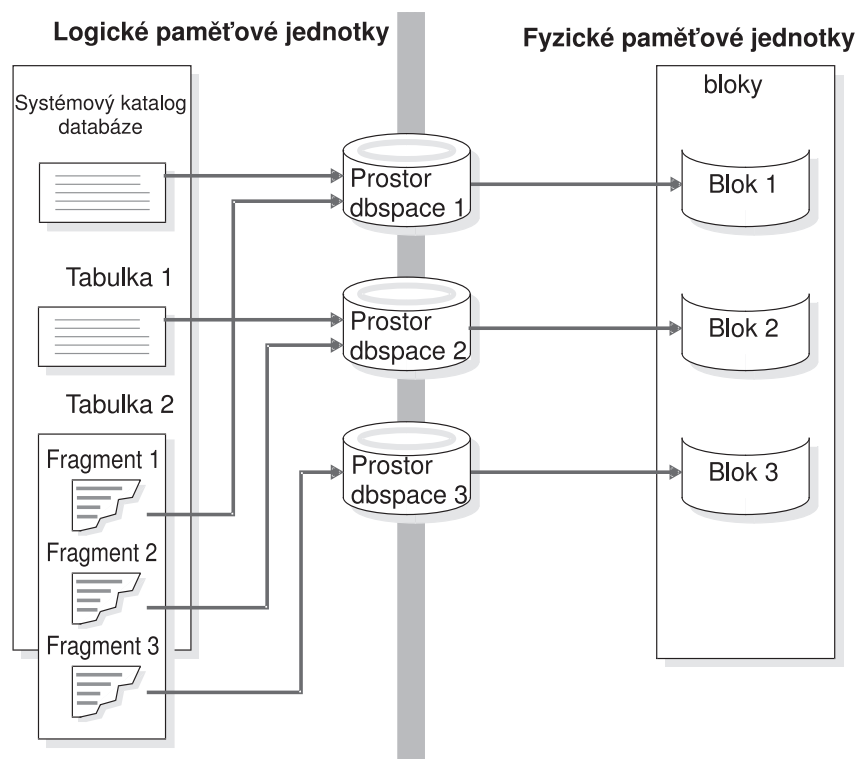
Pokud fragmentujete tabulku, můžete také vytvořit několik oddílů tabulky v jediném prostoru dbspace, jak znázorňuje následující příklad:

```

CREATE TABLE tb1(a int)
 FRAGMENT BY EXPRESSION
 PARTITION part1 (a >=0 AND a < 5) in dbs1,
 PARTITION part2 (a >=5 AND a < 10) in dbs1
 ...
;

```

Obrázek Obrázek 9-17 znázorňuje roli fragmentů při určování umístění dat.



Obrázek 9-17. Prostory dbspace, které propojují logické paměťové jednotky (včetně fragmentů tabulek) a fyzické paměťové jednotky

Informace o prostorech a oddílech uvádí Kapitola 10, “Správa diskového prostoru”, na stránce 10-1.

Další informace o fragmentaci naleznete v příručce *IBM Informix Database Design and Implementation Guide* a v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

## Objem diskového prostoru potřebný k uložení dat

K určení potřebného diskového prostoru proveďte následující kroky:

### Postup určení potřebného diskového prostoru:

1. Vypočtete požadavky na velikost kořenového prostoru dbspace.
2. Odhadněte celkový objem diskového prostoru, který bude přidělen všem databázím databázového serveru včetně prostoru pro režii a růst databáze.

Tyto kroky jsou vysvětleny v následujících částech.

## Velikost kořenového prostoru dbspace

Při výpočtu velikosti kořenového prostoru dbspace vezměte v úvahu následující paměťové struktury:

- Fyzický protokol (minimálně 200 kB).
- Soubory logického protokolu (minimálně 200 kB).
- Dočasné tabulky
- Data
- Systémové databáze (**sysmaster**, **sysutils**, **syscdr**, **sysuuid**) a systémové katalogy (velikost se liší podle verze serveru).
- Rezervované stránky (přibližně 24 kB).
- Prostor tblspace tblspace (minimálně 100 až 200 kB).
- Další prostor.

Tento odhad je počáteční velikost kořenového prostoru dbspace před inicializací databázového serveru. Počáteční velikost kořenového prostoru dbspace závisí na tom, zda plánujete ukládat fyzický protokol, logické protokoly a dočasné tabulky do kořenového prostoru dbspace nebo do jiného prostoru dbspace. Kořenový prostor dbspace musí být dostatečně velký na to, aby pojmul konfiguraci minimální velikosti při inicializaci disku.

Je doporučeno inicializovat systém s malou velikostí protokolu (například tři soubory protokolu po 1000 kB nebo o celkové velikosti protokolu 3000 kB). Po dokončení inicializace vytvořte nové prostory dbspace, přesuňte soubory logického protokolu a změňte jejich velikost a v kořenovém prostoru dbspace původní protokoly vypusťte. Pak přesuňte fyzický protokol do jiného prostoru dbspace. Tímto postupem se minimalizuje vliv používání protokolů na kořenový prostor dbspace, a to z následujících důvodů:

- V kořenovém prostoru dbspace není po přesunutí prostorů velký objem prostoru nevyužit (pokud kořenový prostor dbspace neroste).
- Protokoly nesoupeří o prostor a operace vstupu - výstupu na téže disku, na kterém je uložen kořenový prostor dbspace (pokud kořenový prostor dbspace roste).

Podrobné informace o tom, jak přesunout protokoly naleznete v části “Přesun souboru logického protokolu do jiného prostoru dbspace” na stránce 14-14 a v části “Změna umístění a velikosti fyzického protokolu” na stránce 16-1.

Do kořenového prostoru dbspace můžete přidávat další bloky a prázdné bloky můžete vypouštět. Začněte s malým kořenovým prostorem dbspace a pokud bude systém růst, zvětšete ho podle potřeby. Nelze však začít s velkým počátečním kořenovým blokem a později ho zmenšit.

### Fyzické a logické protokoly

Hodnota uložená v konfiguračním parametru PHYSFILE souboru ONCONFIG určuje velikost fyzického protokolu při počátečním vytvoření databázového serveru. Po uvedení databázového serveru do režimu online a inicializaci diskového prostoru pomocí příkazu **oninit -i** použijte obslužný program **onparams** a změňte umístění logického protokolu a jeho velikost. Pokyny ke správnému určení velikosti logického protokolu naleznete v části “Velikost a umístění fyzického protokolu” na stránce 15-3.

Velikost souborů logického protokolu určíte vynásobením hodnoty parametru LOGSIZE v souboru ONCONFIG počtem souborů logického protokolu. Pokyny ke správnému určení velikosti logického protokolu naleznete v části “Odhad velikosti a počtu souborů protokolu” na stránce 14-2.



## Dočasné tabulky

Analýzou aplikací pro koncové uživatele odhadněte objem diskového prostoru, který bude databázový server pravděpodobně požadovat pro dočasné tabulky. Pokuste se odhadnout, kolik takových dotazů bude spouštěno souběžně. Vhodným ukazatelem, ze kterého můžete vyjít při odhadu velikosti požadovaného prostoru je prostor, který obsadí vrácené řádky a sloupce.

Velikost největší dočasné tabulky, kterou databázový server vytváří v průběhu teplého obnovení, je rovna velikosti logického protokolu. Velikost logického protokolu vypočtete sečtením velikosti všech souborů logického protokolu.

Analýzou aplikací pro koncové uživatele odhadněte také objem diskového prostoru, který bude databázový server pravděpodobně požadovat pro explicitní dočasné tabulky.

Další informace včetně seznamu příkazů požadujících dočasný prostor naleznete v části “Dočasné tabulky” na stránce 9-26.

## Kritická data

Nedoporučujeme ukládat databáze a tabulky do kořenového prostoru dbSPACE. Zrcadlete kořenový prostor dbSPACE a ostatní prostory dbSPACE obsahující kritická data, například fyzický protokol a logické protokoly. Odhadněte objem diskového prostoru, který je zapotřebí přidělit tabulkám ukládaným do kořenového prostoru dbSPACE.

## Další prostor

Ponechte v kořenovém prostoru dbSPACE další prostor pro růst systémových databází, rozšířené rezervované stránky a dostatek volného prostoru. Počet rozšířených rezervovaných stránek závisí na počtu primárních bloků, zrcadlených bloků, souborů logického protokolu a paměťových prostorů v databázovém serveru.

## Objem prostoru požadovaného databázemi

Objem dalšího diskového prostoru potřebného k uložení dat databázového serveru závisí na potřebách uživatelů, režii a růstu databáze. Každá aplikace spuštěná koncovými uživateli má jiné požadavky na paměť. Následující seznam uvádí některé kroky, pomocí kterých můžete vypočítat objem diskového prostoru, který je třeba přidělit (nad rámec kořenového prostoru dbSPACE):

- Určete, kolik databází a tabulek potřebujete uložit. Vypočtete objem prostoru potřebného k uložení každé z nich.
- Vypočtete rychlost růstu všech tabulek a přiřaďte každé tabulce určitý objem diskového prostoru, který bude schopen pojmout narůstající objem dat.
- Určete, které databáze a tabulky budete chtít zrcadlit.

Pokyny, jak vypočítat velikost tabulek naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

---

## Pravidla pro rozvržení disků

Efektivní rozvržení disků obvykle sleduje následující cíle:

- Omezení pohybu diskových hlav.
- Omezení kolizí požadavků na přístup k disku.
- Rozložení zatížení.
- Maximalizace dostupnosti.

Při návrhu rozvržení disků je třeba učinit jisté kompromisy. Můžete například oddělením tabulek systémového katalogu, logického protokolu a fyzického protokolu omezit kolize

požadavků na tyto zdroje. Zvýší se tím však také pravděpodobnost, že bude zapotřebí provést obnovení systému. Podrobná pravidla pro rozvržení disků naleznete v části *Příručka výkonosti serveru IBM Informix Dynamic Server*.

## Pravidla pro prostory dbspace a bloky

V této části jsou uvedeny některé obecné strategie rozvržení disků, které nevyžadují žádné informace o charakteristikách konkrétní databáze:

- Přidruzte diskové oddíly k blokům a přiřďte kořenovému prostoru dbspace alespoň jeden *další* blok.

Pokud je disk již rozdělen na oddíly, bude pravděpodobně nezbytné použít posuny. Další informace naleznete v části “Přidělení diskového prostoru s přímým přístupem v systému UNIX” na stránce 10-5.

**Rada:** Protože je velikost bloku omezena na 4 TB, můžete se vyhnout dělení disku na oddíly a přidělit celou diskovou jednotku jedinému bloku.

- Zrcadlete důležité prostory dbspace: kořenový prostor dbspace, prostory dbspace obsahující fyzický protokol a soubory logického protokolu. Zrcadlete také databáze a tabulky, které jsou často používány.

Zrcadlení se určuje na úrovni prostoru dbspace. Zrcadlení je zapnuté nebo vypnuté pro všechny bloky, které patří do prostoru dbspace. Primární a zrcadlené prostory dbspace umísťujte na různé disky. V ideálním případě by tyto různé disky měly být obsluhovány různými řadiči disků.

- Dočasné tabulky a soubory řazení rozložte na více disků.

K definování několika prostorů dbspace pro dočasné tabulky a soubory řazení použijte příkaz **onspaces -t**. Pokud tyto prostory dbspace umístíte na různé disky a uvedete je v konfiguračním parametru DBSPACETEMP, budete moci operace vstupu - výstupu spojené s použitím dočasných tabulek a souborů řazení rozložit na více disků. Další informace o konfiguračním parametru nebo proměnné prostředí DBSPACETEMP naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.

- Fyzický protokol ponechte v kořenovém prostoru dbspace, ale logické protokoly z kořenového prostoru dbspace přesuňte. Pokud však budete do kořenového prostoru dbspace chtít ukládat systémové katalogy, přesuňte fyzický protokol do jiného prostoru dbspace.

Informace o tom, kam je vhodné umístit protokoly, naleznete v části “Určení umístění fyzického protokolu” na stránce 15-3 a v části “Umístění souborů logického protokolu” na stránce 13-2. Další informace naleznete také v části “Přesun souboru logického protokolu do jiného prostoru dbspace” na stránce 14-14 a v části “Změna umístění a velikosti fyzického protokolu” na stránce 16-1.

- Pokud chcete zvýšit výkon při zálohování a obnovování:

- Systémové katalogy umístěte společně s daty, o kterých tyto katalogy zaznamenávají informace.

- Pokud používáte program ON-Bar k paralelním zálohováním na vysokorychlostní páskovou jednotku, ukládejte databáze do několika malých prostorů dbspace.

Další doporučení týkající se výkonu naleznete v příručce *IBM Informix Backup and Restore Guide*.

## Pravidla umísťování tabulek

V této části jsou uvedeny některé strategie optimalizace rozvržení disků, které předpokládají znalost určitých charakteristik tabulek databáze. Mnoho z těchto strategií můžete implementovat s větší mírou kontroly, pokud použijete fragmentaci tabulek:

- Často používané tabulky izolujte na samostatný disk.

Často používanou tabulku můžete izolovat na samostatné diskové zařízení tak, že toto zařízení přidělíte bloku a tento blok přidělíte prostoru dbspace. Nakonec umístíte často používanou tabulku do vytvořeného prostoru dbspace pomocí volby IN *dbspace* příkazu CREATE TABLE.

Úroveň aktivity operací vstupu - výstupu všech bloků můžete zobrazit pomocí volby příkazu **onstat -g iof**.

- Často používané tabulky fragmentujte na více disků.
- Seskupte související tabulky do prostoru dbspace.

Pokud dojde k selhání zařízení obsahujícího prostor dbspace, budou všechny tabulky v tomto prostoru dbspace nepřístupné. Tabulky v jiných prostorech dbspace však zůstanou přístupné. Přestože po selhání prostoru dbspace obsahujícího kritické informace je třeba provést studené obnovení, po selhání nekritického prostoru dbspace stačí provést pouze teplé obnovení.

- Často používané tabulky umístěte do oddílu uprostřed disku.
- Optimalizujte velikost oblastí tabulek.

Další informace naleznete v kapitole o posouzení výkonu tabulek v příručce *Řízení výkonu systému IBM Informix*. Informace o volbách obslužného programu **onstat** naleznete v příručce *IBM Informix Administrator's Reference*.

---

## Ukázková rozvržení disku

Při návrhu rozvržení diskového prostoru sleduje obvykle administrátor databázového serveru jeden nebo více následujících cílů:

- dosažení vysokého výkonu,
- dosažení vysoké dostupnosti,
- usnadnění zálohování a obnovení a umožnění častého provádění těchto operací.

K dosažení každého z těchto cílů je třeba učinit kompromisní rozhodnutí. Konfigurace systému k dosažení vysoké dostupnosti například obvykle zvyšuje rizika související se ztrátou dostupnosti dat. V následujících částech jsou uvedeny příklady, kdy musí administrátor databázového serveru zvolit rozvržení disků v situaci, kdy jsou diskové zdroje omezené. Tyto části popisují dvě různá řešení problému rozvržení disků. První řešení optimalizuje výkon serveru a druhé řešení optimalizuje dostupnost a obnovitelnost serveru.

Tato ukázková rozvržení disků jsou navrhována pro databázi fiktivního obchodu se sportovními potřebami, která využívá strukturu databáze **stores\_demo** (má však jiný objem dat). V tomto příkladu je databázový server nakonfigurován tak, aby dokázal obsloužit přibližně 350 uživatelů a 3 GB dat. Zdroje diskového prostoru jsou znázorněny v následující tabulce.

| Disková jednotka | Velikost jednotky | Vysoký výkon |
|------------------|-------------------|--------------|
| Disk 1           | 2,5 GB            | Ne           |
| Disk 2           | 3 GB              | Ano          |
| Disk 3           | 2 GB              | Ano          |
| Disk 4           | 1,5 GB            | Ne           |

Databáze obsahuje dvě velké tabulky: **cust\_calls** a **items**. Předpokládejme, že obě tabulky obsahují více než 1000000 řádků. Tabulka **cust\_calls** reprezentuje záznamy všech telefonních hovorů distributorovi uskutečněných zákazníky. Tabulka **items** obsahuje jeden řádek pro každou objednávku, kterou kdy distributor dodal zákazníkovi.

Databáze obsahuje dvě často používané tabulky: **items** a **orders**. K oběma těmto tabulkám neustále přistupují uživatelé z celé země.

Zbývající tabulky jsou malé tabulky, které databázový server používá k vyhledávání takových dat, jako jsou poštovní směrovací čísla, výrobci apod.

| Název tabulky | Maximální velikost | Četnost přístupů |
|---------------|--------------------|------------------|
| cust_calls    | 2,5 GB             | Nízká            |
| items         | 0,5 GB             | Vysoká           |
| orders        | 50 MB              | Vysoká           |
| customers     | 50 MB              | Nízká            |
| stock         | 50 MB              | Nízká            |
| catalog       | 50 MB              | Nízká            |
| manufact      | 50 MB              | Nízká            |
| state         | 50 MB              | Nízká            |
| call_type     | 50 MB              | Nízká            |

## Ukázkové rozvržení upřednostňující výkon

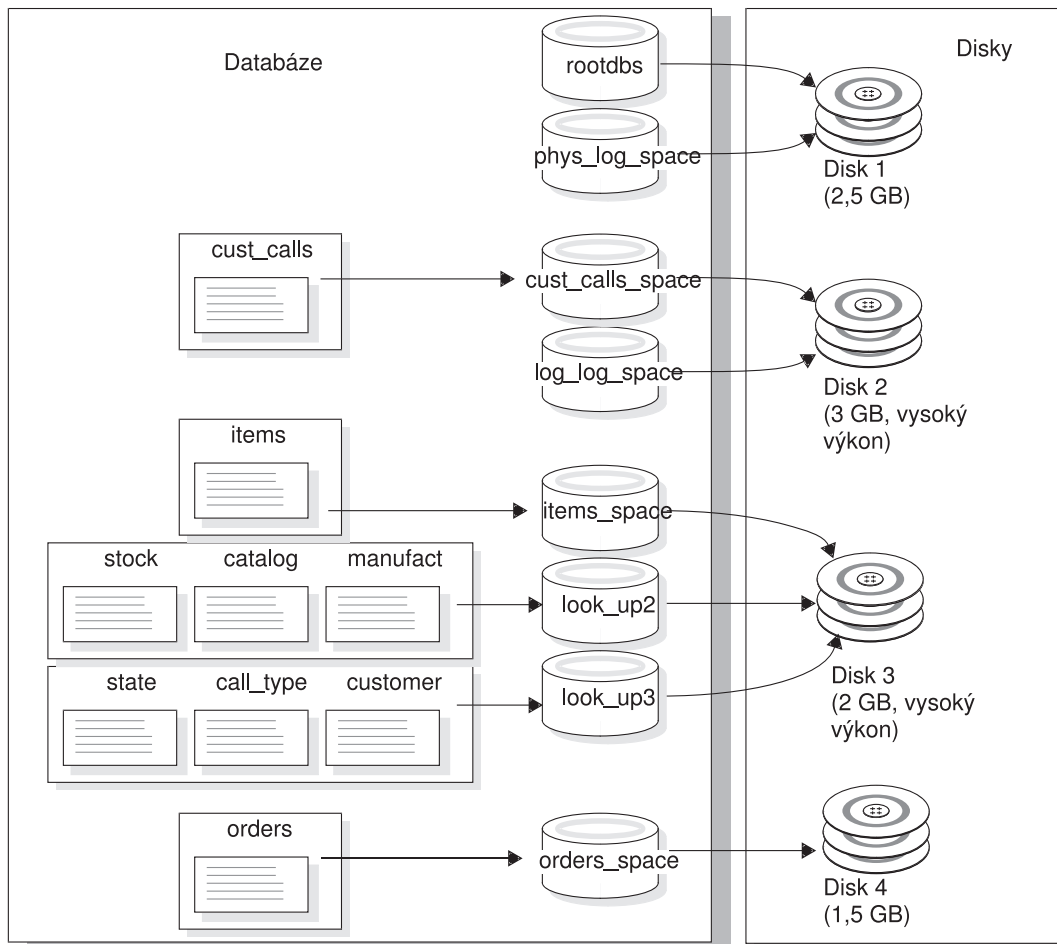
Obrázek Obrázek 9-18 na stránce 9-37 znázorňuje rozvržení disků optimalizované na výkon. Toto rozvržení disků používá ke zvýšení výkonu následující strategie:

- Přesouvá logický protokol z prostoru dbspace rootdbs do prostoru dbspace na samostatném disku.

Tato strategie odděluje logický protokol a fyzický protokol a omezuje kolize požadavků na kořenový prostor dbspace.

- Umísťuje dvě nejpoužívanější tabulky do prostorů dbspace na samostatných discích.

Na žádném z těchto disků není uložen logický ani fyzický protokol. V ideálním případě by měla být tabulka **items** i tabulka **orders** uložena na vlastním samostatném disku s vysokým výkonem. V tomto scénáři však taková strategie není možná, protože jeden z disků s vysokým výkonem je zapotřebí k ukládání velmi rozsáhlé tabulky **cust\_calls** (zbylé dva disky jsou k tomuto účelu příliš malé).

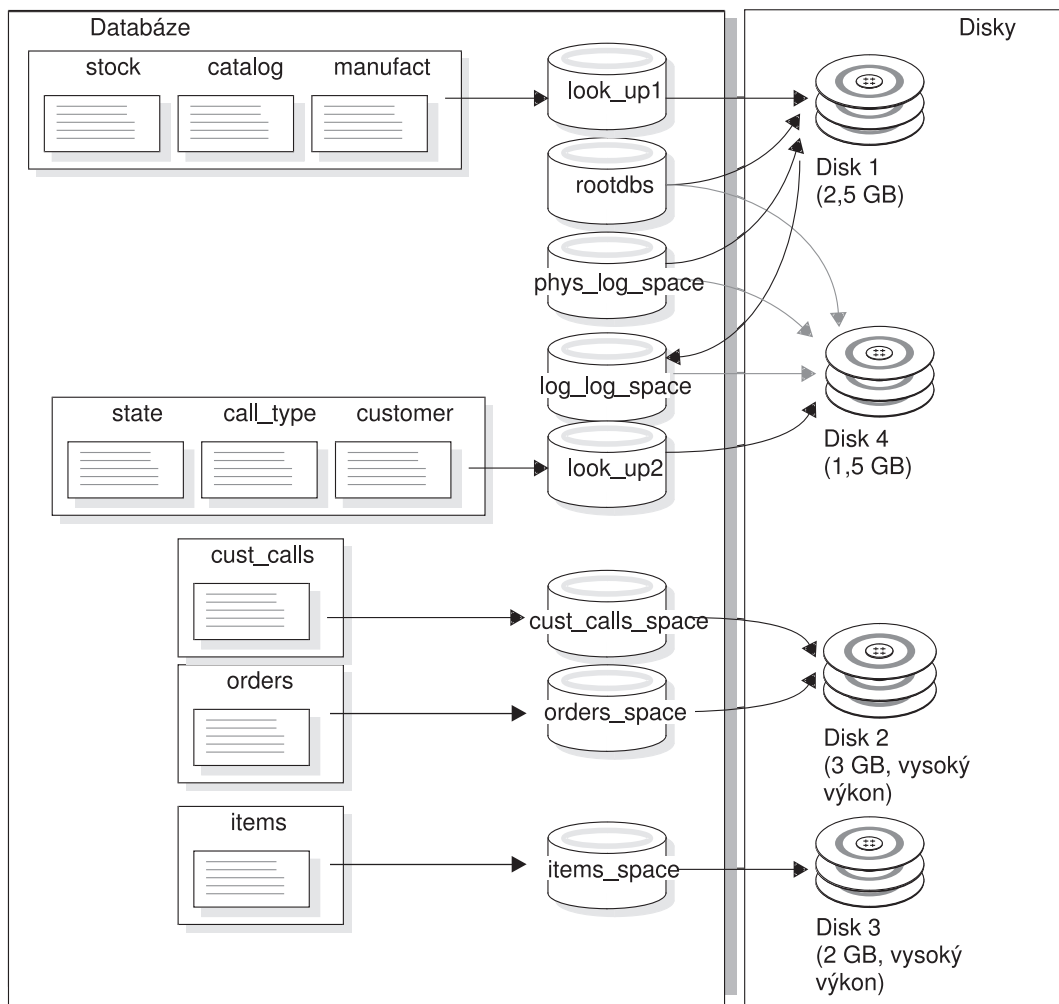


Obrázek 9-18. Rozvržení disků optimalizované na výkon

## Ukázkové rozvržení upřednostňující dostupnost

Slabinou uvedeného rozvržení disků je skutečnost, že pokud dojde k selhání disku číslo 1 nebo 2, bude celý databázový server vypnut, dokud neobnovíte prostory dbspace ze záloh. Jinými slovy, toto rozvržení disků nevyhovuje náročnějším požadavkům na dostupnost.

Obrázek Obrázek 9-19 na stránce 9-38 znázorňuje alternativní rozvržení disků, které používá zrcadlení a optimalizuje dostupnost serveru. V tomto rozvržení jsou všechny kritické datové prostory (tabulky systémového katalogu, fyzický protokol a logický protokol) zrcadleny na samostatném disku. V ideálním případě by byly logický protokol a fyzický protokol odděleny (jako v předchozím rozvržení) a každý disk by byl zrcadlen na vlastní zrcadlový disk. V tomto scénáři však není dostupný požadovaný počet disků, proto jsou logický protokol i fyzický protokol uloženy v kořenovém prostoru dbspace.



Obrázek 9-19. Rozvržení disků optimalizované na dostupnost

## Správce logických disků

Správce logických disků (LVM) je obslužný program, který umožňuje spravovat diskový prostor prostřednictvím uživatelských logických disků.

Mnoho výrobců počítačů dodává počítače s proprietárním správcem LVM. Databázový server můžete použít k ukládání a načítání dat s použitím disků spravovaných většinou proprietárních správců LVM. Použití správců logických disků má některé výhody a některé nevýhody, které jsou popsány ve zbytku této části.

Většina správců LVM dokáže spravovat mnoho gigabajtů diskového prostoru. Velikost bloků databázového serveru je omezena na 4 TB a lze ji dosáhnout pouze tehdy, pokud je blok přidělen s nulovým posunem. Proto byste měli omezit velikost jakýchkoli svazků, které chcete přidělit jako bloky, na hodnotu 4 TB.

Protože správci LVM umožňují rozdělit diskovou jednotku na více logických disků, umožňují řídit, kam budou data na daném disku uložena. Výkon serveru můžete zvýšit tak, že vytvoříte logický disk skládající se z válců uprostřed disku a do něj umístíte často používané tabulky. (Přesněji řečeno, tabulky se neumisťují přímo na logický disk. Vytvořte nejprve blok jako

logický disk, pak přiřaďte blok prostoru dbspace a nakonec umístěte tabulku do tohoto prostoru dbspace. Další informace naleznete v části “Řízení umístění ukládaných dat” na stránce 9-9.)

**Rada:** Pokud se rozhodnete použít velké diskové jednotky, můžete přiřadit celé jednotce jeden blok a nebudete muset dělit disk na oddíly.

Výkon můžete zvýšit také tím, že pomocí správce logických disků definujete logický disk, který se rozkládá přes více disků a pak umístíte tabulku do tohoto disku.

Mnoho správců logických disků poskytuje natolik pružnou správu, jakou standardní formátovací obslužné programy operačního systému poskytnout nemohou. Jednou z těchto funkcí je například schopnost změnit pozici logických disků po jejich definování. Pak není správné rozvržení diskového prostoru na první pokus tak důležité, jako by bylo v případě použití obslužných programů operačního systému.

Správci LVM často poskytují funkce zrcadlení na úrovni operačního systému. Další informace naleznete v části “Alternativy k zrcadlení” na stránce 17-3.





---

## Kapitola 10. Správa diskového prostoru

|                                                                                                      |       |
|------------------------------------------------------------------------------------------------------|-------|
| Obsah kapitoly . . . . .                                                                             | 10-2  |
| Přidělení diskového prostoru . . . . .                                                               | 10-3  |
| Určení posunu . . . . .                                                                              | 10-4  |
| Určení posunu pro počáteční blok kořenového prostoru dbSPACE . . . . .                               | 10-4  |
| Určení posunu pro další bloky . . . . .                                                              | 10-4  |
| Použití posunů při vytváření více bloků . . . . .                                                    | 10-4  |
| Přidělení prostorů předpřipravených souborů v systému UNIX . . . . .                                 | 10-4  |
| Přidělení diskového prostoru s přímým přístupem v systému UNIX . . . . .                             | 10-5  |
| Vytvoření symbolických propojení k přímým zařízením (UNIX) . . . . .                                 | 10-5  |
| Přidělení prostoru souborům systému NTFS v systému Windows . . . . .                                 | 10-6  |
| Přidělení diskového prostoru s přímým přístupem v systému Windows . . . . .                          | 10-6  |
| Určení názvů paměťových prostorů a bloků . . . . .                                                   | 10-7  |
| Určení maximální velikosti bloků . . . . .                                                           | 10-8  |
| Určení maximálního počtu bloků a paměťových prostorů . . . . .                                       | 10-8  |
| Zálohování po změně fyzického schématu . . . . .                                                     | 10-8  |
| Správa prostorů dbSPACE . . . . .                                                                    | 10-9  |
| Vytvoření prostoru dbSPACE, který používá výchozí velikost stránky . . . . .                         | 10-9  |
| Určení velikosti první oblasti a dalších oblastí prostoru tbspace <b>tbspace</b> . . . . .           | 10-10 |
| Vytvoření prostoru dbSPACE s jinou než výchozí velikostí stránky . . . . .                           | 10-12 |
| Vytvoření společné oblasti vyrovnávacích pamětí pro jinou než výchozí velikost stránky . . . . .     | 10-13 |
| Definování velikosti stránky . . . . .                                                               | 10-16 |
| Zvýšení výkonu prostorů dbSPACE pro předpřipravené soubory pomocí přímého vstupu - výstupu . . . . . | 10-16 |
| Vytvoření dočasněho prostoru dbSPACE . . . . .                                                       | 10-17 |
| Co dělat, pokud dojde k vyčerpání diskového prostoru . . . . .                                       | 10-18 |
| Přidání bloku do prostoru dbSPACE nebo blobSPACE . . . . .                                           | 10-18 |
| Přidání bloku pomocí programu ON-Monitor (systém UNIX) . . . . .                                     | 10-19 |
| Přejmenování prostorů dbSPACE . . . . .                                                              | 10-19 |
| Další akce, které může být zapotřebí provést po přejmenování prostoru dbSPACE . . . . .              | 10-20 |
| Správa oddílů prostorů dbSPACE . . . . .                                                             | 10-20 |
| Správa prostorů blobSPACE . . . . .                                                                  | 10-21 |
| Vytvoření prostoru blobSPACE . . . . .                                                               | 10-21 |
| Příprava prostorů blobSPACE k ukládání dat typu TEXT a BYTE . . . . .                                | 10-23 |
| Určení velikosti stránky blobPAGE . . . . .                                                          | 10-23 |
| Určení velikosti stránky databázového serveru . . . . .                                              | 10-23 |
| Získání statistických údajů o paměti prostoru blobSPACE . . . . .                                    | 10-23 |
| Správa prostorů sbSPACE . . . . .                                                                    | 10-24 |
| Vytvoření prostoru sbSPACE . . . . .                                                                 | 10-24 |
| Určení velikosti metadat prostoru sbSPACE . . . . .                                                  | 10-25 |
| Přidání bloku do prostoru sbSPACE . . . . .                                                          | 10-25 |
| Změna paměťových charakteristik inteligentních velkých objektů . . . . .                             | 10-26 |
| Vytvoření dočasněho prostoru dbSPACE . . . . .                                                       | 10-26 |
| Vypuštění bloku . . . . .                                                                            | 10-27 |
| Ověření, zda je blok prázdný . . . . .                                                               | 10-27 |
| Vypuštění bloku z prostoru dbSPACE pomocí obslužného prostoru onspaces . . . . .                     | 10-28 |
| Vypuštění bloku z prostoru blobSPACE . . . . .                                                       | 10-28 |
| Vypuštění bloku z prostoru sbSPACE příkazem onspaces . . . . .                                       | 10-28 |
| Použití možnosti -f (force) . . . . .                                                                | 10-28 |
| Odstranění inteligentních velkých objektů, na které neukazují žádné ukazatele . . . . .              | 10-28 |
| Vypuštění paměťového prostoru . . . . .                                                              | 10-29 |
| Příprava na vypuštění paměťového prostoru . . . . .                                                  | 10-29 |
| Vypuštění zrcadlového paměťového prostoru . . . . .                                                  | 10-29 |
| Vypuštění paměťového prostoru s obslužným programem onspaces . . . . .                               | 10-29 |
| Vypuštění prostorů dbSPACE a blobSPACE pomocí programu ON-Monitor (UNIX) . . . . .                   | 10-30 |
| Zálohování po vypuštění paměťového prostoru . . . . .                                                | 10-30 |
| Správa prostorů extSPACE . . . . .                                                                   | 10-30 |

|                                                                                                   |       |
|---------------------------------------------------------------------------------------------------|-------|
| Vytvoření prostoru extspace . . . . .                                                             | 10-30 |
| Vypuštění prostoru extspace . . . . .                                                             | 10-31 |
| Přeskakování nepřístupných fragmentů . . . . .                                                    | 10-31 |
| Použití konfiguračního parametru DATASKIP . . . . .                                               | 10-31 |
| Použití funkce Dataskip obslužného programu onspaces . . . . .                                    | 10-32 |
| Použití obslužného programu onstat ke kontrole stavu funkce dataskip . . . . .                    | 10-32 |
| Použití příkazu SET DATASKIP jazyka SQL. . . . .                                                  | 10-32 |
| Vliv funkce dataskip na transakce . . . . .                                                       | 10-32 |
| Jak určit, kdy používat parametr dataskip . . . . .                                               | 10-33 |
| Určení, kdy přeskakovat vybrané fragmenty . . . . .                                               | 10-33 |
| Určení, kdy přeskočit všechny fragmenty . . . . .                                                 | 10-33 |
| Monitorování využití fragmentace . . . . .                                                        | 10-33 |
| Zobrazení databází . . . . .                                                                      | 10-34 |
| Použití tabulek SMI . . . . .                                                                     | 10-34 |
| Použití programu ISA. . . . .                                                                     | 10-34 |
| Použití programu ON-Monitor (systém UNIX) . . . . .                                               | 10-34 |
| Monitorování využití disku . . . . .                                                              | 10-34 |
| Monitorování bloků . . . . .                                                                      | 10-34 |
| onstat -d. . . . .                                                                                | 10-35 |
| Příkaz onstat -d update . . . . .                                                                 | 10-35 |
| Příkaz onstat -D . . . . .                                                                        | 10-35 |
| Příkaz onstat -g iof . . . . .                                                                    | 10-35 |
| oncheck -pr. . . . .                                                                              | 10-36 |
| oncheck -pe. . . . .                                                                              | 10-36 |
| Použití programu IBM Informix Server Administrator . . . . .                                      | 10-37 |
| Použití programu ON-Monitor (systém UNIX) . . . . .                                               | 10-37 |
| Použití tabulek SMI . . . . .                                                                     | 10-38 |
| Monitorování prostorů tblspace a oblastí . . . . .                                                | 10-38 |
| Použití tabulek SMI . . . . .                                                                     | 10-38 |
| Monitorování jednoduchých velkých objektů v prostoru blobspace . . . . .                          | 10-38 |
| Příkaz onstat -O . . . . .                                                                        | 10-39 |
| Určení zaplnění stránek blobpage pomocí příkazu oncheck -pB . . . . .                             | 10-40 |
| Monitorování využití prostoru blobspace pomocí příkazu oncheck -pe . . . . .                      | 10-40 |
| Monitorování jednoduchých velkých objektů v prostoru dbspace pomocí příkazu oncheck -pT . . . . . | 10-40 |
| Monitorování prostorů sbspace . . . . .                                                           | 10-41 |
| Použití příkazu onstat -d . . . . .                                                               | 10-42 |
| Použití příkazů oncheck -ce a oncheck -pe . . . . .                                               | 10-43 |
| Použití příkazu oncheck -cs . . . . .                                                             | 10-44 |
| Použití příkazu oncheck -ps . . . . .                                                             | 10-44 |
| Monitorování oblastí metadat a uživatelských dat . . . . .                                        | 10-45 |
| Použití příkazu onstat -g smb c . . . . .                                                         | 10-45 |
| Zavedení dat do tabulky . . . . .                                                                 | 10-46 |

---

## Obsah kapitoly

Tato kapitola obsahuje pokyny pro efektivní správu diskových prostorů a dat, které řídí databázový server. Předpokládá se, že jste obeznámeni s termíny a pojmy, které obsahuje Kapitola 9, "Uložení dat", na stránce 9-1.

Ke správě paměťových prostorů můžete používat následující obslužné programy:

- **onspaces**
- ISA

V této kapitole jsou popsána následující témata:

- přidělení diskového prostoru
- Určení názvů bloků, maximální velikost a počet bloků a paměťových prostorů.
- Zálohování po změně fyzického schématu.
- Vytváření a vypouštění prostorů dbspace, blobspace a sbspace.

- Přejmenování prostorů dbspace.
- Správa oddílů prostoru dbspace.
- Přidávání bloků do prostorů dbspace, blobspace a sbspace a vypouštění bloků z těchto prostorů.
- Vytváření a vypouštění prostorů extspace.
- Přeskakování nepřístupných fragmentů.
- Monitorování využití diskových prostorů.
- Monitorování dat jednoduchých velkých objektů a inteligentních velkých objektů.

Příručka *Příručka výkonnosti serveru IBM Informix Dynamic Server* obsahuje také informace o správě diskových prostorů. Popisuje zejména, jak lze eliminovat prokládané oblasti, uvolnit prostor v prázdné oblasti a zlepšit diskový vstup - výstup.

Další informace o použití administrativních příkazů SQL API namísto některých příkazů obslužného programu **onspaces** naleznete v části “Vzdálená správa pomocí příkazů SQL” na stránce 26-10 and the *IBM Informix Guide to SQL: Syntax*.

---

## Přidělení diskového prostoru

Tato část objasňuje, jak přidělit diskový prostor pro databázový server. Před přidělením diskového prostoru prostudujte následující části:

- “Přístup k disku s vyrovnávací pamětí a bez vyrovnávací paměti v systému UNIX” na stránce 9-4
- “Objem diskového prostoru potřebný k uložení dat” na stránce 9-31
- “Pravidla pro rozvržení disků” na stránce 9-33

Před vytvořením paměťového prostoru nebo bloku nebo před zrcadlením existujícího paměťového prostoru musíte souboru bloku přidělit diskový prostor. Pro diskový prostor databázového serveru můžete přidělit buď prázdný soubor, nebo část disku s přímým přístupem.

---

### Jen pro UNIX

Přidělíte-li diskový prostor s přímým přístupem v systému UNIX, měli byste použít příkaz UNIX **ln** a vytvořit tak propojení mezi názvem zařízení obsahující speciální znaky a dalším názvem souboru. Další informace o tomto tématu naleznete v části “Vytvoření symbolických propojení k přímým zařízením (UNIX)” na stránce 10-5.

Použití souboru systému UNIX a základního rozhraní operačního systému pro diskový prostor databázového serveru je také označováno jako použití *předpřipraveného prostoru*.

---

### Konec Jen pro UNIX

---

### Jen pro Windows

V systému Windows se pro diskový prostor databázového serveru doporučuje používat soubory systému souborů NTFS. Další informace o tomto doporučení naleznete v části “Přístup k disku s vyrovnávací pamětí a bez vyrovnávací paměti v systému UNIX” na stránce 9-4.

---

### Konec Jen pro Windows

Bloky můžete rozložit mezi disky a řadiče. Umístění více bloků na jeden disk může zvýšit propustnost.

## Určení posunu

Když databázovému serveru přidělujete blok diskového prostoru, můžete určit jeho posun. Určení posunu může sloužit k následujícím účelům:

- Zabránění přepsání informací o oddílech databázovým serverem.
- Definice více bloků v oddílu, diskovém zařízení nebo předpřipraveném souboru.

Maximální hodnota posunu je 4 TB.

Mnoho počítačových systémů i někteří výrobci disků uchovávají informace o fyzické diskové jednotce v jednotce samotné. Tyto informace jsou někdy označovány jako *tabulka obsahu disku* (VTOC) nebo jmenovka disku. Tabulka VTOC je obvykle uložena na první stopě jednotky. Na první stopě může být také uložena tabulka náhradních sektorů a mapování vadných sektorů (nazývaná také *tabulka přesměrování*).

Plánujete-li přidělení oddílů na začátku disku, budete pravděpodobně potřebovat použít posuny, které databázovému serveru zabrání v přepsání důležitých informací požadovaných systémem. Přesné informace o požadovaném posunu naleznete v příručkách diskových jednotek.

**Upozornění:** Pokud jsou spuštěny dvě nebo více instancí databázového serveru, věnujte zvýšenou pozornost definicím bloků, které se překrývají. Překrývající se bloky mohou způsobit, že databázový server přepíše data v jednom bloku nesouvisejícími daty druhého bloku. Toto přepsání v podstatě zničí překrývající se data.

### Určení posunu pro počáteční blok kořenového prostoru dbspace

Posuny pro počáteční blok kořenového prostoru dbspace a jeho zrcadla, pokud existuje, určete pomocí parametrů ROOTOFFSET, resp. MIRROROFFSET. Další informace najdete v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.

### Určení posunu pro další bloky

Chcete-li určit posun pro další bloky prostoru databázového serveru, musíte zadat posun jako parametr, když prostor přiřazujete k databázovému serveru. Další informace naleznete v části "Vytvoření prostoru dbspace, který používá výchozí velikost stránky" na stránce 10-9.

### Použití posunů při vytváření více bloků

Více bloků můžete v diskových oddílech, diskových zařízeních nebo souborech vytvářet určením posunů a přiřazením bloků, které jsou menší než celkový dostupný prostor. Posun určuje umístění počátku bloku. Databázový server určuje umístění posledního bajtu bloku sečtením velikosti bloku a posunu.

Přiřaďte počáteční posun pro první blok, je-li potřebný, a zadejte velikost, která musí být menší než celková velikost přiděleného diskového prostoru. Pro každý další blok zadejte posun tak, aby zahrnoval velikosti všech předchozích přidělených bloků a počáteční posun, a přiřaďte velikost, která je menší nebo rovna než velikost zbývajících přidělovaného prostoru.

## Přidělení prostorů předpřipravených souborů v systému UNIX

Následující postup znázorňuje příklad přidělení místa na disku pro předpřipravený soubor `usr/data/my_chunk` v systému UNIX.

### Přidělení prostoru předpřipraveného souboru:

1. Přihlašte se jako uživatel `informix`:  
`su informix`
2. Přejděte do adresáře, ve kterém bude uložen předpřipravený prostor:

- ```
cd /usr/data
```
- Vytvořte blok zapsáním souboru null pomocí příkazu cat do souboru, který bude databázový server používat pro diskový prostor:


```
cat /dev/null > my_chunk
```
 - Nastavte oprávnění souboru na hodnotu 660 (rw-rw----):


```
chmod 660 my_chunk
```
 - Musíte nastavit skupinu i vlastníka souboru na uživatele **informix**:


```
ls -l my_chunk -rw-rw----
  1 informix informix
  0 Oct 12 13:43 my_chunk
```
 - Vytvořte paměťový prostor nebo blok pomocí obslužného programu **onspaces**.

Informace o způsobu vytvoření paměťového prostoru pomocí přiděleného souboru naleznete v částech “Vytvoření prostoru dbspace, který používá výchozí velikost stránky” na stránce 10-9, “Vytvoření prostoru blobspace” na stránce 10-21 a “Vytvoření prostoru sbspace” na stránce 10-24.

Přidělení diskového prostoru s přímým přístupem v systému UNIX

Chcete-li přidělit prostor s přímým přístupem, musíte mít dostupný oddíl, který je vyhrazen pro prostor s přímým přístupem. Diskový prostor s přímým přístupem lze vytvořit buď opětovným vytvořením oddílů na discích nebo odpojením existujícího systému souborů. Před odpojením zařízení zálohujte všechny soubory.

Přidělení diskového prostoru s přímým přístupem:

- Vytvořte a nainstalujte přímé zařízení.

Specifické pokyny týkající se přidělení diskového prostoru s přímým přístupem v systému UNIX naleznete v dokumentaci k operačnímu systému a v části “Přístup k disku s vyrovnávací pamětí a bez vyrovnávací paměti v systému UNIX” na stránce 9-4.
- Změňte vlastnictví a oprávnění znakově orientovaného zařízení na uživatele **informix**.

Název souboru tohoto zařízení obvykle začíná písmenem *r*. Podrobný postup naleznete v částech 4 a 5 v kapitole “Přidělení prostorů předpřipravených souborů v systému UNIX” na stránce 10-4.
- Ověřte, že oprávnění znakově orientovaných zařízení operačního systému je nastaveno na hodnotu **crw-rw----**.
- Vytvořte symbolické propojení mezi názvem znakově orientovaného zařízení a názvem jiného souboru pomocí příkazu **ln -s** systému UNIX. Další informace naleznete v části “Vytvoření symbolických propojení k přímým zařízením (UNIX)” na stránce 10-5.

Upozornění: Po vytvoření přímého zařízení, které databázový server používá pro diskový prostor, nevytvářejte systémy souborů v přímém zařízení, které je přiděleno pro diskový prostor databázového serveru. Nepoužívejte toto přímé zařízení přidělené pro diskový prostor databázového serveru ani jako odkládací prostor.

Vytvoření symbolických propojení k přímým zařízením (UNIX)

Pomocí symbolických propojení lze přiřazovat standardní názvy zařízení a odkazovat na zařízení. Chcete-li vytvořit propojení mezi názvem znakově orientovaného zařízení a názvem jiného souboru, použijte příkaz **ln** systému UNIX (obvykle se nazývá **ln**). Provedením příkazu **ls -l** systému UNIX (**ls -lg** v systému BSD) ověřte, že jak zařízení, tak propojení existují v adresáři příslušného zařízení. `directory`. Následující příklad znázorňuje propojení k přímým zařízením. Nepodporuje-li operační systém symbolická propojení, lze použít pevná propojení.

```
ln -s /dev/rxy0h /dev/my_root # propojení původního zařízení
                                # se symbolickým názvem
ln -s /dev/rxy0a /dev/raw_dev2
ls -l
crw-rw--- /dev/rxy0h
crw-rw--- /dev/rxy0a
lrwxrwxrwx /dev/my_root@->/dev/rxy0h
lrwxrwxrwx /dev/raw_dev2@->/dev/rxy0a
```

Důvod používání symbolických odkazů je následující: Vytvoříte-li bloky v přímém zařízení a toto zařízení selže, nemůžete provést obnovení ze zálohy, dokud přímé zařízení nenahradíte a nepoužijete stejnou cestu. Všechny bloky, které byly přístupné v době posledního zálohování, musejí být přístupné i při obnovení.

Symbolická propojení usnadňují obnovu po selhání disku a umožňují rychlou výměnu disku, na kterém je blok umístěn. Můžete vadné zařízení nahradit jiným, propojit cestu k novému zařízení s názvem souboru, který jste předtím vytvořili pro vadné zařízení, a obnovit data. Nemusíte čekat na opravu původního zařízení.

Přidělení prostoru souborům systému NTFS v systému Windows

Databázový server v systému Windows používá ve výchozím nastavení soubory systému NTFS. Systém souborů NTFS umožňuje použití standardních názvů souborů pro soubory bez vyrovnávací paměti.

Prvním krokem při přidělování prostoru v souboru systému NTFS diskovému nebo zrcadlenému prostoru databázového serveru je vytvoření prázdného souboru (o velikosti 0 bajtů).

Přidělení prostoru souborů systému NTFS prostorům dbspace, blobspace nebo sbospace:

1. Přihlašte se jako člen skupiny **Informix-Admin**.
2. Otevřete prostředí příkazů systému MS-DOS.
3. Přejděte do adresáře, ve kterém chcete prostor přidělit, jako v následujícím příkladu:
c:> cd \usr\data
4. Pomocí následujícího příkazu vytvořte prázdný soubor:
c:> copy nul muj_blok
5. Pomocí příkazu **dir** můžete ověřit, zda byl soubor vytvořen.

Po přidělení prostoru souborů můžete vytvořit prostor dbspace nebo jiný paměťový prostor, který byste obvykle použili, pomocí obslužného programu **onspaces**. Informace o vytváření prostorů dbspace a blobspace naleznete v části “Vytvoření prostoru dbspace, který používá výchozí velikost stránky” na stránce 10-9 a “Vytvoření prostoru blobspace” na stránce 10-21.

Přidělení diskového prostoru s přímým přístupem v systému Windows

Diskový prostor s přímým přístupem můžete v systému Windows konfigurovat jako logickou nebo fyzickou jednotku. Chcete-li vyhledat písmeno jednotky nebo číslo disku, spusíte obslužný program **Správce disků**. Je-li potřeba, aby byly jednotky prokládané (několik fyzických disků je sloučeno do jedné logické jednotky), pracovalo by pouze určování logické jednotky.

Vytváření paměťového prostoru a přidávání bloku vyžaduje, abyste byli členy skupiny **Informix-Admin**. Diskový prostor s přímým přístupem může být naformátovaný i nenaformátovaný diskový prostor.

Určení logické jednotky:

1. Přiřaďte písmeno jednotky k oddílu disku.
2. Zadejte následující hodnotu pro databázový prostor ROOTDBS v souboru ONCONFIG:
`\\.\písmeno_jednotky`
3. Chcete-li vytvořit paměťový prostor nebo přidat blok, zadejte oddíl logické jednotky.
V tomto příkladu je blok velikosti 5000 kilobajtů na jednotce **e**: přidán s posunem 5200 kilobajtů k prostoru dbspace **dpspc3**.
`onspaces -a dbspc3 \\.\e: -o 5200 -s 5000`

Určení logické jednotky:

1. Pokud k oddílu disku *nebylo* přiřazeno písmeno jednotky, zadejte následující hodnotu pro databázový prostor ROOTDBS v souboru ONCONFIG:
`\\.\PhysicalDrive<číslo>`
2. Chcete-li vytvořit paměťový prostor nebo přidat blok, zadejte oddíl fyzické jednotky.
V tomto příkladu je blok velikosti 5000 kilobajtů na jednotce **PhysicalDrive0**, přidán s posunem 5200 kilobajtů k prostoru dbspace **dpspc3**.
`onspaces -a dbspc3 \\.\PhysicalDrive0 : -o 5200 -s 5000`

Upozornění: Pokud přidělíte naformátovanou jednotku nebo část disku jako diskový prostor s přímým přístupem a jednotka nebo část disku obsahuje data, přepíše databázový server tato data, jakmile začne používat diskový prostor. Před přidělením diskového prostoru databázovému serveru musíte zkontrolovat, zda jsou veškerá data v diskovém prostoru s přímým přístupem postradatelná.

Určení názvů paměťových prostorů a bloků

Název bloku musí splňovat stejná kritéria jako název paměťového prostoru. Určete explicitní cestu k paměťovému prostoru nebo bloku. Postupujte takto:

Jen pro UNIX

- Pokud používáte disky s přímým přístupem v systému UNIX, doporučujeme použít propojenou cestu. Další informace naleznete v části “Vytvoření symbolických propojení k přímým zařízením (UNIX)” na stránce 10-5.)

Konec Jen pro UNIX

Jen pro Windows

- Používáte-li disky s přímým přístupem v systému Windows, má cesta následující tvar, kde *x* určuje diskovou jednotku nebo oddíl:
`\\.\x:`

Konec Jen pro Windows

- Používáte-li pro diskový prostor databázového serveru soubor, skládá se cesta z úplné cesty a názvu souboru.

Při vytváření paměťových prostorů a přidávání bloků používejte tato pravidla pro názvy. Název souboru musí mít následující charakteristiky:

- Musí být jedinečný a nesmí překročit 128 bajtů
- Musí začínat písmenem nebo podtržítkem.
- Smí obsahovat pouze písmena, číslice, podtržítka a znaky \$.

V názvu se nerozlišují velká a malá písmena, pokud není uzavřen v uvozovkách. Ve výchozím nastavení převádí databázový server velká písmena v názvu na malá. Chcete-li v názvech používat velká písmena, uzavřete názvy do uvozovek a nastavte proměnnou prostředí **DELIMIDENT** na hodnotu **ON**.

Určení maximální velikosti bloků

Na většině platform je maximální velikost bloků 4 TB, ale na některých platformách může být 8 terabajtů. Chcete-li zjistit, jakou velikost bloků podporuje používaná platforma, prostudujte soubor poznámek k počítači. Nespustíte-li příkaz **onmode -BC**, bude maximální velikost bloků 2 GB.

Určení maximálního počtu bloků a paměťových prostorů

Můžete určit až 32 766 bloků pro jeden paměťový prostor a až 32 766 paměťových prostorů v systému databázového serveru. Paměťové prostory mohou být libovolnou kombinací prostorů **dbspace**, **blobspace** a **sbspace**.

Uvažujeme-li všechna omezení, která lze použít na velikost instance databázového serveru, je maximální velikost instance přibližně 8 petabajtů.

Chcete-li povolit maximální počet bloků a paměťových prostorů, musíte spustit příkaz **onmode -BC**.

Zálohování po změně fyzického schématu

Chcete-li při provádění následujících úloh zajistit, aby bylo možné provést obnovu dat, musíte provést zálohování úrovně 0 kořenového prostoru **dbspace** a změněných paměťových prostorů:

- přidání a vypuštění zrcadlení,
- vypuštění souboru logického protokolu,
- změna velikosti nebo umístění fyzického protokolu,
- změna konfigurace správce paměti,
- přidání, přesun a vypuštění prostorů **dbspace**, **blobspace** a **sbspace**,
- přidání a přesun bloku do prostorů **dbspace**, **blobspace** a **sbspace** a vypuštění bloku z těchto prostorů.

Důležité: Přidáte-li nový logický protokol, nemusíte již provádět zálohování úrovně 0 kořenového prostoru **dbspace** a změněného prostoru **dbspace**, chcete-li nový logický protokol *používat*. Doporučuje se však, abyste zálohování úrovně 0 provedli a zabránili tak selhání zálohování úrovně 1 a 2.

Zálohování úrovně 0 změněných paměťových prostorů musíte provést, aby byla zajištěna možnost obnovy neprotokolovaných dat, než přepnete jiný na typ tabulky s protokolováním:

- Převádíte-li databázi bez protokolování na databázi s protokolováním.
- Převádíte-li tabulku typu RAW na standardní tabulku.

Správa prostorů dbspace

Tato část obsahuje informace o vytváření standardních a dočasných prostorů dbspace s výchozí i jinou velikostí stránky, o určování velikosti první oblasti a dalších oblastí prostoru tbspace **tbspace** v prostoru dbspace při vytváření prostoru dbspace a o přidávání bloku do prostorů dbspace a blobspace.

Vytvoření prostoru dbspace, který používá výchozí velikost stránky

V této části je objasněno použití obslužného programu **onspaces** při vytváření standardního a dočasného prostoru dbspace. Informace o vytváření prostoru dbspace s jinou než výchozí velikostí stránky naleznete v části “Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky” na stránce 10-12.

Informace o vytváření prostoru dbspace pomocí programu ISA naleznete v nápovědě online programu ISA.

Libovolný nově přidaný prostor dbspace a jeho zrcadlo, pokud existuje, jsou okamžitě dostupné. Používáte-li zrcadlení, můžete prostor dbspace zrcadlit při vytvoření. Zrcadlení se projeví okamžitě po vytvoření.

Vytvoření standardního prostoru dbspace pomocí obslužného programu onspaces:

1. Chcete-li vytvořit prostor dbspace v systému UNIX, musíte být přihlášení jako uživatel **informix** nebo **root**.

V systému Windows mohou vytvořit prostor dbspace uživatelé ve skupině **Informix-Admin**.

2. Zajistěte, aby byl databázový server v režimu online, v režimu administrace nebo v klidovém režimu.

3. Přidělte diskový prostor pro prostor dbspace podle popisu v části “Přidělení diskového prostoru” na stránce 10-3.

4. Při vytváření prostoru dbspace použijte volby obslužného programu **onspaces -c -d**.

Výchozí jednotkou pro volby velikost **-s velikost** a **-o posun** je kilobajt (kB). Megabajty převedete na kilobajty vynásobením činitelem 1024 (například 10 MB = 10 * 1024 kB).

Informace o dalších volbách obslužného programu **onspaces** naleznete v části “Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky” na stránce 10-12, vytváříte-li prostor dbspace s jinou než výchozí velikostí stránky.

5. Nechcete-li určit velikost první oblasti pro rozšíření a dalších oblastí pro prostor tbspace **tbspace** v prostoru dbspace, přejděte ke kroku 6.

Pokud chcete určit velikost první oblasti a dalších oblastí pro prostor tbspace **tbspace** v prostoru dbspace, naleznete další informace v části “Určení velikosti první oblasti a dalších oblastí prostoru tbspace **tbspace**” na stránce 10-10.

6. Po vytvoření prostoru dbspace musíte provést zálohování úrovně 0 kořenového prostoru dbspace a nového prostoru dbspace.

Následující příklad znázorňuje vytvoření 10megabajtového zrcadleného prostoru dbspace **dbspc1** s posunem 5000 kB pro primární i zrcadlený blok pomocí diskového prostoru s přímým přístupem v systému UNIX:

```
onspaces -c -d dbspc1 -p /dev/raw_dev1 -o 5000 -s 10240 -m /dev/raw_dev2 5000
```

Následující příklad znázorňuje vytvoření 5megabajtového prostoru dbspace **dbspc3** s posunem 200 kB z diskového prostoru s přímým přístupem (jednotka **e:**) v systému Windows:

```
onspaces -c -d dbspc3 \\.\e: -o 200 -s 5120
```

Další informace o vytváření prostoru dbspace pomocí obslužného programu **onspaces** naleznete v části “Prostory dbspace” na stránce 9-9. Informace o obslužném programu **onspaces** naleznete v příručce *IBM Informix Administrator's Reference*.

Vytvoření prostoru dbspace pomocí programu ON-Monitor (UNIX):

1. Vyberte volbu **Dbspaces > Create**.
2. Zadejte název nového prostoru dbspace do pole **Dbpace Name**.
3. Chcete-li vytvořit zrcadlo pro počáteční blok prostoru dbspace, zadejte do pole **Mirror** hodnotu **Y**.
V opačném případě zadejte hodnotu **N**.
4. Je-li prostor dbspace, který vytváříte, dočasný, zadejte do pole **Temp** hodnotu **Y**.
V opačném případě zadejte hodnotu **N**.
5. Pokud určujete velikost stránky pro standardní prostor dbspace, zadejte tuto velikost v kilobajtech do pole **Page Size**. Tato velikost musí být násobkem velikosti stránky kořenového prostoru dbspace. Další informace o určování velikostí stránky naleznete v části “Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky” na stránce 10-12.
Všechny tabulky, indexy a další objekty v prostoru dbspace budou používat stránky určené velikosti.
6. Zadejte úplnou cestu k počátečnímu primárnímu bloku prostoru dbspace do pole **Full Pathname** v části primárního bloku.
7. Do pole **Offset** zadejte posun.
8. Do pole **Size** zadejte velikost bloku v kilobajtech.
9. Pokud tento prostor dbspace zrcadlíte, zadejte úplný název cesty, velikost a nepovinný posun zrcadleného bloku v části obrazovky se zrcadleným blokem.

Další informace naleznete v kapitole ON-Monitor v příručce *IBM Informix Administrator's Reference*.

Určení velikosti první oblasti a dalších oblastí prostoru tblspace tblspace

Zadáním velikosti první oblasti a dalších oblastí můžete snížit počet oblastí prostoru tblspace **tblspace** a četnost situací, ve kterých je zapotřebí umístit oblasti prostoru tblspace **tblspace** do jiných než primárních bloků. (*Primární blok* je počáteční blok prostoru dbspace.)

Můžete zvolit, zda chcete určit velikost první oblasti, velikost dalších oblastí, velikost první oblasti i dalších oblastí, nebo zda nechcete určit žádnou velikost oblasti. Pokud neurčíte velikost první oblasti a dalších oblastí pro prostor tblspace **tblspace**, použije dynamický server existující výchozí velikost oblasti.

Velikost první oblasti a dalších oblastí pro prostor tblspace **tblspace** v kořenovém prostoru dbspace, který je vytvořen při inicializaci serveru, můžete určit pomocí konfiguračních parametrů TBLTBLFIRST a TBLTBLNEXT.

Velikost první oblasti a dalších oblastí pro prostor tblspace **tblspace** v jiných prostorech než kořenových můžete určit pomocí obslužného programu **onspaces**.

Velikost první oblasti a dalších oblastí můžete určit pouze při vytváření prostoru dbspace. Po vytvoření prostoru dbspace již tyto velikosti nelze měnit. Navíc nelze určit velikosti oblastí pro dočasné prostory dbspace, sbspace, blobspace a externí prostory. Po vytvoření prostoru dbpace již nelze velikost první a dalších oblastí měnit.

Určení velikosti první oblasti a dalších oblastí:

1. Určete celkový počet stránek potřebných v prostoru **tblspace**. Počet stránek je roven součtu počtu tabulek, odpojených indexů a fragmentů tabulek, které budou pravděpodobně uloženy v prostoru **dbspace**, plus jedna stránka pro prostor **tblspace**.
2. Vypočtete počet kilobajtů potřebný pro počet stránek. Tento počet závisí na počtu kilobajtů připadajícím na stránku v systému.
3. Zjistíte potřeby správy prostorů v systému tím, že zvážíte, jak je důležité, aby všechny oblasti pro prostor **tblspace** byly přiděleny během vytváření prostoru **dbspace** a aby tyto oblasti byly přiděleny souvisle. Čím důležitější jsou tyto požadavky, tím větší by měla být velikost první oblasti. Pokud vás příliš nezajímá, zda budou oblasti souvislé, možná v sekundárních blocích, pak velikost první oblasti a dalších oblastí může být menší.
4. Při určování velikosti oblasti postupujte takto:
 - Jedná-li se o požadavek na prostor v kořenovém prostoru **dbspace**, určete velikost první oblasti v konfiguračním parametru **TBLTBLFIRST** a velikost dalších oblastí v konfiguračním parametru **TBLTBLNEXT**. Potom inicializujte instanci databázového serveru.
 - Jedná-li se o požadavek na prostor v jiném prostoru **dbspace** než kořenovém, zadejte velikost první oblasti a dalších oblastí v příkazovém řádku a vytvořte prostor **dbspace** pomocí obslužného programu **onspaces**.

Velikosti oblastí musejí být v kilobajtech a musejí být násobkem velikosti stránky. Při určování velikosti první oblasti a dalších oblastí dodržujte následující pokyny:

Typ oblasti	Minimální velikost	Maximální velikost
První oblast v nekořenovém prostoru dbspace	Ekvivalent 50 stránek v kilobajtech. Jedná se o výchozí hodnotu systému. Například: Minimální délka pro systém se stránkami velikosti 2 kB je 100.	Velikost počátečního bloku minus prostor potřebný pro libovolné systémové objekty, jako jsou například rezervované stránky, databázový prostor tblspace a fyzický a logický protokol.
První oblast v kořenovém prostoru dbspace	Ekvivalent 250 stránek v kilobajtech. Jedná se o výchozí hodnotu systému.	Velikost počátečního bloku minus prostor potřebný pro libovolné systémové objekty, jako jsou například rezervované stránky, databázový prostor tblspace a fyzický a logický protokol.
Nová oblast	Čtyřnásobek velikosti stránky na disku v systému. Výchozí hodnotou je 50 stránek v libovolném typu prostoru dbspace .	Maximální velikost bloku minus tři stránky.

Velikost první oblasti a dalších oblastí pro prostor **tblspace** v nekořenových prostorech **dbspace** můžete určit pomocí voleb **-ef** a **-en** obslužného programu **onspaces**:

- Velikost první oblasti: **-ef velikost_v_kB**
- Velikost dalších oblastí: **-en velikost_v_kB**

Můžete například zadat:

```
onspaces -c -d dbspace1 -p /usr/data/dbspace1 -o 0 -s 1000000 -e 2000 -n 1000
```

Chcete-li zobrazit velikost první oblasti a dalších oblastí prostoru tblspace **tblspace**, můžete použít příkaz **oncheck -pt** a **oncheck -pT**.

Je-li použita replikace data a na primárním databázovém serveru je vytvořen prostor dbspace, je velikost první oblasti a dalších oblastí předána sekundárnímu databázovému serveru prostřednictvím záznamu protokolu ADDCHK.

Další informace o obslužném programu **onspaces**, příkazech **oncheck** a určení velikosti první oblasti a dalších oblastí pro prostor tblspace **tblspace** naleznete v příručce *IBM Informix Administrator's Reference*.

Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky

Pokud chcete větší délku klíče, než jaká je k dispozici pro výchozí velikost stránky, můžete zadat velikost stránky pro standardní nebo dočasný prostor dbspace. Kořenový prostor dbspace sestává ze stránek výchozí velikosti. Pokud chcete určit velikost stránky, musíte zadat celočíselný násobek výchozí velikosti stránky, který nesmí být větší než 16 kB.

U systémů s dostatečně velkým úložištěm se větší velikost stránky vyznačuje následujícími výhodami:

- Snížená hloubka indexů B-stromu i u menších rejstříkových klíčů.
- Snížená doba kontrolního bodu, ke které obvykle dochází u větších velikostí stránek.

Dochází také k dalším zlepšením výkonu, protože můžete:

- Seskupit stejně dlouhé řádky stránky, které aktuálně zahrnují několik stránek velikosti výchozí stránky.
- Definovat odlišnou velikost stránky pro dočasné tabulky, které tak budou mít oddělenou oblast vyrovnávací paměti.

Můžete také pomocí konfiguračního parametru BUFFERPOOL vytvořit společnou oblast vyrovnávací paměti, která odpovídá velikosti stránky prostoru dbspace. (Je možné, že touto akcí budete chtít implementovat formulář "vlastní oblasti pro vyrovnávací paměť.")

Tabulka může být v jednom prostoru dbspace a index pro tuto tabulku může být v jiném prostoru dbspace. Tyto oddíly mohou mít různou velikost stránky.

Chcete-li určit velikost stránky pro prostor dbspace, proveďte tyto úlohy:

1. Není-li povolen režim velkých bloků, povolte jej pomocí příkazu **onmode -BC**. Podle výchozího nastavení se při první inicializaci nebo restartu serveru Dynamic Server 10.0 spustí server Dynamic Server s povoleným režimem velkých bloků. Informace o obslužném programu **onmode** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.
2. Vytvořte společnou oblast vyrovnávacích pamětí, která odpovídá velikosti stránky prostoru dbspace. Můžete použít obslužný program **onparams** nebo konfigurační parametr BUFFERPOOL. Doporučujeme tuto úlohu provést před vytvořením prostoru dbspace.

Vytvoříte-li prostor dbspace s velikostí stránky, která neodpovídá společné oblasti vyrovnávacích pamětí, server Dynamic Server automaticky vytvoří společnou oblast vyrovnávacích pamětí pomocí výchozího parametru definovaného v konfiguračním souboru ONCONFIG.

Nemůžete mít několik společných oblastí vyrovnávacích pamětí se stejnou velikostí stránky.

Další informace naleznete v příručce “Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky” na stránce 10-12.

3. Velikost stránky prostoru dbspace definujte při vytváření prostoru dbspace. Můžete použít obslužný program **onspaces** nebo program ON-Monitor. Další informace naleznete v části “Definování velikosti stránky” na stránce 10-16.

Pokud například vytvoříte prostor dbspace s velikostí stránky 6 kB, je třeba vytvořit také společnou oblast vyrovnávacích pamětí s velikostí stránky 6 kB. Neurčíte-li velikost stránky pro novou společnou oblast vyrovnávacích pamětí, použije server Dynamic Server výchozí velikost stránky operačního systému (4 kB v systému Windows a 2 kB na většině platformem systému UNIX) jako výchozí velikost stránky pro společnou oblast vyrovnávacích pamětí.

Poznámka: Používáte-li jiné než výchozí velikosti stránky, budete pravděpodobně potřebovat zvýšit velikost fyzického protokolu. Provádíte-li mnoho aktualizací stránek jiné velikosti než výchozí, je možné, že budete potřebovat zvýšení velikosti fyzického protokolu o 150 až 200 procent. Vyladění velikosti fyzického protokolu může vyžadovat několik pokusů. Velikost fyzického protokolu můžete nastavit podle potřeby v závislosti na tom, jak často zaplnění fyzického protokolu spouští kontrolní body.

Vytvoření společné oblasti vyrovnávacích pamětí pro jinou než výchozí velikost stránky

Po vytvoření společné oblasti vyrovnávacích pamětí můžete pomocí konfiguračního parametru BUFFERPOOL nebo obslužného programu **onparams** definovat informace o společné oblasti vyrovnávacích pamětí, včetně její velikosti, počtu front LRU ve společné oblasti vyrovnávacích pamětí, počet vyrovnávacích pamětí ve společné oblasti vyrovnávacích pamětí a hodnoty **lru_min_dirty** a **lru_max_dirty**.

Tyto informace lze zadat pomocí konfiguračního parametru BUFFERPOOL.

Konfigurační parametr BUFFERPOOL se skládá ze dvou řádků v souboru **onconfig.std**.

V počítačích se systémem UNIX vypadají řádky takto:

```
BUFFERPOOL default,lrus=8,buffers=5000,lru_min_dirty=50,lru_max_dirty=60
BUFFERPOOL size=2K,buffers=5000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
```

V počítačích se systémem Windows vypadají řádky takto:

```
BUFFERPOOL default,lrus=8,buffers=2000,lru_min_dirty=50,lru_max_dirty=60
BUFFERPOOL size=4K,buffers=2000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
```

První řádek určuje výchozí hodnoty, které budou použity v případě, že vytvoříte prostor dbspace s velikostí stránky, která nemá odpovídající společnou oblast vyrovnávacích pamětí vytvořenou při spuštění databázového serveru. Druhý řádek určuje výchozí hodnoty databázového serveru pro společnou oblast vyrovnávacích pamětí. Tyto hodnoty jsou založeny na výchozí velikosti stránky databázového serveru.

Přidáte-li prostor dbspace s jinou velikostí stránky pomocí obslužného programu **onspaces** nebo přidáte-li novou společnou oblast vyrovnávacích pamětí pomocí obslužného programu **onparams**, bude ke konfiguračnímu parametru BUFFERPOOL v souboru ONCONFIG přidán nový řádek. Velikost stránky každí společné oblasti vyrovnávacích pamětí musí být násobkem výchozí velikosti stránky pro operační systém. Následující příklad znázorňuje třetí řádek konfiguračního parametru BUFFERPOOL, který byl přidán do souboru ONCONFIG:

```
BUFFERPOOL default,lrus=8,buffers=5000,lru_min_dirty=50,lru_max_dirty=60
BUFFERPOOL size=2K,buffers=5000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
BUFFERPOOL size=6K,buffers=3000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
```

Pole v rádcích konfiguračního parametru BUFFERPOOL nerozlišují velká a malá písmena, můžete tedy zadat hodnotu **lrus**, **Lrus** nebo **LRUS**. Tato pole mohou být uvedena v libovolném pořadí.

Neurčíte-li velikost stránky pro novou společnou oblast vyrovnávacích pamětí, použije server Dynamic Server výchozí velikost stránky operačního systému (4 kB v systému Windows a 2 kB na většině platform systému UNIX) jako výchozí velikost stránky pro společnou oblast vyrovnávacích pamětí.

Je-li hodnota pole **buffers** v konfiguračním parametru BUFFERPOOL rovna nule (0) nebo pokud hodnota pole **buffers** chybí, nevytvoří server Dynamic Server společnou oblast vyrovnávacích pamětí se zadanou velikostí stránky.

Poznámka: Informace, které byly před verzí 10.0 určovány pomocí konfiguračních parametrů BUFFERS, LRUS, LRU_MAX_DIRTY a LRU_MIN_DIRTY, jsou nyní určovány pomocí konfiguračního parametru BUFFERPOOL. Informace zadané pomocí konfiguračního parametru BUFFERPOOL nahradí všechny informace zadané dříve pomocí již nepoužívaných parametrů. Další informace o již nepoužívaných parametrech naleznete v dodatku obsahujícím informace o těchto konfiguračních parametrech, který je v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Níže uvedená tabulka obsahuje popis hodnot, které určujete pomocí konfiguračního parametru BUFFERPOOL nebo obslužného programu **onparams**.

Pole	Popis	Rozsah hodnot
Velikost	Udává počet kilobajtů na stránku následovaný příponou k.	Velikost se může měnit od 2 kB nebo 4 kB až do 16 kB. Hodnota 2 kB je výchozí.
Pole buffers	<p>Udává počet vyrovnávacích pamětí pro velikost stránky.</p> <p>Jedná se o maximální počet vyrovnávacích pamětí ve sdílené paměti, které mají k dispozici jednotkové procesy uživatelů databázového serveru pro vstup - výstup na disk z pověření klientské aplikace. Počet vyrovnávacích pamětí, které požaduje databázový server, závisí na těchto aplikacích.</p> <p>Přistupuje-li například databázový server k 15 procentům dat aplikací 90 procent času, musíte přidělit dostatečný počet vyrovnávacích pamětí, které budou schopny těchto 15 % dat udržet. Zvýšení počtu vyrovnávacích pamětí může zvýšit výkon systému.</p> <p>Procento fyzické paměti potřebné pro prostor vyrovnávací paměti závisí na množství dostupné paměti v počítači a množství paměti použité pro jiné aplikace. U systémů s velkým množstvím dostupné fyzické paměti (4 GB a více) může být prostor vyrovnávací paměti až 90 procent fyzické paměti. U systémů s menším množstvím dostupné fyzické paměti se může prostor vyrovnávací paměti pohybovat mezi 20 až 25 procenty fyzické paměti.</p> <p>Po nastavení prostoru vyrovnávacích pamětí doporučujeme vypočítat všechny ostatní parametry sdílené paměti (buffers * velikost_systémové_stránky).</p>	<p>Pro 32bitové platformy se systémem UNIX</p> <ul style="list-style-type: none"> • s velikostí stránky 2048 B: 100 až 1843200 vyrovnávacích pamětí (1843200 = 1800 * 1024) • s velikostí stránky 4096 B: 100 až 921600 vyrovnávacích pamětí (921600 = ((1800 * 1024)/4096) * 2048) <p>Pro 32bitové platformy se systémem Windows: 100 až 524288 vyrovnávacích pamětí (524288 = 512 * 1024)</p> <p>Pro 64bitové platformy: 100 až $2^{31}-1$ vyrovnávacích pamětí (Skutečnou hodnotu pro použitou 64bitovou platformu naleznete v souboru Poznámky k verzi. Maximální počet vyrovnávacích pamětí v systému Solaris je 536870912.)</p>

Pole	Popis	Rozsah hodnot
lrus	<p>Určuje počet front LRU (Least-Recently-Used) ve společné oblasti vyrovnávacích pamětí pro danou velikost stránky. Laděním hodnoty parametru LRUS společně s hodnotami voleb lru_min_dirty a lru_max_dirty můžete řídit, jak často budou vyrovnávací paměti ve sdílené paměti vyprazdňovány na disk.</p> <p>Nastavení parametru LRUS na příliš vysokou hodnotu může způsobit příliš vysokou aktivitu jednotkových procesů čištění stránek.</p>	1 až 128
lru_min_dirty	<p>Určuje procento změněných stránek ve frontách LRU, po jehož dosažení již není vyžadováno čištění dalších stránek. Jednotkové procesy čištění stránek mohou za určitých okolností pokračovat v čištění stránek i po dosažení tohoto bodu.</p> <p>Hodnoty parametru LRU pro vyprazdňování společné oblasti vyrovnávací paměti mezi kontrolními body nejsou pro výkon kontrolních bodů důležité. Hodnota lru_min_dirty je obvykle potřebná pouze pro údržbu dostatečně aktualizovaných stránek v rámci jejich nahrazování. Začněte nastavením vyprazdňovacích parametrů LRU - parametr lru_min_dirty nastavte na hodnotu 70.</p> <p>Další informace naleznete v části "Hodnoty LRU pro vyprazdňování společné oblasti vyrovnávací paměti mezi kontrolními body" na stránce 15-6.</p>	<p>0 až 100 (jsou povoleny i neceločíselné hodnoty)</p> <p>Pokud byla zadána hodnota parametru mimo rozsah povolených hodnot, bude použita výchozí hodnota 50.00.</p>
lru_max_dirty	<p>Určuje procento změněných stránek ve frontách LRU, po jehož dosažení bude fronta vyčištěna.</p> <p>Hodnoty parametru LRU pro vyprazdňování společné oblasti vyrovnávací paměti mezi kontrolními body nejsou pro výkon kontrolních bodů důležité. Hodnota lru_max_dirty je obvykle potřebná pouze pro údržbu dostatečně aktualizovaných stránek v rámci jejich nahrazování. Začněte nastavením vyprazdňovacích parametrů LRU - parametr lru_max_dirty nastavte na hodnotu 80.</p> <p>Další informace naleznete v části "Hodnoty LRU pro vyprazdňování společné oblasti vyrovnávací paměti mezi kontrolními body" na stránce 15-6.</p>	<p>0 až 100 (jsou povoleny i neceločíselné hodnoty)</p> <p>Pokud byla zadána hodnota parametru mimo rozsah povolených hodnot, bude použita výchozí hodnota 60.00.</p>

Pokud se databázový server nachází v režimu online, v klidovém režimu nebo v režimu administrace, můžete také pomocí obslužného programu **onparams** přidat novou společnou oblast vyrovnávacích pamětí jiné velikosti. Pokud použijete obslužný program **onparams**, budou zadané údaje automaticky uloženy do souboru ONCONFIG a nové hodnoty budou určeny pomocí klíčového slova BUFFERPOOL. Hodnoty nelze změnit úpravou souboru **onconfig.std**.

Pokud použijete obslužný program **onparams**, zadejte údaje následovně:

```
onparams -b -g <velikost stránky vyrovnávací paměti v kB> -n
<počet vyrovnávacích pamětí>
-r <počet front LRU> -x <parametr max dirty
  (jsou povoleny neceločíselné hodnoty)>
-m <parametr min dirty (jsou povoleny neceločíselné hodnoty)>
```

Například:

```
onparams -b -g 6 -n 3000 -r 2 -x 2.0 -m 1.0
```

Tento příkaz přidá 3000 vyrovnávacích pamětí s velikostí každé paměti 6 kB a dvě fronty LRU s parametrem **lru_max_dirty** nastaveným na hodnotu 2 % a s parametrem **lru_min_dirty** nastaveným na hodnotu 1 %.

Další informace o obslužném programu **onparams** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Doporučení: Nastavte konfigurační parametr PHYSBUFF na hodnotu alespoň 128 kilobajtů. Pokud je databázový server konfigurován tak, aby používal RTO_SERVER_RESTART, nastavte konfigurační parametr PHYSBUFF na hodnotu alespoň 512 kilobajtů. Nastavení parametru PHYSBUFF na nižší hodnotu by mohlo mít vliv na transakční rychlost, což by během inicializace serveru mělo za následek zobrazení upozornění o snížení výkonu.

Záznamy protokolu LG_ADDDBPOOL a tabulka systémového katalogu **sysbufpool** obsahují údaje o všech společných oblastech vyrovnávacích pamětí.

Společné oblasti vyrovnávacích pamětí přidané do databázového serveru za chodu jsou uloženy ve virtuální paměti, nikoli v rezidentní paměti. Do rezidentní paměti jsou ukládány pouze ty společné oblasti vyrovnávacích pamětí, které byly určeny v souboru ONCONFIG v okamžiku spuštění databázového serveru.

Změna velikosti stávající společné oblasti vyrovnávacích pamětí: Pokud potřebujete změnit velikost existující společné oblasti vyrovnávacích pamětí, bude zapotřebí vypnout databázový server. Potom změňte velikost společné oblasti vyrovnávacích pamětí v souboru ONCONFIG.

Odstranění stávající společné oblasti vyrovnávacích pamětí: Pokud potřebujete odstranit existující společnou oblast vyrovnávacích pamětí, bude zapotřebí vypnout databázový server. Potom odstraňte společnou oblast vyrovnávacích pamětí v souboru ONCONFIG.

Definování velikosti stránky

Pomocí volby **onspaces -k** můžete nastavit velikost stránky v kilobajtech, a to následovně:

```
onspaces -c -d DBspace [-t] [-k pagesize] -p path -o offset -s size [-m path offset]
```

Kořenový prostor dbspace sestává ze stránek výchozí velikosti.

Pokud určíte velikost stránky, musí být velikost stránky násobkem výchozí velikosti stránky, ale nesmí být větší než 16 kB.

Pokud vytváříte prostor dbspace pomocí programu ON-Monitor, zadejte velikost stránky v kilobajtech v poli **Page Size**.

Další informace o programu ON-Monitor a obslužných programech **onspaces** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Zvýšení výkonu prostorů dbspace pro předpřipravené soubory pomocí přímého vstupu - výstupu

V systémech UNIX lze zvýšit výkon předpřipravených souborů použitých pro bloky prostoru dbspace pomocí přímého vstupu - výstupu. Dynamický server Informix umožňuje použít zařízení s přímým přístupem nebo předpřipravené soubory pro bloky prostoru dbspace. Obecně vzato, předpřipravené soubory jsou pomalejší kvůli dodatečnému zahlcení a činnosti vyrovnávací paměti, vyvolané systémem souborů. Přímý vstup - výstup obchází použití

vyrovnávacích pamětí systému souborů, a tak je při čtení a zápisu na disk výkonější. Přímý vstup - výstup lze zadat pomocí konfiguračního parametru `DIRECT_IO`. Pokud systém souborů podporuje přímý vstup - výstup u velikosti stránky použité pro blok prostoru dbSPACE a vy tento přímý vstup používáte, výkon předpřipravených souborů se přiblíží výkonu zařízení s přímým přístupem, které se používají u bloků prostoru dbSPACE.

Předpoklady: Pro velikost stránky použité u bloku prostoru dbSPACE je nutné, aby byl k dispozici přímý vstup - výstup, který musí být podporován systémem souborů.

Jak zvýšit výkon prostorů dbSPACE předpřipravených-souborů pomocí přímého vstupu - výstupu:

1. Ověřte, zda je pro velikost stránky použité u bloku prostoru dbSPACE k dispozici přímý vstup - výstup a zda jej podporuje systém souborů.
2. Nastavením konfiguračního parametru `DIRECT_TO` na hodnotu 1 přímý vstup - výstup povolíte.

Další informace naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Vytvoření dočasného prostoru dbSPACE

Pokud chcete určit, kde mají být přidělovány dočasné soubory, vytvořte dočasné prostory dbSPACE.

Definování dočasných prostorů dbSPACE:

1. Použijte obslužný program `onspaces` s volbami **-c -d -t**.

Další informace naleznete v části “Vytvoření prostoru dbSPACE, který používá výchozí velikost stránky” na stránce 10-9.

2. Pomocí proměnné prostředí **DBSPACETEMP** nebo konfiguračního parametru `DBSPACETEMP` určete prostory dbSPACE, které může databázový server používat jako dočasnou paměť.

Konfigurační parametr `DBSPACETEMP` může obsahovat prostory dbSPACE s velikostí stránky jinou než výchozí. Ačkoliv pro parametr `DBSPACETEMP` můžete do prostorů dbSPACE zařadit v seznamu parametrů různé velikosti stránky, databázový server používá pouze prostory dbSPACE se stejnou velikostí stránky jako je první prostor dbSPACE v seznamu.

Další informace o parametru `DBSPACETEMP` naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.

3. Pokud vytváříte více dočasných prostorů dbSPACE, měly by být tyto prostory dbSPACE z důvodu optimalizace operací vstupu - výstupu uloženy na různých discích.

Pokud vytváříte dočasný prostor dbSPACE, je zapotřebí informovat databázový server o existenci nově vytvořeného dočasného prostoru dbSPACE nastavením konfigurační proměnné `DBSPACETEMP`, proměnné prostředí **DBSPACETEMP** nebo obou těchto proměnných. Databázový server nezačne dočasný prostor dbSPACE používat, dokud neprovedete oba následující kroky:

- Nastavte konfigurační parametr `DBSPACETEMP`, proměnnou prostředí **DBSPACETEMP** nebo obě tyto volby.
- Restartujte databázový server.

Následující příklad znázorňuje, jak vytvořit dočasný prostor dbSPACE **temp_space** o velikosti 5 MB s posunem 5000 kB:

```
onspaces -c -t -d temp_space -p /dev/raw_dev1 -o 5000 -s 5120
```

Další informace naleznete v části “Dočasné prostory dbSPACE” na stránce 9-11.

Co dělat, pokud dojde k vyčerpání diskového prostoru

Pokud je počáteční blok vytvářeného prostoru dbSPACE předpřipraveným souborem systému UNIX nebo souborem systému NTFS v systému Windows, ověří databázový server, zda je k vytvoření počátečního bloku dostatečný diskový prostor. Pokud je velikost bloku větší než dostupný prostor na disku, bude zobrazena zpráva a prostor dbSPACE nebude vytvořen. Předpřipravený soubor, který databázový server vytvořil k uložení počátečního bloku, však nebude odstraněn. Jeho velikost reprezentuje diskový prostor, který zbyl v souborovém systému před vytvořením prostoru dbSPACE. Odstraňte tento soubor, aby se prostor opět uvolnil.

Přidání bloku do prostoru dbSPACE nebo blobSPACE

Blok přidejte tehdy, pokud dojde k zaplnění prostoru dbSPACE, blobSPACE nebo sbSPACE nebo pokud některý z těchto prostorů potřebuje více diskového prostoru. K přidání bloku použijte příkaz **onspaces** nebo program ISA. Informace o přidávání bloků pomocí programu ISA naleznete v nápovědě online programu ISA.

Důležité: Nově přidaný blok (a jeho přidružené zrcadlo, pokud existuje) je dostupný okamžitě. Pokud přidáváte blok do zrcadleného paměťového prostoru, je zapotřebí přidat také zrcadlený blok.

Přidání bloku pomocí obslužného programu onspaces:

1. Chcete-li vytvořit prostor dbSPACE v systému UNIX, musíte být přihlášení jako uživatel **informix** nebo **root**.
V systému Windows mohou bloky přidávat uživatelé, kteří jsou členy skupiny **Informix-Admin**.
2. Zajistěte, aby byl databázový server v režimu online, v režimu administrace, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.
3. Přidělte bloku diskový prostor podle popisu v části “Přidělení diskového prostoru” na stránce 10-3.
4. Přidejte blok pomocí volby **-a** obslužného programu **onspaces**.
Pokud je paměťový prostor zrcadlený, je třeba zadat názvy cest primárního i zrcadleného bloku.
Pokud zadáte nesprávný název cesty, posun nebo velikost, databázový server blok nevytvoří a zobrazí chybovou zprávu. Další informace naleznete také v části “Co dělat, pokud dojde k vyčerpání diskového prostoru” na stránce 10-18.
5. Po vytvoření bloku je třeba provést zálohování úrovně 0 kořenového prostoru dbSPACE a prostoru dbSPACE, blobSPACE nebo sbSPACE, který obsahuje tento blok.

V následujícím příkladu je do prostoru **blobSP3** přidán zrcadlový blok o velikosti 10 MB. Jako posun primárního i zrcadleného bloku je zadána hodnota 200 kB. Pokud nepřidáváte zrcadlený blok, můžete volbu **-m** vynechat.

```
onspaces -a blobSP3 -p /dev/raw_dev1 -o 200 -s 10240 -m /dev/raw_dev2 200
```

V následujícím příkladu je blok diskového prostoru s přímým přístupem o velikosti 5 MB přidán s posunem 5200 kB k prostoru dbSPACE **dbSPC3**.

```
onspaces -a dbSPC3 \\.\e: -o 5200 -s 5120
```

Referenční informace o přidávání bloků k prostorům dbSPACE pomocí obslužného programu **onspaces** naleznete v kapitole o obslužném programu **onspaces** v příručce *IBM Informix Administrator's Reference*.

Přidání bloku pomocí programu ON-Monitor (systém UNIX)

Blok přidejte k prostoru dbspace podle následujících pokynů:

1. Vyberte volbu **Add Chunk > Dbspaces**.
2. Pomocí šipek a klávesy RETURN vyberte prostor blobspace nebo prostor dbspace, ke kterému má být přidán blok a pak stiskněte klávesu CTRL-B nebo F3.
3. Následující obrazovka označuje, zda je prostor blobspace nebo dbspace zrcadlený. Pokud je zrcadlený, zadejte do pole **Mirror** hodnotu Y.
4. Pokud přidáváte blok k dočasnému prostoru dbspace, zadejte do pole **Temp** hodnotu Y.
5. Pokud jste určili, že prostor dbspace nebo blobspace je zrcadlený, je třeba zadat primární blok i zrcadlený blok.
V části týkající se primárního bloku zadejte do pole **Full Pathname** úplný název cesty k primárnímu bloku.
6. Do pole **Offset** zadejte posun.
7. Do pole **Size** zadejte velikost bloku v kilobajtech.
8. Pokud tento blok zrcadlíte, zadejte úplný název cesty, velikost bloku a nepovinný posun bloku v části této obrazovky týkající se zrcadleného bloku.

Přejmenování prostorů dbspace

Pomocí obslužného programu **onspaces** můžete prostor dbspace přejmenovat, pokud jste uživatel **informix** nebo máte oprávnění administrátora DBA a databázový server se nachází v klidovém režimu (nikoli v jiném režimu).

Prostor dbspace přejmenujte pomocí jednoho z následujících příkazů obslužného programu **onspaces**:

```
onspaces -ren starý_název_prostoru_dbspace-n nový_název_prostoru_dbspace
```

Přejmenovat můžete standardní prostory dbspace a všechny další prostory včetně prostorů blobspace, prostorů sbpace, dočasných prostorů a externích prostorů. Nemůžete však přejmenovat žádný z kritických prostorů, například kořenový prostor dbspace nebo prostor dbspace obsahující fyzické protokoly.

Prostory dbspace a prostory sbpace můžete přejmenovat:

- Pokud je povolena replikace Enterprise Replication.
- Na primárním databázovém serveru, pokud je povolena replikace dat.

Soubory dbspace a sbpace nemůžete přejmenovávat v sekundárním databázovém serveru a také v případě, že je sekundární databázový server součástí konfigurace replikace Enterprise Replication

Operace přejmenování prostoru dbspace pouze změní název prostoru dbspace, neprovede reorganizaci dat.

Příkaz přejmenování prostoru dbspace aktualizuje název prostoru dbspace všude, kde je tento název uložen. Mezi tato místa patří rezervované stránky na disku, systémové katalogy, konfigurační soubor ONCONFIG a paměťové datové struktury.

Upozornění: Po přejmenování prostoru dbspace proveďte archivaci úrovně 0 přejmenovaného prostoru dbspace a kořenového prostoru dbspace. Další informace naleznete v kapitole *IBM Informix Backup and Restore Guide*.

Další akce, které může být zapotřebí provést po přejmenování prostoru dbspace

Pokud přejmenujete prostor dbspace, bude zapotřebí přepsat veškerý kód uložených procedur, který bude odkazovat na prostor dbspace jeho původním názvem. Pokud například používáte uloženou proceduru, která obsahuje klíčové slovo ALTER FRAGMENT a odkazuje na název prostoru dbspace, přepište tuto uloženou proceduru a znovu ji zkompilujte.

Pokud přejmenováváte prostory určené v konfiguračním parametru DATASKIP, bude zapotřebí po přejmenování prostoru dbspace aktualizovat konfigurační parametr DATASKIP ručně.

Správa oddílů prostorů dbspace

Pro fragmentované tabulky, které používají schéma distribuce založené na výrazu nebo typu cyklická obsluha můžete vytvořit větší počet *oddílů*, což jsou kolekce stránek tabulky nebo indexu v rámci jediného prostoru dbspace.

Ukládání více fragmentů tabulky nebo rejstříku do jediného prostoru typu dbspace zlepšuje výkon dotazů v porovnání s ukládáním každého fragmentu do různých prostorů dbspace a zjednodušuje správu prostorů dbspace. Pokud rozdělíte tabulku nebo rejstřík do více oblastí uložených ve stejném prostoru dbspace, můžete pomocí těchto oblastí obejít aktuální omezení 16 milionů stránek.

Předpokládejme, že vytváříte fragmentovanou tabulku používající schéma distribuce založené na výrazu, ve kterém každý zadaný výraz určuje, jaká data budou uložena do konkrétních fragmentů. Můžete například určit, aby data jednoho měsíce byla uložena v jednom prostoru dbspace a data jedenácti dalších měsíců byla uložena v jedenácti dalších prostorech dbspace. Pokud však k uložení všech ročních dat chcete použít jediný prostor dbspace, můžete vytvořit oddíly, aby data každého měsíce byla uložena v samostatném oddílu tohoto prostoru dbspace.

Pokud vytvoříte fragmentovanou tabulku s oddíly, bude každý řádek systémového katalogu **sysfragments** obsahovat ve sloupci **Partition** název oddílu. Pokud vytvoříte fragmentovanou tabulku bez oddílů, bude ve sloupci **Partition** uložen název prostoru dbspace. Sloupec **Flags** katalogu **sysfragments** určuje, zda je schéma fragmentace založeno na oddílech.

Můžete vytvářet tabulky a indexy s oddíly a pomocí klíčového slova PARTITION a názvu oddílu můžete oddíly vytvářet, vypouštět a měnit.

K vytvoření fragmentované tabulky s oddíly použijte syntaxi jazyka SQL znázorněnou v následujícím příkladu:

```
CREATE TABLE tb1(a int)
  FRAGMENT BY EXPRESSION
  PARTITION part1 (a >=0 AND a < 5) IN dbspace1,
  PARTITION part2 (a >=5 AND a < 10) IN dbspace1
  ...
;
```

Pokud jste vytvořili fragment indexu nebo tabulky obsahující oddíly, používejte při použití příkazu příkazu ALTER FRAGMENT syntaxi tohoto příkazu zahrnující název oddílu, jak znázorňují následující příklady:

```
ALTER FRAGMENT ON TABLE tb1 INIT FRAGMENT BY EXPRESSION
  PARTITION part_1 (a >=0 AND a < 5) IN dbspace1,
  PARTITION part_2 (a >=5 AND a < 10) IN dbspace1;

ALTER FRAGMENT ON INDEX ind1 INIT FRAGMENT BY EXPRESSION
  PARTITION part_1 (a >=0 AND a < 5) IN dbspace1,
  PARTITION part_2 (a >=5 AND a < 10) IN dbspace1;
```

V příkazech CREATE TABLE, CREATE INDEX a ALTER FRAGMENT ON INDEX můžete místo klauzule FRAGMENT BY EXPRESSION použít klauzuli PARTITION BY EXPRESSION, jak znázorňuje následující příklad:

```
ALTER FRAGMENT ON INDEX idx1 INIT PARTITION BY EXPRESSION
    PARTITION part1 (a <= 10) IN idxdbspc1,
    PARTITION part2 (a <= 20) IN idxdbspc1,
    PARTITION part3 (a <= 30) IN idxdbspc1;
```

Pomocí příkazu ALTER FRAGMENT můžete změnit fragmentované tabulky a indexy nepoužívající oddíly na tabulky a indexy, které používají oddíly. Následující syntaxe znázorňuje, jak můžete převést fragmentovanou tabulku s několika prostory dbspace na tabulku používající oddíly:

```
CREATE TABLE t1 (c1 int) FRAGMENT BY EXPRESSION
    (c1=10) IN dbs1,
    (c1=20) IN dbs2;
ALTER FRAGMENT ON TABLE t1 MODIFY dbs2 TO PARTITION part_3 (c1=20)
    IN dbs1
```

Následující syntaxe znázorňuje, jak můžete převést fragmentovaný index na index používající oddíly:

```
CREATE TABLE t1 (c1 int) FRAGMENT BY EXPRESSION
    (c1=10) IN dbs1, (c1=20) IN dbs2, (c1=30) IN dbs3
CREATE INDEX ind1 ON t1 (c1) FRAGMENT BY EXPRESSION
    (c1=10) IN dbs1, (c1=20) IN dbs2, (c1=30) IN dbs3
ALTER FRAGMENT ON INDEX ind1 INIT FRAGMENT BY EXPRESSION
    PARTITION part_1 (c1=10) IN dbs1, PARTITION part_2 (c1=20) IN dbs1,
    PARTITION part_3 (c1=30) IN dbs1,
```

V příručce *Příručka výkonosti serveru IBM Informix Dynamic Server* naleznete další informace o fragmentaci včetně pravidel používání fragmentace, postupů fragmentování indexů, postupů vytváření připojených a odpojených indexů s oddíly a příkladů jazyka SQL používaných k vytváření připojených a odpojených indexů obsahujících oddíly.

V příručce *IBM Informix Guide to SQL: Syntax* naleznete další podrobné informace o syntaxi včetně zadávání informací o oddílech v příkazech GRANT FRAGMENT a REVOKE FRAGMENT a podrobné informace o používání klauzulí DROP, DETACH a MODIFY příkazu ALTER FRAGMENT.

Správa prostorů blobspace

Tato část popisuje, jak vytvořit prostor blobspace a určit velikost stránky blobpage. Databázový server může ukládat data typu TEXT a BYTE v prostorech dbspace i v prostorech blobspace, ale prostory blobspace jsou efektivnější. Informace o přidávání bloků naleznete v části “Přidání bloku do prostoru dbspace nebo blobspace” na stránce 10-18.

Vytvoření prostoru blobspace

Prostor blobspace můžete vytvořit pomocí obslužného programu **onspaces**, programu ISA nebo programu ON-Monitor. Zadejte název prostoru blobspace o délce až 128 znaků. Tento název musí být jedinečný a musí začínat podtržítkem nebo písmenem. V názvu jsou povolena písmena, číslice, podtržítka a znaky \$.

Důležité: Pokud je v databázovém serveru povoleno zrcadlení, můžete prostor blobspace zrcadlit. Zrcadlení se projeví okamžitě po vytvoření.

Před vytvořením prostoru blobspace:

1. Přidělte prostoru blobospace diskový prostor, jak je popsáno v části “Přidělení diskového prostoru” na stránce 10-3.
2. Určete optimální velikost stránky blobpage ve svém prostředí.
Pokyny naleznete v části “Určení velikosti stránky blobpage” na stránce 10-23.

Vytvoření prostoru blobospace pomocí obslužného programu onspaces:

1. Chcete-li vytvořit prostor blobospace v systému UNIX, musíte být přihlášení jako uživatel **informix** nebo **root**.

Aby mohl uživatel vytvořit prostor blobospace v systému Windows, musí být členem skupiny **Informix-Admin**.

2. Zajistěte, aby byl databázový server v režimu online, v režimu administrace, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.

3. K přidání prostoru blobospace použijte volby obslužného programu **onspaces -c -b**.

- a. Určete explicitní název cesty k prostoru blobospace. Pokud je prostor blobospace zrcadlený, je třeba zadat název cesty a velikost primárního i zrcadleného bloku.
- b. Pomocí volby **-o** zadejte posun prostoru blobospace.
- c. Pomocí volby **-s** určete velikost prostoru blobospace v kilobajtech.

- d. Pomocí volby **-g** určete velikost stránky blobpage. Jednotkou je počet diskových stránek, ze kterých se má skládat jedna stránka blobpage.

Další informace naleznete v části “Určení velikosti stránky blobpage” na stránce 10-23. Pokud například chcete, aby diskové stránky měly velikost 10 kB a instance databázového serveru používá diskové stránky o velikosti 2 kB, zadejte do tohoto pole hodnotu 5.

Pokud zadáte nesprávný název cesty, posun nebo velikost, databázový server prostor blobospace nevytvoří a zobrazí chybovou zprávu. Další informace naleznete také v části “Co dělat, pokud dojde k vyčerpání diskového prostoru” na stránce 10-18.

4. Po vytvoření prostoru blobospace musíte provést zálohování úrovně 0 kořenového prostoru dbspace a nového prostoru blobospace.

Následující příklad znázorňuje, jak vytvořit zrcadlený prostor blobospace **blobsp3** o velikosti 10 MB s velikostí stránky blobpage 10 kB. Velikost stránky databázového serveru je 2 kB. Jako posun primárního a zrcadleného bloku je určena hodnota 200 kB. Prostor blobospace je v systému UNIX vytvořen z diskového prostoru s přímým přístupem.

```
onspaces -c -b blobsp3 -g 5 -p /dev/raw_dev1 -o 200 -s 10240 -m /dev/raw_dev2 200
```

Referenční informace o vytváření prostoru blobospace pomocí obslužného programu onspaces naleznete v kapitole o obslužném programu **onspaces** v příručce *IBM Informix Administrator's Reference*.

Vytvoření prostoru blobospace pomocí programu ON-Monitor (UNIX):

1. Vyberte volbu **Dbspaces > BLOBSpace**.
2. Zadejte název nového prostoru blobospace do pole **BLOBSpace Name**.
3. Chcete-li vytvořit zrcadlo pro počáteční blok prostoru blobospace, zadejte do pole **Mirror** hodnotu Y.

V opačném případě zadejte hodnotu N.

4. Zadejte do pole **BLOBPage Size** počet diskových stránek, které mají tvořit jednu stránku blobpage.

Další informace naleznete v části “Určení velikosti stránky databázového serveru” na stránce 10-23. Pokud například chcete, aby diskové stránky měly velikost 10 kB a instance databázového serveru používá diskové stránky o velikosti 2 kB, zadejte do tohoto pole hodnotu 5.

5. Zadejte úplnou cestu k počítačícímu primárnímu bloku prostoru blobspace do pole **Full Pathname** v části primárního bloku.
6. Do pole **Offset** zadejte posun.
7. Do pole **Size** zadejte velikost bloku v kilobajtech.
8. Pokud tento blok zrcadlíte, zadejte úplný název cesty k bloku, velikost bloku a nepovinný posun bloku v části této obrazovky týkající se zrcadleného bloku.

Příprava prostorů blobspace k ukládání dat typu TEXT a BYTE

Nový prostor blobspace nelze k ukládání dat typu TEXT a BYTE používat okamžitě po vytvoření. Aby bylo možné protokolovat a obnovit prostor blobspace, musejí být záznamy protokolu týkající se příkazů, které do prostoru blobspace vkládají data typu TEXT a BYTE, uloženy v jiných souborech logického protokolu než záznam příkazu, který tento prostor blobspace vytvořil. Tento požadavek se vztahuje na všechny prostory blobspace nezávisle na stavu protokolování databáze. Aby byl tento požadavek splněn, přepněte po vytvoření prostoru blobspace na následující soubor logického protokolu. (Pokyny naleznete v části “Zálohování souborů protokolů za účelem uvolnění stránek blobpage” na stránce 13-7.)

Určení velikosti stránky blobpage

Pokud vytváříte prostor blobspace, použijte jako velikost stránky blobpage velikost nejčastěji se vyskytujícího jednoduchého velkého objektu. Jinými slovy, vyberte takovou velikost stránky blobpage, která způsobí nejmenší plýtvání diskovým prostorem. Informace o výpočtu optimální velikosti stránky blobpage naleznete v části týkající se posouzení velikosti stránek blobpage v kapitole o vlivech konfigurace na aktivitu vstupu - výstupu v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Pokud tabulka obsahuje více sloupců typu TEXT nebo BYTE a objekty v těchto sloupcích mají různou velikost, uložte tyto sloupce do různých prostorů blobspace tak, aby každý měl vhodnou velikost stránky. Další informace naleznete v části “Tabulky” na stránce 9-22.

Určení velikosti stránky databázového serveru

Velikost stránek blobpage se určuje jako násobek velikosti stránek databázového serveru. Velikost stránky databázového serveru můžete určit pomocí jedné z následujících metod:

- Spusťte příkaz **onstat -b**, který zobrazí systémovou velikost stránky jako velikost vyrovnávací paměti v posledním řádku výstupu příkazu.
- Zobrazte obsah rezervované stránky PAGE_PZERO pomocí příkazu **oncheck -pr**.
- Pouze systém **UNIX**: V programu ON-Monitor zobrazte systémovou velikost stránky pomocí volby **Parameters > Shared-Memory** nebo **Parameters > Initialize**.

Získání statistických údajů o paměti prostoru blobspace

Při určování optimální velikosti stránek blobpage všech prostorů blobspace mohou pomoci následující příkazy obsluhujících programů databázového serveru:

- **oncheck -pe**
- **oncheck -pB**

Příkaz **oncheck -pe** poskytuje podrobné informace o objektech uložených v prostoru blobspace:

- Úplné informace o vlastnictví (zobrazené ve formátu *databáze:vlastnik.tabulka*) o každé tabulce, jejíž data jsou uložena v bloku prostoru blobspace.
- Celkový počet stránek používaných každou tabulkou k uložení přidružených dat typu TEXT a BYTE.
- Celkový počet volných stránek a celkový počet režijních stránek v prostoru blobspace.

Příkaz **oncheck -pB** zobrazuje následující statistické údaje o každé tabulce databáze:

- Počet stránek blobpage používaných tabulkou nebo databází v každém z prostorů blobspace.
- Průměrné zaplnění stránek blobpage používaných všemi jednoduchými velkými objekty uloženými jako součást tabulky nebo databáze.

Další informace naleznete v části “Monitorování využití prostoru blobspace pomocí příkazu oncheck -pe” na stránce 10-40, v části “Určení zaplnění stránek blobpage pomocí příkazu oncheck -pB” na stránce 10-40 a v části týkající se optimalizace velikosti stránek blobpage v prostorech blobspace v kapitole o posouzení výkonu tabulek v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Správa prostorů sbpace

Tato část popisuje, jak vytvořit standardní nebo dočasný prostor sbpace, jak monitorovat metadata, jak přidat blok k prostoru sbpace a jak změnit paměťové charakteristiky inteligentního velkého objektu.

Vytvoření prostoru sbpace

K vytvoření prostoru sbpace použijte obslužný program onspaces nebo program ISA.

Vytvoření prostoru sbpace pomocí obslužného programu onspaces:

1. Chcete-li vytvořit prostor sbpace v systému UNIX, musíte být přihlášení jako uživatel **informix** nebo **root**.
Aby mohl uživatel vytvořit prostor sbpace v systému Windows, musí být členem skupiny **Informix-Admin**.
2. Zajistěte, aby byl databázový server v režimu online, v režimu administrace, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.
3. Vytvořte prostor sbpace pomocí příkazu **onspaces -c -S**.
 - a. Pomocí volby **-p** zadejte název cesty, pomocí volby **-o** zadejte posun a pomocí volby **-s** zadejte velikost prostoru sbpace.
 - b. Pokud chcete prostor sbpace zrcadlit, zadejte pomocí volby **-m** cestu k zrcadlu a posun zrcadla.
 - c. Pokud chcete použít výchozí paměťové charakteristiky prostoru sbpace, volbu **-Df** vynechte.
Pokud chcete určit odlišné paměťové charakteristiky, použijte volbu **-Df**. Další informace naleznete v části “Paměťové charakteristiky prostorů sbpace” na stránce 9-15.
 - d. První blok prostoru sbpace musí obsahovat oblast metadat.
Výpočet velikosti oblasti metadat přenechat databázovému serveru nebo můžete oblast metadat prostoru sbpace určit sami. Další informace naleznete v části “Určení velikosti metadat prostoru sbpace” na stránce 10-25.
4. Po vytvoření prostoru sbpace musíte provést zálohování úrovně 0 kořenového prostoru dbspace a nového prostoru sbpace.
5. Abyste mohli začít ukládat inteligentní velké objekty do tohoto prostoru sbpace, určete název tohoto prostoru v konfiguračním parametru SBSPACENAME.
6. K zobrazení informací o prostoru sbpace použijte příkazy **onstat -d**, **onstat -g smb s a oncheck -cs**, **-cS**, **-ps** nebo **-pS**.
Další informace naleznete v části “Monitorování prostorů sbpace” na stránce 10-41.

Tento příklad znázorňuje, jak vytvořit zrcadlený prostor sbspace **sbsp4** o velikosti 20 MB. Jako posun primárního i zrcadleného bloku je určena hodnota 500 kB a velikost oblasti metadat je určena jako 150 kB s posunem 200 kB. Parametr `AVG_LO_SIZE` volby **-Df** určuje, že je očekávána průměrná velikost inteligentního velkého objektu 32 kB.

```
onspaces -c -S sbsp4 -p /dev/rawdev1 -o 500 -s 20480 -m /dev/rawdev2 500  
-Ms 150 -Mo 200 -Df "AVG_LO_SIZE=32"
```

Informace o vytváření prostoru sbspace a o výchozích volbách pro inteligentní velké objekty naleznete v části týkající se obslužného programu **onspaces** v příručce *IBM Informix Administrator's Reference*. Informace o vytváření inteligentních velkých objektů naleznete v příručce *IBM Informix DataBlade API Programmer's Guide* a v příručce *IBM Informix ESQ/L/C Programmer's Manual*.

Vytvoření prostoru sbspace pomocí obslužného programu ISA:

1. Vytvořte prostor sbspace pomocí programu
2. ISA. Další informace naleznete v nápovědě online k programu ISA.
3. Vytvořte zálohu nového prostoru sbspace a kořenového prostoru dbspace.

Určení velikosti metadat prostoru sbspace

První blok prostoru sbspace musí obsahovat oblast metadat. Pokud budete do prostoru sbspace přidávat inteligentní velké objekty a bloky, bude velikost oblasti metadat narůstat. Databázový server kromě toho rezervuje 40 % uživatelské oblasti k použití v případě zaplnění oblasti metadat.

Je důležité správně nastavit velikost oblasti metadat prostoru sbspace, aby nedošlo k zaplnění této oblasti. Můžete postupovat jedním z následujících způsobů:

- Můžete přenechat výpočet velikosti oblasti metadat databázovému serveru.
- Můžete zadat velikost oblasti metadat explicitně.

Pokyny k určení velikosti prostoru sbspace a oblasti metadat naleznete v části o posouzení výkonu tabulek v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*. Další informace naleznete také v části "Monitorování oblastí metadat a uživatelských dat" na stránce 10-45.

Přidání bloku do prostoru sbspace

K přidání bloku k prostoru sbspace nebo k dočasnému prostoru sbspace použijte obslužný program **onspaces** nebo program ISA. Výpočet velikosti oblasti metadat přenechat databázovému serveru, dále můžete oblast metadat prostoru sbspace určit sami nebo můžete používat blok pouze k ukládání uživatelských dat.

Přidání bloku pomocí obslužného programu onspaces:

1. Zajistěte, aby byl databázový server v režimu online, v režimu administrace, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.
2. Vytvořte blok prostoru sbspace pomocí příkazu **onspaces -a**.
 - a. Pomocí volby **-p** zadejte název cesty, pomocí volby **-o** zadejte posun a pomocí volby **-s** zadejte velikost bloku.
 - b. Pokud chcete blok zrcadlit, zadejte pomocí volby **-m** cestu k zrcadlu a posun zrcadla.
 - c. Velikost a posun prostoru metadat určete pomocí voleb **-Mo** a **-Ms**.
Databázový server přidělí v novém bloku určenou oblast metadat.
 - d. Pokud chcete výpočet velikosti oblasti metadat přenechat databázovému serveru, vynechtejте volby **-Mo** a **-Ms**.

Databázový server vydělí odhadnutou průměrnou velikost inteligentních velkých objektů velikostí oblasti uživatelských dat.

e. Pokud chcete blok používat pouze k ukládání uživatelských dat, zadejte volbu **-U**.

Pokud použijete volbu **-U**, databázový server v tomto bloku nepřidělí oblast metadat. Místo toho bude prostor sbspace používat oblast metadat v některém z jiných bloků.

- Po přidání bloku do prostoru sbspace запиše databázový server záznamy protokolu CHRESERV a CHKADJUP.
- Proveďte zálohování úrovně 0 kořenového prostoru dbspace a prostoru sbspace.
- Pomocí příkazů **onstat -d** a **oncheck -pe** monitorujte množství volného prostoru v bloku prostoru sbspace.

Další informace naleznete v části “Monitorování prostorů sbspace” na stránce 10-41.

V následujícím příkladu je do prostoru **sbsp4** přidán zrcadlený blok o velikosti 10 MB. Jako posun primárního i zrcadleného bloku je zadána hodnota 200 kB. Pokud nepřidáváte zrcadlený blok, můžete volbu **-m** vynechat. Volba **-U** určuje, že nový blok bude obsahovat výhradně uživatelská data.

```
onspaces -a sbsp4 -p /dev/rawdev1 -o 200 -s 10240 -m /dev/rawdev2 200 -U
```

Další informace naleznete v části “Přidání bloku do prostoru dbspace nebo blobspace” na stránce 10-18 a v kapitole věnované obsluhnému programu **onspaces** v příručce *IBM Informix Administrator's Reference*.

Změna paměťových charakteristik inteligentních velkých objektů

Pomocí příkazu **onspaces -ch** můžete změnit následující výchozí paměťové charakteristiky prostoru sbspace:

- velikosti oblastí
- průměrnou velikost inteligentního velkého objektu
- režim použití vyrovnávací paměti
- čas posledního přístupu
- režimy uzamykání
- protokolování

Další informace naleznete v části “Paměťové charakteristiky prostorů sbspace” na stránce 9-15 a v části týkající se správy prostorů sbspace v kapitole o posouzení výkonu tabulek v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Vytvoření dočasného prostoru dbspace

Podrobné informace a pravidla k určení, kam ukládat dočasné inteligentní velké objekty naleznete v části “Dočasné prostory sbspace” na stránce 9-19. Dočasné inteligentní velké objekty můžete ukládat do standardního prostoru sbspace nebo do dočasného prostoru sbspace. Bloky dočasného prostoru sbspace můžete přidávat a vypouštět.

Vytvoření dočasného prostoru sbspace pro dočasné inteligentní velké objekty:

- Přidělte dočasnému prostoru sbspace diskový prostor. Další informace naleznete v části “Přidělení diskového prostoru” na stránce 10-3.

Informace o parametru SBSPACETEMP naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.

- Vytvořte dočasný prostor sbspace, jak znázorňuje následující příklad:

```
onspaces -c -S tempsbsp -t -p ./tempsbsp -o 0 -s 1000
```

Můžete zadat libovolné z následujících voleb obslužného programu **onspaces**:

- a. Velikost a posun oblasti metadat (volby **-Ms** a **-Mo**).
- b. Paměťové charakteristiky (volba **-Df**).

Protokolování dočasného prostoru sbspace nelze zapnout.

3. Nastavte konfigurační parametr SBSPACETEMP na název paměťové oblasti výchozího dočasného prostoru sbspace.

Restartujte databázový server.

4. Zobrazte dočasný prostor sbspace pomocí příkazu **onstat -d**.

Více informací o příkazu **onstat -d** a příkladu jeho výstupu naleznete v kapitole týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

5. Při vytváření dočasného inteligentního velkého objektu zadejte příznak **LO_CREATE_TEMP**.

Použití rozhraní DataBlade API:

```
mi_lo_specset_flags(lo_spec,LO_CREATE_TEMP);
```

Použití ESQL/C:

```
ifx_lo_specset_flags(lo_spec,LO_CREATE_TEMP);
```

Informace o vytváření inteligentních velkých objektů naleznete v příručce *IBM Informix DataBlade API Programmer's Guide* a v příručce *IBM Informix ESQL/C Programmer's Manual*.

Vypuštění bloku

K vypuštění bloku z prostoru dbspace použijte obslužný program **onspaces** nebo program ISA.

Před vypuštěním bloku se podle údajů z následující tabulky přesvědčte, zda se databázový server nachází ve správném režimu.

Typ bloku	Databázový server v režimu online	Databázový server v režimu administrace nebo v klidovém režimu	Databázový server v režimu offline
Blok prostoru dbspace	Ano	Ano	Ne
Blok dočasného prostoru dbspace	Ano	Ano	Ne
Blok prostoru blobspace	Ne	Ano	Ne
Blok prostoru sbspace nebo dočasného prostoru sbspace	Ano	Ano	Ne

Ověření, zda je blok prázdný

Aby bylo možné úspěšně vypustit blok z prostoru dbspace pomocí jednoho z následujících obslužných programů, nesmí blok obsahovat žádná data. Všechny stránky kromě režijních musejí být uvolněny.

Pokud zůstanou v bloku stránky přidělené jiným entitám než režijním, vrátí obslužný program následující chybu:

Blok není prázdný.

Pokud se navíc prostor dbspace skládá ze dvou a více bloků a tyto další bloky neobsahují uživatelská data, nelze tyto další bloky odstranit, pokud obsahují prostor tblspace **tblspace**.

Pokud obdržíte zprávu `Chunk is not empty`, zobrazte obsah oblasti pomocí příkazu **oncheck -pe** a určete, která tabulka nebo jiná entita stále zabírá prostor v bloku.

Obvykle lze stránky odstranit po vypuštění tabulky, která je jejich vlastníkem. Pak zadejte příkaz obslužného programu znovu.

Vypuštění bloku z prostoru dbspace pomocí obslužného prostoru onspaces

V následujícím příkladu je v systému UNIX vypuštěn blok z prostoru **dbsp3**. Je zadán posun 300 kB.

```
onspaces -d dbsp3 -p /dev/raw_dev1 -o 300
```

Pomocí syntaxe uvedené v předchozím příkladu nemůžete vypustit počáteční blok prostoru dbspace. Namísto toho musíte vypustit prostor dbspace. Použijte sloupec **fchunk** příkazu **onstat -d** k určení počátečního bloku prostoru dbspace. Další informace o příkazu **onstat** naleznete v kapitole věnované obslužnému programu **onspaces** v příručce *IBM Informix Administrator's Reference*.

Další informace o vypuštění bloku z prostoru dbspace příkazem **onspaces** naleznete v příručce *IBM Informix Administrator's Reference*.

Vypuštění bloku z prostoru blobspace

Postup při vypouštění bloku z prostoru blobspace je totožný s postupem při vypouštění bloku z prostoru dbspace, který je popsán v části “Vypuštění bloku z prostoru dbspace pomocí obslužného prostoru onspaces” na stránce 10-28 s výjimkou toho, že databázový server musí být v klidovém režimu nebo v režimu administrace. Kromě této podmínky je nutné jen nahradit název prostoru blobspace ve všech místech, kde se vyskytne odkaz na prostor dbspace.

Vypuštění bloku z prostoru sbospace příkazem onspaces

V následujícím příkladu je v systému UNIX vypuštěn blok z prostoru **sbsp3**. Je zadán posun 300 kB. Databázový server musí být při vypouštění bloku z prostoru sbospace nebo dočasněho prostoru sbospace v režimu administrace online nebo klidovém režimu.

```
onspaces -d sbsp3 -p /dev/raw_dev1 -o 300
```

Počáteční blok prostoru sbospace se syntaxí v předchozím příkladu nemůžete vypustit. Namísto toho musíte vypustit prostor sbospace. Použijte sloupec **fchunk** příkazu **onstat -d** k určení toho, který blok je počáteční blok prostoru sbospace.

Použití možnosti -f (force)

Můžete použít možnost **-f** příkazu **onspaces** k vypuštění bloku prostoru sbospace bez přidělených metadat. Pokud blok obsahuje metadata pro prostor sbospace, musíte vypustit celý prostor sbospace. Použijte část **Blok** příkazu **onstat -d** k určení, které bloky prostoru sbospace obsahují metadata.

```
onspaces -d sbsp3 -f
```

Upozornění: Pokud vynutíte vypuštění prostoru sbospace, můžete způsobit problémy s konzistencí mezi tabulkami a prostory sbospace.

Odstranění inteligentních velkých objektů, na které neukazují žádné ukazatele

Každý inteligentní velký objekt má určitý počet odkazů - počet ukazatelů na inteligentní velký objekt. Je-li počet odkazů větší než 0, databázový server předpokládá, že je inteligentní velký objekt používán a neodstraní ho.

Inteligentní velký objekt málokdy zůstává s počtem ukazatelů 0. Příkaz **onspaces -cl** můžete použít k odstranění všech inteligentních velkých objektů, které mají počet odkazů rovný 0, pokud nejsou otevřeny nějakou aplikací.

Další informace o příkazu **onspaces** naleznete v kapitole věnované obslužnému programu **onspaces** v příručce *IBM Informix Administrator's Reference*.

Vypuštění paměťového prostoru

Pomocí obslužného programu **onspaces**, ISA nebo ON-Monitor můžete vypustit prostor dbspace, dočasný prostor dbspace, prostor blobspace, prostor sbspace, dočasný prostor sbspace nebo prostor extspace.

Chcete-li vypustit paměťový prostor v systému UNIX, musíte být přihlášení jako uživatel root nebo **informix**. Chcete-li vypustit paměťový prostor v systému Windows, musíte být členem skupiny **Informix-Admin**.

Paměťový prostor je možné vypustit pouze tehdy, je-li databázový server v režimu online, v režimu administrace nebo v klidovém režimu.

Příprava na vypuštění paměťového prostoru

Dříve než vypustíte paměťový prostor dbspace, je třeba vypustit všechny databáze a tabulky, které jste předtím vytvořili v prostoru dbspace. Nelze vypustit kořenový prostor dbspace.

Než vypustíte prostor blobspace, musíte vypustit všechny tabulky obsahující sloupec typu TEXT nebo BYTE, který odkazuje na prostor blobspace.

Příkazem **oncheck -pe** ověříte, zda nejsou uloženy v prostoru dbspace a blobspace nějaké tabulky a soubory protokolů.

Před vypuštěním souboru sbspace je třeba vypustit všechny tabulky, které obsahují sloupec CLOB a BLOB, který odkazuje na objekty uložené v prostoru sbspace. Pro prostor sbspace není potřeba odstranit sloupce, které odkazují na prostor sbspace, tyto sloupce ale musí být nulové; tzn. všechny inteligentní velké objekty nesmí být umístěny v prostoru sbspace.

Rada: Pokud vypustíte tabulky v prostoru dbspace, ve kterém jsou prováděna lehká připojení, mohou lehká připojení být pomalejší, než očekáváte. Příznakem tohoto problému je aktivita fyzického protokolování. Pokud jsou lehká připojení pomalejší, než očekáváte, přesvědčte se, zda před lehkými připojeními nebo v jejich průběhu nebyly v prostoru dbspace vypuštěny žádné tabulky. Pokud jste vypustili tabulky, vynuťte kontrolní bod pomocí příkazu **onmode -c**, než začnete provádět lehká připojení.

Vypuštění zrcadlového paměťového prostoru

Pokud vypustíte paměťový prostor, který je zrcadlen, jsou vypouštěny i zrcadlové prostory.

Pokud chcete vypustit pouze zrcadlo paměťového prostoru, vypněte zrcadlení. (Viz "Ukončení zrcadlení" na stránce 18-7.) Tato akce vypustí zrcadlo prostoru dbspace, blobspace nebo sbspace a uvolní bloky pro jiná použití.

Vypuštění paměťového prostoru s obslužným programem onspaces

Paměťový prostor s obslužným programem onspaces vypustíte pomocí volby **-d** názorně předvedené v následujících příkladech.

Tento příklad vypustí prostor dbspace zvaný **dbspce5** a jeho zrcadla.
onspaces -d dbspce5

Tento příklad vypustí prostor dbspace zvaný **blobsp3** a jeho zrcadla.
onspaces -d blobsp3

Pomocí volby **-d** s volbou **-f** vypustíte prostor sbspace, který obsahuje data. Pokud vynecháte volbu **-f**, nebudete moci vypustit prostor sbspace, který obsahuje data. Tento příklad vypustí prostor sbspace zvaný **sbspc4** a jeho zrcadla.
onspaces -d sbspc4 -f

Upozornění: Pokud použijete volbu **-f**, budou tabulky databázového serveru pravděpodobně obsahovat neplatné ukazatele na odstraněné inteligentní velké objekty.

Další informace o vypouštění paměťového prostoru pomocí obslužného programu **onspaces** naleznete v kapitole **onspaces** v příručce *IBM Informix Administrator's Reference*.

Vypuštění prostorů dbspace a blobspace pomocí programu ON-Monitor (UNIX)

Prostor dbspace nebo blobspace vypustíte pomocí programu ON-Monitor podle následujících pokynů:

1. Vyberte volbu **Dbspaces > Drop**.
2. Pomocí šipek a klávesy RETURN se přesuňte na prostor dbspace nebo blobspace, který chcete odstranit.
3. Stiskněte klávesu CTRL-B nebo F3.

Budete požádáni o potvrzení, zda opravdu chcete vypustit prostor dbspace nebo blobspace.

Zálohování po vypuštění paměťového prostoru

Pokud vytváříte paměťový prostor s totožným názvem, jaký měl odstraněný paměťový prostor, proveďte zálohování úrovně 0, abyste zajistili, že budoucí obnovení nezamění nový paměťový prostor s původním paměťovým prostorem. Další informace naleznete v příručce *IBM Informix Backup and Restore Guide*.

Upozornění: Po vypuštění prostoru dbspace, blobspace nebo sbspace bude nově uvolněné bloky možné znovu přidělit jiným prostorům dbspace, blobspace nebo sbspace. Před opětovným přidělením nově uvolněných bloků je však zapotřebí provést zálohování úrovně 0 kořenového prostoru dbspace a změněného paměťového prostoru. Pokud toto zálohování neprovedete a budete později muset provést obnovení, nemusí se obnovení zdařit, protože rezervované stránky zálohování nebudou aktuální.

Správa prostorů extspace

Prostory extspace nevyžadují přidělení diskového prostoru. Prostory extspace můžete vytvářet a vypouštět pomocí obslužného programu **onspaces**. Další informace o prostorech extspace najdete v části "Prostory extspace" na stránce 9-21.

Vytvoření prostoru extspace

Prostor extspace můžete vytvořit pomocí obslužného programu **onspaces**. Nejprve však budete potřebovat platný zdroj dat a platnou přístupovou metodu, pomocí které budete k tomuto zdroji dat přistupovat. Přestože prostor extspace můžete vytvořit i bez platného zdroje dat nebo bez platné přístupové metody, skončí jakýkoli pokus o načtení dat z takového

prostoru extspace chybou. Informace o přístupových metodách naleznete v příručce *IBM Informix Virtual-Table Interface Programmer's Guide*.

K vytvoření prostoru extspace pomocí obslužného programu **onspaces** použijte volbu **-c**, jak znázorňuje následující příklad. Následující příklad znázorňuje, jak vytvořit prostor extspace **pass_space**, přidružený k souboru hesel systému UNIX.

```
onspaces -c -x pass_space -l /etc/passwd
```

Zadejte název prostoru extspace o délce až 128 znaků. Tento název musí být jedinečný a musí začínat podtržítkem nebo písmenem. V názvu jsou povolena písmena, číslice, podtržítka a znaky \$.

Důležité: Předcházející příklad předpokládá, že jste naprogramovali rutinu poskytující funkce, pomocí kterých lze správně přistupovat k souboru **passwd** a že tento soubor existuje. Po vytvoření prostoru extspace je zapotřebí použít příslušné příkazy, které umožní přístup k datům souboru **passwd**. Další informace o uživatelských přístupových metodách naleznete v příručce *IBM Informix Virtual-Table Interface Programmer's Guide*.

Referenční informace o vytváření prostorů extspace pomocí obslužného programu **onspaces** naleznete v kapitole o obslužných programech v příručce *IBM Informix Administrator's Reference*.

Vypuštění prostoru extspace

K vypuštění prostoru extspace pomocí obslužného programu **onspaces** použijte volbu **-d**, jak znázorňují následující příklady. Prostor extspace nelze vypustit, pokud je přidružen k existující tabulce nebo k existujícímu indexu.

V následujícím příkladu je vypuštěn prostor extspace s názvem **pass_space**.

```
onspaces -d pass_space
```

Přeskakování nepřístupných fragmentů

Jednou z výhod fragmentace je možnost přeskočit fragmenty tabulek, které v průběhu operace vstupu - výstupu nejsou dostupné. Dotaz může například pokračovat dokonce i v případě, že se fragment nachází na disku, který je v okamžiku provádění dotazu vypnut z důvodu selhání disku. Pokud dojde k takové situaci, ovlivní selhání disku pouze část dat fragmentované tabulky. Pokud naopak dojde k selhání disku, na kterém jsou uloženy nefragmentované tabulky, budou tyto tabulky zcela nepřístupné.

Tato funkce je řízena následovně:

- Administrátorem databázového serveru, který nastavuje konfigurační parametr **DATASKIP**.
- Jednotlivými aplikacemi pomocí příkazu **SET DATASKIP**.

Použití konfiguračního parametru **DATASKIP**

Parametr **DATASKIP** můžete nastavit na hodnotu **OFF**, **ALL** nebo **ON** *seznam_prostorů_dbSPACE*. Nastavení **OFF** znamená, že databázový server nebude přeskakovat žádné fragmenty. Pokud bude fragment nedostupný, skončí dotaz chybou. Nastavení **ALL** označuje, že budou přeskočeny všechny nedostupné fragmenty. Nastavení **ON** *seznam_prostorů_dbSPACE* sděluje databázovému serveru, aby přeskočil jakékoli fragmenty umístěné v zadaných prostorech *dbSPACE*.

Použití funkce Dataskip obslužného programu onspaces

Pomocí funkce dataskip obslužného programu onspaces můžete určit prostory dbspace, které mají být přeskočeny, pokud budou nedostupné. Následující příkaz například nastaví parametr DATASKIP tak, aby databázový server přeskakoval fragmenty v prostorech **dbspace1** a **dbspace3**, ale ne v prostoru **dbspace2**:

```
onspaces -f ON dbspace1 dbspace3
```

Úplnou syntaxi této volby obslužného programu **onspaces** naleznete v kapitole o obslužných programech v příručce *IBM Informix Administrator's Reference*.

Použití obslužného programu onstat ke kontrole stavu funkce dataskip

Pomocí obslužného programu **onstat** můžete zobrazit seznam prostorů dbspace, které jsou aktuálně ovlivněny funkcí dataskip. Volba **-f** zobrazí prostory dbspace nastavené pomocí konfiguračního parametru DATASKIP i prostory nastavené pomocí volby **-f** obslužného programu **onspaces**.

Po spuštění příkazu **onstat -f** se zobrazí zpráva oznamující, zda se konfigurační parametr DATASKIP nachází v režimu **on** pro všechny prostory dbspace, v režimu **off** pro všechny prostory dbspace nebo v režimu **on** pro specifické prostory dbspace.

Použití příkazu SET DATASKIP jazyka SQL

Aplikace může řídit, zda mají být přeskakovány nedostupné fragmenty pomocí příkazu SET DATASKIP jazyka SQL. Aplikace by měly tento příkaz používat pouze ve zvláštních případech, protože způsobuje, že dotazy vracejí různé výsledky v závislosti na aktuální dostupnosti bloků, které používají. Podobně jako konfigurační parametr DATASKIP přijímá příkaz SET DATASKIP seznam prostorů dbspace, který databázovému serveru oznamuje, které fragmenty lze přeskočit. Předpokládejme například, že aplikační programátor vložil na začátek aplikace následující příkaz:

```
SET DATASKIP ON dbspace1, dbspace5
```

Tento příkaz způsobí, že databázový server přeskočí prostor **dbspace1** nebo **dbspace5**, pokud budou splněny obě následující podmínky:

- Aplikace se pokusí přistupovat k jednomu z těchto prostorů dbspace.
- Databázový server zjistí, že jeden z prostorů dbspace je nedostupný.

Pokud databázový server, že jsou nedostupné prostory **dbspace1** i **dbspace5**, přeskočí oba tyto prostory dbspace.

Nastavení DEFAULT příkazu SET DATASKIP umožňuje administrátorovi databázového serveru řídit funkci dataskip. Předpokládejme například, že vývojář použije v aplikaci následující příkaz:

```
SET DATASKIP DEFAULT
```

Pokud je po tomto příkazu jazyka SQL spuštěn dotaz, zkontroluje databázový server hodnotu konfiguračního parametru DATASKIP. Pokud koncovým uživatelům doporučíte používat toto nastavení, bude administrátor databázového serveru moci určit, které prostory dbspace mají být přeskakovány, jakmile zjistí, že jeden nebo více prostorů dbspace není dostupných.

Vliv funkce dataskip na transakce

Pokud funkci dataskip zapnete, příkaz SELECT bude vždy proveden. Dále bude vždy úspěšně dokončen příkaz INSERT, pokud je tabulka fragmentovaná v režimu cyklická obsluha a

alespoň jeden fragment se nachází v režimu online. Databázový server však *nedokončí* operace zapisující do databáze, pokud hrozí riziko, že operace ohrozí integritu databáze. Následující operace se nepodaří:

- Všechny operace UPDATE a DELETE, ve kterých se databázovému serveru nepodaří eliminovat vypnuté fragmenty.
Pokud databázový server *dokáže* eliminovat vypnuté fragmenty, bude operace odstranění nebo aktualizace dokončena úspěšně, ale tento výsledek je nezávislý na nastavení parametru DATASKIP.
- Operace INSERT vkládající data do tabulky fragmentované podle schématu distribuce založeného na výrazu, v případě, že příslušný fragment je vypnutý.
- Jakákoli operace vyžadující kontrolu referenčních omezení, pokud jsou k této kontrole zapotřebí data uložená ve vypnutém fragmentu.
Pokud například aplikace odstraňuje řádek, ke kterému existují podřízené řádky, musí být možné odstranit také podřízené řádky.
- Jakákoli operace, která mění hodnotu indexu (například aktualizace indexovaného sloupce), pokud se příslušný index nachází ve vypnutém bloku.

Jak určit, kdy používat parametr dataskip

Tuto funkci používejte zřídka a opatrně, protože výsledky operací s použitím této funkce jsou vždy podezřelé. Použití této funkce zvažte v následujících situacích:

- Narušení integrity transakcí je pro vás přijatelné.
- Dokážete určit, zda integrita transakce nebyla narušena.

Druhá úloha může být obtížná a časově náročná.

Určení, kdy přeskakovat vybrané fragmenty

Za určitých okolností můžete chtít, aby databázový server přeskočil některé fragmenty, ale aby ostatní nepřeskakoval. K tomu obvykle dochází v následujících situacích:

- Fragmenty lze přeskočit, protože nepřispívají významně k výsledku dotazu.
- Přestože jsou některé fragmenty vypnuté, rozhodnete se, že přeskočení těchto fragmentů a vrácení omezeného objemu dat je přijatelnější než zrušení dotazu.

Pokud chcete přeskakovat fragmenty, určete pomocí nastavení ON *seznam-prostorů-dbspace* seznam prostorů dbspace s fragmenty, které má databázový server přeskakovat.

Určení, kdy přeskočit všechny fragmenty

Nastavení konfiguračního parametru DATASKIP na hodnotu ALL způsobí, že databázový server bude přeskakovat všechny nedostupné fragmenty. Tuto volbu používejte opatrně. Pokud se prostor dbspace stane nedostupným, mohou způsobit chybu všechny dotazy spuštěné aplikacemi, které před spuštěním dotazu nepoužily příkaz SET DATASKIP OFF.

Monitorování využití fragmentace

Pro administrátora databáze může být užitečné monitorovat následující aspekty fragmentace:

- Distribuce dat v jednotlivých fragmentech.
- Rozložení požadavků na vstup - výstup do jednotlivých fragmentů.
- Stav bloků obsahujících fragmenty.

Administrátor může monitorovat distribuci dat v jednotlivých fragmentech tabulky. Pokud je úkolem fragmentace zlepšit odezvu administrace, je důležité, aby data byla rovnoměrně rozložena. Abyste mohli monitorovat využití disků fragmenty, musíte monitorovat prostory tblspace databázového serveru, protože diskovou pamětovou jednotkou fragmentu je prostor

tblspace. (Informace o tom, jak monitorovat distribuci dat fragmentované tabulky naleznete v části “Monitorování prostorů tblspace a oblastí” na stránce 10-38.)

Administrátor musí monitorovat fronty požadavků na vstup - výstup dat uložených ve fragmentech. Pokud budou fronty požadavků na vstup - výstup nevyvážené, měl by administrátor spolupracovat s administrátorem DBA na vyladění strategie fragmentace. (Informace o tom, jak monitorovat fronty vstupu - výstupu každého bloku naleznete v části “Monitorování bloků” na stránce 10-34.)

Administrátor musí monitorovat dostupnost bloků a provést vhodné kroky v případě, že prostor dbspace obsahující jeden nebo více fragmentů selže. Informace o tom, jak zjistit, zda je blok vypnutý, naleznete v části “Monitorování bloků” na stránce 10-34.

Zobrazení databází

Vytvořené databáze můžete zobrazit pomocí následujících nástrojů:

- tabulky SMI
- ISA
- ON-Monitor

Použití tabulek SMI

Dotazem na tabulku **sysdatabases** můžete zobrazit jeden řádek pro každou databázi spravovanou databázovým serverem. Popis sloupců této tabulky naleznete v části týkající se tabulky **sysdatabases** v kapitole o databázi **sysmaster** v příručce *IBM Informix Administrator's Reference*.

Použití programu ISA

Dotaz na tabulku **sysdatabases** spusíte pomocí programu ISA podle následujících kroků:

1. Vyberte položku **SQL > Dotaz**.
2. Vyberte v seznamu **Databáze** položku **sysmaster**.
3. Zadejte následující příkaz a klepněte na tlačítko **Předat**:

```
select * from sysdatabases;
```

Použití programu ON-Monitor (systém UNIX)

Pokud chcete zjistit aktuální stav všech databází pomocí programu ON-Monitor, vyberte volbu **Status > Databases**. Program ON-Monitor může zobrazit nejvýše 100 databází. Pokud databázový server spravuje více než 100 databází, zobrazte jejich úplný seznam pomocí tabulek SMI, jak je popsáno v předchozí části.

Monitorování využití disku

Tato část popisuje různé metody sledování diskového prostoru používaného různými paměťovými jednotkami databázového serveru.

Podrobné informace o interních paměťových jednotkách databázového serveru zmiňovaných v této části naleznete v kapitole o diskových strukturách a paměti v příručce *IBM Informix Administrator's Reference*.

Monitorování bloků

Můžete monitorovat následující informace o blocích:

- Velikost bloku.
- Počet volných stránek.

- Tabulky v bloku.

Tyto informace umožňují sledovat využití diskového prostoru bloky, monitorovat aktivitu vstupu - výstupu jednotlivých bloků a kontrolovat fragmentaci.

onstat -d

Příkaz **onstat -d** zobrazí všechny prostory dbspace, blobspace a sbspace a následující informace o blocích v těchto prostorech.

- adresu bloku,
- číslo bloku a číslo přidruženého prostoru dbspace,
- posun vůči počátku zařízení (ve stránkách),
- velikost bloku (ve stránkách),
- počet volných stránek v zařízení,
- název cesty fyzického zařízení.

Pokud zadáte příkaz **onstat -d** v instanci s bloky prostorů blobspace, nebude zobrazený počet volných stránek aktuální. Znak vlnovka (~), který předchází hodnotě free, označuje, že toto číslo je pouze přibližné. Příkaz **onstat -d** neregistruje stránku blobpage jako dostupnou, dokud nedojde k zálohování logického protokolu, ve kterém se nachází záznam o odstranění, a dokud stránka není uvolněna. Pokud například odstraníte 25 jednoduchých velkých objektů a okamžitě spustíte příkaz **onstat -d**, nově uvolněný prostor nebude uveden ve výstupu příkazu **onstat**.

Pokud chcete získat přesný počet volných stránek blobpage v bloku prostoru blobspace, zadejte příkaz **onstat -d update**. Další informace naleznete v části "Příkaz onstat -d update" na stránce 10-35.

Ve výstupu příkazu **onstat -d update** poskytuje sloupec **flags** v části **chunk** následující informace:

- Zda se jedná o primární blok nebo o zrcadlený blok.
- Zda se blok nachází v režimu online, zda je vypnutý, zda právě probíhá jeho obnova nebo zda se jedná o nový blok.

Více informací o příkazu **onstat -d** a příkladu jeho výstupu naleznete v kapitole týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

Důležité: Než bude možné aktivovat zrcadlení, musí být nejprve provedeno zálohování úrovně 0. Totéž je třeba provést i po vypnutí zrcadlení.

Příkaz onstat -d update

Příkaz **onstat -d update** zobrazuje tytéž informace jako příkaz **onstat -d** a navíc zobrazuje přesný počet volných stránek blobpage každého bloku prostoru blobspace.

Příkaz onstat -D

Příkaz **onstat -D** zobrazuje tytéž informace jako příkaz **onstat -d** a navíc zobrazuje také počet stránek načtených z bloku (v poli **page Rd**).

Příkaz onstat -g iof

Příkaz **onstat -g iof** zobrazuje počet čtení z každého bloku a počet zápisů do každého bloku. Pokud je aktivita vstupu - výstupu nepřiměřená, může tento blok představovat problém v systému. Tato volba je užitečná k monitorování distribuce požadavků vstupu - výstupu mezi jednotlivé fragmenty fragmentované tabulky.

Více informací o příkazu **onstat -g iof** a příkladu jeho výstupu naleznete v kapitole týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

oncheck -pr

Databázový server ukládá informace o blocích v rezervovaných stránkách PAGE_1PCHUNK a PAGE_2PCHUNK.

Obsah rezervovaných stránek zobrazte pomocí příkazu **oncheck -pr**. Obrázek 10-1 znázorňuje ukázkou výstupu příkazu **oncheck -pr**. Výstup tohoto příkazu je v podstatě totožný s výstupem příkazu **onstat -d**, pokud však po posledním kontrolním bodu došlo ke změnám informací o blocích, tyto změny se ve výstupu příkazu **oncheck -pr** neprojeví.

```
Validating PAGE_1DBSP & PAGE_2DBSP...
    Using dbspace page PAGE_2DBSP.

    DBspace number          1
    DBspace name            rootdbs
    Flags                   0x20001          No mirror chunks
    Number of chunks        2
    First chunk              1
    Date/Time created        07/28/2006 14:46:55
    Partition table page number 14
    Logical Log Unique Id    0
    Logical Log Position     0
    Oldest Logical Log Unique Id 0
    Last Logical Log Unique Id 0
    Dbspace archive status   No archives have occurred
.
Validating PAGE_1PCHUNK & PAGE_2PCHUNK...
    Using primary chunk page PAGE_2PCHUNK.

    Chunk number            1
    Flags                   0x40           Chunk is online
    Chunk path               /home/server/root_chunk
    Chunk offset             0 (p)
    Chunk size               75000 (p)
    Number of free pages     40502
    DBspace number          1
.
.
.
```

Obrázek 10-1. Výstup příkazu oncheck -pr s informacemi o prostorech dbspace a blocích.

oncheck -pe

Pokud chcete zjistit fyzické rozvržení dat v bloku, spusíte příkaz **oncheck -pe**. Uvedeny jsou prostory dbspace, blobspace a sbspace. Obrázek 10-2 na stránce 10-37 znázorňuje ukázkou výstupu příkazu **oncheck -pe**.

Zobrazí se následující informace:

- Název, vlastník a datum vytvoření prostoru dbspace.
- Velikost bloku ve stránkách, počet použitých stránek a počet volných stránek.
- Seznam všech tabulek v bloku společně s číslem počáteční stránky a s délkou tabulky ve stránkách.

Tabulky v bloku jsou uvedeny sekvenčně. Tento výstup je užitečný při zjišťování fragmentace bloku. Pokud databázový server nemůže v bloku přidělit oblast, přestože je k dispozici odpovídající počet volných stránek, je pravděpodobné, že blok je značně fragmentovaný.

DBSpace Usage Report: rootdbs Owner: informix Created: 08/08/2006

Chunk Pathname	Size	Used	Free
1 /home/server/root_chunk	75000	19420	55580

Description	Offset	Size
RESERVED PAGES	0	12
CHUNK FREELIST PAGE	12	1
rootdbs:'informix'.TBLSpace	13	250
PHYSICAL LOG	263	1000
FREE	1263	1500
LOGICAL LOG: Log file 2	2763	1500
LOGICAL LOG: Log file 3	4263	1500
...		
sysmaster:'informix'.sysdatabases	10263	4
sysmaster:'informix'.systables	10267	8
...		

Chunk Pathname	Size	Used	Free
2 /home/server/dbspace1	5000	53	4947

Description	Offset	Size
RESERVED PAGES	0	2
CHUNK FREELIST PAGE	2	1
dbspace1:'informix'.TBLSpace	3	50
FREE	53	4947

Obrázek 10-2. Výstup příkazu `oncheck -pe`

Použití programu IBM Informix Server Administrator

Pomocí příkazů programu ISA můžete provádět následující úlohy:

- Kontrolovat rezervované stránky.
- Kontrolovat paměťové prostory.
- Přidávat prostory dbspace, dočasné prostory dbspace, prostory blobspace, dočasné prostory sbspace a prostory sbpace.
- Zobrazovat bloky a přidávat bloky k paměťovému prostoru.
- Kontrolovat stav funkce dataskip.
- Zobrazovat a přidávat externí prostory.
- Zobrazovat počet stránek v databázi, procento přiděleného prostoru a využitý prostor.
- Potlačit nastavení ONDBSPACEDOWN.

Použití programu ON-Monitor (systém UNIX)

Pomocí příkazů programu ON-Monitor můžete provádět následující úlohy.

Příkaz programu ON-Monitor Popis

Stav > Prostory	Zobrazuje informace o stavu paměťových prostorů a bloků.
Prostory dbspace > Vytvořit	Vytvoří prostor dbspace.
Prostory dbspace > Prostor blobspace	Vytvoří prostor blobspace.
Prostory dbspace > Zrcadlo	Přidá nebo vypustí zrcadlení paměťového prostoru.
Prostory dbspace > Informace	Zobrazuje informace o paměťových prostorech.
Prostory dbspace > Přidat blok	Přidá blok do paměťového prostoru.

Prostory dbspace > Parametr dataSkip

Spustí nebo zastaví funkci dataskip.

Prostory dbspace > Blok

Přidá blok do prostoru dbspace nebo blobspace.

Prostory dbspace > Vypustit

Vypustí prostor dbspace nebo blobspace.

Prostory dbspace > Stav

Změní stav zrcadlení bloku.

Použití tabulek SMI

Dotazem na tabulku **syschunks** můžete získat stav bloku. Důležité jsou následující sloupce.

sloupec	Popis
chknum	Počet bloků v prostoru dbspace.
dbsnun	Číslo prostoru dbspace.
chksize	Celková velikost bloku ve stránkách.
nfree	Počet volných stránek.
is_offline	Určuje, zda je blok vypnutý.
is_recovering	Určuje, zda právě probíhá obnova bloku.
mis_offline	Určuje, zda je zrcadlený blok vypnutý.
mis_recovering	Určuje, zda právě probíhá obnova zrcadleného bloku.

Tabulka **syschkio** obsahuje následující sloupce.

sloupec	Popis
pagesread	Počet stránek přečtených z bloku.
pageswritten	Počet stránek zapsaných do bloku.

Monitorování prostorů tblspace a oblastí

Monitorujte prostory tblspace a oblasti, abyste zjistili využití disku databázemi, tabulkami a fragmenty tabulek. Monitorování využití disku je zvláště důležité tehdy, pokud používáte fragmentaci tabulek a chcete, aby byla data tabulky a data indexu tabulky vhodným způsobem distribuována mezi jednotlivé fragmenty.

Informace o oblastech získáte spuštěním příkazu **oncheck -pt**. Příkaz **oncheck -pT** zobrazuje všechny informace, které zobrazuje příkaz **oncheck -pt** a navíc zobrazuje další informace o využití stránek a indexů.

Použití tabulek SMI

Dotazem na tabulku **systabnames** získáte informace o všech prostorech tblspace. Tabulka **systabnames** obsahuje sloupce, které ke každému prostoru tblspace udávají odpovídající tabulku, databázi a vlastníka.

Dotazem na tabulku **sysextents** získáte informace o všech oblastech. Sloupce tabulky **sysextents** označují databázi a tabulku, do které oblast patří a fyzickou adresu a velikost oblasti.

Monitorování jednoduchých velkých objektů v prostoru blobspace

Monitorujte prostory blobspace, abyste určili, jak veliký je dostupný prostor a zda je velikost stránky blobpage optimální.

Příkaz onstat -O

Příkaz **onstat -O** zobrazuje informace o prostoru blobspace k dočasnému ukládání velkých objektů a o mezipaměti podsystému Optical Subsystem. Zobrazované souhrnné údaje se v každé relaci přičítají. Databázový server opět nastaví souhrnné údaje na hodnotu 0 pouze při spuštění příkazu **onstat -z**.

Více informací o příkazu **onstat -O** a příkladu jeho výstupu naleznete v kapitole týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

První část výstupu popisuje následující souhrnné údaje o systémové vyrovnávací paměti.

sloupec	Popis
size	Velikost určená konfiguračním parametrem OPCACHEMAX.
alloc	Počet částí o velikosti 1 kB, které databázový server přiděluje vyrovnávací paměti.
avail	Nepoužitá část přidělené paměti (kilobajty).
number	Počet jednoduchých velkých objektů, které databázový server úspěšně uložil do mezipaměti, aniž by došlo k přetečení.
kbytes	Počet kilobajtů jednoduchých velkých objektů, které databázový server úspěšně uložil do mezipaměti, aniž by došlo k přetečení.
number	Počet jednoduchých velkých objektů, které databázový server zapsal do prostoru blobspace k dočasnému ukládání velkých objektů.
kbytes	Počet kilobajtů jednoduchých velkých objektů, které databázový server zapsal do prostoru blobspace k dočasnému ukládání velkých objektů.

Přestože položka **size** výstupu příkazu označuje množství paměti přidělené konfiguračním parametrem OPCACHEMAX, databázový server nepřidělí paměť určenou parametrem OPCACHEMAX, dokud ji nebude potřebovat. Proto položka **alloc** výstupu příkazu představuje pouze počet 1kB částí největšího dosud zpracovaného velkého objektu. Pokud se hodnoty polí **alloc** a **avail** rovnají, mezipaměť je prázdná.

Druhá část výstupu popisuje následující souhrnné údaje o uživatelské vyrovnávací paměti.

sloupec	Popis
SID	ID uživatelské relace
user	ID klientského uživatele
size	Velikost určená proměnnou prostředí INFORMIXOPCACHE , pokud je nastavena. Pokud proměnnou prostředí INFORMIXOPCACHE nenastavíte, použije databázový server velikost určenou konfiguračním parametrem OPCACHEMAX.
number	Počet jednoduchých velkých objektů, které databázový server uložil do mezipaměti, aniž by došlo k přetečení.
kbytes	Počet kilobajtů jednoduchých velkých objektů, které databázový server uložil do mezipaměti, aniž by došlo k přetečení.
number	Počet jednoduchých velkých objektů, které databázový server zapsal do prostoru blobspace k dočasnému ukládání velkých objektů.
kbytes	Velikost jednoduchých velkých objektů (v kilobajtech), které databázový server zapsal do prostoru blobspace k dočasnému ukládání velkých objektů.

Určení zaplnění stránek blobpage pomocí příkazu oncheck -pB

Příkaz **oncheck -pB** zobrazuje statistické údaje popisující průměrné zaplnění stránek blobpage. Pokud zjistíte, že statistické údaje udávají pro značný počet jednoduchých velkých objektů nízké procento zaplnění, databázový server by pravděpodobně zaznamenal zvýšení výkonu při změně velikosti stránky blobpage prostoru blobspace.

Spusťte příkaz **oncheck -pB** a jako parametr zadejte název databáze nebo tabulky. V následujícím příkladu jsou získány informace o všech jednoduchých velkých objektech uložených v tabulce **sriram.catalog** databáze **stores_demo**:

```
oncheck -pB stores_demo:sriram.catalog
```

Podrobné informace o interpretaci výstupu příkazu **oncheck -pB** naleznete v části týkající se optimalizace velikosti stránek blobpage prostoru blobspace v kapitole o posouzení výkonu tabulek v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Monitorování využití prostoru blobspace pomocí příkazu oncheck -pe

Příkaz **oncheck -pe** poskytuje informace o využití prostoru blobspace:

- Názvy tabulek, ve kterých jsou uložena data typu TEXT a BYTE, podle bloků.
- Počet použitých diskových stránek (*nikoli* stránek blobpage), podle tabulek.
- Počet volných diskových stránek, podle bloků.
- Počet použitých režijních stránek, podle bloků.

Obrázek 10-3 znázorňuje ukázkou výstupu příkazu **oncheck -pe**.

```
BLOBSpace Usage Report:  fstblob          Owner:  informix  Created: 03/01/07
      Chunk: 3  /home/server/blob_chunk          Size   Used   Free
                        4000      304   3696

      Disk usage for Chunk 3          Total Pages
      -----
      OVERHEAD                          8
      stores_demo:chrisw.catalog         296
      FREE                               3696
```

Obrázek 10-3. Výstup příkazu **oncheck -pe** znázorňující využití prostoru blobspace

Monitorování jednoduchých velkých objektů v prostoru dbspace pomocí příkazu oncheck -pT

Pomocí příkazu **oncheck -pT** můžete monitorovat počet stránek v prostorech dbspace využitý k uložení dat typu TEXT a BYTE.

Tento příkaz přijímá jako parametr název databáze nebo tabulky. Databázový server zobrazí pro každou tabulku v databázi nebo pro určenou tabulku souhrnnou sestavu o prostoru tbspace.

Po souhrnné sestavě následuje podrobný rozpis využití stránek oblasti podle typu stránky. Informace o datech typu TEXT and BYTE naleznete ve sloupci **Type**.

Databázový server může do jediné stránky blobpage uložit několik jednoduchých velkých objektů. Proto můžete zjistit počet stránek prostoru tbspace, ve kterých jsou uložena data typu TEXT a BYTE, nemůžete však určit počet velkých objektů v tabulce.

Obrázek 10-4 znázorňuje ukázkový výstup.

TBLSpace Usage Report for mydemo:chrisw.catalog

Type	Pages	Empty	Semi-Full	Full	Very-Full
Free	7				
Bit-Map	1				
Index	2				
Data (Home)	9				
Data (Remainder)	0	0	0	0	0
Tblspace BLOBs	5	0	0	1	4
Total Pages	24				

Unused Space Summary

Unused data bytes in Home pages	3564
Unused data bytes in Remainder pages	0
Unused bytes in Tblspace Blob pages	1430

Index Usage Report for index 111_16 on mydemo:chrisw.catalog

Level	Total	Average No. Keys	Average Free Bytes
1	1	74	1058
Total	1	74	1058

Index Usage Report for index 111_18 on mydemo:chrisw.catalog

Level	Total	Average No. Keys	Average Free Bytes
1	1	74	984
Total	1	74	984

Obrázek 10-4. Ukázka výstupu příkazu oncheck -pT s daty typu TEXT a BYTE

Monitorování prostorů sbspace

Jednou z nejdůležitějších oblastí k monitorování je využití stránek metadat. Při vytváření prostoru sbspace můžete určit velikost oblasti metadat. Dále můžete při každém přidání bloku k prostoru sbspace určit, zda bude obsahovat oblast metadat.

Pokud se pokusíte vložit nový inteligentní velký objekt, ale nebude k dispozici žádný prostor pro metadata, dojde k chybě. Aby k této situaci nedošlo, měl by administrátor monitorovat prostor dostupný pro metadata.

K monitorování prostorů sbspace použijte následující příkazy.

Příkaz	Popis
onstat -g smb s	<p>Zobrazí paměťové atributy všech prostorů sbspace v systému:</p> <ul style="list-style-type: none"> Název, příznaky a vlastníka prostoru sbspace. Stav protokolování. Průměrnou velikost inteligentního velkého objektu. Velikost první oblasti, velikost dalších oblastí a minimální velikost oblasti. Maximální přístupovou dobu vstupu - výstupu. Režim uzamykání.

Příkaz	Popis
onstat -g smb c	<p>Zobrazí následující vlastnosti všech bloků prostoru sbpace:</p> <ul style="list-style-type: none"> • Číslo bloku a název prostoru sbpace. • Velikost bloku a název cesty. • Celkový počet uživatelských datových stránek a počet volných uživatelských datových stránek. • Umístění a počet stránek ve všech oblastech uživatelských dat a oblastech metadat. <p>Další informace naleznete v části “Použití příkazu onstat -g smb c” na stránce 10-45.</p>
oncheck -ce oncheck -pe	<p>Zobrazí následující informace o využití prostorů sbpace:</p> <ul style="list-style-type: none"> • Názvy tabulek, ve kterých jsou uložena data inteligentních velkých objektů, podle bloků. • Počet použitých diskových stránek (nikoli stránek sbpage), podle tabulek. • Počet zbývajících volných stránek pro uživatelská data, podle bloků. • Počet zbývajících rezervovaných stránek pro uživatelská data, podle bloků. • Počet použitých stránek metadat, podle bloků. <p>Vystup poskytuje následující souhrnné údaje:</p> <ul style="list-style-type: none"> • Celkový počet použitých stránek ve všech oblastech uživatelských dat a v oblastech metadat. Systém přičítá k souhrnným údajům o oblasti uživatelských dat a oblasti metadat 53 stránek vyhrazené oblasti. • Počet zbývajících volných stránek v oblasti metadat. • Počet zbývajících volných stránek ve všech oblastech uživatelských dat. <p>Další informace naleznete v částech “Použití příkazů oncheck -ce a oncheck -pe” na stránce 10-43 a “Monitorování oblastí metadat a uživatelských dat” na stránce 10-45.</p>
onstat -d	<p>Zobrazí následující informace o blocích v každém prostoru sbpace:</p> <ul style="list-style-type: none"> • Počet volných stránek sbpage v každém bloku prostoru sbpace, v oblasti metadat a v oblastech uživatelských dat. • Celkový počet stránek sbpage v každém bloku prostoru sbpace, v oblasti metadat a v oblastech uživatelských dat. <p>Další informace naleznete v následující části “Použití příkazu onstat -d”.</p>
oncheck -cs oncheck -ps	<p>Ověří oblasti metadat prostorů sbpace a zobrazí informace o nich. Další informace naleznete v částech “Použití příkazu oncheck -cs” na stránce 10-44 a “Použití příkazu oncheck -ps” na stránce 10-44.</p>
oncheck -cS	<p>Zobrazí informace o oblastech inteligentních velkých objektů a o oblastech uživatelských dat v prostorech sbpace.</p>
oncheck -pS	<p>Zobrazí informace o oblastech inteligentních velkých objektů, o oblastech uživatelských dat a o oblastech metadat v prostorech sbpace. Další informace o příkazu oncheck -cS a volbě -pS naleznete v části týkající se správy prostorů sbpace v kapitole o posouzení výkonu tabulek v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i>.</p>

Použití příkazu onstat -d

Pomocí příkazu **onstat -d** můžete zobrazit následující informace o blocích v každém prostoru sbpace:

- Počet volných stránek sbpage v každém bloku prostoru sbpace, v oblasti metadat a v oblasti uživatelských dat.
- Celkový počet stránek sbpage v každém bloku prostoru sbpace, v oblasti metadat a v oblasti uživatelských dat.

Více informací o příkazu **onstat -d** a příkladu jeho výstupu naleznete v kapitole týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

Pokud chcete zjistit celkové množství využitého místa, spusíte příkaz **oncheck -pe**. Další informace naleznete v části “Použití příkazů oncheck -ce a oncheck -pe” na stránce 10-43.

Příkaz **onstat -d** neregistruje stránku sbpace jako dostupnou, dokud nedojde k zálohování logického protokolu, ve kterém se nachází záznam o odstranění, a dokud stránka není uvolněna. Pokud například odstraníte 25 inteligentních velkých objektů a okamžitě spustíte příkaz **onstat -d**, nebude nově uvolněný prostor uveden ve výstupu příkazu **onstat**.

Použití příkazů **oncheck -ce** a **oncheck -pe**

Pomocí příkazu **oncheck -ce** můžete zobrazit velikost všech bloků sbpace, celkové množství využitého prostoru a množství volného prostoru v oblasti uživatelských dat. Příkaz **oncheck -pe** zobrazuje tytéž informace jako příkaz **oncheck -ce** a navíc zobrazuje podrobný výpis využití bloků. Nejprve jsou uvedeny prostory dbspace a pak prostory sbpace. Výstup volby **-pe** poskytuje následující informace o využití prostorů sbpace:

- Názvy tabulek, ve kterých jsou uložena data inteligentních velkých objektů, podle bloků.
- Počet použitých diskových stránek (*nikoli* stránek sbpace), podle tabulek.
- Počet zbývajících volných stránek pro uživatelská data, podle bloků.
- Počet použitých stránek metadat, podle bloků.

Výstup poskytuje následující souhrnné údaje:

- Celkový počet použitých stránek v oblasti uživatelských dat, v oblasti metadat a v rezervované oblasti.
System přičítá k souhrnným údajům o oblasti uživatelských dat a oblasti metadat 53 stránek vyhrazené oblasti.
- Počet zbývajících volných stránek v oblasti metadat.
- Počet zbývajících volných stránek v oblasti uživatelských dat.

Rada: Příkaz **oncheck -pe** poskytuje informace o využití prostorů sbpace, přičemž jednotkou jsou stránky databázového serveru, nikoli stránky sbpace.

Obrázek 10-5 znázorňuje ukázkový výstup. V tomto příkladu se v prostoru sbpace s názvem **s9_sbspc** nachází celkem 214 využitých stránek, 60 volných stránek v oblasti metadat a 726 volných stránek v oblasti uživatelských dat.

Chunk Pathname	Size	Used	Free
2 /ix/ids9.2/./s9_sbspc	1000	940	60
Description	Offset	Size	
-----	-----	-----	
RESERVED PAGES	0	2	
CHUNK FREELIST PAGE	2	1	
s9_sbspc:'informix'.TBLSpace	3	50	
SBLOBSpace LO [2,2,1]	53	8	
SBLOBSpace LO [2,2,2]	61	1	
...			
SBLOBSpace LO [2,2,79]	168	1	
SBLOBSpace FREE USER DATA	169	305	
s9_sbspc:'informix'.sbspace_desc	474	4	
s9_sbspc:'informix'.chunk_adjunc	478	4	
s9_sbspc:'informix'.LO_hdr_partn	482	8	
s9_sbspc:'informix'.LO_ud_free	490	5	
s9_sbspc:'informix'.LO_hdr_partn	495	24	
FREE	519	60	
SBLOBSpace FREE USER DATA	579	421	
Total Used:	214		
Total SBLOBSpace FREE META DATA:	60		
Total SBLOBSpace FREE USER DATA:	726		

Obrázek 10-5. Výstup příkazu **oncheck -pe** znázorňující využití prostoru sbpace

Můžete použít CHECK EXTENTS jako *příkaz* jazyka SQL API, který je ekvivalentní s obslužným programem **oncheck -ce**. Další informace o použití příkazů jazyka SQL API naleznete v části “Vzdálená správa pomocí příkazů SQL” na stránce 26-10 a *IBM Informix Guide to SQL: Syntax*.

Použití příkazu oncheck -cs

Příkazy **oncheck -cs** a **oncheck -Cs** ověřují oblast metadat prostoru sbspace. Obrázek 10-6 znázorňuje příklad výstupu volby **-cs** pro prostor **s9_sbspc**. Pokud neurčíte název prostoru sbspace na příkazovém řádku, příkaz **oncheck** zkontroluje a zobrazí metadata všech prostorů sbspace.

Pomocí výstupu příkazu **oncheck -cs** můžete určit, kolik prostoru zbývá v oblasti metadat. Pokud je oblast zaplněná, přidejte další blok s oblastí metadat odpovídající velikosti. Celkový počet využitých stránek v oblasti metadat zjistíte sečtením čísel ve sloupci **Used**. Celkový počet volných stránek v oblasti metadat zjistíte sečtením čísel ve sloupci **Free**.

Obrázek 10-6 například znázorňuje celkový počet využitých stránek oblasti metadat prostoru **s9_sbspc** vypočtený na základě hodnot v polích jako 33 stránek o velikosti 2 kB (nebo 66 kilobajtů). Oblast metadat obsahuje celkem 62 volných stránek (nebo 124 kilobajtů).

```
Validating space 's9_sbspc' ...
```

SBL0Bspace Metadata Partition	Partnum	Used	Free
s9_sbspc:'informix'.TBLSpace	0x200001	6	44
s9_sbspc:'informix'.sbspace_desc	0x200002	2	2
s9_sbspc:'informix'.chunk_adjunc	0x200003	2	2
s9_sbspc:'informix'.LO_hdr_partn	0x200004	21	11
s9_sbspc:'informix'.LO_ud_free	0x200005	2	3

Obrázek 10-6. Výstup příkazu oncheck -cs

Použití příkazu oncheck -ps

Příkaz **oncheck -ps** ověří a zobrazí informace o oblastech metadat v oddílech prostoru sbspace. Obrázek 10-7 na stránce 10-45 znázorňuje příklad výstupu volby **-ps** pro prostor **s9_sbspc**. Pokud neurčíte název prostoru sbspace na příkazovém řádku, příkaz **oncheck** zkontroluje a zobrazí informace o prostorech tblspace všech paměťových prostorů.

Pokud chcete monitorovat množství dostupného prostoru metadat, spusťte následující příkaz:
`oncheck -ps název_prostoru`

Výstup volby **-ps** obsahuje informace o granularitě zámků, pole **partnum**, počet přidělených a použitých stránek, velikost oblasti a počet řádků v oblasti metadat. Pomocí výstupu příkazu **oncheck -ps** můžete určit, kolik prostoru zbývá v oblasti metadat. Pokud je oblast zaplněná, přidejte další blok s oblastí metadat odpovídající velikosti.

Spuštěním příkazu **oncheck -ps** pro prostor dbspace obsahující tabulky, ve kterých jsou uloženy inteligentní velké objekty můžete zjistit počet řádků tabulky.

Validating space 's9_sbspc' ...

```
TBLSpace Report for
TBLspace Flags          2801          Page Locking
                          TBLspace use 4 bit bit-maps
                          Permanent System TBLspace

Partition partnum       0x200001
Number of rows          92
Number of special columns 0
Number of keys          0
Number of extents      1
Current serial value    1
First extent size      50
Next extent size       50
Number of pages allocated 50
Number of pages used   6
Number of data pages   0
Number of rows         0
Partition lockid       2097153
Optical Cluster Partnum -1
Current SERIAL8 value  1
Current REFID value    1
Created                Thu May 24 14:14:33 2007
```

Obrázek 10-7. Výstup příkazu `oncheck -ps`

Monitorování oblastí metadat a uživatelských dat

Databázový server rezervuje 40 % oblasti uživatelských dat jako *rezervovanou oblast*.

Databázový server používá tento rezervovaný prostor k ukládání metadat nebo uživatelských dat. Využití oblast metadat se zvyšuje, pokud do prostoru sbspace přidáváte inteligentní velké objekty. Pokud databázový server zaplní prostor pro uživatelská data nebo metadata, přesune blok z rezervovaného prostoru do příslušné oblasti.

Pokud již byla použita celá rezervovaná oblast, nemůže databázový server přesunout prostor do oblasti metadat, přestože v oblasti uživatelských dat stále ještě může být dostupný volný prostor.

1. Pokud budete do prostoru sbspace přidávat inteligentní velké objekty, monitorujte prostor v oblasti metadat, v oblasti uživatelských dat a v rezervované oblasti pomocí příkazu **oncheck -pe** nebo **onstat -g smb c**. Příklad naleznete v části “Použití příkazů `oncheck -ce` a `oncheck -pe`” na stránce 10-43.
2. Pomocí protokolu zpráv sledujte přidělování rezervovaného prostoru metadatům.
Databázový server vypisuje zprávy o počtech stránek přidělených z rezervované oblasti do oblasti metadat.
3. K prostoru sbspace přidejte další blok dříve, než bude vyčerpán prostor v oblasti metadat a v rezervované oblasti.
Další informace naleznete v části “Přidání bloku do prostoru sbspace” na stránce 10-25.
4. Když databázový server přesouvá prostor z rezervované oblasti do oblasti metadat nebo uživatelských dat, zapisuje záznamy protokolu `FREE_RE` a `CHKADJUP`.

Další informace naleznete v části “Určení velikosti metadat prostoru sbspace” na stránce 10-25.

Použití příkazu `onstat -g smb c`

Pomocí příkazu `onstat -g smb c` můžete monitorovat množství volného prostoru ve všech blocích prostorů sbspace a velikosti oblasti uživatelských dat, oblasti metadat a vyhrazené oblasti ve stránkách.

Výstup tohoto příkazu zobrazí počet použitých stránek **usr pgs**) a počet volných stránek (**free pg**) v bloku prostoru sbspace. Oblast metadat **Md** zahrnuje informace zobrazující posun počáteční stránky a počet stránek.

Zavedení dat do tabulky

Data můžete do existující tabulky zavést následujícími způsoby.

Metody zavádění dat	Data typu TEXT a BYTE	Data typu CLOB a BLOB	Odkaz
Příkaz LOAD programu DB–Access	Ano	Ano	Příkaz load LOAD v příručce <i>IBM Informix Guide to SQL: Syntax</i>
Obslužný program dbload	Ano	Ano	<i>IBM Informix Migration Guide</i>
Obslužný program dbimport	Ano	Ano	<i>IBM Informix Migration Guide</i>
Programy v jazyku ESQL/C	Ano	Ano	<i>IBM Informix ESQL/C Programmer's Manual</i>
Obslužný program onload	Ne	Ne	<i>IBM Informix Migration Guide</i>
Obslužný program onpladm	Ano, v režimu deluxe	Ano, v režimu deluxe	IBM Informix Server Administrator
High-Performance Loader (zavaděč HPL)	Ano, v režimu deluxe	Ano, v režimu deluxe	<i>IBM Informix High-Performance Loader User's Guide</i>

Důležité: Databázový server neobsahuje mechanismy ke kompresi dat typu TEXT a BYTE po jejich zavedení do databáze.

Část 3. Protokolování a administrace protokolů

Kapitola 11. Protokolování

Obsah kapitoly	11-1
Procesy databázového serveru, které vyžadují protokolování	11-1
Protokolování transakcí	11-3
Protokolování příkazů jazyka SQL a aktivity databázového serveru	11-3
Aktivita, která se vždy protokoluje	11-3
Aktivity protokolované u databázi s protokolováním transakcí	11-4
Aktivita, která se neprotokoluje	11-4
Stav protokolování databáze	11-5
Protokolování transakcí bez vyrovnávací paměti	11-5
Protokolování transakcí s vyrovnávací pamětí	11-6
Protokolování transakcí kompatibilní se standardem ANSI	11-6
Databáze bez protokolování	11-6
Databáze s různými stavy používání vyrovnávací paměti pro protokolování	11-7
Protokolování databáze v prostředí X/Open DTP	11-7
Nastavení nebo změny stavu nebo režimu protokolování	11-7

Obsah kapitoly

Tato kapitola popisuje protokolování databází Informix a zabývá se následujícími otázkami:

- Které procesy databázového serveru vyžadují protokolování?
- Co je protokolování transakcí?
- Jaká aktivita databázového serveru je protokolována?
- Co je stav protokolování databáze?
- Kdo může nastavit nebo změnit stav protokolování databáze?

Všechny databáze spravované jedinou instancí databázového serveru ukládají záznamy protokolu do téhož logického protokolu bez ohledu na to, zda používají protokolování transakcí. Většinu uživatelů databáze může zajímat, zda se jedná o protokolování transakcí s vyrovnávací pamětí nebo zda tabulka používá protokolování.

Chcete-li změnit stav protokolování databáze, prostudujte část “Nastavení nebo změny stavu nebo režimu protokolování” na stránce 11-7.

Procesy databázového serveru, které vyžadují protokolování

Server Dynamic Server při své práci (zpracování transakcí, zaznamenávání uložení dat a zajištění konzistence dat) automaticky generuje *záznamy logického protokolu* pro některé akce, které podniká. Většinu času tyto záznamy logického protokolu databázový server dále nepoužívá, když ovšem například potřebuje odvolat transakci nebo provést rychlou obnovu po selhání systému, jsou záznamy logického protokolu životně důležité. Záznamy logického protokolu jsou klíčové pro mechanismy obnovy dat.

Databázový server ukládá záznamy logického protokolu do *logického protokolu*. Logický protokol je tvořen *soubory logického protokolu*, které databázový server spravuje na disku, dokud nejsou bezpečně převedeny do režimu offline (to znamená *zálohovány*). Administrátor databázového serveru uchovává zálohované soubory logického protokolu, dokud nejsou potřeba během obnovy dat nebo dokud se administrátor nerozhodne, že již nejsou pro obnovu dat dále potřeba. Další informace o logických protokolech uvádí Kapitola 13, “Logický protokol”, na stránce 13-1.

Záznamy logického protokolu mají proměnlivou délku. Toto uspořádání zvětšuje počet záznamů logického protokolu, které je možné zapsat na stránku ve vyrovnávací paměti logického protokolu. Databázový server ovšem často vyprázdní vyrovnávací paměť logického protokolu dříve, než je stránka zaplněna. Další informace o záznamech logického protokolu naleznete v kapitole zabývající se interpretací záznamů logického protokolu v příručce *IBM Informix Administrator's Reference*.

Databázový server používá záznamy logického protokolu při provádění různých funkcí, které obnovují data a zajišťují konzistenci dat:

- **Odvolání transakce.** Pokud databáze používá protokolování transakcí a transakce musí být odvolána, použije databázový server záznamy logického protokolu ke zrušení změn, které byly provedeny během transakce. Další informace naleznete v části “Protokolování transakcí” na stránce 11-3.
- **Rychlá obnova.** Pokud se databázový server vypne neřízeným způsobem, použije databázový server záznamy logického protokolu k obnově všech transakcí, které se vyskytly od nejstarší aktualizace dosud nevyprázdněné na disk a k odvolání jakékoli nepotvrzené transakce. (Když jsou všechna data ve sdílené paměti na disku stejná, jsou *fyziicky konzistentní*.) Databázový server používá záznamy logického protokolu během rychlé obnovy, když vrací celou databázi do stavu logické konzistence, k bodu nejnovějšího záznamu logického protokolu. (Další informace naleznete v části “Rychlá obnova po kontrolním bodu” na stránce 15-8.)
- **Obnova dat.** Databázový server používá většinu nedávných záloh paměťových prostorů a logických protokolu ke znovu vytvoření systému databázového serveru v bodu nejnovějšího zálohovaného záznamu logického protokolu. Logická obnova se vztahuje na všechny záznamy protokolu od poslední zálohy paměťových prostorů.
- **Odložená kontrola.** Pokud transakce používá příkaz SET CONSTRAINTS k nastavení kontroly na hodnotu DEFERRED, databázový server nekontroluje omezení, dokud není transakce potvrzena. Jestliže se vyskytne chyba omezení, zatímco je transakce potvrzována, použije databázový server záznamy logického protokolu k odvolání transakce. Další informace naleznete v části SET Database Object Mode v příručce *IBM Informix Guide to SQL: Syntax*.
- **Vrstvení odstranění.** Vrstvení odstranění u referenčních omezení používá záznamy logického protokolu k tomu, aby bylo zajištěno, že transakce bude možné odvolat zpět, pokud bude odstraněn nadřazený řádek a systém selže před odstraněním podřízených řádků. Další informace o dědičnosti tabulek naleznete v příručce *IBM Informix Database Design and Implementation Guide*. Další informace o omezeních pomocí primárního klíče a cizího klíče naleznete v příručce *IBM Informix Guide to SQL: Tutorial*.
- **Distribuované transakce.** Každý databázový server zapojený do distribuovaných transakcí uchovává záznamy logického protokolu o transakci. Tento proces zajišťuje integritu a konzistenci dat i v případě, že nastane selhání jednoho z databázových serverů, které provádějí transakci. Další informace naleznete v části “Dvoufázové potvrzování a záznamy logického protokolu” na stránce 24-14.
- **Replikace dat.** Prostředí replikace dat, která používají sekundární servery replikace HDR, sekundární servery SD a sekundární servery RS, využívají záznamy logického protokolu k udržování konzistence dat na primárních a sekundárních databázových serverech, takže jeden z těchto databázových serverů může být rychle použit jako záložní databázový server, pokud dojde k poruše druhého databázového serveru. Další informace naleznete v části “Způsob činnosti replikace dat” na stránce 19-6.
- **Enterprise Replication.** Při použití replikace Enterprise Replication musíte používat protokolování transakcí, neboť replikuje data ze záznamů logického protokolu. Další informace naleznete v příručce *IBM Informix Dynamic Server Enterprise Replication Guide*.

Protokolování transakcí

O databázi nebo tabulce se říká, že *má* nebo *používá* protokolování transakcí, když příkazy SQL pro manipulaci s daty generují záznamy logického protokolu.

Stav protokolování transakcí označuje, zda databáze používá protokolování transakcí. *Režim používání vyrovnávací paměti protokolu* označuje, zda databáze používá protokolování s vyrovnávací pamětí nebo bez vyrovnávací paměti, nebo zda se jedná o protokolování kompatibilní se standardem ANSI. Další informace naleznete v částech “Stav protokolování databáze” na stránce 11-5 a Kapitola 12, “Správa režimu protokolování databáze”, na stránce 12-1.

Když vytváříte databázi, určujete, zda bude databáze používat *protokolování transakcí* a případně i jaký mechanismus používání vyrovnávací paměti protokolu bude používat. Po vytvoření databáze můžete vypnout protokolování databáze nebo jej změnit například na protokolování s vyrovnávací pamětí. I když vypnete protokolování transakcí pro všechny databáze, bude databázový server protokolovat některé události. Další informace naleznete v částech “Aktivita, která se vždy protokoluje” na stránce 11-3 a “Protokolování databáze v prostředí X/Open DTP” na stránce 11-7.

V rámci databáze můžete používat tabulky s protokolováním a tabulky bez protokolování. Uživatel, který vytváří tabulku, určuje typ tabulky. I když používáte tabulky bez protokolování, bude databázový server protokolovat některé události. Další informace naleznete v části “Typy tabulek serveru Dynamic Server” na stránce 9-24.

Protokolování příkazů jazyka SQL a aktivity databázového serveru

V databázovém serveru jsou možné tři typy protokolovaných činností:

- Aktivita, která se vždy protokoluje.
- Aktivita, která se protokoluje pouze v databázích používajících protokolování transakcí.
- Aktivita, která se nikdy neprotokoluje.

Aktivita, která se vždy protokoluje

Některé databázové operace vždy generují záznamy logického protokolu, i když vypnete protokolování transakcí nebo používáte tabulky bez protokolování.

U trvalých tabulek jsou vždy protokolovány tyto operace:

- Určité příkazy SQL včetně příkazů SQL, které definují data
- Zálohování paměťových prostorů.
- Kontrolní body.
- Administrativní změny konfigurace databázového serveru, jako například přidání bloku nebo prostoru dbspace.
- Přidělení nových oblastí k tabulkám.
- Změny stavu protokolování databáze.
- Operace s inteligentními velkými objekty:
 - vytvoření
 - odstranění
 - přidělení a zrušení přidělení oblastí
 - zkrácení
 - spojení a rozdělení stránek seznamu volných stránek v bloku.
 - Změna záhlaví LO a referenčního počtu LO.

- Metadata prostoru sbspace.
- Prostory blobspace.

Níže uvedená tabulka obsahuje příkazy generující operace, které jsou protokolovány.

ALTER ACCESS_METHOD	CREATE ROUTINE	DROP SEQUENCE
ALTER FRAGMENT	CREATE ROUTINE FROM	DROP SYNONYM
ALTER FUNCTION	CREATE ROW TYPE	DROP TABLE
ALTER INDEX	CREATE SCHEMA	DROP TRIGGER
ALTER PROCEDURE	CREATE SEQUENCE	DROP TYPE
ALTER ROUTINE	CREATE SYNONYM	DROP VIEW
ALTER SEQUENCE	CREATE TABLE	CREATE XADATASOURCE
ALTER TABLE	CREATE Temporary TABLE	DROP XADATASOURCE TYPE
CLOSE DATABASE	CREATE TRIGGER	GRANT
CREATE ACCESS_METHOD	CREATE VIEW	RENAME COLUMN
CREATE AGGREGATE	CREATE XADATASOURCE	RENAME DATABASE
CREATE CAST	DROP ACCESS_METHOD	RENAME INDEX
CREATE DATABASE	DROP AGGREGATE	RENAME SEQUENCE
CREATE DISTINCT TYPE	DROP CAST	RENAME TABLE
CREATE EXTERNAL TABLE	DROP DATABASE	REVOKE
CREATE FUNCTION	DROP FUNCTION	REVOKE FRAGMENT
CREATE FUNCTION FROM	DROP INDEX	
CREATE INDEX	DROP OPCLASS	
CREATE OPAQUE TYPE	DROP PROCEDURE	
CREATE OPCLASS	DROP ROLE	
CREATE PROCEDURE	DROP ROUTINE	
CREATE PROCEDURE FROM	DROP ROW TYPE	
CREATE ROLE	DROP XADATASOURCE TYPE	

Aktivity protokolované u databází s protokolováním transakcí

Pokud databáze používá protokolování transakcí, generují následující příkazy jazyka SQL jeden nebo více záznamů protokolu. Pokud jsou tyto záznamy odvolány, generuje odvolání transakcí také záznamy protokolu.

DELETE	LOAD	UNLOAD
FLUSH	PUT	UPDATE
INSERT	SELECT INTO TEMP	

Následující příkazy jazyka SQL generují záznamy protokolu ve zvláštních situacích:

Příkaz SQL	Záznam protokolu, který je generován příkazem
BEGIN WORK	Vrátí chybu, pokud databáze nepoužívá protokolování transakcí. Záznam protokolu je vytvořen, pokud transakce provádí nějakou jinou činnost, která je protokolována.
COMMIT WORK	Vrátí chybu, pokud databáze nepoužívá protokolování transakcí. Záznam protokolu je vytvořen, pokud transakce provádí nějakou jinou činnost, která je protokolována.
EXECUTE	Skutečnost, zda tento příkaz generuje záznam protokolu, závisí na vykonávaném příkazu.
EXECUTE FUNCTION	Skutečnost, zda tento příkaz generuje záznam protokolu, závisí na vykonávané funkci.
EXECUTE IMMEDIATE	Skutečnost, zda tento příkaz generuje záznam protokolu, závisí na vykonávaném příkazu.
EXECUTE PROCEDURE	Skutečnost, zda tento příkaz generuje záznam protokolu, závisí na vykonávané proceduře.

Aktivita, která se neprotokuluje

Následující příkazy jazyka SQL nevytvářejí záznamy protokolu, bez ohledu na režim protokolování databáze.

ALLOCATE COLLECTION	DEALLOCATE ROW	LOCK TABLE
ALLOCATE DESCRIPTOR	DECLARE	OPEN
ALLOCATE ROW	DESCRIBE	OUTPUT
CLOSE	DISCONNECT	PREPARE
CONNECT	FETCH	SELECT
DATABASE	FREE	SET
DEALLOCATE COLLECTION	GET DESCRIPTOR	UNLOCK TABLE
DEALLOCATE DESCRIPTOR	GET DIAGNOSTICS	WHENEVER
DEALLOCATE ROW	INFO	

U dočasných tabulek v dočasných prostorech dbspace není protokolováno nic ani příkazy jazyka SQL uvedené v části “Aktivita, která se vždy protokoluje” na stránce 11-3. Pokud zahrnete dočasné prostory dbspace (neprovádějící protokolování) do parametru DBSPACETEMP, databázový server umístí tabulky neprovádějící protokolování nejprve do těchto dočasných prostorů dbspace. Další informace naleznete v části “Dočasné tabulky” na stránce 9-33.

Stav protokolování databáze

Abyste mohli využít všechny vlastnosti uvedené v části “Procesy databázového serveru, které vyžadují protokolování” na stránce 11-1, musíte v databázi používat protokolování transakcí.

Každá databáze, kterou spravuje databázový server, má nastaven stav protokolování. Stav protokolování označuje, zda databáze používá protokolování transakcí, a pokud ano, který mechanismus použití vyrovnávací paměti využívá. Ke zjištění stavu protokolování transakcí databáze použijte obslužné programy databázového serveru podle informací uvedených v části “Monitorování režimu protokolování databáze” na stránce 12-6. Stav protokolování transakce označuje libovolný z následujících typů protokolování:

- protokolování transakcí bez vyrovnávací paměti
- protokolování transakcí s vyrovnávací paměti
- protokolování transakcí kompatibilní se standardem ANSI
- bez protokolování

Všechny záznamy logického protokolu procházejí vyrovnávací paměti logického protokolu ve sdílené paměti, než je databázový server zapíše do logického protokolu na disk. Bod, ve kterém databázový server vyprázdní vyrovnávací paměť logického protokolu, je ovšem jiný pro protokolování transakcí s vyrovnávací paměti a pro protokolování transakcí bez vyrovnávací paměti. Další informace naleznete v částech Obrázek 7-1 na stránce 7-3 a “Vyprazdňování vyrovnávací paměti logického protokolu” na stránce 7-29.

Protokolování transakcí bez vyrovnávací paměti

Pokud jsou v databázi prováděny transakce, které používají protokolování bez vyrovnávací paměti, je zaručen zápis záznamů z vyrovnávací paměti logického protokolu na disk při zpracování potvrzení. Když se po příkazu COMMIT (a před příkazem PREPARE u distribuovaných transakcí) vrátí řízení aplikaci, jsou již záznamy logického protokolu uloženy na disku. Databázový server vyprázdní záznamy okamžitě po potvrzení jakékoli transakce ve vyrovnávací paměti (tedy jakmile je do vyrovnávací paměti logického protokolu zapsán záznam o potvrzení).

Když databázový server vyprazdňuje vyrovnávací paměť, zapisují se na disk pouze použité stránky. Použité stránky zahrnují stránky, které jsou pouze částečně zaplněny, takto ovšem

část prostoru zůstává nevyužita. Z tohoto důvodu se soubory logického protokolu na disku zaplňují rychleji, než kdyby všechny databáze v databázovém serveru používaly protokolování s vyrovnávací pamětí.

Protokolování bez vyrovnávací paměti je nejlepší volbou pro většinu databází, neboť zaručuje, že bude možné obnovit všechny potvrzené transakce. V případě selhání budou ztraceny pouze transakce, které v okamžiku selhání nebyly potvrzeny. Při protokolování bez vyrovnávací paměti vyprazdňuje databázový server vyrovnávací paměť logického protokolu na disk častěji a vyrovnávací paměť obsahuje mnoho pouze částečně zaplněných stránek, takže je logický protokol zaplněn rychleji než v případě protokolování s vyrovnávací pamětí.

Protokolování transakcí s vyrovnávací pamětí

Pokud jsou v databázi, která používá protokolování s vyrovnávací pamětí, prováděny transakce, budou záznamy co možná nejdéle zadržovány (*uchovávány ve vyrovnávací paměti*) ve vyrovnávací paměti logického protokolu. Nebudou vyprázdněny z vyrovnávací paměti logického protokolu ve sdílené paměti do logického protokolu na disku, dokud nenastane jedna z níže uvedených situací:

- Vyrovnávací paměť je zaplněna.
- Potvrzení v databázi s protokolováním bez vyrovnávací paměti vyprázdní vyrovnávací paměť.
- Vyskytne se kontrolní bod.
- Připojení je ukončeno.

Pokud používáte protokolování s vyrovnávací pamětí a dojde k selhání, nelze očekávat, že databázový server bude schopen obnovit transakce, které byly ve vyrovnávací paměti logického protokolu v okamžiku, kdy došlo k poruše. Z tohoto důvodu může dojít i ke ztrátě některých potvrzených transakcí. Pokud však podstoupíte toto riziko, dojde k určitému zvýšení výkonu při změnách databáze. Protokolování s vyrovnávací pamětí je nejvhodnější pro databáze, které jsou aktualizovány často (pokud je rychlost aktualizací důležitá), v případě, že jste schopni v případě selhání aktualizace provést znovu. Můžete vyladit velikost vyrovnávací paměti logického protokolu, abyste pro váš systém našli přijatelnou rovnováhu mezi výkonem a rizikem ztráty transakcí při selhání systému.

Protokolování transakcí kompatibilní se standardem ANSI

Stav protokolování databáze kompatibilní se standardem ANSI označuje, že vlastník databáze vytvořil tuto databázi pomocí klíčových slov `MODE ANSI`. Databáze kompatibilní se standardem ANSI vždy používají protokolování transakcí bez vyrovnávací paměti a pro zpracování transakcí vynucují pravidla standardu ANSI. U databází kompatibilních se standardem ANSI nelze změnit stav používání vyrovnávací paměti.

Databáze bez protokolování

Pokud vypnete v databázi protokolování, nebudou se transakce protokolovat, ale ostatní operace protokolovány budou. Další informace naleznete v části “Aktivita, která se vždy protokoluje” na stránce 11-3. Obvykle je vhodné vypnout protokolování databáze, když zavádíte data nebo pokud provádíte pouze dotazy.

Pokud vám vyhovuje zdroj, ze kterého můžete provádět obnovu, můžete se rozhodnout, že v databázi nebudete používat protokolování transakcí, a snížit tak objem zpracování v databázovém serveru. Pokud například zavádíte do databáze mnoho řádků z obnovitelného zdroje, jako například pásky nebo souboru ASCII, nebudete pravděpodobně potřebovat protokolování transakcí a zavádění bude bez něj zpracováno rychleji. Jestliže ovšem v

databázi budou aktivní další uživatelé, nebudete mít záznamy logického protokolu jejich transakcí, dokud znovu nespustíte protokolování, které musí čekat na provedení zálohy úrovně 0.

Databáze s různými stavy používání vyrovnávací paměti pro protokolování

Všechny databáze v databázovém serveru používají tentýž logický protokol a tytéž vyrovnávací paměti logického protokolu. Proto mohou být transakce v databázích s různými stavy používání vyrovnávací paměti protokolu zapisovány do téže vyrovnávací paměti logického protokolu. V takovém případě, pokud jsou prováděny transakce v databázích s protokolováním s vyrovnávací pamětí *i* v databázích s protokolováním bez vyrovnávací paměti, bude databázový server vyprazdňovat vyrovnávací paměť v případě zaplnění *nebo* při dokončení transakce v databázi s protokolováním bez vyrovnávací paměti.

Protokolování databáze v prostředí X/Open DTP

Databáze v prostředí X/Open DTP (Distributed transaction processing) musí používat *protokolování bez vyrovnávací paměti*. Protokolování bez vyrovnávací paměti zajišťuje, že logické protokoly databázového serveru budou vždy v konzistentním stavu a mohou být synchronizovány se správcem transakcí. Pokud je v prostředí X/Open DTP otevřena databáze vytvořená s protokolováním s vyrovnávací pamětí, stav databáze se automaticky změní na protokolování bez vyrovnávací paměti. Databázový server podporuje databáze kompatibilní se standardem ANSI i databáze, které nejsou kompatibilní se standardem ANSI. Další informace naleznete v části “Správci transakcí” na stránce 24-2.

Nastavení nebo změny stavu nebo režimu protokolování

Stav protokolování a režim používání vyrovnávací nastavuje uživatel, který tuto databázi vytváří pomocí příkazu CREATE DATABASE. Další informace o příkazu CREATE DATABASE naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

Pokud není v příkazu CREATE DATABASE určen stav protokolování, vytvoří se databáze bez protokolování.

Stav protokolování může změnit pouze administrátor databázového serveru. Toto téma popisuje Kapitola 12, “Správa režimu protokolování databáze”, na stránce 12-1. Běžní koncoví uživatelé nemohou měnit stav protokolování databáze.

Pokud databáze nepoužívá protokolování, nemusíte zvažovat, zda je vhodnější protokolování s vyrovnávací pamětí nebo bez vyrovnávací paměti. Pokud zadáte protokolování databáze, ale neurčíte režim použití vyrovnávací paměti, bude výchozím režimem protokolování bez vyrovnávací paměti.

Koncoví uživatelé *mohou* na dobu omezenou *trváním relace* přepínat z protokolování bez vyrovnávací paměti na protokolování s vyrovnávací pamětí (avšak ne na protokolování kompatibilní se standardem ANSI), a naopak z protokolování s vyrovnávací pamětí na protokolování bez vyrovnávací paměti. V rámci aplikace provádí tuto změnu příkaz SET LOG. Další informace o příkazu SET LOG naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

Kapitola 12. Správa režimu protokolování databáze

Obsah kapitoly	12-1
Změna režimu protokolování databáze	12-2
Změna režimu protokolování databáze pomocí obslužného programu ondblog	12-3
Změna režimu protokolování pomocí obslužného programu ondblog	12-3
Zrušení změny režimu protokolování pomocí obslužného programu ondblog	12-3
Ukončení protokolování pomocí obslužného příkazu ondblog	12-3
Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ondblog	12-3
Změna režimu protokolování databáze kompatibilní se standardem ANSI	12-3
Změna režimu protokolování databáze pomocí obslužného programu ontape	12-4
Zapnutí protokolování transakcí pomocí obslužného programu ontape	12-4
Ukončení protokolování pomocí obslužného programu ontape	12-4
Změna režimu protokolování pomocí obslužného programu ontape	12-4
Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ontape	12-5
Změna režimu protokolování databáze pomocí programu ISA	12-5
Změna režimu protokolování pomocí programu ON-Monitor (UNIX)	12-5
Změna režimu protokolování tabulek	12-5
Změna tabulky za účelem vypnutí protokolování	12-5
Změna tabulky za účelem zapnutí protokolování	12-6
Zakázání protokolování v dočasných tabulkách	12-6
Monitorování transakcí	12-6
Monitorování režimu protokolování databáze	12-6
Monitorování režimu protokolování u tabulek typu SMI	12-6
Monitorování režimu protokolování pomocí programu ON-Monitor (UNIX)	12-6
Monitorování režimu protokolování pomocí programu ISA	12-7

Obsah kapitoly

Tato kapitola pokrývá následující témata zabývající se změnami režimu protokolování databáze:

- Jak chápat režim protokolování databáze
- Změna režimu protokolování databáze pomocí obslužného programu **ondblog**
- Změna režimu protokolování databáze pomocí obslužného programu **ontape**
- Změna režimu protokolování databáze pomocí programu **ON-Monitor**
- Monitorování protokolování transakcí

Jako administrátor databázového serveru můžete následujícím způsobem měnit režim protokolování databáze:

- Změnit protokolování transakcí z režimu s vyrovnávací pamětí na režim bez vyrovnávací paměti.
- Změnit protokolování transakcí z režimu bez vyrovnávací paměti na režim s vyrovnávací pamětí.
- Zajistit, aby byla databáze kompatibilní se standardem ANSI.
- Přidat k databázi protokolování transakcí (s vyrovnávací pamětí nebo bez vyrovnávací paměti).
- Ukončit protokolování transakcí pro databázi.

Další informace o režimu protokolování databázi, o tom, kdy je vhodné používat protokolování transakcí a kdy je vhodné pro protokolování transakcí používat vyrovnávací

paměť, uvádí Kapitola 11, “Protokolování”, na stránce 11-1. Informace o tom, jak zjistit aktuální režim protokolování databáze, uvádí “Monitorování režimu protokolování databáze” na stránce 12-6.

Informace o použití administrativních příkazů rozhraní SQL API namísto některých příkazů **onblog** a **ontape** naleznete v příručce “Vzdálená správa pomocí příkazů SQL” na stránce 26-10 a v příručce *IBM Informix Guide to SQL: Syntax*.

Změna režimu protokolování databáze

Protokolování je možné změnit nebo přidat pomocí obslužného programu **ondblog**, **ontape**, nebo programu ISA. Potom pomocí ON-Bar nebo obslužného programu **ontape** proveďte zálohování dat. Při použití ON-Bar nebo obslužného programu **ontape** musí být databázový server online, v režimu administrace nebo v klidovém režimu.

Informace o programu ON-Bar a obslužném programu **ontape** naleznete v příručce *IBM Informix Backup and Restore Guide*.

Tabulka 12-1 znázorňuje, jak může administrátor databázového serveru změnit režim protokolování databáze. Některé změny režimu protokolování vstoupí v platnost okamžitě, zatímco jiné změny vyžadují zálohování úrovně 0.

Tabulka 12-1. Přejechy režimu protokolování

Převést z:	Převést na bez protokolování	Převést na protokolování bez vyrovnávací paměti	Převést na protokolování s vyrovnávací paměti	Převést na kompatibilní s ANSI
Bez protokolování	Nelze použít	Zálohování úrovně 0 (ovlivněných paměťových prostorů)	Zálohování úrovně 0 (ovlivněných paměťových prostorů)	Zálohování úrovně 0 (ovlivněných paměťových prostorů)
Protokolování bez vyrovnávací paměti	Ano	Nelze použít	Ano	Ano
Protokolování s vyrovnávací paměti	Ano	Ano	Nelze použít	Ano
Kompatibilní s ANSI	Neplatné	Neplatné	Neplatné	Nelze použít

Níže jsou uvedeny některé obecné poznatky týkající se změny režimu protokolování databáze:

- Zatímco probíhá změna stavu protokolování, databázový server uzamkne databázi výlučným zámekem, aby zabránil ostatním uživatelům v přístupu k databázi, a tento zámek uvolní po dokončení změny.
- Pokud dojde během změny režimu protokolování k selhání, zkontrolujte po obnově dat databázového serveru režim protokolování pomocí programu ISA nebo příznaků v tabulce **sysdatabases** v databázi **sysmaster**. Další informace naleznete v části “Monitorování režimu protokolování databáze” na stránce 12-6. Potom se pokuste změnit režim protokolování znovu.
- Poté, co zvolíte protokolování s vyrovnávací paměti nebo protokolování bez vyrovnávací paměti, může aplikace pomocí příkazu SQL SET LOG provést změnu z jednoho režimu protokolování na jiný. Tato změna je platná po dobu relace. Další informace o příkazu SET LOG uvádí příručka *IBM Informix Guide to SQL: Syntax*.
- Pokud do databáze přidáte protokolování, nebude změna dokončena, dokud neprovedete zálohování úrovně 0 všech paměťových prostorů databáze.

Změna režimu protokolování databáze pomocí obslužného programu ondblog

Obslužný program **ondblog** můžete použít ke změně režimu protokolování pro jednu nebo více databází. Pokud do databáze přidáte protokolování, musíte vytvořit zálohu úrovně 0 prostorů dbspace, které obsahují databázi, než změna vstoupí v platnost. Další informace naleznete v části týkající se používání obslužného programu **ondblog** v příručce *IBM Informix Administrator's Reference*.

Změna režimu protokolování pomocí obslužného programu ondblog

Chcete-li změnit režim protokolování z režimu s vyrovnávací pamětí na režim bez vyrovnávací paměti v databázi nazvané **stores_demo**, spusíte následující příkaz:

```
ondblog unbuf stores_demo
```

Chcete-li změnit režim použití vyrovnávací paměti z režimu bez vyrovnávací paměti na režim s vyrovnávací pamětí v databázi nazvané **stores_demo**, spusíte tento příkaz:

```
ondblog buf stores_demo
```

Zrušení změny režimu protokolování pomocí obslužného programu ondblog

Pokud chcete zrušit změnu režimu protokolování, než bude vytvořena další záloha úrovně 0, spusíte následující příkaz:

```
ondblog cancel stores_demo
```

Není možné zrušit změny protokolování, které se provedou okamžitě.

Ukončení protokolování pomocí obslužného příkazu ondblog

Pokud chcete ukončit protokolování dvou databází které jsou uvedeny v souboru nazvaném **dbfile**, spusíte následující příkaz:

```
ondblog nolog -f dbfile
```

Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ondblog

Pokud chcete pomocí obslužného programu **ondblog** docílit, aby se databáze nazvaná **stores_demo** stala kompatibilní se standardem ANSI, spusíte následující příkaz:

```
ondblog ansi stores_demo
```

Změna režimu protokolování databáze kompatibilní se standardem ANSI

Poté, co vytvoříte nebo převedete databázi na režim ANSI, není možné tento režim snadno změnit na jiný režim. Pokud převedete databázi na režim ANSI nedopatřením, proveďte při změně režimu protokolování tyto kroky:

Postup změny režimu protokolování:

1. Pomocí obslužného programu **dbexport** nebo jakéhokoli jiného obslužného programu uvolněte data z paměti. Obslužný program **dbexport** vytvoří **soubor schématu**.

Další informace o způsobu zavádění dat a jejich odstraňování z paměti naleznete v příručce *IBM Informix Migration Guide*.

2. K opětovnému vytvoření databáze s vyrovnávací pamětí a k zavedení dat použijte příkaz **dbimport -l buffered**.
K opětovnému vytvoření databáze bez vyrovnávací paměti a k zavedení dat použijte příkaz **dbimport -l**.

Změna režimu protokolování databáze pomocí obslužného programu **ontape**

Pokud používáte obslužný program **ontape** jako zálohovací nástroj, můžete ho použít ke změně režimu protokolování databáze.

Zapnutí protokolování transakcí pomocí obslužného programu **ontape**

Než změníte režim protokolování databáze, přečtěte si část “Změna režimu protokolování databáze” na stránce 12-2.

Obslužný program **ontape** přidá do databáze protokolování v okamžiku, kdy vytváříte zálohu úrovně 0.

Pokud chcete například pomocí obslužného programu **ontape** přidat protokolování s vyrovnávací pamětí do databáze nazvané **stores_demo**, spusíte tento příkaz:

```
ontape -s -B stores_demo
```

Pokud chcete pomocí obslužného programu **ontape** přidat protokolování bez vyrovnávací paměti do databáze nazvané **stores_demo**, spusíte tento příkaz:

```
ontape -s -U stores_demo
```

Kromě zapnutí protokolování transakcí vytvářejí tyto příkazy zálohy paměťových prostorů celého systému. Když vás obslužný program **ontape** vyzve k zadání úrovně zálohy, zadejte zálohu úrovně 0.

Rada: Při použití obslužného programu **ontape** musíte provést zálohu úrovně 0 všech paměťových prostorů.

Ukončení protokolování pomocí obslužného programu **ontape**

Pokud chcete pomocí obslužného programu **ontape** ukončit protokolování databáze nazvané **stores_demo**, spusíte tento příkaz:

```
ontape -N stores_demo
```

Změna režimu protokolování pomocí obslužného programu **ontape**

Pokud chcete pomocí obslužného programu **ontape** změnit režim použití vyrovnávací paměti z protokolování s vyrovnávací pamětí na protokolování bez vyrovnávací paměti v databázi nazvané **stores_demo**, aniž by se vytvořila záloha paměťových prostorů, spusíte tento příkaz:

```
ontape -U stores_demo
```

Pokud chcete pomocí obslužného programu **ontape** změnit režim použití vyrovnávací paměti z protokolování bez vyrovnávací paměti na protokolování s vyrovnávací pamětí v databázi nazvané **stores_demo**, aniž by se vytvořila záloha paměťových prostorů, spusíte tento příkaz:

```
ontape -B stores_demo
```

Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ontape

Pokud chcete pomocí obslužného programu **ontape** změnit databázi s názvem **stores_demo**, která již používá protokolování transakcí (ať již bez vyrovnávací paměti, nebo s vyrovnávací pamětí), na databázi kompatibilní se standardem ANSI, spusíte následující příkaz:

```
ontape -A stores_demo
```

Pokud chcete pomocí obslužného programu **ontape** změnit databázi s názvem **stores_demo**, která dosud nepoužívá protokolování transakcí, na databázi kompatibilní se standardem ANSI, spusíte následující příkaz:

```
ontape -s -A stores_demo
```

Kromě vytvoření databáze kompatibilní se standardem ANSI vytvoří tento příkaz v téže okamžiku zálohu paměťových prostorů. Když budete vyzváni k zadání úrovně, zadejte zálohu úrovně 0.

Rada: Poté, co změníte režim protokolování na kompatibilní se standardem *ANSI*, není možné ho snadno změnit zpět. Pokud chcete změnit režim protokolování databázi kompatibilních se standardem *ANSI*, musíte uvolnit data z paměti, znovu vytvořit databázi s novým režimem protokolování a znovu zavést data. Další informace naleznete v části “Změna režimu protokolování databáze kompatibilní se standardem ANSI” na stránce 12-3.

Změna režimu protokolování databáze pomocí programu ISA

Program ISA používá ke změně režimu protokolování databáze obslužný program **ondblog**. Pokud zapnete protokolování, provede se záloha úrovně 0. Další informace naleznete v nápovědě online k programu ISA a v “Změna režimu protokolování databáze pomocí obslužného programu ondblog” na stránce 12-3.

Změna režimu protokolování pomocí programu ON-Monitor (UNIX)

Pomocí programu ON-Monitor můžete měnit režim použití vyrovnávací paměti při protokolování mezi režimem bez vyrovnávací paměti a režimem s vyrovnávací pamětí. Pokud chcete k databázi přidat protokolování nebo chcete zajistit, aby byla kompatibilní se standardem ANSI, nemůžete použít program ON-Monitor. Je třeba, abyste použili obslužný program **ontape**.

Pokud chcete v databázi změnit režim použití vyrovnávací paměti při protokolování, vyberte volbu **Logical-Logs > Database**.

Změna režimu protokolování tabulek

Databázový server vytváří standardní tabulky, které ve výchozím nastavení používají protokolování. Pokud chcete vytvořit tabulku, která by nepoužívala protokolování, použijte příkaz **CREATE TABLE** s klauzulí **WITH LOG**. Další informace o příkazech **CREATE TABLE** a **ALTER TABLE** naleznete v příručce *IBM Informix Guide to SQL: Syntax*. Další informace uvádí část “Typy tabulek serveru Dynamic Server” na stránce 9-24.

Změna tabulky za účelem vypnutí protokolování

Chcete-li u tabulky přepnout z provádění protokolování na neprovádění protokolování, použijte příkaz jazyka SQL **ALTER TABLE** s volbou **TYPE** s hodnotou **RAW**. Například následující příkaz změní tabulku **tablog** na tabulku typu **RAW**:

```
ALTER TABLE tablog TYPE (RAW)
```

Změna tabulky za účelem zapnutí protokolování

Chcete-li přepnout z tabulky bez protokolování na tabulku s protokolováním, použijte příkaz jazyka SQL ALTER TABLE s volbou TYPE s hodnotou STANDARD. Například následující příkaz změní tabulku **tabnolog** na tabulku typu STANDARD:

```
ALTER TABLE tabnolog TYPE (STANDARD)
```

Upozornění: Když změníte tabulku na tabulku typu STANDARD, zapnete pro tuto tabulku protokolování. Poté, co tabulku změníte, proveďte zálohu úrovně 0, pokud musíte být schopni tuto tabulku obnovit.

Zakázání protokolování v dočasných tabulkách

Chcete-li zlepšit výkon a zabránit serveru Dynamic Server v přenosu dočasných tabulek při používání primárního serveru v prostředí replikace dat (sekundární servery replikace HDR, sekundární servery RS a sekundární servery SD), můžete zakázat protokolování v dočasných tabulkách.

Chcete-li zakázat protokolování v dočasných tabulkách, nastavte konfigurační parametr TEMPTAB_NOLOG na hodnotu 1.

Chcete-li změnit hodnotu parametru TEMPTAB_NOLOG, můžete použít příkaz **onmode -wf**.

Monitorování transakcí

Tato část obsahuje odkazy na možné způsoby monitorování transakcí.

Příkaz	Popis	Odkaz
onstat -x	Monitorování transakcí.	“Monitorování globální transakce” na stránce 24-13
onstat -g sql	Monitorování příkazů jazyka SQL uvedené v seznamu podle ID relace a databáze.	Monitorování výkonu v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i>
onstat -g stm	Monitorování využití paměti připravenými příkazy jazyka SQL.	Využití paměti v příručce <i>Příručka výkonnosti serveru IBM Informix Dynamic Server</i>

Monitorování režimu protokolování databáze

Tato část popisuje, jakým způsobem lze monitorovat režim protokolování databáze a tabulek.

Monitorování režimu protokolování u tabulek typu SMI

Chcete-li určit režim protokolování, spusťte dotaz na tabulku **sysdatabases** v databázi **sysmaster**. Tato tabulka obsahuje řádek pro každou databázi, kterou spravuje databázový server. Pole **flags** určuje režim protokolování databáze. Pole **is_logging** (protokolování), **is_buff_log** (protokolování s vyrovnávací pamětí) a **is_ansi** (kompatibilní se standardem ANSI) označují, zda je protokolování aktivní a zda se používá protokolování s vyrovnávací pamětí nebo kompatibilní se standardem ANSI. Další informace o sloupcích této tabulky naleznete v části zaměřené na **sysdatabases** v kapitole o databázi typu **sysmaster** v příručce *IBM Informix Administrator's Reference*.

Monitorování režimu protokolování pomocí programu ON-Monitor (UNIX)

Pokud chcete ke zjištění režimu protokolování databáze použít program ON-Monitor, vyberte volbu **Status > Databases**. Je-li protokolování databáze zapnuto, zobrazí se v programu ON-Monitor režim použití vyrovnávací paměti. Program ON-Monitor může zobrazit maximálně 100 databází. Jestliže máte na databázovém serveru více než 100 databází,

použijte k zobrazení úplného seznamu tabulky SMI, jak bylo popsáno v části “Monitorování režimu protokolování u tabulek typu SMI” na stránce 12-6.

Monitorování režimu protokolování pomocí programu ISA

Pokud chcete k zobrazení stavu protokolování a režimu použití vyrovnávací paměti použít program ISA, vyberte položku **SQL > Schéma**.

Kapitola 13. Logický protokol

Obsah kapitoly	13-1
Co je logický protokol?	13-1
Umístění souborů logického protokolu	13-2
Identifikace souborů logického protokolu	13-2
Příznaky stavu souborů logického protokolu	13-3
Velikost logického protokolu	13-3
Počet souborů logického protokolu	13-4
Pokyny týkající se výkonu.	13-4
Dynamické přidělování protokolů	13-5
Uvolnění souborů logického protokolu	13-5
Akce v případě, že další soubor logického protokolu není volný	13-5
Akce v případě, že následující logický soubor obsahuje poslední kontrolní bod	13-6
Protokolování prostorů blobspace a jednoduché velké objekty	13-6
Aktivace prostorů blobspace přepnutím souborů protokolu	13-7
Zálohování souborů protokolů za účelem uvolnění stránek blobpage	13-7
Zálohování prostorů blobspace po vložení nebo odstranění dat typu TEXT a BYTE.	13-7
Protokolování prostorů sbospace a inteligentních velkých objektů	13-7
Používání protokolování prostoru sbospace	13-8
Kdy zvolit protokolování inteligentních velkých objektů	13-8
Protokolování aktualizovaných inteligentních velkých objektů	13-8
Zapnutí a vypnutí protokolování prostoru sbospace	13-8
Používání záznamů protokolu inteligentních velkých objektů	13-9
Zabránění dlouhým transakcím při protokolování dat inteligentních velkých objektů	13-9
Proces protokolování	13-10
Protokolování prostorů dbospace	13-10
Protokolování prostorů blobspace	13-10

Obsah kapitoly

Informace, které uvádí Kapitola 11, “Protokolování”, na stránce 11-1 a tato kapitola, vám pomohou porozumět způsobu, jakým databázový server používá logický protokol. Další informace týkající se používání úloh logického protokolu uvádí Kapitola 14, “Správa souborů logických protokolů”, na stránce 14-1 a Kapitola 12, “Správa režimu protokolování databáze”, na stránce 12-1.

Co je logický protokol?

Aby se zachovala historie transakcí a změn databázového serveru od doby poslední zálohy paměťového prostoru, vytváří databázový server logické záznamy. Databázový server ukládá tyto logické záznamy do *logického protokolu*, což je cyklický soubor tvořený třemi nebo více soubory logického protokolu. Tento protokol se nazývá *logický*, protože logické záznamy reprezentují logické operace databázového serveru, na rozdíl od fyzických operací. V libovolnou dobu obsahuje kombinace zálohy paměťového prostoru a zálohy logického protokolu úplnou kopii dat databázového serveru.

Jako administrátor databázového serveru musíte nakonfigurovat a spravovat logický protokol. Pokud například nebudete pravidelně zálohovat soubory logického protokolu, logický protokol se zaplní a databázový server pozastaví zpracování.

Tyto odpovědnosti zahrnují následující úlohy:

- Výběr odpovídajícího umístění pro logický protokol.

Další informace naleznete v části “Umístění souborů logického protokolu” na stránce 13-2.

- Monitorování stavu souboru logického protokolu.
Další informace naleznete v části “Identifikace souborů logického protokolu” na stránce 13-2.
- Přidělení vhodného objemu diskového prostoru pro logický protokol.
Další informace naleznete v části “Velikost logického protokolu” na stránce 13-3.
- Přidělení dodatečných souborů protokolu v případě nutnosti.
Další informace naleznete v části “Přidělení souborů protokolu” na stránce 14-9.
- Zálohování souborů logického protokolu na média.
Další informace naleznete v částech “Zálohování logických souborů” na stránce 14-4 a “Uvolnění souborů logického protokolu” na stránce 13-5.
- Správa protokolování prostorů blobspace a sbspace
Další informace naleznete v částech “Protokolování prostorů blobspace a jednoduché velké objekty” na stránce 13-6 a “Protokolování prostorů sbspace a inteligentních velkých objektů” na stránce 13-7.

Umístění souborů logického protokolu

Když databázový server inicializuje diskový prostor, umísťuje soubory logického protokolu a fyzický protokol do kořenového prostoru dbspace. Nad touto činností nemáte žádnou kontrolu. Pokud chcete zlepšit výkon (zvláště snížit počet zápisů do kořenového prostoru dbspace a minimalizovat kolize požadavků), přesuňte soubory logického protokolu na disk, který není sdílený aktivními tabulkami nebo fyzickým protokolem. Další informace naleznete v části “Přesun souboru logického protokolu do jiného prostoru dbspace” na stránce 14-14.

Pokud chcete dále zlepšovat výkon, rozdělte soubory logického protokolu na dvě skupiny a uložte je na dva samostatné disky (z nichž žádný neobsahuje data). Pokud máte například šest souborů logického protokolu, můžete umístit soubory 1, 3 a 5 na disk 1 a soubory 2, 4 a 6 na disk 2. Toto uspořádání zvýší výkon, protože stejná disková jednotka nemusí nikdy zároveň zapisovat do aktuálního souboru logického protokolu a na zálohy na páse.

Soubory logického protokolu obsahují kritické informace a pro zajištění maximální ochrany dat by měly být zrcadleny. Pokud přesunete soubory logického protokolu do jiného prostoru dbspace, naplánujte si v tomto prostoru dbspace zrcadlení.

Identifikace souborů logického protokolu

Každý soubor logického protokolu, zálohovaný na média nebo ne, má své *jedinečné* identifikační číslo. Posloupnost začíná číslicí 1 pro první soubor logického protokolu, který je naplněn poté, co jste inicializovali diskový prostor databázového serveru. Když se aktuální soubor logického protokolu zaplní, databázový server přepne na další soubor logického protokolu a zvýší jedinečné identifikační číslo pro nový soubor logického protokolu o jednu. Soubory protokolu, které jsou nově přidány nebo jsou označeny pro odstranění, mají jedinečná identifikační čísla rovna 0.

Skutečný diskový prostor přidělený každému souboru logického protokolu má identifikační číslo známé jako *číslo souboru protokolu*. Pokud například nakonfigurujete šest souborů logického protokolu, budou mít tyto soubory čísla protokolu od jedné do šesti. Čísla protokolů nemusí jít za sebou. Jak jsou soubory logického protokolu zálohovány a uvolňovány, používá databázový server opakovaně diskový prostor pro soubory logického protokolu.

Tabulka 13-1 ilustruje vztah mezi čísly protokolu a jedinečnými identifikačními čísly. Protokol 7 je vložen za protokol 5 a používá se poprvé v druhém oběhu.

Tabulka 13-1. Posloupnost číslování souborů logického protokolu

Číslo souboru protokolu	Jedinečné identifikační číslo v prvním oběhu	Jedinečné identifikační číslo v druhém oběhu	Jedinečné identifikační číslo v třetím oběhu
1	1	7	14
2	2	8	15
3	3	9	16
4	4	10	17
5	5	11	18
7	0	12	19
6	6	13	20

Příznaky stavu souborů logického protokolu

Všechny soubory logického protokolu mají na první pozici jeden z následujících příznaků stavu: Added (**A**), Deleted (**D**), Free (**F**) nebo Used (**U**). Tabulka 13-2 uvádí možné kombinace příznaků stavů protokolu.

Tabulka 13-2. Příznaky stavu logického protokolu

Příznak stavu	Stav souboru logického protokolu
A-----	Logický soubor byl <i>přidán (added)</i> , je dostupný, ale nebyl dosud použit.
D-----	Pokud vypustíte soubor protokolu se stavem U-B, bude tento soubor označen jako <i>odstraněný (deleted)</i> . Tento soubor bude vypuštěn a jeho prostor uvolněn nebo opakovaně využit, jakmile provedete zálohu úrovně 0 všech paměťových prostorů.
F-----	Soubor protokolu je <i>volný (free)</i> a dostupný k použití. Soubor logického protokolu je uvolněn po svém zálohování, všechny transakce v rámci souboru logického protokolu jsou uzavřené a nejstarší aktualizace uložená v tomto souboru je vyprázdněna na disk.
U	Soubor protokolu byl <i>použit (used)</i> , ale nebyl zálohován.
U-B----	Soubor protokolu je <i>zálohovaný (backed up)</i> , ale stále je potřebný v případě obnovy. (Soubor protokolu je uvolněn v případě, že již pro obnovu není dále nutný).
U-B---L	Protokol je zálohovaný, ale stále potřebný pro obnovu. Obsahuje záznam posledního kontrolního bodu.
U---C	Databázový server <i>aktuálně</i> zaplňuje soubor protokolu.
U---C-L	Tento aktuální soubor protokolu obsahuje záznam <i>posledního (last)</i> kontrolního bodu.

Pomocí příkazu **onstat -l** můžete zobrazit seznam souborů protokolu seřazený podle čísla a monitorovat příznaky stavu a procentní hodnoty využití prostoru protokolu. Další informace naleznete v části “onstat -l” na stránce 14-8.

Velikost logického protokolu

Rozhodněte, kolik souborů protokolu budete chtít a jaké velikosti. Pokud přidělíte více diskového prostoru než je nutné, budete plýtvat diskovým prostorem. Jestliže ovšem nepřidělíte dostatek diskového prostoru, může být nepříznivě ovlivněn výkon. Při úvahách o velikosti a počtu souborů logického protokolu mějte na paměti tyto skutečnosti:

- Minimální velikost souboru logického protokolu je 200 kilobajtů.
- Maximální velikost souboru logického protokolu je 1048576 stránek (ekvivalent 0x100000).
- Menší soubory protokolu znamenají, že budete moci v případě selhání disku obsahujícího soubory protokolu provést obnovu do pozdějšího času. Pokud je nastaveno souvislé zálohování protokolu, budou se soubory protokolu automaticky zálohovat při jejich zaplnění. Menší protokoly mají za následek poněkud delší logickou obnovu.
- Pokud do protokolů zapisuje mnoho uživatelů současně, používejte větší soubory protokolu.

Počet souborů logického protokolu

Když uvažujete o počtu souborů logického protokolu, mějte na paměti tyto skutečnosti:

- Musíte mít vždy alespoň tři soubory logického protokolu. Maximální počet souborů logického protokolu je 32767. Počet souborů protokolu závisí na velikosti souborů protokolu.
- Počet souborů protokolu ovlivňuje frekvenci zálohování souborů protokolu.
- Počet souborů logického protokolu ovlivňuje rychlost, s jakou je možné uvolnit stránky blobpage prostorů blobspace. Další informace naleznete v části “Zálohování souborů protokolů za účelem uvolnění stránek blobpage” na stránce 13-7.

Pokyny týkající se výkonu

Při dané aktivitě systému platí, že čím menší diskový prostor logickému protokolu přidělíte, tím rychleji se tento diskový prostor zaplní, a tím větší je pravděpodobnost, že aktivita uživatele bude blokována kvůli zálohování a kontrolním bodům. Vyladte velikost logického protokolu a naleznete optimální hodnotu pro váš systém.

- Zálohy logického protokolu
Když se soubory logického protokolu zaplní, musíte je zálohovat. Proces zálohování může bránit zpracování transakcí, které používá data umístěná na stejném disku jako soubory logického protokolu. Umístěte fyzický protokol, logické protokoly a uživatelská data na samostatné disky. (Další informace naleznete v příručce *IBM Informix Backup and Restore Guide*.)
- Velikost logického protokolu
Menší logický protokol se zaplňuje rychleji než větší logický protokol. Můžete přidat větší soubor logického protokolu, jak bylo vysvětleno v části “Ruční přidání souborů logického protokolu” na stránce 14-11.
- Velikost záznamů jednotlivých logických protokolů
Velikost záznamů logických protokolů kolísá v závislosti na zpracovávané operaci a na prostředí databázového serveru. Obecně platí, že čím delší řádky dat jsou zpracovávány, tím delší záznamy jsou zapisovány do logického protokolu. Logický protokol obsahuje obrazy řádků, které byly vloženy, aktualizovány nebo odstraněny. Aktualizace mohou zabírat až dvakrát více místa než vložení a odstranění, neboť obsahují obrazy před aktualizací a po aktualizaci. (Vložení ukládá pouze obrazy po vložení a odstranění ukládá pouze obrazy před odstraněním.)
- Počet záznamů logických protokolů
Čím více záznamů logického protokolu se zapíše do logického protokolu, tím rychleji se zaplní. Databáze s protokolováním transakcí zaplní logický protokol rychleji než transakce prováděné vůči databázi bez protokolování transakcí.
- Typ použití vyrovnávací paměti při protokolování
Databáze používající protokolování transakcí bez vyrovnávací paměti zaplní logický protokol rychleji než databáze, které používají protokolování transakcí s vyrovnávací pamětí.

- Replikace Enterprise Replication databázové tabulky
Jelikož replikace Enterprise Replication generuje v replikačních tabulkách obrazy před transakcí a obrazy po transakci, může způsobit zaplnění logického protokolu.
- Frekvence odvolávání transakcí
Pokud bude častěji docházet k odvolávání transakcí, zaplní se logický protokol rychleji. I samotné odvolání transakce vyžaduje prostor v souboru logického protokolu, přestože jsou záznamy o odvolaných transakcích malé.
- Počet inteligentních velkých objektů
Inteligentní velké objekty s povoleným protokolováním uživatelských dat, jejichž uživatelská data jsou aktualizována ve značném objemu, mohou zaplňovat logické protokoly obrovskou rychlostí. Pokud nechcete protokolovat metadata, použijte dočasné inteligentní velké objekty.

Dynamické přidělování protokolů

Dynamické přidělování protokolů zabraňuje zaplnění souborů protokolu a zastavení systému během dlouhých odvolání transakcí. Tato funkce se aktivuje pouze v okamžiku, kdy následující soubor protokolu obsahuje otevřenou transakci. (*Transakce* je označována jako *dlouhá*, pokud není potvrzená nebo odvolaná v okamžiku, když dosáhne horní meze (high-watermark) dlouhé transakce.)

Databázový server automaticky (dynamicky) přidělí soubor protokolu za aktuální soubor protokolu tehdy, pokud následující soubor protokolu obsahuje otevřenou transakci. Dynamické přidělování protokolů umožňuje provádět následující činnosti:

- Přidat soubor protokolu, zatímco je systém aktivní.
- Vložit soubor protokolu za aktuální soubor protokolu.
- Okamžitě přistupovat k novým souborům protokolu, přestože kořenový prostor dbspace nebyl zálohován.

Nejlépeším způsobem, jak prověřit funkci dynamického přidělování protokolů, je vytvořit transakci, která využije všechny soubory protokolu a potom pomocí příkazu **onstat -l** vyhledat nově přidávané soubory protokolu. Další informace naleznete v části “Přidělení souborů protokolu” na stránce 14-9.

Důležité: Je nezbytné, abyste prováděli zálohování souborů protokolu a zabránili tak jejich zaplnění. Pokud se soubory protokolu zaplní, je systém zastaven, dokud neprovedete zálohu.

Uvolnění souborů logického protokolu

Pokaždé, když databázový server potvrdí nebo odvolá transakci, pokouší se uvolnit soubor logického protokolu ve kterém tato transakce začala. Než databázový server uvolní soubor logického protokolu pro opětovné použití, musí být splněna tato kritéria:

- Soubor protokolu byl zálohován.
- Žádné záznamy v souboru logického protokolu nejsou přidružené k otevřeným transakcím.
- Soubor logického protokolu neobsahuje nejstarší aktualizaci dosud nevyprázdněnou na disk.

Akce v případě, že další soubor logického protokolu není volný

Pokud se databázový server pokouší přepnout na následující soubor logického protokolu a zjistí, že v pořadí následující soubor logického protokolu je stále používán, pozastaví databázový server celé zpracování. Databázový server nemůže přeskočit používaný soubor a

zapisovat do volného souboru mimo pořadí dokonce ani tehdy, pokud jsou volné jiné soubory logického protokolu. Zpracování se zastaví, aby byla chráněna data v souboru logického protokolu.

Soubor logického protokolu může být používán z jakékoli z následujících příčin:

- Soubor obsahuje poslední kontrolní bod nebo nejstarší aktualizaci, která dosud nebyla vyprázdněna na disk.

Zadáním příkazu **onmode -c** se provede kontrolní bod a uvolní se soubor logického protokolu. Další informace naleznete v části “Vynucení kontrolního bodu” na stránce 16-4.

- Soubor obsahuje otevřenou transakci.

Otevřená transakce je dlouhá transakce, popisovaná v části “Nastavení horních mezí pro odvolání dlouhých transakcí” na stránce 14-17.

- Soubor není zálohovaný.

Pokud soubor logického protokolu není zálohovaný, bude zpracování pokračovat, jakmile provedete zálohování souboru logického protokolu pomocí programu ON-Bar nebo obslužného programu **ontape**.

Akce v případě, že následující logický soubor obsahuje poslední kontrolní bod

Databázový server nepozastaví zpracování, když bude následující soubor protokolu obsahovat poslední kontrolní bod nebo nejstarší aktualizaci. Databázový server při přepínání na poslední dostupný protokol vždy vynutí kontrolní bod, pokud se záznam předchozího kontrolního bodu nebo nejstarší aktualizovaný záznam dosud nevyprázdněný na disk nachází v protokolu, který následuje za posledním dostupným protokolem. Pokud například čtyři soubory logického protokolu mají stav uvedený v následujícím seznamu, vynutí databázový server při přepnutí na soubor logického protokolu číslo 3 kontrolní bod.

Číslo souboru protokolu	Stav souboru logického protokolu
1	U-B----
2	U---C--
3	F
4	U-B---L

Protokolování prostorů blobspace a jednoduché velké objekty

Data jednoduchých velkých objektů (typu TEXT nebo BYTE) mohou být příliš objemná, než aby je bylo možné zahrnout do záznamu logického protokolu. Pokud jsou jednoduché velké objekty neustále protokolovány, mohou být tak velké, že mohou zpomalit logický protokol.

Databázový server předpokládá, že jste navrhli databázi tak, aby v prostorech dbspace byly uloženy menší jednoduché velké objekty a větší jednoduché velké objekty byly uloženy v prostorech blobspace:

- Databázový server zahrnuje data jednoduchých velkých objektů do záznamů protokolu v případě jednoduchých velkých objektů uložených v prostorech dbspace.
- Databázový server nezahrnuje data jednoduchých velkých objektů do záznamů protokolu v případě jednoduchých velkých objektů uložených v prostorech blobspace. Logický protokol zaznamenává data prostoru blobspace pouze tehdy, když zálohujete logické protokoly.

Pokud chcete dosáhnout lepšího celkového výkonu aplikací, které provádějí časté aktualizace jednoduchých velkých objektů v prostorech blobspace, zmenšíte velikost logického protokolu. Menší protokoly mohou zlepšit přístup k jednoduchým velkým objektům, které

musí být používány opakovaně. Další informace o vlivu konfigurace na využití vstupu - výstupu naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Aktivace prostorů blobspace přepnutím souborů protokolu

Na následující soubor logického protokolu musíte přepnout v těchto situacích:

- Po vytvoření souboru blobspace, pokud zamýšlíte okamžitě vložit do prostoru blobspace jednoduché velké objekty.
- Po přidání nového bloku do existujícího prostoru blobspace, pokud zamýšlíte do tohoto prostoru blobspace vložit jednoduché velké objekty, které budou používat tento nový blok.

Databázový server vyžaduje, aby se příkazy vytvářející prostor blobspace, příkaz vytvářející v tomto prostoru blobspace nový blok a příkazy vkládající do tohoto prostoru blobspace jednoduché velké objekty objevily v samostatných souborech logických protokolů. Tento požadavek nezávisí na stavu protokolování databáze.

Další pokyny týkající se přepínání na následující soubor protokolu naleznete v části “Přepnutí na následující soubor logického protokolu” na stránce 14-5.

Zálohování souborů protokolů za účelem uvolnění stránek blobpage

Když odstraníte data uložená na stránkách prostoru blobspace, nemusí být tyto stránky nezbytně uvolněny pro opakované použití. Stránky prostoru blobspace jsou uvolněny pouze tehdy, když jsou provedeny *obě* následující akce:

- Data typu TEXT nebo BYTE byla odstraněna, a to příkazem UPDATE použitým k aktualizaci sloupce nebo odstraněním celého řádku.
- Byl zálohován logický protokol, ve kterém byly uloženy příkazy INSERT, které vložily řádek obsahující data typu TEXT nebo BYTE.

Zálohování prostorů blobspace po vložení nebo odstranění dat typu TEXT a BYTE

Ujistěte se, že zálohujete všechny prostory blobspace a logické protokoly obsahující transakce prováděné na jednoduchých velkých objektech uložených v prostoru blobspace. Když databázový server zálohuje protokol, používá ukazatel dat v logickém protokolu ke kopírování změněných dat typu TEXT a BYTE z prostoru blobspace do logického protokolu.

Protokolování prostorů sbospace a inteligentních velkých objektů

Prostory sbospace, popsané v části “Prostory sbospace” na stránce 9-12, obsahují dvě komponenty: metadata a uživatelská data. Ve výchozím nastavení se prostory sbospace *neprotokolují*.

Komponenta *metadata* prostoru sbospace popisuje kritické charakteristiky inteligentních velkých objektů uložených v určitém prostoru sbospace. Metadata obsahují ukazatele na inteligentní velké objekty. Pokud by tato metadata byla poškozena nebo se stala nepřístupná, prostor sbospace by byl poškozen a inteligentní velké objekty uložené v tomto prostoru sbospace by nebyly obnovitelné.

Metadata v běžném prostoru sbospace se *vždy* protokolují, i když je pro databázi protokolování vypnuté. Protokolování metadat prostoru sbospace zajišťuje, že je možné tato metadata vždy obnovit na konzistentní stav transakce. Metadata v dočasném prostoru sbospace se ovšem *neprotokolují*.

Používání protokolování prostoru sbspace

Pokud je protokolován prostor sbspace, databázový server se zpomalí a logické protokoly budou rychle zaplňovány. Pokud chcete používat protokolování prostorů sbspace, musíte zajistit, aby byly logické protokoly dostatečně velké pro uchování protokolovaných dat. Další informace naleznete v části “Odhad velikosti protokolu při protokolování inteligentních velkých objektů” na stránce 14-4.

Pokud zapnete protokolování databáze, databázový server nezačne s protokolováním, dokud neprovedete zálohu úrovně 0. Pokud ovšem zapnete protokolování inteligentních velkých objektů, databázový server začne protokolovat změny okamžitě. Abyste snížili objem záznamů protokolu, můžete při zavádění inteligentních velkých objektů vypnout protokolování a potom opět protokolování zapnout, abyste zachytili aktualizace těchto inteligentních velkých objektů.

Upozornění: Pokud zapnete protokolování inteligentních velkých objektů, musíte okamžitě provést zálohu úrovně 0, abyste byli schopni provádět obnovu a obnovit inteligentní velké objekty.

Další informace naleznete v části “Zálohování prostorů sbspace” na stránce 14-5 a v příručce *IBM Informix Backup and Restore Guide*.

Kdy zvolit protokolování inteligentních velkých objektů

Protokolování inteligentních velkých objektů používejte, pokud uživatelé často aktualizují data nebo pokud je schopnost obnovy aktualizovaných dat životně důležitá. Databázový server zapisuje záznam o operaci (insert (vlození), update (aktualizace), delete (odstranění), read (čtení) nebo write (zápis)) do vyrovnávací paměti logického protokolu. Do záznamu protokolu je zahrnuta modifikovaná část dat CLOB nebo BLOB.

Pokud chcete zvýšit výkon, vypněte protokolování inteligentních velkých objektů. Rovněž vypněte protokolování, pokud uživatelé data především analyzují a aktualizují je nepřilíš často nebo pokud obnova dat není životně důležitá.

Protokolování aktualizovaných inteligentních velkých objektů

Když aktualizujete inteligentní velký objekt, databázový server neprotokoluje celý objekt. Předpokládejme, že uživatel při povoleném protokolování inteligentních velkých objektů zapsal X bajtů dat s posunem Y. Databázový server uloží do protokolu následující údaje:

- Pokud je posun Y nastaven na konec velkého objektu, databázový server uloží do protokolu X bajtů (aktualizovaný rozsah bajtů).
- Pokud posun Y ukazuje na začátek nebo doprostřed velkého objektu, uloží databázový server do protokolu nejmenší z níže uvedených možností:
 - Rozdíl mezi starým a novým obrazem.
 - Obraz před transakcí a obraz po transakci.
 - Jsou-li obrazy před transakcí a po transakci shodné, nebude do protokolu uloženo nic.

Zapnutí a vypnutí protokolování prostoru sbspace

Pokud chcete používat protokolování prostoru sbspace, zadejte při vytváření prostoru sbspace do příkazu **onspace** volbu **-Df LOGGING=ON**. Pokud je protokolování v prostoru sbspace vypnuto, můžete zapnout protokolování inteligentních velkých objektů v určitých sloupcích. Jeden sloupec, který obsahuje inteligentní velké objekty, může mít protokolování zapnuté, zatímco jiný sloupec může mít protokolování vypnuté.

Chcete-li ověřit, že jsou inteligentní velké objekty v prostoru sbspace protokolovány, použijte tento příkaz:

```
oncheck -pS sbspace | grep "Create Flags"
```


Pokud vytvoříte inteligentní velké objekty v prostoru sbspace s výchozí volbou protokolování a ve výstupu se zobrazí příznak LO_NOLOG, znamená to, že inteligentní velké objekty v tomto prostoru sbspace nejsou protokolovány. Pokud se ve výstupu zobrazí příznak LO_LOG, znamená to, že inteligentní velké objekty v tomto prostoru sbspace jsou protokolovány.

Stav protokolování prostoru sbspace můžete změnit libovolným z následujících způsobů.

Použitá funkce nebo příkaz	Akce protokolování	Odkazy
<code>onspaces -ch -Df LOGGING=ON</code>	Zapne nebo vypne protokolování existujícího prostoru sbspace.	“Změna paměťových charakteristik inteligentních velkých objektů” na stránce 10-26 <i>IBM Informix Administrator's Reference</i>
Volba LOG v klauzuli PUT příkazu CREATE TABLE nebo ALTER TABLE	Zapne protokolování všech inteligentních velkých objektů, které zavedete do tohoto sloupce.	“Protokolování” na stránce 9-17 <i>IBM Informix Guide to SQL: Syntax</i>
Funkce <code>mi_lo_create</code> rozhraní DataBlade API	Vypne protokolování inteligentního velkého objektu při jeho počátečním zavedení.	<i>IBM Informix DataBlade API Function Reference</i>
Funkce <code>mi_lo_alter</code> rozhraní DataBlade API	Zapne protokolování po dokončení zavedení.	<i>IBM Informix DataBlade API Function Reference</i>
Funkce <code>ifx_lo_create</code> jazyka ESQL/C	Vypne protokolování inteligentního velkého objektu při jeho počátečním zavedení.	<i>IBM Informix ESQL/C Programmer's Manual</i>
Funkce <code>ifx_lo_alter</code> jazyka ESQL/C	Zapne protokolování po dokončení zavedení.	<i>IBM Informix ESQL/C Programmer's Manual</i>

Používání záznamů protokolu inteligentních velkých objektů

Když vytvoříte inteligentní velký objekt volbou LOG, logický protokol vytvoří *záznam protokolu inteligentního velkého objektu*. Záznamy protokolu inteligentního velkého objektu zaznamenávají změny uživatelských dat nebo metadat. Při aktualizaci inteligentních velkých objektů se v záznamu protokolu zobrazí pouze změněná část stránky sbspace. Záznamy protokolu uživatelských dat se objeví v logickém protokolu pouze tehdy, pokud je povoleno protokolování inteligentních velkých objektů.

Upozornění: Při povolování protokolování často aktualizovaných inteligentních velkých objektů buďte opatrní. Režie protokolování může značně zpomalit provoz databázového serveru.

Další informace o záznamech protokolu inteligentních velkých objektů naleznete v kapitole týkající se interpretace záznamů logického protokolu v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Zabránění dlouhým transakcím při protokolování dat inteligentních velkých objektů

Inteligentní velké objekty můžete používat v situacích, kdy proces shromažďování dat pro jediný inteligentní velký objekt trvá dlouho. Příkladem může být aplikace, která zaznamenává po mnoho hodin zvuková data nízké kvality. Ačkoli množství shromážděných dat může být malé, relace zaznamenávání může trvat dlouho a výsledkem může být stav označovaný jako dlouhá transakce.

Rada: Pokud chcete zabránit vzniku dlouhých transakcí, pravidelně potvrzujte transakce zápisu do inteligentních velkých objektů.

Proces protokolování

Tato část podrobně popisuje proces protokolování u prostorů dbspace, blobspace a sbspace. Uvedené informace nejsou požadovány při provádění běžných úloh administrace databázového serveru.

Protokolování prostorů dbspace

Databázový server používá pro operace, která zahrnují data uložená v prostorech dbspace, následující proces protokolování:

1. Přečte datovou stránku z disku do stránkové vyrovnávací paměti sdílené paměti.
2. V případě potřeby zkopíruje nezměněnou stránku do vyrovnávací paměti fyzického protokolu.
3. V případě potřeby zapíše nová data do stránkové vyrovnávací paměti a vytvoří záznam transakce v logickém protokolu.
4. Vyprázdní vyrovnávací paměť fyzického protokolu do fyzického protokolu na disk.
5. Vyprázdní vyrovnávací paměť logického protokolu do logického protokolu na disk.
6. Vyprázdní stránkovou vyrovnávací paměť a zapíše ji zpět na disk.

Protokolování prostorů blobspace

Databázový server protokoluje data prostoru blobspace, data ovšem neprocházejí ani sdílenou pamětí ani soubory logického protokolu na disku. Databázový server zkopíruje data uložená v prostoru blobspace přímo z disku na pásku. Záznamy změn režijních stránek prostoru blobspace (stránky s mapou volných stránek a stránky bitových map) jsou jedinými daty prostoru blobspace, která jsou ukládána do logického protokolu.

Kapitola 14. Správa souborů logických protokolů

Obsah kapitoly	14-1
Odhad velikosti a počtu souborů protokolu.	14-2
Odhad velikosti protokolu při protokolování inteligentních velkých objektů	14-4
Odhad počtu souborů logického protokolu	14-4
Zálohování logických souborů	14-4
Zálohování prostorů blobspace	14-5
Zálohování prostorů sbspace	14-5
Prepnutí na následující soubor logického protokolu	14-5
Uvolnění souboru logického protokolu	14-6
Odstranění souboru protokolu se stavem D	14-6
Uvolnění souboru protokolu se stavem U	14-6
Uvolnění souboru protokolu se stavem U-B nebo F	14-6
Uvolnění souboru protokolu se stavem U-C nebo U-C-L	14-7
Uvolnění souboru protokolu se stavem U-B-L	14-7
Monitorování aktivity protokolování.	14-7
Monitorování logického protokolu za účelem zjištění jeho zaplnění	14-8
onstat -l	14-8
oncheck -pr	14-8
Monitorování dočasných logických protokolů	14-8
Použití tabulek SMI.	14-9
Použití programu ON-Monitor (systém UNIX).	14-9
Monitorování stavu zálohy protokolů	14-9
Přidělení souborů protokolu	14-9
Dynamické přidávání protokolů	14-9
Velikost a počet dynamicky přidávaných souborů protokolu	14-10
Umístění dynamicky připojovaných souborů protokolu	14-10
Ruční přidání souborů logického protokolu	14-11
Vypuštění souborů logického protokolu	14-12
Změna velikosti souborů logického protokolu	14-13
Přesun souboru logického protokolu do jiného prostoru dspace	14-14
Změna konfiguračních parametrů protokolování	14-14
Použití programu ON-Monitor ke změně parametru LOGFILES (systém UNIX)	14-15
Zobrazení záznamů logického protokolu	14-16
Monitorování událostí dynamického přidávání protokolů	14-16
Nastavení horních mezí pro odvolání dlouhých transakcí	14-17
Horní mez dlouhé transakce (LTXHWM)	14-17
Horní mez dlouhé transakce s výlučným přístupem (LTXEHW)	14-18
Úprava velikosti souborů protokolu za účelem zabránění dlouhých transakcí	14-18
Obnova zablokované dlouhé transakce.	14-18

Obsah kapitoly

Soubory logických protokolů musíte spravovat i v případě, že žádná z databází nepoužívá protokolování transakcí. Základní informace o logických protokolech uvádí Kapitola 13, “Logický protokol”, na stránce 13-1.

Abyste mohli provádět jakoukoli ze změn uvedených v této kapitole, musíte se přihlásit do operačního systému UNIX jako uživatel **informix** nebo **root**. V operačním systému Windows musíte být členem skupiny **Informix-Admin**.

Při nastavování logického protokolu provádíte tyto úlohy:

- Před inicializací databázového serveru zadejte pomocí parametru LOGFILES počet souborů logického protokolu, které chcete vytvořit.

- Poté, co je databázový server online, odhadněte velikost a počet souborů logického protokolu, které budou v systému potřeba.
Další informace naleznete v části “Odhad velikosti a počtu souborů protokolu” na stránce 14-2.
- Pokud nechcete používat výchozí hodnoty, změňte konfigurační parametry LOGSIZE a LOGBUFF.
Další informace naleznete v části “Změna konfiguračních parametrů protokolování” na stránce 14-14.
- Přidejte odhadovaný počet souborů logického protokolu.
Další informace naleznete v části “Přidělení souborů protokolu” na stránce 14-9.

Následující úlohy budete provádět stále:

- Zálohování souboru logického protokolu.
- Přepnutí na následující soubor logického protokolu.
- Uvolnění souboru logického protokolu.
- Monitorování aktivity protokolování a stavu záloh protokolů.

V případě potřeby budete provádět následující činnosti:

- Přidání souboru logického protokolu.
- Vypuštění souboru logického protokolu.
- Změna velikosti souboru logického protokolu.
- Přesun souboru logického protokolu.
- Změna konfiguračních parametrů logického protokolu.
- Monitorování alarmů událostí pro logické protokoly.
- Nastavení horních mezí (high-watermarks) u transakcí.

Informace o použití administrativních příkazů rozhraní SQL API namísto některých příkazů **oncheck**, **onmode**, **onparams** a **onspaces** naleznete v příručce “Vzdálená správa pomocí příkazů SQL” na stránce 26-10 a v příručce *IBM Informix Guide to SQL: Syntax*.

Odhad velikosti a počtu souborů protokolu

Velikost a počet souborů protokolu nastavíte pomocí konfiguračního parametru LOGSIZE.

Velikost místa protokolu, které je pro databázový server optimální, závisí na následujících faktorech:

- Na požadavcích aplikace a na rozsahu aktualizací ostatních aplikací. Zvýšená míra provádění aktualizací vyžaduje přidělení většího prostoru protokolu.
- Cílová doba obnovy (RTO) pro čas v sekundách, během kterého se server musí obnovit z problému poté, co ho restartujete a uvedete do režimu online nebo do klidového režimu.
Zvažte, jaké množství dat můžete v případě katastrofické události tolerovat. Častější zálohování protokolu snižuje riziko ztráty dat a transakcí a vyžaduje větší místo protokolu.
- Zda používáte konfigurace replikace Enterprise Replication nebo replikace dat, například sekundární servery replikace HDR, sekundární servery SD nebo sekundární servery RS.
Tyto služby replikace mohou ovlivnit počet a velikost souborů protokolu. Informace o tom, zda váš systém používá některou z těchto služeb replikace, naleznete v pokynech Kapitola 20, “Použití replikace HDR (verze pro podniky/pracovní skupiny)”, na stránce 20-1 nebo v *IBM Informix Dynamic Server Enterprise Replication Guide*.

Některé pokyny k určení velikosti protokolu:

- Obecně je mnohem jednodušší spravovat několik velkých souborů protokolu než mnoho malých.
- Příliš mnoho místa protokolu výkon neovlivní. Nedostatek souborů protokolu a místa protokolu však výkon ovlivnit může, protože databázový server spouští časté kontrolní body.
- Inteligentní velké objekty v prostorech blobspace nejsou protokolovány, obsahuje je ale záloha protokolu, ve které byl každý objekt vytvořen. To znamená, že objekty nejsou uvolněny, dokud server neprovede zálohu protokolu, ve kterém byly vytvořeny. Pokud jsou tedy inteligentní velké objekty v prostorech blobspace často aktualizované, bude pravděpodobně potřeba provádět častější zálohování protokolů, aby se získal v prostoru blobspace volný prostor navíc.
- Aplikacím generujícím malé množství dat protokolu na začátku přiřadíte 10 souborů protokolu, každý o velikosti 10 MB.
- Aplikacím generujícím velké množství dat protokolu na začátku přiřadíte 10 souborů protokolu, každý o velikosti 100 MB.

Existují dvě možnosti, jak udržet zásady RPO, které určují toleranci ztráty dat v případě katastrofické události, jako je například ztráta datového serveru:

- Jednou z možností, jak udržet zásady RPO, je použít automatické zálohování protokolu, které spustí zálohování protokolu, vždy když se souboru protokolu zaplní. Tím se omezí ztráta dat v důsledku transakcí obsažených v souboru protokolu během zálohy a také jakýchkoliv dalších transakcí, které se v průběhu zálohování protokolu objevily.
- Další možností, jak udržet zásady RPO je použít plánovač. Můžete vytvořit úlohu, která bude automaticky v časovaných intervalech zálohovat jakákoliv nová data protokolu od doby poslední zálohy protokolu. Tím se omezí ztráta dat v důsledku transakcí, které nebyly mezi časovými intervaly zálohovány. Informace o použití plánovače naleznete v příručce “Plánovač” na stránce 26-5.

Pokud jsou vyžadovány zásady RPO, můžete použít plánovač a vložit úlohu, která se s požadovanou frekvencí vykoná. Zásady tak budou zachovány. Tím se v určité časové okamžiky v denním cyklu automaticky provedou zálohy souborů protokolu. Místo protokolu by mělo být vyplněno dříve než proběhne záloha protokolu a jeho recyklace. Můžete proto provést zálohu protokolů a přidat nový soubor protokolu, aby mohlo pokračovat zpracování transakcí, nebo můžete použít plánovač a přidat novou úlohu, která tuto situaci zjistí a provede obě operace automaticky.

Soubory protokolu můžete přidat kdykoliv; databázový server automaticky přidává soubory protokolu, pokud jsou vyžadovány pro konzistenci transakcí, například, pro dlouhé transakce, které by mohly spotřebovávat velké množství místa protokolu.

Nejjednodušší způsob, jak zvětšit velikost prostoru logického protokolu je přidat další soubor logického protokolu. Další informace naleznete v části “Ruční přidání souborů logického protokolu” na stránce 14-11.

Následující výraz poskytuje příklad doporučené konfigurace celkového prostoru protokolu v kilobajtech:

```
LOGSIZE = (((připojení * max_poč_řádků) * velikost_řádku) / 1024) / LOGFILES
```

Prvky výrazu	Popis
LOGSIZE	Určuje velikost každého souboru logického protokolu v kilobajtech.
<i>připojení</i>	Určuje maximální počet připojení pro všechny typy sítě, které zadáváte v souboru nebo registru sqlhosts a v parametru NETTYPE. Pokud jste v konfiguračním souboru nakonfigurovali více než jedno připojení nastavením několika konfiguračních parametrů NETTYPE, sečtete pole uživatelů pro každý parametr NETTYPE a tímto součtem nahradíte hodnotu <i>připojení</i> v předchozím vzorci.
<i>max_počet_řádků</i>	Určuje největší počet řádků, který bude aktualizován jedinou transakcí.
<i>velikost_řádku</i>	Určuje průměrnou velikost řádku tabulky v bajtech. Pokud chcete vypočítat <i>velikost_řádku</i> , připočtete délku sloupců v řádku (z tabulky systémového katalogu syscolumns).
1024	Převádí hodnotu LOGSIZE na kilobajty.
LOGFILES	Určuje počet souborů logického protokolu.

Odhad velikosti protokolu při protokolování inteligentních velkých objektů

Pokud plánujete protokolovat uživatelská data inteligentních velkých objektů, musíte zajistit, aby velikost protokolu byla *značně* větší, než objem zapisovaných dat. Pokud ukládáte inteligentní velké objekty do standardních prostorů sbpace, budou vždy protokolována metadata, i když inteligentní velké objekty protokolovány nejsou. Jestliže ukládáte inteligentní velké objekty do dočasných prostorů sbpace, neprovádí se žádné protokolování.

Odhad počtu souborů logického protokolu

Parametr LOGFILES poskytuje počet souborů logického protokolu při inicializaci systému nebo při restartu. Pokud mají všechny soubory logického protokolu stejnou velikost, můžete vypočítat celkový prostor přidělený souborům logického protokolu následujícím způsobem: celkový prostor logického protokolu = LOGFILES * LOGSIZE

Pokud databázový server obsahuje soubory protokolu různých velikostí, nemůžete k výpočtu velikosti logického protokolu použít výše uvedený výraz (LOGFILES * LOGSIZE). Místo toho musíte sečíst velikosti jednotlivých souborů protokolu na disku. Použijte pole **size** ve výstupu příkazu **onstat -l**. Další informace naleznete v části “onstat -l” na stránce 14-8.

Další informace o parametrech LOGSIZE, LOGFILES a NETTYPE naleznete v kapitole týkající se konfiguračních parametrů v příručce *IBM Informix Administrator's Reference*.

Zálohování logických souborů

Logické protokoly obsahují historii transakcí, které byly provedeny. Proces kopírování souboru logického protokolu na médium je označován jako *zálohování* souboru logického protokolu. Zálohováním souborů logického protokolu dosahujete následujících dvou cílů:

- Záznamy logického protokolu jsou ukládány na média, takže je možné v případě potřeby obnovy dat přehrát protokol.
- Prostor souboru logického protokolu se stává dostupným pro nové záznamy logického protokolu.
Pokud zanedbáte zálohování souborů protokolu, může dojít k vyčerpání prostoru protokolu.

Zálohování logického protokolu můžete spustit ručně nebo můžete nastavit průběžné zálohování logického protokolu. Po obnovení paměťových prostorů musíte obnovit logické

protokoly, abyste uvedli data do konzistentního stavu. Další informace o zálohování protokolů naleznete v příručce *IBM Informix Backup and Restore Guide*.

Zálohování prostorů blobspace

Nezáleží na tom, zda zálohujete dříve logické protokoly nebo soubory blobspace.

Zálohování dat prostoru blobspace:

1. Zavřete aktuální logický protokol, pokud obsahuje transakce prováděné na jednoduchých velkých objektech nebo na prostorech blobspace.
2. Co možná nejdříve po aktualizaci dat jednoduchých velkých objektů proveďte zálohu logických protokolů a prostorů blobspace.

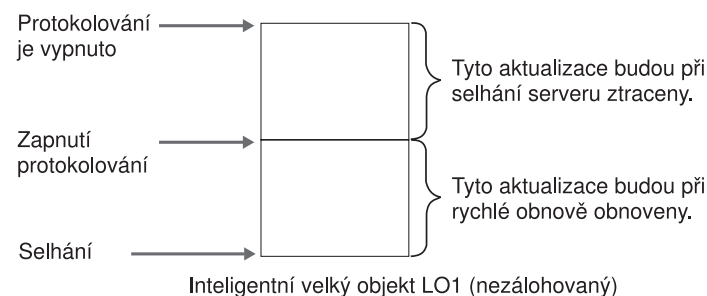
Upozornění: Pokud neprovedete zálohu těchto prostorů blobspace a logických protokolů, nebudete možná schopni obnovit data v prostoru blobspace. Jestliže budete se zálohováním čekat až na okamžik, kdy bude prostor blobspace poškozen, nebude moci databázový server získat přístup k prostoru blobspace a nebude moci zkopírovat změněná data do logického protokolu.

Zálohování prostorů sbpace

Když u inteligentních velkých objektů zapnete zálohování, musíte provést zálohu úrovně 0 prostoru sbpace.

Obrázek 14-1 znázorňuje, co se stane, pokud zapnete protokolování prostoru sbpace a neprovedete jeho zálohování. Neprotokolované změny inteligentního velkého objektu **LO1** jsou při selhání ztraceny, ačkoli protokolované změny je možné obnovit. Objekt **LO1** nebude možné zcela obnovit.

Během rychlé obnovy přehrává databázový server všechny potvrzené transakce pro objekt **LO1**. Pokud není objekt **LO1** protokolován, nebude databázový server schopen odvolat nepotvrzené transakce. Potom by byl obsah objektu **LO1** nesprávný. Další informace uvádí část “Rychlá obnova” na stránce 15-7.



Obrázek 14-1. Zapnutí zálohování prostoru sbpace

Přepnutí na následující soubor logického protokolu

V některých případech budete možná chtít přepnout na následující soubor logického protokolu dříve, než se aktuální soubor protokolu zaplní. Důvody mohou být následující:

- Aby bylo možné provést zálohu aktuálního protokolu.
- Aby se provedla aktivace nových prostorů blobspace a bloků blobspace.

Databázový server může být při provedení této změny v režimu online. Chcete-li přepnout na následující dostupný soubor protokolu, spusíte tento příkaz:

onmode -l

Změna bude platná okamžitě. (Zkontrolujte, zda jste na příkazový řádek napsali malé písmeno L, nikoli číslici 1.)

Uvolnění souboru logického protokolu

Pokud je soubor protokolu nově přidán (stav **A**), je okamžitě dostupný k použití. Může být také okamžitě vypuštěn.

Soubor logického protokolu můžete chtít uvolnit z následujících důvodů:

- Aby databázový server nezastavil zpracování.
- Aby se uvolnil prostor používaný odstraněnými stránkami blobpage.

Postupy při uvolnění souborů protokolu se liší v závislosti na stavu souboru protokolu. Jednotlivé postupy jsou popsány v následujících částech. Chcete-li zjistit stav souborů logického protokolu, prostudujte si část “Příznaky stavu souborů logického protokolu” na stránce 13-3 a “Monitorování aktivity protokolování” na stránce 14-7.

Rada: Další informace o používání programu ON-Bar nebo příkazu **ontape** k zálohování paměťových prostorů a logických protokolů naleznete v příručce *IBM Informix Backup and Restore Guide*.

Odstranění souboru protokolu se stavem D

Pokud vypustíte použitý soubor protokolu, bude označen jako odstraněný (stav **D**) a nebude jej možné použít znovu. Obslužný program **onparams** zobrazí tuto zprávu:

```
Log file číslo_souboru_protokolu has been pre-dropped. It will be
deleted from the log list and its space can be reused
once you take level 0 archives of all BLOBspaces,
Smart BLOBspaces and non-temporary DBspaces.
```

Chcete-li soubor protokolu odstranit, vytvořte zálohu úrovně 0 všech paměťových prostorů.

Uvolnění souboru protokolu se stavem U

Jestliže soubor protokolu obsahuje záznamy, ale nebyl dosud zálohován, (stav **U**), vytvořte zálohu tohoto souboru pomocí nástroje pro zálohování, který obvykle používáte.

Pokud provedení zálohy logického souboru nezmění stav na hodnotu volný (**F**), změní se jeho stav buď na hodnotu **U-B**, nebo na hodnotu **U-B-L**. Další informace naleznete v následující části “Uvolnění souboru protokolu se stavem U-B nebo F” nebo v části “Uvolnění souboru protokolu se stavem U-B-L” na stránce 14-7.

Uvolnění souboru protokolu se stavem U-B nebo F

Pokud byl soubor protokolu zálohován, ale je stále používán (stav **U-B**), znamená to, že se některé transakce v souboru protokolu stále ještě provádějí nebo soubor protokolu obsahuje nejstarší aktualizaci, která je požadována pro rychlou obnovu. Jelikož byl soubor protokolu se stavem **F** používán v minulosti, řídí se stejnými pravidly jako stav **U-B**.

Postup uvolnění zálohovaného souboru protokolu, který je používán:

1. Pokud nechcete čekat na dokončení transakcí, uveďte databázový server do klidového režimu. Další informace naleznete v části “Okamžitá změna z režimu online do klidového režimu” na stránce 4-13. Všechny aktivní transakce budou odvolány.
2. Chcete-li vynutit kontrolní bod, použijte příkaz **onmode -c**. Je potřeba to udělat, protože soubor protokolu se stavem **U-B** může obsahovat nejstarší aktualizaci.

Logický soubor, který byl zálohován, ale *není* používán (stav **U-B**), nepotřebuje být uvolněn. V následujícím příkladu protokol 34 nemusí být uvolněn, zatímco protokoly 35 a 36 uvolněny být musí. Protokol 35 obsahuje poslední kontrolní bod a protokol 36 je zálohován, ale stále používán.

34 U-B-- Protokol je použitý, byl zálohován a není používán.
35 U-B-L Protokol je použitý, byl zálohován a obsahuje poslední kontrolní bod.
36 U-B-- Protokol je použitý, byl zálohován a není používán.
37 U-C-- Toto je aktuální soubor protokolu, nebyl zálohován.

Rada: Logický protokol se stavem U-B (a nikoli se stavem L) je možné uvolnit pouze tehdy, pokud do něj nezasahuje aktivní transakce a pokud neobsahuje nejstarší aktualizaci.

Uvolnění souboru protokolu se stavem U-C nebo U-C-L

Při uvolňování aktuálního souboru protokolu postupujte následovně:

Postup uvolnění aktuálního souboru protokolu (stav C):

1. Přepněte aktuální soubor protokolu na následující dostupný soubor protokolu následujícím příkazem:
onmode -l
2. Vytvořte zálohu původního souboru protokolu pomocí programu ON-Bar nebo příkazu **ontape**.
3. Poté, co jsou všechny zaplněné soubory protokolu zálohovány, budete vyzváni k přepnutí na následující dostupný soubor protokolu a k zálohování nového aktuálního souboru protokolu.
Toto zálohování nebudete muset provádět, protože jste právě přepnuli na tento soubor protokolu.

Pokud bude po uvolnění aktuálního souboru protokolu stav souboru protokolu označen jako **U-B** nebo **U-B-L**, vyhledejte další informace v části “Uvolnění souboru protokolu se stavem U-B nebo F” na stránce 14-6 nebo “Uvolnění souboru protokolu se stavem U-B-L”.

Uvolnění souboru protokolu se stavem U-B-L

Pokud je soubor zálohován a všechny transakce v tomto souboru jsou uzavřeny, ale soubor není uvolněný (stav **U-B-L**), obsahuje tento soubor logického protokolu záznam nejnovějšího kontrolního bodu.

Chcete-li uvolnit soubory protokolu se stavem **U-B-L**, musí databázový server vytvořit nový kontrolní bod. Pomocí následujícího příkazu můžete vynutit vytvoření kontrolního bodu:

```
onmode -c
```

Jen pro UNIX

Pokud chcete vynutit kontrolní bod pomocí programu ON-Monitor, vyberte volbu **Force-Ckpt**.

Konec Jen pro UNIX

Monitorování aktivity protokolování

Pomocí monitorování souborů logického protokolu můžete určit celkový dostupný prostor (ve všech souborech) a stav souboru (například, zda byl soubor protokolu již zálohován). Další informace týkající se monitorování vyrovnávacích pamětí logického protokolu naleznete v části “Monitorování aktivity fyzického a logického protokolování” na stránce 16-2.

Monitorování logického protokolu za účelem zjištění jeho zaplnění

K monitorování souborů logického protokolu můžete použít níže uvedené obslužné programy spuštěné z příkazového řádku.

onstat -l

Příkaz **onstat -l** zobrazí informace o fyzických a logických protokolech.

Výstupní část obsahující informace o každém souboru logického protokolu obsahuje následující informace:

- Adresa deskriptoru souboru logického protokolu.
- Číslo souboru protokolu.
- Příznaky stavu, které oznamují stav každého protokolu (volný, zálohovaný, aktuální, atd.)
- Jedinečný ID souboru protokolu.
- Počáteční stránka souboru.
- Velikost souboru ve stránkách, počet použitých stránek a procentní hodnota využití stránek.

Čísla souborů protokolu v poli **numbers** nemusí jít za sebou, pokud vypustíte některé protokoly z prostřední části seznamu nebo pokud databázový server dynamicky přidá soubory protokolu.

Další informace o programu a příklad výstupu z **onstat -l** naleznete v příručce *IBM Informix Administrator's Reference*.

oncheck -pr

Databázový server ukládá informace o souborech logického protokolu na rezervovaných stránkách vyhrazených informacím o kontrolním bodu. Jelikož databázový server aktualizuje tyto informace pouze během kontrolního bodu, nejsou tak aktuální, jako informace, které zobrazuje volba **onstat -l**. Další informace o používání těchto voleb k zobrazení informací rezervovaných stránek naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Rezervované stránky kontrolního bodu je možné zobrazit pomocí příkazu **oncheck -pr**. Obrázek 14-2 ukazuje vzorový výstup pro jeden ze souborů logického protokolu.

```
...
Log file number          1
Unique identifier        7
Log contains last checkpoint Page 0, byte 272
Log file flags           0x3  Log file in use
                          Current log file
Physical location        0x1004ef
Log size                 750 (p)
Number pages used        1
Date/Time file filled    01/29/2001 14:48:32
...
```

Obrázek 14-2. Výstup příkazu **oncheck -pr** obsahující informace o souboru logického protokolu.

Monitorování dočasných logických protokolů

Databázový server používá *dočasné logické protokoly* k přehrání transakcí během teplého obnovení, protože trvalé protokoly nejsou v tomto okamžiku dostupné. Když se přehrání protokolu dokončí, databázový server uvolní dočasné soubory protokolu. Pokud během teplého obnovení vydáte příkaz **onstat -l**, bude výstup zahrnovat čtvrtou část týkající se

dočasných souborů protokolu, která bude mít stejný formát jako část zaměřená na běžné soubory protokolu. Dočasné soubory protokolu používají pouze příznaky stavu **B**, **C**, **F** a **U**.

Použití tabulek SMI

Pomocí dotazů na tabulku **syslogs** získáte informace o souborech logického protokolu. Tato tabulka obsahuje řádky pro každý soubor logického protokolu. Tato tabulka má následující sloupce:

sloupec	Popis
number	Identifikace počtu souborů logického protokolu.
uniqid	Jedinečný identifikátor souboru protokolu.
size	Velikost soubor ve stránkách.
used	Počet použitých stránek.
is_used	Příznak, který označuje, zda je soubor protokolu používán.
is_current	Příznak, který označuje, zda je soubor protokolu aktuální.
is_backed_up	Příznak, který označuje, zda byl soubor protokolu zálohován.
is_new	Příznak, který označuje, zda byl logický soubor přidán od poslední zálohy paměťových prostorů.
is_archived	Příznak, který označuje, zda byl soubor protokolu zapsán na archivní pásku.
is_temp	Příznak, který označuje, zda má soubor příznak dočasného protokolu.

Použití programu ON-Monitor (systém UNIX)

Volba **Status > Logs** zobrazuje téměř stejné informace o souborech logického protokolu jako volba **onstat -l**. Kromě toho obsahuje sloupec, který udává prostor dbspace, ve kterém je umístěn každý soubor logického protokolu.

Monitorování stavu zálohy protokolů

K monitorování stavu protokolů a zobrazení, které protokoly byly zálohovány, použijte příkaz **onstat -l**. Příznak stavu **B** označuje, že byl protokol zálohován.

Přidělení souborů protokolu

Při inicializaci nebo restartu databázového serveru se vytváří takový počet souborů logického protokolu, jaký jste zadali v konfiguračním parametru **LOGFILES**. Tyto soubory protokolu mají velikost, kterou jste zadali v parametru **LOGSIZE**.

Dynamické přidávání protokolů

Konfigurační parametr **DYNAMIC_LOGS** určuje, kdy databázový server dynamicky přidá soubor logického protokolu. Pokud pro parametr **DYNAMIC_LOGS** použijete výchozí hodnotu **2**, přidá databázový server nový soubor protokolu a způsobí výstrahu, pokud bude následující soubor protokolu obsahovat začátek nejstarší otevřené transakce.

Databázový server zkontroluje prostor logického protokolu v těchto bodech:

- Po přepnutí na nový soubor protokolu.
- Na začátku fáze vyčištění transakcí při logické obnově.

Pokud je parametr **DYNAMIC_LOGS** nastaven na hodnotu **1** a následující aktivní soubor protokolu obsahuje záznamy otevřené transakce, databázový server vás vyzve k ručnímu

přidání souboru protokolu a způsobí výstrahu. Poté, co soubor protokolu přidáte, bude databázový server pokračovat ve zpracování transakce.

Pokud je parametr DYNAMIC_LOGS nastaven na hodnotu 0 a logický protokol při odvolávání dlouhé transakce vyčerpá celý prostor, může se databázový server dostat do stavu, kdy přestane reagovat. (Dlouhá transakce brání prvním souborům logického protokolu v uvolnění, po kterém by jej bylo možno znovu použít.) Nastalou situaci opravíte nastavením parametru DYNAMIC_LOGS na hodnotu 2 a restartováním databázového serveru. Potom bude možné dlouhou transakci dokončit.

Další informace naleznete v části “Monitorování událostí dynamického přidávání protokolů” na stránce 14-16 a v části “Nastavení horních mezí pro odvolání dlouhých transakcí” na stránce 14-17.

Velikost a počet dynamicky přidávaných souborů protokolu

Při dynamickém přidávání souboru protokolu používá databázový server k výpočtu velikosti souboru protokolu tyto faktory:

- průměrnou velikost protokolu,
- objem souvislého dostupného prostoru.

Pokud má logický soubor málo prostoru, databázový server přidá tolik souborů protokolu, kolik je potřeba pro dokončení transakce. Počet souborů protokolu je omezen těmito faktory:

- maximálním podporovaným počtem souborů protokolu,
- objemem diskového prostoru pro soubory protokolu,
- objemem volného souvislého prostoru v kořenovém bloku.

Pokud databázový server zastaví přidávání nových souborů protokolu kvůli nedostatku prostoru na disku, zapíše chybovou zprávu a způsobí výstrahu. Přidejte k existujícímu prostoru dbspace další prostor dbspace nebo blok. Potom bude databázový server automaticky pokračovat ve zpracování transakce.

Na rezervovaných stránkách v kořenovém bloku jsou uloženy informace o každém souboru protokolu. Oblasti, které obsahují tyto informace, se rozšiřují spolu s tím, jak jsou přidávány další soubory protokolu. Kořenový blok požaduje dvě oblasti po 1,4 MB, aby mohl zaznamenávat 32 767 souborů protokolu, což je maximální podporovaný počet.

Pokud je během opětovného vrácení přidělena oblast rezervovaných stránek bloku z nekořenového bloku, pokusí se ji server vrátit do kořenového bloku. Jestliže v kořenovém bloku není dostatek prostoru, opětovné vrácení selže. V protokolu online se zobrazí zpráva obsahující požadovaný prostor. Požadovaný prostor je třeba uvolnit z kořenového bloku předtím, než se znovu pokusíte o opětovné vrácení.

Umístění dynamicky připojovaných souborů protokolu

Databázový server přiděluje soubory protokolu v prostorech dbspace v následujícím pořadí vyhledávání. Prostor dbspace se stane kritickým, pokud obsahuje soubory logického protokolu nebo fyzický protokol.

Průchod

Soubor protokolu bude přidělen

- 1 V prostoru dbspace, který obsahuje nejnovější soubory protokolu.
(Pokud je tento prostor dbspace zaplněn, databázový server vyhledá jiné prostory dbspace.)
- 2 V zrcadleném prostoru dbspace, který obsahuje soubory protokolu (vyjma kořenového prostoru dbspace).

- 3 V některém z prostorů dbspace, které již obsahují soubory protokolu (vyjma kořenového prostoru dbspace).
- 4 V prostoru dbspace, který obsahuje fyzický protokol.
- 5 V kořenovém prostoru dbspace.
- 6 V jakémkoli zrcadleném prostoru dbspace.
- 7 V libovolném prostoru dbspace.

Pokud nechcete používat toto pořadí vyhledávání při přidělování nového souboru protokolu, musíte nastavit parametr DYNAMIC_LOGS na hodnotu 1 a provést příkaz **onparams -a -i** s umístěním, které chcete použít pro nový protokol. Další informace naleznete v části “Monitorování událostí dynamického přidávání protokolů” na stránce 14-16.

Ruční přidání souborů logického protokolu

Soubory logického protokolu budete možná muset přidat ručně z následujících důvodů:

- Pokud chcete zvětšit diskový prostor přidělený logickému protokolu.
- Pokud chcete změnit velikost souborů logického protokolu.
- Pokud chcete umožnit dokončení otevřené transakce.
- Jako součást přesunu souborů logického protokolu do jiného prostoru dbspace.

Upozornění: Nemůžete provádět následující činnosti:

- Připojit soubor protokolu k prostoru blobspace nebo sbpace.
- Připojit logické nebo fyzické protokoly k prostorům dbspace, které mají jiné než výchozí velikosti stránky.

Soubory logického protokolu přidávejte postupně k libovolnému prostoru dbspace až do maximálního počtu 32767 souborů. Jakmile přidáte soubor protokolu k prostoru dbspace, stane se prostor kritickým prostorem dbspace. Soubory logického protokolu můžete přidávat v průběhu zálohování paměťových prostorů.

Existují dva způsoby, kterými můžete přidávat soubor logického protokolu:

- Na konec seznamu souborů pomocí příkazu **onparams -a** nebo programu ISA.
- Za aktuální soubor logického protokolu pomocí příkazu **onparams -a -i** nebo programu ISA.

Přidání souboru logického protokolu pomocí obslužného programu onparams:

1. Přihlaste se do operačního systému UNIX jako uživatel **informix** nebo **root** nebo se přihlaste do operačního systému Windows jako uživatel **Informix-Admin**.
2. Zajistěte, aby byl databázový server v režimu online, v režimu administrace, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.

Databázový server запиše během fáze čištění do protokolu tuto zprávu:

```
Logical recovery has reached the transaction cleanup phase.
```

3. Rozhodněte, zda chcete přidat soubor protokolu na konec seznamu souborů protokolu nebo za aktuální soubor protokolu.

Soubor protokolu můžete vložit za aktuální soubor protokolu bez ohledu na hodnotu parametru DYNAMIC_LOGS. Připojení souboru protokolu nové velikosti nezmění hodnotu parametru LOGSIZE.

- a. Následující příkaz přidá soubor logického protokolu na konec seznamu souborů protokolu v prostoru dbspace **logspace** za využití velikost souboru protokolu určené konfiguračním parametrem LOGSIZE:

```
onparams -a -d logspace
```

- b. Následující příkaz vloží 1000kB soubor logického protokolu za aktuální soubor protokolu v prostoru dbspace **logspace**:

```
onparams -a -d logspace -s 1000 -i
```
 - c. Chcete-li přidat soubor logického protokolu s novou velikostí (v tomto případě 250 kilobajtů), spusíte následující příkaz:

```
onparams -a -d logspace -s 250
```
4. Stav souborů protokolu zkontrolujete pomocí příkazu **onstat -l**. Stav nového souboru protokolu je **A** a je vždy dostupný.
 5. Až budete příště potřebovat zálohovat data, proveďte zálohu úrovně 0 kořenového prostoru dbspace a prostorů dbspace, které obsahují nové soubory protokolu.
 Ačkoli již není třeba provádět zálohování okamžitě po přidání souboru protokolu, měla by být vytvořena záloha úrovně 0, neboť se došlo ke změnám datových struktur. Další informace naleznete v příručce *IBM Informix Backup and Restore Guide*.

Další informace o používání příkazu **onparamsk** přidání souboru logického protokolu naleznete v kapitole *IBM Informix Administrator's Reference*.

Postup přidání souboru logického protokolu pomocí programu ISA:

1. Vyberte položku **Protokoly > Logický** a klepněte na položku **Přidat soubor protokolu**.
2. Pomocí příkazu **onstat -l** zkontrolujte stav souborů protokolu.
 Další informace naleznete v nápovědě online k programu ISA.

Postup přidání souboru logického protokolu pomocí programu ON-Monitor (systém UNIX):

1. Řiďte se pokyny pro přidání souboru protokolu v části “Ruční přidání souborů logického protokolu” na stránce 14-11 s tou výjimkou, že místo příkazu **onparams** použijete program ON-Monitor.
2. Chcete-li přidat soubor logického protokolu, vyberte **Parameters > Add-Log**.
3. Do pole označeného jako **Dbspace Name** zadejte název prostoru dbspace, ve kterém bude soubor logického protokolu uložen.
 Velikost souboru protokolu se automaticky zobrazí v poli **Logical Log Size**. Nový soubor protokolu má vždy velikost zadanou parametrem LOGSIZE.

Vypuštění souborů logického protokolu

Chcete-li vypustit soubor logického protokolu a zvětšit objem prostoru na disku dostupného v rámci prostoru dbspace, můžete použít příkaz **onparams** nebo program ISA. Databázový server vyžaduje vždy minimálně tři soubory logického protokolu. Nemůžete vypustit protokol, pokud je logický protokol tvořen pouze třemi soubory protokolu.

Pravidla pro odstraňování souborů protokolu se změnila následujícím způsobem:

- Pokud vypouštíte soubor protokolu, do kterého nebylo nikdy zapisováno (stav **A**), databázový server jej odstraní a okamžitě uvolní prostor.
- Pokud vypouštíte použitý soubor protokolu (stav **U-B**), databázový server ho označí jako odstraněný (**D**). Poté, co provedete zálohu úrovně 0 prostorů dbspace, které obsahují soubory protokolu, a kořenový prostor dbspace, uvolní databázový server tento prostor.
- Nemůžete vypustit soubor protokolu, který je aktuálně používán nebo obsahuje záznam posledního kontrolního bodu (stav **C** nebo **L**).

Postup vypuštění souboru logického protokolu pomocí příkazu onparams:

1. Zajistěte, aby byl databázový server v režimu online, v režimu administrace nebo v klidovém režimu.

2. Pomocí následujícího příkazu vypustíte soubor logického protokolu, jehož číslo souboru protokolu je 21:

```
onparams -d -1 21
```

Soubory protokolu vypouštějte postupně. Musíte znát číslo souboru protokolu každého logického protokolu, který chcete vypustit.
3. Pokud má soubor protokolu stav Nově přidán (**A**), bude vypuštěn okamžitě.
Pokud má soubor protokolu stav Používaný (**U**), bude označen jako Odstraněný (**D**).
4. Chcete-li vypustit používané soubory protokolu, proveďte zálohu úrovně 0 všech prostorů dbspace.
Tato záloha zabrání databázovému serveru v tom, aby při obnově použil vypuštěné soubory protokolu, a zajistí, aby rezervované stránky obsahovaly informace o aktuálním počtu souborů protokolu.

Další informace o používání příkazu **onparams** k vypuštění souboru logického protokolu naleznete v kapitole *IBM Informix Administrator's Reference*.

Další informace o používání příkazu **onlog** k zobrazení souborů logického protokolu a jedinečných identifikačních čísel naleznete v části “Zobrazení záznamů logického protokolu” na stránce 14-16.

Postup vypuštění souboru logického protokolu pomocí programu ON_Monitor (systém UNIX):

1. Zajistěte, aby byl databázový server v režimu online, v režimu administrace nebo v klidovém režimu.
2. Soubor logického protokolu vypustíte vybráním položky **Parameters > Drop-Log**.
3. Pokud má soubor protokolu stav Nově přidán (**A**), bude vypuštěn okamžitě.
Pokud má soubor protokolu stav Používaný (**U**), bude označen jako Odstraněný (**D**).
4. Chcete-li vypustit používané soubory protokolu, proveďte zálohu úrovně 0 všech prostorů dbspace.

Rada: Pokud nebyl kořenový prostor dbspace nikdy zálohovaný, můžete vypustit **používaný** soubor protokolu okamžitě.

Změna velikosti souborů logického protokolu

Pokud chcete změnit velikost souborů protokolu, je jednodušší přidat nové soubory protokolu požadované velikosti a potom vypustit staré soubory protokolu. Velikost souborů logického protokolu můžete změnit následujícími způsoby:

- Pomocí příkazu **onparams** s volbou **-s** přidejte nový soubor protokolu odlišné velikosti.
Další informace naleznete v části “Ruční přidání souborů logického protokolu” na stránce 14-11.
- Zvětšíte hodnotu LOGSIZE v souboru ONCONFIG, pokud chcete, aby databázový server vytvořil větší soubory protokolu.
Další informace naleznete v části “Změna konfiguračních parametrů protokolování” na stránce 14-14.

Přesun souboru logického protokolu do jiného prostoru dbspace

Možná budete chtít přesunout soubor logického protokolu z důvodů výkonu nebo za účelem získání více prostoru v prostoru dbspace, jak bylo vysvětleno v části “Umístění souborů logického protokolu” na stránce 13-2. Další informace užitečné při hledání umístění souborů logického protokolu naleznete v části “Monitorování aktivity protokolování” na stránce 14-7. Ačkoli přesouvání souborů logického protokolu není obtížné, může být časově náročné.

Přesouvání souborů logického protokolu je kombinací dvou jednoduchých akcí:

- Vypuštění souborů logického protokolu z prostoru dbspace, ve kterém se aktuálně nacházejí.
- Přidání souborů logického protokolu do jiného prostoru dbspace.

Následující postup je příkladem přesunu šesti souborů logického protokolu z **kořenového** prostoru dbspace do jiného prostoru dbspace (do prostoru **dbspace_1**).

Upozornění: Nelze přesouvat soubory logického a fyzického protokolu v prostorech dbspace, které nemají výchozí velikost stránky.

Postup přesunu souborů logického protokolu z kořenového prostoru dbspace (příklad):

1. Zajistěte, aby byl databázový server online, v režimu administrace, v klidovém režimu, případně v režimu rychlé obnovy.
2. Do prostoru **dbspace_1** přidejte šest souborů logického protokolu.
Další informace naleznete v části “Ruční přidání souborů logického protokolu” na stránce 14-11.
3. Proveďte zálohu úrovně 0 všech paměťových prostorů, abyste uvolnili všechny soubory protokolu kromě aktuálního souboru protokolu.
(Pokud použijete příkaz **onbar -l -b -c**, budete zálohovat všechny soubory protokolu včetně aktuálního souboru protokolu.) Viz kapitola “Uvolnění souboru logického protokolu” na stránce 14-6.
4. Pomocí příkazu **onmode -l** přepněte na nový aktuální soubor protokolu.
Další informace naleznete v části “Přepnutí na následující soubor logického protokolu” na stránce 14-5.
5. Vypusťte všech šest souborů logického protokolu z kořenového prostoru dbspace.
Nelze vypustit aktuální soubor logického protokolu.
Další informace naleznete v části “Vypuštění souborů logického protokolu” na stránce 14-12.
6. Vytvořte zálohu úrovně 0 kořenového prostoru dbspace a prostoru **dbspace_1**.
Další informace naleznete v příručce *IBM Informix Backup and Restore Guide*.

Změna konfiguračních parametrů protokolování

Ke změně parametrů ONCONFIG můžete použít textový editor nebo program ISA. V následující tabulce jsou uvedeny konfigurační parametry logických protokolů. Další informace najdete v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.

Konfigurační parametr	Minimální hodnota	Výchozí hodnota	Maximální hodnota
DYNAMIC_LOGS	0 nebo 1	2	2
LOGBUFF	2 * velikost stránky	64 kB	Hodnota parametru LOGSIZE
LOGFILES	3 soubory	6 souborů	32 767 souborů
LOGSIZE	1500 kB v systému UNIX 500 kB v operačním systému Windows	2000 kB	Viz příručka <i>IBM Informix Administrator's Reference</i>
LTXEHWM	Hodnota parametru LTXHWM	90 %	100 %
LTXHWM	1 %	80 %	100 %

Důležité: Změna parametru LOGFILES nebude platná, dokud znovu neinicializujete nebo nerestartujete diskový prostor.

Postup změny konfiguračních parametrů logického protokolu v souboru ONCONFIG:

1. Uveďte databázový server do režimu offline, do klidového režimu nebo do režimu administrace.
2. Pomocí programu ISA nebo textového editoru aktualizujte konfigurační parametry. Parametry DYNAMIC_LOGS, LTXHWM a LTXEHWM se v souboru **onconfig.std** nenacházejí. Pokud chcete změnit hodnoty těchto parametrů, přidejte je do souboru ONCONFIG.
3. Vypněte a restartujte databázový server.
4. Tento krok provádějte pouze tehdy, pokud měníte parametr LOGFILES a chcete, aby byly všechny soubory protokolu v souvislém prostoru. Parametr LOGFILES obvykle přidáte a vypustíte pomocí obslužného **onparams** utility.)
 - a. Uvolněte všechna data databázového serveru. Je potřeba to udělat, protože při uvolňování a obnovení dat nemůžete spoléhat na zálohy paměťových prostorů, protože obnovení vrátí tyto parametry na jejich předchozí hodnoty.
 - b. Znovu inicializujte prostor na disku databázového serveru. Další informace naleznete v části "Inicializace diskového prostoru" na stránce 4-2.
 - c. Znovu vytvořte všechny databáze a tabulky.
 - d. Znovu zaveďte všechna data databázového serveru.
Další informace o zavádění a uvolňování dat naleznete v příručce *IBM Informix Migration Guide*.
5. Proveďte zálohu kořenového prostoru dbspace, aby byly změněné logické protokoly povoleny.

Použití programu ON-Monitor ke změně parametru LOGFILES (systém UNIX)

Ke změně hodnot operačního parametru LOGFILES můžete použít program ON-Monitor.

Postup změny těchto hodnot:

1. Uvolněte všechna data databázového serveru.
Při uvolňování a obnovení dat nemůžete spoléhat na zálohy paměťových prostorů, protože obnovení vrátí tyto parametry na jejich předchozí hodnoty.

2. Vyberte položku **Parameters > Initialize** aby se znovu inicializoval prostor na disku, nebo se změnilly parametry prostoru na disku.
3. Změňte hodnotu parametru LOGSIZE v poli označeném **Log.Log Size** nebo změňte hodnotu parametru LOGFILES v poli označeném **Number of Logical Logs**.
4. Pokračujte v inicializaci prostoru na disku databázového serveru.
5. Znovu vytvořte všechny databáze a tabulky.
6. Znovu zaveďte všechna data databázového serveru.

Další informace o zavádění a uvolňování dat naleznete v příručce *IBM Informix Migration Guide*.

Zobrazení záznamů logického protokolu

K zobrazení a interpretaci záznamů logického protokolu je možné použít obslužný program **onlog**. Informace o použití příkazu **onlog** naleznete v příručce *IBM Informix Administrator's Reference*.

Monitorování událostí dynamického přidávání protokolů

Provádějte monitorování následujících alarmů událostí, které jsou spouštěny dynamickým přidáním souborů protokolu (viz Tabulka 14-1). Když je alarm spuštěn, zapíše se zpráva do protokolu zpráv. Další informace naleznete v kapitole týkající se alarmů zpráv a konfiguračních parametrů v příručce *IBM Informix Administrator's Reference*.

Příkaz **onparams**, který bude přidávat soubory protokolu, můžete zahrnout do skriptu pro alarmy událostí s ID třídy události 27 (je požadován soubor protokolu). Skript také může provést příkaz **onstat -d**, aby vyhledal odpovídající prostor a provedl příkaz **onparams -a -i** s umístěním, ve kterém se nachází dostatek prostoru. Chcete-li přidat nový protokol přímo za aktuální soubor protokolu, musíte použít volbu **-i**.

Tabulka 14-1. Alarmy událostí dynamického přidávání souborů protokolu

ID třídy	Závažnost	Zpráva třídy	Zpráva
26	3	Dynamicky přidán soubor protokolu <i>číslo_protokolu</i>	Tato zpráva se zobrazí, když databázový server dynamicky přidá soubor protokolu. Dynamicky přidáný soubor protokolu <i>číslo_protokolu</i> do prostoru dbspace <i>číslo_prostoru_dbspace</i> .
27	4	Požadován soubor protokolu	Tato zpráva se zobrazí, když je parametr DYNAMIC_LOGS nastaven na hodnotu 1 a databázový server čeká na vás, aby mohl přidat soubor protokolu. Varování: Nejstarší logický protokol <i>číslo_protokolu</i> obsahuje záznamy otevřené transakce <i>adresa_transakce</i> . Logické protokolování zůstane zablokované, dokud nebude přidán soubor protokolu. Přidejte soubor protokolu pomocí příkazu onparams -a s použitím volby -i (vlození) následujícím způsobem: onparams -a -d prostor_dbspace -s velikost-i Potom co možná nejdříve dokončete transakci.
28	4	Není prostor pro soubor protokolu.	Varování: Protože nejstarší logický protokol <i>číslo_protokolu</i> obsahuje záznamy otevřené transakce <i>adresa_transakce</i> , pokouší se server dynamicky přidat soubor protokolu. K dispozici však není žádný prostor. Přidejte prostor dbspace nebo blok. Potom co možná nejdříve dokončete transakci.

Tabulka 14-2 obsahuje akce, které databázový server podnikne při jednotlivých nastaveních konfiguračního parametru DYNAMIC_LOGS.

Tabulka 14-2. Nastavení konfiguračního parametru DYNAMIC_LOGS

DYNAMIC_LOGS	Význam	Alarm události	Čekat na přidání protokolu	Dynamicky přidat protokol
2 (výchozí hodnota)	Umožňuje automatické přidělení nových souborů protokolu, aby otevřené transakce nemohly uvést systém do stavu, kdy přestane reagovat.	Ano (26, 28)	Ne	Ano
1	Umožňuje ruční přidání nových souborů protokolu.	Ano (27)	Ano	Ne
0	Ne přiděluje soubory protokolu, ale vydá následující zprávu týkající se otevřených transakcí: Varování: Nejstarší soubor logického protokolu <i>číslo_protokolu</i> obsahuje záznamy otevřené transakce <i>adresa_transakce</i> , ale funkce dynamického protokolu je vypnutá.	Ne	Ne	Ne

Nastavení horních mezí pro odvolání dlouhých transakcí

Databázový server používá konfigurační parametry LTXHWM a LTXEHW M k nastavení horních mezí pro dlouhé transakce. Pokud je konfigurační parametr DYNAMIC_LOGS nastaven na hodnotu 1 nebo 2, bude výchozí hodnota konfiguračního parametru LTXHWM 80 procent a výchozí hodnota parametru bude LTXEHW M je 90 procent. Pokud je konfigurační parametr DYNAMIC_LOGS nastaven na hodnotu 0, bude výchozí hodnota konfiguračního parametru LTXHWM je 50 procent a výchozí hodnota konfiguračního parametru LTXHEWM bude 60 procent.

Pokud snížíte hodnoty horních mezí, zvýšíte pravděpodobnost dlouhých transakcí. To můžete vynahradiť přidělením dalšího prostoru protokolu. Další informace o konfiguračních parametrech LTXHWM a LTXEHW M, naleznete v kapitole týkající se konfiguračních parametrů v příručce *IBM Informix Administrator's Reference*.

Horní mez dlouhé transakce (LTXHWM)

Horní mez dlouhé transakce je procento celkového prostoru protokolu, který smí transakce zaplnit, než bude odvolána. Pokud databázový server nalezne otevřenou transakci v nejstarším používaném souboru protokolu, dynamicky přidá soubory protokolu. Protože se prostor protokolu zvětšuje, zvyšuje se i horní mez. Když prostor protokolu dosáhne horní meze, databázový server transakci odvolá. Odvolání transakce a jiné procesy rovněž generují záznamy logického protokolu. Databázový server pokračuje v přidávání souborů protokolu, dokud není odvolání transakce dokončeno, aby zabránil vyčerpání prostoru logického protokolu. Pokud existuje více dlouhých transakcí, může být odvoláno více transakcí.

Příklad: Databázový server má deset logických protokolů a konfigurační parametr LTXHWM je nastaven na hodnotu 98. Transakce začne v souboru protokolu číslo 1 a aktivita aktualizace vyplní protokoly číslo 1 až 9. Databázový server dynamicky přidá soubor protokolu číslo 11 za soubor protokolu číslo 10. Pokud transakce nebude nedokončena, bude tento proces pokračovat dokud databázový server nepřidá 40 souborů protokolu. Jakmile databázový server přidá padesátý soubor protokolu, dosáhne transakce horní meze a databázový server ji odvolá.

Horní mez dlouhé transakce s výlučným přístupem (LTXEHW)

K dosažení *horní meze dlouhé transakce s výlučným přístupem* dochází tehdy, pokud dlouhá transakce, která je odvolávána, získá *výlučný* přístup k logickému protokolu. Databázový server dramaticky omezí vytváření záznamů protokolu. K logickému protokolu mají povolen přístup pouze jednotkové procesy, které aktuálně odvolávají transakce, a jednotkové procesy aktuálně zapisující záznamy COMMIT. Omezení přístupu k logickému protokolu ušetří maximální množství prostoru pro záznamy o odvolaných transakcích, které jsou zapisovány uživatelskými jednotkovými procesy odvolávajícími transakce.

Upozornění: Pokud nastavíte oba konfigurační parametry *LTXHWM* a *LTXEHW* na hodnotu *100*, nebudou dlouhé transakce nikdy zastaveny. Pro běžný provoz databázového serveru byste měli nastavit konfigurační parametr *LTXHWM* pod hodnotu *100*. Konfigurační parametr *LTXHWM* nastavte na hodnotu *100*, pokud chcete spustit plánované transakce neznámé délky. Konfigurační parametr *LTXEHW* nastavte na hodnotu *100*, pokud nechcete nikdy během odvolávání dlouhé transakce blokovat jiné uživatele a máte dostatek prostoru na disku.

Úprava velikosti souborů protokolu za účelem zabránění dlouhých transakcí

Pokud do protokolů zapisuje mnoho uživatelů současně, používejte větší soubory protokolu. Pokud použijete malé protokoly a existuje pravděpodobnost vzniku dlouhých transakcí, snižte mezní hodnotu. Nastavte konfigurační parametr *LTXHWM* na hodnotu *50* a *LTXEHW* na hodnotu *60*.

Pokud jsou soubory protokolu příliš malé, může databázový server při odvolávání dlouhé transakce vyčerpávat prostor protokolu. V takovém případě nebude databázový server dost rychlý, aby přidal nový soubor protokolu předtím, než se poslední soubor protokolu zaplní. Pokud se poslední soubor protokolu zaplní, systém přestane reagovat a zobrazí chybovou zprávu. Problém vyřešte vypnutím a restartováním databázového serveru. Další informace naleznete v části “Obnova zablokované dlouhé transakce” na stránce 14-18.

Obnova zablokované dlouhé transakce

Pokud má systém nadbytek diskového prostoru a chcete provádět transakce neznámé délky zvažte nastavení konfiguračního parametru *LTXHWM* na hodnotu *100*, kterým přinutíte databázový server přidávat soubory protokolu, dokud nebude transakce dokončena.

Transakce může způsobit, že systém přestane reagovat, protože databázový server vyčerpá prostor na disku. Databázový server zastaví přidávání nových souborů protokolu, zapíše chybovou zprávu a způsobí výstrahu.

Jak pokračovat v transakci:

1. Chcete-li pokračovat v transakci, přidejte prostor dbspace nebo k prostoru dbspace přidejte blok.
2. Pokračujte ve zpracování transakce.

Pokud nemůžete databázovému serveru přidat další prostor na disku, transakci ukončete.

Jak ukončit transakci:

- Zadejte příkaz **onmode -z**.
- Vypněte a restartujte databázový server.

Když se databázový server spustí v režimu rychlé obnovy, bude transakce odvolána. Potom proveďte následující kroky:

Postup obnovy zablokované dlouhou transakcí:

1. Přidejte více prostoru na disku nebo další disk, dokud se transakce úspěšně neodvolá.
2. Proveďte obnovu do bodu v čase před zahájením dlouhé transakce nebo dostatečně brzy na to, aby databázový server mohl transakci odvolat.
3. Z instance databázového serveru vypusťte přebytečné soubory protokolu, prostory dbspace a bloky.
4. Proveďte zálohu úrovně 0, abyste uvolnili prostor logického protokolu.

Kapitola 15. Fyzické protokolování, kontrolní body a rychlá obnova

Obsah kapitoly	15-1
Kritická část	15-2
Fyzické protokolování	15-2
Jak rychlá obnova používá fyzicky protokolované stránky	15-2
Jak zálohování používá fyzicky protokolované stránky	15-2
Aktivity databázového serveru, které jsou fyzicky protokolovány	15-2
Zprávy fyzické obnovy	15-2
Fyzické protokolování a jednoduché velké objekty	15-3
Fyzické protokolování a inteligentní velké objekty	15-3
Velikost a umístění fyzického protokolu	15-3
Určení umístění fyzického protokolu	15-3
Postup k určení velikosti fyzického protokolu	15-3
Přetečení fyzického protokolu po vypnutí protokolování transakcí	15-4
Kontrolní body	15-5
Hodnoty LRU pro vyprázdňování společné oblasti vyrovnávací paměti mezi kontrolními body	15-6
Kontrolní body během zálohování	15-6
Rychlá obnova	15-7
Potřeba rychlé obnovy	15-7
Situace, při kterých se spouští rychlá obnova	15-7
Rychlá obnova a protokolování s vyrovnávací pamětí	15-7
Možné přetečení fyzického protokolu během rychlé obnovy	15-7
Rychlá obnova a žádné protokolování	15-8
Rychlá obnova po kontrolním bodu	15-8
Server se vrací do stavu v době posledního kontrolního bodu	15-8
Server hledá záznam kontrolního bodu v logickém protokolu	15-8
Server přehraje záznamy logického protokolu	15-9
Server odvolává nepotvrzené transakce	15-9

Obsah kapitoly

Tato kapitola popisuje tři procedury, které databázový server používá k dosažení konzistence dat:

- Fyzické protokolování.
- Kontrolní body.
- rychlá obnova

Fyzický protokol je sada stránek na pevném disku, na které databázový server ukládá nezměněnou kopii stránky nazvanou *předobraz*. *Fyzické protokolování* je proces ukládání předobrazu stránky, kterou databázový server hodlá změnit. *Kontrolní bod* označuje bod, ve kterém databázový server synchronizuje stránky na pevném disku se stránkami ve vyrovnávacích pamětech sdílené paměti. *Rychlá obnova* je proces spouštěný automaticky po neočekávaném vypnutí databázového serveru, který ho uvede zpět do konzistentního stavu.

Tyto procedury zajišťují, že více logicky souvisejících zápisů je zaznamenáváno jako jedna jednotka a zajišťují pravidelné udržování konzistence dat ve sdílené paměti a dat na pevném disku.

Další informace o úlohách správy a monitorování kontrolních bodů fyzických protokolů uvádí Kapitola 16, “Správa fyzického protokolu”, na stránce 16-1.

Kritická část

Kritická část je část kódu (strojové instrukce), kterou je nutné provést jako nedělitelnou jednotku. Kritická část zajišťuje integritu jednotkového procesu, neboť mu umožňuje provést řadu instrukcí předtím, než je uvolněn.

Fyzické protokolování

Fyzické protokolování je proces ukládání stránek, které databázový server hodlá změnit, předtím, než jsou změněné stránky skutečně zaznamenány na disk. Než databázový server změní určité stránky ve společné oblasti vyrovnávacích pamětí sdílené paměti, uloží předobrazy stránek do vyrovnávací paměti fyzického protokolu ve sdílené paměti.

Databázový server udržuje pro tyto stránky předobrazy ve vyrovnávací paměti fyzického protokolu ve sdílené paměti, dokud jeden nebo více jednotkových procesů čištění stránek nevyprázdní tyto stránky na disk. Nezměněné stránky jsou dostupné v případě, že databázový server selže nebo že je potřebuje zálohovací procedura, aby mohly poskytnout přesný snímek dat databázového serveru. Tyto snímky používá rychlá obnova a zálohy databázového serveru.

Nenastanou-li mimořádné podmínky, recykluje databázový server fyzický protokol v každém kontrolním bodě. Další informace o kontrolních bodech naleznete v části “Kontrolní body” na stránce 15-5.

Jak rychlá obnova používá fyzicky protokolované stránky

Po poruše používá databázový server předobrazy stránek k jejich obnovení na pevný disk do stavu při posledním kontrolním bodu. Potom databázový server použije záznamy logického protokolu k uvedení všech dat do fyzické a logické konzistence až do bodu poslední dokončené transakce. Podrobnější informace o této proceduře naleznete v části “Rychlá obnova” na stránce 15-7.

Jak zálohování používá fyzicky protokolované stránky

Při zálohování vytvoří databázový server kontrolní bod a koordinací s fyzickým protokolem identifikuje správnou verzi stránek, které je třeba do zálohování zahrnout. V případě zálohy úrovně 0 zálohuje databázový server všechny stránky na disku. Podrobnější informace naleznete v příručce *IBM Informix Backup and Restore Guide*.

Aktivity databázového serveru, které jsou fyzicky protokolovány

Bylo-li na stránce provedeno v době mezi kontrolními body více změn, je obvykle do fyzického protokolu zahrnut pouze první předobraz.

Fyzický protokol je cyklický protokol, ve kterém jsou obsažené stránky použity pro každý kontrolní bod jednou. Pokud je nastaven konfigurační parametr `RTO_SERVER_RESTART`, bude za účelem vyššího výkonu procesurychlé obnovy probíhat přídavné fyzické protokolování.

Zprávy fyzické obnovy

Na začátku rychlé obnovy databázový server zaprotokoluje spolu s názvem bloku a posunem následující zprávu:

```
Physical recovery started at page blok:posun.
```

Po dokončení rychlé obnovy databázový server zaprotokoluje spolu s počtem prozkoumaných a obnovených stránek následující zprávu:

Physical recovery complete: (počet prozkoumaných stránek) pages examined, (počet obnovených stránek)

Fyzické protokolování a jednoduché velké objekty

Ve fyzickém protokolu mohou být jakékoliv stránky databázového serveru, včetně jednoduchých velkých objektů v prostorech tabulek (prostory typu tblspace). Do fyzického protokolu jsou před úpravou dat ve stránce a před jejím zápisem na pevný disk zkopírovány také i stránky overhead (jako jsou stránky chunk free-list, stránky blobspace free-map a stránky blobspace bit-map).

Stránky blobpage prostoru blobspace nejsou protokolovány do fyzického protokolu. Další informace o prostoru blobspace naleznete zde: “Protokolování prostorů blobspace a jednoduché velké objekty” na stránce 13-6.

Fyzické protokolování a inteligentní velké objekty

Část uživatelských dat inteligentních velkých objektů není fyzicky protokolována. Metadata se ovšem fyzicky protokolují. Další informace o inteligentních velkých objektech naleznete v části “Prostory sbpace” na stránce 9-12.

Velikost a umístění fyzického protokolu

Tato část popisuje, jak konfigurovat velikost a umístění fyzického protokolu.

Určení umístění fyzického protokolu

Když databázový server inicializuje diskový prostor, umísťuje soubory logického protokolu a fyzický protokol do kořenového prostoru dbspace. Nad tímto umístěním nemáte žádnou počáteční kontrolu. Chcete-li zlepšit výkon (zvláště snížit počet zápisu do kořenového prostoru dbspace a minimalizovat kolize požadavků na pevné disky) můžete přesunout fyzický protokol do jiného prostoru dbspace, nejlépe na disk, který neobsahuje aktivní tabulky ani soubory logického protokolu.

Doporučení: Kritické prostory typu dbspace vyhledávejte v paměťovém zařízení, které je tolerantní vůči poruchám. Pokud paměťové zařízení, ve kterém je uložen fyzický protokol, není tolerantní vůči chybám, použijte pro prostor typu dbspace, který obsahuje fyzický protokol, dynamické zrcadlení serveru. Tímto v případě selhání paměťového zařízení ochráníte databázi.

Postup k určení velikosti fyzického protokolu

Velikost fyzického protokolu závisí na dvou faktorech:

- Četnost s jakou transakce vytvářejí aktivitu fyzického protokolu
- Nastavení nebo nenastavení konfiguračního parametru RTO_SERVER_RESTART

Četnost s jakou transakce vytvářejí aktivitu fyzického protokolu může postihnout výkon tvorby kontrolních bodů. Pokud je při zpracovávání kontrolního bodu fyzický protokol v důsledku pokračování tvorby dat transakcemi přeplněný, zablokuje databázový server transakce, aby umožnil kontrolní bod dokončit a vyhnul se přetečení fyzického protokolu.

Chcete-li se vyhnout blokování transakcí, musí mít databázový server dostatek prostoru pro fyzický protokol, aby se do něj vešly všechny data z transakcí, které proběhnou během zpracovávání kontrolního bodu. Tvorba kontrolního bodu je spuštěna když je fyzický protokol ze 75 % plný. Zpracování kontrolního bodu tedy musí být dokončeno než se zaplní zbylých 25%. Transakce jsou blokovány, jakmile systém zjistí hrozbu přetečení fyzického protokolu, protože každá aktivní transakce může být zdrojem aktivity fyzického protokolu.

Řekněme například, že máte fyzický protokol o velikosti 1 GB a 1000 aktivních transakcí. 1000 aktivních transakcí má za předpokladu, že každá z těchto transakcí je v kritické fázi

současně, potenciál vytvořit zhruba 80 MB dat pro fyzický protokol. Když se zaplní 750 MB z fyzického protokolu, spustí databázový server vytváření kontrolního bodu. Pokud není v době zaplnění 920 MB z fyzického protokolu tvorba kontrolního bodu dokončena, bude databázový server blokovat transakce až do jejího dokončení. Pokud dojde k blokování transakcí, nastaví databázový server automaticky častější tvorbu kontrolních bodů, aby se takové situaci příště vyhnul. Automatické tvoření kontrolních bodů lze vypnout.

Další informace o zpracování kontrolních bodů a jejich automatické tvorbě naleznete v části “Kontrolní body” na stránce 15-5.

Druhý faktor, který je třeba zvážit při odhadování velikosti fyzického protokolu, je použití konfiguračního parametru `RTO_SERVER_RESTART` k určení cílového času potřebného k provedení rychlé obnovy. Pokud není nutné brát čas provedení rychlé obnovy v potaz, není třeba povolovat konfigurační parametr `RTO_SERVER_RESTART`. Zadáte-li hodnotu pro konfigurační parametr `RTO_SERVER_RESTART`, bude transakční aktivita tvořit další data pro fyzický protokol.

Zpracování těchto dalších dat má obvykle minimální nebo dopad na výkon, nebo se to na něm vůbec neprojeví. Přídavné protokolování je používáno za účelem podpory společné oblasti vyrovnávací paměti během procesu rychlé obnovy, aby přehrání proběhlo co nejlépe. Pokud je fyzický protokol znatelně větší než velikost všech společných oblastí vyrovnávací paměti, nastávají během procesu rychlé obnovy chyby stránek a jejich vyprázdňování. Vyprázdňování a chyby stránek podstatně snižují výkon procesu rychlé obnovy a databázový server nemůže udržovat metodu danou parametrem `RTO_SERVER_RESTART`.

U systémů s méně než čtyřmi gigabajty společné oblasti vyrovnávací paměti může fyzický protokol zabírat až 110 % místa potřebného pro velikost všech společných oblastí vyrovnávací paměti dohromady. U větších společných oblastí vyrovnávací paměti začínejte se 4 GB paměti pro fyzický protokol a poté sledujte aktivitu při tvorbě kontrolních bodů. Pokud kontrolní body nastávají příliš často a zdá se, že mají dopad na výkon, zvyšte velikost fyzického protokolu.

Vzácný stav, zvaný přetečení fyzického protokolu, může nastat je-li databázový server nakonfigurován pro malou velikost fyzického protokolu a přitom má velké množství uživatelů. Budete-li dbát výše uvedených pokynů, nemělo by se přetečení fyzického protokolu přihodit. Když databázový server zjistí nevhodné konfigurace, vytváří v protokolu zpráv upozornění na špatný výkon.

Je-li zjištěna nevhodná konfigurace, můžete pomocí příkazu `onstat -g ckp` zobrazit doporučení ke konfiguraci.

Přetečení fyzického protokolu po vypnutí protokolování transakcí

Fyzický protokol může přetéci, Používáte-li jednoduché velké objekty nebo inteligentní velké objekty v databázi, u které je vypnuté protokolování transakcí, jak je uvedeno v následujícím příkladu:

Když databázový server zpracovává jednoduché velké objekty, každá část těchto objektů, které databázový server ukládá na disk, může být zaprotokolována zvlášť, aby proces mohl mezi oběma částmi ukončit kritické sekce kódu. Pokud je ovšem protokolování vypnuto, musí databázový server provádět všechny operace na jednoduchých velkých objektech v jedné kritické části. Je-li jednoduchý velký objekt velký a velikost fyzického protokolu malá, může takto dojít k jeho zaplnění. Jestliže tato situace nastane, odešle databázový server do protokolu zpráv následující zprávu:

```
Physical log file overflow
```

Databázový server se potom vypne. Doporučenou opravnou akcí naleznete v protokolu zpráv.

Kontrolní body

Databázový server periodicky přesouvá transakce a data ze společné oblasti vyrovnávací paměti na disk. Dokud nejsou transakce a data přesunuty na disk, jsou ve stavu přesunu. Místo vynucování přesunu každé transakce na disk ihned po jejím dokončení zapíše databázový server transakce do logického protokolu. Databázový server protokoluje transakce v pořadí, ve kterém se udály. V případě selhání systému server provede následující akce:

- Přehraje protokol pro zopakování a obnovení transakcí
- Navrátí databázi do stavu konzistentního se stavem databázového systému v době selhání.

K ulehčení obnovení nebo logické obnovy databázového systému vytváří databázový server záchytný konzistentní bod, nazývaný *kontrolní bod*. Kontrolní bod je bod v čase protokolu, ve kterém byl potvrzen známý konzistentní stav databázového systému. Obvykle tvorba kontrolního bodu zahrnuje uložení určitého množství informací tak, aby databázový server mohl v případě selhání od tohoto bodu pokračovat.

Účelem kontrolních bodů je soustavně posouvat bod, od kterého jde v případě potřeby pokračovat, v logickém protokolu vpřed. Pokud kontrolní body neexistují a selhání nastane, bude databázový server muset zpracovat všechny transakce uložené v logickém protokolu od posledního restartování systému.

Kontrolní bod se může tvořit v následujících situacích:

- Při specifických událostech. Kontrolní bod je tvořen například kdykoli je do serveru přidán prostor typu dbspace nebo kdykoli je zálohována databáze.

Tyto typy událostí obvykle spustí tvorbu kontrolního bodu, která blokuje zpracovávání transakcí. Proto jsou tyto kontrolní body nazývané jako *blokující* kontrolní body.

- Nastanou-li omezení prostředků. Kontrolní bod je například vyžadován pro každý úsek prostoru logického protokolu, aby byla záruka, že tento protokol má kontrolní bod, od kterého lze začít proces rychlé obnovy. Databázový server vytváří kontrolní bod jakmile je fyzický protokol ze 75% zaplněn.

Kontrolní body vyvolané omezeními prostředků obvykle neblokují transakce. Proto jsou tyto kontrolní body nazývané jako *neblokující* kontrolní body.

Nicméně přiblíží-li se databázový server během zpracovávání kontrolního bodu k vyčerpání prostředků, zablokují se ihned transakce aby bylo zajištěno dokončení kontrolního bodu před úplným vyčerpáním prostředků. Jsou-li transakce blokovány, pokusí se server tvořit kontrolní body častěji, aby k tomu při jejich tvorbě již nedocházelo. Další informace naleznete v části "Postup k určení velikosti fyzického protokolu" na stránce 15-3.

Automatické kontrolní body způsobují častější tvorbu kontrolních bodů za účelem zabránění blokování transakcí. Automatické kontrolní body se pokoušejí sledovat aktivitu systému a využití prostředků (využití fyzického protokolu a logického protokolu spolu s aktualizací společných oblastí vyrovnávací paměti) a tvořit tak kontrolní body dostatečně včas na to, aby mohly být dokončeny předtím, než je fyzický protokol nebo logický protokol zaplněn.

Databázový server tvoří alespoň jeden automatický kontrolní bod pro každý úsek prostoru logického protokolu. To zaručuje existenci kontrolního bodu, který může být výchozím bodem procesu rychlé obnovy.

Použitím konfiguračního parametru `AUTO_CKPTS` můžete povolit nebo zakázat automatické kontrolní body při startu databázového serveru. (dynamicky lze automatické kontrolní body povolit nebo zakázat pomocí příkazu **onmode -wm** nebo **onmode -wf**.)

Ruční kontrolní body jsou tvořeny za událostí, které můžete sami spustit.

Databázový server umožňuje dvě metody určování délky procesu rychlé obnovy v případě neplánovaného výpadku.

- Konfiguračním parametrem CKPTINTVL určíte frekvenci, se kterou bude server tvořit kontrolní body.
- Konfiguračním parametrem RTO_SERVER_RESTART určíte délku procesu rychlé obnovy.

Používáte-li konfigurační parametr RTO_SERVER_RESTART, platí že:

- Databázový server ignoruje konfigurační parametr CKPTINTVL.
- Databázový server sledováním využití fyzického logu a logického protokolu odhaduje délku trvání procesu rychlé obnovy. Pokud server odhadne, že rychlá obnova zabere více času než kolik je určeno konfiguračním parametrem RTO_SERVER_RESTART, vytvoří server automaticky kontrolní bod.

Hodnota konfiguračního parametru RTO_SERVER_RESTART je míněna jako cílová doba, nikoli zaručená doba.

Některé faktory, které mohou prodloužit dobu restartování, mohou také ovlivnit dobu trvání rychlé obnovy. Tyto faktory zahrnují odvolávání dlouhých transakcí aktivních v době neplánovaného výpadku.

Více informací o konfiguračních parametrech RTO_SERVER_RESTART a AUTO_CKPTS naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.

Hodnoty LRU pro vyprázdňování společné oblasti vyrovnávací paměti mezi kontrolními body

Hodnoty LRU pro vyprázdňování společné oblasti vyrovnávací paměti mezi kontrolními body nejsou z hlediska výkonu tvorby kontrolních bodů důležité. Hodnoty **lru_max_dirty** a **lru_min_dirty**, které se nastavují v rámci konfiguračního parametru BUFFERPOOL, jsou obvykle potřebné pouze pro údržbu dostatečně čistých stránek pro náhradu. Zpočátku nastavte hodnotu **lru_min_dirty** na 70 a hodnotu **lru_max_dirty** na 80.

Pokud jsou během tvorby kontrolního bodu blokovány transakce, pokusí se poté databázový server o zvýšení frekvence kontrolních bodů, aby blokování transakcí omezil. Hledá-li server volnou stránku pro nahrazení a dojde k zápisu na popředí, zvýší server automaticky frekvenci vyprázdňování LRU, aby k tomu příště již nedošlo. Dokončí-li databázový server náhradu stránky a nalezne často používanou stránku, zvýší automaticky vyprázdňování LRU. Libovolné automatické změny hodnot vyprázdňování LRU se v souboru ONCONFIG neprojeví.

Více informací o sledování a ladění parametrů kontrolních bodů a informace o úpravách a ladění hodnot LRU naleznete v příručce *Příručka výkonnosti serveru IBM Informix Dynamic Server*.

Kontrolní body během zálohování

Zálohujete-li databázi, vytvoří databázový server kontrolní bod a vyprázdní všechny změněné stránky na disk. Během obnovy databázový server znovu použije všechny záznamy logického protokolu.

Další informace o programu ON-Bar nebo obslužném programu **ontape** naleznete v příručce *IBM Informix Backup and Restore Guide*

Rychlá obnova

Rychlá obnova je automatická vůči chybám tolerantní funkce, kterou databázový server provádí, změni kdykoli stav z režimu offline do režimu klidu, administrace nebo online. V souvislosti s rychlou obnovou není nutné provádět žádné administrativní akce, tato funkce je automatická.

Proces rychlé obnovy kontroluje, zda při posledním přechodu databázového serveru do režimu offline proběhl tento přechod neřízeně. Pokud ano, vrátí rychlá obnova databázový server do stavu fyzické a logické konzistence.

Jestliže proces rychlé obnovy zjistí, že se databázový server dostal do režimu offline řízeným způsobem, proces rychlé obnovy se ukončí a databázový server se přepne do režimu online.

Další informace naleznete v kapitole “Rychlá obnova po kontrolním bodu” na stránce 15-8.

Potřeba rychlé obnovy

Rychlá obnova obnovuje databázový server do stavu fyzické a logické konzistence po jakémkoliv selhání, které vede ke ztrátě obsahu paměti databázového serveru. Operační systém může například selhat bez upozornění. Selhání systému nepoškodí databázi, ale místo toho ovlivní transakce, které jsou v okamžiku selhání zpracovávány.

Situace, při kterých se spouští rychlá obnova

Kdykoli správce přepne stav databázového serveru z režimu offline do režimu klidu, administrace nebo online, databázový server zkontroluje, zda je potřeba provést rychlou obnovu.

Jako součást inicializace sdílené paměti kontroluje databázový server obsah fyzického protokolu. Fyzický protokol je prázdný, pokud vypnutí databázového serveru proběhlo řízeně. Přesun z režimu online do klidového režimu zahrnuje kontrolní bod, který vyprázdní fyzický protokol. Pokud tedy databázový server nalezne stránky ve fyzickém protokolu, je zřejmé, že databázový server se dostal do stavu offline neřízeným způsobem a je zahájena rychlá obnova.

Rychlá obnova a protokolování s vyrovnávací pamětí

Pokud databáze používá protokolování s vyrovnávací pamětí (jak bylo popsáno v části “Protokolování transakcí s vyrovnávací pamětí” na stránce 11-6), nemusí být v okamžiku selhání některé záznamy logického protokolu přidružené k potvrzeným transakcím zapsané do logického protokolu. Pokud nastane taková situace, nemůže rychlá obnova tyto transakce obnovit. Rychlá obnova může obnovit pouze transakce přidružené k záznamu COMMIT uloženému v logickém protokolu na disku. (Z tohoto důvodu protokolování s vyrovnávací pamětí představuje kompromis mezi výkonem a zranitelností dat.)

Možné přetečení fyzického protokolu během rychlé obnovy

Během rychlé obnovy může fyzický protokol přetéci. Pokud k tomu dojde, pokusí se databázový server zvětšit prostor fyzického protokolu o diskový soubor nazvaný **plog_extend.číslo_serveru**. Výchozí umístění tohoto souboru je adresář **\$INFORMIXDIR/tmp**.

Pomocí parametru **ONCONFIG PLOG_OVERFLOW_PATH** definujte umístění pro vytvoření tohoto souboru.

Databázový server odstraní soubor **plog_extend.servernum**, když je během rychlé obnovy proveden první kontrolní bod.

Rychlá obnova a žádné protokolování

U databází nebo tabulek, které nepoužívají protokolování, obnovuje rychlá obnova jejich stav na okamžik nejnovějšího kontrolního bodu. Všechny změny provedené do databáze od posledního kontrolního bodu jsou ztraceny.

Rychlá obnova po kontrolním bodu

Rychlá obnova vrací databázový server do stavu konzistence jako část inicializace sdílené paměti. Všechny proběhlé transakce jsou obnoveny a všechny neproběhlé transakce jsou odvolány.

Rychlá obnova probíhá těmito kroky:

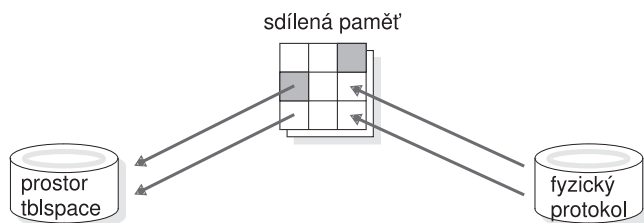
1. Databázový server používá data ve fyzickém protokolu k navrácení všech stránek na disku do jejich stavu v době posledního kontrolního bodu. Tento bod je nazýván *fyzickou konzistencí*.
2. Databázový server vyhledá poslední záznam o kontrolního bodu v souborech logického protokolu.
3. Databázový server přehraje všechny záznamy logického protokolu zapsané po posledním záznamu kontrolního bodu.
4. Databázový server odvolá všechny transakce, které nebyly potvrzeny. Některé transakce XA mohou být nevyřešené, dokud nebude dostupný správce prostředků XA.

Následující části detailně popisují každý krok:

- “Server se vrací do stavu v době posledního kontrolního bodu.”
- “Server hledá záznam kontrolního bodu v logickém protokolu”
- “Server přehraje záznamy logického protokolu” na stránce 15-9
- “Server odvolává nepotvrzené transakce” na stránce 15-9

Server se vrací do stavu v době posledního kontrolního bodu.

Aby mohly být všechny stránky disku vrácené do stavu v době posledního kontrolního bodu, zapíše databázový server předobrazy uložené ve fyzickém protokolu do sdílené paměti a poté zpět na disk. Každý předobraz ve fyzickém protokolu obsahuje adresu stránky, která byla aktualizovaná po kontrolním bodu. Když databázový server zapíše každou stránku předobrazu do fyzického protokolu do sdílené paměti a potom zpět na disk, budou změny v datech databázového serveru od okamžiku posledního kontrolního bodu vráceny zpět. Databázový server je nyní fyzicky konzistentní. Tento krok ilustruje Obrázek 15-1.



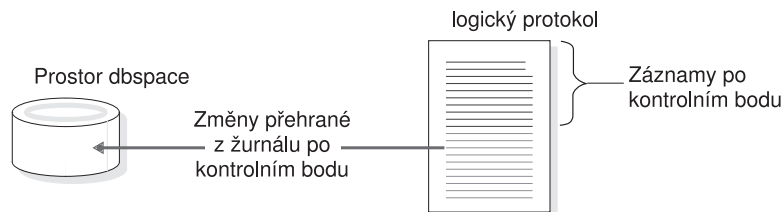
Obrázek 15-1. Zápis všech zbývajících předobrazů ve fyzickém protokolu na disk.

Server hledá záznam kontrolního bodu v logickém protokolu

Po navrácení ke stavu v době posledního kontrolního bodu databázový server vyhledá adresu posledního záznamu kontrolního bodu v logickém protokolu. Záznam posledního kontrolního bodu se zaručeně nachází v logickém protokolu na disku.

Server přehraje záznamy logického protokolu

Po vyhledání záznamu kontrolního bodu v logickém protokolu, databázový server přehraje záznamy v logickém protokolu, které byly napsány po posledním záznamu kontrolního bodu. Tato akce zreprodukuje všechny změny v databázi, které nastaly po posledním kontrolním bodu, až po moment nekontrolovaného vypnutí. Tento krok ilustruje Obrázek 15-2.



Obrázek 15-2. Přehrání záznamů logického protokolu zapsaných od posledního kontrolního bodu.

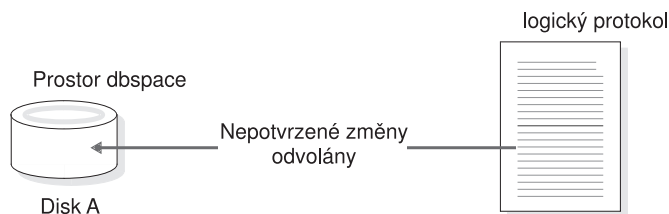
Server odvolává nepotvrzené transakce

Po přehrání logického protokolu z něj databázový server odvolá všechny záznamy transakcí, které nebyly potvrzeny v době selhání systému. Všechny databáze jsou logicky konzistentní, neboť všechny potvrzené transakce byly přehrány a všechny nepotvrzené transakce byly odvolány. Některé transakce XA mohou být nevyřešené, dokud nebude dostupný správce prostředků XA.

Transakce, které by dokončily první fázi u dvoufázového potvrzení, jsou výjimečné případy. Další informace naleznete v části “Jak protokol dvoufázového potvrzování ošetřuje selhání” na stránce 24-7.

Protože jeden jedna nebo více transakcí mohly přetrvat přes několik kontrolních bodů aniž by byly potvrzeny, může toto odvolání zasahovat v logickém protokolu zpět až za poslední kontrolní bod. Všechny soubory logického protokolu, které obsahují záznamy pro otevřené transakce jsou dostupné na databázovém serveru, protože soubor protokolu není uvolněn, dokud nejsou uzavřené všechny transakce, které obsahuje.

Obrázek 15-3 znázorňuje proceduru odvolání transakce. Zde jsou nepotvrzené změny odvolány z logického protokolu do prostoru dbspace na některém pevném disku. Když je proces rychlé obnovy dokončen, přepne se databázový server zpět do režimu klidu, administrace nebo online.



Obrázek 15-3. Odvolání všech nedokončených transakcí.

Kapitola 16. Správa fyzického protokolu

Obsah kapitoly	16-1
Změna umístění a velikosti fyzického protokolu	16-1
Kontrola odpovídajícího souvislého prostoru	16-2
Použití obslužného programu <code>onparams</code> ke změně umístění nebo velikosti fyzického protokolu	16-2
Použití programu <code>ON-Monitor</code> ke změně umístění a velikosti fyzického protokolu (systém UNIX)	16-2
Monitorování aktivity fyzického a logického protokolování.	16-2
Monitorování informací o kontrolních bodech.	16-3
Zapnutí a vypnutí kontrolních bodů	16-4
Vynucení kontrolního bodu	16-4
Použití statistických údajů o kontrolních bodech poskytovaných serverem	16-5
Použití tabulek <code>SMI</code>	16-5
Zapnutí a vypnutí automatického ladění <code>LRU</code>	16-5

Obsah kapitoly

Tato kapitola popisuje následující postupy:

- Změnu umístění a velikosti fyzického protokolu.
- Monitorování fyzického protokolu, vyrovnávací paměti fyzického protokolu a vyrovnávací paměti logického protokolu.
- Monitorování a vynucení kontrolních bodů.

Související informace uvádí Kapitola 15, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 15-1.

Změna umístění a velikosti fyzického protokolu

Umístění a velikost fyzického protokolu můžete změnit libovolným z následujících způsobů:

- Pomocí obslužného programu `onparams` z příkazového řádku.

Jen pro UNIX

- Pomocí programu `ON-Monitor`.

Konec Jen pro UNIX

Až budete provádět změny, přihlaste se do systému UNIX jako uživatel `informix` nebo `root` nebo jako člen skupiny `Informix-Admin` v systému Windows.

Soubor fyzického protokolu můžete přesunout a pokusit se tak zvýšit výkon. Když databázový server inicializuje diskový prostor, umístí stránky na disku přidělené logickému a fyzickému protokolu do kořenového prostoru `dbspace`. Výkon můžete zvýšit přesunutím fyzického protokolu, souborů logického protokolu nebo obou do jiných prostorů `dbspace`.

Poznámka: Nelze připojit logické nebo fyzické protokoly k prostorům `dbspace`, které mají jiné než výchozí velikosti stránky.

Informace o tom, kam je vhodné umístit fyzický protokol, naleznete v části “Určení umístění fyzického protokolu” na stránce 15-3. Informace o určování velikosti fyzického protokolu naleznete v části “Velikost a umístění fyzického protokolu” na stránce 15-3. Další informace o fyzickém protokolu naleznete v části “Monitorování aktivity fyzického a logického protokolování” na stránce 16-2.

Kontrola odpovídajícího souvislého prostoru

Prostor přidělený pro fyzický protokol musí souvislý. Pokud jsou při inicializaci databázového serveru pomocí příkazu **oninit -i** nedostatečné zdroje, inicializace se nepodaří.

Chcete-li změnit velikost nebo umístění fyzického protokolu a cílový prostor dbspace neobsahuje dostatečné souvislé místo, server fyzický protokol nezmění.

Pomocí volby **oncheck -pe** můžete zjistit, zda je dostupný odpovídající souvislý prostor. Informace o použití příkazu **oncheck** s volbami **-ce** a **-pe** při kontrole seznamu volných stránek v bloku naleznete v příručce *IBM Informix Administrator's Reference*.

Použití obslužného programu onparams ke změně umístění nebo velikosti fyzického protokolu

Chcete-li změnit velikost a umístění fyzického protokolu, převedte databázový server do klidového režimu nebo do režimu administrace a proveďte následující příkaz:

```
onparams -p -s velikost -d dbspace -y
```

velikost Nová velikost fyzického protokolu v kilobajtech.

dbspace Určuje prostor dbspace, ve kterém má být uložen fyzický protokol.

V následujících příkladech se mění velikost a umístění fyzického protokolu. Velikost nového fyzického protokolu je 400 kilobajtů a protokol bude uložen do prostoru dbspace **dbspace6**:

```
onparams -p -s 400 -d dbspace6 -y
```

Informace o změně fyzického protokolu pomocí obslužného programu **onparams** naleznete v příručce *IBM Informix Administrator's Reference*.

Použití programu ON-Monitor ke změně umístění a velikosti fyzického protokolu (systém UNIX)

Chcete-li změnit velikost nebo umístění prostoru dbspace nebo obojí, pomocí programu ON-Monitor, vyberte příkaz **Parametry > Fyzický protokol**.

Monitorování aktivity fyzického a logického protokolování

Monitorováním fyzického protokolu můžete určit v procentech využití souboru fyzického protokolu v okamžiku, kdy nastane kontrolní bod. Tato informace vám umožní zjistit optimální velikost souboru fyzického protokolu. Měl by být dostatečně velký, aby databázový server nemusel vynucovat kontrolní body příliš často, a dostatečně malý, aby šetřil prostorem na disku a zaručoval rychlou obnovu.

Monitorováním vyrovnávacích pamětí fyzického protokolu a logického protokolu můžete určit, zda mají optimální velikost pro aktuální úroveň zpracování. Důležitou statistikou, kterou je vhodné monitorovat, je statistika stránek na zápis na disk. Další informace o ladění vyrovnávacích pamětí fyzického protokolu a logického protokolu naleznete v příručce *Řízení výkonu systému IBM Informix*.

K monitorování souboru fyzického protokolu, vyrovnávacích pamětí fyzického protokolu a vyrovnávacích pamětí logického protokolu použijte následující příkazy.

Obslužný program	Příkaz	Dodatečné informace
Příkazový řádek nebo program ISA	onstat -l	<p>První řádek zobrazuje následující informace pro každou vyrovnávací paměť fyzického protokolu:</p> <ul style="list-style-type: none"> • Počet použitých stránek vyrovnávací paměti (bufused). • Velikost každé vyrovnávací paměti fyzického protokolu ve stránkách (bufsize). • Počet stránek zapsaných do vyrovnávací paměti (numpages). • Počet zápisů z vyrovnávací paměti na disk (numwrits). • Poměr stránek zapsaných do vyrovnávací paměti k počtu zápisů na disk (pages/IO). <p>Druhý řádek zobrazují následující informace o fyzickém protokolu:</p> <ul style="list-style-type: none"> • Číslo první stránky v souboru fyzického protokolu (phybegin). • Velikost souboru fyzického protokolu ve stránkách (physize). • Aktuální pozice, na které dojde k následujícímu zápisu, určená jako číslo stránky (physpos). • Počet stránek v protokolu, které byly použity (phyused). • Procento z celkového počtu všech stránek fyzického protokolu, které byly použity (%used). <p>Třetí řádek zobrazují následující informace o vyrovnávací paměti logického protokolu:</p> <ul style="list-style-type: none"> • Počet použitých stránek vyrovnávací paměti (bufused). • Velikost každé vyrovnávací paměti logického protokolu ve stránkách (bufsize). • Počet záznamů zapsaných do vyrovnávací paměti (numrecs). • Počet stránek zapsaných do vyrovnávací paměti (numpages). • Počet zápisů z vyrovnávací paměti na disk (numwrits). • Poměr záznamů ke stránkám ve vyrovnávací paměti (recs/pages). • Poměr stránek zapsaných do vyrovnávací paměti k počtu zápisů na disk (pages/IO).
Příkazový řádek nebo program ISA	onparams -p	Přesune fyzický protokol nebo změní jeho velikost.
Příkazový řádek nebo program ISA	onmode -l	Přechod na následující soubor logického protokolu.
ISA	Protokoly> Logické	Klepněte na volbu Přejít na následující soubor protokolu .

Poznámka: Další informace o výstupu příkazu **onstat -l** output spolu s příkladem naleznete v příručce *IBM Informix Administrator's Reference*.

Informace o použití administrativních příkazů rozhraní SQL API namísto některých příkazů **onparams** and **onmode** naleznete v příručce “Vzdálená správa pomocí příkazů SQL” na stránce 26-10 a v příručce *IBM Informix Guide to SQL: Syntax*.

Monitorování informací o kontrolních bodech

Monitorováním aktivity kontrolních bodů získáte základní informace o kontrolních bodech. Tyto informace zahrnují počet čekání jednotkových procesů na dokončení kontrolního bodu. Tyto informace jsou užitečné pro určení, zda je interval kontrolních bodů vhodně nastaven. Další informace o ladění intervalu kontrolních bodů naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

K monitorování kontrolních bodů můžete použít následující příkazy.

Obslužný program	Příkaz	Dodatečné informace
Příkazový řádek nebo program ISA	onstat -m	Zobrazí posledních 20 řádků v protokolu zpráv. Pokud se v posledních 20 řádcích nezobrazuje zpráva o kontrolním bodu, přečtěte si protokol zpráv přímo pomocí textového editoru. Databázový server zapisuje jednotlivé zprávy o kontrolních bodech do protokolu po dokončení kontrolního bodu. Jestliže se vyskytne kontrolní bod, ale databázový server nemá žádné stránky pro zápis na disk, nezapiše databázový server do protokolu zpráv žádnou zprávu.
Příkazový řádek nebo program ISA	onstat -p	Získá následující statistiky kontrolních bodů: <ul style="list-style-type: none"> • numckpts: Počet kontrolních bodů, které se vyskytly od uvedení databázového serveru do režimu online. • ckptwaits: Počet opakování, kdy uživatelský jednotkový proces čekal na dokončení kontrolního bodu. Databázový server zabraňuje během kontrolního bodu uživatelským jednotkovým procesům ve vstupu do kritické části.
ON-Monitor (systém UNIX)	Stav > Profil	Pole Checkpoints a Check Waits zobrazují stejné informace jako pole numckpts a ckpwaits v příkazu onstat -p .

Zapnutí a vypnutí kontrolních bodů

Chcete-li zapnout nebo vypnout automatické ladění kontrolních bodů, zadejte příkaz **onmode -wf RTO_SERVER_RESTART value**.

Vynucení kontrolního bodu

Chcete-li vynutit kontrolní bod, proveďte jeden z následujících příkazů.

Obslužný program	Příkaz	Dodatečné informace
Příkazový řádek nebo program ISA	onmode -c	Žádné.
ISA	Protokoly > Logické	Klepněte na volbu Vynutit kontrolní bod .
ON-Monitor (systém UNIX)	Volba Force-Ckpt v hlavní nabídce	Čas uvedený v poli Last Checkpoint Done se nezmění, dokud nenastane kontrolní bod. Pole Last Checkpoint Check zobrazuje čas poslední kontroly kontrolního bodu. Pokud od doby posledního kontrolního bodu nedošlo k žádným změnám, neprovedl databázový server žádný kontrolní bod.

Kontrolní bod byste měli vynutit v libovolné z následujících situací:

- Chcete-li uvolnit soubor logického protokolu, který obsahuje záznam nejnovějšího kontrolního bodu a který je zálohovaný, ale není uvolněný (stav **U-B-L** nebo **U-B** příkazu **onstat -l**).
- Předtím, než vydáte příkaz **onmode -sy** k uvedení databázového serveru do klidového režimu.
- Po vytvoření velkého indexu, pokud se databázový server ukončí před následujícím kontrolním bodem (vytváření indexu bude znovu spuštěno při dalším restartování databázového serveru).
- Pokud se kontrolní bod nevyskytl po dlouhou dobu a hodláte spustit systémovou operaci, která by mohla narušit provoz databázového serveru.

- Pokud zápisy na popředí zabírají více zdrojů, než chcete (vynucením kontrolního bodu snížíte dočasně jejich počet na nulu).
- Před spuštěním příkazu **dbexport** nebo před uvolněním tabulky, abyste zajistili fyzickou konzistenci všech dat předtím, než ji vyexportujete nebo uvolníte.
- Po provedení velkého zavádění tabulek pomocí příkazů PUT nebo INSERT.

Informace o použití administrativních příkazů rozhraní SQL API namísto některých příkazů **onmode** naleznete v příručce “Vzdálená správa pomocí příkazů SQL” na stránce 26-10 a v příručce *IBM Informix Guide to SQL: Syntax*.

Použití statistických údajů o kontrolních bodech poskytovaných serverem

Databázový server poskytuje informace o historii předchozích dvaceti kontrolních bodů. Tyto informace jsou přístupné prostřednictvím tabulek **sysckpthist** a **sysckptinfo** rozhraní SMI.

Použití tabulek SMI

Dotazováním na tabulky **sysprofile** získáte statistiky týkající se vyrovnávacích pamětí fyzického a logického protokolu. Tabulka **sysprofile** také poskytuje stejné statistiky kontrolních bodů, které jsou dostupné prostřednictvím volby **onstat -p**. Tyto řádky obsahují následující statistiky.

Řádek	Popis
plpgawrites	Počet stránek zapsaných do vyrovnávací paměti fyzického protokolu.
plgwrites	Počet zápisů z vyrovnávací paměti fyzického protokolu do souboru fyzického protokolu.
llgrecs	Počet záznamů zapsaných do vyrovnávací paměti logického protokolu.
llpgawrites	Počet stránek zapsaných do vyrovnávací paměti logického protokolu.
llgwrites	Počet zápisů z vyrovnávací paměti logického protokolu do souboru logického protokolu.
numckpts	Počet kontrolních bodů, které se vyskytly od uvedení databázového serveru do režimu online.
ckptwaits	Počet situací, kdy jednotkové procesy čekaly na dokončení kontrolního bodu, aby mohly vstoupit do <i>kritické části</i> .
value	Hodnoty pro numckpts a ckptwaits .

Zapnutí a vypnutí automatického ladění LRU

Je-li nastaven konfigurační parametr **RTO_SERVER_RESTART**, databázový server automaticky nastaví počet virtuálních procesorů AIO a jednotkových procesů vyčištění a automaticky vyladí LRU (naposledy použitá) vyprázdnění. Databázový server vytiskne varovné zprávy do protokolu zpráv, nemůže-li server dodržet metody **RTO_SERVER_RESTART**.

Pomocí konfiguračního parametru **AUTO_LRU_TUNING** můžete povolit nebo zakázat automatické ladění LRU při spuštění databázového serveru.

Chcete-li automatické ladění LRU pro určitou relaci zakázat, zadejte příkaz **onmode -wm AUTO_LRU_TUNING 0**.

Chcete-li automatické ladění povolit po zákazu během relace, zadejte příkaz **onmode -wm AUTO_LRU_TUNING 1**.

Změny automatického ladění LRU ovlivňují všechny společné oblasti vyrovnávací paměti a nastavují hodnoty **lru_min_dirty** a **lru_max_dirty** v konfiguračním parametru BUFFERPOOL.

Další informace o ladění LRU naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Část 4. Odolnost vůči selhání

Kapitola 17. Zrcadlení

Obsah kapitoly	17-1
Zrcadlení	17-1
Výhody zrcadlení	17-2
Požadavky na zrcadlení	17-2
Důsledky nepoužívání zrcadlení	17-2
Data vhodná k zrcadlení	17-2
Alternativy k zrcadlení	17-3
Správci logických disků	17-3
Hardwarové zrcadlení	17-3
Externí zálohování a obnovení	17-3
Průběh zrcadlení.	17-3
Vytvoření zrcadleného bloku	17-4
Stavové příznaky zrcadlení	17-4
Obnova	17-4
Činnosti v průběhu zpracování	17-5
Diskové zápisy do zrcadlených bloků	17-5
Diskové čtení ze zrcadlených bloků	17-5
Zjišťování poruch média	17-5
Obnova bloku	17-5
Výsledek zastavení zrcadlení	17-6
Struktura zrcadleného bloku	17-6

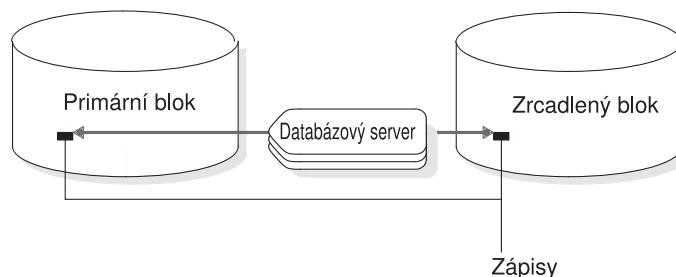
Obsah kapitoly

Tato kapitola popisuje funkci zrcadlení databázového serveru. Další pokyny pro provádění úloh zrcadlení uvádí Kapitola 18, “Použití zrcadlení”, na stránce 18-1.

Zrcadlení

Zrcadlení je strategie, která páruje *primární blok* jednoho definovaného prostoru dbspace, blobspace nebo sbspase se stejně velkým *zrcadleným blokem*.

Každý zápis do primárního bloku je automaticky doprovázen identickým zápisem do zrcadleného bloku. Tento pojem ilustruje Obrázek 17-1. Pokud dojde k chybě primárního bloku, umožňuje zrcadlení číst a zapisovat do zrcadleného bloku, dokud se primární blok neobnoví, a to bez přerušení přístupu uživatele k datům.



Obrázek 17-1. Zápis dat do primárního bloku i do zrcadleného bloku

Zrcadlení není podporováno na discích, které jsou spravovány prostřednictvím sítě. Všechny bloky zrcadlené sady musí spravovat stejná instance databázového serveru.

Výhody zrcadlení

Pokud dojde k selhání média, poskytuje zrcadlení administrátorovi databázového serveru prostředek pro obnovu dat, aniž by musel být databázový server převeden do stavu offline. Zrcadlení tedy poskytuje vyšší spolehlivost a nižší prostoje systému. Aplikace mohou navíc pokračovat ve čtení a zápisu do databáze, jejíž primární bloky jsou na poškozeném médiu, za předpokladu, že bloky zrcadlicí tato data jsou umístěny na samostatném médiu.

Každá důležitá databáze by měla být umístěna v zrcadleném prostoru dbSPACE. Především by měl být zrcadlen kořenový prostor dbSPACE, který obsahuje rezervované stránky databázového serveru.

Požadavky na zrcadlení

Se zrcadlením jsou spojeny náklady na diskový prostor a náklady na výkon. Náklady na diskový prostor jsou důsledkem potřeby dodatečného prostoru pro uložení zrcadlených dat. Náklady na výkon vyplývají z nutnosti provádět zápis do primárních i do zrcadlených bloků. Použití několika virtuálních procesorů pro zápisy na disk tyto náklady na výkon snižuje. Použití *rozdělených čtení*, při kterých databázový server čte data buď z primárního, nebo ze zrcadleného bloku podle umístění dat v bloku, ve skutečnosti výkon zvyšuje, pokud jsou data pouze čtena. Další informace o tom, jak databázový server provádí čtení a zápis zrcadlených bloků, najdete v části "Činnosti v průběhu zpracování" na stránce 17-5.

Důsledky nepoužívání zrcadlení

Není-li prostor dbSPACE zrcadlen, musíte častěji provádět obnovení ze zálohy paměťových prostorů, protože se zvyšuje počet selhání médií.

Po selhání média zrcadleného bloku čte databázový server výhradně z bloku, který je dosud online, dokud nepřevéte poškozený blok zpět do režimu online. Přestane-li *zrcadlený* blok fungovat, nemá databázový server přístup k datům uloženým v tomto bloku. Pokud blok obsahuje soubory logického protokolu, fyzický protokol nebo kořenový prostor dbSPACE, přejde databázový server ihned do režimu offline. Neobsahuje-li blok soubory logického protokolu, fyzický protokol ani kořenový prostor dbSPACE, může být databázový server v provozu, ale jednotkové procesy nemohou číst z poškozeného bloku ani do něj zapisovat. Je-li poškozen nezrcadlený blok, je nutné ho obnovit obnovou dbSPACE ze zálohy.

Data vhodná k zrcadlení

V ideálním případě by měla být zrcadlena všechna data. Máte-li potíže s nedostatkem místa na disku, nelze tohoto ideálního stavu dosáhnout. V tomto případě vyberte určité kritické bloky, které mají být zrcadleny.

K těmto kritickým blokům vždy patří bloky, které jsou částí kořenového prostoru dbSPACE, blok, ve kterém jsou uloženy soubory logického protokolu, a blok, ve kterém jsou uloženy fyzické protokoly. Pokud některý z těchto kritických bloků chybí, přejde databázový server ihned do režimu offline.

Pokud některé bloky obsahují data, která jsou důležitá pro vaše podnikání, dejte těmto blokům vysokou prioritu pro zrcadlení.

Dejte vyšší prioritu pro zrcadlení také blokům, ve kterých jsou uložena často používaná data. Zajistíte tak, že nebudou ukončeny aktivity mnoha uživatelů při vypnutí jednoho široce používaného bloku.

Alternativy k zrcadlení

Zrcadlení, o němž pojednává tato příručka, je funkce databázového serveru. Operační systém nebo hardware mohou poskytovat alternativní řešení zrcadlení.

Pokud zvažujete použití funkce zrcadlení, kterou poskytuje operační systém, namísto funkce zrcadlení databázového serveru, porovnejte implementaci obou funkcí, než se rozhodnete, kterou z nich použijete. Nejpomalejším krokem v procesu zrcadlení je vlastní zápis dat na disk. Strategie databázového serveru spočívající v paralelním provádění zápisů do zrcadlených bloků pomáhá zkracovat dobu nutnou pro tento krok. (Další informace naleznete v části “Diskové zápisy do zrcadlených bloků” na stránce 17-5.) Zrcadlení databázového serveru navíc zvyšuje výkon čtení pomocí rozděleného čtení. (Další informace naleznete v části “Diskové čtení ze zrcadlených bloků” na stránce 17-5.) Funkce zrcadlení operačního systému, které při zrcadlení nevyužívají paralelní zápisy a rozdělené čtení, mohou mít nižší výkon.

Můžete také používat zrcadlení databázového serveru a zrcadlení operačního systému současně. Tyto funkce jsou na sobě nezávislé. V některých případech se můžete rozhodnout, že budete současně používat zrcadlení databázového serveru i funkci zrcadlení, kterou poskytuje operační systém. Můžete mít například data databázového serveru na jedné diskové jednotce s dalšími daty. Tato další data můžete zrcadlit pomocí zrcadlení operačního systému a data databázového serveru můžete zrcadlit pomocí zrcadlení databázového serveru.

Správci logických disků

Správci logických disků jsou alternativním řešením zrcadlení. Někteří výrobci operačních systémů poskytují obslužný program tohoto typu, který umožňuje, aby se mohlo několik disků jevit jako jeden systém souborů. Uložení dat na více než dvou discích poskytuje další ochranu před selháním médií, ale další zápisy zvyšují požadavky na výkon.

Hardwarové zrcadlení

Dalším řešením je použití hardwarového zrcadlení, jako je například pole RAID (redundant array of inexpensive disks). Výhodou tohoto typu hardwarového zrcadlení je, že uložení stejného množství dat, které chrání před selháním médií, vyžaduje méně diskového prostoru než zrcadlení databázového serveru.

Některé hardwarové systémy zrcadlení podporují *výměnu disků za chodu*. Poškozený disk můžete vyměnit, zatímco databázový server zůstane online. Před provedením výměny za chodu se doporučuje snížit aktivity vstupu - výstupu.

Důležité: Máte-li při použití hardwarového zrcadlení potíže s databázovým serverem, prostudujte dokumentaci operačního systému nebo disku, nebo požádejte o pomoc technickou podporu.

Externí zálohování a obnovení

Je-li použito hardwarové zrcadlení disku, je možné uvést systém do činnosti rychleji pomocí externího zálohování a obnovení, než pomocí obvyklých ON-Bar příkazů. Další informace o externím zálohování a obnovení naleznete v části *IBM Informix Backup and Restore Guide*.

Průběh zrcadlení

Tato část popisuje podrobněji průběh zrcadlení. Návod k provádění operací zrcadlení, jako je vytvoření zrcadlených bloků, zahájení zrcadlení, změna stavu zrcadlených bloků atd. uvádí Kapitola 18, “Použití zrcadlení”, na stránce 18-1.

Vytvoření zrcadleného bloku

Zadáte-li zrcadlený blok, databázový server kopíruje všechna data z primárního bloku do zrcadleného bloku. Tento proces kopírování se nazývá *obnova*. Zrcadlení je spuštěno, jakmile je obnova dokončena.

Procedura obnovy, která určuje začátek zrcadlení, se opozdí, pokud se začnou zrcadlit bloky uvnitř prostoru dbSPACE, který obsahuje soubor logického protokolu. Zrcadlení prostorů dbSPACE obsahujících soubory logického protokolu nebude zahájeno, dokud nevytvoříte zálohu úrovně 0 kořenového prostoru dbSPACE. Zpoždění zajistí, že databázový server bude moci použít soubory logického protokolu, i když primární blok obsahující tyto soubory logického protokolu bude během obnovy prostoru dbSPACE nedostupný.

Záloha úrovně 0 kopíruje údaje o aktualizované konfiguraci databázového serveru včetně údajů o novém zrcadleném bloku z rezervovaných stránek kořenového prostoru dbSPACE do zálohy. Jestliže provádíte obnovu dat a primární blok není dostupný, údaje o aktualizované konfiguraci dají na začátku zálohování pokyn databázovému serveru, aby hledal zrcadlené kopie souborů logického protokolu. Pokud tyto údaje o novém zálohování paměťového prostoru neexistují, databázový server nemůže využít zrcadlených souborů protokolu.

Z podobných důvodů není možné zrcadlit prostor dbSPACE, který obsahuje soubor logického protokolu, pokud se vytváří záloha prostoru dbSPACE. Novou informaci, která se musí zobrazovat v prvním bloku pásky zálohy prostoru dbSPACE, není do tohoto místa možné kopírovat po zahájení zálohování.

Více informací o vytváření zrcadlených bloků najdete v části Kapitola 18, "Použití zrcadlení", na stránce 18-1.

Stavové příznaky zrcadlení

Prostory dbSPACE, blobSPACE a sbSPACE mají stavové příznaky, které označují, zda jsou zrcadleny nebo ne.

Zálohu úrovně 0 kořenového prostoru dbSPACE je zapotřebí provést před spuštěním zrcadlení.

Bloky mají stavové příznaky, které poskytují následující údaje:

- Zda je blok primární nebo zrcadlený.
- Zda je blok v současné době v činnosti nebo je poškozený, je-li to nový zrcadlený blok, který vyžaduje zálohu úrovně 0 kořenového prostoru dbSPACE nebo je v průběhu obnovy.

Popis stavových příznaků bloků naleznete v popisu volby **onstat -d** v příručce *IBM Informix Administrator's Reference*. Další údaje, jak zobrazit tyto stavové příznaky, najdete v části "Monitorování využití disku" na stránce 10-34.

Obnova

Když databázový server obnoví zrcadlený blok, provede stejnou obnovovací proceduru, jako když zrcadlení začíná. Proces obnovy zrcadlení kopíruje data z existujícího funkčního bloku do nového opraveného bloku do té doby, než jsou bloky identické.

Při zahájení obnovy databázový server uvede nefunkční blok do režimu obnovy a kopíruje data z funkčního bloku do obnovovaného bloku. Když je obnova dokončena, blok automaticky získá stav online. Stejný postup se provádí při obnově primárního bloku i při obnově zrcadleného bloku.

Rada: Během procesu obnovy je stále možné použít blok, který je v činnosti. Jestliže se data zapisují na stránku, která již byla zkopírována do obnovovaného bloku, databázový server aktualizuje odpovídající stránku v obnovovaném bloku před tím, než pokračuje v procesu obnovování.

Informace o tom, jak obnovit nefunkční blok najdete v informacích o tom, jak obnovit zrcadlený blok, na straně 18-6.

Činnosti v průběhu zpracování

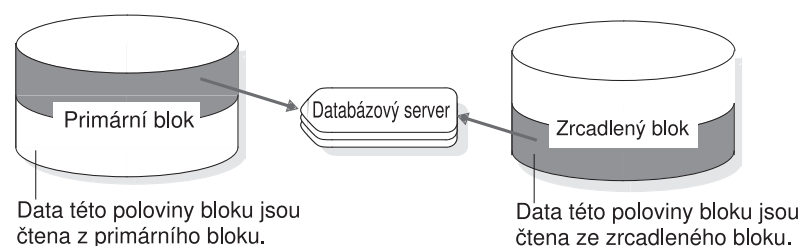
V této části se projednávají některé detaily diskového vstupu - výstupu pro zrcadlené bloky, a to, jak se databázový server u těchto bloků vypořádá s poruchou média.

Diskové zápisy do zrcadlených bloků

Během své činnosti provádí databázový server zrcadlení tak, že provádí pro každou změnu dva paralelní zápisy: jeden do primárního bloku a druhý do zrcadleného bloku.

Diskové čtení ze zrcadlených bloků

Databázový server používá zrcadlení ke zlepšení výkonu čtení, protože se obě verze dat nacházejí samostatných discích. Datová stránka se čte buď z primárního, nebo ze zrcadleného bloku, podle toho, která polovina bloku obsahuje adresu datové stránky. Tato funkce se nazývá *rozdělené čtení*. Rozdělené čtení zlepšuje výkon, protože snižuje dobu přístupu na disk. Doba přístupu na disk se zkrátí, protože maximální vzdálenost, kterou musí urazit disková hlava, se zkrátí o polovinu. Obrázek 17-2 ilustruje rozdělené čtení.



Obrázek 17-2. Rozdělené čtení snižuje maximální vzdálenost, kterou musí urazit disková hlava

Zjišťování poruch média

Databázový server zkontroluje návratový kód, když poprvé otevře blok, a po každém čtení a zápisu. Kdykoli databázový server zjistí, že zařízení primárního (nebo zrcadleného) bloku selhalo, nastaví stavový příznak bloku na vypnuto (D). Informace o stavových příznacích bloků najdete v části “Stavové příznaky zrcadlení” na stránce 17-4.

Pokud databázový server zjistí, že zařízení primárního (nebo zrcadleného) bloku selhalo, pokračují čtení a zápisy jen v bloku, který zůstal v činnosti. To platí i tehdy, pokud administrátor záměrně vyřadí z činnosti jeden z bloků.

Poté, co administrátor obnoví vyřazený blok a vrátí jej do stavu online, čtení se opět rozdělí mezi dva primární zrcadlené bloky a zápisy se provádí v obou blocích.

Obnova bloku

Databázový server používá asynchronní vstup - výstup, aby minimalizoval dobu, která je zapotřebí k obnově bloku. Čtení z bloku, který je online, se může překrývat se zápisem na vypnutý blok, místo toho, aby se oba procesy uskutečňovaly jeden za druhým. To znamená, že jednotkový proces provádějící čtení nemusí při čtení dalších dat čekat, než druhý jednotkový proces dokončí zápis.

Výsledek zastavení zrcadlení

Když se zrcadlení ukončí, databázový server ihned uvolní zrcadlené bloky a prostor je pak dostupný k opětovnému přidělení. Akce ukončení trvá jen několik sekund.

Po ukončení zrcadlení vytvořte zálohu úrovně 0 kořenového prostoru dbSPACE, abyste zajistili, že se rezervované stránky s aktualizovanými údaji o zrcadlených blocích zkopírují do zálohy. Tato akce zabraňuje tomu, aby procedura obnovení předpokládala, že zrcadlená data jsou stále dostupná.

Struktura zrcadleného bloku

Zrcadlený blok obsahuje tytéž řídicí struktury jako primární blok, a to následovně:

- Zrcadla bloků prostoru blobSPACE obsahují režijní stránky prostoru blobSPACE.
- Zrcadla bloků prostoru dbSPACE obsahují režijní stránky prostoru dbSPACE.
- Zrcadla prostorů sbSPACE obsahují stránky metadat.

Další informace o těchto strukturách najdete v části o struktuře zrcadleného bloku v kapitole o diskových strukturách a paměti, v příručce *IBM Informix Administrator's Reference*.

Zobrazení využití diskového prostoru, získané jednou z metod uvedených v části "Monitorování bloků" na stránce 10-34, vždy ukazuje, že zrcadlený blok je zaplněný, dokonce i pokud primární blok obsahuje volné místo. *Zaplnění* zrcadleného bloku označuje, že v bloku není k dispozici volné místo k jiným účelům než k uložení zrcadlených dat primárního bloku. Stav zůstává zaplněný, dokud je primární i zrcadlený blok v činnosti.

Jestliže dojde k vypnutí primárního bloku a ze zrcadleného bloku se stane primární blok, budou zprávy o přidělení diskového prostoru přesně popisovat, do jaké míry je nový primární blok zaplněný.

Kapitola 18. Použití zrcadlení

Obsah kapitoly	18-1
Příprava k zrcadlení dat	18-1
Povolení konfiguračního parametru MIRROR	18-2
Změna parametru MIRROR pomocí programu ON-Monitor (systém UNIX)	18-2
Přidělení diskového prostoru pro zrcadlená data	18-2
Propojení bloků (UNIX)	18-3
Opětovné propojení bloku se zařízením po poruše disku	18-3
Použití zrcadlení.	18-3
Zrcadlení kořenového prostoru dbspace během spuštění	18-4
Změna stavu zrcadlení	18-4
Správa zrcadlení.	18-4
Spuštění zrcadlení u nezrcadlených paměťových prostorů	18-4
Spuštění zrcadlení nezrcadlených prostorů dbspace pomocí programu onspaces	18-4
Spuštění zrcadlení pomocí programu ISA	18-5
Spuštění zrcadlení nezrcadlených prostorů dbspace pomocí programu ON-Monitor (UNIX)	18-5
Spuštění zrcadlení nových paměťových prostorů	18-5
Spuštění zrcadlení nových prostorů pomocí programu onspaces.	18-5
Spuštění zrcadlení nových prostorů pomocí programu ISA	18-5
Spuštění zrcadlení nových prostorů dbspace pomocí programu ON-Monitor (UNIX)	18-5
Přidání zrcadlených bloků	18-5
Přidání zrcadlených bloků pomocí programu onspaces	18-6
Přidání zrcadlených bloků pomocí programu ISA	18-6
Přidání zrcadlených bloků pomocí programu ON-Monitor (UNIX)	18-6
Vypnutí zrcadleného bloku	18-6
Vypnutí zrcadlených bloků pomocí obslužného programu onspaces	18-6
Vypnutí zrcadlených bloků pomocí programu ON-Monitor (systém UNIX)	18-6
Obnovení zrcadleného bloku	18-6
Obnovení zrcadleného bloku pomocí programu onspaces	18-6
Obnovení zrcadleného bloku pomocí programu ISA	18-6
Obnovení zrcadleného bloku pomocí programu ON-Monitor (UNIX)	18-7
Ukončení zrcadlení	18-7
Ukončení zrcadlení pomocí programu onspaces	18-7
Ukončení zrcadlení pomocí programu ON-Monitor (systém UNIX)	18-7
Ukončení zrcadlení pomocí programu ISA	18-7

Obsah kapitoly

Tato kapitola popisuje různé úlohy zrcadlení, které jsou zapotřebí při využívání funkce zrcadlení databázového serveru. Poskytuje přehled kroků potřebných pro zrcadlení dat.

Příprava k zrcadlení dat

Tato část popisuje, jak spustit zrcadlení dat na databázovém serveru, u něhož není funkce zrcadlení povolena.

Příprava k zrcadlení dat:

1. Uvedte databázový server do režimu offline a povolte zrcadlení.
Další informace naleznete v části “Povolení konfiguračního parametru MIRROR” na stránce 18-2.
2. Vraťte databázový server zpět do režimu online.
3. Přidejte zrcadleným blokům diskový prostor.

Tento diskový prostor je možné přidělit vždy, když je diskový prostor k dispozici, jsou-li v dalším kroku určeny zrcadlené bloky. Zrcadlené bloky by měly být na jiném disku, než odpovídající primární bloky. Další informace naleznete v části “Přidělení diskového prostoru pro zrcadlená data” na stránce 18-2.

4. Zvolte prostory dbSPACE, blobSPACE a sSPACE, které chcete zrcadlit a určete název cesty pro zrcadlené bloky a posun každého primárního bloku v tomto paměťovém prostoru. Proces zrcadlení začne, jakmile tento krok provedete. Opakujte tento krok pro všechny paměťové prostory, které chcete zrcadlit. Další informace naleznete v části “Použití zrcadlení” na stránce 18-3.

Povolení konfiguračního parametru MIRROR

Povolení zrcadlení vyvolá funkci databázového serveru potřebnou pro úlohy zrcadlení. Povolením zrcadlení se však proces zrcadlení nespustí. Zrcadlení ve skutečnosti nebude spuštěno, dokud se nevytvoří zrcadlené bloky pro prostory dbSPACE, blobSPACE a sSPACE. Další informace naleznete v části “Použití zrcadlení” na stránce 18-3.

Povolte zrcadlení při spuštění databázového serveru, jestliže máte v plánu vytvořit zrcadlo kořenového prostoru dbSPACE jako součást spuštění, jinak zrcadlení nepovolujte. Jestliže se později rozhodnete zrcadlit paměťový prostor, můžete změnit hodnotu konfiguračního parametru MIRROR.

Pokud chcete povolit zrcadlení databázového serveru, je nutné nastavit parametr MIRROR v souboru **ONCONFIG** na hodnotu 1. Výchozí hodnota parametru MIRROR je 0, což označuje, že zrcadlení je zakázáno.

Nenastavujte parametr MIRROR na 1, jestliže zrcadlení nepoužíváte.

Hodnotu parametru MIRROR je možné změnit úpravou souboru **ONCONFIG** pomocí textového editoru nebo pomocí programu ISA, pokud je databázový server v režimu online. Po změně souboru **ONCONFIG** je třeba uvést databázový server do režimu offline a pak do klidového režimu, aby změna nabyla platnosti.

Změna parametru MIRROR pomocí programu ON-Monitor (systém UNIX)

Chcete-li povolit zrcadlení, zvolte možnost **Parameters > Initialize**. V poli označeném **Mirror**, zadejte Y. Změny zaznamenáte stisknutím tlačítka ESC.

Po poslední obrazovce se objeví výzva, abyste potvrdili, že chcete pokračovat (inicializovat prostor databázového serveru a odstranit všechna existující data). Na tuto výzvu odpovězte N (ne).

Upozornění: Pokud na tuto výzvu odpovíte Y (ano), ztratíte všechna data.

Uveďte databázový sever do režimu offline a pak do klidového režimu, aby změna nabyla platnosti.

Přidělení diskového prostoru pro zrcadlená data

Před vytvořením zrcadleného bloku je nutné pro tento účel přidělit diskový prostor. Zrcadleným blokům je možné přidělit diskový prostor s přímým přístupem nebo prostor předpřipraveného souboru. Informace o přidělování diskového prostoru najdete v části “Přidělení diskového prostoru” na stránce 10-3.

Diskový prostor pro zrcadlený blok vždy přiřadíte na jiném disku, než je odpovídající primární blok, v ideálním případě na disku s jiným řadičem. Toto nastavení umožňuje přístup k zrcadlenému bloku, je-li disk, na němž je umístěn primární blok, nefunkční, a naopak.

Propojení bloků (UNIX)

Chcete-li propojit aktuální soubory nebo přímá zařízení zrcadlených bloků s názvy cesty zrcadlení, použijte příkaz propojení systému UNIX (**ln**). Pokud dojde k poruše disku, je možné propojit nový soubor nebo přímé zařízení s názvem cesty a odstranit tak nutnost fyzicky nahradit vadný disk před tím, než bude blok znovu uveden do činnosti.

Opětovné propojení bloku se zařízením po poruše disku

Pokud v systému UNIX dojde k selhání disku, na kterém je umístěn aktuální zrcadlený soubor nebo přímé zařízení, můžete blok opětovně propojit se souborem nebo přímým zařízením na jiném disku. Tato akce umožňuje obnovit zrcadlený blok před tím, než je disk, který selhal, znovu uveden do činnosti.

Typické příkazy systému UNIX, které můžete použít pro opětovné propojení, jsou zobrazeny v následujících příkladech.

Původní nastavení se skládá z primárního kořenového bloku a ze zrcadleného kořenového bloku, které jsou propojeny se skutečnými přímými diskovými zařízeními následovně:

```
ln -l
lrwxrwxrwx 1 informix 10 May 3 13:38 /dev/root@->/dev/rxy0h
lrwxrwxrwx 1 informix 10 May 3 13:40 /dev/mirror_root@->/dev/rsd2b
```

Předpokládejme, že disk, na němž sídlí přímé zařízení **/dev/rsd2b**, je nefunkční. Odpovídající symbolické propojení můžete odstranit příkazem **rm** následujícím způsobem:

```
rm /dev/mirror_root
```

Nyní je možné znovu propojit název cesty zrcadleného bloku s přímým diskovým zařízením na disku, který je v činnosti a pokračovat v obnovení bloku, a to následovně:

```
ln -s /dev/rab0a /dev/mirror_root
```

Použití zrcadlení

Zrcadlení bude spuštěno, když pro jednotlivé primární bloky v prostoru dbspace, blobspace nebo sbspace vytvoříte zrcadlené bloky.

Když vytvoříte zrcadlený blok, databázový server zkopíruje data z primárního bloku do zrcadleného bloku. Když tento proces skončí, databázový server začne zrcadlit data. Jestliže primární blok obsahuje soubory logického protokolu, databázový server nezkopíruje data ihned po vytvoření zrcadleného bloku, ale čeká na provedení zálohy úrovně 0. Vysvětlení tohoto chování najdete v části “Vytvoření zrcadleného bloku” na stránce 17-4.

Důležité: Vždy je nutné spustit zrcadlení celého prostoru dbspace, blobspace nebo sbspace. Databázový server nedovoluje v prostorech dbspace, blobspace nebo sbspace vybrat pro zrcadlení jednotlivé bloky. Je nutné vytvořit zrcadlený blok pro všechny bloky v prostoru.

Zrcadlení paměťového prostoru se spustí po provedení následujících operací:

- Vytvoření zrcadleného kořenového prostoru dbspace během inicializace systému.
- Změna stavu prostoru dbspace z nezrcadleného na zrcadlený.
- Vytvoření zrcadleného prostoru dbspace, blobspace, nebo sbspace.

Všechny tyto operace vyžadují, aby pro existující bloky byly v paměťovém prostoru vytvořeny zrcadlené bloky.

Zrcadlení kořenového prostoru dbspace během spuštění

Pokud je povoleno zrcadlení při inicializaci databázového serveru, je možné také určit název cesty zrcadlení a posun pro kořenový blok. Databázový server vytvoří zrcadlený blok při inicializaci serveru. Protože však kořenový blok obsahuje soubory logického protokolu, nebude ve skutečnosti zrcadlení spuštěno, dokud nebude provedena záloha úrovně 0.

K určení názvu cesty a posunu zrcadla kořenového prostoru nastavte hodnoty proměnných MIRRORPATH a MIRROROFFSET v souboru ONCONFIG před tím, než spustíte databázový server.

Pokud nezádáte název cesty a posun zrcadlení, ale chcete spustit zrcadlení kořenového prostoru typu dbspace, je nutné změnit stav zrcadlení kořenového prostoru typu dbspace po spuštění databázového serveru.

Změna stavu zrcadlení

Můžete provést následující dvě změny stavu zrcadleného bloku:

- Převést zrcadlený blok z režimu online do vypnutého stavu.
- Změnit stav zrcadleného bloku z vypnutého stavu na stav obnovení.

Blok je možné uvést do vypnutého stavu nebo obnovit pouze tehdy, pokud je částí zrcadleného páru. Primární blok i zrcadlený blok můžete vypnout pouze tehdy, pokud je druhý blok v činnosti.

Informace o tom, jak zjistit stav bloku, najdete v části “Monitorování využití disku” na stránce 10-34.

Správa zrcadlení

Ke správě zrcadlení je možné použít obslužný program **onspaces**. V systému UNIX můžete ke správě zrcadlení použít program ON-Monitor. Úplný popis syntaxe obslužného programu **onspaces** naleznete v informacích o obslužném programu **onspaces** v příručce *IBM Informix Administrator's Reference*.

Spuštění zrcadlení u nezrcadlených paměťových prostorů

Zrcadlení prostorů dbspace, blobspace nebo sbspace můžete připravit kdykoli. Zrcadlení však nebude spuštěno, dokud není provedeno zálohování úrovně 0.

Spuštění zrcadlení nezrcadlených prostorů dbspace pomocí programu onspaces

Zrcadlení prostorů dbspace, blobspace nebo sbspace je možné spustit obslužným programem **onspaces**. Následující příkaz programu **onspaces** například spustí zrcadlení prostoru dbspace **db_project**, který obsahuje dva bloky **data1** a **data2**:

```
onspaces -m db_project\  
-p /dev/data1 -o 0 -m /dev/mirror_data1 0\  
-p /dev/data2 -o 5000 -m /dev/mirror_data2 5000
```

Následující příklad ukazuje, jak zapnout zrcadlení prostoru dbspace, který má název **sp1**. Cestu a posun primárního bloku a cestu a posun zrcadleného bloku je možné zadat buď příkazem, nebo v souboru.

```
onspaces -m sp1 -f mirfile
```

Soubor **mirfile** obsahuje následující řádek:

```
/ix/9.3/sp1 0 /ix/9.2/sp1mir 0
```

V tomto řádku je **/ix/9.3/sp1** cesta k primárnímu bloku, 0 je posun primárního bloku, **/ix/9.3/sp1mir** je cesta k zrcadlenému bloku a 0 je posun zrcadleného bloku.

Spuštění zrcadlení pomocí programu ISA

Zrcadlení se pomocí programu ISA spouští následovně:

1. Vyberte volbu **Storage >Chunks**.
2. Zvolte jméno prostoru dbspace a klepněte na volbu **Spustit zrcadlení**.

Spuštění zrcadlení nezrcadlených prostorů dbspace pomocí programu ON-Monitor (UNIX)

Zrcadlení prostoru dbspace se spustí volbou **Dbspaces > Mirror**.

Prostor dbspace, který chcete zrcadlit, vyberte tak, že přesunete kurzor na správný prostor dbspace v seznamu, a stisknete klávesu **CTRL-B**. Volba **Mirror** pak zobrazí obrazovku pro každý blok v prostoru dbspace. Na této obrazovce pak můžete zadat název cesty a posun **zrcadla**. Po zadání údajů pro všechny bloky opusťte volbu stisknutím klávesy **ESC**. Databázový server obnoví nové zrcadlené bloky, to znamená, že zkopíruje data z primárního bloku do zrcadleného. Jestliže blok obsahuje soubory logického protokolu, bude obnovení odloženo, dokud nebude vytvořena záloha úrovně 0.

Spuštění zrcadlení nových paměťových prostorů

Zrcadlení je také možné spustit při vytváření nového prostoru dbspace, blobspace nebo sbspace.

Spuštění zrcadlení nových prostorů pomocí programu onspaces

Program **onspaces** je možné použít k vytvoření zrcadleného prostoru dbspace. Následující příkaz například vytváří prostor dbspace **db_acct** s původním blokem **/dev/chunk1** a zrcadleným blokem **/dev/mirror_chk1**:

```
onspaces -c -d db_acct -p /dev/chunk1 -o 0 -s 2500 -m /dev/mirror_chk1 0
```

Další způsob, jak spustit zrcadlení, je vybrat volbu **Rejstřík podle obslužného programu > onspaces > -m**.

Spuštění zrcadlení nových prostorů pomocí programu ISA

Spuštění zrcadlení nových paměťových prostorů pomocí programu ISA se provádí takto:

1. Vyberte volbu **Paměť > Prostory**.
2. Klepněte na volbu **Přidat prostor dbspace**, **Přidat prostor blobspace** nebo **Přidat prostor sbspace**.
3. Zadejte cestu a posun zrcadleného bloku.

Spuštění zrcadlení nových prostorů dbspace pomocí programu ON-Monitor (UNIX)

Prostor dbspace se zrcadlením se vytvoří výběrem volby **Dbspaces > Create**. Tato volba zobrazí obrazovku, v níž můžete určit název cesty, posun a velikost primárního bloku a název cesty a posun zrcadleného bloku nového prostoru dbspace.

Přidání zrcadlených bloků

Přidáte-li blok k zrcadlenému prostoru dbspace, blobspace, nebo sbspace, musíte také přidat odpovídající zrcadlený blok.

Přidání zrcadlených bloků pomocí programu onspaces

K přidání primárního bloku a jeho zrcadleného bloku k prostoru dbspace, blobspace, nebo sbospace můžete použít obslužný program **onspaces**. Následující příklad přidává blok **chunk2** k prostoru dbspace **db_acct**. Protože prostor dbspace je zrcadlený, je také přidán zrcadlený blok **mirror_chk2**.

```
onspaces -a db_acct -p /dev/chunk2 -o 5000 -s 2500 -m /dev/mirror_chk2 5000
```

Přidání zrcadlených bloků pomocí programu ISA

Přidat zrcadlení bloků pomocí programu ISA:

1. Vyberte volbu **Storage >Chunks**.
2. Vyberte jméno prostoru dbspace a klepněte na volbu **Add Chunk**.
3. Zadejte cestu a posun zrcadleného bloku.

Přidání zrcadlených bloků pomocí programu ON-Monitor (UNIX)

Volba **Dbspaces > Add-chunk** v programu ON-Monitor zobrazí pole, do kterých má být zadán název cesty, posun a velikost primárního zrcadleného bloku a název cesty a posun zrcadleného bloku.

Vypnutí zrcadleného bloku

Je-li zrcadlený blok *vypnutý*, databázový server do něj nemůže zapisovat ani z něj číst. Zrcadlený blok můžete vypnout a pak ho znovu propojit s jiným zařízením. Další informace naleznete v části “Opětovné propojení bloku se zařízením po poruše disku” na stránce 18-3.)

Vypnutí bloku není totéž, co ukončení zrcadlení. Zrcadlení se ukončuje pro celý prostor dbspace a způsobí, že server vypustí všechny zrcadlené bloky tohoto prostoru dbspace.

Vypnutí zrcadlených bloků pomocí obslužného programu onspaces

Blok můžete vypnout pomocí obslužného programu **onspaces**. V následujícím příkladu je vypnut blok, který je součástí prostoru dbspace **db_acct**:

```
onspaces -s db_acct -p /dev/mirror_chk1 -o 0 -D
```

Vypnutí zrcadlených bloků pomocí programu ON-Monitor (systém UNIX)

Chcete-li použít program ON-Monitor k vypnutí zrcadleného bloku, zvolte položku **Dbspaces > Status**. Umístěte kurzor na prostor dbspace obsahující blok, který chcete vypnout a stiskněte klávesu F3 nebo CTRL-B. Databázový server zobrazí na obrazovce seznam všech bloků v prostoru dbspace. Umístěte kurzor na blok, který chcete vypnout a stisknutím klávesy F3 nebo CTRL-B změňte jeho stav (na vypnutý).

Obnovení zrcadleného bloku

Chcete-li zrcadlit data v bloku, který je v režimu online, musíte vypnutý blok obnovit.

Obnovení zrcadleného bloku pomocí programu onspaces

Zrcadlený blok můžete obnovit příkazem **onspaces -s**. Chcete-li například obnovit blok, jehož cesta je **/dev/mirror_chk1** a posun 0 kilobajtů, zadejte následující příkaz:

```
onspaces -s db_acct -p /dev/mirror_chk1 -o 0 -0
```

Obnovení zrcadleného bloku pomocí programu ISA

Obnovení zrcadleného bloku pomocí programu ISA:

Vyberte volbu **Rejstřík podle obslužného programu > onspaces > -s**.

Obnovení zrcadleného bloku pomocí programu ON-Monitor (UNIX)

Chcete-li k použití program ON-Monitor k obnově vypnutého bloku, zvolte možnost **Dbspaces > Status**.

Ukončení zrcadlení

Při ukončení zrcadlení prostoru dbspace, blobspace nebo sbspace databázový server okamžitě uvolní zrcadlené bloky tohoto prostoru. Tyto bloky jsou ihned k dispozici k opětovnému přidělení jiným paměťovým prostorům.

Zrcadlení mohou v systému UNIX ukončit pouze uživatelé **informix** a **root** a v systému Windows členové skupiny **Informix-Admin**.

Zrcadlení není možné ukončit, jestliže jsou některé z primárních bloků v prostoru dbspace vypnuté. Systém může být při ukončení zrcadlení ve režimu online.

Ukončení zrcadlení pomocí programu onspaces

Zrcadlení je možné ukončit pomocí programu **onspaces**. Chcete-li například ukončit zrcadlení kořenového prostoru dbspace, zadejte následující příkaz:

```
onspaces -r rootdbs
```

Další způsob, jak ukončit zrcadlení, je vybrat volbu **Rejstřík podle obslužného programu > onspaces > -r**.

Ukončení zrcadlení pomocí programu ON-Monitor (systém UNIX)

Chcete-li ukončit zrcadlení prostoru dbspace nebo blobspace pomocí programu ON-Monitor, vyberte volbu **Dbspaces > Mirror**. Vyberte zrcadlený prostor dbspace nebo blobspace a stiskněte klávesu CTRL-B nebo F3.

Ukončení zrcadlení pomocí programu ISA

Chcete-li ukončit zrcadlení pomocí programu ISA, proveďte následující kroky:

1. Vyberte volbu **Storage > Chunks**.
2. Vyberte název prostoru dbspace a klepněte na volbu **Zastavit zrcadlení**.

Kapitola 19. Přehled replikace dat (vydání Enterprise/Workgroup)

Obsah kapitoly	19-2
Replikace dat	19-2
Režimy serveru	19-3
Typy replikovaných dat	19-3
Primární a sekundární databázové servery	19-3
Výhody replikace dat	19-4
Porovnání replikace dat a zrcadlení	19-4
Porovnání replikace dat a dvoufázového potvrzování	19-5
Replikace dat a replikace Enterprise Replication	19-5
Způsob činnosti replikace dat	19-6
Průběh prvotní replikace dat	19-6
Reprodukce aktualizací primárního databázového serveru	19-7
Jak jsou odesílány záznamy protokolu na sekundární server replikace HDR	19-7
Jak jsou odesílány záznamy protokolu na sekundární server RS a SD	19-7
Vyrovnávací paměti replikace dat HDR	19-7
Kdy jsou záznamy odesílány	19-8
Synchronní aktualizace replikace HDR	19-8
Asynchronní aktualizace replikace HDR	19-8
Jednotkové procesy, které obsluhují replikaci dat	19-10
Kontrolní body mezi databázovými servery	19-11
Jak je zaznamenávána synchronizace dat	19-11
Příklady konfigurace replikace dat	19-12
Příklady konfigurace vzdálených samostatných serverů	19-12
Příklady konfigurace sekundárních sdílených disků	19-14
Použití replikace Enterprise Replication jako části obnovitelné skupiny	19-18
Příklad konfigurace vysoce dostupných klastrů s replikací Enterprise Replication	19-18
Příklad komplexní strategie obnovy pro případ poruchy	19-19
Přesměrování a propojitelnost pro klienty replikace dat	19-23
Návrh klientů pro přesměrování	19-24
Automatické přesměrování klientů pomocí proměnné prostředí DBPATH	19-24
Jak pracuje metoda přesměrování pomocí proměnné prostředí DBPATH	19-24
Co by měli provést administrátoři	19-25
Co by měli provést uživatelé	19-25
Směrování klientů pomocí informací o propojitelnosti	19-25
Jak pracuje přesměrování pomocí informací o propojitelnosti	19-25
Změna informací o propojitelnosti	19-26
Připojení k databázovému serveru	19-27
Automatické přesměrování ve skupině serverů	19-28
Směrování klientů pomocí proměnné prostředí INFORMIXSERVER	19-28
Jak pracuje přesměrování pomocí proměnné prostředí INFORMIXSERVER	19-28
Co by měli provést administrátoři	19-29
Co by měli provést uživatelé	19-29
Přesměrování z aplikace	19-29
Porovnání různých mechanismů přesměrování	19-30
Návrh klientů skupiny replikace dat	19-31
Nastavení režimu uzamykání při přístupu k primárnímu databázovému serveru na čekání	19-31
Návrh klientů k použití sekundárního databázového serveru	19-32
Žádné příkazy ke změně dat	19-32
Úroveň uzamčení a izolace	19-32

Obsah kapitoly

Tato kapitola obsahuje obecné informace o možnostech replikace dat, včetně sekundárního serveru HDR, sekundárního serveru SD a sekundárního serveru RS. Na vašem serveru IDS Dynamic Server tyto možnosti replikace dat mohou a nemusí být k dispozici (viz níže uvedená tabulka).

Tabulka 19-1. Dostupné možnosti replikace dat na serveru IDS Dynamic Server

Vlastnost	IDS Express	IDS vydání Workgroup	IDS vydání Enterprise
sekundární server RS	Není k dispozici	Není k dispozici	Ano
sekundární server HDR	Není k dispozici	Volitelný	Ano
sekundární server SD	Není k dispozici	Volitelný	Volitelný

V této kapitole jsou popsána následující témata:

- Pojem replikace dat
- Způsob činnosti replikace dat
- Způsob zpracování chyb při replikaci dat
- Jaká posouzení návrhu je třeba provést v případě aplikací, které se připojují k sekundárnímu serveru replikace dat.

Replikace dat

Replikace dat označuje proces reprezentace objektů databáze na více než jednom samostatném místě. Replikace dat poskytuje několik možností konfigurace.

Tabulka 19-2. Typy sekundárních serverů replikace dat.

Typ serveru	Popis
Sekundární server replikace HDR	Umožňuje synchronní replikaci dat Dynamic Server. Sekundární server HDR použijte tehdy, pokud potřebujete okamžitě použitelnou zálohu. Sekundární server HDR poskytuje způsob, jak spravovat záložní kopii celého databázového serveru, ke které mohou aplikace rychle získat přístup v případě závažného selhání.
Sekundární server SD	Server, který sdílí místo na disku s primárním serverem. Primární server má k disku nebo diskovému poli přístup pouze pro zápis a všechny sekundární servery SD mají přístup pouze pro čtení. Sekundární server SD neuchovává kopii fyzické databáze ve vlastním prostoru na disku, ale sdílí disky s primárním serverem.
Sekundární server RS	Server, který je asynchronně aktualizován z primárního serveru. Sekundární servery RS mohou být geograficky vzdálené od primárního serveru a v případech obnovení při havárii mohou sloužit jako vzdálené záložní servery. Každý sekundární server RS uchovává úplnou kopii databáze, aktualizace jsou přenášeny asynchronně z primárního serveru prostřednictvím zabezpečeného síťového připojení.

Jedním ze způsobů replikace dat je prosté zkopírování databáze do databázového serveru nainstalovaného v jiném počítači. Tato kopie umožňuje sestavám přistupovat k datům, aniž by byl narušen provoz klientských aplikací, které přistupují k původní databázi.

Pomocí replikace dat databázový server implementuje téměř transparentní aktualizace celých databázových serverů. Všechna data spravovaná jedním databázovým serverem jsou replikována a dynamicky aktualizována na jednom nebo více databázových serverech, které se často nachází na geograficky vzdáleném místě.

Režimy serveru

V následující tabulce jsou uvedeny možné režimy serveru pro konfigurace replikace dat.

Režim serveru	Popis
Standardní režim	Není součástí systému replikace dat.
Primární režim	Primární režim systému replikace dat. Data lze aktualizovat.
Sekundární režim	Sekundární režim systému replikace dat. Data nelze aktualizovat, lze je však číst.

Databázový server spuštěný ve standardním režimu nebo databázový server HDR spuštěný v primárním nebo v sekundárním režimu se mohou nacházet v jakémkoli provozním režimu databázového serveru, například v klidovém režimu nebo v režimu online. Informace o provozních režimech naleznete v části “Provozní režimy databázového serveru” na stránce 4-7.

Typy replikovaných dat

Replikace dat replikuje data prostorů dbspace a sbspace, ale nereplikuje data v prostorů blobspace.

Všechny vestavěné i rozšířené datové typy jsou replikovány na sekundární server. Uživatelské datové typy (UDT) musejí být protokolovány a uloženy v jediném databázovém serveru. Datové typy s daty ukládanými mimo řádek jsou replikovány tehdy, pokud jsou jejich data uložena v prostoru sbspace nebo v jiné tabulce téhož databázového serveru. Aby mohla být replikována data prostoru sbspace, musí být prostor sbspace protokolován.

Data uložená v souborech operačního systému, v trvalých externích souborech ani v paměťových objektech přidružených k uživatelským rutinám nejsou replikována.

Uživatelské datové typy, uživatelské rutiny a moduly DataBlade vyžadují zvláštní postup instalace a registrace. Pokyny naleznete v části “Průběh prvotní replikace dat” na stránce 19-6.

Primární a sekundární databázové servery

Pokud konfiguruje sadu databázových serverů k použití replikace dat, bude jeden databázový server nazýván *primární* databázový server a ostatní servery budou nazývány *sekundární* databázové servery. (V tomto kontextu bude databázový server, který nepoužívá replikaci dat, nazýván *standardní* databázový server.) Sekundární server může zahrnovat libovolnou kombinaci sekundárních serverů SD, RS a HDR.

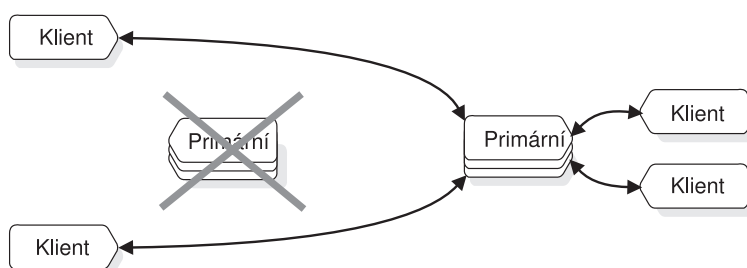
Při normálním provozu se klienti mohou připojovat k primárnímu databázovému serveru a používat ho jako běžný databázový server. Klienti mohou při normálním provozu používat také sekundární databázový server, ale pouze ke čtení dat. Sekundární databázový server nedovoluje klientským aplikacím data aktualizovat.

Jak znázorňuje Obrázek 19-1, bude sekundární databázový server dynamicky aktualizován, jakmile dojde ke změně dat spravovaných primárním databázovým serverem.



Obrázek 19-1. Primární a sekundární databázový server v konfiguraci replikace dat.

Pokud jeden z databázových serverů selže, jak znázorňuje Obrázek 19-2 budete moci přeměrovat klienty používající tento databázový server na jiný databázový server v páru, který se stane primárním serverem.



Obrázek 19-2. Databázové servery v konfiguraci replikace dat a klienti po selhání.

Výhody replikace dat

Replikace dat má následující výhody:

- Klienti na straně, do které jsou data replikována, zaznamenají zvýšení výkonu, protože mohou k datům přistupovat přímo a nemusí se připojovat ke vzdálenému serveru prostřednictvím sítě.
- Klienti všech serverů zaznamenají zlepšení dostupnosti replikovaných dat. Pokud je místní kopie replikovaných dat nedostupná, mohou klienti stále ještě přistupovat ke vzdálené kopii dat.

Tyto výhody však s sebou přinášejí i nedostatky. Replikace dat vyžaduje větší množství diskové paměti a zpracování aktualizace replikovaných dat může trvat déle než aktualizace jediného objektu.

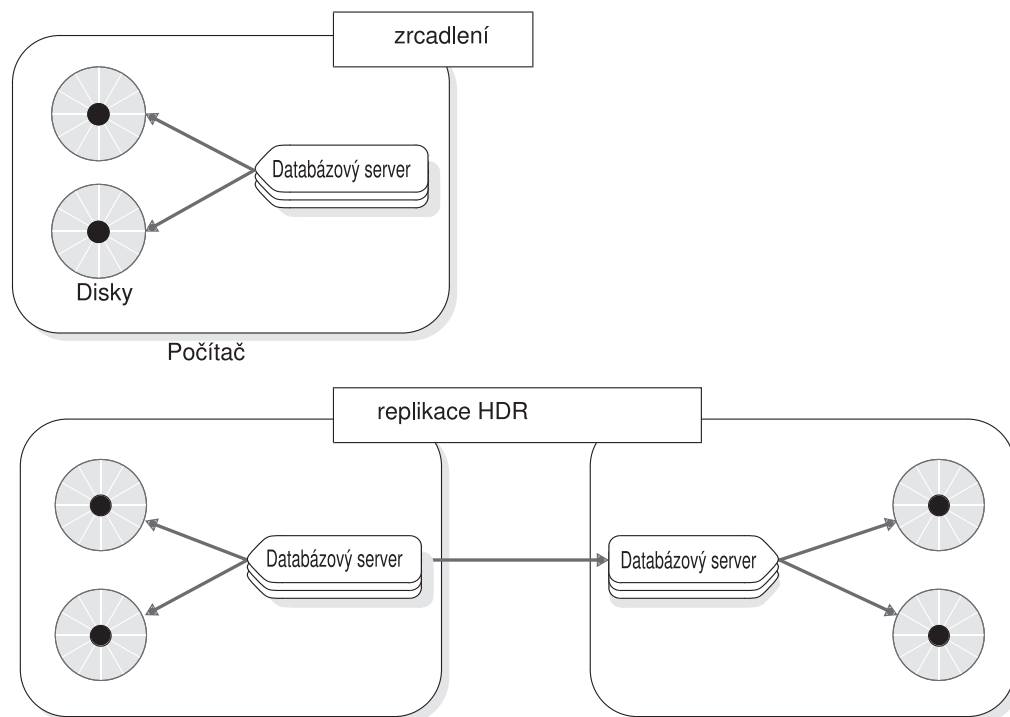
Replikaci dat by bylo možné implementovat v klientské aplikaci pomocí explicitního určení umístění dat, která mají být aktualizována. Tento způsob replikace dat je však náročný, není odolný vůči chybám a znesnadňuje správu. Koncept replikace dat je naopak často spojován s *transparentní replikací*. Replikace vestavěná do databázového serveru (nikoli do aplikace) je transparentní, databázový server sám vyhledává a spravuje repliky dat.

Porovnání replikace dat a zrcadlení

Replikace dat i zrcadlení jsou transparentní způsoby zvýšení odolnosti databázového serveru proti selhání. Jak však znázorňuje Obrázek 19-3, mezi oběma funkcemi jsou značné rozdíly.

Zrcadlení, které popisuje část "Zrcadlení" na stránce 17-1, je mechanismus, pomocí kterého databázový server udržuje kopii konkrétního prostoru dbspace na samostatném disku. Tento mechanismus ochraňuje data zrcadlených prostorů dbspace v případě selhání disku, protože databázový server automaticky aktualizuje data na obou discích a v případě selhání jednoho z prostorů dbspace automaticky použije druhý disk.

Replikace dat však všechna data spravovaná databázovým serverem (nikoli pouze určené prostory dnbspace) replikuje ve zcela odděleném počítači. Protože replikace dat používá dva samostatné databázové servery, chrání data spravovaná těmito servery nejen před selháním disků, ale proti všem typům selhání databázového serveru, včetně selhání počítače nebo v případě závažného selhání celé lokality.



Obrázek 19-3. Porovnání zrcadlení a replikace dat

Porovnání replikace dat a dvoufázového potvrzování

Protokol dvoufázového potvrzování, který podrobně popisuje Kapitola 24, “Protokoly vícefázového potvrzování”, na stránce 24-1, zaručuje, že transakce budou shodně potvrzovány nebo odvolávány ve více databázových serverech současně.

Teoreticky můžete využít k replikaci dat dvoufázové potvrzování, a to tak, že nakonfigurujete dva databázové servery s identickými daty a pak vytvoříte spouštěče, které budou replikovat aktualizace do druhé databáze. Tento způsob se však v různých scénářích selhání potýká s mnoha problémy synchronizace. Výkon distribuovaných transakcí je také menší než výkon replikace dat.

Replikace dat a replikace Enterprise Replication

K vytvoření robustního systému replikace můžete použít kombinaci replikace dat a replikace Enterprise Replication. Replikace dat může zaručit, že systém replikace Enterprise Replication zůstane trvale zcela připojen tím, že poskytuje záložní databázové servery pro kritické replikační uzly.

Pokud použijete kombinaci replikace dat a replikace Enterprise Replication, bude k systému replikace Enterprise Replication připojen pouze primární server. Replikace Enterprise Replication se nebudou účastnit žádné sekundární servery, pokud nedojde k selhání primárního serveru.

Další informace naleznete v částech *IBM Informix Dynamic Server Enterprise Replication Guide* a “Použití replikace Enterprise Replication jako části obnovitelné skupiny” na stránce 19-18.

Způsob činnosti replikace dat

Tato část popisuje mechanismy, které databázový server používá k provádění replikace dat na sekundární servery. Pokyny, jak nastavit, spustit a spravovat různé typy sekundárních serverů uvádí:

Tabulka 19-3.

Typ sekundárního serveru	Další informace naleznete v části
Sekundární replikace HDR	Další informace naleznete v části Kapitola 20, “Použití replikace HDR (verze pro podniky/pracovní skupiny)”, na stránce 20-1, a informace o spuštění páru replikace HDR pomocí externí zálohy a obnovy v části <i>IBM Informix Backup and Restore Guide</i> .
Sekundární server RS	další informace naleznete v části Kapitola 21, “Použití serverů RS (remote standalone) (vydání Enterprise)”, na stránce 21-1.
Sekundární server SD	Další informace naleznete v části Kapitola 22, “Použití sekundárních serverů se sdíleným diskem (vydání Enterprise)”, na stránce 22-1.

Průběh prvotní replikace dat

Sekundární servery replikace HDR a RS použijí zálohy paměťových prostorů a logického protokolu (uložené na pásku nebo na disk), aby provedl počáteční replikaci dat primárního databázového serveru do sekundárního databázového serveru.

Sekundární servery SD nevyžadují zálohu a obnovení z primárního serveru, protože sekundární servery SD sdílí jako primární stejné disky.

Postup replikace dat:

1. Nainstalujte v obou databázových serverech uživatelské typy, uživatelské rutiny a moduly DataBlade a zaregistrujte je pouze v primárním databázovém serveru.
2. Vytvořte zálohu úrovně 0 všech paměťových prostorů primárního databázového serveru a obnovte všechny paměťové prostory z této zálohy v sekundárním databázovém serveru páru replikace dat. Po této operaci bude většina dat spravovaných oběma databázovými servery totožná.

Tento krok není u sekundárních serverů SD vyžadován.

Sekundární databázový server, který byl v předchozím kroku obnoven ze zálohy paměťových prostorů, pak načte všechny záznamy logického protokolu vytvořené po zálohování primárního databázového serveru.

Databázový server načte záznamy logického protokolu nejprve ze všech zálohovaných souborů logického protokolu, které se již nenacházejí na disku a pak je začne načítat ze souborů logického protokolu na disku.

Podrobné pokyny k provedení těchto kroků naleznete v části “První spuštění replikace HDR” na stránce 20-6. Příručka *IBM Informix Backup and Restore Guide* popisuje, jak spustit replikaci pomocí obslužného programu ON-Bar.

Počáteční inicializaci zálohy je třeba provést s použitím zálohy paměťových prostorů. K inicializaci nelze použít nástroje k migraci dat (například **onload** nebo **onunload**), protože aby replikace dat mohla pracovat správně, musejí být rozvržení fyzických stránek jednotlivých tabulek v obou databázových serverech totožná.

Jakmile začne replikace dat pracovat, bude se primární databázový server nacházet v režimu online a bude přijímat aktualizace a dotazy stejným způsobem, jako kdyby se jednalo o standardní databázový server. Sekundární databázový server se bude nacházet v režimu logického obnovení a nebude moci přijímat příkazy jazyka SQL, které by způsobily zápisy na disk (s výjimkou řazení a dočasných tabulek).

Reprodukce aktualizací primárního databázového serveru

Všechny typy sekundárních serverů používají k replikaci dat z primárního serveru protokoly. Sekundární servery RS vyžadují protokolování indexových stránek, ale sekundární servery HDR a SD také používají protokolování indexových stránek, pokud je povoleno. Pro sekundární servery replikace HDR a RS jsou aktualizace prováděné na primárním serveru replikovány na sekundární server tak, že primární server odesílá všechny vytvářené záznamy logického protokolu do sekundárního serveru. Sekundární servery replikace HDR a RS přijímají záznamy logického protokolu generované primárním serverem a používají je na vlastní prostory dbspace. V případě sekundárních serverů SD odesílá primární server sekundárnímu serveru SD umístění ve stránce logického protokolu. Podle umístění v protokolu získaném z primárního serveru čte sekundární server SD z disku stránku logického protokolu a použije ji na vyrovnávací paměti dat.

Důležité: Databázový server nemůže replikovat aktualizace do databázi, které nepoužívají protokolování transakcí.

Jak jsou odesílány záznamy protokolu na sekundární server replikace HDR

Jak znázorňuje Obrázek 19-4 na stránce 19-8, jakmile primární databázový server začne vyprazdňovat obsah vyrovnávací paměti logického protokolu ve sdílené paměti do logického protokolu na disku, zkopíruje také obsah vyrovnávací paměti logického protokolu do *vyrovnávací paměti replikace dat* primárního databázového serveru. Primární databázový server pak odešle tyto záznamy logického protokolu do sekundárního databázového serveru replikace HDR.

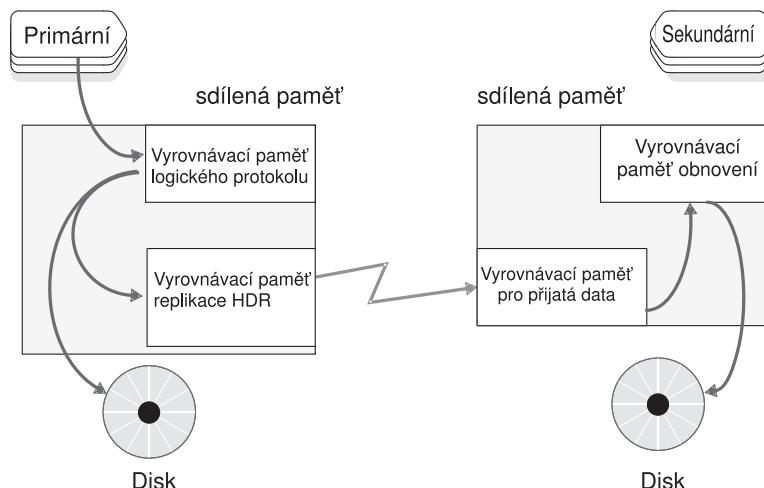
Sekundární databázový server replikace HDR přijme záznamy logického protokolu z primárního databázového serveru do *vyrovnávací paměti pro přijatá data* ve sdílené paměti (její velikost databázový server automaticky přizpůsobuje množství odeslaných dat). Sekundární databázový server pak použije záznamy logického protokolu k aktualizaci dat pomocí logické obnovy.

Jak jsou odesílány záznamy protokolu na sekundární server RS a SD

U sekundárních serverů RS se mohou stránky protokolu ukládat do mezipaměti při jejich vyprázdňení na disk nebo je lze číst přímo z disku. V případě sekundárních serverů SD se protokoly vůbec neodesílají, odesílá se pouze umístění v protokolu.

Vyrovnávací paměti replikace dat HDR

Vyrovnávací paměti replikace dat jsou součástí sdílené paměti, kterou spravuje databázový server. Ve vyrovnávacích pamětech replikace HDR jsou uloženy záznamy logického protokolu, než je primární databázový server odešle do sekundárního databázového serveru HDR. Vyrovnávací paměti replikace HDR mají stejnou velikost, jako vyrovnávací paměť logického protokolu. Tento koncept znázorňuje Obrázek 19-4.



Obrázek 19-4. Jak primární databázový server odesílá záznamy logického protokolu do sekundárního databázového serveru.

Kdy jsou záznamy odesílány

Primární databázový server odesílá obsah vyrovnávací paměti replikace dat do sekundárního serveru HDR *synchronně* nebo *asynchronně*. Zda databázový server bude používat synchronní nebo asynchronní aktualizaci určuje hodnota konfiguračního parametru DRINTERVAL souboru ONCONFIG. Další informace o parametru DRINTERVAL naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.

V případě sekundárních serverů RS se indexové stránky odesílají *asynchronně* z primárního serveru pomocí protokolování indexových stránek. V případě sekundárních serverů SD se replikuje pouze umístění v logickém protokolu.

Synchronní aktualizace replikace HDR

Pokud nastavíte parametr DRINTERVAL na hodnotu -1, bude replikace dat na sekundární server HDR probíhat *synchronně*. Jakmile databázový server zapíše obsah vyrovnávací paměti logického protokolu do vyrovnávací paměti replikace HDR, odešle tyto záznamy z vyrovnávací paměti do sekundárního databázového serveru replikace HDR. K vyprázdnění vyrovnávací paměti logického protokolu v primárním databázovém serveru dojde až po přijetí potvrzení o přijetí záznamů protokolu od sekundárního databázového serveru replikace HDR.

Pokud dojde k selhání při synchronní aktualizaci, žádné transakce potvrzené v primárním databázovém serveru nezůstanou v sekundárním databázovém serveru replikace HDR nepotvrzené ani částečně potvrzené.

Asynchronní aktualizace replikace HDR

Pokud nastavíte parametr DRINTERVAL na jinou hodnotu než -1, bude replikace dat na sekundární server replikace HDR probíhat *asynchronně*. Primární databázový server po zkopírování obsahu vyrovnávací paměti logického protokolu do vyrovnávací paměti replikace HDR vyprázdní vyrovnávací paměť logického protokolu. Nezávisle na této akci odešle primární databázový server obsah vyrovnávací paměti replikace HDR prostřednictvím sítě, pokud nastane jeden z následujících stavů:

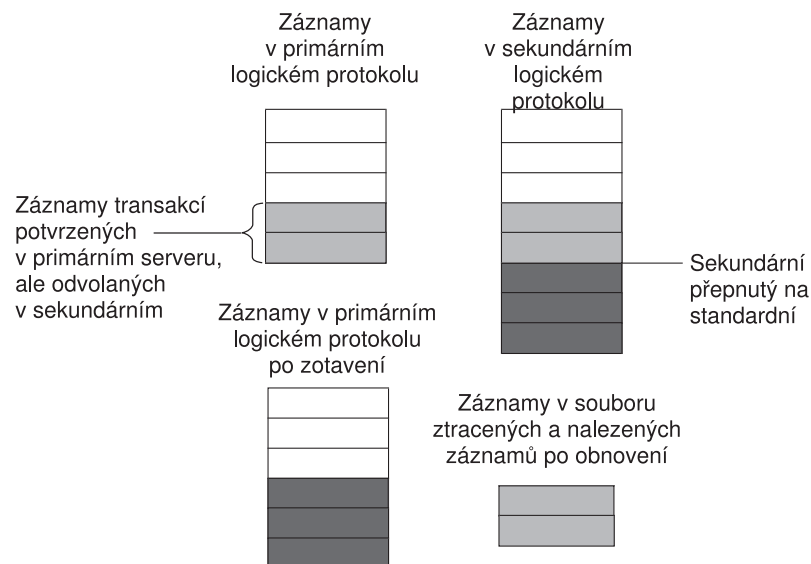
- Dojde k zaplnění vyrovnávací paměti replikace HDR.

- Od posledního odeslání záznamů do sekundárního databázového serveru uplynul časový interval určený konfiguračním parametrem DRINTERVAL primárního databázového serveru.

Tato metoda aktualizace poskytuje vyšší výkon než synchronní aktualizace. Jak je však vysvětleno v následující části, může dojít ke ztrátám transakcí.

Ztracené a nalezené transakce: Při asynchronní aktualizaci může dojít k tomu, že transakce potvrzená v primárním serveru nebude replikovaná do sekundárního databázového serveru. K této situaci může dojít, pokud dojde k selhání po zkopírování záznamu o potvrzení transakce do vyrovnávací paměti replikace HDR a před odesláním tohoto záznamu databázovým serverem do sekundárního databázového serveru.

Pokud je sekundární databázový server po selhání primárního databázového serveru změněn na standardní databázový server, odvolá všechny otevřené transakce. Mezi tyto transakce patří i ty transakce, které byly v primárním databázovém serveru potvrzeny, ale sekundární databázový server neobdržel záznam o jejich potvrzení. V důsledku toho mohou transakce, které jsou v primárním serveru potvrzené, zůstat v sekundárním serveru nepotvrzené. Jakmile po selhání restartujete replikaci dat, umístí databázový server v průběhu obnovy primárního databázového serveru všechny záznamy logického protokolu týkající se ztracených transakcí do souboru, jehož název určuje konfigurační parametr DRLOSTFOUND. Tento proces znázorňuje Obrázek 19-5.



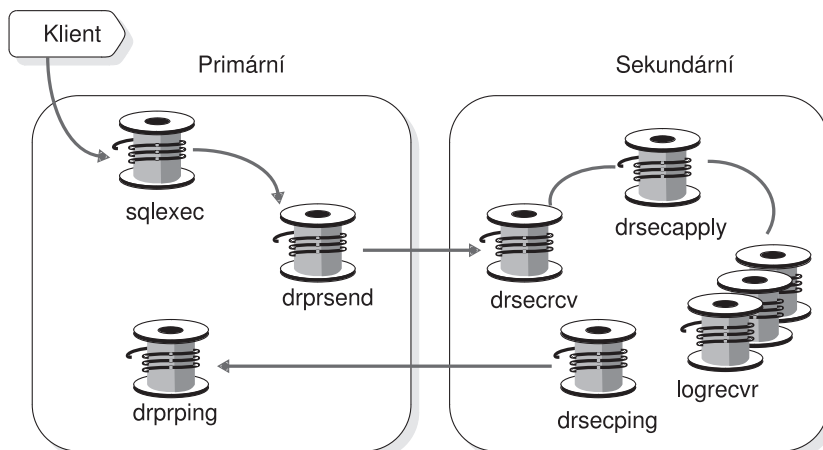
Obrázek 19-5. Použití souboru ztracených a nalezených transakcí.

Pokud v počítači, ve kterém je spuštěn primární databázový server, naleznete po restartování replikace dat soubor ztracených a nalezených transakcí, znamená to, že došlo ke ztrátě transakcí. Databázový server nemůže znovu použít záznamy o transakcích uložené v souboru ztracených a nalezených transakcí k aktualizaci sekundárního serveru, protože v sekundárním databázovém serveru mohly být mezitím uskutečněny konfliktní aktualizace, zatímco byl sekundární databázový server spuštěn v režimu standardního databázového serveru.

Pokud chcete snížit riziko ztráty transakcí při replikaci dat spuštěné v asynchronním režimu, používejte pro všechny databáze protokolování bez vyrovnávací paměti. Tato metoda omezuje dobu, která uplyne mezi zápisem a přenosem záznamů o transakci z primárního do sekundárního databázového serveru.

Jednotkové procesy, které obsluhují replikaci dat

Primární databázový server spouští k podpoře replikace dat specializované jednotkové procesy. Jak znázorňuje Obrázek 19-6, jednotkový proces primárního databázového serveru nazývaný **drprsend** odesílá obsah vyrovnávací paměti primárního serveru prostřednictvím sítě do jednotkového procesu sekundárního databázového serveru nazývaného **drsecrcv**.

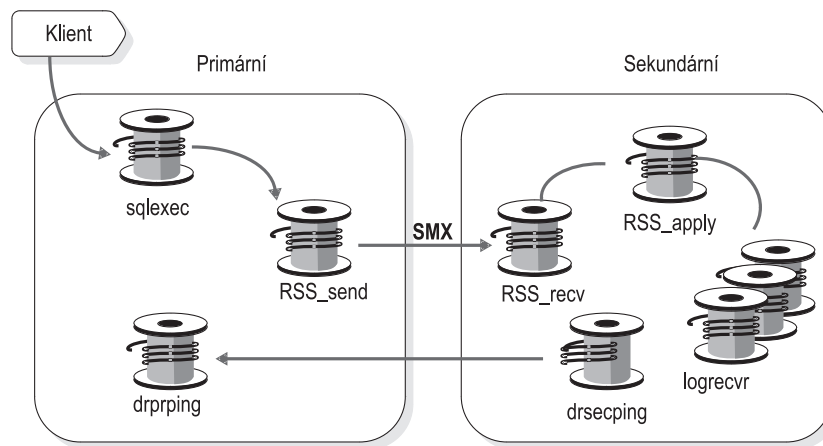


Obrázek 19-6. Jednotkové procesy, které spravují replikaci dat.

Jednotkový proces sekundárního databázového serveru zvaný **drsecapply** kopíruje obsah přijímací vyrovnávací paměti do vyrovnávací paměti obnovy. Jeden nebo více jednotkových procesů **logrecvr** provádí logickou obnovu pomocí obsahu vyrovnávací paměti obnovy a používá záznamy logického protokolu k aktualizaci prostorů dbspace spravovaných databázovým serverem. Počet použitých jednotkových procesů **logrecvr** určuje konfigurační parametr `OFF_RECOVERY_THREADS`.

Nakonec databázový server spouští také jednotkové procesy **drpping** a **drsecping**, které zodpovídají za odesílání a přijímání zpráv, které udávají, zda jsou oba databázové servery propojené.

Pro podporu sekundárních serverů RS primární server také kontroluje, zda jsou připojeny sekundární servery RS. Pokud je tomu tak, zkopíruje stránku do mezipaměti protokolu, která se používá k odeslání stránky na sekundární server RS. Další informace naleznete v části Obrázek 19-6



Obrázek 19-7. Jednotkové procesy, které spravují replikaci dat pro sekundární servery RS.

Jednotkový proces RSS_Send přenáší stránku protokolu na sekundární server RS. Další stránka, kterou je třeba odeslat, možná není v mezipaměti protokolu. V tom případě bude jednotkový proces RSS_Send číst stránky protokolu přímo z disku. Jednotkový proces RSS_Send odesílá data prostřednictvím SMX plně duplexně. Při plně duplexní komunikaci jednotkový proces nečeká před odesláním další vyrovnávací paměti na potvrzení od sekundárního serveru RS. Předtím, než primární server vyžaduje potvrzení, se odešle až 32 vyrovnávacích pamětí. Pokud je dosažen limit 32 vyrovnávacích pamětí, počká odesílací jednotkový proces, až jednotkový proces RSS_Recv přijme potvrzení od sekundárního serveru RS.

Na sekundárním serveru RS se jednotkový proces RSS_Recv prostřednictvím SMX přijme stránky protokolu z primárního serveru.

Kontrolní body mezi databázovými servery

Kontrolní body mezi primárním serverem a sekundární replikací HDR nebo mezi primárním serverem a sekundárním serverem SD jsou synchronní bez ohledu na hodnotu DRINTERVAL. (Viz “Kontrolní body” na stránce 15-5.) Kontrolní bod je v primárním databázovém serveru dokončen až po dokončení kontrolního bodu v sekundárním databázovém serveru. Pokud kontrolní bod nebyl dokončen v čase určeném konfiguračním parametrem DRTIMEOUT, bude primární databázový server předpokládat, že došlo k selhání. Další informace naleznete v části “Definice selhání replikace HDR” na stránce 20-18.

Rada: Synchronizujte čas v operačních systémech obou databázových serverů v páru a nastavte v obou serverech parametr DRTIMEOUT na shodnou hodnotu.

Kontrolní body mezi primárním serverem a sekundárním serverem RS jsou asynchronní a vyžadují, aby bylo povoleno protokolování indexových stránek. Další informace naleznete v části “Protokolování indexových stránek” na stránce 21-3.

Jak je zaznamenávána synchronizace dat

Aby mohl zaznamenávat synchronizaci, zaznamenává každý z databázových serverů v páru ve své vyhrazené stránce následující informace:

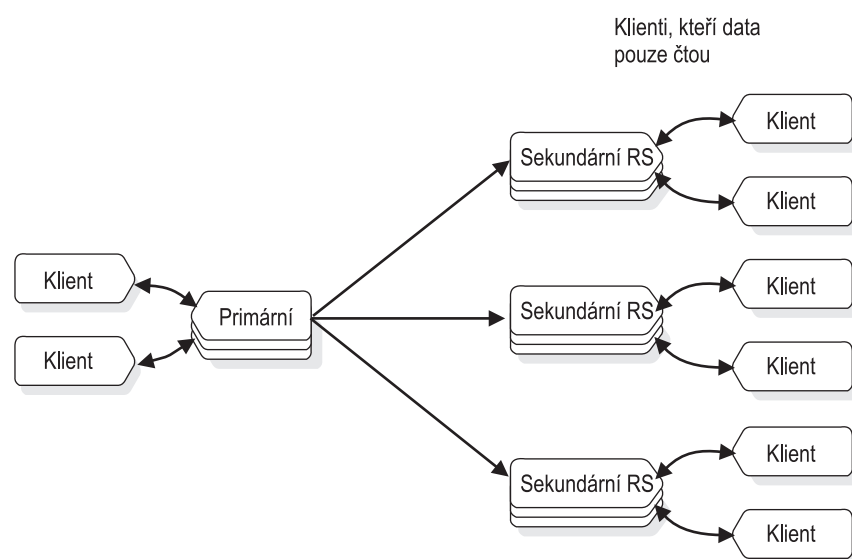
- ID souboru logického protokolu, který obsahuje poslední dokončený kontrolní bod.
- Pozici záznamu o kontrolním bodu v souboru logického protokolu.
- ID posledního odeslaného (nebo přijatého) souboru logického protokolu.
- Číslo stránky posledního odeslaného (nebo přijatého) záznamu logického protokolu.

Příklady konfigurace replikace dat

Tato část popisuje některé příklady, jak lze nakonfigurovat prostředí replikace dat.

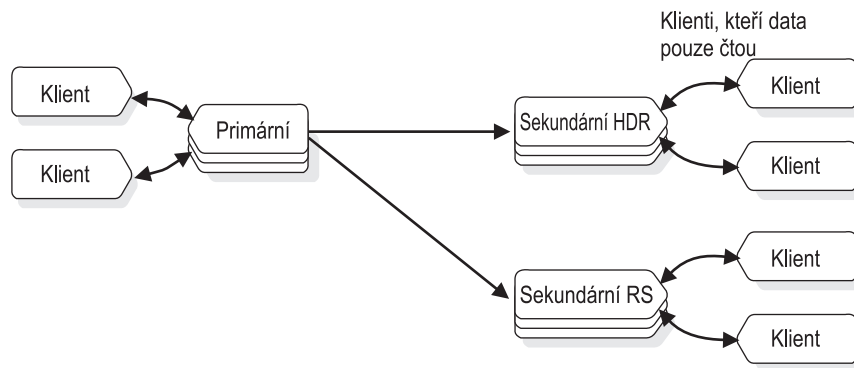
Příklady konfigurace vzdálených samostatných serverů

Na následujícím obrázku je uveden příklad konfigurace, která se skládá s několika sekundárních serverů RS. Tato konfigurace může být vhodná v situaci, ve které je primární server umístěn ve velké vzdálenosti od sekundárního serveru RS, nebo pokud je síť mezi primárním serverem a sekundárním serverem RS pomalá nebo chybová. Vzhledem k tomu, že sekundární servery RS používají plně duplexní komunikační protokoly a nepožadují zpracování kontrolních bodů v režimu SYNC, neměly by mít další servery významný vliv na výkon primárního serveru.



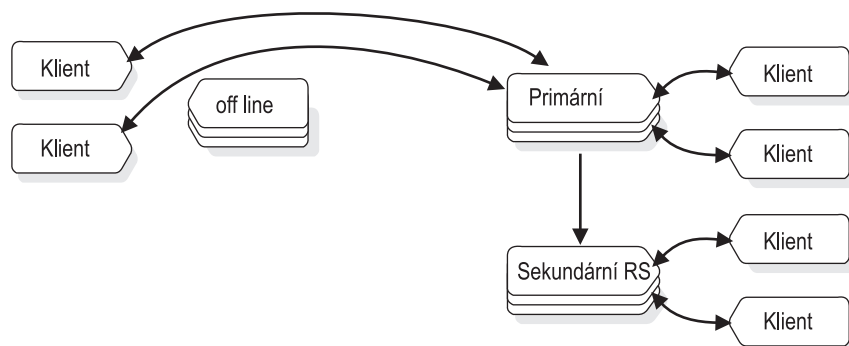
Obrázek 19-8. Primární server se třemi sekundárními servery RS.

Následující obrázek zobrazuje příklad konfigurace sekundárního serveru RS se sekundárním serverem replikace HDR. V tomto příkladu poskytuje sekundární replikace HDR vysokou dostupnost a sekundární server RS poskytuje další možnost obnovení při havárii v případě, že dojde k současnému výpadku primárního serveru a sekundárního serveru replikace HDR. Sekundární server RS může být geograficky vzdálený od primárního serveru a sekundárního serveru replikace HDR, takže by na něj nemělo mít vliv místní poškození, například zemětřesení nebo povodně.



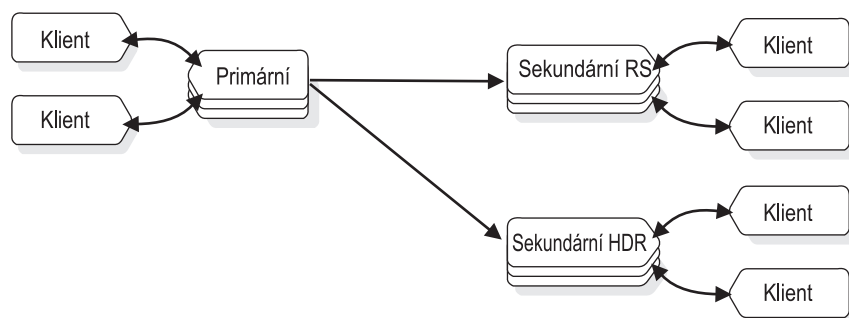
Obrázek 19-9. Primární server se sekundárními servery replikace HDR a RS.

Pokud primární databázový server selže, budete moci převést existující sekundární server replikace HDR na primární server, jako v následujícím diagramu.



Obrázek 19-10. Zabezpečení pro případ poruchy primárního serveru na sekundární server replikace HDR.

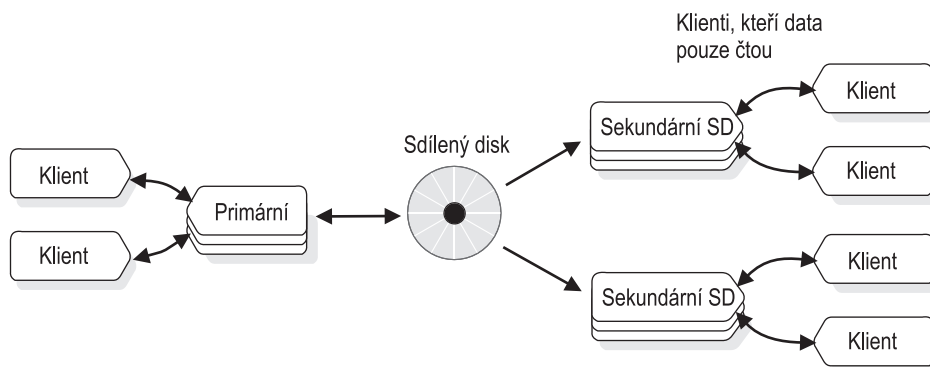
Pokud je možné, že bude primární server delší dobu v režimu offline, je možné sekundární server RS převést na sekundární server replikace HDR. Pokud se původní primární server vrátí zpět do režimu online, můžete jej převést ma sekundární server RS, jak je uvedeno na následujícím obrázku.



Obrázek 19-11. Sekundární server RS, který přebírá roli sekundárního serveru replikace HDR.

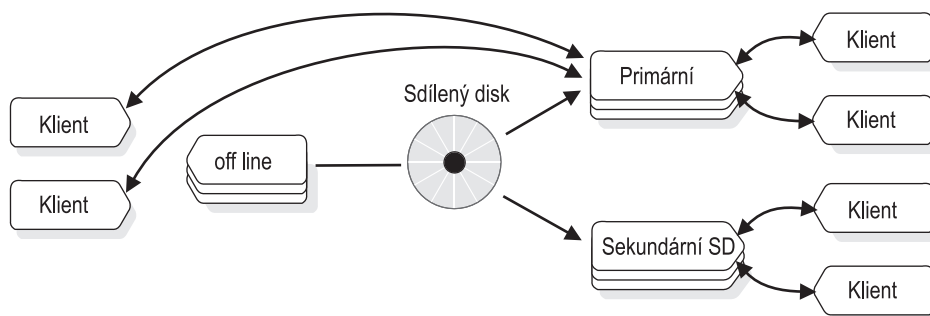
Příklady konfigurace sekundárních sdílených disků

Na následujícím obrázku je zobrazen příklad primárního serveru se dvěma sekundárními servery. V tomto případě lze přenést roli primárního serveru na jeden ze dvou sekundárních serverů SD. To je možné v případě, že je třeba primární server odebrat z důvodu plánovaného vypnutí nebo z důvodu selhání primárního serveru.



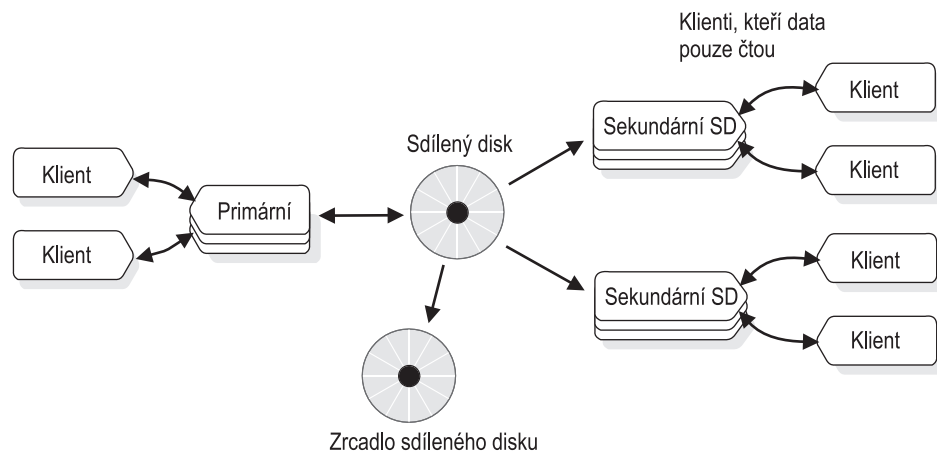
Obrázek 19-12. Primární server nakonfigurovaný se dvěma sekundárními servery SD.

Protože oba sekundární servery SD čtou ze stejného diskového podsystému, jsou oba stejně schopné převzít roli primárního serveru. Na následujícím obrázku je uvedena situace, kdy je primární server v režimu offline.



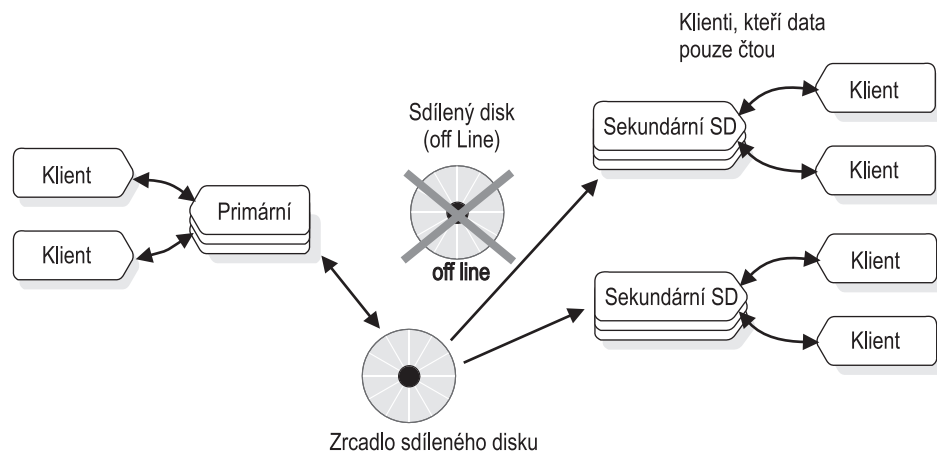
Obrázek 19-13. Sekundární server SD, který přebírá roli primárního serveru.

Existuje několik způsobů ochrany proti hardwarovému selhání sdíleného disku. Nejčastěji používaným způsobem je zřejmě konfigurace diskového pole založeného na technologii RAID (například RAID-5). Dalším způsobem ochrany proti selhání disku je použití technologie SAN (Storage Area Technology), která umožňuje některé druhy vzdáleného zrcadlení disku. Protože je možné disky SAN umístit do velké vzdálenosti od primárního disku a jeho zrcadel, poskytuje tato technologie vysoký stupeň dostupnosti pro plánované i neplánované vypnutí serveru nebo diskového podsystému. Následující obrázek popisuje takovou konfiguraci:



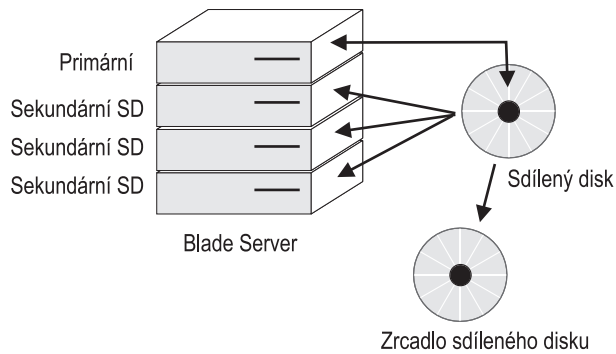
Obrázek 19-14. Primární server a sekundární servery SD se zrcadlenými disky.

V případě selhání disku lze servery překonfigurovat podle následujícího obrázku:



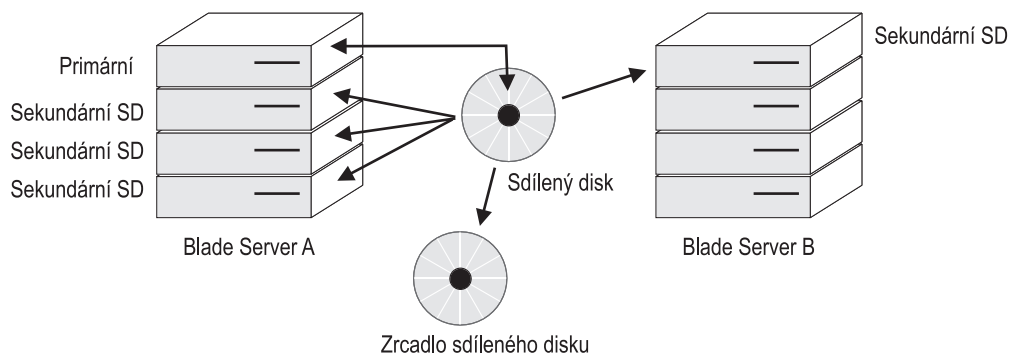
Obrázek 19-15. Zrcadlo sdíleného disku po selhání primárního sdíleného disku.

Kromě konfigurace podsystému zrcadleného disku jako na předchozím obrázku můžete používat další redundantní servery. Primární server a dva sekundární servery SD z předchozího obrázku můžete mít například umístěny v rámci jednoho zařízení kartového serveru). Protože je tím skupina serverů umístěna v rámci jednoho kartového serveru, může být toto zařízení potenciálním místem selhání. Konfigurace na následujícím obrázku je zajímavým řešením, pokud pravidelně potřebujete zvýšit výkon čtení procesoru, například při provádění velkých informačních úloh.



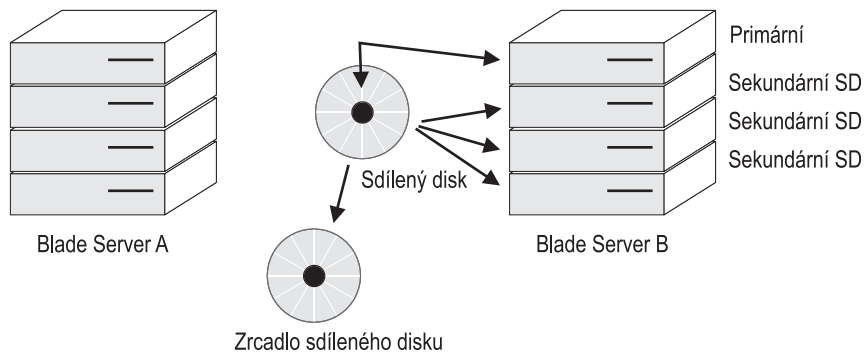
Obrázek 19-16. Primární server a sekundární servery SD v kartovém serveru

Samostatný kartový server představuje možné místo selhání, protože jsou všechny servery umístěny v rámci jednoho zařízení. V této konfiguraci můžete chtít předejít možnému místu selhání použitím více kartových serverů, jako na následujícím obrázku.



Obrázek 19-17. Konfigurace více kartových serverů jako ochrana před selháním v jednom místě

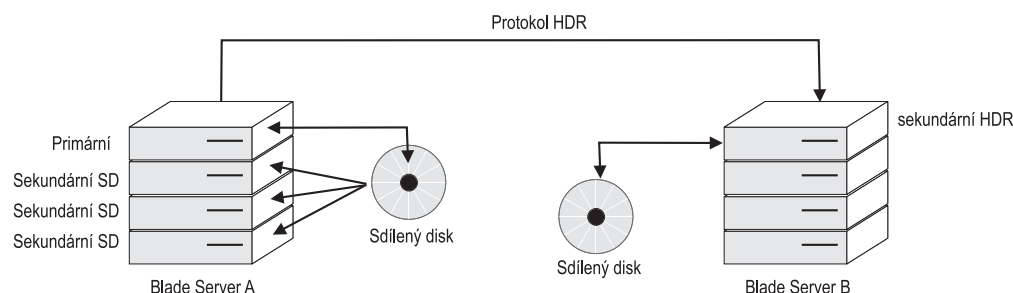
Pokud serverová karta A z předchozího obrázku selže, bude možné přenést roli primárního serveru na sekundární server SD na serverové kartě B. Protože je možné velmi rychle přivést sekundární servery SD do režimu online, bude možné dynamicky přidat další sekundární servery SD do kartového serveru B jako na následujícím obrázku.



Obrázek 19-18. Zabezpečení pro případ poruchy kartového serveru

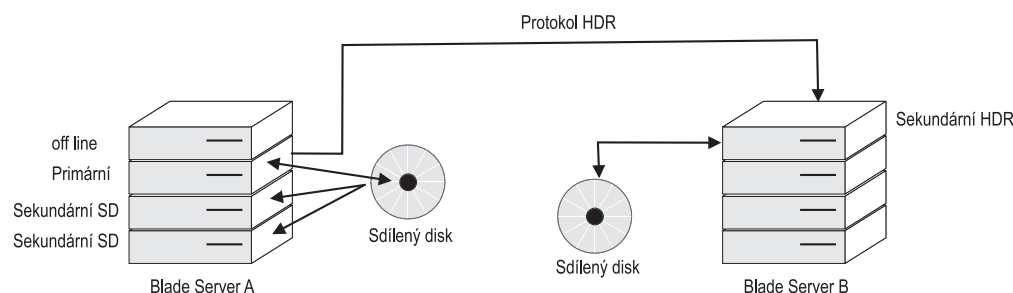
Z důvodu omezení vzdálenosti mezi primárními a zrcadlenými disky, které podporují zrcadlení, můžete uvažovat o použití sdílených disků a zajistit dostupnost disků pomocí zrcadlení sdílených disků. Můžete například dát přednost tomu, aby mezi dvěma kopiemi

diskového podsystému byla významná vzdálenost. V tom případě můžete pro sekundární kopii diskového podsystému zvolit sekundární server replikace HDR nebo sekundární server RS. Pokud je síťové připojení dostatečně rychlé (to znamená že je odezva příkazu ping od sekundárního serveru nižší než 50 ms), můžete zvážit použití sekundárního serveru replikace HDR. U pomalejších síťových připojení zvažte použití sekundárního serveru RS. Následující obrázek uvádí příklad sekundárního serveru replikace HDR v konfiguraci kartového serveru.



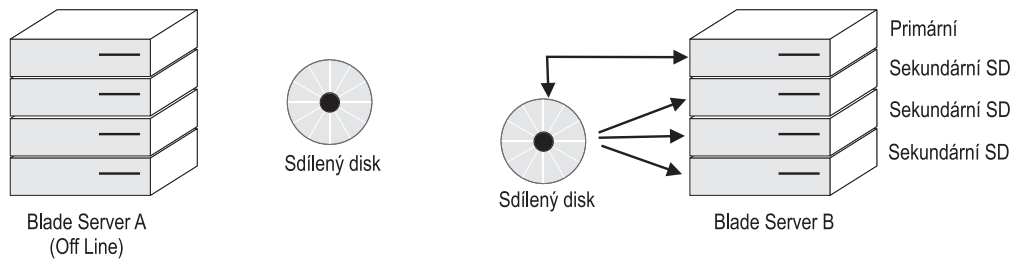
Obrázek 19-19. Sekundární server HDR v konfiguraci kartového serveru

Pokud v konfiguraci uvedené na předchozím obrázku dojde k selhání primárního uzlu, ale sdílené disky budou neporušené a kartový server bude stále funkční, bude možné převést roli primárního serveru z prvního serveru na kartovém serveru A na jiný server na stejném kartovém serveru. Při změně primárního serveru dojde k automatickému přesměrování ze vzdáleného sekundárního serveru replikace HDR na nový primární server, viz následující diagram:



Obrázek 19-20. Zabezpečení pro případ poruchy primárního serveru na sekundární server SD

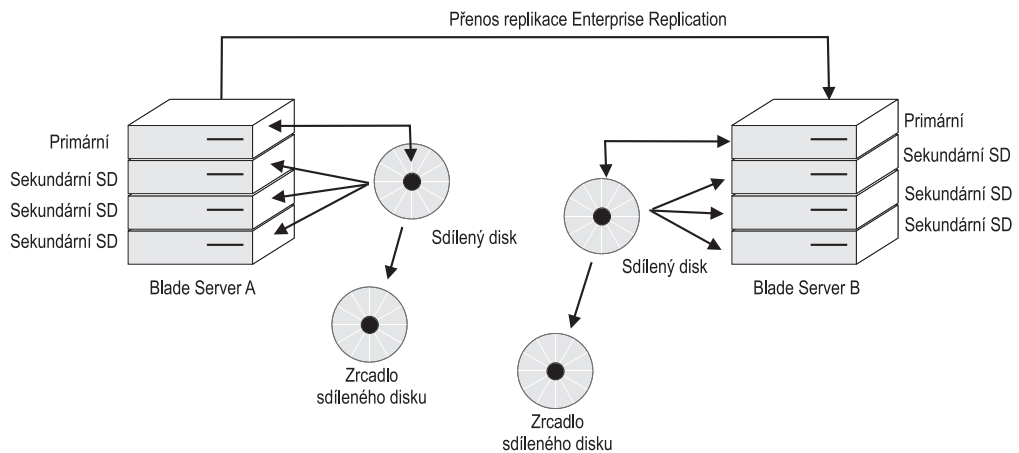
Předpokládejme, že při selhání popsaném na předchozím obrázku nedošlo k selhání pouze jedné serverové karty v kartovém serveru, ale celého kartového serveru. V takovém případě můžete přejít na sekundární server replikace HDR. Protože je spuštění sekundárního serveru SD velmi rychlé, můžete snadno přidat další sekundární servery SD. Sekundární server SD může pracovat pouze na primárním uzlu. Jakmile byl primární server přenesen na kartový server B, je také možné na tomto serveru spustit sekundární servery SD, viz následující obrázek.



Obrázek 19-21. Selhání celého kartového serveru

Použití replikace Enterprise Replication jako části obnovitelné skupiny

Protože replikace Enterprise Replication nepodporuje synchronní režim provozu (SYNC), neumožňuje podporu prostředí s několika aktivními servery. Během události selhání je replikace Enterprise Replication schopna sloučit rozdíly mezi dvěma servery. Replikaci Enterprise Replication můžete použít jako prostředek zlepšení synchronizace mezi servery, protože každý systém replikace Enterprise Replication uchovává nezávislý protokolovací systém. Konfigurace, která používá replikaci Enterprise Replication, je zobrazena na následujícím obrázku:



Obrázek 19-22. Konfigurace replikace Enterprise Replication jako části obnovitelné skupiny

Příklad konfigurace vysoce dostupných klastrů s replikací Enterprise Replication

Předpokládejme, že potřebujete replikaci Enterprise Replication mezi dvěma vysoce dostupnými serverovými klastry:

Klastr 1:

- Primární server
- Sekundární server HDR
- Sekundární server SD 1
- Sekundární server SD 2
- Sekundární server RS 1
- Sekundární server RS 2

Klastr 2:

- Primární server
- Sekundární server HDR
- Sekundární server SD 1
- Sekundární server SD 2
- Sekundární server RS 1
- Sekundární server RS 2

Dále předpokládejme, že každý server je pojmenován podle následující konvence:

- První tři znaky: název podniku
- Znak 4: krátké číslo hostitele
- Znaky 5, 6 a 7: název klastru
- Znaky 8, 9 a 10: typ serveru: "pri" v případě primárního serveru, "sec" v případě sekundárního serveru
- Znaky 11, 12 a 13: typ připojení: "shm" nebo "tcp"

Například server s názvem: **srv4_1_pri_shm** je popsán následovně:

- srv = název podniku
- 4 = krátké číslo hostitele
- _1_ = název klastru
- pri = jedná se o primární server
- shm = typ připojení využívá komunikaci prostřednictvím sdílené paměti

Výše uvedenou konfiguraci by podporovaly následující položky souboru **sqlhosts**:

```
srv4_1_pri_shm onipcshm sun-mach4 srv4_1_pri_shm
srv4_1_sec_shm onipcshm sun-mach4 srv4_1_sec_shm
srv5_1_rss_shm onipcshm sun-mach5 srv5_1_rss_shm
srv5_1_sds_shm onipcshm sun-mach5 srv5_1_sds_shm
srv6_1_rss_shm onipcshm sun-mach6 srv6_1_rss_shm
srv6_1_sds_shm onipcshm sun-mach6 srv6_1_sds_shm
srv_1_cluster group - - i=1
srv4_1_pri_tcp onlitcp sun-mach4 21316 g=srv_1_cluster
srv4_1_sec_tcp onlitcp sun-mach4 21317 g=srv_1_cluster
srv5_1_rss_tcp onlitcp sun-mach5 21316 g=srv_1_cluster
srv5_1_sds_tcp onlitcp sun-mach5 21317 g=srv_1_cluster
srv6_1_rss_tcp onlitcp sun-mach6 21316 g=srv_1_cluster
srv6_1_sds_tcp onlitcp sun-mach6 21317 g=srv_1_cluster

srv4_2_pri_shm onipcshm sun-mach4 srv4_2_pri_shm
srv4_2_sec_shm onipcshm sun-mach4 srv4_2_sec_shm
srv5_2_rss_shm onipcshm sun-mach5 srv5_2_rss_shm
srv5_2_sds_shm onipcshm sun-mach5 srv5_2_sds_shm
srv6_2_rss_shm onipcshm sun-mach6 srv6_2_rss_shm
srv6_2_sds_shm onipcshm sun-mach6 srv6_2_sds_shm
srv_2_cluster group - - i=2
srv4_2_pri_tcp onlitcp sun-mach4 21318 g=srv_2_cluster
srv4_2_sec_tcp onlitcp sun-mach4 21319 g=srv_2_cluster
srv5_2_rss_tcp onlitcp sun-mach5 21318 g=srv_2_cluster
srv5_2_sds_tcp onlitcp sun-mach5 21319 g=srv_2_cluster
srv6_2_rss_tcp onlitcp sun-mach6 21318 g=srv_2_cluster
srv6_2_sds_tcp onlitcp sun-mach6 21319 g=srv_2_cluster
```

Příklad komplexní strategie obnovy pro případ poruchy

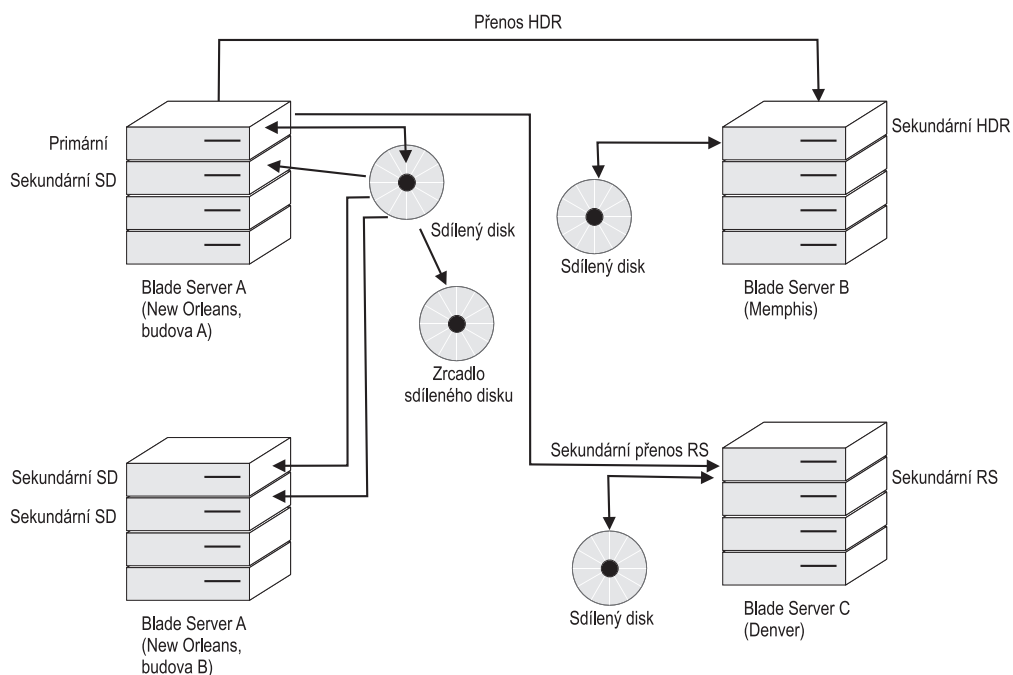
Tato část popisuje tříúrovňový server, který umožňuje dosáhnout maximální dostupnost v případě pohromy v rozsáhlejší oblasti.

Sekundární server replikace HDR obvykle poskytuje zálohu sekundárních serverů SD a poskytuje podporu vysoce dostupných systémů, které jsou geograficky vzdálené od hlavního systému. Sekundární servery RS poskytují další dostupnost sekundárnímu serveru replikace HDR a lze je použít jako řešení dostupnosti v případě pohrom. Pokud musíte z důvodu dostupnosti používat sekundární server RS, bude třeba ostatní systémy pro návrat k normálnímu provozu znovu ručně vytvořit provedením zálohy a obnovy. Další informace o tomto tématu poskytuje scénář pohromy, která zasahuje velkou oblast, například hurikán.

Chcete-li zajistit maximální dostupnost pro případ oblastní pohromy, je třeba použít dostupnost založenou na *vrstvách*. První vrstva poskytuje řešení dostupnosti pro případ přechodných místních selhání. Může zahrnovat například dvojici kartových serverů připojených k jednomu diskovému podsystému se spuštěnými sekundárními servery SD. Umístěním sekundárních serverů SD do několika umístění v rámci areálu lze dosáhnout souvislého provozu i v případě místního selhání.

Dále je možné přidat druhou vrstvu, která umožňuje zvýšit dostupnost díky alternativnímu umístění s vlastní kopii disků. Chcete-li se chránit proti větší oblastní pohromě, můžete zvážit konfiguraci sekundárního serveru replikace HDR umístěného ve větší vzdálenosti, například stovek kilometrů. Vzdálený systém můžete také vytvořit jako kartový server nebo jiný víceserverový systém. Pokud je k dispozici druhá vrstva, dojde k poruše a vzdálený sekundární server replikace HDR se stane primárním serverem, bude možné ve vzdáleném umístění snadno spustit sekundární servery SD.

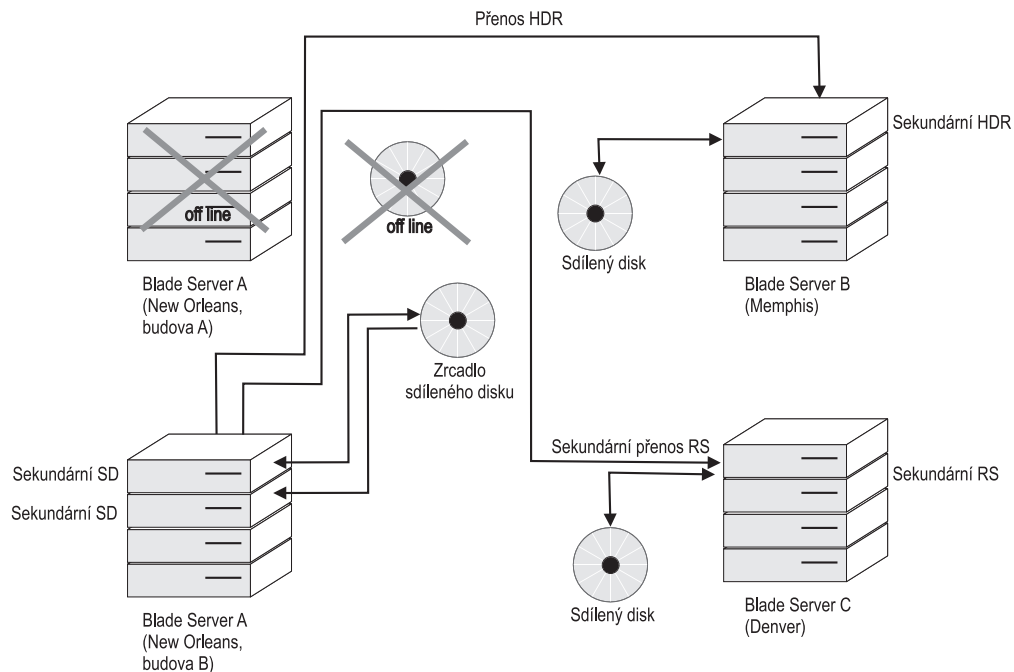
Může však postačovat i dvouúrovňové řešení. Hurikán v jedné oblasti může vyvolat stovky tornád vzdálených stovek kilometrů. Před nimi je možné se chránit přidáním třetí ochranné úrovně, například sekundárnímu serveru RS umístěného ve vzdálenosti tisíce kilometrů. Toto tříúrovňové řešení poskytuje další redundanci, která významně snižuje riziko výpadku.



Obrázek 19-23. Konfigurace dostupnosti tříúrovňového serveru

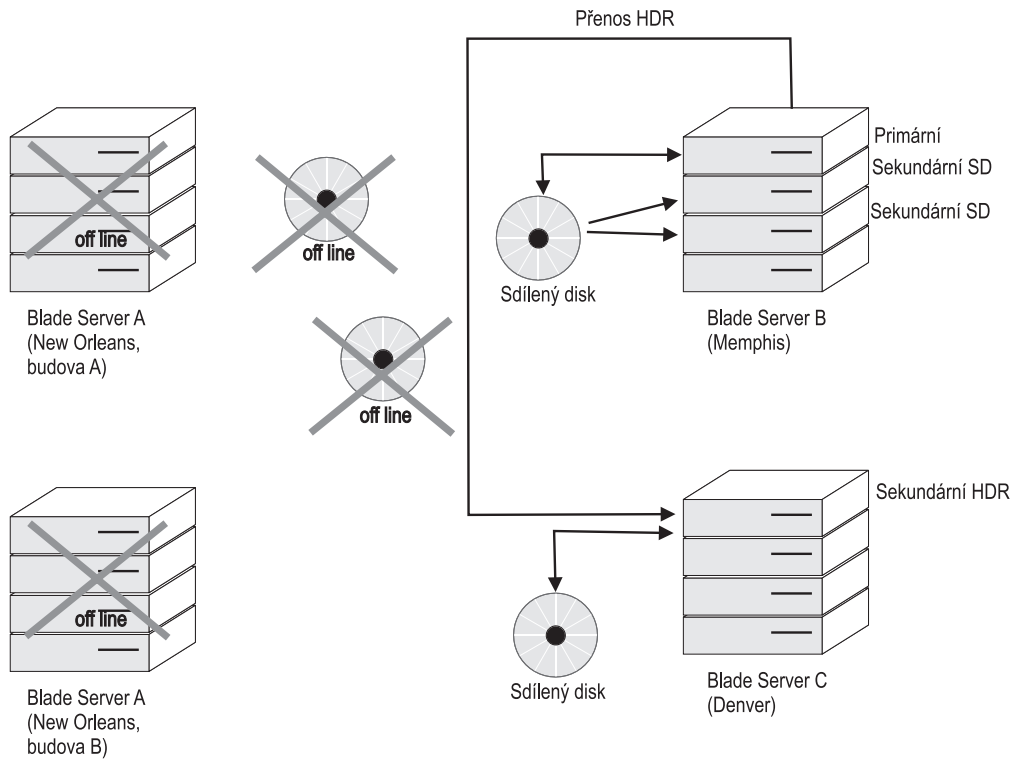
Nyní předpokládejme, že došlo k místnímu výpadku v budově A v areálu v Novém Orleansu. Řekněme, že došlo k popraskání potrubí v serverové místnosti a kartový server a primární kopie subsystému sdíleného disku byla poškozena vodou. Nyní můžete spuštěním

příkazu `onmode -d make primary` název-serveru na libovolném sekundárním serveru SD spuštěném na kartovém serveru v budově B přepnout roli primárního serveru na budovu B. To způsobí, že se budou všechny další sekundární uzly automaticky připojovat k novému primárnímu uzlu.



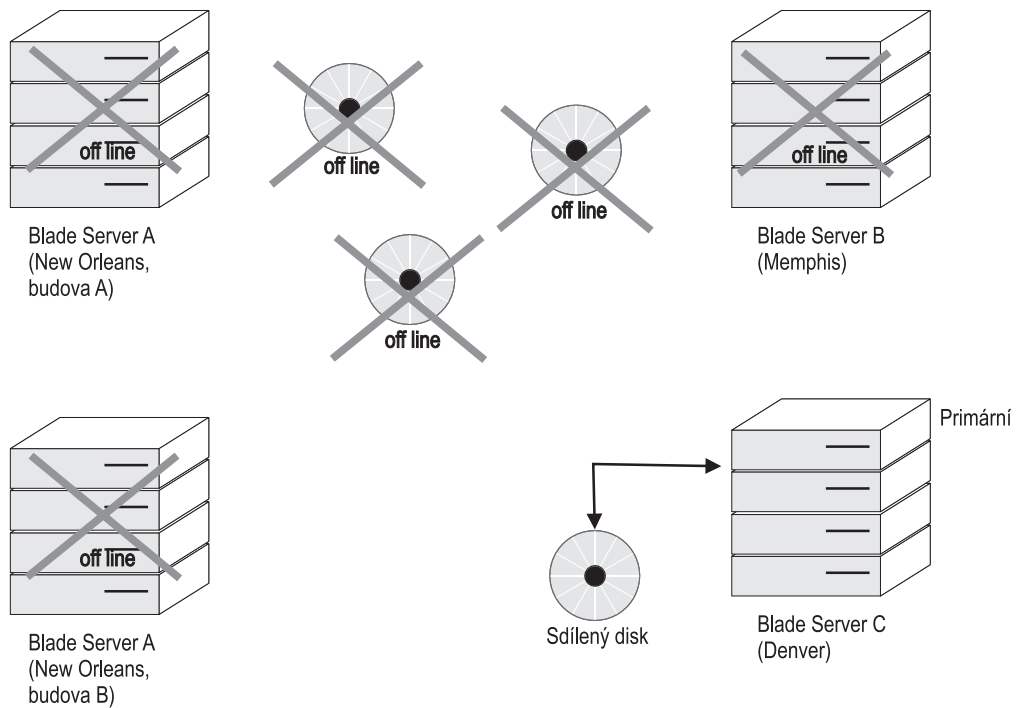
Obrázek 19-24. První úroveň ochrany

Také může dojít k oblastnímu výpadku v Novém Orleansu, například ke ztrátě obou budov A i B. Kromě toho můžete umístit v Denveru sekundární replikaci HDR a další sekundární servery SD v počítači v Memphisu.



Obrázek 19-25. Druhá úroveň ochrany

Při ještě větším výpadku, který má vliv na obě místa, může být třeba přepnout na nejvzdálenější systém.



Obrázek 19-26. Třetí úroveň ochrany

Tabulka 19-4. Doporučené konfigurace při různých požadavcích

Požadavky	Doporučené konfigurace
Pravidelně je třeba zvýšit přenosovou kapacitu	Použijte sekundární servery SD
Používáte zařízení SAN, která poskytují dodatečnou dostupnost diskového hardwaru, ale chcete se chránit před selháním serverů	Použijte sekundární servery SD
Používáte zařízení SAN, která poskytují dodatečné zrcadlení hardwaru disků, ale chcete mít druhou sadu serverů, které je možné uvést do režimu online při případné ztrátě hlavního provozu (a v omezení zrcadlených disků není problém)	Zvažte použití dvou kartových center v různých umístěních, na kterých jsou spuštěny sekundární servery SD
Chcete mít záložní umístění ve střední vzdálenosti, ale nechcete připustit žádnou ztrátu dat při výpadku	Zvažte použití dvou kartových center se sekundárními servery SD v hlavním kartovém centru a sekundární replikací HDR ve vzdáleném.
Chcete vysoce dostupný systém, ve kterém nikdy nedojde ke ztrátě transakce, a musíte mít také vzdálený systém v jiné části světa	Zvažte použití sekundární replikace HDR v režimu SYNC umístěné nedaleko a sekundárního serveru RS v jiné části světa
Chcete vysoce dostupné řešení, ale v oblasti jsou k dispozici pouze sítě s nejlepší odezvou ping kolem 200 ms	Zvažte použití sekundárního serveru RS
Chcete použít záložní umístění, ale nemáte s tímto umístěním přímé spojení	Můžete použít obnovení nepřetržitého protokolu se zálohováním a obnovou
Můžete připustit zpoždění v doručení dat, pokud data nakonec dorazí, ale potřebujete rychlé zabezpečení pro libovolný případ poruchy	Zvažte použití sekundárních serverů SD s hardwarovým zrcadlením disku společně s replikací ER.
Potřebujete přídavný výkon pro zápis, můžete připustit určité zpoždění v doručení zapisovaných dat, potřebujete vysokou dostupnost a můžete rozdělovat pracovní zátěž	Zvažte použití replikace ER se sekundárními servery SD

Přesměrování a propojitelnost pro klienty replikace dat

Klienti využívají k připojení k databázovým serverům v replikačním páru tytéž metody, kterými se připojují k standardním databázovým serverům. Podrobný popis těchto metod naleznete v popisech příkazů CONNECT a DATABASE v příručce *IBM Informix Guide to SQL: Syntax*.

Po selhání jednoho z databázových serverů v páru můžete chtít *přesměrovat* klienty využívající databázový server, který selhal. (Těž může nastat situace, kdy nebudete chtít přesměrovat klienty. Pokud například předpokládáte, že databázové servery budou brzy opět v provozu, nemusí být přesměrování klientů vhodné.)

Chcete-li automaticky přesměrovat klienty na jiné databázové servery v páru replikace, nakonfigurujte aplikace, aby se připojovaly ke skupině serverů, ke které patří oba servery replikace HDR. Pokud je vytvořeno připojení ke skupině serverů a je v provozu replikace HDR, vytvoří se ve výchozím nastavení připojení k aktuálnímu primárnímu serveru ve skupině. Pokud je replikace vypnuta z důvodu selhání jednoho ze serverů, bude vytvořeno připojení k serveru v režimu online (ve standardním režimu nebo v primárním režimu bez sekundárního serveru). Tuto akci můžete také automatizovat z prostředí aplikace, jak popisuje část "Přesměrování z aplikace" na stránce 19-29. Některé z ovladačů propojitelnosti klientů obsažené v sadě IBM Informix Client Software Developer's Kit jsou vybaveny specifickými mechanismy, které přesměrování automatizují. Podrobnosti naleznete v dokumentaci sady IBM Informix Client Software Developer's Kit.

Návrh klientů pro přesměrování

Když navrhujete klientské aplikace, musíte učinit několik rozhodnutí týkajících se strategie přesměrování. Musíte rozhodnout, zda budete přesměrování ovládat z aplikace a který mechanismus přesměrování budete využívat. Existují tři různé mechanismy přesměrování:

- Automatické přesměrování pomocí proměnné prostředí **DBPATH**.
- Administrátorem řízené přesměrování pomocí údajů o propojitelnosti.
- Uživatelem řízené přesměrování pomocí proměnné prostředí **INFORMIXSERVER**.

Mechanismus, který použijete, určuje syntaxi příkazu **CONNECT**, kterou můžete použít v aplikaci. Následující části popisují všechny tři mechanismy přesměrování.

Automatické přesměrování klientů pomocí proměnné prostředí **DBPATH**

Tato část popisuje kroky, které je nutno učinit, chcete-li přesměrovat klienty pomocí proměnné prostředí **DBPATH** a také strategií propojitelnosti, která podporuje tuto metodu.

Jak pracuje metoda přesměrování pomocí proměnné prostředí **DBPATH**

Pokud aplikace neurčí databázový server v rámci příkazu **CONNECT** explicitně a databázový server určený proměnnou prostředí **INFORMIXSERVER** není dostupný, použije klient k nalezení databáze (a databázového serveru) proměnnou prostředí **DBPATH**.

Pokud je jeden z databázových serverů v replikačním páru nepoužitelný, aplikace, které používají tento databázový server, nemusejí znovu nastavit proměnnou prostředí **INFORMIXSERVER**, je-li jejich proměnná prostředí **DBPATH** nastavena na druhý databázový server z tohoto páru. Proměnná prostředí **INFORMIXSERVER** by měla vždy obsahovat název databázového serveru, který obvykle používají a proměnná prostředí **DBPATH** by měla vždy obsahovat název alternativního databázového serveru v replikačním páru.

Pokud aplikace za normálních okolností používají databázový server s názvem **cliff_ol** a databázový server v páru se serverem **cliff_ol** do replikačního páru se nazývá **beach_ol**, budou proměnné prostředí pro tyto aplikace vypadat následovně:

```
INFORMIXSERVER cliff_ol
DBPATH          //beach_ol
```

Protože proměnnou prostředí **DBPATH** lze pouze číst (je-li to potřeba), když aplikace vydá příkaz **CONNECT**, dojde k přesměrování až po restartu aplikace.

Aplikace může obsahovat kód, který prověří případné selhání připojení a zjistí-li jej, zopakuje pokus o připojení. Obsahuje-li aplikace tento kód, není třeba ji restartovat.

U této metody přesměrování můžete použít příkaz **CONNECT TO** *databáze*. Aby tato metoda pracovala, nepoužívejte následující příkazy:

- **CONNECT TO DEFAULT**
- **CONNECT TO** *databáze@databázovýserver*
- **CONNECT TO** *@databázovýserver*

Důvodem tohoto omezení je skutečnost, že aplikace nepoužije proměnnou prostředí **DBPATH**, pokud příkaz **CONNECT** určuje databázový server. Další informace o proměnné prostředí **DBPATH** naleznete v příručce *IBM Informix Guide to SQL: Reference*.

Co by měli provést administrátoři

Administrátoři nezasahují při přesměrování klientů, ale v závislosti na typu databázového serveru může být jejich účast nutná.

Co by měli provést uživatelé

Obsahují-li aplikace kód, který prověří selhání spojení a v případě potřeby vydá příkaz k opakovanému připojení, bude přesměrování provedeno automaticky. Uživatel nemusí učinit žádné kroky.

Pokud aplikace takový kód neobsahuje, bude nutné, aby uživatelé klientů ukončili a restartovali všechny aplikace.

Směrování klientů pomocí informací o propojitelnosti

Tato část popisuje kroky při přesměrování klientů pomocí informací o propojitelnosti a strategii propojitelnost, která podporuje tuto metodu.

Operační systém

UNIX

Umístění informací o propojitelnosti

Proměnná prostředí **INFORMIXSQLHOSTS** určuje úplný název cesty k souboru **\$INFORMIXDIR/etc/sqlhosts** obsahujícímu údaje o připojení. Další informace o proměnné prostředí **INFORMIXSQLHOSTS** naleznete v příručce *IBM Informix Guide to SQL: Reference*.

Windows

Informace o propojitelnosti jsou uloženy v klíči registru systému Windows nazvaném **HKEY_LOCAL_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS**.

Jak pracuje přesměrování pomocí informací o propojitelnosti

Metoda přesměrování pomocí informací o propojitelnosti vychází ze skutečnosti, že aplikace, která se připojuje k databázovému serveru, vyhledává databázový server právě prostřednictvím informací o propojitelnosti.

Pokud je jeden z databázových serverů v replikačním páru nepoužitelný, může administrátor změnit definici nedostupného databázového serveru v informacích o propojitelnosti. Jak popisuje část “Změna informací o propojitelnosti” na stránce 19-26, pole nedostupného databázového serveru (kromě pole **dbservername**) se změní a odkazují se na zbývající databázový server v páru replikace.

Protože jsou informace o propojitelnosti načítány po vydání příkazu **CONNECT**, bude pravděpodobně nutné restartovat aplikace, aby mohlo dojít k přesměrování. Aplikace mohou obsahovat kód, který prověří případné selhání připojení a zjistí-li jej, zopakuje pokus o připojení. Pokud došlo k selhání připojení, je přesměrování automatické a není kvůli němu nutné restartovat aplikace.

Aplikace mohou použít následující připojovací příkazy podporující tuto metodu přesměrování:

- **CONNECT TO** *databáze@databázovýserver*
- **CONNECT TO** *@databázovýserver*

Aplikace mohou též za předpokladu, že proměnná prostředí **INFORMIXSERVER** obsahuje vždy stejný název databázového serveru a proměnná prostředí **DBPATH** není nastavena, použít následující připojovací příkazy:

- **CONNECT TO** **DEFAULT**

- CONNECT TO *databáze*

Změna informací o propojitelnosti

Chcete-li používat informace o propojitelnosti k přesměrování klientů, musíte změnit informace o propojitelnosti klientů a jiné soubory obsahující údaje o připojení, je-li to nutné.

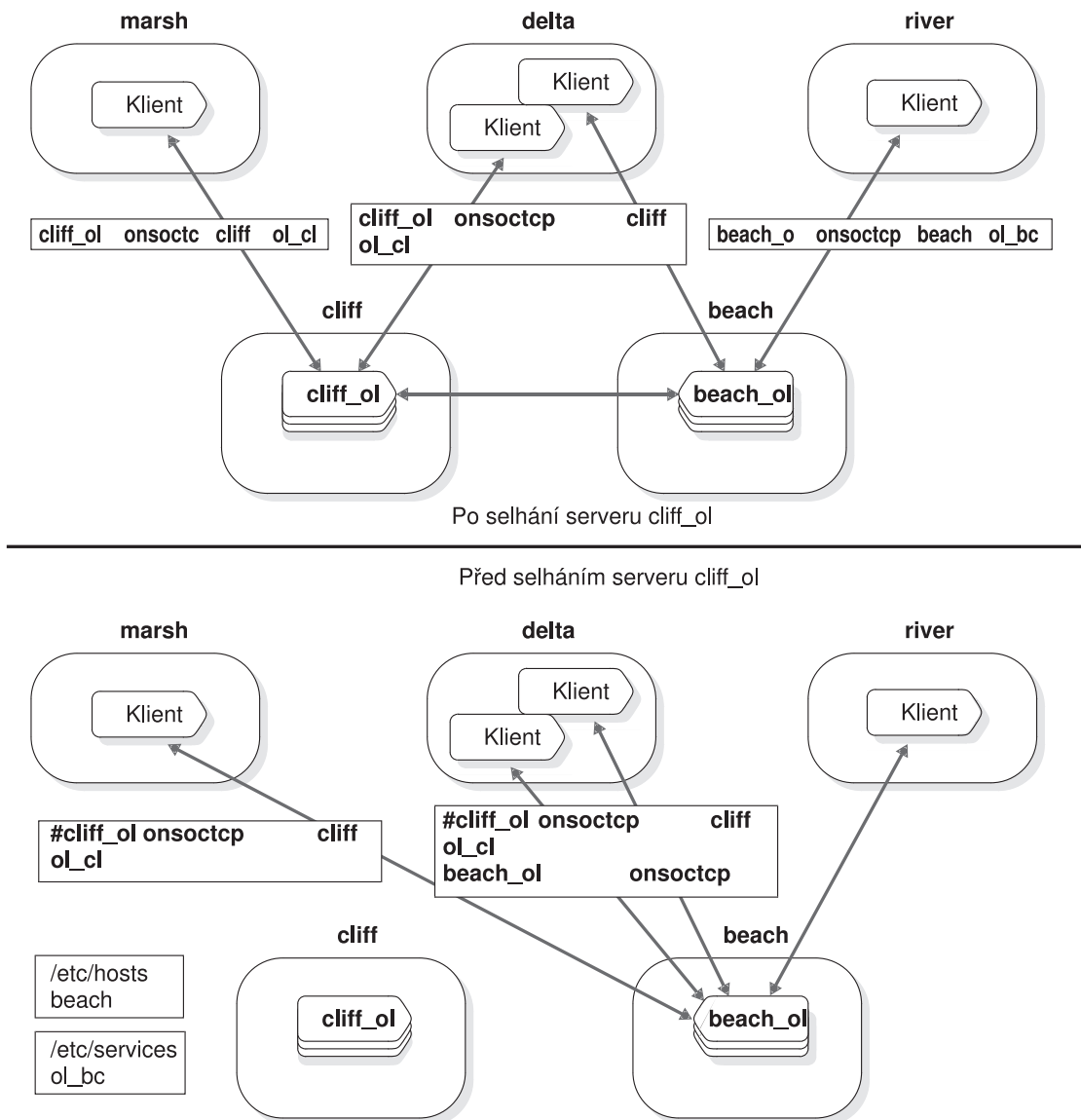
Další informace uvádí část “Konfigurace propojitelnosti replikace HDR” na stránce 20-6 a Kapitola 3, “Komunikace mezi klientem a serverem”, na stránce 3-1.

Změna informací o propojitelnost na klientském počítači:

1. Uzavřete záznam databázového serveru s poruchou do poznámky.
2. Vložte záznam, kterým v poli **servername** určíte název dbservername databázového serveru, který selhal, a v polích **nettype**, **hostname**, a **servicename** určíte údaje databázového serveru, k němuž přesměrujete klienty.
3. Pokud dojde k selhání, použijte v souboru nebo registru **sqlhosts** k přesměrování aplikací na jiný databázový server následující volby:
 - a. “Volba přesměrování připojení” na stránce 3-22
 - b. “Volba konce skupiny” na stránce 3-23
 - c. “Volba skupiny” na stránce 3-23

Obrázek 19-27 na stránce 19-27 znázorňuje, jak lze upravit hodnoty propojitelnosti, aby došlo k přesměrování klientů.

Není nutné měnit záznamy v informacích o propojitelnosti v kterémkoli z počítačů, na kterých jsou spuštěné databázové servery.



Obrázek 19-27. Hodnoty propojitelnosti před selháním a po selhání databázového serveru cliff_ol

Změna dalších propojovacích souborů:

Než se klient znovu připojí k jinému databázovému serveru, musíte zajistit, aby byly splněny také následující podmínky.

1. Soubor `/etc/hosts` v systému UNIX nebo soubor `hosts` v systému Windows obsahuje položku `hostname` počítače se spuštěným databázovým serverem, na který jsou přesměrování klienti.
2. Soubor `/etc/services` v systému UNIX nebo soubor `services` v systému Windows obsahuje položku `servicename` databázového serveru, na který jsou přesměrování klienti.

Připojení k databázovému serveru

Jakmile administrátor změní informace o propojitelnosti a další připojovací soubory (je-li to nutné), připojí se klienti po odeslání následujícího příkazu `CONNECT` k databázovému serveru, k němuž je administrátor přesměroval.

Obsahují-li aplikace kód, který prověří selhání spojení a v případě potřeby vydá příkaz k opakovanému připojení, bude přesměrování provedeno automaticky. Uživatel nemusí učinit žádné kroky. Pokud aplikace takový kód neobsahují, bude nutné, aby uživatelé klientů ukončili a restartovali všechny aplikace.

Automatické přesměrování ve skupině serverů

Možnost skupiny v souboru SQLHOSTS můžete použít k zadání skupiny serverů, ke které se připojují aplikace, místo zadání jednotlivého databázového serveru. Chcete-li provádět přesměrování připojení automaticky, přidejte do definice skupiny serverů definici databázového serveru pro primární i sekundární server. Ve výchozím nastavení se při uskutečnění připojení ke skupině serverů replikace HDR připojí také primární server, pokud je k dispozici. Pokud primární server není k dispozici, provede se připojení ke skupině serverů replikace HDR k sekundárnímu serveru.

Následující položky SQLHOSTS například reprezentují skupinu serverů replikace HDR **g_hdr** s definicí primárního serveru **hdr_prim** a definicí sekundárního serveru **hdr_sec**.

Tabulka 19-5. Položky SQLHOSTS pro skupinu serverů replikace HDR

dbservername	nettype	hostname	servicename	options
g_hdr	group	-	-	i=1
hdr_prim	ontlitcp	machine1pri	port1	g=g_hdr
hdr_sec	ontlitcp	machine1sec	port1	g=g_hdr

Aplikace mohou použít následující připojovací příkazy podporující tuto metodu přesměrování:

- CONNECT TO *database@dbserver_group*
- CONNECT TO *@dbserver_group*

Obsahují-li aplikace kód, který prověří selhání spojení a v případě potřeby vydá příkaz k opakovanému připojení, bude přesměrování provedeno automaticky. Uživatel nemusí učinit žádné kroky. Pokud aplikace takový kód neobsahují, bude nutné, aby uživatelé klientů ukončili a restartovali všechny aplikace.

Směrování klientů pomocí proměnné prostředí INFORMIXSERVER

Tato část popisuje kroky při přesměrování klientů pomocí proměnné prostředí **INFORMIXSERVER** a strategii propojitelnosti, která podporuje tuto metodu.

Jak pracuje přesměrování pomocí proměnné prostředí INFORMIXSERVER

Metoda přesměrování pomocí proměnné prostředí **INFORMIXSERVER** vychází ze skutečnosti, že pokud aplikace explicitně neurčí databázový server v příkazu **CONNECT**, připojí se klient k databázovému serveru, který je určen proměnnou prostředí **INFORMIXSERVER**.

Pokud je jeden z databázových serverů v replikačním páru nepoužitelný, mohou aplikace, které využívají tento databázový server, znovu nastavit vlastní proměnnou prostředí **INFORMIXSERVER** na druhý databázový server v páru, aby získaly přístup ke stejným datům.

Aplikace načítají hodnotu proměnné prostředí **INFORMIXSERVER** pouze při spuštění. Proto je nutné aplikaci restartovat, aby rozpoznala změnu v hodnotě proměnné prostředí.

Pro podporu této metody přesměrování můžete použít následující připojovací příkazy:

- CONNECT TO DEFAULT
- CONNECT TO *databáze*

U této metody nelze použít příkazy CONNECT TO *databáze@databázovýserver* nebo CONNECT TO *@databázovýserver*. Pokud je databázový server určen explicitně, příkaz CONNECT nepoužije k vyhledání databázového serveru proměnnou prostředí **INFORMIXSERVER**.

Co by měli provést administrátoři

Administrátoři nezasahují při přesměrování klientů, ale může být nutné, aby změnili typ databázového serveru.

Co by měli provést uživatelé

Uživatelé spouštějící klientské aplikace musejí provést následující tři kroky, rozhodnou-li se přesměrovat klienty pomocí proměnné prostředí **INFORMIXSERVER**.

Přesměrování klientů pomocí proměnné prostředí **INFORMIXSERVER**:

1. Aplikaci je nutné ukončit.
2. Pak musí uživatelé změnit proměnnou prostředí **INFORMIXSERVER** tak, aby obsahovala název druhého databázového serveru v replikačním páru.
3. Nakonec je nutné aplikaci restartovat.

Přesměrování z aplikace

Pokud k přesměrování používáte proměnnou prostředí **DBPATH** nebo informace o propojitelnosti, můžete do svých klientů integrovat rutinu, která ošetří chyby, setkají-li se klienti se selháním replikace HDR. Rutina může volat jinou funkci obsahující smyčku, která se bude opakovaně pokoušet připojit k druhému databázovému serveru v replikačním páru. Tato rutina přesměruje klienty, aniž by žádala uživatele o ukončení a restartování aplikace.

Obrázek 19-28 znázorňuje příklad funkce v klientské aplikaci využívající mechanismus přesměrování pomocí proměnné **DBPATH**, který se bude ve smyčce pokoušet o opakované připojení. Jakmile bude vytvořeno připojení, prověří aplikace rovněž typ databázového serveru, aby se ujistila, že se nejedná o sekundární databázový server. Pokud se jedná o databázový server sekundárního typu, zavolá další funkci, která upozorní uživatele (nebo administrátora databázového serveru), že databázový server nepřijímá aktualizace.

```

/* Rutina předpokládá, že proměnná prostředí INFORMIXSERVER
 * je nastavená na databázový server, který klient
 * obvykle používá, a že je proměnná prostředí DBPATH
 * nastavena na druhý databázový server v páru.
 */

#define SLEEPTIME 15
#define MAXTRIES 10

main()
{
    int connected = 0;
    int tries;
    for (tries = 0; tries < MAXTRIES && connected == 0; tries++)
    {
        EXEC SQL CONNECT TO "superstores";
        if (strcmp(SQLSTATE,"00000"))
        {
            if (sqlca.sqlwarn.sqlwarn6 != 'W')
            {
                notify_admin();
                if (tries < MAXTRIES - 1)
                    sleep(SLEEPTIME);
            }
            else connected =1;
        }
    }
    return ((tries == MAXTRIES)? -1:0);
}

```

Obrázek 19-28. Příklad smyčky příkazů CONNECT pro mechanismus přesměrování prostřednictvím proměnné prostředí DBPATH

Tento příklad předpokládá mechanismus přesměrování prostřednictvím proměnné prostředí **DBPATH** a využívá takovou formu příkazu CONNECT, která podporuje metodu přesměrování prostřednictvím proměnné prostředí **DBPATH**. Pokud jste k přesměrování použili informace o propojitelnosti, můžete používat odlišný připojovací příkaz, například: EXEC SQL CONNECT TO "superstores@cliff_ol";

V tomto příkladu odkazuje název **superstores@cliff_ol** na databázi v databázovém serveru, který klientský počítač rozpoznává. Aby mohlo dojít k přesměrování, musí administrátor změnit informace o propojitelnosti tak, aby tento název odkazoval na jiný databázový server. Bude pravděpodobně nezbytné upravit časový interval, po který klient čeká, než se pokusí o připojení, případně počet pokusů, které tato funkce vykoná. Poskytněte dostatečně dlouhý časový interval pro administrativní akci, kterou je třeba provést v databázovém serveru (změně informací o propojitelnosti nebo změně typu sekundárního databázového serveru na standardní).

Porovnání různých mechanismů přesměrování

Tabulka 19-6 shrnuje rozdíly mezi třemi mechanismy přesměrování.

Tabulka 19-6. Porovnání metod přesměrování pro různé strategie propojitelnosti

Kritéria porovnání	DBPATH		Informace o propojitelnosti		INFORMIXSERVER
	Automatické přesměrování	Přesměrování uživatelem	Automatické přesměrování	Přesměrování uživatelem	Přesměrování uživatelem
Kdy je klient přesměrován?	Když se klient opět pokusí připojit k určené databázi.		Po změně informací o propojitelnosti ze strany administrátora, kdy se klient opět pokusí vytvořit spojení s databázovým serverem.		Když se klient restartuje a načte novou hodnotu proměnné prostředí INFORMIXSERVER .
Musí být klienti restartováni nebo přesměrováni?	Ne	Ano	Ne	Ano	Ano
Jaký je rozsah přesměrování?	Přesměrování jednotlivých klientů	Přesměrování jednotlivých klientů	Přesměrování všech klientů využívajících příslušný databázový server	Přesměrování jednotlivých klientů	Přesměrování jednotlivých klientů
Jsou vyžadovány změny proměnných prostředí?	Ne		Ne		Ano

Návrh klientů skupiny replikace dat

Tato část pojednává o různých aspektech návrhu (kromě aspektu přesměrování popsaného výše) klientů, kteří se připojují k databázovým serverům využívajícím replikaci dat.

Nastavení režimu uzamykání při přístupu k primárnímu databázovému serveru na čekání

Když databázový server provádí logické obnovení, obvykle odloží vytvoření indexu až na konec obnovení. Pokud je ale databázový server spuštěn jako sekundární databázový server, nachází se v režimu logické obnovy, dokud probíhá replikace dat. Sekundární databázový server musí tedy používat k vytváření indexu jiný mechanismus.

Použitý mechanismus pracuje následujícím způsobem. Pokud sekundární databázový server obdrží záznam logického protokolu, který vyžaduje odpovídající vytvoření indexu, odešle primárnímu databázovému serveru zpětnou zprávu s žádostí o fyzickou kopii indexu. V primárním databázovém serveru je aktualizovaná tabulka uzamčena. Vlastníkem zámku je jednotkový proces **dr_btssend**. Jednotkový proces aplikace, který je spuštěn, může pokračovat ve zpracování. Jednotkový proces **dr_btssend** ale nemůže uvolnit zámek, dokud sekundární databázový server nepotvrdí příjem indexu. Pokud se aplikace pokusí přistoupit do tabulky, zatímco je tabulka zamčena, tento pokus selže, pokud ovšem aplikace nenastavila režim uzamykání na čekání.

Nemají-li aplikace nastavený režim uzamykání na režim WAIT, mohou se vyskytnout neočekávané chyby. Mnoho příkazů jazyka SQL například způsobuje aktualizace indexů tabulek katalogu. Následující sekvence příkazů jazyka SQL selže, není-li režim uzamykání aplikace nastaven na WAIT:

```
CREATE DATABASE db_name;
DATABASE db_name;
CREATE TABLE tab_name;
```

Tyto příkazy jazyka SQL by selhaly, protože příkaz CREATE DATABASE vytváří indexy tabulky katalogu **systables**, a proto umístí na tabulku zámek, dokud nejsou indexy

zkopírované na sekundární databázový server. Mezitím se příkaz CREATE TABLE pokouší vložit řádek do tabulky katalogu **systables**. Vložení řádku ale nebude úspěšné, protože tabulka je uzamčena.

Tato aplikace selže, protože oba příkazy CREATE DATABASE a CREATE TABLE způsobí aktualizaci indexu tabulky katalogu **systables**.

Návrh klientů k použití sekundárního databázového serveru

Chcete-li dosáhnout určitého stupně rozložení zatížení při použití replikaci dat, umožněte některým klientským aplikacím používat sekundární databázový server v replikačním páru. Všechny klientské aplikace využívající sekundární databázový server navrhnete s ohledem na následující body:

- Všechny příkazy jazyka SQL, které se pokoušejí měnit data, se nepodaří.
- Úroveň uzamčení a izolace nejsou totožné s úrovněmi ve standardním databázovém serveru.
- Pro dočasné tabulky a řazení musejí být použity dočasné prostory dbSPACE.

Následující odstavce se těmito body zabývají podrobněji.

Žádné příkazy ke změně dat

Příkazy SQL, které aktualizují prostory dbSPACE v režimu logické obnovy nejsou v sekundárním databázovém serveru povolené. Například příkazy ALTER FRAGMENT a CREATE DATABASE by měly za následek vznik chyb. Úplný seznam příkazů měnících data data naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

Chcete-li klientům užívajícím sekundární databázový server zabránit ve vydávání příkazů aktualizujících data, můžete provést jeden z následujících kroků:

- Vytvořte klientské aplikace, které nevydávají příkazy k aktualizaci.
- Všechny příkazy k aktualizaci změňte na podmíněné.

Aby byly příkazy provádějící aktualizaci podmíněnými příkazy, ujistěte se, že klientské aplikace kontrolují položku **slqwarn6** pole **sqlwarn** ve struktuře **sqlca** jazyka ESQL/C (nebo ekvivalentní hodnoty v případě dalších rozhraní API jazyka SQL). Pokud je databázový server spuštěn jako sekundární databázový server, nastaví položku **slqwarn6** na hodnotu **W**.

Úroveň uzamčení a izolace

Protože všichni klienti, kteří využívají sekundární databázový server mohou data pouze číst, není uzamčení zajišťující izolaci mezi těmito klienty nutné. Klient používající sekundární databázový server ale není chráněn před aktivitou uživatelů na primárním databázovém serveru, protože jednotkové procesy **logrecvr**, které provádějí logickou obnovu, nepoužívají uzamykání.

Pokud například klient připojený k sekundárnímu databázovému serveru načte řádek, nic nezabrání uživateli na primárním databázovém serveru, aby tento řádek aktualizoval, a to i v případě, že klient připojený k sekundárnímu databázovému serveru vydal příkaz SET ISOLATION TO REPEATABLE READ. Aktualizace se na sekundárním databázovém serveru projeví při zpracování záznamů logického protokolu pro potvrzenou transakci. Všechny dotazy směřované na sekundární databázový server tedy při zohlednění změn, k nimž dochází na primárním databázovém serveru odpovídají úrovni izolace neaktualizované čtení, a to i přesto, že klient, který používá sekundární databázový server, může explicitně nastavit úroveň izolace na jinou než neaktualizované čtení.

Důležité: Všechny dotazy na sekundární databázový server probíhají v režimu neaktualizovaného čtení. Nedotazujte sekundární databázový server, zatímco

probíhá replikace záznamů protokolu obsahujících příkazy jazyka pro definici dat (*DDL*) a jejich použití v sekundárním databázovém serveru.

Provedení operací typu DDL na primárním databázovém serveru:

1. Ukončete veškerou aktivitu (například transakce) v sekundárním databázovém serveru.
2. Spusťte příkazy jazyka DDL v primárním databázovém serveru.
Úplný seznam příkazů jazyka DDL naleznete v příručce *IBM Informix Guide to SQL: Syntax*.
3. Vynuťte v primárním databázovém serveru kontrolní bod pomocí příkazu **onmode -c**.
4. Počkejte, než bude tento kontrolní bod replikován do sekundárního databázového serveru.
5. Obnovte aktivitu v sekundárním databázovém serveru.

Použití dočasných prostorů dbspace k řazení a pro dočasné tabulky

I přesto, že je sekundární databázový server spuštěn v režimu pouze pro čtení, zapisuje data, pokud potřebuje provést řazení nebo vytvořit dočasnou tabulku. “Dočasné prostory dbspace” na stránce 9-11 popisuje, kde nalezne databázový server dočasný prostor, který může používat k řazení nebo pro dočasné tabulky. Aby sekundární databázový server nemusel zapisovat do prostoru dbspace, který se nachází v režimu logické obnovy, musíte provést následující kroky:

- Ujistěte se, že existuje jeden nebo více dočasných prostorů dbspace. Pokyny pro vytváření dočasného prostoru dbspace naleznete v části “Vytvoření prostoru dbspace, který používá výchozí velikost stránky” na stránce 10-9.
- Pak se rozhodněte pro jeden z následujících kroků:
 - Nastavte parametr **DBSPACETEMP** v souboru **ONCONFIG** sekundárního databázového serveru na dočasný prostor nebo prostory dbspace.
 - Nastavte proměnnou prostředí **DBSPACETEMP** klientských aplikací na dočasný prostor nebo prostory dbspace.

V případě sekundárních serverů SD nastavte ke konfiguraci dočasných prostorů dbspace pro použití na sekundárním serveru SD konfigurační parametr **SDS_TEMPDBS**.

Kapitola 20. Použití replikace HDR (verze pro podniky/pracovní skupiny)

Obsah kapitoly	20-2
Plánování replikace HDR	20-2
Konfigurace systému pro replikaci HDR	20-2
Splnění požadavků na hardware a operační systém	20-3
Splnění požadavků na databázi a data	20-3
Splnění požadavků na konfiguraci databázového serveru.	20-3
Verze databázového serveru	20-4
Konfigurace paměťových prostorů a bloků	20-4
Použití jiné než výchozí velikosti stránky v prostředí replikace HDR	20-4
Zrcadlení	20-4
Konfigurace fyzického protokolu.	20-5
Pásková zařízení pro zálohování prostorů dbpace a logického protokolu.	20-5
Konfigurace logického protokolu.	20-5
Konfigurační parametry replikace HDR	20-5
Konfigurace propojitelnosti replikace HDR.	20-6
První spuštění replikace HDR.	20-6
Zkrácení doby nastavení alternativní metodou zálohování	20-9
Provádění základních úloh administrace.	20-9
Změna konfiguračních parametrů databázového serveru	20-9
Zálohování paměťových prostorů a souborů logického protokolu	20-10
Změna režimu protokolování databázi	20-10
Přidání a vypuštění bloků a paměťových prostorů	20-10
Přejmenování bloků	20-10
Uložení stavu bloku na sekundární databázový server	20-11
Použití a změna zrcadlení bloků	20-11
Správa fyzického protokolu	20-12
Správa logického protokolu	20-12
Správa virtuálních procesorů.	20-12
Správa sdílené paměti.	20-12
Replikace indexu sekundárního databázového serveru	20-13
Šifrování přenosu dat mezi databázovými servery HDR.	20-14
Úprava vyprázdnění LRU a automatické ladění v serverových párech HDR	20-15
Změna režimu databázového serveru	20-15
Změna typu databázového serveru	20-16
Monitorování stavu replikace HDR.	20-17
Použití obslužných programů příkazového řádku.	20-17
onstat -g dri.	20-17
oncheck -pr.	20-17
Použití tabulek SMI	20-18
Použití programu ON-Monitor (systém UNIX)	20-18
Selhání replikace HDR	20-18
Definice selhání replikace HDR.	20-18
Zjišťování selhání replikace HDR	20-19
Akce, které je třeba provést při selhání replikace HDR	20-19
Co provést v případě selhání replikace HDR	20-19
Postup v případě selhání sekundárního databázového serveru	20-20
Postup v případě selhání primárního databázového serveru	20-20
Obnovení dat po selhání prostředku	20-22
Obnovení po selhání prostředku na primárním databázovém serveru	20-22
Obnovení po selhání prostředku na sekundárním databázovém serveru	20-23
Replikace indexu na sekundární server.	20-23
Restartování replikace HDR po selhání	20-24
Restartování po poškození kritických dat	20-24
Selhání kritického prostředku na primárním databázovém serveru.	20-24

Selhání kritického prostředku na sekundárním databázovém serveru	20-25
Selhání kritického prostředku na obou databázových serverech	20-25
Restartování bez poškození kritických dat.	20-25
Restartování po chybě sítě	20-25
Restartování po selhání sekundárního databázového serveru	20-26
Restartování po selhání primárního databázového serveru	20-26

Obsah kapitoly

Tato kapitola popisuje způsob plánování, konfigurace, spouštění a sledování replikace dat HDR (HDR) pro server Dynamic Server a způsob obnovení dat po selhání prostředku. Chcete-li používat replikaci HDR, přečtěte si nejdříve tuto kapitolu. Chcete-li používat replikaci IBM Informix Enterprise Replication, naleznete další informace v příručce *IBM Informix Dynamic Server Enterprise Replication Guide*.

Kapitola 19, “Přehled replikace dat (vydání Enterprise/Workgroup)”, na stránce 19-1 vysvětluje, co je replikace HDR, jak pracuje a jaké jsou možnosti návrhu klientských aplikací v prostředí replikace HDR.

Replikace HDR je dostupná se standardní verzí dynamického serveru. Replikace HDR nefunguje se serverem IBM Informix Dynamic Server Express Edition.

Plánování replikace HDR

Než začnete nastavovat počítače a databázové servery k použití replikace HDR, měli byste replikaci nejprve naplánovat. Následující seznam obsahuje úlohy při plánování:

- Vyberte a získejte vhodný hardware.
- Pokud k ukládání dat, která chcete replikovat, používáte více databázových serverů, přemístěte tato data a znovu je rozdělte tak, aby mohla být spravována jedním databázovým serverem.
- Zkontrolujte, zda všechny databáze, které chcete replikovat, používají protokolování transakcí. Další informace o protokolování transakcí uvádí Kapitola 12, “Správa režimu protokolování databáze”, na stránce 12-1.
- Vytvořte klientské aplikace, které využijí oba databázové servery v replikačním páru. Informace o návrhu naleznete v části “Přesměrování a propojitelnost pro klienty replikace dat” na stránce 19-23 a v části “Návrh klientů k použití sekundárního databázového serveru” na stránce 19-32.
- Vytvořte harmonogram pro první spuštění replikace HDR.
- Navrhněte paměťový prostor a harmonogram zálohování logického protokolu pro primární databázový server.
- Naplánujte, jak budete řešit selhání databázového serveru a jak budete po selhání restartovat replikaci HDR. Přečtěte si část “Přesměrování a propojitelnost pro klienty replikace dat” na stránce 19-23.

Konfigurace systému pro replikaci HDR

Pro konfiguraci vašeho systému pro replikaci HDR je třeba provést následující kroky:

- Splnit požadavky na hardware a operační systém.
- Splnit požadavky na databázi a data.
- Splnit požadavky na konfiguraci databázového serveru.
- Nakonfigurovat propojitelnost replikace HDR.

V této části je každé z těchto témat vysvětleno.

Splnění požadavků na hardware a operační systém

Aby mohl pár databázových serverů replikace HDR správně pracovat, musí splňovat následující hardwarové požadavky:

- Počítače, ve kterých je spuštěn primární a sekundární databázový server, musejí být identické (stejný dodavatel a architektura).
- Operační systémy na počítačích, ve kterých je spuštěn primární a sekundární databázový server, musejí být identické.
- Hardware, na kterém je spuštěn primární a sekundární databázový server, musí podporovat síťový provoz.
- Prostory dbspace pro primární a sekundární databázové servery musejí mít stejnou velikost přiděleného diskového prostoru. Druh diskového prostoru není podstatný, na dvou databázových serverech lze použít jakoukoli kombinaci přímých a předpřipravených prostorů.

Splnění požadavků na databázi a data

Aby mohl pár databázových serverů replikace HDR správně pracovat, musí splňovat následující požadavky na databázi a na data:

- **Všechna data musí být protokolována**

Všechny databáze, které chcete replikovat, musejí mít zapnuté protokolování transakcí.

Tento požadavek je důležitý, protože databázové servery používají k aktualizaci spravovaných dat záznamy logického protokolu z primárního serveru. Pokud databáze spravovaná primárním databázovým serverem nepoužívá protokolování, aktualizace těchto databází nevytváří záznamy protokolu, takže sekundární databázový server nemá prostředky k aktualizaci replikovaných dat. Protokolování může probíhat s vyrovnávací pamětí i bez vyrovnávací paměti.

Pokud potřebujete před zahájením replikace HDR zapnout protokolování transakcí, naleznete další informace v části “Zapnutí protokolování transakcí pomocí obslužného programu ontape” na stránce 12-4.

- **Data musí být uložena v prostorech dbspace nebo sbspace.**

Pokud má primární databázový server v prostorech blobspace uložené jednoduché velké objekty, nejsou změny dat uvnitř těchto prostorů blobspace replikovány jako součást normálního zpracování replikace HDR. Jednoduché velké objekty uvnitř prostorů dbspace jsou však replikovány.

Inteligentní velké objekty, které jsou uloženy v prostorech sbspace, jsou replikovány.

Prostory sbspace musí být protokolovány. Uživatelské typy (UDT) jsou replikovány, pokud neukládají data mimo řádky do souborů operačního systému. Datové typy s daty ukládanými mimo řádek jsou replikovány tehdy, pokud jsou jejich data uložena v prostoru sbspace nebo v jiné tabulce téhož databázového serveru.

Splnění požadavků na konfiguraci databázového serveru

Oba databázové servery musí být zcela nakonfigurovány, aby tento pár databázových serverů replikace HDR pracoval správně. Informace o konfiguraci databázového serveru uvádí Kapitola 1, “Instalace a konfigurace databázového serveru”, na stránce 1-1. Důležité aspekty této konfigurace potom můžete použít při konfiguraci druhého databázového serveru. Další informace o konfiguračních parametrech naleznete v příručce *IBM Informix Administrator's Reference*.

Tato část popisuje následující aspekty konfigurace párů databázových serverů replikace HDR, které je třeba vzít v úvahu:

- Verze databázového serveru
- Konfigurace paměťových prostorů a bloků

- Zrcadlení
- Konfigurace fyzického protokolu
- Pásková zařízení pro zálohování prostorů dbspace a logického protokolu
- Konfigurace logického protokolu
- Konfigurační parametry replikace HDR

Verze databázového serveru

Verze databázového serveru na primárním a sekundárním databázovém serveru musejí být shodné.

Konfigurace paměťových prostorů a bloků

Počet prostorů dbspace, počet bloků, jejich velikosti, jejich názvy cest a jejich posuny musejí být shodné v primárním i v sekundárním databázovém serveru. Konfigurace navíc obsahuje alespoň jeden dočasný prostor dbspace. Další informace naleznete v části “Použití dočasných prostorů dbspace k řazení a pro dočasné tabulky” na stránce 19-33.

Jen pro UNIX

Pro názvy cest bloků byste měli použít symbolická propojení, která jsou popsána v části “Přidělení diskového prostoru s přímým přístupem v systému UNIX” na stránce 10-5.

Důležité: Pokud pro názvy cest bloků nepoužíváte symbolická propojení, nelze název bloku snadno změnit. Další informace naleznete v části “Přejmenování bloků” na stránce 20-10.

Konec Jen pro UNIX

Následující parametry v souboru ONCONFIG musí mít stejnou hodnotu na každém databázovém serveru:

- ROOTNAME
- ROOTOFFSET
- ROOTPATH
- ROOTSIZE

Použití jiné než výchozí velikosti stránky v prostředí replikace HDR

Velikost stránky prostoru dbspace a specifikace společné oblasti vyrovnávacích pamětí jsou automaticky šířeny z primárního do sekundárního databázového serveru. Primární i sekundární databázové servery musí mít stejnou společnou oblast vyrovnávacích pamětí, ale počet vyrovnávacích pamětí ve společných oblastech vyrovnávacích oblastí se shodovat nemusí.

Zrcadlení

Ve dvojici databázových serverů nemusíte nastavovat parametr MIRROR na stejnou hodnotu. Na jednom databázovém serveru můžete zrcadlení povolit a na druhém zakázat. Pokud však pro kořenový blok primárního databázového serveru určíte zrcadlený blok, musíte také určit zrcadlený blok pro kořenový blok na sekundárním databázovém serveru. Z toho důvodu musejí být následující parametry souboru ONCONFIG nastaveny na obou databázových serverech na stejnou hodnotu:

- MIRROROFFSET
- MIRRORPATH

Konfigurace fyzického protokolu

Fyzický protokol by měl být na obou databázových serverech identický. Následující parametry v souboru ONCONFIG musí mít stejnou hodnotu na každém databázovém serveru:

- PHYSDBS
- PHYSFILE

Pásková zařízení pro zálohování prostorů dbspace a logického protokolu

Pro primární a sekundární databázové servery můžete určit různá pásková zařízení.

Pokud používáte program ON-Bar, nastavte konfigurační parametry programu ON-Bar na obou databázových serverech na stejnou hodnotu. Další informace o parametrech programu ON-Bar naleznete v části *IBM Informix Backup and Restore Guide*.

Pokud používáte obslužný program **ontape**, měla by mít zařízení pro zálohování paměťového prostoru a logického protokolu stejnou velikost pásky a velikost bloku na pásce. Následující parametry v souboru ONCONFIG musí mít stejnou hodnotu na každém databázovém serveru:

- LTAPEBLK
- LTAPESIZE
- TAPEBLK
- TAPESIZE

Chcete-li využít veškerou fyzickou kapacitu pásky, nastavte parametry LTAPESIZE a TAPESIZE na hodnotu 0.

Konfigurace logického protokolu

Všechny záznamy protokolu jsou replikovány na sekundární server. Velikost a počet souborů logického protokolu musejí být na obou databázových serverech nakonfigurovány stejně. Následující parametry v souboru ONCONFIG musí mít stejnou hodnotu na každém databázovém serveru:

- LOGFILES
- LOGSIZE
- DYNAMIC_LOGS

Databázový server protokoluje přidání souborů logického protokolu. Soubory logického protokolu dynamicky přidané v primárním serveru jsou automaticky replikovány na sekundární server. Hodnota DYNAMIC_LOGS sice nemá na sekundárním serveru žádný účinek, ale měli byste hodnotu DYNAMIC_LOGS udržovat tak, aby byla synchronizována s hodnotou na primárním serveru pro případ, že by se jejich role vyměnily.

Konfigurační parametry replikace HDR

Následující konfigurační parametry replikace HDR musejí být na obou databázových serverech v replikačním páru nastaveny na stejnou hodnotu:

- DRAUTO
- DRINTERVAL
- DRTIMEOUT

Konfigurace propojitelnosti replikace HDR

Aby mohl pár databázových serverů replikace HDR pracovat, musejí být oba databázové servery schopné navázat vzájemné připojení. K tomu je nutné, aby údaje o propojitelnosti na každém počítači, na kterém je spuštěný databázový server v replikačním páru, obsahovaly alespoň následující položky, které určují:

- databázový server spuštěný v tomto počítači,
- druhý databázový server v replikačním páru.

Důležité: Nastavte pole **nettype** v souboru **sqlhosts** nebo v registru a konfigurační parametr **NETTYPE** na síťový protokol, jako je například **ontlftp**, **onsoctcp** nebo **ontlisp**, tak, aby mohly databázové servery ve dvou různých počítačích navzájem komunikovat. Replikace HDR nefunguje, určuje-li pole **typ sítě** jiný protokol než síťový, například **onipcshm**, **onipcstr** nebo **onipcnmp**.

Další informace o přesměrování klientů a změně propojitelnosti naleznete v části “Přesměrování a propojitelnost pro klienty replikace dat” na stránce 19-23.

První spuštění replikace HDR

Po dokončení konfigurace replikace HDR je spuštění replikace HDR připraveno. Tato část podrobně popisuje kroky nezbytné pro spuštění replikace HDR.

Předpokládáme, že chcete replikaci HDR spustit na dvou databázových serverech, na serveru **ServerA** a na serveru **ServerB**. V následujících krocích je podrobně popsán postup při spuštění replikace HDR se serverem **ServerA** jako primárním databázovým serverem a se serverem **ServerB** jako sekundárním databázovým serverem. Tabulka 20-1 na stránce 20-7 uvádí příkazy nutné k provedení každého kroku a zprávy odeslané do protokolu zpráv. Zálohování a obnovení můžete provést pomocí obslužného programu **ontape** nebo pomocí programu **ON-Bar**. V průběhu celého postupu musíte používat stejný obslužný program.

Důležité: Pokud zálohování a obnovení provádíte pomocí programu **ON-Bar**, je na obou databázových serverech vyžadován obslužný program **ontape**. Obslužný program **ontape** nelze z databázových serverů, které se účastní replikace HDR, odstranit. V případě potřeby můžete replikaci HDR také nastavit pomocí standardního programu **ON-Bar** nebo pomocí příkazů obslužného programu **ontape** pro externí zálohování a obnovení.

Postup při spouštění replikace HDR:

1. Nainstalujte na oba databázové servery uživatelské typy, uživatelské rutiny a moduly **DataBlade** a poté je zaregistrujte pouze na serveru **ServerA**.
2. Vytvořte zálohu úrovně 0 serveru **ServerA**.
3. Proveďte fyzické obnovení serveru **ServerB** ze zálohy úrovně 0, kterou jste vytvořili v kroku 1. Neprovádějte logické obnovení.
Pokud používáte:
 - Program **ON-Bar**, použijte k provedení fyzického obnovení příkaz **onbar -r -p**.
 - Program **ON-Bar** a provádíte externí obnovení, použijte k provedení fyzického obnovení příkaz **onbar -r -p -e**.
 - Obslužný program **ontape**, použijte volbu **ontape -p**. Volbu **ontape -r** nelze použít, protože provádí fyzické i logické obnovení.
 - Obslužný program **ontape** a provádíte externí obnovení, použijte k provedení fyzického obnovení příkaz **ontape -p -e**.
4. Pomocí příkazu **onmode -d** nastavte typ serveru **ServerA** jako primární a zadejte název přidruženého sekundárního databázového serveru (v tomto případě **ServerB**).

Pokud použijete příkaz **onmode -d**, pokusí se databázový server navázat spojení s druhým databázovým serverem v páru replikace HDR a spustí operaci replikace HDR. Pokus o navázání spojení uspěje pouze tehdy, je-li druhý databázový server již nastaven na správný typ.

V tuto chvíli není server **ServerB** online a není nastaven na sekundární typ, nebude tedy spojení replikace HDR navázáno.

5. Pomocí příkazu **onmode -d** nastavte typ serveru **ServerB** na sekundární a zadejte přidružený primární databázový server.

Server **ServerB** se pokusí navázat spojení replikace HDR s primárním databázovým serverem (**ServerA**) a spustí operaci. Připojení by mělo být úspěšně navázáno.

Před zahájením replikace HDR provede sekundární databázový server logickou obnovu pomocí záznamů logického protokolu, které byly zapsány na primární databázový server během kroku 2. Pokud se všechny tyto záznamy logického protokolu stále nacházejí na primárním databázovém serveru, odešle primární databázový server tyto záznamy přes síť přímo sekundárnímu databázovému serveru a k logické obnově dojde automaticky.

Pokud jste zálohovali a uvolnili soubory logického protokolu na primárním databázovém serveru, nejsou již záznamy v těchto souborech na disku. Sekundární databázový server vás vyzve k obnovení těchto souborů z pásky. V tomto případě musíte provést krok 6.

Důležité: Kroky 3 a 5 musíte provést společně. Pokud po kroku 3 potřebujete vypnout a restartovat sekundární databázový server, musíte krok 3 opakovat.

6. Pokud záznamy logického protokolu, které byly zapsány na primární databázový server, již na primárním disku nejsou, sekundární databázový server vás vyzve k obnovení těchto souborů z pásky.

Pokud musí sekundární databázový server číst zálohované soubory logického protokolu přes síť, nastavte parametry páskového zařízení na sekundárním databázovém serveru na zařízení v počítači, ve kterém je spuštěn primární databázový server, nebo na zařízení ve stejném umístění jako primární databázový server.

Po obnovení všech souborů logického protokolu z pásky bude obnovení dokončeno pomocí souborů logického protokolu na disku primárního databázového serveru.

Tabulka 20-1. Postup prvního spuštění replikace HDR

Krok	Na primárním serveru	Na sekundárním serveru
1.	Nainstalujte uživatelské rutiny (UDR), uživatelské typy (UDT) a moduly DataBlade. Zaregistrujte uživatelské rutiny, uživatelské typy a moduly DataBlade.	Nainstalujte uživatelské rutiny (UDR), uživatelské typy (UDT) a moduly DataBlade.
2.	Příkaz ontape ontape -s -L 0 ON-Bar Příkaz onbar -b -L 0 Zprávy do protokolu zpráv Archivace úrovně 0 v prostoru rootdbs byla zahájena. Archivace v prostoru rootdbs byla dokončena.	

Tabulka 20-1. Postup prvního spuštění replikace HDR (pokračování)

Krok	Na primárním serveru	Na sekundárním serveru
3	<p>Obslužný program onmode</p> <p>onmode -d primary <i>název_sek</i></p> <p>Zprávy do protokolu zpráv</p> <p>DR: nový typ = primární, sekundární server name = <i>název_sek</i></p> <p>DR: Pokus o připojení k sekundárnímu serveru</p> <p>DR: Nelze se připojit k sekundárnímu serveru</p>	
4.		<p>Příkaz ontape</p> <p>ontape -p nebo ontape -r -p -e</p> <p>Na výzvu k zálohování protokolů odpovězte ne.</p> <p>Příkaz ON-Bar</p> <p>onbar -r -p nebo onbar -r -p -e</p> <p>Zprávy do protokolu zpráv</p> <p>Databázový server IBM Informix byl inicializován - sdílená paměť byla inicializována Režim obnovy Bylo spuštěno fyzické obnovení prostoru rootdbs. Bylo dokončeno fyzické obnovení prostoru rootdbs.</p>
5.		<p>Příkaz onmode</p> <p>onmode -d secondary <i>název_prim</i></p> <p>Zprávy do protokolu zpráv</p> <p>DR: nový typ = sekundární, primární server name = <i>název_prim</i></p> <p>Pokud jsou na disku primárního databázového serveru stále uloženy všechny záznamy logického protokolu, které byly zapsány na primární databázový server během kroku 1, pak sekundární databázový server tyto záznamy přečte a provede logickou obnovu. (Jinak musí být proveden krok 5).</p>
	<p>Zprávy do protokolu zpráv</p> <p>DR: Primární server byl připojen DR: Primární server je schopný provozu</p>	<p>Zprávy do protokolu zpráv</p> <p>DR: Probíhá pokus o připojení k primárnímu serveru DR: Sekundární server byl připojen DR: Probíhá obnova z disku po selhání. Počet pracovních jednotkových procesů obnovy, které budou spuštěny, je <i>n</i>. Byla spuštěna logická obnova Spustit logickou obnovu - spustit protokol <i>n</i>, ukončit protokol? Počáteční umístění protokolu - <i>n 0xnnnnn</i>DR: Sekundární je server schopný provozu.</p>

Tabulka 20-1. Postup prvního spuštění replikace HDR (pokračování)

Krok	Na primárním serveru	Na sekundárním serveru
6.		<p>Příkaz ontape</p> <p>ontape -l</p> <p>ON-Bar Příkaz</p> <p>onbar -r -l</p>
	<p>Zprávy do protokolu zpráv</p> <p>DR: Primární server byl připojen DR: Primární server je schopný provozu</p>	<p>Zprávy do protokolu zpráv</p> <p>DR: Sekundární server byl připojen DR: Probíhá obnova z disku po selhání. Počet pracovních jednotkových procesů obnovy, které budou spuštěny, je <i>n</i>. Byla spuštěna logická obnova Spustit logickou obnovu - spustit protokol <i>n</i>, ukončit protokol? <i>Počáteční umístění protokolu - n 0xnnnnn</i> DR: Sekundární server je schopný provozu</p>

Zkrácení doby nastavení alternativní metodou zálohování

Používáte-li při zálohování a obnově dat obslužný program **ontape**, můžete namísto zálohování souboru do zařízení zvýšit rychlost nastavení replikace HDR nastavením konfiguračního parametru TAPEDEV na hodnotu STDIO. Konfigurační parametr TAPEDEV používá hodnotu STDIO jako speciální hodnotu pro konfiguraci standardních datových proudů vstupu - výstupu.

Důležité: Používáte-li hodnotu STDIO, nepracuje obslužný program **ontape** stejně jako při zálohování do zařízení.

Další informace o použití hodnoty STDIO naleznete v části *IBM Informix Backup and Restore Guide*.

Provádění základních úloh administrace

Tato část obsahuje pokyny pro provádění úloh administrace databázového serveru, když je v systému spuštěná replikace HDR.

Změna konfiguračních parametrů databázového serveru

Některé konfigurační parametry musejí být na obou databázových serverech v replikačním páru nastaveny na stejnou hodnotu (jak je uvedeno v části “Splnění požadavků na konfiguraci databázového serveru” na stránce 20-3.) Ostatní konfigurační parametry serveru Dynamic Server mohou být nastaveny na rozdílné hodnoty.

Provádění změn v souborech ONCONFIG:

1. Převeďte oba databázové servery do režimu offline pomocí volby **onmode -k**. Pokud je parametr DRAUTO nastaven na hodnotu RETAIN_TYPE nebo REVERSE_TYPE, bude snadnější převést do režimu offline nejprve sekundární databázový server.
2. Změňte parametry v obou databázových serverech.
3. Počínaje databázovým serverem, který jste do režimu offline převedli jako poslední, převedte všechny databázové servery zpět do režimu online.

Pokud jste například převedli do režimu offline jako poslední sekundární databázový server, převedte do režimu online nejprve sekundární databázový server. Tabulka 20-1

na stránce 20-7 a Tabulka 20-1 na stránce 20-7 ukazují postupy, jak převést primární a sekundární databázový server do režimu online.

Pokud konfigurační parametr nevyžaduje stejnou hodnotu na obou databázových serverech v replikačním páru, můžete hodnotu na primárním i sekundárním databázovém serveru změnit samostatně.

Zálohování paměťových prostorů a souborů logického protokolu

Pokud používáte replikaci HDR, musíte soubory logického protokolu a paměťové prostory zálohovat pouze na primárním databázovém serveru. Pokud je ale typ databázového serveru změněn na standardní, buďte připraveni provést zálohování paměťového prostoru a logického protokolu na sekundárním databázovém serveru.

Na obou databázových serverech musíte používat pro zálohování a obnovení tentýž nástroj.

Použitá velikost bloku a velikost pásky (pro zálohování paměťového prostoru i logického protokolu) musí být na primárním i sekundárním databázovém serveru stejná.

Chcete-li automaticky využít celou fyzickou kapacitu pásky, můžete pomocí příkazu **ontape** nastavit velikost pásky na hodnotu 0.

Změna režimu protokolování databází

Používáte-li replikaci HDR, nemůžete zapínat protokolování transakcí pro databáze na primárním databázovém serveru. Protokolování můžete pro danou databázi vypnout, další změny této databáze pak ale nebudou duplikovány na sekundární databázový server.

Zapnutí protokolování databáze:

1. Vypněte replikaci HDR vypnutím sekundárního databázového serveru.
2. Zapněte protokolování databáze.
Spustíte-li replikaci dat po zapnutí protokolování databáze, aniž provedete zálohování úrovně 0 na primárním databázovém serveru a obnovení na sekundárním databázovém serveru, mohly by databáze na primárním a sekundárním databázovém serveru obsahovat rozdílná data. Tato situace by mohla způsobit problémy při replikaci dat.
3. Proveďte zálohování úrovně 0 na primárním databázovém serveru a na sekundárním databázovém serveru proveďte obnovení. Tento postup je popsán v části “První spuštění replikace HDR” na stránce 20-6.

Přidání a vypuštění bloků a paměťových prostorů

Operace rozvržení disků, například přidávání nebo vypouštění bloků nebo prostorů dbspace, lze provádět pouze z primárního databázového serveru. Tato operace je na sekundárním databázovém serveru replikována. Toto opatření zajišťuje konzistenci rozvržení disků na obou databázových serverech.

Název cesty k adresáři nebo skutečný soubor bloku musí existovat předtím, než ho vytvoříte. Před vytvořením bloku na primárním databázovém serveru vždy zkontrolujte, zda na sekundárním databázovém serveru existují názvy cest (a posuny, pokud je lze použít), jinak databázový server replikaci dat vypne.

Přejmenování bloků

Pokud pro názvy cest k blokům používáte symbolická propojení, můžete bloky přejmenovat, zatímco replikace HDR pracuje. Další informace o přejmenování bloků naleznete v příručce *IBM Informix Backup and Restore Guide*.

Pokud pro názvy cest k blokům nepoužíváte symbolická propojení, musíte převést při přejmenování bloků oba databázové servery do režimu offline, než bude dokončeno studené obnovení databázového serveru.

Přejmenování bloků na serveru replikace HDR, který selhal:

1. Změňte režim nepoškozeného serveru na standardní.
2. Proveďte zálohování úrovně 0 standardního serveru.
3. Vypněte standardní server.
4. Přejmenujte bloky na standardním serveru během studeného obnovení z nového archivu úrovně 0 (pokyny naleznete v příručce *IBM Informix Backup and Restore Guide*).
5. Spusťte standardní server.
6. Proveďte další zálohování úrovně 0 standardního serveru. Zkontrolujte, zda je server ve standardním režimu.
7. Obnovte server, který selhal, pomocí nové zálohy úrovně 0 a znovu vytvořte pár replikace HDR.

Uložení stavu bloku na sekundární databázový server

Je-li stav bloku *vypnut* nebo *online*) pro pár replikace dat na sekundárním databázovém serveru změněn a tento sekundární server bude před dokončením kontrolního bodu restartován, nebude aktualizovaný stav bloku uložen.

Vyprázdnění stavu bloku do rezervovaných stránek na sekundárním databázovém serveru zajistíte vynucením kontrolního bodu na primárním databázovém serveru a ověřením, zda byl kontrolní bod dokončen také na sekundárním databázovém serveru. Nový stav bloku je nyní zachován, i když je sekundární databázový sever restartován.

Je-li primární blok na sekundárním databázovém serveru vypnut, můžete primární blok obnovit ze zrcadleného bloku.

Obnovení primárního bloku ze zrcadleného bloku:

1. Přepněte primární blok do režimu online spuštěním příkazu **onspaces -s** na sekundárním databázovém serveru.
Primární blok můžete přepnout do režimu online také pomocí programu ISA.
2. Spuštěním příkazu **onmode -c** na primárním databázovém serveru vynuťte kontrolní bod.
3. Spuštěním příkazu **onmode -m** na primárním databázovém serveru ověřte, zda byl kontrolní bod skutečně proveden.
4. Spuštěním příkazu **onmode -m** na sekundárním databázovém serveru ověřte, zda byl kontrolní bod dokončen také na sekundárním databázovém serveru.

Po dokončení těchto kroků bude primární blok po restartování sekundárního databázového serveru v režimu online.

Použití a změna zrcadlení bloků

Než budete moci přidat zrcadlený blok, musí být diskový prostor na primárním i sekundárním databázovém serveru bloku již přidělen. Pokud chcete na jednom z databázových serverů v replikačním páru zrcadlit prostor dbspace, musíte vytvořit zrcadlené bloky pro daný prostor dbspace na *obou* databázových serverech. Obecné informace o přidělování diskového prostoru naleznete v části "Přidělení diskového prostoru" na stránce 10-3.

Nenastavujte konfigurační parametr MIRROR na hodnotu 1, jestliže zrcadlení nepoužíváte.

Operace rozvržení disku lze provádět pouze z primárního databázového serveru. Zrcadlený blok můžete tedy přidat nebo vypustit pouze na primárním databázovém serveru. Zrcadlený blok, který přidáte na primární databázový server nebo který z něj vypustíte, je přidán také na sekundární databázový server nebo z něj vypuštěn. Pro zrcadlený blok nově přidávaný na sekundární databázový server musíte provést obnovu zrcadla. Další informace naleznete v části “Obnova zrcadleného bloku” na stránce 18-6.

Pokud z primárního databázového serveru vypustíte blok, server Dynamic Server automaticky vypustí odpovídající blok na sekundárním databázovém serveru. Tento postup platí jak pro primární, tak pro zrcadlené bloky.

Pokud pro prostor dbspace na primárním databázovém serveru vypnete zrcadlení, server Dynamic Server nevypne zrcadlení pro odpovídající prostor dbspace na sekundárním databázovém serveru. Zrcadlení pro prostory dbspace na sekundárním databázovém serveru, které je nezávislé na primárním databázovém serveru, můžete vypnout pomocí příkazu **onspaces -r**. Další informace o vypínání zrcadlení naleznete v části “Ukončení zrcadlení” na stránce 18-7.

Zrcadlený blok můžete vypnout nebo obnovit na primárním i na sekundárním databázovém serveru. Tyto procesy jsou pro replikaci HDR transparentní.

Správa fyzického protokolu

Velikost fyzického protokolu musí být na obou databázových serverech stejná. Změníte-li velikost nebo umístění fyzického protokolu na primárním databázovém serveru, je tato změna replikována na sekundární databázový server. Hodnoty souboru ONCONFIG na sekundárním serveru jsou aktualizovány automaticky.

Další informace o změně velikosti nebo umístění fyzického protokolu uvádí Kapitola 16, “Správa fyzického protokolu”, na stránce 16-1.

Správa logického protokolu

Velikost logického protokolu musí být na obou databázových serverech stejná. Soubor logického protokolu můžete přidat nebo vypustit pomocí obslužného programu **onparams**, který popisuje Kapitola 14, “Správa souborů logických protokolů”, na stránce 14-1. Server Dynamic Server tuto změnu replikuje na sekundární databázový server. Parametr LOGFILES na sekundárním databázovém serveru však aktualizován není. Po zadání příkazu **onparams** z primárního databázového serveru musíte na sekundárním databázovém serveru ručně změnit parametr LOGFILES na požadovanou hodnotu. Tato změna se projeví až po provedení zálohování úrovně 0 kořenového prostoru dbspace na primárním databázovém serveru.

Pokud na primární databázový server přidáte soubor logického protokolu, bude tento soubor k dispozici pro použití, jakmile provedete zálohování úrovně 0, a bude označen příznakem F. Na sekundárním databázovém serveru je nový soubor logického protokolu stále opatřen příznakem A. Tato podmínka ale nebrání sekundárnímu databázovému serveru v zápisu do souboru.

Správa virtuálních procesorů

Počet virtuálních procesorů nemá na replikaci dat žádný vliv. Každý databázový server v páru lze konfigurovat a optimalizovat samostatně.

Správa sdílené paměti

Pokud na jednom databázovém serveru změňte parametry sdílené paměti v souboru ONCONFIG, musíte současně provést stejné změny parametrů sdílené paměti v souboru

ONCONFIG na druhém databázovém serveru. Postup při provádění těchto změn naleznete v části “Změna konfiguračních parametrů databázového serveru” na stránce 20-9.

Replikace indexu sekundárního databázového serveru

Pokud se poškodí index na sekundárním databázovém serveru replikace HDR a musí se znovu vytvořit, můžete provést jednu z následujících akcí:

- Ručně replikovat index z primárního serveru na sekundární server.
- Umožnit sekundárnímu serveru automaticky replikovat index, pokud jste tuto možnost na sekundárním serveru povolili.

Automatickou replikaci indexu na sekundárním databázovém serveru můžete povolit jednou z následujících akcí:

- Nastavte parametr **onmode -d idxauto** na hodnotu on.
- Nastavte konfigurační parametr DRIDXAUTO na hodnotu 1.

Zjistí-li po nastavení jedné z těchto hodnot některý jednotkový proces na sekundárním databázovém serveru, že je poškozen index, je tento index automaticky replikován na sekundární databázový server. Restartování replikace indexu může probíhat maximálně po dobu určenou konfiguračním parametrem DRTIMEOUT (v sekundách).

V některých případech byste mohli chtít replikovat index ručně, chcete-li například odložit opravu indexu, protože je zamknutá tabulka. Chcete-li replikovat index na sekundárním serveru replikace HDR ručně, vypněte funkci automatické replikace.

Funkci automatické replikace indexu můžete vypnout jedním z následujících způsobů:

- Nastavte volbu **onmode -d idxauto** na hodnotu Off.
- Nastavte konfigurační parametr DRIDXAUTO na hodnotu 0.

Je-li volba **onmode -d idxauto** nastavena na hodnotu off nebo je-li konfigurační parametr DRIDXAUTO nastaven na hodnotu 0 a sekundární server zjistí, že je index poškozen, můžete index na sekundárním serveru replikace HDR replikovat ručně pomocí příkazu **onmode -d index** v následujícím formátu.

```
onmode -d index databáze:[jméno_vlastníka].tabulka#index
```

Například:

```
onmode -d index cash_db:user_dx.table_12#index_z
```

V případě, že je index fragmentovaný a poškozený je jeden jeho fragment, přenese volba **onmode -d idxauto** pouze tento jeden dotčený fragment, zatímco volba **onmode -d index** přenese celý index.

Poznámka: Chcete-li zapnout nebo vypnout funkci automatické replikace indexu, můžete použít buď obslužný program **onmode** nebo konfigurační parametr DRIDXAUTO. Pokud použijete obslužný program **onmode**, nemusíte databázový server vypnout a restartovat. Pokud použijete parametr DRIDXAUTO, je databázový server restartován s nastavením, které jste zadali. Příkaz obslužného programu **onmode** nemění hodnotu konfiguračního parametru DRIDXAUTO. Pokud použijete příkaz obslužného programu **onmode**, je potřeba změnit hodnotu konfiguračního parametru DRIDXAUTO ručně.

V souboru **online.log** vytvořeném sekundárním serverem jsou uloženy informace o každém replikovaném indexu.

Šifrování přenosu dat mezi databázovými servery HDR

Chcete-li šifrovat přenos dat mezi párem databázových serverů HDR, můžete použít možnosti šifrování dynamického serveru. Udělejte to v případě, kdy chcete zajistit bezpečný přenos dat.

Poté, co šifrování umožníte, první databázový server v páru HDR data před posláním druhému serveru v páru zašifruje. Server přijímající data je dešifruje, jakmile data obdrží.

Nezbytné předpoklady: Chcete-li podpořit šifrovaná spojení HDR ve spojení se šifrováním typu klient/server modulu CSM (Communication Support Module) musí být nakonfigurovány dva následující síťové porty:

- Jeden síťový port musí být nakonfigurován pro replikaci HDR.
- Další síťový port musí být nakonfigurován pro připojení typu klient/server s modulem CSM.

Důležité: Replikaci HDR není možné spustit v síťovém připojení, které je konfigurováno k použití šifrování CSM pro připojení typu klient/server.

Možná bude potřeba dalších nebo větších vyrovnávacích pamětí, aby vyhovovaly velikosti šifrovaných dat.

Chcete-li šifrovat přenos dat mezi dvěma databázovými servery HDR:

1. U prvního serveru páru HDR nastavte následující konfigurační parametry.
 - Parametr ENCRYPT_HDR, který povoluje nebo zakazuje šifrování HDR.
 - Parametr ENCRYPT_CIPHERS, který určuje šifry a režimy, které se mají používat při šifrování.
 - Parametr ENCRYPT_MAC, který ovládá úroveň generování ověřovacího kódu zprávy (MAC)- message authentication code (MAC).
 - Parametr ENCRYPT_MACFILE, který určuje seznam názvů úplných cest souborů MAC klíčů.
 - Parametr ENCRYPT_SWITCH, který určuje počet minut mezi automatickým opětovným jednáním šifer a klíčů.

Chcete-li tyto parametry změnit, postupujte podle pokynů v části “Změna konfiguračních parametrů databázového serveru” na stránce 20-9.

2. Nastavte konfigurační parametry šifrování na sekundárním serveru. Konfigurační parametry ENCRYPT_HDR, ENCRYPT_CIPHERS, ENCRYPT_MAC a ENCRYPT_SWITCH musí mít stejnou hodnotu jako jim odpovídající konfigurační parametry na primárním serveru. Konfigurační parametr ENCRYPT_MACFILE může mít na každém serveru rozdílnou hodnotu, soubory však musí obsahovat stejné MAC klíče.

Určete například následující informace na primárním a sekundárním serveru v páru HDR:

Konfigurační parametr	Ukázka nastavení na primárním serveru	Ukázka nastavení na sekundárním serveru
parametr ENCRYPT_HDR	1	1
ENCRYPT_CIPHERS	all	all
ENCRYPT_MAC	medium	medium
ENCRYPT_MACFILE	/vobs/tristan/sqlldist/etc/ mac1.dat	vobs/tristan/sqlldist/etc/ mac2.dat

Konfigurační parametr	Ukázka nastavení na primárním serveru	Ukázka nastavení na sekundárním serveru
ENCRYPT_SWITCH	60,60	60,60

V tomto příkladě je **mac1.dat** jméno souboru v cestě parametru ENCRYPT_MACFILE pro primární server a **mac2.dat** je jméno souboru v cestě parametru ENCRYPT_MACFILE pro sekundární server. Jinak jsou všechna nastavení na obou serverech stejná.

Tyto konfigurační parametry použijte jen k určení šifrovací informace pro HDR. Informaci pro šifrování HDR nelze určit pomocí volby CSM v souboru SQLHOSTS.

Šifrování replikace HDR pracuje společně s šifrováním replikace Enterprise Replication a funguje nezávisle na tom, zda je šifrování replikace Enterprise Replication povoleno. Když replikace HDR a replikace Enterprise Replication pracují společně, sdílejí stejné konfigurační parametry ENCRYPT_CIPHER, ENCRYPT_MAC, ENCRYPT_MACFILE a ENCRYPT_SWITCH.

Další informace o těchto konfiguračních parametrech naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Úprava vyprázdnění LRU a automatické ladění v serverových párech HDR

Když je server nakonfigurován pro replikaci HDR, kontrolní body spuštěné sekundárním databázovým serverem nejsou blokovány. Tyto typy kontrolních bodů se vyskytují vzácně. Pokud je neblokovaný kontrolní bod spuštěný sekundárním serverem, transakce budou na primárním serveru zablokovány, aby se ověřilo, zda není ohrožena integrita sekundárního serveru. Pokud se neblokované kontrolní body spuštěné sekundárním serverem vyskytnou ve vašem systému, měli byste výrazněji optimalizovat vyprázdnění LRU na primárním serveru, abyste snížili blokování transakcí.

Chcete-li zvýšit vyprázdnění LRU, zmenšete hodnoty **lru_min_dirty** a **lru_max_dirty** v konfiguračním parametru BUFFERPOOL.

Automatické ladění LRU může být na každém uzlu HDR zapnuto nebo vypnuto nezávisle. Nastavení může být na každém databázovém serveru HDR odlišné. Informace o tom, jak vypnout automatické ladění LRU naleznete v části "Zapnutí a vypnutí automatického ladění LRU" na stránce 16-5.

Další informace o ladění LRU naleznete v příručce *Příručka výkonosti serveru IBM Informix Dynamic Server*.

Změna režimu databázového serveru

Chcete-li změnit režim databázového serveru, spusíte obslužný program **onmode** v příkazovém řádku nebo v programu ISA. Informace o příkazu **onmode** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Tabulka 20-2 shrnuje důsledky změny režimu primárního databázového serveru.

Tabulka 20-2. Změny režimu primárního databázového serveru

Na primárním serveru	Na sekundárním serveru	Restartování replikace HDR
Libovolný režim do režimu offline. (onmode -k)	Sekundární server zobrazí: DR: Chyba příjmu. Replikace HDR je vypnuta. Režim zůstane jen pro čtení. Je-li parametr DRAUTO nastaven na hodnotu 0 (OFF), zůstane server v režimu jen pro čtení. Je-li parametr DRAUTO nastaven na hodnotu 1 (RETAIN_TYPE), přepne se sekundární server do standardního režimu a začne přijímat aktualizace. (Pokud je parametr DRAUTO nastaven na hodnotu 2 (REVERSE_TYPE), stane se sekundární databázový server primárním databázovým serverem v okamžiku, jakmile bude ukončeno spojení z důvodu selhání původního primárního serveru.)	Postupujte stejně jako při selhání primárního serveru. Jsou možné dva scénáře, které závisejí na tom, co se stane se sekundární databází, zatímco je primární databázový server v režimu offline. Další informace naleznete v těchto částech: <ul style="list-style-type: none"> • “Sekundární databázový server nebyl změněn na standardní databázový server” na stránce 20-26 • “Sekundární databázový server je automaticky změněn na standardní databázový server” na stránce 20-27
Do režimu online, do klidového režimu nebo do režimu administrace. (onmode -s / onmode -u) (onmode -j)	Sekundární server nepřijímá chyby. Replikace HDR zůstane zapnutá. Režim zůstane jen pro čtení.	Použijte příkaz onmode -m na primárním serveru.

Tabulka 20-3 shrnuje důsledky změn režimu sekundárního databázového serveru.

Tabulka 20-3. Změny režimu sekundárního databázového serveru

Na sekundárním serveru	Na primárním serveru	Restartování replikace HDR
Jen pro čtení v režimu offline (onmode -k)	Primární server zobrazí: DR: Chyba příjmu. Replikace HDR je vypnuta.	Postupujte stejně jako při selhání sekundárního serveru. Postupujte podle pokynů v části “Restartování po selhání sekundárního databázového serveru” na stránce 20-26.

Poznámka: Jednouživatelský režim pracuje na sekundárním databázovém serveru replikace HDR stejným způsobem, jako na primárním databázovém serveru.

Změna typu databázového serveru

Typ primárního nebo sekundárního databázového serveru můžete změnit.

Změnit typ databázového serveru ze sekundárního na standardní můžete pouze tehdy, pokud je replikace HDR na sekundárním databázovém serveru vypnutá. Replikace HDR je vypnutá, pokud je přerušené připojení replikace dat k primárnímu databázovému serveru nebo dojde k selhání replikace dat na sekundárním databázovém serveru. Pokud standardní databázový server přepnete z režimu offline do režimu online, nepokusí se o připojení k druhému databázovému serveru v replikačním páru.

Chcete-li přepnout typ serveru, použijte následující příkazy:

- Skripty **hdrmksec.[sh|bat]** a **hdrmkpri.[sh|bat]**.

Přepnutí typu databázového serveru pomocí skriptů **hdrmkpri** a **hdrmksec**:

1. Vypněte primární databázový server (**ServerA**):
onmode -ky
2. Zatímco se sekundární databázový server (**ServerB**) nachází v režimu online, spusťte v systému UNIX skript **hdrmkpri.sh** nebo v systému Windows skript **hdrmkpri.bat**. **ServerB** je nyní primárním databázovým serverem.
3. Na serveru **ServerA** spusťte v systému UNIX skript **hdrmksec.sh** nebo v systému Windows skript **hdrmksec.bat**. **ServerA** je nyní sekundárním databázovým serverem.
4. Přepněte server **ServerB** (primární databázový server) do režimu online.

Monitorování stavu replikace HDR

Monitorování stavu replikace HDR databázového serveru umožňuje zjistit následující informace:

- Typ databázového serveru (primární, sekundární nebo standardní).
- Název druhého databázového serveru v páru.
- Zda je replikace HDR zapnutá.
- Hodnoty parametrů replikace HDR.

Použití obslužných programů příkazového řádku

Záhlaví zprávy, která se zobrazí při každém provedení příkazu **onstat**, obsahuje pole, které ukazuje, zda databázový server pracuje jako primární nebo sekundární databázový server.

Následující příklad znázorňuje záhlaví databázového serveru, který je primárním databázovým serverem v replikačním páru a je v režimu online:

```
IBM Informix Dynamic
Server Version 9.30.UC1    -- online(Prim) -- Up 45:08:57
```

Tento příklad znázorňuje databázový server, který je sekundárním databázovým serverem v replikačním páru a je v režimu jen pro čtení.

```
IBM Informix Dynamic Server Version 9.30.UC1    -- Read-Only (Sec) -- Up 45:08:57
```

Následující příklad znázorňuje záhlaví databázového serveru, který není zapojen do replikace HDR. Typ tohoto databázového serveru je standardní.

```
IBM Informix Dynamic Server Version 9.30.UC1    -- online -- Up 20:10:57
```

onstat -g dri

Pokud chcete monitorovat veškeré informace o replikaci HDR, spusťte volbu **onstat -g dri**. Zobrazí se následující pole:

- Typ databázového serveru (primární, sekundární nebo standardní).
- Stav replikace HDR (zapnuto nebo vypnuto).
- Párování databázový server.
- Poslední kontrolní bod replikace HDR.
- Hodnoty konfiguračních parametrů replikace HDR.

Příklad výstupu **onstat -g dri** naleznete v části *IBM Informix Administrator's Reference*.

oncheck -pr

Pokud je na databázovém serveru spuštěna replikace HDR, jsou informace o kontrolním bodu, který replikace HDR používá k synchronizaci primárního a sekundárního

databázového serveru, uloženy na rezervovaných stránkách PAGE_1ARCH a PAGE_2ARCH. Příklad odpovídajícího výstupu příkazu **oncheck -pr** ukazuje Obrázek 20-1.

```
Validating Informix Database Server reserved pages - PAGE_1ARCH &
PAGE_2ARCH
Using archive page PAGE_1ARCH.
```

```
Archive Level          0
Real Time Archive Began 01/11/95 16:54:07
Time Stamp Archive Began 11913
Logical Log Unique Id   3
Logical Log Position    b018

DR Ckpt Logical Log Id  3
DR Ckpt Logical Log Pos 80018
DR Last Logical Log Id  3
DR Last Logical Log Page 128
```

Obrázek 20-1. Výstup příkazu **oncheck -pr PAGE_1ARCH** pro databázový server se spuštěnou replikací HDR

Použití tabulek SMI

Tabulka **sysdri**, kterou popisuje kapitola týkající se databáze **sysmaster** v příručce *IBM Informix Administrator's Reference*, obsahuje následující sloupce.

sloupec	Popis
type	Typ serveru replikace HDR.
state	Stav serveru replikace HDR.
name	Název databázového serveru
intvl	Interval vyprazdňování vyrovnávací paměti replikace HDR.
timeout	Síťová prodleva.
lostfound	Název cesty ke ztraceným a nalezeným transakcím replikace HDR.

Použití programu ON-Monitor (systém UNIX)

Pomocí příkazu **Status > Replication** můžete zobrazit informace o replikaci HDR. Tato volba zobrazí stejné informace jako volba **onstat -g dri**.

Selhání replikace HDR

Tato část popisuje příčiny a následky selhání replikace HDR a volby, které může administrátor učinit v případě selhání a při restartování replikace dat.

Definice selhání replikace HDR

Selhání replikace HDR je ztráta spojení mezi databázovými servery v replikačním páru. Selhání replikace dat může způsobit jakákoli z následujících situací:

- Selhání v důsledku živelné pohromy (například požáru nebo rozsáhlého zemětřesení) v lokalitě jednoho z databázových serverů.
- Přerušování síťových kabelů, které spojují oba databázové servery.
- Přílišné zpoždění zpracování v jednom z databázových serverů.
- Selhání disku v sekundárním databázovém serveru, které se serveru nepodaří vyřešit pomocí zrcadleného bloku.

Rada: Selhání replikace *HDR* nemusí nezbytně znamenat, že selhal jeden z databázových serverů, znamená to pouze, že došlo ke ztrátě připojení replikace *HDR* mezi oběma databázovými servery.

Zjišťování selhání replikace HDR

Databázový server jako selhání replikace HDR interpretuje následující stavy:

- Byla překročena určená prodleva.

V běžném provozu replikace HDR databázový server očekává, že druhý databázový server v páru bude komunikaci potvrzovat. V obou databázových serverech určuje parametr `DRTIMEOUT` souboru `ONCONFIG` počet sekund. Pokud databázový server neobdrží od druhého databázového serveru v páru potvrzení v čase určeném parametrem `DRTIMEOUT`, bude databázový server předpokládat, že došlo k selhání replikace HDR.

- Druhý databázový server v páru primárního a sekundárního serveru neodpovídá na pokusy o periodické zasílání zprávy (*příkaz ping*) prostřednictvím sítě.

Databázový server odesílá pakety příkazu ping nezávisle na tom, zda primární databázový server odesílá či neodesílá záznamy do sekundárního databázového serveru. Pokud jeden z databázových serverů v páru neodpoví na čtyři příkazy ping v řadě, bude druhý databázový server předpokládat, že došlo k selhání replikace HDR.

Každý databázový server v páru odešle do druhého databázového serveru v páru paket příkazu ping, jakmile uplyne čas určený v sekundách parametrem `DRTIMEOUT` databázového serveru.

Akce, které je třeba provést při selhání replikace HDR

Jakmile databázový server zjistí selhání replikace HDR, zapíše zprávu do protokolu zpráv (například zprávu `DR: receive error`) a vypne replikaci. Pokud dojde k selhání replikace HDR, bude připojení replikace HDR mezi dvěma databázovými servery ukončeno a sekundární databázový server zůstane v režimu pouze pro čtení.

Pokud sekundární databázový server zůstane po selhání replikace HDR v režimu online a konfigurační parametr `DRAUTO` je nastaven na hodnotu 1(`RETAIN_TYPE`), změní se typ databázového serveru automaticky na standardní. Pokud je parametr `DRAUTO` nastaven na hodnotu 0 (vypnuto), bude se sekundární databázový server automaticky pokoušet obnovit komunikaci s primárním databázovým serverem. Pokud je parametr `DRAUTO` nastaven na hodnotu 2 (`REVERSE_TYPE`), stane se sekundární databázový server primárním databázovým serverem v okamžiku přerušení připojení po selhání primárního databázového serveru, ne po restartování původního primárního serveru.

Co provést v případě selhání replikace HDR

V případě selhání replikace HDR zvažte následující záležitosti:

- Jak by na selhání měli reagovat klienti.

Pokud dojde ke skutečnému selhání (ne k pouhému dočasnému zpomalení nebo selhání sítě), budete pravděpodobně chtít *přesměrovat* klienty používající databázový server, který selhal, na druhý databázový server v páru. Pokyny, jak přesměrovat klienty, naleznete v části “Přesměrování a propojitelnost pro klienty replikace dat” na stránce 19-23.

- Jak by na selhání měly reagovat databázové servery.

Administrativní akce, které je třeba provést po selhání replikace HDR, závisí na tom, zda selhal primární nebo sekundární databázový server. Další informace o tomto tématu naleznete v části “Postup v případě selhání sekundárního databázového serveru” na stránce 20-20 a v části “Postup v případě selhání primárního databázového serveru” na stránce 20-20.

Pokud přeměrujete klienty, zvažte, jaké zatížení způsobí dodateční klienti ve zbývajícím databázovém serveru. Je možné, že bude zapotřebí zvětšit prostor vyhrazený logickému protokolu nebo častěji zálohovat soubory logického protokolu.

Postup v případě selhání sekundárního databázového serveru

Pokud dojde k selhání sekundárního databázového serveru, zůstane primární databázový server v režimu online.

K přeměrování klientů používajících sekundární databázový server na primární databázový server použijte jakoukoli z metod popsaných v části “Přeměrování a propojitelnost pro klienty replikace dat” na stránce 19-23. Pokud přeměrujete tyto klienty, bude primární databázový server pravděpodobně potřebovat další dočasný prostor dbspace k ukládání dočasných tabulek a k řazení.

Typ databázového serveru není třeba měnit z primárního na standardní.

Restartování replikace dat po selhání sekundárního databázového serveru proveďte podle kroků popsaných v části “Restartování po selhání sekundárního databázového serveru” na stránce 20-26.

Postup v případě selhání primárního databázového serveru

Pokud dojde k selhání primárního databázového serveru, může sekundární databázový server pracovat následujícími způsoby:

- Sekundární databázový server může zůstat v režimu logické obnovy. V takovém případě se neprovádí žádná akce. To je žádoucí, pokud očekáváte, že bude připojení replikace HDR obnoveno velmi brzy.
- Databázový server může přejít do režimu standardního databázového serveru automaticky. Tato akce se nazývá *automatické přepnutí*.
- Sekundární databázový server může přejít do režimu standardního databázového serveru, pokud změníte režim databázového serveru na standardní pomocí *ručního přepnutí*.

Automatické přepnutí: Automatické přepnutí znamená, že sekundární databázový server automaticky přejde do režimu standardního databázového serveru, jakmile zjistí selhání replikace HDR. Nejprve odvolá všechny otevřené transakce a pak přejde do režimu online jako standardní databázový server. K automatickému přepnutí dochází pouze tehdy, pokud je parametr DRAUTO v souboru ONCONFIG nastaven na hodnotu 1 (RETAIN_TYPE) nebo 2 (REVERSE_TYPE).

Protože se ze standardního databázového serveru stane v takovém případě standardní databázový server, zajistěte, aby byla splněna jedna z následujících podmínek:

- V sekundárním databázovém serveru je dostatek diskového prostoru pro logický protokol, aby mohl server pokračovat ve zpracování i bez zálohování souborů logického protokolu.
- Logické protokoly budou zálohovány.

Automatické přepnutí změní pouze typ databázového serveru. Nepřesměruje však klientské aplikace na sekundární databázový server. Informace o tom, jak přeměrovat klienty, naleznete v části “Přeměrování a propojitelnost pro klienty replikace dat” na stránce 19-23.

Automatické přepnutí má v porovnání s ručním přepnutím následující výhody:

- Klienti přeměrovaní z primárního databázového serveru na sekundární databázový server mohou pokračovat v zapisování a aktualizaci dat.
- Přepnutí není závislé na tom, zda operátor monitorující protokol zpráv zaznamená, že došlo k selhání replikace HDR a ručně přepne sekundární databázový server na standardní.

Hlavní nevýhodou automatického přepnutí je, že k tomu, aby pracovalo správně, vyžaduje velmi stabilní síť. Další informace naleznete v části “Používání automatického přepnutí v nespolehlivé síti”.

V části “Sekundární databázový server je automaticky změněn na standardní databázový server” na stránce 20-27 naleznete kroky nezbytné k restartování replikace dat po automatickém přepnutí.

Akce, ke kterým dojde po automatickém přepnutí: Jakmile se vám podaří opět převést primární databázový server do režimu online, bude automaticky vytvořeno připojení replikace HDR.

- Pokud je parametr DRAUTO nastaven na hodnotu RETAIN_TYPE, bude standardní databázový server (který byl původně sekundárním serverem) nenásilně vypnut, aby mohl odpojit všechny klienty, kteří by mohli do databáze zapisovat, a pak přejde zpět do režimu sekundárního databázového serveru.
- Pokud je parametr DRAUTO nastaven na hodnotu REVERSE_TYPE, přejde standardní databázový server (který byl původně sekundárním databázovým serverem) přímo do primárního režimu. Nedojde k jeho vypnutí. Všechny aplikace připojené k databázovému serveru mohou zůstat připojené. Původní primární databázový server přejde do režimu sekundárního databázového serveru.

Používání automatického přepnutí v nespolehlivé síti: Přestože se automatické přepnutí může zdát nejlepším řešením, nemusí být jeho použití vhodné ve všech prostředích.

Zvažte například situaci, ve které by primární databázový server ve skutečnosti neselhal, ale sekundárnímu databázovému serveru by se jevilo, že k selhání došlo. Pokud například sekundární databázový server nepřijímá signály (pakety příkazu ping) primárního databázového serveru, protože síť je pomalá nebo nestabilní, bude sekundární databázový server předpokládat, že došlo k selhání primárního databázového serveru a automaticky přejde do standardního režimu. Pokud primární databázový server nebude přijímat odpovědi na signály zasílané sekundárnímu databázovému serveru, bude předpokládat, že došlo k selhání sekundárního databázového serveru a vypne replikaci dat, zůstane však v režimu online. Nyní se budou primární databázový server i sekundární databázový server (přepnutý do standardního režimu) nacházet v režimu online.

Pokud klienti mohou aktualizovat data v obou databázových serverech nezávisle, dosáhnou pak oba databázové servery stavu, kdy každý z těchto databázových serverů bude potřebovat záznamy logického protokolu uložené v druhém databázovém serveru. V takovém případě je třeba začít od začátku a provést počáteční replikaci dat s použitím zálohy úrovně 0 prostorů dbspace celého databázového serveru, jak popisuje část “První spuštění replikace HDR” na stránce 20-6. Proto pravděpodobně nebudete chtít používat automatické přepnutí, pokud síť není zcela stabilní. Replikace HDR nemůže být na předchozím sekundárním serveru znovu obnovena bez rizika ztráty transakcí.

Ruční přepnutí: Ručním přepnutím se rozumí změna typu sekundárního databázového serveru na standardní administrátorem sekundárního databázového serveru. Sekundární databázový server odvolá všechny otevřené transakce a pak přejde do režimu online jako standardní databázový server, aby mohl přijímat aktualizace od klientských aplikací. Vysvětlení, jak provést přepnutí naleznete v části “Změna typu databázového serveru” na stránce 20-16.

Restartování po ručním přepnutí: Seznam kroků, které je třeba provést k restartování replikace dat po ručním přepnutí naleznete v příručce “Sekundární databázový server byl změněn na standardní databázový server” na stránce 20-26.

Restartování v případě, pokud sekundární databázový server není přepnut na standardní: Pokud sekundární databázový server není přepnut na standardní, proveďte kroky popsané v části “Sekundární databázový server nebyl změněn na standardní databázový server” na stránce 20-26.

Obnovení dat po selhání prostředku

Výsledek selhání disku závisí na tom, zda k němu došlo na primárním nebo sekundárním databázovém serveru, zda bloky na disku obsahují kritické prostředky (kořenový prostor dbspace, soubor logického nebo fyzického protokolu) a zda jsou bloky zrcadleny.

Obnovení po selhání prostředku na primárním databázovém serveru

Tabulka 20-4 na stránce 20-22 uvádí přehled různých scénářů pro obnovení dat v případě selhání prostředku na primárním databázovém serveru. Podstatné jsou následující skutečnosti:

- Pokud jsou bloky zrcadleny, můžete obnovu provést stejným způsobem, jako v případě standardního databázového serveru, který používá zrcadlení.
- Pokud bloky nejsou zrcadleny, závisí postup při obnově primárního databázového serveru na tom, zda vadný disk obsahuje kritické prostředky.

Pokud ano, primární databázový server selže. Musíte provést úplné obnovení pomocí záloh primárních prostorů dbspace (nebo záloh sekundárních prostorů dbspace, pokud byl sekundární databázový server přepnut do standardního režimu a aktivita byla přesměrována. Další informace naleznete v části “Restartování po poškození kritických dat” na stránce 20-24.

Pokud disk neobsahuje kritické prostředky, můžete dotčené prostory obnovit jednotlivě pomocí teplého obnovení. Teplé obnovení se skládá ze dvou částí: nejprve je ze zálohy obnoven vadný prostor dbspace a potom je provedeno logické obnovení všech záznamů logického protokolu, které byly zapsány od zálohování tohoto prostoru dbspace. Další informace o provádění teplého obnovení naleznete v příručce *IBM Informix Backup and Restore Guide*. Před provedením teplého obnovení musíte zálohovat všechny soubory logického protokolu.

Tabulka 20-4. Scénáře pro selhání prostředků na primárním databázovém serveru.

replikace HDR	Důležité prostředky	bloky zrcadleny	Dopad selhání a postup při obnovení prostředků
replikace HDR			
Primární	Ano	Ne	Primární databázový server selže. Postupujte podle pokynů v části “Restartování po poškození kritických dat” na stránce 20-24.
Primární	Ano	Ano	Primární databázový server zůstane v režimu online. Postupujte podle pokynů v části “Obnovení zrcadleného bloku” na stránce 18-6.
Primární	Ne	Ne	Primární databázový server zůstane v režimu online. Proveďte teplé obnovení prostorů dbspace ze zálohy prostorů dbspace podle pokynů v příručce pro zálohování a obnovení serveru Dynamic Server. Před provedením teplého obnovení zálohujte všechny soubory logického protokolu.
Primární	Ne	Ano	Primární databázový server zůstane v režimu online. Postupujte podle pokynů v části “Obnovení zrcadleného bloku” na stránce 18-6.

Obnovení po selhání prostředku na sekundárním databázovém serveru

Replikace HDR: Tabulka 20-5 uvádí přehled různých scénářů pro obnovení dat v případě selhání prostředku na sekundárním databázovém serveru. Podstatné jsou následující skutečnosti:

- Pokud jsou bloky zrcadleny, můžete obnovu provést úplně stejně jako pro standardní databázový server, který používá zrcadlení.
- Pokud bloky nejsou zrcadleny, pak sekundární databázový server selže, pokud disk obsahuje kritické prostředky, ale zůstane v režimu online, pokud disk kritické prostředky neobsahuje. V obou případech musíte provést úplné obnovení pomocí záloh prostorů dbspace na primárním databázovém serveru. (Viz “Restartování po poškození kritických dat” na stránce 20-24.) V druhém případě nelze vybrané prostory dbspace obnovit ze zálohy sekundárních prostorů dbspace, protože se v danou chvíli již mohou lišit od odpovídajících prostorů dbspace na primárním databázovém serveru. Musíte provést úplné obnovení.

Tabulka 20-5. Scénáře pro selhání prostředků na sekundárním databázovém serveru

replikace HDR	Důležité prostředky	Bloky jsou zrcadleny	Dopad selhání
Sekundární	Ano	Ne	Sekundární databázový server selže. Primární databázový server přijme chybové zprávy. Replikace HDR je vypnuta. Postupujte podle pokynů v části “Restartování po poškození kritických dat” na stránce 20-24.
Sekundární	Ano	Ano	Sekundární databázový server zůstane online v režimu jen pro čtení. Postupujte podle pokynů v části “Obnovení zrcadleného bloku” na stránce 18-6.
Sekundární	Ne	Ne	Sekundární databázový server zůstane online v režimu jen pro čtení. Primární databázový server přijme chybové zprávy. Replikace HDR je vypnuta. Postupujte podle pokynů v části “Restartování po poškození kritických dat” na stránce 20-24.
Sekundární	Ne	Ano	Sekundární databázový server zůstane online v režimu jen pro čtení. Postupujte podle pokynů v části “Obnovení zrcadleného bloku” na stránce 18-6.

Replikace indexu na sekundární server

Poruší-li se index na sekundárním databázovém serveru replikace HDR, je obvyklým postupem při opravě tohoto problému vypustit index na primárním databázovém serveru a potom ho znovu vytvořit. Tento proces vyžaduje uzamčení tabulky a jeho dokončení může trvat poměrně dlouho. Namísto toho můžete index replikovat z primárního databázového serveru na sekundární databázový server, aniž by se na primárním databázovém serveru znovu vytvářel.

Ruční spuštění replikace indexu lze provést pomocí příkazu **onmode -d**.

Chcete-li nastavit automatickou replikaci indexu v případě, že je index na sekundárním databázovém serveru poškozen, nastavte konfigurační parametr DRIDXAUTO. Hodnotu konfiguračního parametru DRIDXAUTO můžete pro tuto relaci aktualizovat pomocí příkazu **onmode -d idxauto**.

Restartování replikace HDR po selhání

Informace o různých typech selhání replikace HDR naleznete v části “Definice selhání replikace HDR” na stránce 20-18. Postup při restartování replikace HDR závisí na tom, zda kritická data na jednom z databázových serverů byla, či nebyla poškozena. V této části jsou probírány oba případy.

Restartování po poškození kritických dat

Pokud se na jednom z databázových serverů vyskytne porucha, která poškodí kořenový prostor dbspace, prostor dbspace obsahující soubory logického protokolu nebo prostor dbspace obsahující fyzický protokol, musíte s poškozeným databázovým serverem pracovat tak, jako by na discích neměl žádná data a spouštěli jste replikaci HDR poprvé. Použijte funkční databázový server s nedotčenými disky jako databázový server s daty.

Selhání kritického prostředku na primárním databázovém serveru

Po selhání kritického prostředku na primárním databázovém serveru byste měli restartovat replikaci HDR. Tabulka 20-6 uvádí příkazy požadované pro provedení tohoto postupu.

Restartování replikace HDR po selhání kritického prostředku:

1. Pokud byl původní sekundární databázový server změněn na standardní databázový server, převedte tento databázový server do klidového režimu (DRAUTO = 0) a potom změňte typ zpět na sekundární pomocí příkazu **onmode -d**.

Je-li DRAUTO = 1 (RETAIN_TYPE), nelze tento krok použít. Databázový server automaticky provede nenásilné vypnutí a přepne se zpět na sekundární typ, až převedete primární databázový server zpět do režimu online.

Je-li DRAUTO = 2 (REVERSE_TYPE), stane se sekundární databázový server primárním databázovým serverem, jakmile bude spojení ukončeno selháním původního primárního databázového serveru, ne restartováním původního primárního databázového serveru.

2. Obnovte primární databázový server z poslední zálohy prostorů dbspace.
3. Provedením příkazu **onmode -d** nastavte typ primárního databázového serveru a spusťte replikaci HDR.

Příkaz **onmode -d** spustí logickou obnovu primárního databázového serveru ze souborů logického protokolu na disku sekundárního databázového serveru. Nelze-li logickou obnovu dokončit proto, že jste zálohovali a uvolnili soubory logického protokolu na původní sekundární databázový server, replikace HDR nebude spuštěna, dokud neprovedete krok 4.

4. Použijte soubory logického protokolu ze sekundárního databázového serveru, které byly zálohovány na pásku, na primární databázový server.

Je-li tento krok vyžadován, odešle primární databázový server zprávu, která vyzve k obnově souborů logického protokolu z pásky. Tato zpráva je uložena v protokolu zpráv. Po obnově všech požadovaných souborů logického protokolu z pásky budou obnoveny všechny zbývající soubory logického protokolu na sekundárním disku.

Tabulka 20-6. Postup při restartování replikace HDR po selhání kritického prostředku na primárním databázovém serveru

Krok	Na primárním databázovém serveru	Na sekundárním databázovém serveru
1.		Obslužný program onmode onmode -s onmode -d secondary <i>název_prim</i>

Tabulka 20-6. Postup při restartování replikace HDR po selhání kritického prostředku na primárním databázovém serveru (pokračování)

Krok	Na primárním databázovém serveru	Na sekundárním databázovém serveru
2.	<p>ON-Bar Příkaz</p> <p>onbar -r -p</p> <p>Příkaz ontape</p> <p>ontape -p</p>	
3.	<p>Obslužný program onmode</p> <p>onmode -d primary <i>název_sek</i></p>	
4.	<p>Příkaz ontape</p> <p>ontape -l</p>	

Selhání kritického prostředku na sekundárním databázovém serveru

Dojde-li na sekundárním databázovém serveru k selhání kritického prostředku, můžete postupovat podle pokynů v části “První spuštění replikace HDR” na stránce 20-6.

Selhání kritického prostředku na obou databázových serverech

Pokud na obou počítačích, ve kterých jsou spuštěny databázové servery v replikačním páru, dojde k selhání, která poškodí kořenový prostor dbspace, prostor dbspace obsahující soubory logického protokolu nebo fyzický protokol, musíte restartovat replikaci HDR.

Restartování replikace HDR po selhání kritického prostředku na obou databázových serverech:

1. Obnovte primární databázový server ze zálohy paměťového prostoru a logického protokolu.
2. Po obnovení primárního databázového serveru musíte s druhým poškozeným databázovým serverem pracovat tak, jako by na discích neměl žádná data a spouštěli jste replikaci HDR poprvé.

(Viz “První spuštění replikace HDR” na stránce 20-6.) Použijte funkční databázový server s nedotčenými disky jako databázový server s daty.

Restartování bez poškození kritických dat

Nejsou-li kritická data na žádném z obou databázových serverů poškozena, jsou možné následující čtyři scénáře. Každý z nich vyžaduje jiný postup restartování replikace HDR:

- Dojde k chybě sítě.
- Selže sekundární databázový server.
- Selže primární databázový server a sekundární databázový server není změněn na standardní databázový server.
- Selže primární databázový server a sekundární databázový server je změněn na standardní databázový server.

Restartování po chybě sítě

Po chybě sítě je primární databázový server v režimu online a sekundární databázový server je v režimu jen pro čtení. Replikace HDR je na obou databázových serverech vypnuta (state = off). Jakmile bude spojení znovu navázáno, můžete replikaci HDR restartovat příkazem **onmode -d secondary *název_prim*** na sekundárním databázovém serveru. Restartování replikace HDR by nemuselo být nutné, protože se primární databázový server pokouší

každých 10 s znovu připojit a každé 2 minuty zobrazuje zprávu, že se nemůže připojit. Nemusíte restartovat spojení pomocí obslužného programu onmode.

Restartování po selhání sekundárního databázového serveru

Potřebujete-li restartovat replikaci HDR po selhání sekundárního databázového serveru, postupujte podle pokynů, které uvádí Tabulka 20-7. Tento postup předpokládá, že jste od selhání sekundárního databázového serveru zálohovali soubory logického protokolu na primárním databázovém serveru podle potřeby.

Tabulka 20-7. Postup při restartování po selhání sekundárního databázového serveru

Krok	Na primárním serveru	Na sekundárním serveru
1.	Primární databázový server by měl být v režimu online.	oninit Zobrazí-li se v protokolu zpráv následující zpráva, pokračujte krokem 2: DR: Zahájení obnovy z pásky po selhání
2.		Příkaz ontape ontape -l ON-Bar příkaz onbar -r -l

Restartování po selhání primárního databázového serveru

Následující části popisují restartování replikace HDR, pokud za různých okolností selže primární databázový server.

Sekundární databázový server nebyl změněn na standardní databázový server:

Potřebujete-li restartovat replikaci HDR po selhání primárního databázového serveru v případě, že sekundární databázový server nebyl změněn na standardní databázový server, stačí převést primární databázový server zpět do režimu online pomocí příkazu **oninit**.

Sekundární databázový server byl změněn na standardní databázový server:

Potřebujete-li restartovat replikaci HDR po selhání primárního databázového serveru a změnili jste sekundární databázový server na standardní databázový server, postupujte podle pokynů, které uvádí Tabulka 20-9.

Tabulka 20-8. Postup restartování po změně sekundárního databázového serveru na standardní

Krok	Na primárním databázovém serveru	Na sekundárním databázovém serveru
1.		onmode -s Tento krok převádí sekundární databázový server (nyní standardní) do klidového režimu. Všichni klienti, kteří jsou připojeni k tomuto databázovému serveru, se musejí odpojit. Aplikace, které provádějí aktualizace, musejí být přeměrovány na primární databázový server. Další informace naleznete v části "Přeměrování a propojitelnost pro klienty replikace dat" na stránce 19-23.
2.		onmode -d secondary <i>název_prim</i>

Tabulka 20-8. Postup restartování po změně sekundárního databázového serveru na standardní (pokračování)

Krok	Na primárním databázovém serveru	Na sekundárním databázovém serveru
3.	<p>oninit</p> <p>Pokud všechny záznamy logického protokolu, které byly zapsány na sekundární databázový server, jsou stále na disku sekundárního databázového serveru, pak primární databázový server obnoví tyto záznamy z tohoto disku, zadáte-li příkaz oninit.</p> <p>Pokud jste záložovali a uvolnili soubory logického protokolu na sekundárním databázovém serveru, záznamy v těchto souborech již na disku nejsou. V takovém případě budete vyzváni, abyste tyto soubory logického protokolu obnovili z pásky (krok 4).</p> <p>Pro uživatele příkazu ontape:</p> <p>Chcete-li číst záznamy logického protokolu přes síť, nastavte páskové zařízení logického protokolu na zařízení v počítači, ve kterém je spuštěn sekundární databázový server.</p>	
4.	<p>Až budete vyzváni k obnově záznamů logického protokolu z pásky, proveďte tento krok.</p> <p>Příkaz ontape</p> <p>ontape -l</p> <p>ON-Bar příkaz</p> <p>onbar -r -l</p>	

Sekundární databázový server je automaticky změněn na standardní databázový

server: Potřebujete-li restartovat replikaci HDR po selhání primárního databázového serveru a sekundární databázový server byl automaticky změněn na standardní databázový server (podle popisu v části “Automatické přepnutí” na stránce 20-20), postupujte podle pokynů v následující tabulce.

Tabulka 20-9. Postup restartování po automatické změně sekundárního databázového serveru na standardní

Krok	Na primárním databázovém serveru	Na sekundárním databázovém serveru
1.	<p>% oninit</p> <p>Je-li DRAUTO = 1, bude typ tohoto databázového serveru nastaven na primární.</p> <p>Je-li DRAUTO = 2, bude typ tohoto databázového serveru při restartování nastaven na sekundární.</p> <p>Pokud všechny záznamy logického protokolu, které byly zapsány na sekundární databázový server, jsou stále na disku sekundárního databázového serveru, pak primární databázový server obnoví tyto záznamy z tohoto disku, zadáte-li příkaz oninit.</p> <p>Pokud jsou soubory logického protokolu, které jste záložovali a uvolnili, na sekundárním databázovém serveru, záznamy v těchto souborech již nejsou na disku. V takovém případě budete vyzváni, abyste tyto soubory logického protokolu obnovili z pásky (krok 2).</p> <p>Pro uživatele příkazu ontape:</p> <ul style="list-style-type: none"> Nastavte páskové zařízení logického protokolu na zařízení v počítači, ve kterém je spuštěn sekundární databázový server. 	<p>Je-li DRAUTO = 1, sekundární databázový server bude automaticky nenásilně vypnut, až primární databázový server znovu zapnete. Tím bude zajištěno odpojení všech klientů. Typ bude potom přepnut na sekundární. Všechny aplikace, které provádějí aktualizace, musejí být přesměrovány zpět na primární databázový server. Další informace naleznete v části “Přesměrování a propojitelnost pro klienty replikace dat” na stránce 19-23.</p> <p>Je-li DRAUTO = 2, přepne se sekundární databázový server na primární a potom automaticky na standardní. Původní primární databázový server se po restartování a připojení k dalšímu serveru stane sekundárním databázovým serverem a určí, že se tento server stane primárním databázovým serverem.</p>

Tabulka 20-9. Postup restartování po automatické změně sekundárního databázového serveru na standardní (pokračování)

Krok	Na primárním databázovém serveru	Na sekundárním databázovém serveru
2.	Až budete vyzváni k obnovení záznamů logického protokolu z pásky, proveďte tento krok. Příkaz ontape % ontape -l ON-Bar příkaz onbar -r -l	

Kapitola 21. Použití serverů RS (remote standalone) (vydání Enterprise)

Obsah kapitoly	21-1
Sekundární server RS	21-1
Rozdíly mezi sekundárním serverem RS a replikací HDR	21-2
Kdy používat sekundární Server RS	21-2
Charakteristika a omezení sekundárních serverů RS	21-3
Protokolování indexových stránek	21-3
Způsob práce protokolování indexových stránek	21-3
Povolení nebo zakázání protokolování indexových stránek	21-3
Zobrazení statistiky protokolování indexových stránek	21-3
Připojení rozhraní SMX (Server Multiplexer Group)	21-4
Povolení šifrování SMX	21-4
Získání statistiky SMX.	21-4
Konfigurace sekundárního serveru RS	21-4
Hardwarové a softwarové požadavky	21-4
První spuštění sekundárního serveru RS.	21-4
Získání statistiky sekundárního serveru RS	21-7
Konfigurace sekundárního serveru RS	21-7
Přeměna sekundárního serveru RS na standardní server	21-7
Přeměna sekundárního serveru RS na sekundární server replikace HDR	21-7
Opětovné zřízení sekundárního serveru replikace HDR jako sekundárního serveru RS	21-7
Opětovné vrácení softwaru v prostředí serveru RS	21-7
Odstranění sekundárního serveru RS.	21-8
Obnovení při havárii v prostředí sekundárního serveru RS	21-8
Zabezpečení sekundárního serveru RS	21-8
Vytvoření nebo změna hesla na sekundárním serveru RS.	21-8
Návrh klientů sekundárních serverů RS	21-8
Návrh klientů používajících sekundární Server RS	21-8
Monitorování alarmů událostí sekundárního serveru RS	21-8

Obsah kapitoly

Tato kapitola poskytuje přehled nastavení a konfigurace sekundárních serverů RS (remote standalone) ve vysoce dostupném prostředí.

V této kapitole jsou popsána následující témata:

- Kdy používat sekundární servery RS
- Výhody sekundárních serverů RS
- Způsob replikace dat na sekundární servery RS
- Zabezpečení sekundárního serveru RS
- Získání statistiky sekundárního serveru RS

Sekundární server RS

Sekundární server RS (remote standalone) je součástí vysoce dostupné konfigurace. Sekundární servery RS mohou být geograficky vzdálené od primárního serveru a v případech obnovení při havárii mohou sloužit jako vzdálené záložní servery. Každý sekundární server RS uchovává úplnou kopii databáze, aktualizace jsou přenášeny asynchronně z primárního serveru prostřednictvím zabezpečeného síťového připojení.

Sekundární server RS poskytuje za účelem zvýšení dostupnosti možnost konfigurace více sekundárních databázových serverů. Pokud potřebujete více zálohovacích míst, můžete například nakonfigurovat několik sekundárních serverů RS jako další servery do páru replikace HDR. Zatímco základní prostředí replikace HDR poskytuje jeden primární a jeden sekundární uzel, umožňuje sekundární server RS rozšířit toto prostředí o podporu několika sekundárních uzlů v režimu pouze pro čtení.

Rozdíly mezi sekundárním serverem RS a replikací HDR

Sekundární server RS je v mnoha ohledech podobný sekundárnímu serveru HDR. Na sekundární serveru RS se protokoly odesílají téměř stejným způsobem jako odesílá primární server protokoly na sekundární server replikace HDR serveru. Sekundární server RS je však navržen pouze pro asynchronní komunikační rozhraní, aby byl minimalizován jeho vliv na primární server. Z toho důvodu nejsou mezi primárním serverem a sekundárními servery RS ani potvrzení transakce ani kontrolní body. Jinými slovy není u žádné transakce potvrzené na primárním serveru garantováno, že bude ve stejnou chvíli potvrzena i na sekundárním serveru RS. Protože se servery RS aktualizují asynchronně, nelze sekundární server RS přímo přeměnit na primární server. Místo toho je možné jej změnit na sekundární server replikace HDR, který lze potom přeměnit na primární server. Také je možné sekundární replikaci HDR přeměnit zpět na sekundární server RS.

I když je sekundární server RS podobný sekundární replikaci HDR, existují některé funkce, které sekundární replikace HDR podporuje a sekundární server RS nepodporuje, například:

- Sekundární server RS nepodporuje režim SYNC
- Sekundární server RS nepodporuje DRAUTO.
- Sekundární server RS nemá synchronizované kontrolní body.
- Sekundární server RS nelze přímo převést na primární server.

Kdy používat sekundární Server RS

Při konfiguraci prostředí sekundárního serveru RS byste měli vzít v úvahu následující důvody:

- Zvýšená dostupnost
Jediný server obvykle k zajištění dostupnosti nestačí. Jeden nebo více sekundárních serverů RS poskytuje další zajištění díky existenci několika serverů, které lze použít ke zvýšení dostupnosti.
- Geograficky vzdálená podpora zásobníku
Pokud je sekundární server umístěn v síti WAN, může vysoká úroveň koordinace mezi primárním a sekundárním párem replikace HDR způsobit problémy s výkonem. Primární a sekundární servery je často vhodnější umístit relativně blízko sebe jednak kvůli zjednodušení údržby a také kvůli zachování výkonu. Mimo to je často vhodné umístit do určité vzdálenosti sekundární server, který lze použít jako "záložní zásobník" při nejhorších scénářích obnovení při havárii. Sekundární server RS je ideálním řešením záložního zásobníku.
- Zvýšený výkon přenosu dat
S podporou více sekundárních serverových uzlů je možné převést funkčnost přenosu dat na jiné servery, aniž by byl ovlivněn primární server. Konfigurace sekundárního serveru RS navíc umožňuje snadno izolovat požadavky na přenos dat od požadavků na vysokou dostupnost, což vede v obou prostředích k lepšímu řešení.
- Dostupnost v nestabilních sítích
V některých případech může prostředí pomalé nebo nestabilní sítě při synchronním provádění kontrolních bodů způsobovat zpoždění primárního i sekundárního serveru. Konfigurace sekundárního serveru RS jsou určeny k tomu, aby používaly plně duplexní síťovou komunikaci a nevyžadovaly takovou koordinaci. Sekundární server RS může

poskytovat atraktivní řešení v situacích, kdy není síťové prostředí mezi primárním serverem a sekundárním serverem RS optimální.

Charakteristika a omezení sekundárních serverů RS

Protože sekundární server RS uchovává úplnou kopii fyzické databáze, musí mít stejnou fyzickou paměť jako zdrojový uzel.

Sekundární server RS replikuje všechny vestavěné i rozšířené datové typy. Uživatelské datové typy (UDT) musejí být protokolovány a uloženy v jediném databázovém serveru. Datové typy s daty ukládanými mimo řádek jsou replikovány tehdy, pokud jsou jejich data uložena v prostoru sbspase nebo v jiné tabulce téhož databázového serveru. Aby mohla být replikována data prostoru sbspase, musí být prostor sbspase protokolován.

Sekundární server RS nereplikuje data uložená v souborech operačního systému ani trvalé externí soubory nebo paměťové objekty přidružené k uživatelským rutinám.

Uživatelské datové typy, uživatelské rutiny a moduly DataBlade vyžadují zvláštní postup instalace a registrace. Pokyny naleznete v části “Průběh prvotní replikace dat” na stránce 19-6.

Protokolování indexových stránek

Z důvodu synchronizace vytvoření indexu mezi servery ve vysoce dostupných prostředích při vytvoření indexu zapíše protokolování indexových stránek stránky do logického protokolu. Pro použití sekundárního serveru RS musí být povoleno protokolování indexových stránek.

Způsob práce protokolování indexových stránek

Protokolování indexových stránek zapíše úplný index do souboru protokolu, který je potom asynchronně přenesen na sekundární server. Sekundární server může být buď sekundární server RS, nebo sekundární server replikace HDR. Potom jsou transakce ze souboru protokolu načteny do databáze na sekundárním serveru, a tím umožní, aby sekundární server nemusel při obnově znovu vytvářet index. V případě sekundárních serverů RS nečeká primární server na potvrzení od sekundárního serveru, což umožňuje okamžitý přístup k indexu na primárním serveru.

Protokolování indexových stránek je řízeno parametrem onconfig LOG_INDEX_BUILDS. Pokud je parametr LOG_INDEX_BUILDS nastaven na hodnotu 1 (povoleno), jsou indexy vytvořeny na primárním serveru a potom odeslány na sekundární server.

Povolení nebo zakázání protokolování indexových stránek

Konfigurační parametr LOG_INDEX_BUILDS použijte k povolení nebo zakázání protokolování indexových stránek při spuštění databázového serveru. Hodnotu parametru LOG_INDEX_BUILDS můžete změnit v souboru onconfig spuštěním příkazu **onmode -wf LOG_INDEX_BUILDS=1** (povolit) nebo **0** (zakázat).

Pokud ve vysoce dostupném prostředí existuje sekundární server RS, musí být povoleno protokolování indexových stránek a nelze je zakázat.

Zobrazení statistiky protokolování indexových stránek

Zda je protokolování indexových stránek povoleno nebo zakázáno můžete zobrazit pomocí nástroje onstat nebo pomocí tabulek rozhraní SMI (system-monitoring interface). Statistika zobrazuje také datum a čas, kdy bylo protokolování indexových stránek povoleno nebo zakázáno.

Chcete-li zobrazit statistiku protokolování indexových stránek, použijte příkaz **onstat -g ipl**. Můžete také použít dotaz na tabulku **sysipl**.

Další informace o výstupu příkazu **onstat -g ipl** naleznete v kapitole týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

Připojení rozhraní SMX (Server Multiplexer Group)

Rozhraní SMX (Server Multiplexer Group) je komunikační rozhraní, které podporuje šifrovaná síťová připojení mezi servery ve vysoce dostupných prostředích. Rozhraní SMX poskytuje zabezpečený vysoce výkonný komunikační mechanismus mezi instancemi databázového serveru.

Povolení šifrování SMX

Ve vysoce dostupných konfiguracích nastavuje úroveň šifrování komunikační parametr **ENCRYPT_SMX**. Pokud je parametr **ENCRYPT_SMX** nastaven na hodnotu 1, použije se pro transakce SMX šifrování, pouze pokud databázový server, ke kterému probíhá připojení, také podporuje šifrování. Pokud je parametr **ENCRYPT_SMX** nastaven na hodnotu 2, jsou povolena pouze připojení k šifrovaným databázovým serverům. Nastavením parametru **ENCRYPT_SMX** na hodnotu 0 se šifrování mezi dvěma servery zakáže.

Získání statistiky SMX

Statistiku připojení SMX můžete zobrazit pomocí nástroje **onstat** nebo pomocí tabulek rozhraní SMI (system-monitoring interface).

Chcete-li zobrazit statistiku připojení SMX, použijte příkaz **onstat -g smx**.

Chcete-li zobrazit statistiku relace SMX, použijte příkaz **onstat -g smx ses**.

Příklady výstupů příkazů **onstat -g smx** a **onstat -g smx ses** naleznete v kapitole týkající se obslužného programu **onstat** v příručce *IBM Informix Administrator's Reference*.

Konfigurace sekundárního serveru RS

Tato část popisuje hardwarové požadavky na konfiguraci prostředí sekundárního serveru RS.

Hardwarové a softwarové požadavky

Sekundární server RS uchovává úplnou kopii fyzické databáze. Z toho důvodu se musí shodovat následující vlastnosti s primárním serverem:

- Hardware počítače, na kterém je spuštěn databázový server.
- Objem diskového prostoru přiděleného pro prostory dbspaces.
- Posuny ve fyzickém zařízení použité k vytvoření prostorů dbspaces.

Úplný seznam požadavků na hardware a na operační systém sekundárních serverů naleznete v části “Konfigurace systému pro replikaci HDR” na stránce 20-2.

První spuštění sekundárního serveru RS

Po dokončení konfigurace hardwaru sekundárního serveru RS je možné jej spustit. Tato část podrobně popisuje kroky nezbytné ke spuštění primárního serveru se sekundárním serverem RS.

Předpokládejme, že chcete spustit primární server a sekundární server RS, server **ServerA** a server **ServerB**. V následujících krocích je podrobně popsán postup spuštění serverů se serverem **ServerA** jako primárním databázovým serverem a se serverem **ServerB** jako sekundárním databázovým serverem RS. Tabulka 21-1 na stránce 21-6 uvádí příkazy požadované k provedení jednotlivých kroků. Zálohování a obnovení můžete provést pomocí obslužného programu **ontape** nebo pomocí programu **ON-Bar**. V průběhu celého postupu musíte používat stejný obslužný program.

Důležité: Pokud zálohování a obnovení provádíte pomocí programu ON-Bar, je na obou databázových serverech vyžadován obslužný program **ontape**. Obslužný program **ontape** nelze z databázových serverů, které se podílí na sekundárním serveru RS, odstranit.

V případě potřeby můžete sekundární server RS také nastavit pomocí standardního programu ON-Bar nebo pomocí příkazů obslužného programu **ontape** pro externí zálohování a obnovení.

Postup spuštění primárního serveru se sekundárním serverem RS:

1. Nainstalujte na oba databázové servery uživatelské typy, uživatelské rutiny a moduly DataBlade a poté je zaregistrujte pouze na serveru **ServerA**.
2. Aktivujte na primárním serveru protokolování indexových stránek.
3. Zaznamenejte na primárním serveru identitu sekundárního serveru RS. Heslo je nepovinné a lze jej zadat pouze při prvním spuštění tohoto příkazu. Pokus o změnu hesla po připojení sekundárního serveru RS způsobí chybu.
4. Vytvořte zálohu úrovně 0 serveru **ServerA**.
5. Proveďte fyzické obnovení serveru **ServerB** ze zálohy úrovně 0, kterou jste vytvořili v kroku 4. Neprovádějte logické obnovení.

Pokud používáte:

- Program ON-Bar, použijte k provedení fyzického obnovení příkaz **onbar -r -p**.
 - Program ON-Bar a provádíte externí obnovení, použijte k provedení fyzického obnovení příkaz **onbar -r -p -e**.
 - Obslužný program **ontape**, použijte volbu **ontape -p**. Volbu **ontape -r** nelze použít, protože provádí fyzické i logické obnovení.
 - Obslužný program **ontape** a provádíte externí obnovení, použijte k provedení fyzického obnovení příkaz **ontape -p -e**.
6. Spuštěním příkazu **onmode -d add RSS ServerB** na primárním serveru zaregistrujte název sekundárního serveru RS do databáze **sysha**.
V tuto chvíli není server **ServerB** online a není nastaven na typ RSS, spojení tedy nebude navázáno.
 7. Pomocí příkazu **onmode -d RSS ServerA heslo** nastavte typ serveru **ServerB** na sekundární server RS a zadejte přidružený primární databázový server.

Server **ServerB** se pokusí navázat spojení s primárním databázovým serverem (**ServerA**) a spustí operaci. Připojení by mělo být úspěšně navázáno.

Sekundární databázový server provede logickou obnovu pomocí záznamů logického protokolu, které byly zapsány na primární databázový server během kroku 2. Pokud se všechny tyto záznamy logického protokolu stále nacházejí na primárním databázovém serveru, odešle primární databázový server tyto záznamy přes síť přímo sekundárnímu serveru RS a k logické obnově dojde automaticky.

Pokud jste zálohovali a uvolnili soubory logického protokolu na primárním databázovém serveru, nejsou již záznamy v těchto souborech na disku. Sekundární databázový server vás vyzve k obnovení těchto souborů z pásky. V tomto případě musíte provést krok 8.

Důležité: Kroky 5 a 7 musíte provést společně. Pokud po kroku 5 potřebujete vypnout a restartovat sekundární databázový server, musíte krok 5 opakovat.

8. Pokud záznamy logického protokolu, které byly zapsány na primární databázový server, již na primárním disku nejsou, sekundární databázový server vás vyzve k obnovení těchto souborů z pásky.

Pokud musí sekundární databázový server číst zálohované soubory logického protokolu přes síť, nastavte parametry páskového zařízení na sekundárním databázovém serveru na

zařízení v počítači, ve kterém je spuštěn primární databázový server, nebo na zařízení ve stejném umístění jako primární databázový server.

Po obnovení všech souborů logického protokolu z pásky bude obnovení dokončeno pomocí souborů logického protokolu na disku primárního databázového serveru.

Tabulka 21-1. Postup prvního spuštění primárního serveru se sekundárním serverem RS

Krok	Na primárním serveru	Na sekundárním serveru RS
1.	Nainstalujte uživatelské rutiny (UDR), uživatelské typy (UDT) a moduly DataBlade. Zaregistrujte uživatelské rutiny, uživatelské typy a moduly DataBlade.	Nainstalujte uživatelské rutiny (UDR), uživatelské typy (UDT) a moduly DataBlade modules.
2.	Obslužný program onmode onmode -wf LOG_INDEX_BUILDS=1	
3.	Obslužný program onmode onmode -d add RSS <i>rss_servername heslo</i>	
4.	Příkaz ontape ontape -s -L 0 Příkaz ON-Bar onbar -b -L 0	
6.		Příkaz ontape ontape -p or ontape -p -e Na výzvu k zálohování protokolů odpovězte ne. Příkaz ON-Bar onbar -r -p nebo onbar -r -p -e
7.		Příkaz onmode onmode -d RSS <i>primary_servername heslo</i> Pokud jsou na disku primárního databázového serveru stále uloženy všechny záznamy logického protokolu, které byly zapsány na primární databázový server během kroku 1, pak sekundární databázový server tyto záznamy přečte a provede logickou obnovu. (Jinak musí být proveden krok 5).
8.		Příkaz ontape ontape -l Příkaz ON-Bar onbar -r -l Tento krok je třeba pouze v případě, že vás sekundární databázový server vyzve k obnovení těchto souborů z páskové zálohy.

Zkrácení doby nastavení alternativní metodou zálohování: Používáte-li při zálohování a obnovení dat obslužný program **ontape**, můžete namísto zálohování souboru do zařízení zvýšit rychlost nastavení sekundárního serveru RS nastavením konfiguračního parametru

TAPEDEV na hodnotu STDIO. Konfigurační parametr TAPEDEV používá hodnotu STDIO jako speciální hodnotu pro konfiguraci standardních datových proudů vstupu - výstupu.

Důležité: Používáte-li hodnotu STDIO, nepracuje obslužný program **ontape** stejně jako při zálohování do zařízení.

Další informace o použití hodnoty STDIO naleznete v části *IBM Informix Backup and Restore Guide*.

Získání statistiky sekundárního serveru RS

Příkaz `onstat` lze použít k zobrazení informací o stavu sekundárních serverů RS. Příkaz:

```
onstat -g rss
```

zobrazuje, které sekundární servery RS existují, která data byla odeslána sekundárním serverům RS a které sekundární servery RS potvrdily příjem. Tento příkaz má možnosti zobrazení rozšířených informací o jednom serveru nebo o více sekundárních serverech.

Příklady výstupu příkazu **onstat -g rss** naleznete v informacích o obslužném programu `onstat` v příručce *IBM Informix Administrator's Reference*.

Konfigurace sekundárního serveru RS

Přeměna sekundárního serveru RS na standardní server

Sekundární server RS lze převést na standardní server vyvoláním následujícího příkazu na sekundárním serveru RS:

```
onmode -d standard
```

Sekundární server RS převedený na standardní server nelze převést zpět na sekundární server RS.

Přeměna sekundárního serveru RS na sekundární server replikace HDR

Sekundární server RS lze přeměnit na sekundární server replikace HDR zadáním následujícího příkazu na sekundárním serveru RS:

```
onmode -d secondary <název primárního serveru>
```

Pokud je zdrojem protokolů standardní server, bude převeden na primární server replikace HDR.

Opětovné zřízení sekundárního serveru replikace HDR jako sekundárního serveru RS

Chcete-li znovu zřídit sekundární server replikace HDR jako sekundární server RS, zadejte následující příkazy:

1. Zaregistruje na primárním serveru HDR název sekundárního serveru RS:

```
onmode -d add RSS název_serveru_rss [nepovinné heslo]
```
2. Spuštěním následujícího příkazu na sekundárním serveru replikace HDR převedte server na sekundární server replikace RS

```
onmode -d RSS název_primárního_serveru [nepovinné heslo]
```

Opětovné vrácení softwaru v prostředí serveru RS

Návrat ke dřívější verzi softwaru vyžaduje, aby byly všechny aktivní sekundární servery RS v síti převedeny na standardní servery spuštěním příkazu **onmode -d standard**. Po opětovném vrácení bude odstraněna systémová databáze **sysba**.

Odstranění sekundárního serveru RS

Sekundární server RS můžete odstranit z konfigurace provedením následujícího příkazu:

```
onmode -d delete RSS název_serveru_rss
```

Obnovení při havárii v prostředí sekundárního serveru RS

Pokud sekundární server HDR selže nebo se stane primárním serverem replikace HDR, může být sekundární server RS převeden na sekundární server replikace HDR.

Zabezpečení sekundárního serveru RS

Sekundární servery RS podporují podobná šifrovací pravidla jako replikace HDR. Podrobnosti naleznete v části "Šifrování přenosu dat mezi databázovými servery HDR" na stránce 20-14.

Další informace o nastavení a konfiguraci šifrování mezi servery a sekundárními servery RS naleznete v části o připojení SMX (Server Multiplexer Group).

Vytvoření nebo změna hesla na sekundárním serveru RS

Na primárním serveru musí být definována identita jednotlivých sekundárních serverů RS, aby se zabránilo neoprávněným serverům v připojení k síti a v získání informací protokolu. Název a heslo jednotlivých sekundárních serverů RS lze definovat buď před nebo po zálohování primárního serveru úrovně 0. To lze provést zadáním následujícího příkazu:

```
onmode -d add RSS název_serveru_rss [optional password]
```

Heslo můžete změnit, pouze pokud server nebyl připojen do vysoce dostupného prostředí. Při pokusu o změnu hesla již připojeného sekundárního serveru RS dojde k následující chybě:
Heslo RSS nelze změnit po připojení sekundárního serveru

Předtím, než je sekundární server RS připojen, můžete změnit heslo pomocí následujícího příkazu:

```
onmode -d change RSS název_serveru_rssheslo
```

Výše uvedený příkaz lze spustit na primárním serveru.

Návrh klientů sekundárních serverů RS

Tato část pojednává o různých aspektech návrhu (kromě aspektu přesměrování popsaného výše) klientů, kteří se připojují k databázovým serverům využívajícím replikaci dat.

Návrh klientů používajících sekundární Server RS

Viz část *Přesměrování a konektivita klientů replikace dat* v příručce *IBM Informix Administrator's Reference*.

Monitorování alarmů událostí sekundárního serveru RS

Provádějte monitorování následujících alarmů událostí, které jsou spouštěny sekundárními servery RS (viz Tabulka 21-2). Když je alarm spuštěn, zapíše se zpráva do protokolu zpráv. Další informace naleznete v kapitole týkající se alarmů zpráv a konfiguračních parametrů v příručce *IBM Informix Administrator's Reference*.

Tabulka 21-2. Alarmy událostí sekundárních serverů RS

ID třídy	Závažnost	Zpráva třídy	Zpráva
40	ALRM_ATTENTION	Pozice přehrání protokolu RSS <i>název_serveru</i> spadá příliš daleko za zdroj RSS	Tato zpráva se zobrazuje, pokud pozice přehrání protokolu sekundárního serveru RS spadá příliš daleko za primární. Pokud by tento trend pokračoval, nemusí mít primární server k dispozici soubor logického protokolu, až ho bude sekundární server RS požadovat.
40	ALRM_ATTENTION	Byl přidán server RSS <i>název_serveru</i>	Do klastru byl přidán sekundární server RS s názvem <i>název_serveru</i> .
40	ALRM_ATTENTION	Byl odstraněn server RSS <i>název_serveru</i>	Z klastru byl odstraněn sekundární server RS s názvem <i>název_serveru</i> .
40	ALRM_ATTENTION	Heslo HA pro server RSS <i>název_serveru</i> bylo změněno	Heslo ke konfiguraci sekundárního serveru RS s názvem <i>název_serveru</i> bylo změněno.
40	ALRM_ATTENTION	Server RSS <i>název_serveru</i> nepotvrzuje transakce protokolu	Sekundární server RS s názvem <i>název_serveru</i> nepotvrzuje transakce protokolu. Primární server nebude odesílat žádné další stránky protokolu, dokud nebude přijato potvrzení.
40	ALRM_ATTENTION	Chyba při přijetí vyrovnávací paměti ze serveru RSS <i>název_serveru</i> , vypíná se	Na primárním serveru došlo k chybě při příjmu zprávy ze sekundárního serveru RS s názvem <i>název_serveru</i> .

Kapitola 22. Použití sekundárních serverů se sdíleným diskem (vydání Enterprise)

Obsah kapitoly	22-1
Sekundární server SD	22-1
Kdy používat sekundární Server SD	22-2
Charakteristika a omezení prostředí sekundárního serveru SD	22-2
Požadavky na konfiguraci sekundárního serveru SD	22-2
Hardwarové a softwarové požadavky na sekundární serveru SD	22-2
Nastavení sekundárního serveru SD	22-2
Obnovení při havárii v prostředích sekundárního serveru SD	22-3
Zabezpečení pro případ poruchy, pokud je aktuální primární server připojen k novému primárnímu serveru.	22-3
Zabezpečení pro případ poruchy, pokud není primární server připojen k novému primárnímu serveru	22-3
Zabezpečení pro případ poruchy, pokud jsou všechny uzly ve vysoce dostupném klastru mimo provoz	22-3
Získání statistiky sekundárního serveru SD	22-4
Konfigurace sekundárního serveru SD	22-4
Přeměna sekundárního serveru SD na primární server.	22-4
Zakázání primárního serveru v prostředí sekundárního serveru SD.	22-4
Obnovení při havárii v prostředí sekundárního serveru SD	22-4
Zabezpečení sekundárního serveru SD	22-5
Monitorování alarmů událostí sekundárního serveru SD	22-6

Obsah kapitoly

Tato kapitola poskytuje přehled nastavení a konfigurace sekundárních serverů SD (sdílený disk) ve vysoce dostupném prostředí. Ve standardní verzi dynamického serveru jsou k dispozici možnosti sekundárního serveru SD. Ve verzi IBM Informix Dynamic Server Express Edition nepracuje mnoho možností vysoké dostupnosti.

V této kapitole jsou popsána následující témata:

- Kdy používat sekundární serveru SD
- Výhody sekundárních serverů SD
- Obnovení při havárii v prostředí sekundárního serveru SD
- Konfigurace sekundárního serveru SD
- Zabezpečení sekundárního serveru SD
- Získání statistiky sekundárního serveru SD

Sekundární server SD

Sekundární server SD je součástí vysoce dostupné konfigurace. V takové konfiguraci má primární server k disku nebo diskovému poli přístup pouze pro zápis a všechny sekundární serveru SD mají přístup pouze pro čtení. Sekundární server SD neuchovává kopii fyzické databáze ve vlastním prostoru na disku, ale sdílí disky s primárním serverem.

Sekundární serveru SD lze použít společně se sekundárními serveru replikace HDR, sekundárními serveru RS a s replikací Enterprise Replication. Účelem sekundárního serveru SD je poskytovat další dostupnost bez nutnosti poskytování dalších disků. Sekundární serveru SD jsou vhodné pro aplikace, ve kterých je vysoký poměr čtení, například ve webových aplikacích.

Sekundární serveru SD lze do vysoce dostupného prostředí přidat velmi rychle, protože nevyžadují samostatnou kopii disku. Protože serveru SD sdílí disk primárních serverů, je

doporučeno používat další způsob zálohování, například zrcadlení, použití sekundárního serveru RS nebo sekundárního serveru replikace HDR.

Kdy používat sekundární Server SD

Při konfiguraci prostředí sekundárního serveru SD byste měli vzít v úvahu následující příčiny:

- Zvýšení kapacity přenosu dat

S podporou více sekundárních serverů je možné převést funkčnost přenosu dat na jiné servery, aniž by byl ovlivněn primární server.

- Záloha pro případ poruchy serveru

V případě poruchy primárního serveru může být sekundární server SD rychle a snadno přeměněn na primární server. Pokud například používáte zařízení SAN (storage area network), která poskytují dostatečně velkou a spolehlivou diskovou paměť, ale chcete mít ochranu proti selhání serveru.

Charakteristika a omezení prostředí sekundárního serveru SD

V prostředích sekundárního serveru platí následující omezení:

- Sekundární server SD nelze přeměnit na sekundární server RS
- Sekundární server SD nelze přeměnit na standardní server, který existuje mimo primární prostředí s vysokou dostupností

Požadavky na konfiguraci sekundárního serveru SD

Tato část popisuje hardwarové požadavky na konfiguraci prostředí sekundárního serveru SD.

Hardwarové a softwarové požadavky na sekundární servery SD

Kromě požadavků na disk (které jsou sdílené s primárním serverem) jsou hardwarové a softwarové požadavky stejné jako na sekundární servery replikace HDR. Systém primárního disku musí být navíc sdílený s počítači, které jsou hostiteli databázových serverů. To znamená, že cesta k prostorům dbspace sekundárního serveru SD je stejná, jako cesta k prostoru dbspace primárního serveru. Viz část: "Konfigurace systému pro replikaci HDR" na stránce 20-2.

Nastavení sekundárního serveru SD

Sekundární server SD se nastaví nejdříve nastavením konfiguračního parametru SDS_ENABLE na hodnotu 1, který povolí prostředí sdíleného disku. Dále se pomocí nástroje onmode nastaví alias, který bude sekundární server používat k připojení k primárnímu serveru. Konfigurační soubor sekundárního serveru SD je potom upraven, aby obsahoval odpovídající možnosti. Nakonec se sekundární server SD spustí pomocí nástroje **oninit**.

1. Nastavte v konfiguračním serveru na primárním serveru hodnotu SDS_ENABLE na 1.
2. Nastavte konfigurační parametr SDS_TIMEOUT

Informace o konfiguračním parametru SDS_TIMEOUT naleznete v příručce *IBM Informix Administrator's Reference*.

3. Nakonfigurujte název aliasu primárního serveru:

```
onmode -d set SDS primary <alias>
```

Název serveru zadaný hodnotou <alias> se stane novým primárním serverem v prostředí sdíleného disku a zdrojem protokolů sekundárního serveru SD.

4. Na sekundárním serveru SD nastavte v konfiguračním souboru sekundárního serveru SD následující parametry konfigurace:

```
SDS_ENABLE  
SDS_PAGING  
SDS_TEMPDBS
```


5. Na uzlu sekundárního serveru SD nastavte následující parametry konfigurace, aby se shodovaly s parametry na primárním serveru:

- ROOTNAME
- ROOTPATH
- ROOTOFFSET
- ROOTSIZE
- PHYSDBS
- PHYSFILE
- LOGFILES
- LOGSIZE

Ostatní konfigurační parametry je vhodné mapovat, aby se shodovaly s parametry primárního serveru, kromě hodnot DBSERVERALIASES, DBSERVERNAME a SERVERNUM.

6. Pomocí příkazu `oninit` spusíte sekundární server SD.

`oninit`

Před spuštěním sekundárního serveru SD musí být aktivní primární server.

Vezměte prosím na vědomí, že při přidání sekundárního serveru dojde ke zvýšení využití paměti ve společné oblasti LGR.

Informace o konfiguračních parametrech naleznete v příručce *IBM Informix Administrator's Reference*.

Obnovení při havárii v prostředích sekundárního serveru SD

Zabezpečení pro případ poruchy, pokud je aktuální primární server připojen k novému primárnímu serveru

Pokud je aktivní vysoce dostupné prostředí, oznámí nový primární server starému serveru, že přebírá vlastnictví sdíleného disku. Starý primární server potom odvolá všechny otevřené transakce a přepne se do sekundárního stavu. Až starý primární server dokončí tento proces, oznámí novému primárnímu serveru, že bylo odvolání dokončeno. Pro primární server to bude signál k pokračování. Tento proces se provádí spuštěním příkazu `onmode -d set sds primary` na novém primárním serveru.

Zabezpečení pro případ poruchy, pokud není primární server připojen k novému primárnímu serveru

V tomto scénáři neexistuje připojení mezi novým a starým serverem. V takovém případě je třeba převod vynutit. Provádí se to vyvoláním příkazu `onmode -d set sds primary force`. Tento příkaz lze vyvolat pouze v případě, že je jisté, že původní primární server není aktivní. Protože klíčové slovo `force` způsobí, že se nový primární server stane zdrojovým serverem bez komunikace se starým primárním serverem, je možné, že na starém primárním serveru, pokud byl ještě aktivní, dojde k porušení databáze.

Zabezpečení pro případ poruchy, pokud jsou všechny uzly ve vysoce dostupném klastru mimo provoz

Specifický problém nastává, pokud při obnovení provozu po selhání všech serverů nelze spustit existující primární server. Příčina je v tom, že primární server musí být schopen se připojit, aby bylo možné spustit sekundární server ve vysoce dostupném klastru. Pokud primární server není aktivní, nelze uskutečnit připojení a nelze spustit sekundární server. Pokud nelze spustit sekundární server, nebude fungovat příkaz pro změnu primárního serveru `onmode command`. Tento problém lze obejít použitím příkazu `oninit -SDS=<nový alias>`, kde `<nový alias>` je alias TCP nového primárního serveru. To umožňuje spustit existující

sekundární server a způsobí, že server v tu chvíli převezme vlastnictví prostředí. Tuto možnost příkazu `oninit` lze použít pouze při spouštění prvního serveru v klastru.

Získání statistiky sekundárního serveru SD

Statistiku sekundárního serveru můžete zobrazit pomocí příkazu `onstat` nebo pomocí tabulek rozhraní SMI (system-monitoring interface).

Chcete-li zobrazit statistiku sekundárního serveru SD, použijte příkaz `onstat -g sds`. Výstup nástroje `onstat` závisí na tom, zda je nástroj spuštěn na primárním nebo sekundárním serveru.

Informace o statistice sdíleného disku na primárním serveru můžete získat dotazem z tabulky `syssrcsds`.

Informace o statistice sdíleného disku na sekundárním serveru můžete získat dotazem z tabulky `sysstrgsds`.

Informace o příkazu `onstat` a tabulkách SMI naleznete v příručce *IBM Informix Administrator's Reference*.

Konfigurace sekundárního serveru SD

Přeměna sekundárního serveru SD na primární server

Sekundární server SD lze převést na primární server vyvoláním následujícího příkazu na sekundárním serveru SD:

```
onmode -d set SDS primary <alias>
```

Sekundární server SD nelze převést na standardní server.

Zakázání primárního serveru v prostředí sekundárního serveru SD

Primární server lze zakázat následujícím příkazem:

1. Na primárním serveru zadejte následující příkaz:

```
onmode -d clear SDS primary <alias>
```

Tento příkaz způsobí, že se primární server stane standardním serverem a zakáže prostředí sdíleného disku.

Obnovení při havárii v prostředí sekundárního serveru SD

V případě selhání primárního serveru by měly být provedeny opravné kroky v tomto pořadí:

1. Na sekundární server SD
2. Na sekundární server replikace HDR
3. Na sekundární server RS

Pokud je primární server aktivní, je možné sekundární server SD přeměnit na primární server pomocí následujících možností příkazu nástroje `onmode`:

```
onmode -d set SDS primary <alias>
```

Na sekundárním serveru SD, který chcete převést na primární server, by měl být spuštěn výše uvedený příkaz. Pokud je vytvořen nový primární server, upraví se všechny sekundární servery automaticky na nový primární server a původní primární server se vypne.

Sekundární server SD se při spuštění pokusí připojit ke svému primárnímu serveru. Pokud je primární server nedostupný, zjištění poruchy a upozornění uživatele může při spuštění sekundárního serveru SD trvat až sedm minut.

Po poruše primárního serveru bude sekundární server SD až do restartování serveru využívat dočasné prostory dbspace určené konfiguračním parametrem SDS_TEMPBS místo dočasných prostorů dbspace určených konfiguračním parametrem DBSPACETEMP.

Pokud používáte konfiguraci, která se skládá z jednoho primárního serveru a jednoho nebo více sekundárních serverů SD, vezměte prosím na vědomí, že v případě vypnutí primárního serveru máte dvě možnosti zotavení:

- Můžete primární server znovu spustit. Pokud je primární server restartován, sekundární server SD bude automaticky vypnut a administrátor databáze bude muset ručně znovu spustit sekundární server SD. Jakmile bude jeden ze sekundárních serverů SD převeden na primární server, původní primární server může být znovu spuštěn jako sekundární server SD. Pokud to bude později nutné, může být převeden zpět na primární server.
- K převedení jednoho ze sekundárních serverů SD na primární server můžete použít příkaz `onmode -d make primary`. Převedení sekundárního serveru SD na primární server vyžaduje zásah administrátora databáze a nedojde k němu automaticky.

Další informace o volbách příkazu `onmode` naleznete v příručce *IBM Informix Administrator's Reference*.

Příkaz `onmode -d make primary` lze spustit na libovolném typu sekundárního serveru za účelem jeho převedení na primární server. Následující tabulka ukazuje, jakým způsobem jsou jednotlivé typy serverů ovlivněny

Tabulka 22-1.

Pokud je primární server:	Rovnocenné servery tohoto typu:	Jsou ovlivněny tímto způsobem:
Sekundární server SD	Sekundární server SD	Připojí se k novému primárnímu serveru a pokračuje
	Sekundární server RS	Připojí se k novému primárnímu serveru a pokračuje
	Sekundární replikace HDR	Připojí se k novému primárnímu serveru a pokračuje
	Starý primární server	Vypnutí
Sekundární replikace HDR	Sekundární server SD	Vypnutí
	Sekundární server RS	Připojí se k novému primárnímu serveru a pokračuje
	Primární replikace HDR	Závisí na akci uživatele
Sekundární server RS	Sekundární server SD	Vypnutí
	Sekundární replikace HDR	Vypnutí
	Sekundární server RS	Vypnutí

Zabezpečení sekundárního serveru SD

Sekundární servery SD podporují podobná šifrovací pravidla jako replikace HDR. Podrobnosti naleznete v části “Splnění požadavků na konfiguraci databázového serveru” na stránce 20-3.

Šifrování mezi libovolným párem primárního a sekundárního serveru může být povoleno nebo zakázáno. To znamená, že je možné šifrovat provoz mezi zdrojovým serverem a jedním sekundárním serverem SD a případně šifrovat provoz na jiný server.

Další informace o nastavení a konfiguraci šifrování mezi primárními servery a sekundárními servery SD naleznete v části “Připojení rozhraní SMX (Server Multiplexer Group)” na stránce 21-4.

Monitorování alarmů událostí sekundárního serveru SD

Provádějte monitorování následujících alarmů událostí, které jsou spouštěny sekundárními servery SD (viz Tabulka 22-2). Když je alarm spuštěn, zapíše se zpráva do protokolu zpráv. Další informace naleznete v kapitole týkající se alarmů zpráv a konfiguračních parametrů v příručce *IBM Informix Administrator's Reference*.

Tabulka 22-2. Alarmy událostí sekundárních serverů SD

ID třídy	Závažnost	Zpráva třídy	Zpráva
41	ALRM_ATTENTION	CHYBA: Odebrání uzlů SDS <i>servername d</i> vypršelo - odstraňuje se	Sekundární server SD byl odstraněn z důvodu situace vypršení časového limitu. Další informace naleznete v popisu konfiguračního parametru SDS_TIMEOUT v části <i>IBM Informix Administrator's Reference</i>

Kapitola 23. Kontrola konzistence

Obsah kapitoly	23-1
Provádění periodické kontroly konzistence	23-2
Ověření konzistence	23-2
Ověření tabulek systémového katalogu	23-2
Ověření datových stránek	23-3
Ověření oblastí	23-3
Ověření indexů	23-3
Ověření logických protokolů	23-3
Ověření rezervovaných stránek	23-3
Ověření metadat	23-3
Monitorování nekonzistence dat	23-3
Čtení chyb aserce v protokolu zpráv a souborech výpisu	23-3
Ověření dat v tabulkách a tabulkových prostorech	23-4
Zachování konzistentních záloh úrovně 0	23-5
Co provést v případě poškození	23-5
Zjištění příznaků poškození	23-5
Oprava poškození indexu	23-5
Oprava chyb vstupu - výstupu bloku	23-5
Shromažďování diagnostických dat	23-6
Chyby vstupu - výstupu bránící v provozu	23-7
Monitorování chyb vstupu - výstupu bránících v provozu v databázovém serveru	23-8
Používání protokolu zpráv ke monitorování chyb vstupu - výstupu bránících v provozu	23-8
Používání alarmů událostí k monitorování chyb vstupu - výstupu bránících v provozu	23-8
Používání vypnutého mapování chybných sektorů	23-9

Obsah kapitoly

Databázové servery Informix jsou navrženy tak, aby rozpoznaly selhání databázového serveru nebo potíže způsobené chybami hardwaru nebo operačního systému. Potíže jsou rozpoznány pomocí *asercí* prováděných v mnoha kritických funkcích serveru. *Aserce* je kontrola konzistence, která ověřuje, zda se obsah stránky, struktury nebo jiné entity shoduje s předpokládaným obsahem.

Pokud jedna z těchto kontrol zjistí, že obsah neodpovídá očekávanému obsahu, databázový server oznámí *chybu aserce* a do protokolu zpráv databázového serveru zapíše text s popisem kontroly, která se nezdařila. Databázový server rovněž shromažďuje další diagnostické informace ve zvláštním souboru, který může být užitečný pro IBM Informix tým technické podpory.

Tato kapitola obsahuje přehledné informace o opatřeních kontroly konzistence a způsobech zacházení s nekonzistencemi. Jsou popsána následující témata:

- Provádění periodické kontroly konzistence.
- Postup při poškození dat.
- Sběr upřesňujících diagnostických informací.
- Monitorování chyb vstupu - výstupu, které zabraňují databázovému serveru v provozu.

Provádění periodické kontroly konzistence

Chcete-li kontrolu konzistence maximálně využít a zajistit integritu zálohování prostorů dbspace, měli byste pravidelně provádět následující akce:

- Ověřit konzistenci všech dat a doplňujících informací databázového serveru.
- Při ověřování konzistence zkontrolovat protokol zpráv a vyhledat v něm chyby aserce.
- Po ověření konzistence vytvořit zálohu prostorů dbspace úrovně 0.

V následujících částech jsou popsány jednotlivé akce.

Ověření konzistence

Protože tato kontrola vyžaduje dostatek času a může způsobit kolize požadavků, naplánujte spuštění kontroly na dobu, kdy je aktivita nejnižší. Tuto kontrolu byste měli provádět před vytvořením zálohy úrovně 0.

Jako součást kontroly konzistence spusíte příkazy, které uvádí Tabulka 23-1.

Tabulka 23-1. Kontrola konzistence dat

Typ ověření	Příkaz
Tabulky systémového katalogu	oncheck -cc
Data	oncheck -cD <i>název_databáze</i>
Oblasti	oncheck -ce
Indexy	oncheck -cI <i>název_databáze</i>
Rezervované stránky	oncheck -cr
Logické protokoly a rezervované stránky	oncheck -cR
Metadata a inteligentní velké objekty	oncheck -cs

Všechny tyto příkazy můžete spustit, pokud je databázový server v režimu online. Další informace o tom, jak *jednotlivé příkazy* uzamykají objekty při jejich kontrole a kteří uživatelé mohou provádět ověřování, naleznete v části týkající se příkazu **oncheck** v příručce *IBM Informix Administrator's Reference*.

Pokud jedna nebo více procedur ověření zjistí chybu, lze ve většině případů potíže vyřešit obnovením databáze ze zálohy prostorů dbspace. Nicméně původcem chyby může být také hardware nebo operační systém.

Ověření tabulek systémového katalogu

K ověření tabulek systémového katalogu použijte příkaz **oncheck -cc**.

Každá databáze obsahuje vlastní systémový katalog s informacemi o tabulkách databáze, sloupcích, indexech, pohledech, omezeních, uložených procedurách a oprávněních.

Zobrazí-li se po skončení ověření upozornění, je jeho jediným účelem oznámit vám, že nebyly nalezeny žádné záznamy konkrétního typu. Tato upozornění neznamenají žádný problém s daty, systémovým katalogem ani s návrhem databáze. Toto upozornění znamená pouze, že pro žádnou tabulku neexistuje žádné synonymum, tedy že systémový katalog neobsahuje žádné záznamy v tabulce **syssyntable**. Například při ověřování tabulek systémového katalogu pro databázi, která nemá definovaná synonyma v žádné její tabulce, se může zobrazit následující upozornění:

```
WARNING: No syssyntable records found.
```

Pokud se však zobrazí chybová zpráva během ověřování tabulky systémového katalogu, je situace zcela odlišná. Kontaktujte neprodleně technickou podporu systému IBM Informix.

Ověření datových stránek

Chcete-li ověřit datové stránky, použijte příkaz **oncheck -cD**.

Pokud ověření datové stránky rozpozná chyby, zkuste uvolnit data z určené tabulky, potom vypusťte tabulku, vytvořte ji znovu a potom znovu zaveďte data do tabulky. Informace o načítání a odstraňování dat naleznete v příručce *IBM Informix Migration Guide*. Pokud se tento postup nezdaří, obnovte data ze zálohy paměťových prostorů.

Ověření oblastí

Chcete-li ověřit oblasti ve všech databázích, použijte příkaz **oncheck -ce**.

Oblasti se nesmí překrývat. Pokud tento příkaz rozpozná chyby, obnovte data ze zálohy paměťových prostorů.

Ověření indexů

Chcete-li ověřit indexy všech tabulek v databázi, použijte příkaz **oncheck -cI**.

Pokud tento příkaz rozpozná chyby, vypusťte a znovu vytvořte poškozený index.

Ověření logických protokolů

Chcete-li ověřit logické protokoly a rezervované stránky, použijte příkaz **oncheck -cR**.

Ověření rezervovaných stránek

Chcete-li ověřit rezervované stránky, použijte příkaz **oncheck -cr**.

Rezervované stránky jsou stránky umístěné na začátku počátečního bloku kořenového prostoru dbspace. Tyto stránky obsahují primární režijní informace databázového serveru. Pokud tento příkaz rozpozná chyby, obnovte data ze zálohy paměťových prostorů.

Při provádění tohoto příkazu se mohou zobrazit upozornění. Ve většině případů se tato upozornění vztahují k situacím, kterých jste si již vědomi.

Ověření metadat

Chcete-li ověřit metadata pro všechny inteligentní velké objekty v databázi, spusťte v každé databázi příkaz **oncheck -cs**. V případě nutnosti obnovte data ze zálohy prostorů dbspace.

Monitorování nekonzistence dat

Pokud kód kontroly konzistence rozpozná nekonzistenci během činnosti databázového serveru, odešle se do protokolu zpráv databázového serveru zprávu o chybě aserce. (Další informace naleznete v kapitole o protokolech zpráv v příručce *IBM Informix Administrator's Reference*.)

Čtení chyb aserce v protokolu zpráv a souborech výpisu

Obrázek 23-1 znázorňuje formát zprávy o chybě aserce v protokolu zpráv.

```
Assert Failed: krátký popis toho, co se nezdařilo
Who: popis přihlášeného uživatele nebo spuštěné relace nebo podprocesu
Result: stav zasažené entity databázového serveru
Action: jakou akci by měl správce databázového serveru provést
See Also: soubory obsahující další diagnostické údaje
```

Obrázek 23-1. Forma chyb aserce v protokolu zpráv

Řádek **See Also:** obsahuje jeden nebo více následujících názvů souborů:

- **af.xxx**
- **shmem.xxx**

Jen pro UNIX

- **gcore.xxx**
- **gcore.xxx**
- */název cesty/core*

Konec Jen pro UNIX

Řetězec znaků **xxx** je ve všech případech hexadecimální číslo společné všem souborům přidruženým k chybám aserce jediného jednotkového procesu. Soubory **af.xxx**, **shmem.xxx**, a **gcore.xxx** se nacházejí v adresáři, který je určen parametrem DUMPDIR v souboru ONCONFIG.

V souboru **af.xxx** je uložena kopie zprávy o chybě aserce, která byla odeslána do protokolu zpráv, a zároveň je v souboru zaznamenán obsah aktuálních důležitých struktur a datových vyrovnávacích pamětí.

V souboru **shmem.xxx** je uložena úplná kopie sdílené paměti databázového serveru v čase výskytu chyby aserce, ale jen pokud je hodnota parametru DUMPSHMEM v souboru ONCONFIG nastavena na 1.

Jen pro UNIX

V operačním systému UNIX obsahuje souboru **gcore.xxx** výpis stavu jádra virtuálního procesu databázového serveru, na kterém byl v okamžiku výskytu chyby spuštěn jednotkový proces, ale pouze tehdy, pokud je hodnota parametru DUMPGCORE v souboru ONCONFIG nastavena na hodnotu 1 a pokud používaný operační systém podporuje obslužný program **gcore**. Soubor **core** obsahuje výpis stavu virtuálního procesu databázového serveru, na kterém byl v okamžiku výskytu chyby spuštěn jednotkový proces, ale pouze tehdy, pokud je hodnota parametru DUMPCORE v souboru ONCONFIG nastavena na hodnotu 1. Názvem cesty k souboru **core** je adresář, ze kterého byl databázový server naposledy vyvolán.

Konec Jen pro UNIX

Ověření dat v tabulkách a tabulkových prostorech

Chcete-li ověřit data v tabulkách a tabulkových prostorech, použijte příkaz **oncheck -cD** v databázi nebo tabulce.

Většinu obecných zpráv o chybách aserce následuje podrobnější informace, která obvykle obsahuje název tabulkového prostoru, ve kterém byla chyba rozpoznána. Pokud tato kontrola zjistí nekonzistenci, uvolněte data z tabulky, tabulku zrušte a vytvořte znovu a potom znovu zaveďte do tabulky data. V opačném případě není potřeba provádět žádnou akci.

V řadě případů se databázový server zastaví, jakmile dojde k chybě aserce. Avšak pokud je zřejmé, že se chyby týkají pouze tabulky nebo menší entity, databázový server pokračuje v činnosti.

Pokud nastane chyba aserce z důvodu nekonzistence datové stránky, ke které databázový server přistupoval na požadavek uživatele, je zpráva o chybě odeslána také do procesu aplikace. Příslušná chyba jazyka SQL závisí na typu probíhající operace. Chyby přístupové metody ISAM mají však téměř vždy číslo -105 nebo -172, jako v následujících příkladech:

-105 ISAM error: bad isam file format
-172 ISAM error: Unexpected internal error

Další podrobnosti o cílech a obsahu zpráv naleznete v kapitole o zprávách v protokolu zpráv v příručce *IBM Informix Administrator's Reference*.

Zachování konzistentních záloh úrovně 0

Zálohu úrovně 0 vytvořte po provedení kontrol popsaných v části “Ověření konzistence” na stránce 23-2, pokud kontroly nezaznamenají žádné chyby. Zachovejte tuto zálohu paměťového prostoru a všechny následující záložní pásy logických protokolů do doby, než provedete další kontrolu konzistence. Kontrolu konzistence byste měli provádět před každým vytvořením zálohy úrovně 0. Pokud tak neučiníte, je vhodné alespoň zachovat všechny pásy potřebné k obnově ze zálohy paměťového prostoru, která byla vytvořena ihned po ověření konzistence databázového serveru.

Co provést v případě poškození

V této části jsou popsány některé příznaky poškození systému databázového serveru a akce, jejichž provedením může databázový server nebo administrátor potíže vyřešit. K poškození databáze může dojít v důsledku potíží hardwaru nebo operačního systému, nebo nějakých neznámých potíží databázového serveru. Poškození může ovlivnit jak data, tak i režijní informace databázového serveru.

Zjištění příznaků poškození

Databázový server upozorní uživatele a správce na možná poškození následujícími způsoby:

- Chybové zprávy zobrazované aplikaci uvádějí, že stránky, tabulky nebo databáze nelze najít. Pokud se operace nezdaří z důvodu nekonzistence příslušných dat nebo režijních informací, zobrazí aplikace vždy jednu z následujících chybových zpráv:
-105 ISAM error: bad isam file format
-172 ISAM error: Unexpected internal error
- Zprávy o chybě aserce jsou zaznamenány do protokolu zpráv databázového serveru. Vždy označují soubory obsahující další diagnostické informace, které vám mohou pomoci odhalit zdroj potíží. Další informace najdete v části “Ověření konzistence” na stránce 23-2.
- Obslužný program **oncheck** zobrazí chybové zprávy.

Oprava poškození indexu

Při prvním podezření na poškození spusťte příkaz **oncheck -cI**, kterým zjistíte, zda je index poškozen.

Pokud je databázový server při kontrole indexů spuštěn v režimu online, příkaz **oncheck** odhalí poškození, ale nevyzve vás k provedení opravy. Pokud k poškození došlo, můžete v režimu online indexy odstranit a znovu vytvořit pomocí příkazů jazyka SQL (databázový server uzamkne tabulku a index). Pokud je příkaz **oncheck -cI** spuštěn v klidovém režimu a odhalí poškození, budete dotázáni, zda se má obslužný program pokusit poškození opravit.

Oprava chyb vstupu - výstupu bloku

Pokud se během činnosti databázového serveru objeví chyba vstupu - výstupu, změní se stav bloku, ve kterém k chybě došlo, na *vypnutý*.

Pokud je blok *vypnutý*, příkaz **onstat -d** zobrazí stav bloku jako PD- v případě primárního bloku a MD- v případě zrcadleného bloku. Příklad výstupu příkazu **onstat -d** naleznete v příručce *IBM Informix Administrator's Reference*.

Protokol zpráv také obsahuje zprávu o umístění chyby a navrženém způsobu řešení. Navržené řešení potíží může opravu vyřešit, ale ne vždy se tím potíže odstraní.

Pokud je vypnutý blok zrcadlený, databázový server pokračuje v činnosti a bude používat zrcadlený blok. Chcete-li odhalit důvod potíží s poškozeným blokem a vyřešit je, použijte obslužné programy operačního systému. Potom musíte přimět databázový server k obnovení dat zrcadleného bloku.

Další informace o obnovení zrcadleného bloku naleznete v části "Obnovení zrcadleného bloku" na stránce 18-6.

Není-li poškozený blok zrcadlený a obsahuje-li soubory logického protokolu, fyzický protokol nebo kořenový paměťový prostor dbspace, iniciuje databázový server okamžitě akci zastavení. V opačném případě může databázový server pokračovat v činnosti, ale nemůže zapisovat do vypnutého bloku nebo číst z vypnutého bloku ani jiných bloků v témže prostoru dbspace. Postupnými kroky zjistěte, proč k chybě vstupu - výstupu došlo, opravte ji a obnovte prostor dbspace ze zálohy.

Pokud je blok označen jako vypnutý (D) a databázový server přepnete do režimu offline, můžete databázový server restartovat, pokud blok označený jako vypnutý neobsahuje kritická data (soubory logického protokolu, fyzický protokol nebo kořenový prostor dbspace).

Shromáždování diagnostických dat

Způsob, jakým databázový server shromažďuje diagnostická data, ovlivňuje několik parametrů v souboru ONCONFIG. Protože chyba aserce může obecně znamenat nepředvídaný problém, upozorněte technickou podporu systému IBM Informix, pokud se chyba objeví. Shromážděná diagnostická data jsou určena pro potřeby týmu technické podpory systému IBM Informix. Obsah a použití souborů **af.xxx** a výpisů sdílené paměti nejsou podrobněji dokumentovány.

Aby bylo možné zjistit příčinu potíží, které způsobily chybu aserce, neodstraňujte diagnostická data, dokud technická podpora systému IBM Informix nepotvrdí, že tak můžete učinit. Soubor **af.xxx** často obsahuje informace, které tým technické podpory potřebuje k vyřešení potíží.

Několik parametrů v souboru ONCONFIG určuje, jaké diagnostické informace má databázový server zachovat, kdykoliv je zjištěna chyba vyhodnocení nebo kdykoliv bude databázový server ukončen:

Jen pro UNIX

- DUMPDIR
- DUMPSHMEM
- DUMPCNT
- DUMPCORE
- DUMPGCORE

Konec Jen pro UNIX

Další informace o konfiguračních parametrech naleznete v příručce *IBM Informix Administrator's Reference*.

Záleží na vás, zda se rozhodnete nastavit tyto parametry. Výstup diagnostických dat může zabrat hodně místa na disku. (Přesný obsah závisí na nastavených proměnných prostředí a na operačním systému.) Prvky výstupu mohou zahrnovat kopii sdílené paměti a snímek stavu procesu v souboru *core*.

Rada: Snímek stavu procesu v souboru *core* je obraz procesu v paměti v okamžiku, kdy došlo k chybě aserce. V některých systémech zahrnuje výpis do souboru *core* i kopii sdílené paměti. Soubory *core* jsou užitečné pouze v těchto případech.

Jsou-li správci databázového serveru omezeni velikostí místa na disku, mohou upřednostnit vytvoření skriptu, který rozpozná výstup diagnostických dat v určitém adresáři a odešle výstup na pásku. Toto řešení zachová diagnostická data a minimalizuje velikost použitého místa na disku.

Chyby vstupu - výstupu bránící v provozu

Systém Informix dělí chyby vstupu - výstupu bránící v provozu do dvou kategorií: destruktivní a nedestruktivní. Chyba vstupu - výstupu bránící v provozu je destruktivní, pokud je disk obsahující databázi nějakým způsobem poškozený. Tento typ události ohrožuje integritu dat a databázový server označí blok a prostor *dbspace* jako vypnutý. Databázový server zakáže přístup k poškozenému disku, dokud nebude disk opraven nebo vyměněn a provedeno fyzické a logické obnovení.

Chyba vstupu - výstupu bránící v provozu je nedestruktivní, pokud neohrožuje integritu dat. K nedestruktivním chybám dochází, pokud omylem odpojíte kabel, omylem odstraníte symbolické propojení, které ukazuje na blok nebo když dojde k poškození řadiče disku.

Aby mohl databázový server označit chybu vstupu - výstupu jako chybu bránící v provozu, musí být splněna dvě kritéria. Zprv, k chybě musí dojít, když se databázový server pokouší provést operaci v bloku, pro který platí alespoň jedna z následujících charakteristik:

- Blok nemá žádné zrcadlo.
- Primární nebo zrcadlený blok, který tomuto bloku odpovídá, je v režimu offline.

Zadruhé musí chyba nastat, když se databázový server pokouší neúspěšně provést jednu z následujících operací:

- Vyhledávání v bloku, čtení bloku nebo zapisování do bloku.
- Otvírání bloku.
- Ověřování, zda jsou informace o bloku na první použité stránce platná.

Databázový server provádí toto ověření jako logickou kontrolu ihned po otevření bloku.

Databázovému serveru můžete během vyšetřování chyb vstupu - výstupu bránících v provozu zabránit, aby označil prostor *dbspace* jako vypnutý. Pokud usoudíte, že potíže jsou triviální (pokud například došlo k odpojení kabelu), můžete databázový server přepnout do režimu offline a hned poté do režimu online, aniž by bylo nutné obnovit zasažený prostor *dbspace* ze zálohy. Pokud usoudíte, že se jedná o vážnější potíže, jako je poškozený disk, můžete pomocí příkazu **onmode -O** označit zasažený prostor *dbspace* jako vypnutý a pokračovat v činnosti.

Monitorování chyb vstupu - výstupu bránících v provozu v databázovém serveru

Databázový server vás na chyby vstupu - výstupu bránící v provozu upozorní dvěma způsoby:

- protokol zpráv
- alarmy událostí

Používání protokolu zpráv ke monitorování chyb vstupu - výstupu bránících v provozu

Dojde-li k chybě vstupu - výstupu bránící v provozu, odešle databázový server do protokolu zpráv následující zprávu:

```
Assert Failed: Chunk {číslo bloku} is being taken OFFLINE.  
Who: popis přihlášeného uživatele nebo spuštěné relace/podprocesu  
Result: stav zasažené entity databázového serveru  
Action: jakou akci by měl administrátor databázového serveru provést  
See Also: soubor DUMPDIR/af.uniqid obsahující další diagnostická data
```

Výsledek a akce závisejí na aktuálním nastavení ONDBSPACEDOWN, popsaném v následující tabulce.

Nastavení ONDBSPACEDOWN	Výsledek	Akce
0	Prostor dbSPACE {název_prostoru} je zakázán.	Obnovení prostoru dbSPACE {název_prostoru}.
	Prostor blobSPACE {název_prostoru} je zakázán.	Obnovení prostoru blobSPACE {název_prostoru}.
1	Databázový server se musí zastavit.	Vypněte a restartujte databázový server.
2	Databázový server bude blokován v příštím kontrolním bodě.	Pomocí příkazu onmode -k databázový server vypnete nebo můžete použít příkaz onmode -O pro potlačení blokování.

Hodnota ONDBSPACEDOWN nemá žádný dopad na dočasné prostory dbSPACE. Databázový server pokračuje ve zpracování dočasných prostorů dbSPACE bez ohledu na nastavení ONDBSPACEDOWN. Vyžaduje-li dočasný prostor dbSPACE opravu, můžete ho vypustit a vytvořit znovu.

Další informace o interpretaci zpráv, které databázový server odesílá do protokolu zpráv, naleznete v kapitole o protokolech zpráv v příručce *IBM Informix Administrator's Reference*.

Používání alarmů událostí k monitorování chyb vstupu - výstupu bránících v provozu.

V případě, že v prostoru dbSPACE dojde k chybám vstupu - výstupu bránícím v provozu, předá databázový server následující hodnoty spustitelnému souboru alarmů událostí jako parametry.

Parametr	Hodnota
Závažnost	4 (stav nouze)
Třída	5

Zpráva třídy	Prostor dbspace je zakázán: 'název_prostoru_dbspace'
Specifická zpráva	Blok {číslo bloku} je přepnut do režimu OFFLINE.

Chcete-li, aby vás databázový server upozorňoval na chyby vstupu - výstupu bránící v provozu prostřednictvím alarmů událostí, napište skript, který databázový server spustí pokaždé, když rozpozná chybu vstupu - výstupu bránící v provozu. Další informace o nastavení tohoto vlastního spustitelného souboru naleznete v dodatcích k alarmům událostí a v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.

Používání vypnutého mapování chybných sektorů

Server Dynamic Server přenechává mapování chybných sektorů operačnímu systému hostitelského počítače. Databázový server se o chybném sektoru nebo chybné stopě dozví, jakmile obdrží chybovou zprávu při systémovém volání. Když tato situace nastane, pokusí se databázový server několikrát o přístup, aby se ujistil, že tento stav není falešný. Pokud je stav potvrzen, označí databázový server jako *vypnutý* ten blok, ze kterého se pokoušel číst nebo do kterého se pokoušel zapisovat.

Databázový server nemůže žádným způsobem identifikovat umístění chybného válce, stopy nebo sektoru, protože jedinou dostupnou informací je umístění dat v bloku, na kterém se pokusil provést operaci vstupu - výstupu, vyjádřené v bajtech posunem od začátku bloku.

Pokud databázový server rozpozná chybu vstupu - výstupu v bloku, který *není* zrcadlený, označí blok jako vypnutý. Pokud blok, který je označený jako vypnutý, obsahuje soubory logického protokolu, fyzický protokol nebo kořenový prostor dbspace, databázový server ihned iniciuje akci zastavení. V opačném případě může databázový server pokračovat v činnosti, ale aplikace nemají přístup k bloku, který je vypnutý, dokud nebude obnoven jeho prostor dbspace.

Část 5. Distribuovaná data

Kapitola 24. Protokoly vícefázového potvrzování

Obsah kapitoly	24-2
Správci transakcí	24-2
Použití knihovny TP/XA se správcem transakcí	24-2
Použití serveru MTS/XA (Microsoft Transaction Server).	24-2
Použití podpory transakcí dynamického serveru pro externí zdroje dat vyhovující specifikaci XA.	24-3
Použití volně vázaného režimu a provázaného režimu	24-4
Protokol dvoufázového potvrzování	24-5
Kdy se používá protokol dvoufázového potvrzování	24-5
Koncepce dvoufázového potvrzování	24-5
Fáze protokolu dvoufázového potvrzování	24-6
Fáze před potvrzením	24-6
Fáze po rozhodnutí	24-6
Jak protokol dvoufázového potvrzování ošetřuje selhání	24-7
Typy selhání ošetřené automatickou obnovou	24-7
Role administrátora v procesu automatické obnovy.	24-7
Mechanismus automatického zotavení při selhání koordinátora	24-7
Mechanismus automatického zotavení při selhání účastníka	24-7
Optimalizace předpokládaného konce	24-8
Nezávislé akce	24-8
Situace, které vyvolávají nezávislé akce.	24-8
Možné důsledky nezávislých akcí	24-8
Nezávislé akce, které umožní úspěšné dokončení transakcí	24-9
Nezávislé akce, které vyústí v chybový stav	24-9
Nezávislé akce, jejichž důsledkem je heuristické rozhodnutí.	24-9
Scénář heuristického odvolání transakce	24-10
Podmínky, jejichž výsledkem je heuristické odvolání transakce	24-10
Výsledky heuristického odvolání transakce	24-11
Scénář heuristického ukončení transakce	24-12
Kdy provést heuristické ukončení transakce	24-12
Jak používat příkaz onmode -Z	24-13
Akce při heuristickém ukončení transakce.	24-13
Monitorování globální transakce	24-13
Chyby protokolu dvoufázového potvrzování	24-14
Dvoufázové potvrzování a záznamy logického protokolu	24-14
Záznamy logického protokolu při potvrzení transakce	24-15
Záznamy logického protokolu zapsané během heuristického odvolání transakce	24-16
Záznamy logického protokolu zapsané během heuristického ukončení transakce	24-17
Konfigurační parametry používané při dvoufázovém potvrzování	24-18
Funkce parametru DEADLOCK_TIMEOUT	24-18
Funkce parametru TXTIMEOUT	24-19
Protokol dvoufázového potvrzování	24-19
Brány, které se mohou účastnit heterogenního potvrzování transakcí	24-20
Povolení a zakázání heterogenního potvrzování	24-20
Jak funguje heterogenní potvrzování	24-21
Fáze před potvrzením	24-21
Fáze potvrzení brány	24-21
Optimalizace heterogenního potvrzování	24-22
Důsledky selhání heterogenního potvrzování	24-22
Selhání koordinačního databázového serveru	24-22
Selhání účastníka	24-22

Obsah kapitoly

Protokol dvoufázového potvrzování zajišťuje, že transakce budou ve více databázových serverech shodně potvrzeny nebo shodně odvolány. *** K manipulaci s daty v jiných databázích než databázích Informix můžete používat databázové servery Informix s produkty IBM Informix Enterprise Gateway nebo správce transakcí. *** Distribuované dotazy v databázových serverech Informix podporují dvoufázové potvrzování.

Protokol heterogenního potvrzování zajišťuje, že aktualizace v jedné nebo více databázích Informix a *jedné* databázi jiné než Informix budou shodně potvrzeny nebo shodně odvolány.

Tato kapitola obsahuje informace o použití protokolu dvoufázového potvrzování. Informace o ruční obnově z nezdařené dvoufázově potvrzované transakce naleznete v kapitole Kapitola 25, "Ruční obnova při selhání dvoufázového potvrzování", na stránce 25-1.

Tato kapitola obsahuje také informace o použití podpory pro přenos externích zdrojů dat vyhovujících specifikaci XA, které se mohou na dvoufázově potvrzovaných transakcích podílet. Další informace naleznete v části "Použití podpory transakcí dynamického serveru pro externí zdroje dat vyhovující specifikaci XA" na stránce 24-3.

Správci transakcí

Správci transakcí podporují dvoufázové potvrzování a odvolávání transakcí. Pokud například používáte databázi Informix jako účetní systém databázi Oracle a jako převodní systém databázi Sybase, můžete ke komunikaci mezi různými databázemi používat správce transakcí. Správce transakcí můžete také použít k zajištění konzistence dat mezi databázemi Informix a databázemi jinými než Informix pomocí distribuovaných transakcí místo replikace ER (Enterprise Replication) nebo replikace HDR (High-Availability Data Replication).

Použití knihovny TP/XA se správcem transakcí

Globální transakce je distribuovaný dotaz, při kterém je do dotazu zahrnut více než jeden databázový server. Prostředí globální transakce je tvořeno následujícími částmi:

- klientskou aplikací,
- *** Správce zdrojů (databázový server Informix)
- správcem transakcí (softwaru jiného dodavatele).

TP/XA je knihovna funkcí, které umožňují databázovému serveru, aby se choval jako správce zdrojů v prostředí X/Open DTP. Knihovna TP/XA se instaluje jako část produktu IBM Informix ESQL/C, aby umožnila komunikaci mezi správcem transakcí jiného dodavatele a databázovým serverem. Prostředí X/Open podporuje rozsáhlé a vysoce výkonné aplikace OLTP. Další informace naleznete v příručce *IBM Informix TP/XA Programmer's Manual*.

Knihovnu TP/XA použijte, pokud bude mít vaše databáze následující charakteristiky:

- Data jsou distribuována mezi několika databázemi od různých dodavatelů.
- Transakce zahrnují data systému Informix data systémů jiných než Informix.

Použití serveru MTS/XA (Microsoft Transaction Server)

Databázový server jako správce transakcí v prostředí XA podporuje server MTS/XA (Microsoft Transaction Server). Chcete-li používat server MTS/XA, nainstalujte sadu IBM Informix Client Software Developer's Kit, nejnovější verzi ovladače IBM Informix ODBC Driver a server MTS/XA. Server MTS/XA pracuje v operačním systému Windows. Další

informace získáte od technické podpory systému IBM Informix nebo v příručce *IBM Informix Client Products Installation Guide* a v dokumentaci serveru MTS/XA.

Použití podpory transakcí dynamického serveru pro externí zdroje dat vyhovující specifikaci XA

Správce transakcí dynamického serveru, který je nedílnou součástí a nikoli odděleným modulem dynamického serveru, rozpozná externí zdroje dat vyhovující specifikaci XA. Tyto zdroje dat se mohou podílet na dvoufázově potvrzovaných transakcích.

Správce transakcí vyvolává podpůrné rutiny pro každý externí XA-kompatibilní datový zdroj, který se podílí na distribuované transakci při určité transakční události, jako je příprava, potvrzení nebo odvolání. Tato interakce odpovídá standardům pro rozhraní XA sdružení X/Open XA.

Podpora transakcí XA-kompatibilních externích zdrojů dat, které jsou také nazývány *správci zdrojů*, vám umožňuje následující:

- Vytváření typů XA-kompatibilních externích zdrojů dat a instancí XA-kompatibilních externích zdrojů dat.
- Vytváření nebo úpravy uživatelských rutin (UDR), rozhraní virtuální tabulky nebo rozhraní virtuálního rejstříku, za účelem umožnění XA-kompatibilním externím zdrojům dat poskytování mechanismů přístupu k externím datům.
Modul MQ DataBlade je příkladem sady uživatelských rutin (UDR), které poskytují tento druh přístupu k externím datům.
- Registrování XA-kompatibilních externích zdrojů dat prostřednictvím dynamického serveru.
- Odregistrování XA-kompatibilních externích zdrojů dat.
- Použití několika XA-kompatibilních externích zdrojů dat v rámci jediné globální transakce.

*** Koordinace transakcí s XA-kompatibilním externím zdrojem dat je podporována pouze v protokolovaných databázích Informix a v databázích kompatibilních se standardem ANSI, jelikož tyto databáze podporují transakce. Koordinace transakcí s XA-kompatibilním externím zdrojem dat není u neprotokolovaných databází podporována.

Můžete použít následující příkazy DLL, které jsou rozšířením příkazů jazyka SQL pro správu zdrojů dat a jejich typů:

Příkaz	Popis
CREATE XADATASOURCE TYPE	Vytvoří typ XA-kompatibilního externího zdroje dat.
CREATE XADATASOURCE	Vytvoří instanci XA-kompatibilního externího zdroje dat.
DROP XADATASOURCE	Odstraní instanci XA-kompatibilního externího zdroje dat.
DROP XADATASOURCE TYPE	Odstraní typ XA-kompatibilního externího zdroje dat.

Více informací o těchto příkazech naleznete v části *IBM Informix Guide to SQL: Syntax*.

Spolupráce mezi dynamickým serverem a XA-kompatibilním externím zdrojem dat probíhá prostřednictvím sady uživatelských rutin pro podporu standardu XA, jako je **xa_open**, **xa_end**, **xa_commit** a **xa_prepare**. Tyto pomocné rutiny vytvoříte před použitím příkazu CREATE XADATASOURCE TYPE. Další informace naleznete v příručce *IBM Informix DataBlade API Programmer's Guide*.

Po vytvoření XA-kompatibilního externího zdroje dat můžete zdroj dat zaregistrovat k aktuální transakci a odregistrovat jej pomocí funkcí **mi_xa_register_xadatasource()** nebo **ax_reg()** a **mi_xa_unregister_xadatasource()** nebo **ax_unreg()**. V distribuovaném prostředí je třeba zdroj dat zaregistrovat k lokálnímu koordinačnímu serveru. Tato registrace je dočasná a přetrvá pouze po dobu trvání transakce. Více informací o použití těchto funkcí naleznete v příručkách *IBM Informix DataBlade API Function Reference* a *IBM Informix DataBlade API Programmer's Guide*.

Pomocí následujících voleb pomocného programu **onstat** můžete zobrazit informace o transakcích, ve kterých jsou zapojeny XA-kompatibilní zdroje dat:

Volba programu onstat	Informace o XA-kompatibilním zdroji dat, kterou příkaz zobrazí
onstat -x	Zobrazí informace o účastnících transakce vyhovujících standardu XA.
onstat -G	Zobrazí informace o účastnících globální transakce vyhovujících standardu XA.
onstat -g ses session id	Zobrazí informace o relaci, včetně informací o zdrojích dat účastnících se transakce.

Modul IBM Informix MQ DataBlade poskytuje pro XA-kompatibilní zdroje dat externí mechanismy přístupu k datům. Další informace o modulu DataBlade naleznete v příručce *IBM Informix Database Extensions User's Guide*.

Použití volně vázaného režimu a provázaného režimu

Databázový server podporuje globální transakce XA ve volně vázaném režimu a provázaném režimu.

- *Volně vázaný režim* znamená, že různé databázové servery koordinují transakce, ale nesdílejí zdroje. Záznamy ze všech větví transakcí se v logickém protokolu zobrazují jako samostatné transakce.
- *Provázaný režim* znamená, že různé databázové servery koordinují transakce a sdílejí zdroje, například zamykání a protokolování. Záznamy ze všech větví transakcí se v logickém protokolu zobrazují jako jediná transakce.

Správce transakcí Tuxedo poskytovaný systémy BEA podporuje volně vázaný režim. Správce transakcí Tuxedo funguje na platformách UNIX i Windows.

Jen pro Windows

Správce transakcí MTS/XA Transaction Manager, který pracuje pouze v operačním systému Windows, podporuje provázaný režim. Podpora provázaných transakcí MTS v databázovém serveru zahrnuje:

- Podporu aplikačních programů se dvěma vrstvami (vrstvou aplikační logiky a vrstvou přístupu k datům).
- Společné oblasti připojení a relací.

Podpora provázaných transakcí serverem MTS neovlivňuje existující podporu volně vázaných transakcí. Databázový server může současně používat podporu volně vázaných i provázaných transakcí.

Podpora provázaných transakcí serverem MTS má tato omezení:

- Dočasné tabulky se omezují na jednu větev transakce. Různé větve transakce v rámci globální transakce nemohou sdílet dočasnou tabulku.
- Různé větve transakce v rámci globální transakce nemohou sdílet kurzory.

- Různé větve transakce v rámci globální transakce nemohou sdílet úroveň izolace nebo režim čekání zámku. Úroveň izolace a režim čekání zámku musejí být nastaveny jednotlivě nebo musí být nastaveny na výchozí úroveň. Pokud chcete mít stejnou úroveň izolace pro všechny větve transakce, musíte k určení této informace pro všechny větve transakce použít jazyk SQL.

Konec Jen pro Windows

Přejete-li si kompletní seznam podporovaných správců transakcí, obraťte se na svého marketingového zástupce.

Protokol dvoufázového potvrzování

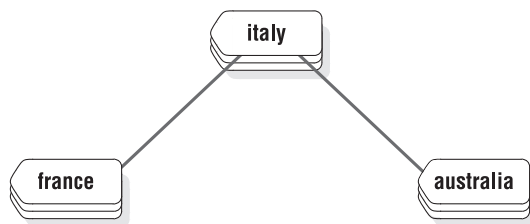
Protokol dvoufázového potvrzování poskytuje automatický mechanismus obnovy v případě selhání systému nebo média během vykonávání transakce. Protokol dvoufázového potvrzování zajistí, že všechny zúčastněné databázové servery obdrží a implementují totožnou akci (potvrzení transakce nebo její odvolání), bez ohledu na místní selhání nebo selhání sítě.

Pokud některý databázový server není schopný potvrdit svou část transakce, musí být zabráněno *všem* databázovým serverům účastnícím se na transakci v potvrzení jejich práce.

Kdy se používá protokol dvoufázového potvrzování

Databázový server automaticky používá protokol dvoufázového potvrzování pro jakoukoli transakci, která mění data v několika databázových serverech.

Předpokládejme například, že tři databázové servery s názvy **australia**, **italy** a **france**, jsou propojeny způsobem, který ilustruje Obrázek 24-1.



Obrázek 24-1. Propojené databázové servery

Pokud spustíte příkaz, který uvádí Obrázek 24-2, bude výsledkem jedna aktualizace a dvě vložení ve třech různých databázových serverech.

```

CONNECT TO stores_demo@italy
BEGIN WORK
  UPDATE stores_demo:manufact SET manu_code = 'SHM' WHERE manu_name = 'Shimara'
  INSERT INTO stores_demo@france:manufact VALUES ('SHM', 'Shimara', '30')
  INSERT INTO stores_demo@australia:manufact VALUES ('SHM', 'Shimara', '30')
COMMIT WORK
  
```

Obrázek 24-2. Příklad distribuované transakce

Koncepce dvoufázového potvrzování

Každá globální transakce má svého *koordinátora* a jednoho nebo více *účastníků*. Koordinátor a účastníci jsou definováni následovně:

- *Koordinátor* dohlíží na řešení konfliktů v globální transakci. Rozhoduje, zda má být globální transakce potvrzena nebo zastavena.

Protokol dvoufázového potvrzování vždy přiřazuje roli koordinátora *aktuálnímu* databázovému serveru. Roli koordinátora nelze v rámci jedné transakce změnit. Ve vzorové transakci, kterou uvádí Obrázek 24-2 na stránce 24-5, je koordinátorem server **italy**. Pokud změňte první řádek v tomto příkladu na následující příkaz, přiřadí protokol dvoufázového potvrzování roli koordinátora databázovému serveru **france**:

```
CONNECT TO stores_demo@france
```

Koordinátora distribuované transakce zobrazíte pomocí volby **onstat -x**. Další informace naleznete v části “Monitorování globální transakce” na stránce 24-13.

- Každý *účastník* dohlíží na vykonání jedné z *větví transakce*, která je součástí globální transakce zahrnující jednotlivou lokální databázi. Globální transakce zahrnuje několik větví transakce v těchto případech:

- Aplikace používá k práci v rámci globální transakce několik procesů.
- V rámci téže globální transakce pracuje několik vzdálených aplikací.

Obrázek 24-1 na stránce 24-5 uvádí jako účastníky databázové servery **france** a **australia**. Koordinační databázový server **italy** rovněž působí jako účastník, protože také provádí aktualizaci.

Protokol dvoufázového potvrzování spoléhá na dva způsoby komunikace, na *zprávy* a na *záznamy logického protokolu*:

- Mezi koordinátorem a každým účastníkem jsou předávány zprávy. Zprávy od koordinátora zahrnují identifikační číslo transakce a instrukci (například **prepare to commit** (přípravit k potvrzení), **commit** (potvrdit) nebo **roll back** (odvolat)). Zprávy od každého účastníka zahrnují stav transakce a výkaz provedených akcí (například lze potvrdit nebo nelze potvrdit, potvrzena nebo odvolána).
- Záznamy logického protokolu týkající se transakce jsou uchovávány na disku nebo pásce, aby mohly zajistit integritu a konzistenci dat i v případě, že by na zúčastněném databázovém serveru došlo k selhání (účastníka nebo koordinátora).

Další informace naleznete v části “Dvoufázové potvrzování a záznamy logického protokolu” na stránce 24-14.

Fáze protokolu dvoufázového potvrzování

Při transakci s dvoufázovým potvrzováním odesílá koordinátor všechny instrukce změn dat (například vložení) všem účastníkům. Potom koordinátor spustí protokol dvoufázového potvrzování. Protokol dvoufázového potvrzování se skládá ze dvou částí, z *fáze před potvrzením* a z *fáze po rozhodnutí*.

Fáze před potvrzením

Během fáze před potvrzením vedou koordinátor a účastníci následující dialog:

1. **Koordinátor.** Koordinátor nařídí každému účastnickému databázovému serveru, aby se připravil na potvrzení transakce.
2. **Účastníci.** Každý účastník sdělí koordinátorovi, zda může potvrdit svou větev transakce.
3. **Koordinátor.** Koordinátor na základě odezvy od každého účastníka se rozhodne transakci potvrdit nebo odvolat. Rozhodne se pro potvrzení pouze tehdy, pokud *všichni* účastníci oznamují, že mohou potvrdit svou větev transakce. Pokud některý z účastníků oznámí, že *není* připraven potvrdit svou větev transakce (nebo pokud neodpovídá), rozhodne se koordinátor tuto globální transakci ukončit.

Fáze po rozhodnutí

Během fáze po rozhodnutí koordinátor a účastníci vedou následující dialog:

1. **Koordinátor:** Koordinátor zapíše do logického protokolu koordinátora záznam o potvrzení nebo odvolání transakce a nařídí každému účastnickému databázovému serveru, aby transakci potvrdil nebo odvolal.

2. **Účastníci:** Pokud koordinátor vydal zprávu o potvrzení, potvrdí účastníci transakci zapsáním záznamu o potvrzení do logického protokolu a následně odešlou zprávu koordinátorovi, ve které mu potvrzují, že transakce byla potvrzena. Pokud koordinátor vydal zprávu o odvolání transakce, účastníci odvolají transakci, ale koordinátorovi neodesílají žádné potvrzení.
3. **Koordinátor:** Pokud koordinátor vydal zprávu o potvrzení transakce, čeká před ukončením globální transakce na obdržení potvrzení od každého účastníka. Jestliže koordinátor vydal zprávu o odvolání transakce, nečeká na potvrzení od účastníků.

Jak protokol dvoufázového potvrzování ošetřuje selhání

Protokol dvoufázového potvrzování je navržen k ošetření selhání systému a médií takovým způsobem, aby byla integrita dat zachována ve všech účastnických databázových serverech. Protokol dvoufázového potvrzování provádí v případě selhání *automatické obnovy*.

Typy selhání ošetřené automatickou obnovou

Následující události mohou způsobit ukončení nebo zastavení koordinačního nebo účastnického jednotkového procesu, a tudíž vyžadují automatickou obnovu:

- Selhání systému koordinátora.
- Selhání systému účastníka.
- Selhání sítě.
- Ukončení koordinačního jednotkového procesu administrátorem.
- Ukončení účastnického jednotkového procesu administrátorem.

Role administrátora v procesu automatické obnovy.

Jediná úloha, kterou má administrátor v automatické obnově, je uvést koordinátora nebo účastníka (případně oba) zpět do režimu online po selhání systému nebo sítě.

Důležité: Zpomalení sítě nemůže způsobit automatické zotavení, nebo by ho způsobit nemělo. Pokud nedojde k selhání systému koordinátora nebo k selhání sítě nebo pokud administrátor neukončí koordinační jednotkový proces, žádný ze zde popsaných mechanismů zotavení se nespustí.

Mechanismus automatického zotavení při selhání koordinátora

Pokud selže koordinační jednotkový proces, každý účastnický databázový server se musí rozhodnout, zda vyvolá automatické zotavení *předtím*, než potvrdí nebo odvolá transakci, nebo *poté*, co odvolá transakci. Tato zodpovědnost je součástí optimalizace předpokládaného konce. (Další informace naleznete v části “Optimalizace předpokládaného konce” na stránce 24-8.)

Mechanismus automatického zotavení při selhání účastníka

K zotavení účastníka dochází vždy, když účastnický jednotkový proces předběžně potvrdí položku práce, která byla ukončena předtím, než mohl být dokončen protokol dvoufázového potvrzování. Cílem zotavení účastníka je dokončit protokol dvoufázového potvrzování podle rozhodnutí, které učinil koordinátor.

Zotavení účastníka je vyvoláno koordinátorem nebo účastníkem, podle toho zda se koordinátor rozhodnul globální transakci potvrdit nebo odvolat.

Důležité: Kvůli podpoře automatického zotavení po tom, co je podřízený server vypnut nebo restartován, zatímco je otevřená transakce mezi servery, by měl soubor **sqlhosts** obsahovat záznam pro každý databázový server, ze kterého mohou být distribuované transakce zahájeny. Během procesu automatického zotavení je název koordinátora obnoven z logických protokolů a podřízený server za účelem dokončení transakce znovu naváže spojení s koordinátorem. Protože koordinátor se

vždy účastníkům identifikuje pomocí názvu, který je v konfiguračním parametru DBSERVERNAME ve vlastním souboru ONCONFIG, měl by koordinátorův parametr DBSERVERNAME být účastníkům známý název adresy protokolu Internetu. Podřízený server se by měl být schopen použitím koordinátorova parametru DBSERVERNAME připojit ke koordinátorovi.

Optimalizace předpokládaného konce

Optimalizace předpokládaného konce je pojem, který označuje způsob, kterým protokol dvoufázového potvrzení řeší odvolání transakce.

Odvolání transakce je zpracováváno následujícím způsobem. Když koordinátor určí, že transakce musí být odvolána, odešle všem účastníkům zprávu, aby odvolali svou část práce. Koordinátor nečeká na potvrzení této zprávy, ale pokračuje v uzavření transakce a v jejím odstranění ze sdílené paměti. Pokud se účastník pokusí určit stav této transakce - to znamená zjistit, zda byla tato transakce potvrzena nebo odvolána (například během zotavení účastníka) - nenalezne ve sdílené paměti žádný stav této transakce. Účastník si to pak tuto skutečnost interpretuje jako fakt, že byla transakce odvolána.

Nezávislé akce

Nezávislá akce je v kontextu dvoufázového potvrzení akce, která se uskuteční nezávisle na protokolu dvoufázového potvrzení. Nezávislé akce mohou, avšak zároveň nemusejí působit v protikladu k akcím, které určuje protokol dvoufázového potvrzení. Pokud akce *působí* v protikladu k protokolu dvoufázového potvrzení, bude výsledkem akce chyba nebo *heuristické rozhodnutí*. Heuristická rozhodnutí mohou mít za následek nekonzistenci databáze a vyžadují ruční obnovu po nezdařeném dvoufázovém potvrzení. Ruční obnova je velmi složitá administrativní procedura, které byste se měli vyhnout. (Další informace o procesu ruční obnovy uvádí Kapitola 25, "Ruční obnova při selhání dvoufázového potvrzení", na stránce 25-1.)

Situace, které vyvolávají nezávislé akce

Nezávislé akce jsou během protokolu dvoufázového potvrzení vzácné, ale mohou se vyskytnout v následujících situacích:

- Část práce účastníka se vyvine v chybu dlouhé transakce a je odvolána.
- Administrátor během fáze pozdního rozhodování protokolu pomocí příkazu **onmode -z** ukončí účastnický proces.
- Administrátor během fáze pozdního rozhodování protokolu pomocí příkazu **onmode -Z** ukončí účastnickou transakci.
- Administrátor pomocí příkazu **onmode -z** nebo **onmode -Z** ukončí globální transakci v databázovém serveru koordinátora *po* tom, co se koordinátor rozhodl pro potvrzení *a* byl uvědoměn o selhání účastníka. Tato akce vždy způsobí chybu, konkrétně způsobí chybu číslo -716.

Možné důsledky nezávislých akcí

Jak bylo zmíněno výše, nepůsobí všechny nezávislé akce v protikladu k protokolu dvoufázového potvrzení. Nezávislé akce mohou přinášet tyto možné výsledky:

- Úspěšné dokončení protokolu dvoufázového potvrzení.
- Chybový stav.
- Heuristické rozhodnutí.

Pokud akce nepůsobí v protikladu k dvoufázovému protokolu, měla by transakce být potvrzena nebo odvolána normálním způsobem. Pokud akce ukončí globální transakci předčasně, bude výsledkem chybový stav. Ukončení globální transakce u koordinátora není

považováno za heuristické rozhodnutí. Pokud akce působí v protikladu k protokolu dvoufázového potvrzování, bude výsledkem heuristické rozhodnutí. V následujících částech jsou podrobně probrány všechny uvedené situace.

Nezávislé akce, které umožní úspěšné dokončení transakcí

Nezávislé akce nepůsobí nezbytně v protikladu k protokolu dvoufázového potvrzování. Pokud je například část práce na účastnickém databázovém serveru odvolána, neboť se vyvinula v dlouhou transakci, a koordinátor vydá rozhodnutí odvolat globální transakci, databáze zůstane konzistentní.

Nezávislé akce, které vyústí v chybový stav

Pokud jako administrátoři koordinačního databázového serveru spustíte příkaz **onmode -z** (zastavit proces koordinátora) nebo příkaz **onmode -Z** (zastavit globální transakci) po tom, co koordinátor vydá konečné rozhodnutí *potvrdit*, odstraníte tak veškeré informace o transakci ze sdílené paměti databázového serveru.

Tato akce není považována za heuristické rozhodnutí, protože se nestřetává s dvoufázovým protokolem. Je buď přijatelná, nebo se střetává se zotavením účastníka a způsobí chybu.

Akce je přijatelná, pokud jsou všichni účastníci schopni bez potíží potvrdit transakci. V tomto případě je ruční vynucené ukončení transakce zbytečné. Oznámení o tom, že jste spustili příkaz **onmode -Z**, dojde ke koordinátorovi až v okamžiku, kdy se koordinátor bude připravovat na ukončení transakce.

V praxi to znamená, že budete pravděpodobně zvažovat spuštění příkazu **onmode -z** nebo **onmode -Z** na koordinačním databázovém serveru pouze tehdy, pokud budete chtít urychlit uzavření globální transakce, která zůstala otevřená po nezvykle dlouhou dobu. V takovém scénáři je zdrojem potíží pravděpodobně selhání na některém účastnickém databázovém serveru. Koordinátor neobdržel potvrzení, že účastník potvrdil svou část práce a pokusí se navázat komunikaci s účastníkem, aby mohl provést šetření.

Pokud spustíte příkaz **onmode -z** nebo **onmode -Z**, zatímco se koordinátor aktivně pokouší navázat komunikaci, uposlechně koordinační jednotkový proces požadavek na ukončení, ale předtím запиše do protokolu zpráv databázového serveru chybu číslo -716. Tato akce je považována za chybu, protože protokol dvoufázového potvrzování byl vynuceně přerušena a koordinátor proto nemohl určit, zda je databáze konzistentní.

Zastavení globální transakce v koordinační databázi není považováno za heuristické rozhodnutí, avšak může být příčinou nekonzistentní databáze. Pokud se například účastník nakonec vrátí do režimu online a nenalezne globální transakci ve sdílené paměti koordinátora, odvolá svou část práce, a tak způsobí nekonzistenci databáze.

Nezávislé akce, jejichž důsledkem je heuristické rozhodnutí

Některé nezávislé akce se mohou vyvinout v heuristické rozhodnutí, pokud jsou splněny *obě* následující podmínky:

- Účastnický databázový server již odeslal koordinátorovi zprávu lze potvrdit a potom transakci odvolá.
- Koordinátor se rozhodl potvrdit transakci.

Pokud jsou obě podmínky platné, bude čistým výsledkem globální transakce, která je nekonzistentně implementována (potvrzena jedním nebo více databázovými servery a odvolána jinými). Databáze se stane nekonzistentní.

Jsou možná tato heuristická rozhodnutí:

- Heuristické odvolání transakce (popsané v části “Scénář heuristického odvolání transakce” na stránce 24-10)
- Heuristické ukončení transakce (popsané v části “Scénář heuristického ukončení transakce” na stránce 24-12)

Po heuristickém odvolání nebo po konci transakce je možné, že bude potřeba provést ruční zotavení, složitý a na čas náročný proces. Abyste se mohli heuristickým rozhodnutím vyhnout, musíte jim zcela porozumět. Při spouštění příkazu **onmode -z** nebo **onmode -Z** v kontextu dvoufázového potvrzování buďte vždy obezřetní.

Scénář heuristického odvolání transakce

Při *heuristickém odvolání transakce* buď každý databázový server, nebo administrátor odvolá svou část práce, pro kterou již byla odeslána zpráva lze potvrdit.

Podmínky, jejichž výsledkem je heuristické odvolání transakce

Heuristické odvolání transakce mohou způsobit tyto dvě podmínky:

- Logický protokol se zaplní do bodu definovaného konfiguračním parametrem LTXEHWM. (Další informace naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix Administrator's Reference*.) Původcem stavu dlouhé transakce je část práce prováděná za globální transakci.
- Administrátor zadá příkaz **onmode -z ID realce** k zastavení procesu databázového serveru, který zpracovává část práce pro globální transakci.

V druhém případě, pokud tato část práce již odeslala svému koordinátorovi zprávu lze potvrdit, bude tato akce považována za heuristické rozhodnutí.

Podmínka 1: Logický protokol se zaplní do horní meze: Při dvoufázovém potvrzování je dokončení transakce v účastnickém databázovém serveru čekajícím na instrukci od koordinátora blokováno. Jelikož transakce zůstává otevřená, nemohou být uvolněny soubory logického protokolu obsahující záznamy přidružené k této transakci. Výsledkem je, že zaplňování tohoto logického protokolu pokračuje díky aktivitě souběžně pracujících uživatelů.

Pokud se během čekání účastníka logický protokol zaplní až k horní mezi dlouhé transakce (LTXHWM), databázový server nařídí všem jednotkovým procesům databázového serveru, které vlastní dlouhé transakce, aby tyto transakce odvolaly. Pokud problematickou dlouhou transakci představuje již předběžně potvrzená část práce, znamená to, že databázový server zahájil heuristické odvolání transakce. To znamená, že databázový server odvolává zpět předběžně potvrzenou část práce, aniž by obdržel instrukce nebo informace od koordinátora.

Při dvoufázovém potvrzování jsou soubory logického protokolu obsahující záznamy, které jsou přidruženy k části této práce, považované za otevřené, dokud není zapsán záznam logického protokolu ENDTRANS. Tento typ transakce se liší od transakce zahrnující jediný databázový server, ve kterém odvolání transakce skutečně transakci uzavře.

Logický protokol se může dále zaplňovat, dokud nedosáhne horní meze dlouhé transakce s výlučným přístupem (LTXEHWM). V takovém případě budou pozastaveny všechny uživatelské jednotkové procesy kromě těch, které aktuálně provádějí odvolání transakce nebo potvrzení transakce. Ve scénáři dvoufázového potvrzování brání otevřená transakce v zálohování souborů logického protokolu a v uvolnění prostoru v logickém protokolu. Za takovýchto konkrétních podmínek se logický protokol může zaplnit zcela. Pokud to nastane, databázový server se vypne a musíte provést obnovu dat.

Podmínka 2: Administrátor systému spustí příkaz onmode -z: Jako administrátor můžete rozhodnout o zahájení heuristického odvolání transakce předběžně potvrzené části práce

spuštěním příkazu **onmode -z**. Toto rozhodnutí můžete učinit, pokud budete chtít uvolnit zdroje zadržované touto částí práce. (Pokud zastavíte účastnický proces použitím příkazu **onmode -z**, uvolníte všechny zámky a prostředky sdílené paměti, které jsou zabrány uživatelským procesem i přesto, že neukončíte transakci.)

Výsledky heuristického odvolání transakce

Tato část popisuje, co nastane u koordinátora a účastníka v případě heuristického odvolání transakce a jak tento proces může způsobit, že databáze bude nekonzistentní:

1. Na účastnickém databázovém serveru, na kterém došlo k odvolání transakce, je do logického protokolu databázového serveru uložen záznam (typu HEURTX). Zámky a zdroje zadržované transakcí jsou uvolněny. Účastnický jednotkový proces zapíše do protokolu zpráv databázového serveru následující zprávu, která označuje, že se vyskytnul stav dlouhé transakce a že došlo k odvolání transakce:

```
Transaction Completed Abnormally (rollback):
```

```
tx=address flags=0xnn
```

2. Koordinátor ve fázi po rozhodnutí vydá instrukce, které transakci potvrdí.

Účastnický jednotkový proces na databázovém serveru, na kterém se vyskytnulo heuristické odvolání transakce, vrátí koordinátorovi následující chybovou zprávu -699:

```
-699 Transaction heuristically rolled back.
```

Tato chybová zpráva není v tomto okamžiku vracena aplikacím, jedná se o interní oznámení koordinátorovi. Koordinátor čeká, dokud všichni účastníci nezareagují na instrukci k potvrzení transakce. Koordinátor nezjišťuje konzistenci databáze, dokud se neohlásí všichni účastníci.

3. Následující kroky závisí na akcích, které se vyskytnou u ostatních účastníků. Jsou možné dvě situace.

Situace 1: Koordinátor vydá instrukci k potvrzení transakce a všichni účastníci oznámí heuristické odvolání transakce: Koordinátor shromáždí všechny odpovědi od účastníků. Pokud každý účastník vykáže heuristické odvolání transakce, budou důsledkem následující události:

1. Koordinátor zapíše do vlastního protokolu zpráv databázového serveru tuto zprávu:

```
Transaction heuristically rolled back.
```
2. Koordinátor odešle všem účastníkům zprávu, aby ukončili transakci.
3. Každý účastník zapíše do vyrovnávací paměti vlastního logického protokolu záznam ENDTRANS. (Záznam transakce je odstraněn z tabulky transakcí.)
4. Koordinátor vrátí aplikaci chybu -699 v tomto tvaru:

```
-699 Transaction heuristically rolled back.
```
5. V takové situaci zůstanou všechny databáze konzistentní.

Situace 2: Koordinátor vydá instrukci potvrdit, jeden účastník transakci potvrdí a jeden oznámí heuristické odvolání transakce: Koordinátor shromáždí všechny odpovědi od účastníků. Pokud alespoň jeden účastník oznámí heuristické odvolání transakce a alespoň jeden účastník potvrdí potvrzení transakce, je výsledek označován jako *smíšený výsledek transakce*. V důsledku této situace nastanou následující události:

1. Koordinátor zapíše do vlastního protokolu zpráv databázového serveru tuto zprávu:

```
Mixed transaction result. (pid=nn user=id_uživatele)
```

Hodnota `pid` je identifikační číslo uživatelského procesu pro koordinační proces. Hodnota `user` je ID uživatele přidružené ke koordinačnímu procesu. K této zprávě jsou přidruženy další zprávy, které budou uvedeny na každém účastnickém databázovém serveru, který nahlásil heuristické odvolání transakce. Tyto dodatečné zprávy mají následující formát:

```
Participant database server název_dbservername heuristically rolled back.
```

2. Koordinátor odešle každému účastníkovi, který heuristicky odvolal svou část práce, zprávu, ve které jim přikazuje, aby ukončili transakci.
3. Každý účastník запиše do vyrovnávací paměti logického protokolu zprávu ENDTRANS. (Záznam transakce je odstraněn z tabulky transakcí.)
4. Koordinátor запиše do vyrovnávací paměti logického protokolu zprávu ENDTRANS. (Záznam transakce je odstraněn z tabulky transakcí sdílené paměti.)
5. Koordinátor vrátí tímto způsobem aplikaci chybu -698:
-698 Inconsistent transaction.
Počet a názvy serverů, které byly navráceny zpět. .
6. K této chybové zprávě je přidružen seznam účastnických databázových serverů, které nahlásily heuristické odvolání transakce. Pokud transakci odvolal velký počet databázových serverů, může být tento seznam zkrácen. Úplný seznam je vždy k dispozici v protokolu zpráv na koordinačním databázovém serveru.

V této situaci prozkoumejte logický protokol v každém místě databázových serverů a určete, zda je databázový systém konzistentní. (Další informace naleznete v části "Určení, zda byla transakce implementována nekonzistentně" na stránce 25-1.)

Scénář heuristického ukončení transakce

Heuristické ukončení transakce je nezávislá akce, kterou provádí administrátor, aby odvolal část práce a odstranil veškeré informace o transakci z tabulky transakcí. Proces heuristického ukončení transakce je zahájen, když administrátor spustí příkaz **onmode -Z address**.

Pokud vyvoláte heuristické ukončení transakce spuštěním příkazu **onmode -Z**, odstraníte tím kritické informace, které databázový server potřebuje pro podporu protokolu dvoufázového potvrzování a jeho funkci automatického zotavení. Pokud spustíte příkaz **onmode -Z**, budete pak sami zodpovědni za určení, zda je váš síťový databázový systém konzistentní.

Kdy provést heuristické ukončení transakce

Heuristické ukončení transakce příkazem **onmode -Z** byste měli spouštět pouze ve výjimečných situacích. Taková situace nastane, když část práce, která byla heuristicky odvolána, zůstane otevřená a zabraňuje uvolnění souborů logického protokolu. Důsledkem je, že se logický protokol nebezpečně přiblíží úplnému zaplnění.

Obecně platí, že koordinátor vydává rozhodnutí o potvrzení nebo odvolání transakce v rozumně krátké době. Pokud ovšem koordinátor selže a nevrátí se do režimu online, aby mohl ukončit transakci, která byla na účastnickém databázovém serveru heuristicky odvolána, mohou nastat velmi závažné potíže.

Scénář potíží začíná tímto způsobem:

1. Účastnický jednotkový proces, který vykonává část práce za globální transakci, odeslal koordinátorovi zprávu `can commit` (lze potvrdit).
2. Část práce čeká na pokyny od koordinátora.
3. Zatímco část práce čeká, logický protokol se zaplní nad horní mez dlouhé transakce.
4. Část práce čekající na pokyny je původcem dlouhé transakce. Účastnický databázový server nařídí vykonávanému jednotkovému procesu odvolat část práce. Tato akce se nazývá heuristické odvolání transakce.
5. Účastník poté stále čeká, až koordinátor určí, jak má účastník ukončit transakci. Transakce zůstává otevřená. Logický protokol se stále zaplňuje.

Pokud koordinátor v rozumném časovém intervalu kontaktuje účastníka a nařídí mu ukončit transakci, nenastane žádný problém. K vážným potížím dojde, pokud na účastnickém

databázovém serveru dojde k heuristickému odvolání transakce a následně selže koordinátor. Porucha zabrání koordinátorovi sdělit účastníkovi, jak má ukončit transakci.

V důsledku toho zůstane transakce otevřená. Otevřená transakce brání v zálohování souborů logického protokolu a v uvolnění prostoru logického protokolu. Jak se bude logický soubor neustále zaplňovat, může dosáhnout bodu určeného horní mezí dlouhé transakce s výlučným přístupem (LTXEHWM). Při dosažení tohoto bodu je normální zpracování pozastaveno. Po dosažení horní meze se musíte rozhodnout, zda otevřená transakce ohrožuje logický protokol. Nebezpečí spočívá v tom, že pokud se logický protokol zaplní zcela, databázový server se vypne a budete muset provést obnovu dat.

Musíte se rozhodnout, zda ukončíte transakci, abyste ochránili systém před možným zaplněním logického protokolu bez ohledu na všechny problémy související se spuštěním příkazu **onmode -Z**, nebo zda počkáte, abyste zjistili, zda se ještě podaří včas obnovit komunikaci s koordinátorem, který transakci ukončí, než dojde k zaplnění logického protokolu.

Jak používat příkaz **onmode -Z**

Příkaz **onmode -Z adresa** je vhodné použít pouze v situaci, kdy je komunikace mezi koordinátorem a účastníkem přerušena. Pokud chcete zkontrolovat, zda je komunikace skutečně přerušena, nepouštějte příkaz **onmode -Z**, dokud jednotkový proces, který prováděl část práce, není v nečinnosti a bez odezvy (dead) po dobu určenou parametrem **TXTIMEOUT**. Další informace o této možnosti naleznete v části *IBM Informix Administrator's Reference*.

Parametr *adresa* získáte z výstupu příkazu **onstat -x**. Další informace o možnosti **onstat -x** naleznete v části *IBM Informix Administrator's Reference*.

Akce při heuristickém ukončení transakce

Když spustíte příkaz **onmode -Z**, nařídíte tím obslužnému programu **onmode** odstranit z tabulky transakcí účastníka záznam transakce, který je umístěný na určené adrese.

Tuto akci dokumentují dva záznamy zapsané do logického protokolu. Jedná se o záznamy typu **ROLLBACK** a **ENDTRANS**, nebo v případě heuristického odvolání transakce pouze typu **ENDTRANS**. Do protokolu zpráv účastnického databázového serveru je zapsána následující zpráva:

(časová_značka) Transaction Completed Abnormally (endtx): tx=*adresa* flags:0xnn
user *jméno_uživatele* tty *ttyid*

Koordinátor obdrží chybovou zprávu od účastníka, ve kterém byl spuštěn příkaz **onmode -Z**, jako odezvu na svou instrukci **COMMIT** (potvrdit). Koordinátor se dotáže účastnického databázového serveru, který již nemá dále informace o transakci. Skutečnost, že v tabulce transakcí na databázovém serveru není příslušný záznam, označuje, že byla transakce potvrzena. Koordinátor předpokládá, že od účastníka byla odeslána potvrzující zpráva, ale že ji z nějakého důvodu neobdržel. Protože koordinátor *neví*, že účastnická část práce nebyla potvrzena, nevytvoří zprávu oznamující, že globální transakce byla implementována nekonzistentně. Pouze administrátor, který spustil příkaz **onmode -Z**, si je vědom nekonzistentní implementace.

Monitorování globální transakce

Pomocí příkazu **onstat -x** můžete zaznamenávat otevřené transakce a určovat, zda byly heuristicky odvolány.

Příznak H v poli **flags** například identifikuje heuristické odvolání, příznak G identifikuje globální transakci, příznak L označuje volně provázaný režim a příznak T označuje provázaný režim.

Pole **curlog** a **logposit** poskytují přesnou pozici záznamu logického protokolu. Pokud transakce není odvolávána, popisují pole **curlog** a **logposit** pozici nejnovějšího zapsaného záznamu protokolu. Pokud je transakce odvolávána, popisují tato pole pozici nejnovějšího “vráceného” (undone) záznamu protokolu. Jak je transakce odvolávána, hodnoty **curlog** a **logposit** se snižují. V případě dlouhé transakce vám může rychlost, s níž se hodnoty **logposit** a **beginlg** sblíží, pomoci odhadnout, jak dlouho bude odvolání transakce trvat.

Další informace a příklad výstupu příkazu **onstat -x** naleznete v části *IBM Informix Administrator's Reference*.

Rovněž můžete pomocí příkazů **onstat -u** a **onstat -k** zaznamenávat transakce a zámky, které drží. Další podrobnosti naleznete v části týkající se monitorování transakcí v příručce *Řízení výkonu systému IBM Informix*. Popis polí zobrazovaných příkazem **onstat -x** naleznete v části *IBM Informix Administrator's Reference*.

Chyby protokolu dvoufázového potvrzování

Následující chyby protokolu dvoufázového potvrzování vyžadují zvláštní pozornost administrátora.

Číslo chyby	Popis
-698	Pokud obdržíte chybu -698, došlo k heuristickému odvolání transakce, což způsobilo nekonzistentně implementovanou transakci. Okolnosti vedoucí k této události jsou popsány v části “Výsledky heuristického odvolání transakce” na stránce 24-11. Další informace o tom, jak se vyvinula nekonzistentní transakce a jaké máte možnosti, naleznete v této části příručky.
-699	Pokud obdržíte chybu -699, došlo k heuristickému odvolání transakce. Okolnosti vedoucí k této události jsou popsány v části “Výsledky heuristického odvolání transakce” na stránce 24-11. Další informace o tom, jak se vyvinula nekonzistentní transakce, naleznete v této části příručky.
-716	Pokud obdržíte chybu -716, byl koordinující jednotkový proces ukončen administrátorem poté, co vydal konečné rozhodnutí. Tento scénář je popsán v části “Nezávislé akce, které vyústí v chybový stav” na stránce 24-9.

Dvoufázové potvrzování a záznamy logického protokolu

Databázový server k implementaci protokolu dvoufázového potvrzování používá záznamy logického protokolu. Tyto záznamy logického protokolu můžete použít ke zjištění heuristických rozhodnutí a v případě nutnosti vám pomohou provést ruční obnovu. (Další informace uvádí Kapitola 25, “Ruční obnova při selhání dvoufázového potvrzování”, na stránce 25-1.)

Do distribuovaných transakcí jsou zapojeny následující záznamy logického protokolu:

- BEGPREP
- PREPARE
- TABLOCKS
- HEURTX
- ENDTRANS

Další informace o těchto záznamech logického protokolu naleznete v kapitole o interpretaci záznamů logického protokolu v příručce *IBM Informix Administrator's Reference*.

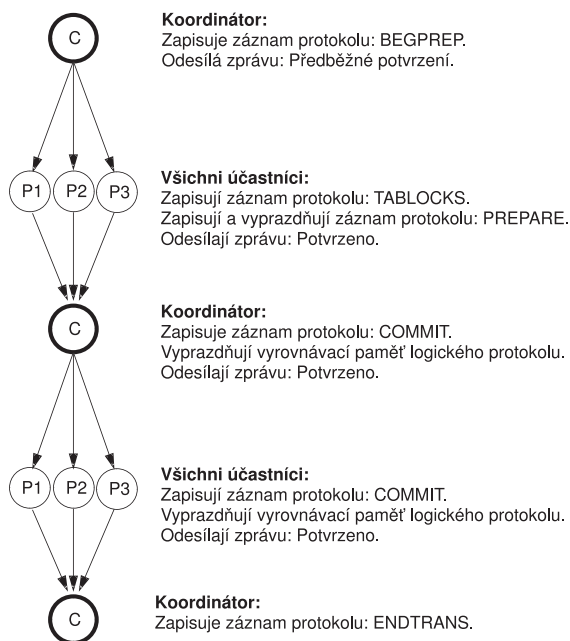
V této části je popsáno pořadí záznamů logického protokolu, které se zapisují během následujících scénářů databázového serveru:

- Transakce je potvrzena.
- Část práce je heuristicky odvolána.
- Část práce je heuristicky ukončena.

Záznamy logického protokolu při potvrzení transakce

Obrázek 24-3 ilustruje pořadí zápisu záznamů logického protokolu během úspěšného protokolu dvoufázového potvrzování, který vyústí v potvrzenou transakci.

Zahájení protokolu



Ukončení protokolu

Obrázek 24-3. Záznamy logického protokolu zapsané během potvrzené transakce

Některé záznamy logického protokolu musí být vyprázdněny z logického protokolu okamžitě, zatímco vyprázdnění jiných není životně důležité.

Záznam koordinátora o potvrzení (záznam COMMIT) obsahuje všechny informace potřebné ke spuštění protokolu dvoufázového potvrzování. Slouží také jako počáteční bod pro automatické zotavení v případě selhání hostitelského počítače koordinátora. Jelikož je tento záznam kritický pro zotavení, není povoleno ponechávat jej ve vyrovnávací paměti logického protokolu. Koordinátor musí záznam logického protokolu COMMIT okamžitě vyprázdnit.

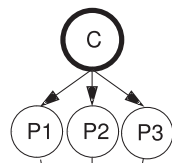
Účastníci, které uvádí Obrázek 24-3 na stránce 24-15, musí okamžitě vyprázdnit záznamy logického protokolu PREPARE a COMMIT. Vyprázdnění záznamu PREPARE zajišťuje, že v případě poruchy hostitelského počítače účastníka bude rychlá obnova schopná zjistit, že je tento účastník částí globální transakce. Jako část zotavení se bude moci účastník dotázat koordinátora a zjistit tak konečný stav transakce.

Vyprázdnění účastníkového záznamu COMMIT zajistí, že v případě, že by hostitelský počítač účastníka selhal, bude mít účastník záznam o tom, jakou akci v souvislosti s transakcí provedl. Abyste lépe pochopili zásadní význam této informace, zvažte situaci, ve které se dojde k selhání účastníka po zapsání záznamu PREPARE, ale před vyprázdněním záznamu COMMIT. Po rychlé obnově bude záznam PREPARE obnoven, ale záznam COMMIT bude ztracen (protože byl v okamžiku selhání ve vyrovnávací paměti logického protokolu). Existence záznamu PREPARE vyvolá dotaz na koordinátora, který se bude týkat transakce. Koordinátor ovšem nebude vědět o transakci nic, protože ji ukončil poté, co obdržel účastníkovou potvrzení, že došlo k potvrzení transakce. V této situaci bude účastník interpretovat nedostatek informací jako konečný důvod k odvolání transakce. Protokol dvoufázového potvrzování vyžaduje, aby byl účastníkův záznam COMMIT okamžitě vyprázdněn, aby se tak zabránilo takovému nedorozumění.

Záznamy logického protokolu zapsané během heuristického odvolání transakce

Obrázek 24-4 na stránce 24-17 ilustruje pořadí, ve kterém databázový server zapisuje záznamy logického protokolu během heuristického odvolání transakce. Jelikož k heuristickému odvolání transakce dochází pouze poté, co účastník odešle zprávu, že může potvrdit transakci, a koordinátor odešle zprávu nařizující transakci potvrdit, je první fáze tohoto protokolu stejná jako situace, kterou uvádí Obrázek 24-3 na stránce 24-15. Když dojde k heuristickému odvolání transakce, předpokládá se, že odvolání transakce je důsledkem stavu dlouhé transakce, který nastal na databázovém serveru Účastníka 1 (P1). Konečným výsledkem je nekonzistentně implementovaná transakce. Další informace naleznete v části “Scénář heuristického odvolání transakce” na stránce 24-10.

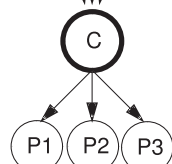
Zahájení protokolu



Koordinátor:
Zapisuje záznam protokolu: BEGPREP.
Odesílá zprávu: Předběžné potvrzení.

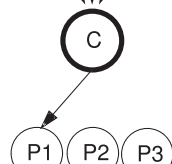
Všichni účastníci:
Zápis záznamu protokolu: TABLOCKS.
Zápis záznamu protokolu: PREPARE.
Záznam protokolu je vyprázdněn.
Je odeslána zpráva: Lze provést potvrzení.

V prostředí účastníka P1:
Databázový server zjistil, že probíhá dlouhá transakce. Je zahájeno odvolání.
Zapisuje záznam protokolu: HEURTX.
Zapisuje záznam protokolu: ROLLBACK.
Do protokolu zpráv je zapsána zpráva.



Koordinátor:
Zapisuje záznam protokolu: COMMIT.
Vyprazdňuje záznam protokolu.
Odesílá zprávu: Potvrzeno.

Účastník 1:
Odesílá zprávu: Transakce byla heuristicky odvolána. Nelze potvrdit.
Účastníci 2 a 3:
Zápis a vyprázdnění záznamu protokolu: COMMIT.
Odesílají zprávu: Potvrzeno.



Koordinátor:
Zapisuje zprávu do protokolu zpráv (-698).
Odesílá zprávu účastníkovi 1: Konec transakce.

Účastník 1:
Zapisuje záznam protokolu: ENDTRANS.
Odesílá zprávu: Transakce byla ukončena.



Koordinátor:
Zapisuje záznam protokolu: ENDTRANS.
Vrací uživateli chybovou zprávu: Chyba -698.

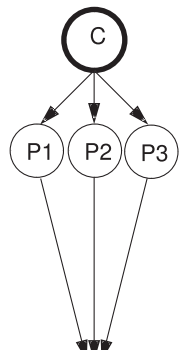
Ukončení protokolu

Obrázek 24-4. Záznamy logického protokolu zapsané během heuristického odvolání transakce

Záznamy logického protokolu zapsané během heuristického ukončení transakce

Obrázek 24-5 na stránce 24-18 ilustruje pořadí zápisu záznamů logického protokolu během heuristického ukončení transakce. Tato událost je vždy důsledkem ukončení transakce na účastnickém databázovém serveru, které provedl administrátor (další informace naleznete v části týkající se příkazu **onmode** v příručce *IBM Informix Administrator's Reference*) poté, co účastník odeslal zprávu **can commit** (lze potvrdit). Obrázek 24-5 ilustruje situaci, kdy se předpokládá, že k heuristickému ukončení transakce došlo na databázovém serveru Účastníka 1 (P1). Výsledkem bude nekonzistentně implementovaná transakce. Další informace naleznete v části “Scénář heuristického ukončení transakce” na stránce 24-12.

Zahájení protokolu

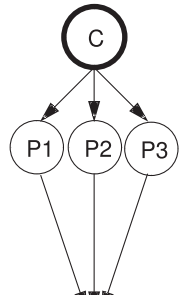


Koordinátor:

Zapisuje záznam protokolu: BEGPREP.
Odesílá zprávu: Předběžné potvrzení.

Všichni účastníci:

Zapisují záznam protokolu: TABLOCKS.
Zapisují a vyprazdňují záznam protokolu: PREPARE.
Odesílají zprávu: Potvrzeno.
Prostředí účastníka P1:
Transakce byla přerušena.
Zapisuje záznam protokolu: ROLLBACK.
Zapisuje záznam protokolu: ENDTRANS.
Do protokolu zpráv je zapsána zpráva.



Koordinátor:

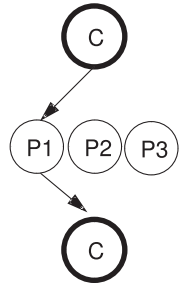
Zapisuje záznam protokolu: COMMIT.
Vyprazdňuje vyrovnávací paměť logického protokolu.
Odesílá zprávu: Potvrzeno.

Účastník 1:

Vrací chybovou zprávu.

Účastníci 2 a 3:

Zapisují záznam protokolu: COMMIT.
Vyprazdňují vyrovnávací paměť logického protokolu.
Odesílají zprávu: Potvrzeno.



Koordinátor:

Přijímá chybovou zprávu z počítače P1.
Vytváří nové připojení k počítači P1 a odesílá mu zprávu TX Inquire.

Účastník 1:

Odesílá koordinátorovi zprávu o neznámém stavu transakce.

Koordinátor:

Předpokládá, že neznámý stav znamená potvrzení.
Zapisuje záznam protokolu: ENDTRANS.

Ukončení protokolu

Obrázek 24-5. Záznamy logického protokolu zapsané během heuristického ukončení transakce

Konfigurační parametry používané při dvofázovém potvrzování

Pro distribuované prostředí jsou specifické následující dva konfigurační parametry:

- DEADLOCK_TIMEOUT
- TXTIMEOUT

Ačkoli oba parametry určují doby prodlevy, jsou vzájemně nezávislé. Více informací o těchto konfiguračních parametrech naleznete v příručce *IBM Informix Administrator's Reference*.

Funkce parametru DEADLOCK_TIMEOUT

Pokud je distribuovaná transakce nucena čekat na zdroj ve sdílené paměti déle než po dobu určenou v sekundách parametrem DEADLOCK_TIMEOUT, bude jednotkový proces vlastníci tuto transakci předpokládat, že na více serverech došlo k zablokování. Bude vrácena tato chybová zpráva:

-154 ISAM error: deadlock timeout expired - Possible deadlock.

Výchozí hodnota konfiguračního parametru `DEADLOCK_TIMEOUT` je 60 sekund. Tuto hodnotu upravujte opatrně. Pokud tuto hodnotu nastavíte příliš malou, jednotlivé databázové servery ukončí transakce, které nejsou zablokovány. Pokud bude nastavena příliš velká hodnota, mohla by zablokování na více serverech narušovat souběžné zpracování.

Funkce parametru `TXTIMEOUT`

Konfigurační parametr `TXTIMEOUT` je specifický pro protokol dvoufázového potvrzování. Používá se pouze tehdy, pokud byla komunikace mezi koordinátorem a účastníkem narušena a musí být znovu navázána.

Konfigurační parametr `TXTIMEOUT` určuje časový interval, po který účastnický databázový server během distribuované transakce čeká na obdržení instrukce `commit` od koordinačního databázového serveru. Pokud časový interval určený v konfiguračním parametru `TXTIMEOUT` vyprší, účastnický databázový server zkontroluje stav transakce, aby zjistil, zda má účastník zahájit automatické zotavení účastníka.

Konfigurační parametr `TXTIMEOUT` je zadáván v sekundách. Výchozí hodnota je 300 (pět minut). Optimální hodnota tohoto parametru se liší v závislosti na konkrétním prostředí a aplikaci. Předtím, než se rozhodnete změnit tento parametr, přečtěte si informace uvedené v části "Jak protokol dvoufázového potvrzování ošetřuje selhání" na stránce 24-7.

Protokol dvoufázového potvrzování

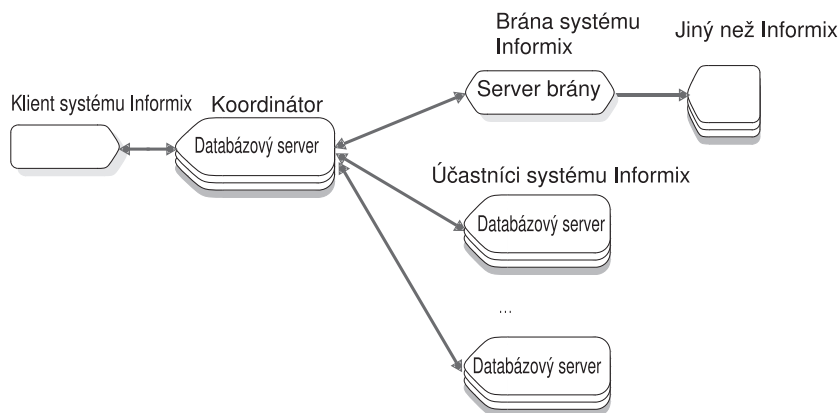
Termín heterogenní prostředí používaný v souvislosti s databázovými servery Informix označuje skupinu databázových serverů, z nichž alespoň jeden není databázový server Informix. Heterogenní potvrzování zajišťuje, že se distribuované transakce v heterogenním prostředí řídí principem "všechno nebo nic".

Na rozdíl od protokolu dvoufázového potvrzování podporuje protokol heterogenního potvrzování účast účastníků jiných než Informix. Účastník jiný než Informix, nazývaný účastník prostřednictvím brány, musí komunikovat s koordinátorem prostřednictvím brány IBM Informix.

Databázový server používá protokol heterogenního potvrzování v případě, že jsou splněny následující podmínky:

- Heterogenní potvrzování je povoleno. (To znamená, že konfigurační parametr `HETERO_COMMIT` je nastaven na hodnotu 1.)
- Koordinátor potvrzování je server IBM Informix Dynamic Server verze 7.2 nebo novější.
- Účastník jiný než Informix komunikuje s databázovým serverem Informix prostřednictvím brány systému Informix.
- Aktualizaci provádí v rámci jediné transakce nanejvýš jeden účastník jiný než Informix.

Tento scénář ilustruje Obrázek 24-6.



Obrázek 24-6. Konfigurace, která vyžaduje heterogenní potvrzení distribuovaných transakcí.

Brány, které se mohou účastnit heterogenního potvrzení transakcí

Brána IBM Informix funguje jako most mezi aplikací Informix (v tomto případě databázovým serverem) a databázovým serverem jiným než IBM Informix. Brána IBM Informix umožňuje používat aplikace IBM Informix k přístupu k datům a ke změnám dat, která jsou uložena v databázích jiných než Informix.

Následující tabulka uvádí seznam bran a odpovídajících databázových serverů, které se mohou účastnit v transakci, ve které databázový server používá protokol heterogenního potvrzení.

Tabulka 24-1. Brány a odpovídající databázové servery/heterogenní potvrzení transakcí

Brána	Databázové servery
IBM Informix Enterprise Gateway with DRDA	IBM DB2, OS/400, SQL/DS
IBM Informix Enterprise Gateway for EDA/SQL	EDA/SQL
IBM Informix Enterprise Gateway Manager	Jakýkoli databázový server s propojitelností ODBC

Povolení a zakázání heterogenního potvrzení

Pomocí textového editoru nebo programu ISA můžete změnit konfigurační parametr `HETERO_COMMIT`, který povoluje nebo zakazuje heterogenní potvrzení. Změny budou platné po vypnutí a restartování databázového serveru.

Pokud nastavíte konfigurační parametr `HETERO_COMMIT` na hodnotu 1, bude koordinátor transakcí zjišťovat, zda se jedná o distribuované transakce vyžadující použití heterogenního potvrzení. Když koordinátor zjistí takovou transakci, automaticky provede protokol heterogenního potvrzení.

Pokud nastavíte konfigurační parametr `HETERO_COMMIT` na hodnotu 0 nebo libovolné číslo jiné než 1, koordinátor transakcí protokol heterogenního potvrzení zakáže. Následující tabulka podává souhrnný přehled o tom, jaký protokol koordinátor transakcí použije, zda heterogenní potvrzení nebo dvoufázové potvrzení, aby zajistil integritu distribuované transakce.

Nastavení konfiguračního parametru HETERO_COMMIT	Účastník prostřednictvím brány je aktualizován	Protokol databázového serveru
Zakázán	Ne	dvoufázové potvrzování
Zakázán	Ano	dvoufázové potvrzování
Povolen	Ne	dvoufázové potvrzování
Povolen	Ano	heterogenní potvrzování

Jak funguje heterogenní potvrzování

Protokol heterogenního potvrzování je pozměněnou verzí standardního protokolu dvoufázového potvrzování. Fáze po rozhodnutí v protokolu heterogenního potvrzování je stejná jako fáze po rozhodnutí v protokolu dvoufázového potvrzování. Fáze před potvrzením obsahuje malé změny, a dále je k protokolu heterogenního potvrzování přidána nová fáze, nazvaná fáze potvrzení brány.

V následujících částech je popsána změna fáze před potvrzením a fáze potvrzení brány. Podrobné vysvětlení fází po rozhodnutí naleznete v části "Fáze po rozhodnutí" na stránce 24-6.

Fáze před potvrzením

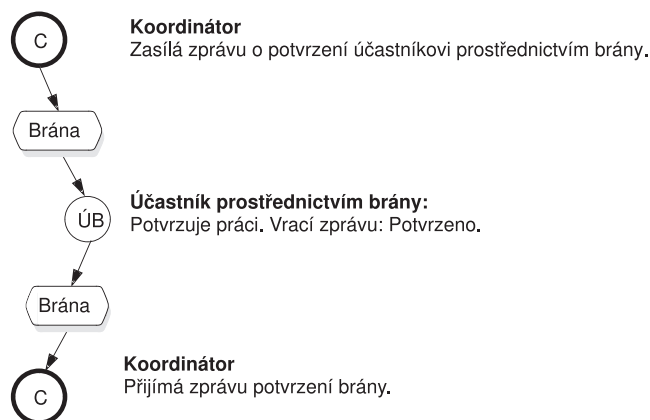
Koordinátor nařídí každému účastníkovi (kromě účastníka brány), aby se připravil na potvrzení transakce.

Pokud aktualizace vyhovují všem odloženým omezením, všichni účastníci (kromě účastníka brány) vrátí koordinátorovi zprávy oznamující, že mohou potvrdit svou část práce.

Fáze potvrzení brány

Pokud všichni účastníci úspěšně vrátí zprávu oznamující, že jsou připraveni potvrdit svou část práce, odešle koordinátor zprávu "commit" (potvrdit) bráně. Brána naopak odešle koordinátorovi zprávu oznamující, zda potvrzuje svou část transakce. Pokud brána potvrdí transakci, rozhodne se koordinátor potvrdit celou transakci. Tento proces ilustruje Obrázek 24-7 na stránce 24-21.

Zahájení fáze potvrzení prostřednictvím brány



Konec fáze potvrzení prostřednictvím brány

Obrázek 24-7. Fáze heterogenního potvrzování, jejímž výsledkem je potvrzená transakce

Pokud brána selže při potvrzení transakce, koordinátor odvolá celou transakci, tak jak to ilustruje Obrázek 24-7.

Optimalizace heterogenního potvrzování

Pokud jediným účastníkem, který obdrží aktualizaci, je databáze jiná než Informix, bude databázový server protokol heterogenního potvrzování optimalizovat. V takovém případě koordinátor odešle všem účastníkům jedinou zprávu "commit" (potvrdit), aniž by použil protokol heterogenního potvrzování.

Důsledky selhání heterogenního potvrzování

V každém okamžiku během distribuované transakce, kdy databázový server používá heterogenní potvrzování, může koordinátor nebo jakýkoli počet účastníků selhat. Databázový server řeší tato selhání stejným způsobem jako u protokolu dvoufázového potvrzování, kromě určitých případů. Následující části se podrobně zabývají těmito speciálními případy.

Selhání koordinačního databázového serveru

Konzistence dat po selhání koordinátora závisí na bodu v procesu heterogenního potvrzování, ve kterém koordinátor selhal. Pokud koordinátor selže předtím, než bráně odešle zprávu "commit" (potvrdit), bude celá akce při zotavení zastavena, jako je tomu v případě dvoufázového potvrzování.

Pokud koordinátor selže po zapsání potvrzovacího záznamu protokolu, celá akce bude při zotavení úspěšně potvrzena, jako je tomu v případě dvoufázového potvrzování.

Pokud koordinátor selže po odeslání zprávy "commit" (potvrdit) bráně, ale před zapsáním potvrzovacího záznamu protokolu, budou při zotavení vzdálené databázové servery Informix účastníci se transakce zastaveny. To může mít za následek nekonzistence, pokud brána obdržela zprávu "commit" (potvrdit) a transakci potvrdila.

Tyto scénáře jsou shrnuty v následující tabulce:

Bod selhání koordinačního databázového serveru	Očekávaný výsledek
Poté, co koordinátor zapíše záznam protokolu PREPARE (připravit), a před fází potvrzení brány.	Konzistence dat je zachována.
Poté, co koordinátor odešle bráně zprávu "commit" (potvrdit), ale dříve, než obdrží odpověď.	Data jsou pravděpodobně konzistentní. Koordinátor neoznámí, zda mohlo dojít ke vzniku nekonzistence dat.
Po fázi potvrzení brány, ale předtím, než koordinátor zapíše záznam COMMIT (potvrzeno) do logického protokolu.	Konzistence dat je ztracena. Koordinátor neoznámí, že došlo ke vzniku nekonzistence dat.

Selhání účastníka

Pokud účastník při distribuované transakci používající protokol heterogenního potvrzování selže, odešle koordinátor tuto chybovou zprávu:

-441 Možná nekonzistence dat na cílovém systému DBMS *název* z důvodu přerušení potvrzení.

Navíc odešle databázový server do protokolu zpráv tuto zprávu:

Zdroj dat, ke kterému bylo přístupováno prostřednictvím brány *název*, může být v nekonzistentním stavu.

Selhání účastníka se neomezuje pouze na selhání databázového serveru nebo brány. Selhání komunikačního propojení mezi koordinátorem a bránou je také považováno za selhání brány. Brána se ukončí, pokud dojde k selhání propojení. Brána se musí ukončit, protože neudrží

transakční protokol, a není proto schopna znovu vytvořit spojení s koordinátorem a pokračovat v transakci. Kvůli těmto omezením existují některé scénáře, ve kterých mohou při selhání brány data zůstat v nekonzistentním stavu. Tyto scénáře jsou shrnuty v následující tabulce:

Bod selhání účastníka	Očekávaný výsledek
Poté, co účastník obdrží od koordinátora zprávu <code>commit transaction</code> (potvrdit transakci), ale předtím, než účastník provede potvrzení.	Konzistence dat je zachována.
Poté, co účastník obdrží od koordinátora zprávu <code>commit transaction</code> (potvrdit transakci) a transakci potvrdí, ale předtím, než účastník odpoví koordinátorovi.	Data jsou nekonzistentní.
Poté, co účastník potvrdí transakci a odešle koordinátorovi odpověď.	Pokud komunikační propojení selže předtím, než koordinátor obdrží odpověď, budou data nekonzistentní. Pokud koordinátor obdrží odpověď, budou data konzistentní (za předpokladu, že koordinátor neseleže před zapsáním záznamu COMMIT (potvrzení)).

Postup zotavení, kterým se databázový server po selhání účastníka řídí, je stejný jako postup při dvoufázovém potvrzování. Další informace o tomto postupu naleznete v části "Selhání účastníka" na stránce 24-22.

Interpretace chybových zpráv heterogenního potvrzování

Pokud databázový server selže při zpracování distribuované transakce používající heterogenní potvrzování, vrátí jednu ze dvou chybových zpráv, které jsou popsány v následujících částech.

Aplikace se pokouší aktualizovat několik účastníků brány: Pokud se klientská aplikace pokouší aktualizovat data ve více než jednom účastníku brány, když je konfigurační parametr `HETERO_COMMIT` nastaven na hodnotu 1, vrátí koordinátor tuto chybovou zprávu:

-440 Cannot update more than one non-Informix DBMS within a transaction.

Pokud obdržíte tuto chybovou zprávu, přepište příslušnou aplikaci tak, aby v jediné distribuované transakci aktualizovala maximálně jednoho účastníka brány.

Selhání pokusu potvrdit distribuovanou transakci pomocí heterogenního potvrzování:

Databázovému serveru se nemusí potvrzení distribuované transakce s použitím protokolu heterogenního potvrzování podařit z jedné nebo několika následujících příčin:

- chyba komunikace,
- selhání počítače,
- selhání brány,
- jiná neznámá chyba.

Pokud nastane takové selhání, vrátí koordinátor tuto zprávu:

-441 Possible inconsistent data at the target DBMS *název* due to an aborted commit.

Poté, co databázový server odešle tuto zprávu, odvolá transakci ve všech aktualizovaných počítačích zúčastněných v transakci. Výjimkou může být práce provedená v počítači účastníka brány. Účastník brány mohl potvrdit své aktualizace, pokud k selhání došlo poté, co zprávu "commit" (potvrdit) zpracoval. Jestliže účastník brány aktualizace potvrdil, budete muset tyto aktualizace odvolat ručně.

Kapitola 25. Ruční obnova při selhání dvoufázového potvrzování

Obsah kapitoly	25-1
Určení, zda je požadována ruční obnova	25-1
Určení, zda byla transakce implementována nekonzistentně	25-1
Předčasné ukončení globální transakce	25-1
Heuristické ukončení transakce	25-2
Heuristické odvolání transakce	25-2
Určení, zda distribuovaná databáze obsahuje nekonzistentní data	25-2
Získání informací z logického protokolu	25-3
Získání identifikátoru globální transakce	25-3
Rozhodnutí, zda je nezbytné k nápravě situace provést akci	25-4
Příklad ruční obnovy	25-5

Obsah kapitoly

Distribuované transakce se řídí protokolem dvoufázového potvrzování. Určité akce mohou nastat nezávisle na protokolu dvoufázového potvrzování a mohou způsobit nekonzistentní implementaci transakce. (Další informace naleznete v části “Nezávislé akce” na stránce 24-8.) V takových situacích může být nezbytná ruční obnova z transakce.

V této kapitole jsou popsána následující témata:

- Určení, zda je nezbytná ruční obnova z nekonzistentně implementované transakce při dvoufázovém potvrzování.
- Provedení ruční obnovy.

Určení, zda je požadována ruční obnova

Následující seznam poskytuje přehled kroků při zjišťování konzistence databáze a při nápravě situace, je-li to nezbytné.

Postup zjištění konzistence databáze:

1. Určete, zda byla transakce implementována nekonzistentně.
2. Určete, zda síťový databázový systém obsahuje nekonzistentní data.
3. Určete, zda je nezbytná akce pro nápravu situace.

Každý z těchto kroků je popsán v následujících částech.

Určení, zda byla transakce implementována nekonzistentně

Vášim prvním úkolem je určit, zda byla transakce v důsledku nezávislé akce implementována nekonzistentně.

Předčasné ukončení globální transakce

Pokud jste spuštěním příkazu **onmode -z** na koordinátorovi ukončili globální transakci, může být transakce implementována nekonzistentně. (Vysvětlení okolností vzniku této situace naleznete v části “Nezávislé akce, které vyústí v chybový stav” na stránce 24-9.) Zjištění nekonzistentní transakce provedete nejdříve přezkoumáním protokolu zpráv databázového serveru pro koordinátora. Hledejte následující chybovou zprávu:

-716 Possible inconsistent transaction. Unknown servers are *seznam_název_serverů*.

Tato zpráva vypisuje všechny databázové servery, které byly účastníky. Prozkoumejte logický protokol každého účastníka. Pokud alespoň jeden účastník provedl potvrzení transakce a jiný provedl odvolání transakce, byla transakce implementována nekonzistentně.

Heuristické ukončení transakce

Pokud jste spustili příkaz **onmode -Z address**, abyste ukončili část práce prováděnou účastníkem *a* koordinátor se rozhodl potvrdit transakci, bude transakce implementována nekonzistentně. (Popis tohoto scénáře naleznete v části “Scénář heuristického ukončení transakce” na stránce 24-12.) Prozkoumejte logický protokol každého účastníka. Pokud alespoň jeden účastník provedl potvrzení transakce a jiný provedl odvolání transakce, byla transakce implementována nekonzistentně.

Heuristické odvolání transakce

Konkrétní účastnické databázové servery, ovlivněné heuristickým rozhodnutím odvolat transakci můžete určit následujícím způsobem:

- Prozkoumejte v aplikaci návratový kód příkazu COMMIT WORK.
Následující zpráva označuje, že jeden z účastníků provedl heuristické odvolání transakce:
-698 Inconsistent transaction. *Number and names of servers rolled back.*
- Přezkoumejte zprávy v souboru protokolu zpráv databázového serveru.
Pokud kvůli heuristickému rozhodnutí na zúčastněném databázovém serveru mohlo dojít k nekonzistenci databáze, bude v souboru protokolu zpráv koordinačního databázového serveru uvedena tato zpráva:
Mixed transaction result. (pid=*nn* user=*id_uživatele*)
Tato zpráva se zapíše vždy, když je vrácena chyba -698. K této zprávě je přidružen seznam účastnických databázových serverů, na kterých byla tato transakce odvolána. Jedná se o úplný seznam. Seznam, který se zobrazuje s chybovou zprávou -698, může být zkrácen, pokud transakci odvolal větší počet účastníků.
- Prozkoumejte logický protokol každého účastníka.
Pokud alespoň jedním účastníkem svou část práce odvolá a jeden z účastníků svou část práce potvrdí, je transakce implementována nesprávně.

Určení, zda distribuovaná databáze obsahuje nekonzistentní data

Pokud zjistíte, že transakce byla implementována nekonzistentně, musíte určit, co tato situace znamená pro databázový systém. Konkrétně musíte určit, zda nebyla narušena integrita dat.

Transakce, která byla nekonzistentně implementována, způsobuje problémy, kdekoli je část práce odvolaná jedním účastníkem závislá na části práce aktualizované jiným účastníkem. Tyto závislosti není možné definovat pomocí jazyka SQL, protože distribuované transakce nepodporují referenční omezení dat na více databázových serverech. Části práce jsou nezávislé (neexistuje žádná závislost) pouze tehdy, pokud by data mohla být aktualizována ve dvou nezávislých transakcích. V opačném případě jsou části práce považovány za závislé.

Než budete pokračovat, zamyslete se nad transakcí, která způsobila chybu. Jsou části dat, kterých se týkala aktualizace transakce a odvolání transakce, navzájem závislé? Do jediné transakce mohlo být několik aktualizací zahrnuto z jiných důvodů, než je udržení integrity dat. Tři možné důvody jsou například tyto:

- snížení režie transakcí
- zjednodušení programování
- preference programátora

Ověřte také, zda skutečně měnily data všechny účastnické databázové servery, u kterých se předpokládá, že transakci potvrdily. Jako účastník, který transakci potvrdil, mohl být uveden databázový server v režimu pouze pro čtení.

Pokud nekonzistentní transakce nevede k narušení integrity dat, můžete v tomto bodu ukončit tuto proceduru.

Získání informací z logického protokolu

Chcete-li určit, zda byla nekonzistentně implementovanou globální transakcí ovlivněna integrita dat, musíte rekonstruovat tuto globální transakce a určit, které části transakce byly potvrzeny a které byly odvolány. Nezbytné informace získáte pomocí obslužného programu **onlog**. Postup je následující:

1. Zrekonstruujte transakci na účastníkovi, který obsahuje záznam HEURTX.
 - a. Výchozím bodem pro získávání informací je logický protokol účastnického databázového serveru. Každý záznam v protokolu má identifikační číslo místní transakce (**xid**). Získejte hodnotu **xid** záznamu HEURTX.
 - b. Pomocí místního identifikátoru **xid** vyhledejte všechny přidružené záznamy protokolu, které byly odvolány jako část této části práce.
2. Určete, který databázový server byl koordinátorem této globální transakce.
 - a. Vyhledejte na účastníkovi záznam PREPARE, který obsahuje stejný místní identifikátor **xid**. Záznam PREPARE označuje začátek protokolu dvoufázového potvrzování pro tohoto účastníka.
 - b. Pomocí volby **onlog -l** získejte dlouhý výstup záznamu PREPARE. Tento záznam obsahuje identifikátor globální transakce (GTRID) a název koordinujícího databázového serveru. Další informace o identifikátoru globální transakce GTRID naleznete v části “Získání identifikátoru globální transakce” na stránce 25-3.
3. Z protokolu koordinátora získejte seznam ostatních účastníků.
 - a. Prozkoumejte záznamy protokolu na koordinačním databázovém serveru. Vyhledejte záznam BEGPREP.
 - b. Prozkoumejte dlouhý výstup pro záznam BEGPREP. Pokud prvních 32 bajtů identifikátoru GTRID v tomto záznamu odpovídá identifikátoru GTRID u účastníka, je záznam BEGPREP součástí téže globální transakce. Poznamenejte si účastníky zobrazené v části ASCII dlouhého výstupu záznamu BEGPREP.
4. U každého účastníka zrekonstruujte transakci.
 - a. Na každém účastnickém databázovém serveru si přečtete logický protokol a vyhledejte záznam PREPARE obsahující identifikátor GTRID přidružený k této transakci a získejte místní identifikátor **xid** pro část práce prováděnou tímto účastníkem.
 - b. Na každém účastnickém databázovém serveru použijte místní identifikátor **xid** k vyhledání všech záznamů logického protokolu přidružených k této transakci (potvrzených nebo odvolaných).

Pokud se budete řídit tímto postupem, zjistíte všechny účastníky dané transakce, které části práce byly přiřazeny každému účastníkovi a u každé části práce budete vědět, zda byla odvolána nebo potvrzena. Z těchto informací můžete určit, zda nezávislá akce narušila integritu dat.

Získání identifikátoru globální transakce

Když se globální transakce spouští, obdrží jedinečné identifikační číslo nazývané identifikátor globální transakce (GTRID). Identifikátor GTRID zahrnuje název koordinátora. Identifikátor GTRID je zapsán do záznamu BEGPREP v logickém protokolu koordinátora a záznamu PREPARE v logickém protokolu každého účastníka.

Identifikátor GTRID lze zobrazit pomocí volby **onlog -l**. Identifikátor GTRID je posunut o 20 bajtů vůči počátku datové části záznamu a je 144 bajtů dlouhý. Obrázek 25-1 ukazuje výstup příkazu **onlog -l** pro záznam BEGPREP. Koordinátor je server **chrisw**.

```

4a064  188 BEGPREP 4      0 4a038      0 1
000000bc 00000043 00000004 0004a038 .....C .....8
00087ef0 00000002 63687269 73770000 ..~..... chrisw..
00000000 00000000 00000000 00087eeb ..... ~:
000006b16 00000000 00000000 00000000 ..k.....
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000001 6a756469 74685f73 ..... judith_s
6f630000 736f6374 63700000          oc..soct cp..

```

Obrázek 25-1. Výstup volby **onlog -l** pro záznam BEGPREP

Prvních 32 bajtů identifikátoru GTRID je shodných v záznamu BEGPREP koordinátora a v záznamech PREPARE účastníků, kteří jsou součástí téže globální transakce. Porovnejte například identifikátor GTRID záznamu PREPARE, který uvádí Obrázek 25-2, s identifikátorem záznamu BEGPREP, který ukazuje Obrázek 25-1.

```

c7064  184 PREPARE 4      0 c7038      chrisw
000000b8 00000044 00000004 000c7038 .....D .....p8
00005cd6 00000002 63687269 73770000 ..... chrisw..
00000000 00000000 00000069 00087eeb ..... .i..~.
000006b16 00000000 00000010 00ba5a10 ..k.....Z.
00000002 00ba3a0c 00000006 00000000 .....:.....
00ba5a10 00ba5a1c 00000000 00000000 ..Z...Z. ....
00ba3a0e 00254554 00ba2090 00000001 ...:%ET .. ....
00000000 00ab8148 0005fd70 00ab8148 .....H ...p...H
0005fe34 0000003c 00000000 00000000 ...4...< .....
00000000 00ab80cc 00000000 00ab80c4 .....
00ba002f 63687269 73770000 00120018 .../chrisw.....
00120018 00ba0000          .....

```

Obrázek 25-2. Výstup volby **onlog -l** pro záznam PREPARE

Rozhodnutí, zda je nezbytné k nápravě situace provést akci

Pokud nekonzistentní transakce způsobí nekonzistenci databáze, máte tyto tři možnosti:

- Ponechat síťovou databázi v nekonzistentním stavu.
- Odstranit změny provedené transakcí všude, kde byla potvrzena, a tudíž odvolat celou transakci.
- Znovu použít změny provedené transakcí všude, kde byla odvolána, a tudíž transakci potvrdit.

Databázi můžete ponechat v nekonzistentním stavu, pokud transakce neovlivnila data databáze zásadním způsobem. S takovou situací se můžete setkat, pokud aplikace provádějící tuto transakci může dále pracovat i s databází v tomto stavu, a rozhodnete se, že cena (čas a vynaložené úsilí) za návrat databáze do konzistentního stavu, ať již odstraněním vlivů nebo přehráním transakce, je příliš vysoká.

Toto rozhodnutí nemusíte provést okamžitě. Pomocí metod popsaných v následujících částech můžete určit, která data transakce aktualizovala a které záznamy byly ovlivněny.

Až budete rozhodovat, zvažte, že žádný automatický proces nebo obslužný program neumí provést odvolání potvrzené transakce ani neumí potvrdit část transakce, která byla odvolána. V následujících odstavcích získáte informace o tom, jak prohlížet protokol zpráv a logický protokol databázového serveru a hledat ovlivněné záznamy. Bez podrobné znalosti aplikace

však nelze pouze pomocí zpráv zjistit, co se skutečně stalo. Rozhodnutí, zda transakci odvolat nebo potvrdit, musíte založit na znalostech aplikace a systému. Musíte rovněž naprogramovat kompenzační transakci, která provede odvolání nebo potvrzení transakce.

Příklad ruční obnovy

Tento příklad ilustruje, jaká práce je součástí ruční obnovy. Následující příkazy jazyka SQL byly spuštěny uživatelem **nhowe**. Byla vrácena chyba -698.

```
dbaccess
CREATE DATABASE tmp WITH LOG;
CREATE TABLE t (a int);
CLOSE DATABASE;
CREATE DATABASE tmp@apex WITH LOG;
CREATE TABLE t (a int);
CLOSE DATABASE;
DATABASE tmp;
BEGIN WORK;
INSERT INTO t VALUES (2);
INSERT INTO tmp@apex:t VALUES (2);
COMMIT WORK;
### return code -698
```

Následující výňatek je vybrán z logického protokolu aktuálního databázového serveru:

addr	len	type	xid	id	link
..... 17018	16	CKPOINT	0	0	13018 0
18018 3482	20	BEGIN nhowe	2	1	0 08/27/91 10:56:57
1802c 4	32	HINSERT	2	0	18018 1000018 102
1804c	40	CKPOINT	0	0	17018 1
		begin	xid	id	addr user
		1	2	1	1802c nhowe
19018	72	BEGPREP	2	0	1802c 6d69 1
19060	16	COMMIT	2	0	19018 08/27/91 11:01:38
1a018	16	ENDTRANS	2	0	19060 580543

Následující výňatek je vybrán z logického protokolu databázového serveru **apex**:

addr	len	type	xid	id	link
..... 16018 10:57:07	20	BEGIN pault	2		1 0 08/27/91
1602c 4	32	HINSERT	2	0	16018 1000018 102
1604c	68	PREPARE	2	0	1602c eh
17018	16	HEURTX	2	0	1604c 1
17028	12	CLR	2	0	1602c
17034	16	ROLLBACK	2	0	17018 08/27/91 11:01:22
17044	40	CKPOINT	0	0	15018 1
		begin	xid	id	addr user

```

          1          2          1 17034  -----
18018    16          ENDTRANS 2      0      17034      8806c3
.....

```

Nejdříve se pokusíte nalézt shodné transakce v protokolu aktuálního databázového serveru s transakcemi v protokolu databázového serveru **apex**. Záznamy protokolu BEGPREP a PREPARE obsahují identifikátor GTRID. Identifikátor GTRID můžete získat pomocí příkazu **onlog -l** a z datové části záznamů protokolu BEGPREP a PREPARE. Identifikátor GTRID je posunut o 22 bajtů vůči počátku datové části a jeho délka je 68 bajtů. Mnohem jednodušší, ačkoli méně přesný přístup je zjištění času v záznamech COMMIT nebo ROLLBACK. Časy by měly být podobné, ačkoli zde může být malé zpoždění kvůli době nutné k přenosu zprávy o potvrzení (nebo odvolání) transakce od koordinátora k účastníkovi. (Tento druhý přístup postrádá přesnost, neboť souběžné transakce mohou být potvrzeny ve stejný čas, ačkoli souběžné transakce od jednoho koordinátora by pravděpodobně nebyly potvrzeny ve stejný čas.)

Postup nápravy vzorové situace:

1. Vyhledejte všechny záznamy, které byly aktualizovány.
2. Identifikujte jejich typ (vlození, odstranění, aktualizace) pomocí obslužného programu **onlog** a tabulky typů záznamů.
3. Použijte výstup příkazu **onlog -l** pro každý záznam k získání místního identifikátoru **xid**, čísla prostoru tblspace a čísla rowid.
4. Namapujte číslo prostoru tblspace na název tabulky porovnáním čísla prostoru tblspace s hodnotou ve sloupci **partnum** tabulky systémového katalogu **systables**.
5. S využitím znalostí aplikace určete akci, která je potřeba pro nápravu situace.

V tomto příkladu jsou si časové značky záznamů COMMIT a ROLLBACK v různých protokolech velmi blízké. V daném čase nejsou aktivní žádné jiné transakce, které by mohly být souběžně potvrzeny nebo odvolány. V tomto případě bylo v aktuálním databázovém serveru potvrzeno vložení (záznam HINSERT) s přiřazenou hodnotou rowid 102 hexadecimálně (258 desítkově). Kompenzační transakce je tedy následující:

```
DELETE FROM t WHERE rowid = 258
```

Část 6. Automatické monitorování a nápravné akce

Kapitola 26. Automatické monitorování a nápravné akce

Tato kapitola poskytuje informace o rozhraní API administrace, plánovači a provádění analýz historie příkazů jazyka SQL pomocí funkce procházení dotazů Query Drill-Down. Další informace naleznete v částech “Přehledné informace o automatickém monitorování a o nápravných akcích”, “Vzdálená správa pomocí příkazů SQL” na stránce 26-10, “Plánovač” na stránce 26-5 a “Procházení dotazů” na stránce 26-11.

Přehledné informace o automatickém monitorování a o nápravných akcích

Pomocí rozhraní API administrace, plánovače, informací uložených v databázi **sysadmin** a pomocí funkce procházení dotazů Query Drill-Down můžete automatizovat správu, monitorování a úlohy administrace. Tyto komponenty serveru Dynamic Server umožňují zjednodušit sběr informací a správu serveru ve složitých systémech.

Rozhraní API administrace

Rozhraní API administrace umožňuje provádět vzdálenou administraci pomocí určitých příkazů jazyka SQL. Další informace naleznete v části “Vzdálená správa pomocí příkazů SQL” na stránce 26-10.

Plánovač

Plánovač umožňuje databázovému serveru provádět databázové funkce a procedury v předdefinovanou dobu nebo v dobu interně určenou serverem. Tyto funkce shromažďují informace a monitorují a nastavují server pomocí systému administrace založeného na jazyku SQL a na sadě úloh. Další informace naleznete v části “Plánovač” na stránce 26-5.

Databáze sysadmin

Databáze **sysadmin** obsahuje tabulky, ve kterých jsou uloženy vlastnosti úloh. K definování informací, které má plánovač shromažďovat, a úkolů, které má plánovač spouštět, se používají vlastnosti úloh (nikoli konfigurační parametry). Další informace o této databázi naleznete v části “Databáze sysadmin” na stránce 26-2.

Tabulka **command_history** v databázi **sysadmin** obsahuje historické informace o příkazech SQL rozhraní API administrace spuštěných v tomto datovém serveru. Další informace naleznete v části “Tabulka command_history” na stránce 26-3.

Funkce procházení dotazů Query Drill-Down umožňující analýzu historie příkazů SQL

Funkce procházení dotazů Query Drill-Down poskytuje statistické informace o nedávno provedených příkazech jazyka SQL, umožňuje sledovat výkon jednotlivých příkazů jazyka SQL a analyzovat historii příkazů. Další informace naleznete v části “Procházení dotazů” na stránce 26-11.

Administrativní rozhraní API a plánovač můžete používat na primárním serveru páru replikace HDR.

Protože se všechny operace rozhraní API administrace provádějí pomocí jazyka SQL, mohou tyto funkce k administraci databázového serveru používat klientské nástroje.

Při administraci několika instancí databázových serverů z jednoho umístění můžete použít také webový administrativní nástroj OpenAdmin pro server IDS založený na technologii PHP. Mezi úlohy, které můžete s nástrojem OpenAdmin provádět, patří:

- Definování a správa automatizovaných úloh prostřednictvím rozhraní SQL Administration API

- Vytváření a zobrazování histogramů výkonu pro analýzu a ladění
- Monitorování řešení vysoké dostupnosti, které zahrnují replikaci HDR, sekundární servery se sdílenými disky a vzdálené samostatné sekundární servery.

Do nástroje OpenAdmin můžete zapojit vlastní rozšíření a vytvořit tak funkčnost, kterou potřebujete.

Nástroj OpenAdmin je program s otevřeným zdrojem, který je možné stáhnout z tohoto webového serveru: <http://www-306.ibm.com/software/data/informix/downloads.html>.

Zvýšené požadavky na paměťový prostor

Pokud budete používat rozhraní API administrace a plánovač a pokud budete používat funkci procházení dotazů Query Drill-Down, budete potřebovat místo na disku pro tyto funkce a k ukládání tabulek s historií.

Pokud zapnete funkci procházení dotazů Query Drill-Down, budou vyžadovány nejméně 2 MB. Tuto paměť můžete nastavit pomocí rozhraní API administrace, zatímco se systém nachází v režimu online, nebo změnou konfiguračního parametru SQLTRACE v souboru ONCONFIG.

Úlohy, které pracují jako senzory (shromažďují informace a ukládají je v tabulkách) také mohou spotřebovávat místo na disku databázového serveru. K odhadu využití místa na disku jedním senzorem můžete použít následující vzorec:

*počet shromažďovaných řádků * velikost shromažďovaného řádku * frekvence sběrů dat v průběhu dne * doba zachování*

Tento odhad zopakujte pro všechny senzory. Tak můžete určit přibližnou velikost požadovaného místa.

Senzory, které shromažďují a ukládají informace o historii, můžete vypnout. Vypnutí těchto senzorů však může ovlivnit další funkce databázového serveru.

Databáze sysadmin

Databáze **sysadmin**, která obsahuje tabulky s vlastnostmi úloh, je protokolovaná databáze. K definování informací, které má plánovač shromažďovat, a úkolů, které má plánovač spouštět, se používají vlastnosti úloh (nikoli konfigurační parametry).

Databáze **sysadmin** obsahuje také:

- Vestavěnou funkci **task()**.
- Vestavěnou funkci **admin()**.
- Tabulku **command_history**, která obsahuje informace o příkazech spuštěných pomocí rozhraní API administrace.

Důležité: Databázi **sysadmin** nikdy nevypouštějte ani neměňte, protože ji používá několik komponent databázového serveru.

Databáze **sysadmin** je při výchozím nastavení vytvořena v kořenovém prostoru dbspace. Pokud chcete databázi přesunout, naleznete další informace v části “Přesun databáze sysadmin do nového databázového prostoru” na stránce 26-4.

Tabulky databáze sysadmin

Databáze **sysadmin** obsahuje následující tabulky:

Tabulka 26-1. Tabulky databáze sysadmin

Tabulka	Popis
PH_ALERT	Obsahuje seznam chyb, upozornění a informačních zpráv, které musejí být monitorovány.
PH_GROUP	Obsahuje seznam názvů skupin. Každá úloha je členem skupiny.
PH_RUN	Obsahuje informace o tom, jak a kdy se která úloha provedla.
PH_TASK	Obsahuje seznam úloh a informace o tom, jak a kdy databázový server jednotlivé úlohy bude provádět.
PH_THRESHOLD	Obsahuje seznam definovaných prahových hodnot. Pokud je dosažena prahová hodnota, úloha se může rozhodnout provést jinou akci, například vložení výstrahy do tabulky PH_ALERT.

Každý řádek v tabulce **ph_task** je samostatná *úloha* (definovaná jako jediná událost monitorování) a každý sloupec představuje vlastnost úlohy. Vlastnosti úlohy například označují, kdy má být spuštěn příkaz SQL, uložená procedura nebo rutina UDR, a jak má být úloha zpracována. Příkladem úlohy může být automatické spuštění konkrétní úlohy každé pondělí o půlnoci.

Podrobnosti o těchto tabulkách naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Funkce jazyka SQL rozhraní API administrace

V databázi **sysadmin** jsou k dispozici dvě vestavěné administrativní funkce: Funkce **task()** a funkce **admin()**. Funkce **task()** a **admin()** smí spouštět pouze administrátor DBSA. Ve výchozím nastavení se však smí připojit k databázi **sysadmin** pouze uživatel **informix**.

Funkce **task()** a **admin()** provádějí totéž, liší se však návratovým kódem. Funkce **task()** vrací řetězec, který popisuje výsledky příkazu. Funkce **admin()** vrací celé číslo.

K definování nebo úpravám těchto funkcí se používá jazyk SQL. Můžete například definovat úlohu, která vytváří prostor dbspace, a to následujícím příkazem:

```
EXECUTE FUNCTION admin('create dbspace', 'dbspace2', '/work/dbspace2', "20 MB");
```

Informace o definování funkcí **admin()** a **task()** a příklady naleznete v příručce *IBM Informix Guide to SQL: Syntax*.

Tabulka command_history

Tabulka **command_history** obsahuje seznam všech příkazů, které byly spuštěny pomocí rozhraní API administrace. Tabulka zobrazuje také výsledky příkazů. Tato tabulka, která se nachází v databázi **sysadmin**, je neprotokolovaná tabulka (typu RAW).

Tabulka **command_history** obsahuje informace o tom, zda byla úloha administrace spuštěna pomocí funkce **admin()** nebo **task()**, a také informace o uživateli, který příkaz spustil, o čase, kdy byl příkaz spuštěn, samotný příkaz a zprávu vrácenou poté, co databázový server provádění příkazu dokončil.

Následující tabulka znázorňuje ukázkové příkazy a odpovídající výsledky v ukázkové tabulce **command_history**. Popis všech informací v tabulce **command_history** naleznete v příručce *IBM Informix Dynamic Server Administrator's Reference*.

Tabulka 26-2. Příklady některých informací v tabulce **command_history**

Spuštěný příkaz	Ukázkové vrácené zprávy
SET SQL TRACING ON	SQL tracing on with 1000 buffers of 2024 bytes.
vytvoření prostoru dbspace	Space 'space12' added.
kontrolní bod	Kontrolní bod dokončen.
add log	K prostoru dbspace logdbs byly přidány 3 logické protokoly.

Chcete-li zobrazit historii příkazů, spusťte tento příkaz SQL:

```
SELECT * from command_history
```

Úlohy v tabulce **command_history** jsou po určité pevné době automaticky odstraněny. Tuto dobu můžete změnit úpravou údajů v řádku **COMMAND HISTORY RETENTION** v tabulce **ph_threshold**. Parametr **COMMAND HISTORY RETENTION** nastavuje časové období, po které by řádky měly zůstat v tabulce **command_history**.

Pomocí příkazů SQL, jako je **delete** nebo **truncate table**, můžete data z této tabulky odstranit ručně. Můžete také vytvořit úlohu v tabulce **ph_task**, která vymaže data z tabulky **command_history**. Následující příkaz znázorňuje úlohu, která monitoruje množství dat v tabulce **command_history** a vymaže data, která jsou zastaralá.

```
INSERT INTO ph_task
( tk_name, tk_type, tk_group, tk_description, tk_execute,
tk_start_time, tk_stop_time, tk_frequency )
VALUES
('mon_command_history',
'TASK',
'TABLES',
'Monitorovat, kolik dat uchovává tabulka historie příkazů',
'delete from command_history where cmd_exec_time < (
select current - value::INTERVAL DAY TO SECOND
from ph_threshold
where name = 'COMMAND HISTORY RETENTION' ) ',
DATETIME(02:00:00) HOUR TO SECOND,
NULL,
INTERVAL ( 1 ) DAY TO DAY);
```

Přesun databáze sysadmin do nového databázového prostoru

Databáze **sysadmin** je při výchozím nastavení vytvořena v kořenovém prostoru dbspace. Pokud v kořenovém prostoru dbspace není dostatek místa pro uložení informací o vlastnostech úloh a historii příkazů, můžete databázi **sysadmin** přesunout do jiného prostoru dbspace pomocí příkazu API administrace v jazyce SQL **"reset sysadmin"**. Tento příkaz odstraní databázi **sysadmin** z kořenového prostoru dbspace a vytvoří ji znovu v určeném prostoru dbspace.

Postup při přesunu databáze **sysadmin**:

1. Zkontrolujte, zda se v protokolu zpráv zobrazila po spuštění serveru následující zpráva:

SCHAPI: Byly spuštěny 2 jednotkové procesy dbWorker.

2. V případě potřeby vytvořte pro databázi **sysadmin** nový prostor typu dbSpace, například *new_dbSpace*.
3. Spuštěte následující příkaz jako uživatel **informix**:
dbaccess sysadmin -
execute function task("reset sysadmin","new_dbSpace");

kde *new_dbSpace* je název prostoru dbSpace, ve kterém bude uložena databáze **sysadmin**.

Příkaz vrací následující zprávu:

SCHAPI: Databáze 'sysadmin' bude přesunuta do 'new_dbSpace'.
Další informace naleznete v protokolu zpráv online.

Vnitřní jednotkový proces **bld_sysadmin** čeká po dobu až pěti minut na získání výlučného přístupu k databázi **sysadmin**. Průběh jednotkového procesu **bld_sysadmin** je zaznamenáván do online protokolu zpráv.

4. Ukončete databázovou relaci příkazem **close database**.

Skončí-li tato operace úspěšně, je databáze **sysadmin** vypuštěna a znovu vytvořena v novém prostoru dbSpace. Jednotkové procesy plánovače a dbWorker jsou spuštěny automaticky.

Plánovač

Plánovač spravuje a provádí plánované úlohy údržby, monitorování a administrace. Tento nástroj umožňuje monitorovat aktivity, jako je například správa volných míst nebo automatické zálohování všech nových dat protokolu v načasovaných intervalech od posledního zálohování protokolu, a vytvářet nápravné akce, které běží automaticky.

Plánovač spravuje:

- Úlohy, které poskytují prostředky pro spuštění specifických úloh v určitém čase nebo intervalu.
- Sensory, které shromažďují a ukládají informace.
- Úlohy při spuštění, které jsou spuštěny pouze jednou, když je spuštěn databázový server.
- Sensory při spuštění, které jsou spuštěny pouze jednou, když je spuštěna databáze.

Plánovač řídí sada vlastností úloh, které určují, jaké informace mají být shromážděny nebo jaké úlohy mají být provedeny. Vlastnosti úloh jsou uloženy v tabulce **ph_task** v databázi **sysadmin**. Každý řádek této tabulky představuje samostatnou úlohu a každý sloupec vlastnost úloh. Vlastnosti úloh informují systém o tom, kdy má být spuštěn příkaz SQL, uložená procedura nebo funkce a jak má být úloha zpracována.

Můžete například definovat úlohu, která denně zjišťuje volné místo v protokolu každou hodinu od 9:00:00 do 19:00:00.

Informace, které plánovač shromažďuje a úlohy, které provádí, jsou definovány pouze vlastnostmi úloh, ne konfiguračními parametry. Plánovač provádí úlohy v předdefinovaném čase nebo podle vnitřního určení podle požadavků databázového serveru.

Plánovač dodávaný s dynamickým serverem obsahuje úlohy uvedené v následující tabulce.

Tabulka 26-3. Vestavěné úlohy

Úkol	Popis
mon_command_history	Vyprázdní tabulku historie.
mon_config	Uloží všechny změny souboru ONCONFIG.

Tabulka 26-3. Vestavěné úlohy (pokračování)

Úkol	Popis
mon_config_startup	Uloží soubor ONCONFIG při každém spuštění serveru.
mon_profile	Uloží informace o profilu serveru.
mon_vps	Shromáždí informace o virtuálním procesoru.
mon_checkpoint	Uloží informace o kontrolních bodech.
mon_table_profile	Uloží informace profilech tabulek včetně celkového počtu aktualizací, vložení a odstranění, které byly v této tabulce provedeny.
mon_table_names	Uloží názvy tabulek spolu s časem jejich vytvoření.
mon_users	Uloží informace o profilu pro každého uživatele.
check_backup	Zkontroluje, zda je spuštěno zálohování.

Tyto úlohy můžete měnit a můžete také vytvořit další úlohy. Další informace naleznete v částech “Vytváření úloh” a “Úprava úloh” na stránce 26-10.

Vytváření úloh

“Plánovač” na stránce 26-5 spravuje a provádí plánované úlohy údržby, monitorování a administrace. Nové úlohy můžete vytvořit vložением řádků do tabulky **ph_task**. Použijte tento způsob, chcete-li vytvořit nové aktivity monitorování nebo nápravné akce, které se spouštějí automaticky.

V tabulce **ph_task** jsou uvedeny úlohy a obsahuje také informace o tom, kdy a jak bude databázový server jednotlivé úlohy provádět.

Když databázový server provádí úlohu, vyvolá server objekt SQL obsažený ve sloupci **pk_execute** tabulky **ph_task**. Sloupec **pk_execute** může obsahovat uživatelskou funkci, samotný příkaz SQL nebo objekt připravený několika příkazy, který byl vytvořen pomocí příkazu PREPARE jazyka SQL, aby umožnil sestavení jednoho nebo více příkazů za běhu.

Procedura: Vytvoření úlohy

- Při plánování úlohy by neměly chybět následující údaje:
 - Popis úlohy, kterou chcete monitorovat.
 - Tabulka, do které chcete uložit data.
 - Příkaz SQL, uložená procedura nebo funkce pro sběr dat.
 - Informace o tom, kdy a jak chcete úlohu spouštět.
- Vložte nový řádek do tabulky **ph_task** pomocí jazyka SQL podle následujících příkladů. Zadejte hodnoty, které budou zobrazeny ve sloupcích tabulky.

Následující příklad uvádí kód úlohy, která je spouštěna jednou denně ve 2:00 a zajišťuje, aby tabulka **command_history** obsahovala pouze nejnovější data. V tomto příkladu je definice nejnovějších dat uložena ve sloupci **Interval historie příkazů** v tabulce **ph_threshold**.

```

INSERT INTO ph_task
(
tk_name,
tk_group,
tk_description,
tk_type,
tk_execute,
tk_start_time,
tk_frequency
)
VALUES
(
"mon_command_history",
"TABLES",
"Monitorovat, kolik dat uchováva tabulka historie příkazů",
"MONITOR",
"delete from command_history where cmd_exec_time < (
select current - value::INTERVAL DAY to SECOND
from ph_threshold
where name = 'COMMAND HISTORY INTERVAL' ) ",
"2:00:00",
"1 0:00:00"
);

```

Obrázek 26-1. Ukázkový kód pro úlohu, která je spouštěna denně

Následující příklad uvádí kód pro vytvoření tabulky s názvem **mon_prof**, do které se ukládají data a pohled s názvem **mon_profile**. V tomto příkladu jsou data shromažďována každých pět minut a jsou použity výchozí metody plánování (například 8:00 až 17:00, pět dní v týdnu). Předdefinovanou hodnotou je \$DATA_SEQ_ID; tuto hodnotu nahradí ID datové posloupnosti.

```

INSERT INTO ph_task
(
tk_description,
tk_result_table,
tk_create,
tk_execute,
tk_frequency
)
VALUES
(
"Shromažďování informací obecných profilů",
"mon_prof",
"create table mon_prof (ID integer, number integer, value int8 );
create view mon_profile as select ID, A.name,
B.value from sysmaster:sysshmhdr A,
mon_prof B where B.number = A.number ",
"insert into mon_prof select $DATA_SEQ_ID, number,
value from ysmaster:sysshmhdr",
"0 0:05:00"
);

```

Obrázek 26-2. Ukázkový kód pro vytvoření tabulky, která ukládá data

Následující příklad uvádí kód pro vytvoření tabulky s názvem **mon_chunkio**, která shromažďuje a ukládá data pomocí uložené procedury. Tato úloha nařídí databázovému serveru, aby každých pět minut shromažďoval data provedením uložené procedury s názvem **chunkio**. Jsou použity výchozí metody plánování (například 8:00 až 17:00, pět dní v týdnu),

ale jakmile budou data sedm dní stará, jejich platnost vyprší a budou odstraněna.

```
INSERT INTO ph_task
(
tk_description,
tk_result_table,
tk_create,
tk_execute,
tk_frequency,
tk_delete
)
VALUES
(
"Počet bloků vstupu - výstupu a využití prostoru",
"mon_chunkio",
"create table mon_chunkio (ID integer, chunknum smallint,
  free integer, size integer, reads integer, pagereads integer,
  writes integer, pageswritten integer)",
"chunkio",
"0 0:05:00",
"7 0:00:00"
);
```

Obrázek 26-3. Ukázkový kód pro vytvoření tabulky, která při shromažďování a ukládání dat používá uloženou proceduru

Následující příklad uvádí kód snímače, který zaznamenává spouštěcí prostředí databázového serveru.

```

INSERT INTO ph_task
(
tk_name,
tk_type,
tk_group,
tk_description,
tk_result_table,
tk_create,
tk_execute,
tk_stop_time,
tk_start_time,
tk_frequency,
tk_delete
)
VALUES
(
"mon_sysenv",
"STARTUP SENSOR",
"SERVER",
"Zaznamenává spouštěcí prostředí databázového serveru.",
"mon_sysenv",
"create table mon_sysenv (ID integer, name varchar(250), value lvarchar(1024))",
"insert into mon_sysenv select $DATA_SEQ_ID, env_name, env_value FROM sysmaster:sysenv",
NULL,
NULL,
"0 0:01:00",
"60 0:00:00"
);

```

```

tk_id          7
tk_name        mon_sysenv
tk_description  Zaznamenává spouštěcí prostředí databázového serveru.
tk_type        STARTUP SENSOR
tk_executing_sid 0
tk_sequence    0
tk_result_table mon_sysenv
tk_create      create table mon_sysenv (ID integer, name varchar(250),
               value lvarchar(1024))
tk_execute     insert into mon_sysenv select $DATA_SEQ_ID, env_name,
               env_value FROM sysmaster:sysenv
tk_delete      60 00:00:00
tk_start_time
tk_stop_time
tk_frequency    0 00:01:00
tk_next_execution 2006-07-24 17:09:50
tk_total_execution+ 0
tk_total_time  0.00
tk_monday      t
tk_tuesday     t
tk_wednesday   t
tk_thursday    t
tk_friday      t
tk_saturday    t
tk_sunday      t
tk_attributes  0
tk_group       SERVER
tk_enable      t
tk_priority    0

```

Obrázek 26-4. Snímač, který zaznamenává spouštěcí prostředí

Následující příklad uvádí kód snímače, který shromažďuje informace o množství použité paměti a ukládá informace do tabulky **mon_memory_system**. Pokud tabulka neexistuje,

úloha ji vytvoří. Tato úloha, která se spouští každých 30 minut, odstraňuje z tabulky **mon_memory_system** veškerá data, která existují déle než 30 dní.

```
INSERT INTO ph_task
(tk_name, tk_type, tk_group, tk_description, tk_result_table, tk_create,
tk_execute, tk_stop_time, tk_start_time, tk_frequency, tk_delete )
VALUES
("mon_memory_system",
"SENSOR",
"MEMORY",
"Server memory consumption",
"mon_memory_system",
"create table mon_memory_system (ID integer, class smallint, size int8,
used int8, free int8 )",
"insert into mon_memory_system select $DATA_SEQ_ID, seg_class, seg_size,
seg_blkused, seg_blkfree FROM sysmaster:sysseg1st",
NULL,
NULL,
INTERVAL ( 30 ) MINUTE TO MINUTE,
INTERVAL ( 30 ) DAY TO DAY);
```

Obrázek 26-5.

Úprava úloh

K úpravě úloh lze použít řádky v tabulkách, které začínají řetězcem **ph_**.

Úprava úloh:

1. V databázi **sysadmin** přejděte k tabulce **ph_task** nebo k jiné tabulce, kterou chcete upravit.
2. Ručně změňte informace o úloze.

Vzdálená správa pomocí příkazů SQL

Administrační rozhraní API jazyka SQL umožňuje provádění vzdálené administrace pomocí různých specifických příkazů jazyka SQL pro úlohy, jako například správu prostorů, správu konfigurace, spouštění rutinních úloh a ověření systému.

Pomocí příkazu EXECUTE FUNCTION můžete vyvolat vestavěné funkce **admin()** nebo **task()** a provádět administrativní úlohy, které jsou shodné se spuštěním různých administračních nástrojů serveru Dynamic Server.

V příkazech EXECUTE FUNCTION určují položky v seznamu argumentů nástroj a jeho argumenty příkazové řádky. Následující příkaz SQL je například shodný s příkazem **oncheck -ce**, který nařizuje databázovému serveru, aby provedl kontrolu oblastí:

```
EXECUTE FUNCTION admin('check extents');
```

Předpokládejme, že chcete zvětšit velikost virtuální paměti databázového serveru, kterou lze využít v aplikaci. Aplikace může provést tento příkaz SQL:

```
EXECUTE FUNCTION task('add memory', '10 MB')
```

Předpokládejme, že chcete vypustit všechny logické protokoly v kořenovém bloku kromě aktuálního logického protokolu. Příkaz administračního rozhraní API můžete použít následujícím způsobem v příkazu SELECT:

```
select task("drop log", number) from sysmaster:syslogfil
where chunk = 1 and sysmaster:bitval(flags,"0x02")==0;
```

Musíte provést funkce **task()** a **admin()** na databázi **sysadmin**.

Další informace o příkazech administračního rozhraní API a informace o definování funkcí **admin()** a **task()** naleznete v části *IBM Informix Guide to SQL: Syntax*.

Použití syntaxe onmode v příkazech administračního rozhraní API

Administrační rozhraní API jazyka SQL umožňuje provádění vzdálené administrace pomocí různých specifických příkazů jazyka SQL pro úlohy, jako například správu prostorů, správu konfigurace, spouštění rutinních úloh a ověření systému. Pomocí příkazů EXECUTE FUNCTION můžete v databázi **sysadmin** vyvolat vestavěné funkce **admin()** nebo **task()** a provádět administrativní úlohy, které jsou shodné s prováděním příkazů pomocí obslužného programu **onmode**.

Funkce **task()** a **admin()** smí spouštět pouze administrátor DBSA. Ve výchozím nastavení se však smí připojit k databázi **sysadmin** pouze uživatel **informix**.

Použití syntaxe onmode v příkazech rozhraní Administration API:

1. Vytvořením příkazu EXECUTE FUNCTION vyvolejte úlohu.
2. V části argumentů zadejte příkaz **onmode** a příslušné funkce bez spojovníku před písmenem možnosti.

Pokud je například databáze **sysadmin** aktuální databází, následující příkaz spustí úlohu, která nenásilně vypne databázový server:

```
EXECUTE FUNCTION task ('onmode', 'k');
```

Pokud databáze **sysadmin** není aktuální databází, ale jste uživateli, kteří mají oprávnění k připojení k databázi **sysadmin**, můžete provést následující příkaz:

```
execute function sysadmin:task ('omode', 'k');
```

Další informace o příkazech **onmode** naleznete v části *IBM Informix Dynamic Server Administrator's Reference*.

Procházení dotazů

Dotazy SQL můžete procházet za účelem shromažďování statistických údajů o jednotlivých příkazech SQL provedených v systému a analýzy historie příkazů. Funkce procházení dotazů umožňuje odpovědět například na následující otázky:

- Jak dlouho trvaly příkazy SQL?
- Kolik zdrojů používaly jednotlivé příkazy?
- Jak dlouho trvalo provedení příkazů?
- Kolik času vyžadovalo čekání na jednotlivé zdroje?

Statistické informace jsou uloženy v kruhové vyrovnávací paměti, která je nepravou tabulkou umístěnou v paměti s názvem **sysqltrace** uložené v databázi **sysmaster**. Velikost kruhové vyrovnávací paměti je možné dynamicky měnit.

Ve výchozím nastavení je tato funkce vypnuta, ale můžete ji zapnout pro všechny uživatele nebo pro určitou skupinu uživatelů. Pokud je tato funkce povolena výchozí konfigurací, databázový server zaznamenává posledních 1000 spuštěných příkazů SQL včetně statistiky profilu těchto příkazů.

Tato funkce vyžaduje velkou paměť, pokud chcete zachovat mnoho informací o historii. Výchozí velikost místa požadovaného pro historii trasování je 2 MB. Množství paměti můžete zvětšit nebo zmenšit podle svých požadavků. Pokud pro tuto funkci nechcete používat paměť, můžete trasování historie SQL také zakázat.

K zobrazeným informacím patří:

- ID uživatele, který spustil příkaz
- ID databázové relace
- Název databáze
- Typ příkazů SQL
- Doba provádění příkazu SQL
- Čas dokončení tohoto příkazu SQL
- Text příkazu SQL nebo seznam volání funkcí (také nazývaný *trasování zásobníku*) s typem příkazu, například:
procedura1() volá procedura2() volá procedura3()
- Statistika včetně:
 - Počet vyrovnávacích pamětí pro čtení i zápis
 - Počet čtení i zápisů stran
 - Počet řazení a řazení disku
 - Počet požadavků o zamknutí a čekání
 - Počet záznamů logického protokolu
 - Počet čtení vyrovnávací paměti indexu
 - Odhadovaný počet řádků
 - Odhadovaná nákladovost optimalizátoru
 - Počet vrácených řádků
- Stupeň izolace databáze.

Také můžete zadat následující úrovně překročení informací, které ze mají zahrnovat do trasování:

- *Nízká úroveň* trasování, která je ve výchozím nastavení povolena, sbírá informace uvedené v následujícím příkladu. Tato informace zahrnuje statistiku příkazů, text příkazu a iterátory příkazů.
- *Střední úroveň* trasování sbírá všechny informace, které obsahuje trasování nízké úrovně, a navíc názvy tabulek, názvy databází a zásobníky uložených procedur.
- *Vysoká úroveň* trasování sbírá všechny informace, které obsahuje trasování střední úrovně, a navíc hostitelské proměnné.

Trasované informace mají vliv na množství požadované paměti pro data historie.

Trasování můžete kdykoliv povolit nebo zakázat a zatímco je spuštěn databázový server, můžete změnit počet a velikost vyrovnávacích pamětí trasování. Pokud měníte velikost vyrovnávací paměti trasování, pokusí se databázový server zachovat obsah vyrovnávací paměti. Pokud se parametry zvětší, nebudou zkrácena žádná data. Pokud však počet vyrovnávacích pamětí snížíte, mohou být data ve vyrovnávací paměti trasování zkrácena nebo ztracena.

Počet vyrovnávacích pamětí určuje počet příkazů SQL, kolik příkazů se trasuje. Každá vyrovnávací paměť obsahuje informace o jednom příkazu SQL. Ve výchozím nastavení má vyrovnávací paměť jednotlivých trasování pevnou velikost. Pokud textová informace uložená ve vyrovnávací paměti překročí velikost vyrovnávací paměti trasování, jsou data zkrácena.

Níže je uveden příklad informací o trasování jazyka SQL:

```
select * from syssqltrace where sql_id = 5678;
```

sql_id	5678
sql_address	4489052648
sql_sid	55
sql_uid	2053
sql_stmttype	6
sql_stmtname	INSERT
sql_finishtime	1140477805
sql_begintxtime	1140477774
sql_runtime	30.86596333400
sql_pgreads	1285
sql_bfreads	19444
sql_rdcache	93.39127751491
sql_bfixreads	5359
sql_pgwrites	810
sql_bfwrites	17046
sql_wrcache	95.24815205913
sql_lockreq	10603
sql_lockwaits	0
sql_lockwtime	0.00
sql_logspace	60400
sql_sorttotal	0
sql_sortdisk	0
sql_sortmem	0
sql_executions	1
sql_totaltime	30.86596333400
sql_avgtime	30.86596333400
sql_maxtime	30.86596333400
sql_numioawaits	2080
sql_avgioawaits	0.014054286131
sql_totaliowaits	29.23291515300
sql_rowspersec	169.8958799132
sql_estcost	102
sql_estrows	1376
sql_actualrows	5244
sql_sqlerror	0
sql_isamerror	0
sql_isollevel	2
sql_sqlmemory	32608
sql_numiterators	4
sql_database	db3
sql_numtables	3
sql_tablelist	t1
sql_statement	insert into t1 select {+ AVOID_FULL(sysindices) } 0, tablename

Obrázek 26-6. Vzorové informace o trasování jazyka SQL

Vysvětlení všech řádků tabulky naleznete v tabulce **syssqltrace** v databázové části **sysmaster** příručky *IBM Informix Dynamic Server Administrator's Reference*.

Trasování historie jazyka SQL můžete povolit globálně a zobrazovat tak příkazy, které každý uživatel spustil. Můžete také povolit trasování historie jazyka SQL u jednotlivých uživatelů. Zakázat trasování historie jazyka SQL můžete také globálně nebo u konkrétního uživatele.

Určení spouštěcích informací o trasování jazyka SQL pomocí konfiguračního parametru SQLTRACE

Pomocí konfiguračního parametru SQLTRACE můžete při spuštění databázového serveru řídit výchozí chování trasování. Stanovené informace zahrnují počet příkazů jazyka SQL určených k trasování a režim trasování.

Předpoklad: Každý uživatel, který může měnit soubor `$INFORMIXDIR/etc/SONCONFIG`, může také měnit hodnotu konfiguračního parametru SQLTRACE a provést konfiguraci spuštění. Pouze uživatel **informix**, **root**, správce DBSA nebo uživatel, kterému byla udělena oprávnění k připojení se do databáze **sysadmin** však může pomocí administračních příkazů rozhraní API změnit stav doby spuštění trasovací funkce jazyka SQL.

Postup určení informací o trasování jazyka SQL při spuštění databázového serveru:

1. Do pole **level** doplňte jednu z níže uvedených hodnot:
 - **Low** (nízká). Tato úroveň trasování povolena ve výchozím nastavení shromažďuje statistické údaje o příkazech, texty příkazů a iterátory a jejich iterátory.
 - **Medium** (střední). Tato úroveň trasování shromažďuje všechny informace zahrnuté v nízké úrovni trasování, a navíc zahrnuje názvy tabulek a databází a uložené zásobníky procedur.
 - **High** (vysoká). Tato úroveň trasování shromažďuje všechny informace zahrnuté ve střední úrovni trasování, a navíc hostitelské proměnné.
 - **Off** (vypnutá). Tímto není zadáno žádné trasování jazyka SQL.
2. V poli **nttraces** určete počet příkazů jazyka SQL určených k trasování.
3. V poli **size** určete velikost vyrovnávací paměti trasování v kilobajtech. Dojde-li k překročení zadané velikosti vyrovnávací paměti, databázový server data vyřadí.
4. V poli **mode** určete:
 - **Global** pro všechny uživatele systému nebo
 - **User** pro uživatele, u kterých je povoleno trasování pomocí funkce Administration API **task()** (tuto možnost zvolte, pokud chcete získat ukázkou jazyka SQL, který je spuštěn jen malým počtem uživatelů.)

Více informací o konfiguračním parametru SQLTRACE včetně minimálních a maximálních hodnot některých polí naleznete v části *IBM Informix Dynamic Server Administrator's Reference*.

Příklad: Následující příkaz určuje, že bude databázový server shromažďovat informace o trasování na nízké úrovni až u 2000 příkazů jazyka SQL provedených všemi uživateli systému, a že bude přidělovat přibližně 4 MB paměti (2000 * 2 kilobajty).

```
SQLTRACE level=LOW,nttraces=2000,size=2,mode=global
```

Pokud použijete pouze procentní část přiděleného prostoru vyrovnávací paměti (například 42 procent prostoru vyrovnávací paměti), množství přidělené paměti bude stále 2 kB.

Další informace o trasování jazyka SQL naleznete v části "Procházení dotazů" na stránce 26-11.

Pokud si nepřejete nastavením konfiguračního parametru SQLTRACE restartovat server, můžete spustit následující příkaz ADMIN API, který se vyznačuje stejnou funkcí jako nastavení konfiguračního parametru SQLTRACE:

```
execute function task("set sql tracing on", 100,"1k","med","user");
```

Jakmile v uživatelském režimu povolíte trasovací systém jazyka SQL, můžete potom povolit trasování u každého uživatele. Další informace naleznete v části “Povolení trasování jazyka SQL u konkrétního uživatele”.

Další informace o definování funkcí **admin()** a **task()** naleznete v části *IBM Informix Guide to SQL: Syntax*.

Zakázání trasování historie jazyka SQL globálně nebo pro relaci

Přestože je režim určený v konfiguračním parametru SQLTRACE nastavený na **global** nebo na **user**, v případě, že si přejete úplně vypnout veškerá uživatelská a globální trasování a uvolnit zdroje, které právě používá funkce trasování jazyka SQL, je možné toto trasování zakázat. Ve výchozím nastavení je trasování historie jazyka SQL vypnuto u všech uživatelů.

Pokud si přejete zakázat globální trasování jazyka SQL, spusťte funkci **task()** nebo **admin()**, kde zadejte `set sql tracing off`.

Zákaz globálního trasování jazyka SQL provedete spuštěním následujícího příkazu:

```
EXECUTE FUNCTION task('set sql tracing off');  
(expression) SQL tracing off.  
Byl nalezen 1 řádek (řádky).
```

Pokud si přejete zakázat trasování jazyka SQL pro konkrétní relaci, spusťte funkci **task()** nebo **admin()**, kde zadejte `set sql tracing off` a identifikační číslo relace v následujícím formátu:

```
EXECUTE FUNCTION task("set sql user tracing off",session id);
```

Zákaz trasování jazyka SQL pro relaci s ID 47 provedete pomocí tohoto příkazu:

```
EXECUTE FUNCTION task("set sql user tracing off",47);
```

Další informace o definování funkcí **admin()** a **task()** naleznete v části *IBM Informix Guide to SQL: Syntax*.

Povolení trasování jazyka SQL u konkrétního uživatele

Po zadání uživatele **user** jako režimu v konfiguračním parametru SQLTRACE je nutné spustit funkci Administration API **task()** nebo funkci **admin()**, pomocí kterých zapnete trasování historie jazyka SQL u vybraného uživatele.

Pokud je globální trasování jazyka SQL zakázáno, můžete spustit funkci Administration API **task()** nebo funkci **admin()**, pomocí kterých povolíte trasování SQL u konkrétního uživatele.

Předpoklad: Pouze uživatel **informix**, **root**, správce DBSA nebo uživatel, kterému byla udělena oprávnění k připojení se do datábase **sysadmin** však může pomocí administračních příkazů rozhraní API změnit stav doby spuštění trasovací funkce jazyka SQL. feature.

Pokud si přejete povolit trasování jazyka SQL u konkrétního uživatele, spusťte funkci **task()** nebo funkci **admin()**, kde zadejte `set sql tracing on`.

Povolení trasování jazyka SQL u konkrétního uživatele provedete spuštěním následujícího příkazu:

```
EXECUTE FUNCTION task("set sql user tracing on", session_id);
```

Pokud si přejete povolit trasování historie jazyka SQL na nízké úrovni pro všechny uživatele kromě uživatele **root** nebo **informix**, můžete spustit funkci **task()** nebo funkci **admin()**, kde zadejte `set sql tracing on` a informace definující uživatele.

Například můžete provést následující příkaz, pomocí kterého povolíte trasování příkazů jazyka SQL u uživatelů, kteří jsou právě připojeni k systému, pokud nejsou přihlášení jako uživatel **root** nebo **informix**.

```
dbaccess sysadmin -<<END
execute function task("set sql tracing on", 1000, 1,"low","user");
select task("set sql user tracing on", session_id)
  FROM sysmaster:syssessions
 WHERE username not in ("root","informix");
END
```

Další informace o definování funkcí **admin()** a **task()** naleznete v části *IBM Informix Guide to SQL: Syntax*.

Povolení trasování jazyka SQL na nízké úrovni u všech uživatelů

Ve výchozím nastavení je režim trasování historie jazyka SQL na nízké úrovni povolený u všech uživatelů. Pokud jste zakázali trasování historie jazyka SQL, můžete je obnovit.

Postup povolení trasování historie jazyka SQL na nízké úrovni:

- Jako úroveň zadejte **low** a jako režim v konfiguračním parametru **SQLTRACE** zadejte **global**, čímž povolíte trasování jazyka SQL při spuštění serveru nebo
- spusíte funkci **task()** nebo **admin()**, kde zadejte **set sql tracing on**, čímž trasování jazyka SQL zapnete.

Pomocí následujícího příkazu můžete například povolit globální trasování jazyka SQL na nízké úrovni pro všechny uživatele:

```
EXECUTE FUNCTION task("set sql tracing on", 1000, 1,"low","global");
```

Pokud se k systému přihlásí nový uživatel po spuštění příkazu, je možné u tohoto uživatele trasování povolit. Další informace naleznete v části "Povolení trasování jazyka SQL u konkrétního uživatele" na stránce 26-15.

Další informace o definování funkcí **admin()** a **task()** naleznete v části *IBM Informix Guide to SQL: Syntax*.

Část 7. Dodatky a přílohy

Dodatek. Usnadnění

Cílem IBM je poskytovat produkty pro každého, bez ohledu na věk či zdravotní handicap.

Funkce usnadňující přístup v rámci produktu IBM Informix Dynamic Server

Funkce usnadňující přístup pomáhají uživateli s tělesným postižením, například s postižením pohybového ústrojí či se zrakovým postižením, úspěšně využívat produkty informačních technologií.

Funkce usnadnění přístupu

Následující seznam obsahuje hlavní funkce systému IBM Informix Dynamic Server. Tyto funkce podporují:

- Operace prováděné pouze pomocí klávesnice.
- Rozhraní, která obvykle používají čtečky obrazovky.
- Přílohy alternativního vstupu a výstupu zařízení.

Rada: Informační centrum systému IBM Informix Dynamic Server a související příručky usnadňují přístup pro čtečky IBM Home Page Reader. Všechny funkce můžete ovládat pomocí klávesnice, nemusíte používat myš.

Navigace pomocí klávesnice

Tento produkt používá standardní Microsoft Windows navigační klávesy.

Informace související s usnadněním přístupu

IBM se zavázala usnadnit přístup ke své dokumentaci postiženým osobám. Naše příručky jsou k dispozici ve formátu HTML, umožňují tedy přístup pomocí pomocných technologií, jako je například software pro čtení obrazovky. Diagramy syntaxe jsou v našich příručkách k dispozici ve formátu desítkových čísel oddělených tečkami.

Příručky pro systém IBM Informix Dynamic Server můžete prohlížet ve formátu PDF (Adobe Portable Document Format) pomocí aplikace Adobe Acrobat Reader.

IBM a usnadnění přístupu

Další informace o záležitostech, které má IBM k usnadnění přístupu, naleznete v *Centru IBM pro usnadnění přístupu* na webu <http://www.ibm.com/able>.

Upozornění

IBM nemusí ve všech zemích nabízet produkty, služby a funkce popsané v tomto dokumentu. Informace o produktech a službách, které jsou momentálně ve vaší zemi dostupné, můžete získat od zástupce IBM pro vaši oblast. Žádný z odkazů na produkty, programové vybavení nebo služby IBM není zamýšlen jako tvrzení, že lze použít pouze tyto produkty, programové vybavení nebo služby. Jako náhrada mohou být použity libovolné funkčně ekvivalentní produkty, programové vybavení nebo služby, které neporušují žádná práva IBM na duševní vlastnictví. Za vyhodnocení a ověření provozu jakýchkoli produktů, programů a služeb, které nejsou od IBM, nese odpovědnost uživatel.

IBM může mít patenty nebo podané žádosti o patent, které zahrnují předmět tohoto dokumentu. Získání tohoto dokumentu neposkytuje uživateli licenci na tyto patenty. Písemné dotazy ohledně licencí můžete zaslat na adresu:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Pokud máte zájem o licenci v zemi s double-bajtovou znakovou sadou (DBCS), kontaktujte zastoupení společnosti IBM ve vaší zemi nebo písemně zastoupení společnosti IBM na adrese:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

SPOLEČNOST INTERNATIONAL BUSINESS MACHINES CORPORATION POSKYTUJE TUTO PUBLIKACI TAK "JAK JE" BEZ ZÁRUKY JAKÉHOKOLIV DRUHU, VÝSLOVNĚ VYJÁDŘENÉ NEBO VYPLÝVAJÍCÍ Z OKOLNOSTÍ, VČETNĚ, A TO ZEJMÉNA, ZÁRUK NEPORUŠENÍ PRÁV, PRODEJNOSTI NEBO VHODNOSTI PRO URČITÝ ÚČEL VYPLÝVAJÍCÍ Z OKOLNOSTÍ. Právní řády některých zemí u určitých transakcí nepřipouštějí vyloučení záruk výslovně vyjádřených nebo vyplývajících z okolností, a proto se na vás výše uvedené omezení nemusí vztahovat.

Tato publikace může obsahovat technické nepřesnosti nebo typografické chyby. Informace zde uvedené jsou pravidelně aktualizovány a v nových vydáních této publikace již budou tyto změny zahrnuty. IBM má právo kdykoli bez upozornění zdokonalovat nebo měnit produkty a/nebo programy popsané v této publikaci.

Jakékoli odkazy na webové stránky jiných společností než IBM obsažené v této publikaci mají pouze informační charakter a v žádném případě se nejedná o podporu těchto stránek. Materiály na těchto webových stránkách nejsou součástí materiálů k danému produktu IBM a používání těchto webových stránek je na vaše vlastní nebezpečí.

IBM může použít nebo distribuovat jakékoliv informace, které jí sdělíte, libovolným způsobem, který považuje za odpovídající, aniž by jí tím vznikl jakýkoliv závazek vůči vám.

Držitelé licence tohoto programu, kteří si přejí mít přístup i k takovým informacím za účelem (i) výměny informací mezi nezávisle vytvořenými programy a jinými programy (včetně tohoto) a (ii) vzájemného použití sdílených informací, mohou kontaktovat:

IBM Corporation

J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
USA

Informace tohoto typu mohou být za odpovídajících podmínek dostupné. V některých případech připadá v úvahu zaplacení poplatku.

Program popsáný v tomto dokumentu a všechny materiály s ním související, které podléhají licenci, jsou dodávány IBM v souladu s textem smlouvy mezi zákazníkem a IBM nebo ekvivalentní smlouvy.

Všechny informace o provozu byly určeny v řízeném prostředí. Výsledky obdržené v jiném operačním prostředí se tudíž mohou výrazně lišit. Některá měření byla provedena v systémech s vývojovým prostředím a neexistuje žádná záruka, že tato měření budou stejná v obecně dostupných systémech. Některá měření byla odhadnuta extrapolací. Skutečné výsledky se mohou lišit. Uživatelé tohoto dokumentu by měli ověřit vhodnost dat pro svá specifická prostředí.

Informace týkající se produktů jiných společností byly získány od dodavatelů těchto produktů, z jejich tištěných materiálů nebo z jiných veřejně dostupných zdrojů. IBM netestovala tyto produkty a nemůže potvrdit spolehlivost jejich provozu, kompatibilitu nebo jiné tvrzení týkající se těchto produktů. Otázky týkající se možností produktů jiných společností by měly být adresovány dodavatelům těchto produktů.

Všechna tvrzení o budoucím zaměření nebo úmyslech IBM mohou být bez upozornění změněna nebo zrušena a představují pouze hrubý nástin cílů a podmínek společnosti.

Všechny uvedené ceny jsou současnými cenami navrženými IBM a mohou být bez upozornění změněny. Ceny prodejců se mohou od těchto cen lišit.

V rámci informací zde uvedených jsou uvedeny příklady sestav a dat, které jsou používány při každodenních operacích. Kvůli úplnosti obsahují příklady jména jednotlivců, společností, značek a produktů. Všechna tato jména jsou smyšlená a jakákoli podobnost s existujícími jmény či adresami je čistě náhodná.

LICENCE NA AUTORSKÁ PRÁVA:

Tyto informace obsahují ukázkové programy ve zdrojovém jazyce, které ilustrují programovací techniky na různých platformách. Můžete kopírovat, měnit a distribuovat tyto vzorové programy v libovolné formě bez nutnosti úhrady IBM, za účelem vývoje, používání, marketingu nebo distribuce aplikačních programů přizpůsobených aplikačnímu programovému rozhraní pro operační platformu, pro kterou byly vzorové programové napsány. Tyto příklady nebyly důkladně testovány při veškerých podmínkách. IBM proto nemůže zaručit nebo spolehlivost, možnost opravy či funkčnost těchto programů ani tyto záruky nelze nijak vyvozovat. Můžete kopírovat, změnit a distribuovat tyto vzorové programy v libovolné formě bez úhrady IBM, za účelem vývoje, použití, marketingu nebo distribuce aplikačních programů odpovídajícím aplikačnímu programovému rozhraní IBM.

Každá kopie nebo část těchto vzorových programů nebo jakákoli odvozená práce musí zahrnovat následující doložku o autorských právech:

© (název vaší firmy) (rok). Části tohoto kódu jsou odvozeny od kódu vzorových programů IBM Corp. Vzorové programy. © Copyright IBM Corp. (zadejte rok nebo roky). Všechna práva vyhrazena.

Pokud si prohlédíte tyto informace v souboru, nemusí se fotografie a barevné ilustrace zobrazit.

Ochranné známky

Následující termíny se používají v publikacích k produktu IBM Informix a jsou zároveň ochrannými známkami společnosti International Business Machines Corporation ve USA a případně v dalších jiných zemích:

AIX	OS/2
AnalystStudio	OS/390
C-ISAM	OS/400
Cloudscape	Passport Advantage
DataBlade	ProjectConsole
DB2	Rational
DB2 Connect	Rational Suite
DB2 Universal Database	Rational Unified Process
Distributed Relational Database Architecture	RedBack
Domino	Red Brick
DRDA	RequisitePro
Dynamic Connect	RETAIN
IBM	SQL/DS
Informix	SystemBuilder
Lotus	Tivoli
Lotus Notes	TME 10
MQSeries	UniData
NetView	UniVerse
Notes	WebSphere
NUMA-Q	z/OS

Adobe, Acrobat, Portable Document Format (PDF) a PostScript jsou ochranné známky nebo registrované ochranné známky Adobe Systems Incorporated v USA a případně v dalších jiných zemích.

Intel, logo Intel, Intel Inside, logo Intel Inside, Intel Centrino, logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium a Pentium jsou ochranné známky nebo registrované ochranné známky společnosti Intel Corporation či jejích dceřiných společností v USA a případně v dalších jiných zemích.

Java a všechny ochranné známky a loga související s jazykem Java jsou ochranné známky Sun Microsystems, Inc. v USA a případně v dalších jiných zemích.

Microsoft, Windows, Windows NT a logo Windows jsou ochranné známky Microsoft Corporation v USA a případně v dalších jiných zemích.

UNIX je registrovaná ochranná známka The Open Group v USA a případně v dalších jiných zemích.

Linux je registrovaná ochranná známka Linuse Torvaldse v USA a případně v dalších jiných zemích.

Další názvy společností, produktů nebo služeb mohou být ochranné známky nebo servisní známky jiných stran.

Rejstřík

Speciální znaky

(*), hvězdička

Viz též zástupný znak.

zástupný znak v poli názvu hostitele 3-28

Čísla

32bitová platforma

a společná oblast vyrovnávacích pamětí 10-14

64bitové adresování

definované 7-33

maximální počet vyrovnávacích pamětí 7-11

podpora databázového serveru 1-4

společná oblast vyrovnávací paměti 7-11

využití paměti 7-33

64bitové platformy

a společná oblast vyrovnávacích pamětí 10-14

A

ADM.

Viz Administrační třída virtuálních procesorů.

administrační rozhraní API 26-10

přehled 26-1

administrační třída virtuálních procesorů 5-11

adresa IP

použití v poli názvu hostitele 3-27

zjištění 3-28

adresáře

NFS 9-3

adresy IPv4 3-30

adresy IPv6 3-30

ADT.

Viz Virtuální procesor typu audit.

agregace paketů 3-42

agregační funkce

paralelní zpracování 5-6

AIO

virtuální procesory 5-21

aktualizace dat

tabulky typu RAW 9-25

tabulky typu STANDARD 9-24

alarm události

definovaný 1-23, 2-8

dynamicky přidáné protokoly 14-16, 14-17

sekundární server RS 21-8

sekundární server SD 22-6

asynchronní vstup - výstup

definované 5-19

jádra (KAIO) 5-18

asynchronní vstup - výstup jádra

na platformách Linux 5-20

na platformách UNIX 5-20

neprotokolující diskový vstup - výstup 5-18

přehled 5-19

automatická obnova, dvoufázové potvrzování 24-7

automatické

spuštění databázového serveru 1-14

vypnutí 1-15

B

běžné vyrovnávací paměti

události, které způsobují vyprázdnění 7-27

blokové zařízení 9-4

bloky

činnost během obnovy zrcadlení 17-4

definované 1-16

chyby vstupu - výstupu během zpracování 23-5

koncepty 9-3

kontrola stavu 10-35, 23-5

maximální počet 1-16, 9-3, 10-8

maximální velikost 1-5, 1-16, 9-3, 10-8

monitorování 10-34, 10-35, 23-5

název při přidělení jako přímé zařízení 9-4

oblastí 9-8

obnovení vypnutého bloku 18-4

počáteční přidělení 10-6

podpora velkých bloků 1-17

pravidla pro rozvržení disků 9-34

propojení s názvem cesty 1-6, 10-5

propojení s názvem cesty 18-3

překročení maximální velikosti při použití správce LVM 9-38

přidání pomoci

obslužný program ON-Monitor 10-19

přidat k

pomocí programu ISA nebo obslužného programu

onspaces 10-18

prostoru dbspace 10-18

prostory sbspace 9-18, 10-25

zrcadlený prostor dbspace 18-5

seznam volných stránek

monitorování 16-2

spojení a rozdělení stránek 11-3

tabulka, definovaná 7-16

uložení stavu na sekundární server 20-11

určení oblasti metadat 10-25

vypuštění z

blobspace 10-28

prostoru dbspace 10-28

prostory sbspace 10-28

zálohování 10-18

zápis

kontrolní body 7-29

monitorování 8-11

brána IBM Informix Gateway při heterogenním potvrzování 24-19

C

centrální registr

sqlhosts 3-14

cron

obslužný program 1-26

Č

čísla portů 3-16

čítač programu a data jednotkových procesů 7-18

čtení napřed

definované 7-26

kdy se používá 7-26

čtení napřed (*pokračování*)
konfigurační parametr RA_PAGES 7-26
konfigurační parametr RA_THRESHOLD 7-26

D

další oblast

velikost 9-8

data

odhad diskového prostoru 9-33

data typu TEXT a BYTE

absence komprese 10-46

monitorování v prostoru dbspace 10-40

zápis do prostorů blobspace 7-30

zavádění 10-46

databáze

asynchronní vstup - výstup 5-19

definované 9-21

fragmentace 9-22

kompatibilní s ANSI 11-7

monitorování 10-34, 12-6

odhadnutí velikosti 9-33

omezení velikosti 9-22

sysutils 4-7

účel 9-21

umístění 9-21

zobrazení stavu protokolování 12-7

databáze bez protokolování

podporované typy tabulek 9-24

příkazy jazyka SQL

nikdy neprotokolované 11-4

tabulky typu RAW 9-24

databáze kompatibilní se standardem ANSI

obslužný program ondblog 12-3

obslužný program ontape 12-5

změna režimu protokolování 12-3

databáze s protokolováním

podporované typy tabulek 9-24

příkazy jazyka SQL

nikdy neprotokolované 11-4

vždy protokolované 11-3, 11-4

tabulky typu RAW 9-24

tabulky typu STANDARD 9-24

databáze sysadmin 26-1

databáze sysadmin 26-4

popis 26-2

tabulky 26-3

databáze sysmaster

tabulky SMI 1-25

vytvoření 4-7

databáze sysutils

vytvoření 4-7

databázové servery

32bitové a 64bitové verze 1-4

plánování podle priorit 1-13

požadavky vytvoření databáze 11-7

protokol zpráv 1-23

Připojení prostřednictvím SQL API 1-16

s vícenásobnými jednotkovými procesy 5-2

skupiny 3-23, 3-24, 3-25, 3-27

Enterprise Replication 3-24

replikace HDR 3-25

spuštění 1-14

vzdálené 3-12

datové typy

CLOB a BLOB 9-14

replikováno v replikaci dat 19-3

datové typy (*pokračování*)

uživatelský 9-14

datový typ BLOB 9-12

DDL.

Viz jazyk Data Definition Language.

definice fyzické konzistence 15-8

diagnostická data

nastavení parametrů 23-6

požadavky na diskový prostor 23-7

sběr 23-6

disková stránka

čtení napřed 7-26

předobrazy ve fyzickém protokolu 7-27

diskový prostor

blok, maximální velikost 1-16, 10-8

cesty k blokům, posuny 10-4

definovaný 10-3

inicializace 1-14, 4-2, 4-3, 4-5

monitorování pomocí programu ISA 10-37

oddíly uprostřed disku 9-35

odhadnutí velikosti 9-31

požadavky 9-33

pravidla pro rozvržení 9-33

přídělení

diskový prostor s přímým přístupem 10-5

prostor v předpřipraveném souboru 10-5

v systému Windows 10-6

sledování využití prostorem tblspace 9-28

ukládání dat typu TEXT a BYTE 10-23

vytvoření propojení s názvem cesty bloku 1-6, 10-5

diskový prostor s přímým přístupem

definovaný 9-4

přídělení v systému UNIX 9-4, 10-5

přídělení v systému Windows 9-3

znakově orientované rozhraní 9-4

diskový vstup - výstup

asynchronní vstup - výstup jádra 5-18

čtení ze zrcadlených bloků 17-5

fronty 5-21

fyzický protokol 5-18

chyby během zpracování 23-5

inteligentní velké objekty 9-13

logický protokol 5-18

priority 5-18

přímý vstup - výstup 9-4

role sdílené paměti při omezování 7-2

třídy virtuálních procesorů 5-17

vstup - výstup prostřednictvím operačního systému 9-4

vyrovnávací paměti 10-14

zapisuje do zrcadlených bloků 17-5

distribuované databáze 1-22

distribuované dotazy

definované 1-22

dvoufázové potvrzování 1-23

nastavení sqlhosts 1-10

distribuované transakce 11-2

protokol dvoufázového potvrzování 24-5

protokolování bez vyrovnávací paměti 11-7

určení, zda byla implementována nekonzistentně 25-2

distribuované zpracování 24-2

dlouhá transakce

definovaný 13-5

důsledky 13-6

dvoufázové potvrzování 24-8, 24-11, 24-12

heuristické odvolání transakce 24-13

prevence 1-20

dočasné logické protokoly 14-8

- dočasné prostory dbspace
 - vytvoření 10-17
- dočasné tabulky
 - Enterprise Replication 9-24
 - inteligentní velký objekt, dočasný 9-19
 - obnovení 9-25
 - záloha 9-26
- dočasný inteligentní velký objekt
 - definovaná 9-20
 - vytvoření 9-20
- dočasný prostor dbspace
 - DBSPACETEMP 9-26
 - definovaný 10-17
 - objem požadovaný pro dočasné tabulky 9-33, 19-33
 - replikace dat 19-32, 20-4
- dočasný prostor sbpace
 - definovaný 9-19
 - charakteristiky 9-19
 - přidávání bloků 9-18, 10-26
 - příklad 10-26
 - příznak LO_CREATE_TEMP 10-27
 - vypouštění bloků 9-20, 10-26, 10-27
 - vypuštění prostoru sbpace 10-29
 - vytvoření 10-26
 - zálohování a obnovení 9-20
- doména
 - systému Windows
 - definovaná 3-3
 - důvěryhodná 3-3
 - řadič 3-3
 - uživatelské účty 3-3, 3-4
 - Windows
 - spuštění jako určitý uživatel 1-27
- doporučení
 - kontrola konzistence 23-2
 - přidělování diskového prostoru 9-5
 - zrcadlení fyzického protokolu 15-3
- dostupnost
 - jako cíl efektivního rozvržení disků 9-33
 - kritická data 9-9
 - ukázkové rozvržení disků 9-35
- dotaz pro podporu rozhodování
 - konfigurační parametr DS_MAX_QUERIES 2-7
- důvěryhodná
 - doména 3-3
- důvěryhodná doména 24-7
- důvěryhodné
 - databázový server 3-3
 - domény systému Windows 3-3
 - klienty 3-3
- Dynamic Server
 - služba 1-14
- dynamické přidělování protokolu
 - alarmy událostí a zprávy 14-16
 - definice 14-9
 - přehled 1-20
 - umístění souborů 14-10
 - velikost souboru protokolu 14-10
- dynamické přidělování zámků 7-13

E

- e-mail
 - oznamování alarmů událostí 2-8
- Enterprise Replication
 - použití skupin databázových serverů 3-24
 - prostory sbpace 9-13

- Enterprise Replication (*pokračování*)
 - protokolování databáze 11-2
 - tabulky typu RAW 9-24
 - tabulky typu STANDARD 9-24
 - tabulky typu TEMP 9-25
 - určení prostorů sbpace 9-13
- externí zálohování
 - a obnovení
 - použití při nastavení replikace HDR 20-6
 - použití při nastavení sekundárního serveru RS 21-5
- externí zdroje specifikace XA 24-3

F

- FAT.
 - Viz* Tabulka FAT.
- fáze po rozhodnutí 24-6
- fáze před potvrzením 24-6
- FQDN 3-12
- fragment
 - monitorování
 - požadavků na vstup - výstup 10-35
 - využití disku 10-38
 - na více disků 9-35
 - přeskakování
 - nedostupné fragmenty 10-33
 - nepřístupných fragmentů 10-31
 - pomocí parametru DATASKIP 10-32
 - vybrané fragmenty 10-33
 - tabulky a indexy 9-23
- fronta čekajících procesů
 - definované 5-11
 - zámky vyrovnávacích pamětí 7-22
- fronta připravených procesů
 - definované 5-10
 - přesunutí jednotkového procesu 5-11, 19-7
- fronta spících procesů, definovaná 5-11
- fronty
 - čekajících procesů 5-11
 - definované 5-10
 - diskový vstup - výstup 5-17
 - připravených procesů 5-11
 - spánek 5-11
- fronty FIFO/LRU
 - definované 7-22
 - určení informací 7-22
- fronty FLRU
 - definované 7-22
 - způsob výběru databázovým serverem 7-23
- fronty LRU
 - definované 7-22
 - fronty FLRU 7-22
 - fronty MLRU 7-22
 - konfigurace více front 7-23
 - správa společné oblasti vyrovnávací paměti 7-23
 - stránky seřazené sestupně podle doby nepoužívání 7-23
 - určení informací 7-22
- fronty MLRU
 - definované 7-22
 - konec čištění 7-25
 - omezení počtu stránek 7-24
 - ukončení čištění stránek 7-25
 - umisťování vyrovnávacích pamětí 7-23
- funkce admin() 26-3
- funkce GLS (Global Language Support) 1-8
- funkce ifx_lo_copy 9-20
- funkce ifx_lo_specset_flags 9-20, 10-27

- funkce `mi_lo_copy()` 9-20
- funkce `mi_lo_specset_flags()` 9-20, 10-27
- funkce Nastavení serveru 1-12
 - vlastní nastavení konfigurace 1-12
- funkce `task()` 26-3
- fyzické paměťové jednotky, seznam 9-2
- fyzické protokolování
 - definice 15-2
 - inteligentní velké objekty 15-2
 - lehká připojení 10-29
 - podrobnosti o protokolování 15-3
 - protokolovaná aktivita 15-2
 - zálohy 15-3
- fyzický protokol
 - máte-li jiné než výchozí velikosti stránek 10-13
 - monitorování 16-2
 - pokyny týkající se velikost 15-3
 - přehled 1-20
 - přetečení 15-4
 - přetečení během rychlé obnovy 15-7
 - role v rychlé obnově 15-7
 - role v rychlé obnově po kontrolním bodu 15-8
 - souvislý prostor 16-2
 - virtuální procesory 5-19
 - vstup - výstup, virtuální procesory 5-19
 - vyrovnávací paměť 7-12
 - zálohování 10-8
 - změna velikosti a umístění 16-2
 - možné metody 16-1
 - pomocí programu ON-Monitor 16-2
 - pomocí textového editoru 16-2
 - zprávy fyzické obnovy 15-2
 - zvětšení velikosti 10-13, 16-2

G

- gcrcore
 - obslužný program 2-9
 - soubor 23-4
- globální společná oblast, definovaná 7-20
- GLS
 - Viz* Global Language Support
- GTRID 25-3

H

- haldy 7-19
- hardwarové zrcadlení 17-3
- hashovací tabulka 7-16
- HDR
 - Viz* High-Availability Data Replication.
- hesla
 - změna pro uživatele informix 1-26
- heterogenní potvrzování 24-19, 24-20, 24-21, 24-23
- heuristické odvolání transakce
 - definice 24-10
 - dlouhá transakce 24-13
 - ilustrace a záznamy protokolu 24-16
 - indikace výskytu odvolání transakce 25-2
 - monitorování, příkaz `onstat -x` 24-13
 - výsledkem (čeho) 24-11
- heuristické rozhodnutí
 - nezávislá akce 24-9
 - typy 24-9
- heuristické ukončení transakce
 - definice 24-12

- heuristické ukončení transakce (*pokračování*)
 - ilustrace a záznamy protokolu 24-17
 - kdy je nezbytné 24-12
 - určení vlivu na transakci 25-2
 - vracené zprávy 24-13
 - výsledkem (čeho) 24-13
- historie trasování SQL
 - vyrovnávací paměť pro 26-12
- hodnota klíče
 - sdílená paměť 7-7
- hodnota `lru_min_dirty` 15-6
- hodnota parametru `lru_max_dirty` 7-22, 7-25, 10-15
 - příklad použití 7-25
- hodnota parametru `lru_min_dirty` 7-22, 7-25, 10-15
 - ve vztahu k čištění stránek 7-26
- hodnota parametru `lrus` 10-15
- hodnota pole buffers 10-14
- hodnota `ru_max_dirty` 15-6
- hodnota STDIO 20-9, 21-6
- HPL
 - Viz* zavaděč High Performance Loader
- hvězdička (*)
 - Viz též* zástupný znak.
 - zástupný znak v poli názvu hostitele 3-28

CH

- chyba aserce
 - během kontroly konzistence 23-3
 - během zpracování požadavku uživatele 23-4
 - definovaná 23-1
 - formát protokolu zpráv 23-3
 - poškození dat 23-5
 - zjištění příčiny 23-6
- chybové zprávy
 - chyby vstupu - výstupu na bloku 23-5
 - protokol dvoufázového potvrzování 24-14
- chyby vstupu - výstupu bránící v provozu
 - destruktivní a nedestruktivní 23-7
 - kdy nastane 23-7
 - monitorování
 - alarmy událostí 23-8
 - protokol zpráv 23-8

I

- IBM Informix Server Administrator 1-23
 - funkce Nastavení serveru 1-12
 - konfigurace `sqlhosts` 1-10
 - monitorování diskové paměti 10-37
 - monitorování zámků latch a zámků spin 8-8
 - obnovení vypnutého bloku 18-6
 - použití obslužného programu `onmode` 20-15
 - provádění příkazů obslužného programu 8-10
 - přidání zrcadlených bloků 18-6
 - přidávání bloků 10-18, 10-25
 - režimy databázového serveru 4-10
 - sdílená paměť 8-5
 - spustit zrcadlení 18-5
 - spuštění 1-12
 - stav protokolování databáze 12-5
 - ukončení zrcadlení 18-7
 - úpravy konfigurace 2-11
 - úpravy souboru `sqlhosts` 1-10
 - uživatelská oprávnění 4-10
 - virtuální procesory 6-2

- IBM Informix Server Administrator *(pokračování)*
 - vytvoření
 - prostorů dbspace 1-16, 10-9
 - prostory sbspace 10-25
 - vytvoření prostorů dbspace 1-16
 - změna provozních režimů 4-10
 - změna režimů databázového serveru 4-10
 - zobrazení stavu protokolování 12-7
 - zobrazení zpráv 1-23
 - IBM Informix Storage Manager 1-17
 - ID
 - v logickém protokolu 13-2
 - ID databázového serveru 2-2
 - ID protokolu 13-2
 - identifikátor globální transakce 25-3
 - identifikátor segmentu 7-7
 - imc jako část pole určující protokol 3-18
 - index
 - fragmentace 9-23
 - ověření 23-3
 - paralelní vytváření 5-6
 - prostor tblspace 9-29
 - index B-stromu
 - jednotkový proces čištění 7-18
 - INFORMIXSERVER 1-7
 - inicializace
 - databáze sysmaster 4-7
 - databáze sysutils 4-7
 - diskového prostoru 1-14
 - diskový prostor 4-2, 4-3, 4-5
 - konfigurační soubory 4-3, 4-6
 - kontrolní bod 4-7
 - kroky 4-3
 - maximální počet připojení 4-7
 - protokol zpráv 4-6
 - převod interních souborů 4-5
 - sdílená paměť 4-1, 4-3
 - tabulky SMI 4-7
 - virtuální procesory 4-5
 - vynucená rezidence 4-6
 - instalace
 - klíč registru sqlhosts 3-14
 - MaxConnect 3-42
 - Windows 1-6
 - Instance Manager 1-11, 1-12, 1-13
 - instmgr.exe 1-11
 - integrita předávaných zpráv 3-8
 - inteligentní velké objekty
 - AVG_LO_SIZE 9-16, 10-25
 - čas posledního přístupu 9-16
 - definovaný 9-12
 - fyzické protokolování 15-3
 - metadata 9-13
 - oblasti a bloky 9-8
 - odhadovaná velikost 10-26
 - odlehčená verze vstupu - výstupu 9-17
 - odlehčené verze vstupu - výstupu 7-11
 - odstranění dat CLOB nebo BLOB 10-28
 - odstraňování, nulový počet odkazů 10-29
 - paměť 9-18
 - paměťové charakteristiky 9-13, 9-15, 10-26
 - použití protokolování 13-8
 - použití replikace Enterprise Replication 9-13
 - použití vyrovnávací paměti
 - doporučení 7-30
 - režim 7-32
 - prostorů dbspace 9-7
 - inteligentní velké objekty *(pokračování)*
 - protokolování 9-17, 13-7
 - přístup v aplikacích 9-19
 - příznak LO_CREATE_TEMP 9-20
 - statistické údaje pro uživatelem definovaná data 2-3
 - stránky sbpage 9-7
 - ukládání do vyrovnávací paměti 9-16
 - určení datových typů 9-14
 - určení velikosti oblasti metadat 10-45
 - uživatelská data 9-13
 - vlastnosti vstupu - výstupu 9-14
 - vyhledávání dat 9-13
 - výpočet průměrné velikosti velkého objektu 9-15
 - výpočet velikosti oblasti 9-8, 10-25
 - vyrovnávací paměti ve sdílené paměti 7-32
 - vytvoření 9-14, 13-9
 - zamykání 9-13, 9-17
 - zamykání rozsahu bajtů 7-13
 - zapnutí nebo vypnutí protokolování 13-8
 - Internet Assigned Numbers Authority 3-16
 - ipcshm
 - jednotkové procesy cyklického dotazování odpovídající paměťovým segmentům 6-4
 - procesy, které komunikují prostřednictvím této sdílené paměti 7-6
 - sdílená paměť
 - velikost 7-20
 - ISA
 - Viz IBM Informix Server Administrator.
 - ISM
 - Viz IBM Informix Storage Manager.
 - ixpasswd.exe 1-26
 - izolace tabulek 9-34
- ## J
- J/Foundation
 - instalační adresář ovladače JDBC 1-7
 - konfigurace 1-13
 - nastavení proměnné CLASSPATH 1-7
 - proměnné prostředí 1-9
 - Java Development Kit
 - soubor krakatoa.jar 1-7
 - jazyk DDL
 - příkazy 19-33
 - jazyk DDL (Data definition language)
 - protokolované příkazy 11-3
 - jazyk ESQL/C
 - přístup k inteligentním velkým objektům 9-13, 9-19
 - JDK
 - Viz Java Development Kit.
 - jednoduché velké objekty
 - deskriptor 7-32
 - ilustrace paměťového prostoru blobspace 7-32
 - vyrovnávací paměti 7-31
 - vytvoření v prostoru blobspace 7-31
 - zápis na disk 7-31
 - jednoprocesorový počítač 2-7, 5-14
 - jednotkové procesy
 - asynchronní vstup - výstup jádra 5-20
 - definované 5-3
 - haldy 7-19
 - interní 5-3, 5-12
 - jak pracují virtuální procesory 5-20
 - jednotkové procesy čištění stránek 5-2, 5-3, 7-17
 - klientské aplikace 5-3
 - kontext 5-8

- jednotkové procesy (*pokračování*)
 - migrace 5-10
 - několik současných 5-8
 - obnova 5-3
 - onmode 5-3
 - plánování a synchronizace 5-8
 - primární relace 7-18
 - probuzení 5-9
 - prohledávání B-stromu 5-3
 - předávání řízení 5-8
 - přepínání mezi jednotkovými procesy 5-8
 - přístup ke sdíleným vyrovnávacím pamětím 7-22
 - přístup ke zdrojům 5-8
 - relace 5-10, 7-18
 - řídící blok 5-8
 - řízení souběžného přístupu 7-21
 - spící 5-11
 - uživatelský 5-3, 7-18
 - vztah k procesu 5-3, 5-12
 - zásobníky 7-18
 - zrcadlení 5-3
- jednotkové procesy cyklického dotazování
 - definované 5-23
 - fronty zpráv 5-24
 - kolik 5-22, 5-23
 - konfigurační parametr DBSERVERNAME 5-23
 - konfigurační parametr FASTPOLL 5-25
 - na virtuálních procesorech CPU nebo síťových 5-22
 - položka nettype 5-23
 - připojení 5-23
 - více procesů pro protokol 5-22
- jednotkové procesy čištění stránek
 - definované 7-27
 - monitorování 7-17
 - role při zápisu bloků 7-29
 - upozorňované při zápisech na popředí 7-28
 - vyprazdňování běžných vyrovnávací paměti 7-27
 - vyprazdňování společné oblasti vyrovnávací paměti 7-27
- jednotkové procesy předávající řízení
 - definované 5-10
 - fronta připravených procesů 5-10
 - podmínky 5-8
 - přepínání mezi jednotkovými procesy 5-8
 - v předem stanoveném okamžiku 5-9
- jednotkové procesy typu listener
 - definované 5-23
 - přidání 5-26
 - více karet rozhraní 5-27
- jednotkový proces DRDAEXEC 3-36
- jednotkový proces KAIO 5-18, 5-20
- jednotkový proces sqlxec
 - klientská aplikace 5-24
 - role při spojení klienta/serveru 5-24
 - uživatelský jednotkový proces 5-3
- jiná než výchozí velikost stránky
 - pro prostor dbspace 10-12
 - v prostředí replikace HDR 20-4
 - vytvoření společné oblasti vyrovnávacích pamětí pro 10-13

K

- klávesové zkratky
 - klávesnice A-1
- klíč shmkey
 - definovaná 7-7
 - připojení dalších segmentů 7-8

- klíčová slova MODE ANSI
 - stav protokolování databáze 11-6
- klíčové slovo ALL
 - DATASKIP 10-33
- klíčové slovo DEFAULT
 - v příkazu SET DATASKIP 10-32
- klidový režim 4-8
- klient-server
 - konfigurace
 - jednotkové procesy typu listener a cyklického dotazování 5-23
 - místní zpětná smyčka 3-8
 - sdílená paměť 3-6
 - příklad konfigurace
 - místní zpětná smyčka 3-38
 - prostřednictvím protokolu IPX/SPX 3-40
 - sdílená paměť 3-37
 - síťové připojení 3-39
 - více typů připojení 3-40
 - vícenásobné databázové servery 3-41
 - vícenásobné uložení 3-41
- klientská aplikace
 - definovaná 3-2
 - globální transakce 24-2
 - konfigurace propojitelnosti 1-9, 1-16, 3-9
 - konfigurace prostředí 1-8
 - konfigurace velikosti zásobníku 7-19
 - multiplexní připojení 3-4
 - podporovaná připojení 3-5
 - pole názvu hostitele 3-19
 - pole typu připojení 3-17
 - pole voleb 3-21
 - položky sqlhosts 3-13, 3-15, 3-28
 - použití replikace dat 19-4
 - proměnná prostředí ONCONFIG 1-7
 - přesměrování při replikaci dat 19-23, 20-20
 - připojení k hostiteli 3-13
 - připojení k primárnímu nebo sekundárnímu serveru 19-3
 - připojení ke sdílené paměti 7-6
 - připojení prostřednictvím místní zpětné smyčky 3-8
 - připojení prostřednictvím sdílené paměti 3-7
 - reakce na selhání replikace HDR 20-19
 - síťové domény systému 3-4
 - soubory zabezpečení sítě 3-12
 - určení adresy pomocí zástupných znaků 3-30
 - určení názvu dbservername 3-31
 - vzdálení hostitelé 3-12
 - zahájení relace 7-18
- kolize požadavků na přístup k disku
 - omezení 9-33
- komunikační prostředky DRDA
 - nastavení 3-34
 - přehled 3-33
 - připojení ke klientům 3-34
 - určení informací pomocí parametru NETTYPE 3-35
 - určení velikosti vyrovnávací paměti pomocí parametru DRDA_COMMBUFFSIZE 3-35
 - zobrazení informací 3-37
- komunikační protokol TCP/IP
 - číslo naslouchajícího portu 3-30
 - pole názvu hostitele 3-19
 - pole názvu služby 3-20
 - použití
 - adresa IP sítě Internet 3-30
 - číslo naslouchajícího portu TCP 3-30
 - protokol IPX/SPX 3-40
 - soubor hosts.equiv 3-11

komunikační protokol TCP/IP (*pokračování*)
 použití (*pokračování*)
 více portů 3-11
 zástupné znaky 3-28

konfigurace
 J/Foundation 1-13
 monitorování 2-11
 odhad potřebného diskového prostoru 9-33
 plánování databázového serveru 1-3
 použití funkce Nastavení serveru 1-12
 požadavky 1-2
 prostředí databázového serveru 1-2
 více portů 3-11
 vlastnosti relace 1-17
 Windows 1-4

konfigurace disku 1-3
 konfigurace paměťových zařízení 1-17
 konfigurační parametr ADMIN_MODE_USERS 4-8, 4-15
 konfigurační parametr ADMIN_USER_MODE_WITH_DBSA 4-8, 4-9
 konfigurační parametr ADTERR 2-10
 konfigurační parametr ADTMODE 2-10, 5-28
 konfigurační parametr ADTPATH 2-11
 konfigurační parametr ADTSIZE 2-11
 konfigurační parametr AFF_NPROCS 6-2
 konfigurační parametr AFF_SPROC 5-13, 6-2
 konfigurační parametr ALARMPROGRAM 1-23, 2-9
 konfigurační parametr ALRM_ALL_EVENTS 2-9
 konfigurační parametr AUTO_AIOVPS 2-7, 5-21
 konfigurační parametr AUTO_CKPTS 15-5
 konfigurační parametr AUTO_LRU_TUNING 16-5
 konfigurační parametr BUFFERPOOL 2-5, 7-10, 7-22, 10-13, 15-6
 a inteligentní velké objekty 7-32
 konfigurační parametr CDR_QDATA_SBSpace 9-13
 konfigurační parametr CKPTINTVL 15-6
 definovaný 2-5
 konfigurační parametr CLEANERS 2-6
 účel 7-17
 konfigurační parametr CONSOLE 1-25, 2-4
 konfigurační parametr DATASKIP 2-7
 definovaný 10-31
 klíčové slovo ALL 10-33
 konfigurační parametr DB_LIBRARY_PATH 2-10
 konfigurační parametr DBCREATE_PERMISSION 2-10
 konfigurační parametr DBSERVERALIAS 1-7, 2-2
 definovaný 3-32
 příklad 3-32
 soubor sqlhosts 3-17
 více typů připojení 3-40
 konfigurační parametr DBSERVERNAME 1-7, 2-2
 definovaný 3-31
 pravidla syntaxe 3-17
 přidružený protokol 5-23
 soubor sqlhosts 3-17
 virtuální procesor pro jednotkový proces cyklického dotazování 5-23
 konfigurační parametr DBSPACETEMP 2-3, 9-26, 10-17
 definování dočasných tabulek 9-34
 rozložení zatížení 9-34
 konfigurační parametr DD_HASHMAX 2-5
 konfigurační parametr DD_HASHSIZE 2-5
 konfigurační parametr DEADLOCK_TIMEOUT 2-8, 24-18
 protokol dvoufázového potvrzování 24-18
 konfigurační parametr DEF_TABLE_LOCKMODE 2-5, 7-13
 konfigurační parametr DIRECT_IO 10-17
 konfigurační parametr DIRECTIVES 2-9
 konfigurační parametr DRAUTO 2-8
 konfigurační parametr DRAUTO (*pokračování*)
 role v obnově po selhání replikace dat 20-19, 20-20, 20-24
 konfigurační parametr DRDA_COMMBUFFSIZE 3-35
 konfigurační parametr DRIDXAUTO 2-8, 20-13
 konfigurační parametr DRINTERVAL 2-8
 nastavení pro
 asynchronní aktualizace 19-8
 synchronní aktualizace 19-8
 konfigurační parametr DRLOSTFOUND 2-8, 19-9
 konfigurační parametr DRTIMEOUT 2-8, 20-13
 zjištění selhání replikace dat 20-19
 konfigurační parametr DS_HASHSIZE 7-19, 8-4
 konfigurační parametr DS_MAX_QUERIES 2-7
 konfigurační parametr DS_MAX_SCANS 2-7
 konfigurační parametr DS_NONPDQ_QUERY_MEM 2-7, 7-15
 konfigurační parametr DS_POOLSIZE 7-19, 8-4
 konfigurační parametr DS_TOTAL_MEMORY 2-7, 7-14
 konfigurační parametr DUMPCORE 2-9, 23-6
 konfigurační parametr DUMPDIR 2-9, 23-4, 23-6
 konfigurační parametr DUMPGCORE 2-9, 23-4, 23-6
 konfigurační parametr DUMPSHMEM 2-9, 23-4, 23-6
 konfigurační parametr DYNAMIC_LOGS 2-3
 povolení a zakázání 14-9
 přidání souborů protokolu 14-11, 14-16
 úprava hodnoty 14-15
 konfigurační parametr ENCRYPT_CIPHERS 20-14
 konfigurační parametr ENCRYPT_HDR 20-14
 konfigurační parametr ENCRYPT_MAC 20-14
 konfigurační parametr ENCRYPT_MACFILE 20-14
 konfigurační parametr ENCRYPT_SWITCH 20-14
 konfigurační parametr EXT_DIRECTIVES 2-9
 konfigurační parametr FASTPOLL 5-25
 konfigurační parametr FILLFACTOR 2-3
 konfigurační parametr HETERO_COMMIT 2-8, 24-20
 konfigurační parametr IFX_EXTEND_ROLE 2-10
 konfigurační parametr LISTEN_TIMEOUT 1-10, 2-10
 konfigurační parametr LOCKS 2-5
 konfigurační parametr LOGBUFF 2-3, 14-15
 inteligentní velké objekty 7-32
 vyrovnávací paměti logického protokolu 7-12
 konfigurační parametr LOGFILES 2-3, 14-15
 nastavení velikosti logického protokolu 14-4
 úprava hodnoty ONCONFIG 14-15
 konfigurační parametr LOGSIZE 2-3, 14-15
 přidání souborů protokolu 14-11
 velikost logického protokolu 14-2
 změna 14-14
 zvětšení velikosti protokolu 14-13
 konfigurační parametr LTAPEBLK 2-4, 20-5
 konfigurační parametr LTAPEDEV
 konfigurační parametry
 LTAPEDEV 2-4
 konfigurační parametr LTAPESIZE 2-4, 20-5
 konfigurační parametr LTXEHWM 2-3, 24-10
 definice 14-15
 role v heuristickém odvolání transakce 24-10
 zabránění dlouhým transakcím 14-18
 konfigurační parametr LTXHWM 2-3
 definice 14-15
 role v heuristickém odvolání transakce 24-10
 zabránění dlouhým transakcím 14-17
 konfigurační parametr MAX_INCOMPLETE_CONNECTIONS 1-10, 2-10
 konfigurační parametr MAX_PDQPRIORITY 2-7
 konfigurační parametr MIRROR 2-2
 počáteční konfigurační hodnota 18-2

konfigurační parametr MIRROR (*pokračování*)
změna 18-2

konfigurační parametr MIRROROFFSET 2-2, 9-11
v případě potřeby 10-4

konfigurační parametr MIRRORPATH 2-2, 9-11

konfigurační parametr MSGPATH 1-24, 2-4

konfigurační parametr MULTIPROCESSOR 2-7, 5-13, 6-2

konfigurační parametr NETTYPE 2-7, 3-32, 3-35, 6-2
jednotkové procesy cyklického dotazování 5-22
role při určení protokolu 5-22
účel 3-32
vícenásobné síťové adresy 5-27
záznam třídy VP 5-23

konfigurační parametr NUMCPUVPS 5-13

konfigurační parametr OFF_RECVRY_THREADS 2-8, 19-10

konfigurační parametr ON_RECVRY_THREADS 2-8

konfigurační parametr ONDBSPACEDOWN 2-3

konfigurační parametr OPCACHEMAX 2-11, 10-39
konfigurace paměti 10-39

konfigurační parametr OPTCOMPIND 2-7

konfigurační parametr PC_HASHSIZE 2-6, 7-20, 8-4

konfigurační parametr PC_POOLSIZE 2-6, 7-20, 8-4

konfigurační parametr PHYSBUFF 2-4, 7-12

konfigurační parametr PHYSDBS 2-4

konfigurační parametr PHYSFILE 2-4

konfigurační parametr PLOG_OVERFLOW_PATH 15-7

konfigurační parametr RA_PAGES 2-6, 7-26

konfigurační parametr RA_THRESHOLD 2-6, 7-26

konfigurační parametr RESIDENT 2-6
během inicializace 4-6
nastavení pomocí obslužného programu onmode 8-7

konfigurační parametr ROOTNAME 2-2, 9-11

konfigurační parametr ROOTOFFSET 2-2, 9-11, 10-4

konfigurační parametr ROOTPATH 2-2, 9-11

konfigurační parametr ROOTSIZE 2-2

konfigurační parametr RTO_SERVER_RESTART 2-4, 15-3, 15-4, 15-6

konfigurační parametr SBSPACENAME 2-3, 9-14, 9-19, 10-24

konfigurační parametr SBSPACETEMP 2-3, 9-19, 10-26

konfigurační parametr SECURITY_LOCALCONNECTION 2-10

konfigurační parametr SERVERNUM 2-2
definovaný 7-7
použití 7-7

konfigurační parametr SHMADD 2-5, 7-14

konfigurační parametr SHMBASE
definovaný 2-5, 7-7
připojení prvního segmentu sdílené paměti 7-7

konfigurační parametr SHMTOTAL 2-5, 7-4, 7-5

konfigurační parametr SHMVIRTSIZE 2-5, 7-14

konfigurační parametr SINGLE_CPU_VP 2-7, 5-13, 6-2
jednoprocesorový počítač 5-14

konfigurační parametr SQLTRACE 26-14
režimy 26-12

konfigurační parametr STACKSIZE 2-6
definovaný 7-19
změna velikosti zásobníku 7-19

konfigurační parametr STAGEBLOB 2-11

konfigurační parametr STMT_CACHE 2-6, 8-6

konfigurační parametr STMT_CACHE_HITS 2-6, 8-6

konfigurační parametr STMT_CACHE_NOLIMIT 2-6, 8-6

konfigurační parametr STMT_CACHE_NUMPOOL 2-6, 8-6

konfigurační parametr STMT_CACHE_SIZE 2-6, 8-6

konfigurační parametr SYSSBSPACENAME 2-3

konfigurační parametr TAPEBLK 2-4, 20-5

konfigurační parametr TAPEDEV 2-4, 20-9, 21-6

konfigurační parametr TAPESIZE 2-4, 20-5

konfigurační parametr TBLTBLFIRST 2-2, 9-11, 10-10

konfigurační parametr TBLTBLNEXT 2-2, 3-30, 9-11, 10-10

konfigurační parametr TEMPTAB_NOLOG 2-3, 12-6

konfigurační parametr TXTIMEOUT 2-8, 24-13, 24-19
definice 24-19
onmode -Z 24-13
protokol dvoufázového potvrzování 24-18

konfigurační parametr UNSECURE_ONSTAT 2-10

konfigurační parametr USELASTCOMMITTED 7-14

konfigurační parametr USEOSTIME 2-8

konfigurační parametr VP_MEMORY_CACHE_KB 5-12

konfigurační parametr VPCLASS 2-7, 5-17, 6-2
konfigurace virtuálních procesorů CPU 5-13
uživatelé definované VP 5-16
virtuální procesory Java (JVP) 5-17

konfigurační parametry
ADMIN_MODE_USERS 4-8, 4-15
ADMIN_USER_MODE_WITH_DBSA 4-8, 4-9
ADTERR 2-10
ADTMODE 2-10
ADTPATH 2-11
ADTSIZE 2-11
AFF_NPROCS 6-2
AFF_SPROC 5-13, 6-2
ALARMPROGRAM 1-23, 2-9
ALRM_ALL_EVENTS 2-9
AUTO_CKPTS 15-5
AUTO_LRU_TUNING 16-5
BUFFERPOOL 2-5, 7-10, 10-13
CDR_QDATA_SBSpace 9-13
CKPTINTVL 2-5, 15-6
CLEANERS 2-6
CONSOLE 1-25, 2-4
DATASKIP 10-31
DB_LIBRARY_PATH 2-10
DBCREATE_PERMISSION 2-10
DBSERVERALIASES 2-2, 3-32
DBSERVERNAME 2-2, 3-17, 3-31
DBSPACETEMP 2-3, 9-34, 10-17
DD_HASHMAX 2-5
DD_HASHSIZE 2-5
DEADLOCK_TIMEOUT 2-8, 24-18
DEF_TABLE_LOCKMODE 2-5, 7-13
diagnostických informací 23-6
DIRECT_IO 10-17
DIRECTIVES 2-9
DRAUTO 2-8, 20-19, 20-20
DRIDXAUTO 2-8, 20-13
DRINTERVAL 2-8
DRLOSTFOUND 2-8, 19-9
DRTIMEOUT 2-8, 20-13
DS_HASHSIZE 7-19, 8-4
DS_MAX_QUERIES 2-7
DS_MAX_SCANS 2-7
DS_NONPDQ_QUERY_MEM 2-7, 7-15
DS_POOLSIZE 7-19, 8-4
DS_TOTAL_MEMORY 2-7, 7-14
DUMPCNT 2-9, 23-6
DUMPCORE 2-9, 23-6
DUMPDIR 2-9, 23-4, 23-6
DUMPGCORE 2-9, 23-4, 23-6
DUMPSHMEM 2-9, 23-4, 23-6
DYNAMIC_LOGS 14-11, 14-15, 14-16
ENCRYPT_CIPHERS 20-14
ENCRYPT_MAC 20-14
ENCRYPT_SWITCH 20-14
EXT_DIRECTIVES 2-9
EXTSHMADD 2-5, 7-14

konfigurační parametry *(pokračování)*

FILLFACTOR 2-3
HETERO_COMMIT 2-8, 24-20
IFX_EXTEND_ROLE 2-10
konfigurační parametr RTO_SERVER_RESTART 2-4, 15-3,
15-4, 15-6
LISTEN_TIMEOUT 1-10, 2-10
LOCKS 2-5
LOGBUFF 2-3, 7-12
LOGFILES 2-3
LOGSIZE 2-3
LTAPEBLK 2-4, 20-5
LTAPESIZE 2-4, 20-5
LTXEHWM 2-3, 14-15, 14-18, 24-10
LTXHWM 2-3, 14-15, 14-17, 24-10
MAX_INCOMPLETE_CONNECTIONS 1-10, 2-10
MIRROR 2-2, 18-2
MIRROROFFSET 2-2, 9-11, 10-4
MIRRORPATH 2-2, 9-11
MSGPATH 1-24, 2-4
MULTIPROCESSOR 2-7, 5-13
nastavení pomocí textového editoru 8-5
NETTYPE 2-7, 3-32
NUMCPUVPS 5-13
OFF_RECVRY_THREADS 2-8, 19-10
ON_RECVRY_THREADS 2-8
ONDBSPACEDOWN 2-3
OPCACHEMAX 2-11, 10-39
OPTCOMPIND 2-7
parametr AUTO_AIOVPS 2-7, 5-21
parametr DATASKIP 2-7
parametr DYNAMIC_LOGS 2-3
parametr ENCRYPT_HDR 20-14
parametr ENCRYPT_MAFILEC 20-14
parametr FASTPOLL 5-25
parametr MAX_PDQPRIORITY 2-7
parametr RA_PAGES 2-6, 7-26
parametr RA_THRESHOLD 2-6, 7-26
parametr SECURITY_LOCALCONNECTION 2-10
parametr TEMPTAB_NOLOG 2-3
parametr UNSECURE_ONSTAT 2-10
parametr VP_MEMORY_CACHE_KB 5-12
PC_HASHSIZE 2-6, 7-20, 8-4
PC_POOLSIZE 2-6, 7-20, 8-4
PHYSBUFF 2-4, 7-12
PHYSDBS 2-4
PHYSFILE 2-4
PLOG_OVERFLOW_PATH 15-7
počáteční blok kořenového prostoru dbspace 9-11
RESIDENT 2-6, 8-7
ROOTNAME 2-2, 9-11
ROOTOFFSET 2-2, 9-11, 10-4
ROOTPATH 2-2
ROOTSIZE 2-2
SBSPACENAME 2-3, 9-14, 9-19, 10-24
SBSPACETEMP 2-3, 9-19, 10-26
inteligentní velké objekty 10-24
sdílená paměť 8-2
SERVERNUM 2-2, 7-7
SHMADD 2-5, 7-14
SHMBASE 2-5, 7-7
SHMTOTAL 2-5, 7-4
SHMVIRTSIZE 2-5, 7-14
SINGLE_CPU_VP 2-7, 5-13
SQLTRACE 26-14
STACKSIZE 2-6, 7-19
STAGEBLOB 2-11

konfigurační parametry *(pokračování)*

STMT_CACHE 2-6, 8-6
STMT_CACHE_HITS 2-6, 8-6
STMT_CACHE_NOLIMIT 2-6, 8-6
STMT_CACHE_NUMPOOL 2-6, 8-6
STMT_CACHE_SIZE 2-6, 8-6
SYSSBSPACENAME 2-3
TAPEBLK 2-4, 20-5
TAPEDEV 2-4, 20-9, 21-6
TAPESIZE 2-4, 20-5
TBLTBLFIRST 2-2, 9-11, 10-10
TBLTBLNEXT 2-2, 3-30, 9-11, 10-10
TXTIMEOUT 2-8, 24-13, 24-19
USELASTCOMMITTED 7-14
USEOSTIME 2-8
VPCCLASS 2-7, 5-17, 6-2
konfigurační parametry EXTSHMADD 2-5, 7-14
konfigurační parametry Java 1-13
konfigurační soubor
neprovádění změn v souboru onconfig.std 1-11
onconfig.std 1-11
propojitelnost 3-9
síť 1-10
konfigurační soubor ONCONFIG
během inicializace 4-3, 4-4
během nastavení serveru 1-12
definovaný 2-11
chybějící parametry 1-11
parametry 1-11, 2-11
parametry jazyka Java 1-13
propojitelnost 1-11
připojení prostřednictvím sdílené paměti 3-38
příprava 1-11
úprava 1-13
vícenásobné uložení 3-41
zobrazení pomocí příkazu onstat -c 2-11
kontrola konzistence
data a režie 23-2
indexy 23-3
logické protokoly 23-3
monitorování konzistence dat 23-3
oblastí 23-3
ověření metadat 23-3
periodické úlohy 23-2
poškození dat 23-5
poškození indexu 23-5
prostory sbpace 9-19
přehled 1-19, 23-1
tabulky systémového katalogu 23-2
kontrolní bod
krok při inicializaci sdílené paměti 4-7
lehká připojení 10-29
maximální počet připojení 4-7
poslední dostupný protokol 13-6
replikace dat 20-11
spuštění 4-5
vynucený 20-11
vyprazdňování běžných vyrovnávací paměti 7-27
vyrovnávací paměť fyzického protokolu 7-27
vyrovnávací paměť logického protokolu 7-11
zápisy bloků 7-29
kontrolní body
automatické 15-5
automatické ladění 16-4
blokování 15-5
blokování transakcí 15-3, 15-5
definice 15-5

- kontrolní body *(pokračování)*
 - monitorování aktivity 16-3
 - neblokující 15-5
 - pokyny týkající se zálohování 15-6
 - role v rychlé obnově 15-8
 - ruční 15-5
 - rychlá obnova po 15-8
 - statistika 16-5
 - vynucení 16-4
 - vyprázdňování společné oblasti vyrovnávací paměti mezi 15-6
- konzistence dat
 - jak lze dosáhnout 15-1
 - monitorování 23-3
 - ověření 23-2
 - příznaky poškození 23-5
 - rychlá obnova 15-7
- konzole
 - zprávy 1-25
- koordinující databázový server 24-5
- kořenový prostor dbspace
 - definovaný 9-11
 - dočasné tabulky 9-25, 9-28
 - umístění souborů logického protokolu 13-2
 - výchozí umístění 9-11, 9-26
 - výpočet velikosti 9-31, 9-32
 - zrcadlení 18-4
- kritická část kódu
 - definice 15-2
- kritické prostory dbspace
 - paměť 9-9
 - zrcadlení 9-34, 9-37

L

- ladění LRU 16-5, 20-15
- ladění výkonu
 - logický protokol, nové umístění 16-1
 - objem protokolovaných dat 15-2
 - pravidla pro rozvržení disků 9-33
 - přesun fyzického protokolu 16-1
 - rozkládání dat na více disků 9-39
 - správci logických disků 9-39
 - ukázkové rozvržení disků optimalizované na výkon 9-36
 - velikost logického protokolu 13-4, 16-1
 - zápis front LRU 7-29
 - zápisy na popředí 7-28
- lehká připojení
 - fyzické protokolování 10-29
 - tabulky typu RAW 9-24
 - tabulky typu STANDARD 9-24
- licencování uživatelé, maximální povolený počet 4-3, 4-7
- logické obnovení, replikace dat 19-7
- logické paměťové jednotky
 - seznam 9-2
- logický protokol
 - definovaný 7-11, 11-1, 13-1
 - dynamické přidělování 1-20
 - globální transakce 24-4, 24-14
 - konfigurační parametry 14-14
 - monitorování za účelem zjištění zaplnění pomoci obslužného programu onstat 14-8
 - obslužný program onlog 14-16
 - ověření 23-3
 - pokyny týkající se velikosti 13-3, 14-2
 - pokyny týkající se výkonu 13-4
 - pozice v protokolu 24-14
 - protokolování inteligentních velkých objektů 14-4

- logický protokol *(pokračování)*
 - typy záznamů 7-11
 - úlohy administrace 1-20, 13-7
 - záloha
 - definovaná 1-19
 - harmonogram 1-19
 - kontrolní body 15-6
 - záznam
 - procesy databázového serveru vyžadující 11-2
 - protokol dvoufázového potvrzování 24-6, 24-14, 24-15
 - příkazy jazyka SQL, které generují 11-4
- LVM.
 - Viz správce logických disků.

M

- mapování chybného sektoru, absence 23-9
- mapování chybných sektorů 23-9
- MaxConnect
 - agregace paketů 3-43
 - definovaný 3-42
 - imc jako část pole určující protokol 3-18
 - instalace 3-42
 - monitorování 3-43
 - protokol onsocimc 3-19
 - protokol ontlimc 3-19
 - příkaz imcadmin 3-43
- maximum
 - počet bloků 10-8
 - počet paměťových prostorů 10-8
 - uživatelských připojení 4-3, 4-7
 - velikost bloku 10-8
- mechanizmy obnovy dat
 - inteligentní velké objekty 9-13
 - rychlá obnova 15-7
- metadata
 - bloky 10-25
 - definovaná 7-32, 9-13
 - dočasný prostor sbspace 9-19
 - ověření 23-3
 - protokolování 15-3
 - protokolování prostoru sbspace 13-7
 - přesunutí prostoru z rezervované oblasti 10-45
 - přidělení 10-24
 - tabulka prostorů dbspace 7-16
 - určení správné velikosti 9-16, 10-25
 - výpočet oblasti 9-19, 10-25
 - vypuštění bloků prostoru sbspace 10-28
 - vytvoření 9-16
- metody RTO_SERVER_RESTART 16-5
- mezipaměť
 - distribuce dat 7-19, 8-4
 - mezipaměť pro příkazy SQL
 - konfigurační parametry 2-6
 - povolení 8-6
 - určení velikosti 8-6
 - mezipaměť rutin SPL
 - velikost hashovací tabulky 2-6, 8-4
 - velikost společné oblasti 8-4
 - monitorování vyrovnávací paměti ve sdílené paměti 8-8, 8-10
 - optického podsystému 10-39
- mezipaměť distribuce dat 7-19, 8-4
- mezipaměť optického podsystému
 - ID klientského uživatele 3-22, 10-39
 - ID uživatelské relace 10-39
 - kilobajty zapsaných dat typu TEXT a BYTE 10-39
 - počet zapisovaných objektů 2-11, 10-39

- mezipaměť optického podsystemu *(pokračování)*
 - přídělení 10-39
 - velikost 10-39
 - mezipaměť pro příkazy SQL
 - definovaná 7-19
 - konfigurace 8-6
 - monitorování 8-6
 - přehled 2-6
 - umístění sdílené paměti 7-14
 - určení velikosti 2-6
 - mezipaměť rutin UDR 7-20
 - mezipaměť slovníku 7-19
 - meziprocesová komunikace
 - sdílená paměť 7-2
 - v poli nettype 3-18
 - Microsoft Transaction Server 24-2
 - místní zpětná smyčka
 - omezení 3-8
 - příklad 3-38
 - připojení 3-8, 5-22
 - modul pro
 - sdílená paměť
 - definovaný 7-20
 - velikost 7-20
 - způsob připojení klientu 7-6
 - modul pro podporu komunikace
 - konfigurační soubor 3-9
 - pole voleb informací sqlhosts 3-23
 - zabezpečení sítě 3-9
 - modul pro podporu komunikačních prostředků
 - virtuální procesor 5-27
 - moduly DataBlade
 - virtuální procesory 5-13
 - monitorování
 - aktivita uživatele 24-14
 - databázového serveru 1-23
 - globální transakce 24-13
 - licencovaných uživatelů 4-3, 4-7
 - mezipaměť pro příkazy SQL 8-6
 - oblasti 10-38
 - oblastí metadat a uživatelských dat 10-45
 - prostory sbpace 9-19, 10-42
 - prostory tblspace 10-38
 - uživatelských připojení 4-3, 4-7
 - využití programu MaxConnect 3-43
 - zámky 24-14
 - zámky spin 8-8
 - monitorování databázového serveru 1-24
 - aktivita společné oblasti vyrovnávací paměti 8-10
 - bloky 10-34
 - databáze 10-34, 12-6
 - délka front diskového vstupu - výstupu 5-21
 - fronty diskového vstupu - výstupu 5-21
 - globální transakce 1-23
 - hodnoty konfiguračních parametrů 2-11
 - jednoduchý velké objekty v prostorech dbspace 10-39, 10-40
 - kontrolní body 16-3
 - oblasti 10-38
 - paměť prostoru blobspace 10-23
 - pomocí programu ON-Monitor 1-24
 - pomocí programu oncheck 1-24
 - použití programu onstat 1-25
 - používání tabulek SMI 1-25
 - profil aktivity 8-8
 - sdílená paměť 8-7
 - soubor fyzického protokolu 16-2
 - soubory protokolu 14-7
 - monitorování databázového serveru *(pokračování)*
 - stav protokolování 12-6
 - stav replikace dat 20-17
 - uživatelské jednotkové procesy 7-18
 - virtuální procesory 6-5
 - vyrovnávací paměť fyzického protokolu 7-12, 16-2
 - vyrovnávací paměti 8-8
 - vyrovnávací paměti logického protokolu 16-2
 - využití disku fragmenty 10-35, 10-38
 - zámky latch 8-7, 8-8
 - MTS/XA 24-2
 - multiplexní připojení 3-4
 - mutex
 - definované 5-12
 - definovaný 7-21
 - použití 7-21
 - synchronizace 5-12
- ## N
- nastavení
 - obslužný program ON-Bar 1-17
 - obslužný program ontape 1-17
 - proměnné prostředí 1-9
 - nástroj OpenAdmin pro server IDS 1-27, 26-1
 - nástroje k monitorování
 - sledování výkonu v systému Windows 1-26
 - UNIX 1-25
 - název
 - paměťové prostory 10-7
 - název hostitele
 - změna 1-27
 - názvy služeb 3-16
 - neaktualizovaná vyrovnávací paměť, definovaná 7-26
 - nejstarší aktualizace, uvolnění souboru logického protokolu 13-3
 - několik karet síťového rozhraní 5-27
 - několik souběžných jednotkových procesů 5-8
 - nekonistence, rozpoznání 23-2
 - nesprávně pracující uživatelská rutina 5-16
 - New Technology File System 1-6
 - ntchname.exe 1-27
- ## O
- oblast uživatelských dat, prostor sbpace 7-32
 - oblasti
 - definované 9-8
 - monitorování 10-38
 - monitorování pomocí tabulky sysextents 10-38
 - ověření 23-3
 - prostory sbpace 9-7, 9-15
 - přídělení 9-8
 - související klíčové koncepty 9-8
 - struktura 9-8, 9-15
 - účel 9-8
 - velikost další oblasti 9-8, 9-16, 10-10
 - velikost pro prostor tblspace 10-10
 - velikost v prostorech sbpace 9-7, 9-15
 - velikost, počáteční 9-8, 9-16, 10-10
 - vztah k bloku 9-8
 - způsob přidělování databázovým serverem 9-8
 - obnova
 - paralelní zpracování 5-6
 - po selhání média 17-2
 - protokol dvoufázového potvrzování 24-5

- obnova (*pokračování*)
 - režim
 - definovaný 4-8
 - tabulky typu RAW 9-24
 - tabulky typu STANDARD 9-24
- obnovení
 - tabulky typu RAW 9-24
 - tabulky typu STANDARD 9-24
 - typy tabulek 9-26
- obnovení k určitému datu
 - protokolování 9-24
- obsah svazku 10-4
- obslužné programy
 - cron 1-26
 - chkenv 1-9
 - iostat 1-26
 - ISA 10-34
 - obslužný program onstat
 - volba -d 10-43
 - připojení ke sdílené paměti 7-5
 - ps 1-26
 - sar 1-26
 - UNIX 1-25
 - vmstat 1-26
- obslužný program archecker
 - přehled 1-19
- obslužný program convert systému Windows 1-6
- obslužný program dbload 10-46
- obslužný program chkenv 1-9
- obslužný program ixsu.exe 1-27
- obslužný program ON-Bar
 - nastavení 1-17
- obslužný program ON-Monitor
 - kopírování konfigurace 2-11
 - monitorování databázového serveru 1-24
 - nastavení parametrů
 - sdílená paměť 8-5
 - virtuální procesory 6-3
 - obnovení bloku 18-7
 - přidání
 - blok 10-19
 - soubor logického protokolu 14-12
 - zrcadlené bloky 18-6
 - spuštění zrcadlení 18-5
 - vypnutí bloku 18-6
 - vypouštění paměťových prostorů 10-29, 10-30
 - vypuštění souboru logického protokolu 14-13
 - vytváření prostorů blobspace 10-22
 - vytvoření prostorů dbspace 10-10
 - změna režimů databázového serveru 4-10
- obslužný program ondblog
 - databáze kompatibilní se standardem ANSI 12-5
 - změna režimu protokolování
 - ISA 12-3
 - obslužný program ondblog 12-2
- obslužný program oncheck
 - informace o stránkách blobpage 10-40
 - kontrola konzistence 23-3
 - monitorování oblastí metadat a uživatelských dat 10-44
 - monitorování prostorů sbpace 10-24, 10-42, 10-43
 - ověření
 - datové stránky 23-2
 - indexy 23-3
 - metadata 23-2, 23-3
 - oblasti 23-2, 23-3
 - protokoly a rezervované stránky 23-2
 - rezervované stránky 23-2
- obslužný program oncheck (*pokračování*)
 - ověření (*pokračování*)
 - tabulky systémového katalogu 23-2
 - volba -cc 23-2
 - volba -cD 23-3
 - volba -ce 23-2
 - volba -cI 23-2, 23-3
 - volba -cr 23-2
 - volba -cR 23-2
 - volba -cs 10-42, 10-44, 23-3
 - volba -pB 10-40
 - volba -pe 10-36, 10-42, 10-43
 - volba -pr 2-11, 14-8
 - volba -ps 10-44, 23-2
 - volba -pS 10-44
 - získání informací
 - bloky 10-36
 - konfigurace 10-40
 - logické protokoly 14-8
 - oblasti a fragmentace 10-38
 - prostory blobspace 10-36, 10-43
 - prostory tbspace 10-38
- obslužný program oninit
 - dočasné tabulky 9-28
 - spuštění databázového serveru 4-1, 4-2
 - volba -p 4-6, 9-20
- obslužný program onlog
 - rekonstrukce globální transakce 25-3
 - zobrazení záznamů protokolu 14-16
- obslužný program onmode
 - konfigurace mezipaměti příkazů jazyka SQL 8-6
 - nastavení
 - typ databázového serveru 20-15
 - použití v programu ISA 20-15
 - přepínání souborů logického protokolu 14-5
 - přidání segmentu 8-7
 - ukončení
 - relace 25-1
 - transakce 24-8, 24-13
 - účastnický jednotkový proces 24-10
 - uvolnění souboru logického protokolu 14-7
 - uživatelský jednotkový proces obsluhující požadavky 5-3
 - volba -a 8-7
 - volba -c 8-7
 - volba -d idxauto 20-13
 - volba -e 8-6
 - volba -O 23-7
 - volba -W 8-6
 - vynucení kontrolního bodu 10-29, 16-4, 20-11
 - vypuštění virtuálních procesorů CPU 6-4
 - vzdálené provedení příkazů 26-11
 - změna uložení sdílené paměti 8-6
- obslužný program onmode -d index 20-13
- obslužný program onparams
 - přidání souboru logického protokolu 14-11
 - vypuštění souboru logického protokolu 14-12
 - změna fyzického protokolu
 - umístění 16-2
 - velikost 16-2
- obslužný program onperf 1-24
- obslužný program onpladm 10-46
- obslužný program onsocimc 3-19
- obslužný program onspaces
 - definovaný 1-16
 - obnovení vypnutého bloku 18-6
 - přidávání bloků prostoru sbpace 10-25
 - přidávání zrcadlených bloků 10-25

- obslužný program onspaces *(pokračování)*
 - ukončení zrcadlení 18-7
 - volba -a 10-25, 18-6
 - volba -c -b 10-22
 - volba -c -d 10-9
 - volba -c -S 10-24
 - volba -c -t 10-17
 - volba -c -x 10-31
 - volba -cl 10-29
 - volba -d 10-24, 10-29
 - volba -f 10-32
 - volba -g 10-22
 - volba -ch 9-15, 9-18, 10-26, 20-11
 - volba -k 10-16
 - volba -s 9-14
 - volba -t 9-34
 - volba -U 10-26
 - volby -Df 9-15, 10-28
 - vypnutí bloku 18-6
 - vypuštění bloků prostoru sbspace 10-28
 - vytváření prostorů sbspace 10-24
 - vytvoření dočasného prostoru sbspace 9-19
 - změna parametru DATASKIP 10-32
 - změna stavu bloku 20-11
- obslužný program onstat
 - aktualizace statistických údajů o stránkách blobpage 10-35
 - monitorování
 - blobspace 10-35
 - fyzický protokol 16-3
 - konfigurace 2-11
 - mezipaměť pro příkazy SQL 8-6
 - profil databázového serveru 8-8
 - příkazy jazyka SQL 12-6
 - replikace dat 20-17
 - sdílená paměť 8-7
 - soubory logického protokolu 14-12, 16-3
 - společná oblast vyrovnávací paměti 8-11
 - stav bloku 10-34
 - transakce 6-5
 - virtuální procesory 6-5
 - vyrovnávací paměť logického protokolu 14-8, 16-2
 - využití vyrovnávací paměti 8-9, 8-11
 - zámky latch 8-7
 - zatížení fragmentů 10-35
 - profilování uživatelské aktivity 24-13
 - přehled 1-25
 - příznaky dočasného prostoru sbspace 10-27
 - virtuální procesory CPU 5-3
 - volba -c 2-11
 - volba -d 10-35, 10-42
 - volba -d update 1-24, 10-35
 - volba -g ath 6-5
 - volba -g cac 8-6
 - volba -g glo 6-5
 - volba -g imc 3-43
 - volba -g ioq 6-6
 - volba -g rea 6-6
 - volba -g smb c 10-42, 10-45
 - volba -g sql 12-6
 - volba -g stm 12-6
 - volba -k 24-14
 - volba -l 14-12
 - volba -m 1-24
 - volba -p 8-8
 - volba -s 8-8
 - volba -u 5-3, 24-14
 - volba -x 12-6, 24-13
- obslužný program onstat *(pokračování)*
 - volby -g ssc 8-6
 - zaznamenání
 - globální transakce 24-14
 - zámky 24-14
 - zobrazení
 - zprávy 1-24
- obslužný program ontape
 - alternativní metoda zálohování 20-9, 21-6
 - nastavení 1-17
 - zálohování souborů logického protokolu 13-6
 - změna stavu protokolování databáze 12-4
- obslužný program Setnet32 1-10
- oddíly
 - uvedené v příkazu ALTER FRAGMENT 10-20
 - vytváření a používání 10-20
 - vytváření v již vytvořené tabulce nebo indexu 10-21
 - vytváření ve fragmentované tabulce nebo indexu 10-20
- odkládání paměti na disk 7-9
- odlehčená verze vstupu - výstupu 7-11, 9-16
- odlehčené procesy 5-2
- odlehčené prohledávání 7-18
- odložená kontrola omezení 11-2
- odolnost vůči selhání
 - replikace dat 19-4
 - rychlá obnova 15-7
- odstranění
 - data inteligentních velkých objektů 10-29, 10-30
 - soubory protokolu 14-12
 - tabulky typu RAW 9-24
 - tabulky typu STANDARD 9-24
- odvolání transakce
 - heuristické, monitorování 24-13
 - inteligentní velké objekty 9-21
 - protokolované příkazy jazyka SQL 11-4
 - tabulky typu RAW 9-24
 - tabulky typu STANDARD 9-20
- omezení
 - kontrola, odložená 11-2
 - počet bloků 10-8
 - uživatelé definované VP 5-15
 - velikost bloku 10-8
 - virtuální procesory Java (JVP) 5-17
 - VP CPU 5-12
- operační systém
 - 32bitové a 64bitové verze 8-2
 - nástroje 1-25
 - parametry 8-2
- operační systém UNIX
 - nastavení proměnných prostředí 1-8
 - příkaz link 1-6, 10-5
 - spouštěcí skript 1-15
 - ukončovací skript 1-15
 - změna jádra 1-4
- oprávnění
 - nastavení 1-6
- optický virtuální procesor 5-28
- optimalizace předpokládaného konce 24-8
- ověření
 - datové stránky 23-3
 - indexy 23-3
 - logické protokoly 23-3
 - metadata 23-3
 - oblasti 23-3
 - rezervované stránky 23-3
 - tabulky systémového katalogu 23-2

- ověřování
 - definované 3-8
 - výchozí metoda 3-9
- ovladač IBM Informix ODBC 24-2
- ovladač JDBC 1-7
- ovladač ODBC 24-2
- ovladač síťového protokolu
 - definovaný 3-2

P

- paměť
 - 64bitové platformy 7-33
 - přidání segmentu 8-7
- paměť uložených procedur
 - Viz mezipaměť UFR.
- paměťová zařízení
 - nastavení 1-17
- paměťové charakteristiky
 - hierarchie 9-17
 - inteligentní velké objekty 9-16, 10-26
 - paměťové prostory 1-17
 - prostory sbspace 9-15
 - příkaz onspaces -Df 10-24
 - příkaz onspaces -ch 9-15, 10-26
- paměťové jednotky 9-2
- paměťové prostory
 - harmonogram zálohování 1-17
 - spuštění zrcadlení 18-4
 - ukončení zrcadlení 18-7
 - uložení oddílů NTFS 1-6
- paměťový prostor v logickém disku nebo jednotce
 - definovaný 1-16
- paralelní databázové dotazy
 - dotaz pro podporu rozhodování
 - Viz konfigurační parametr DS_MAX_QUERIES.
 - konfigurační parametr DS_MAX_QUERIES 2-7
 - konfigurační parametr DS_NONPDQ_QUERY_MEM 2-7, 7-15
 - konfigurační parametr DS_TOTAL_MEMORY 2-7, 7-15
 - konfigurační parametr MAX_PDQPRIORITY 2-7
- paralelní zpracování
 - virtuální procesory 5-6
- parametr AVG_LO_SIZE 9-16, 10-25
- parametr BUFFERING volby příkazu onspaces -c -Df 9-16
- parametr LOGGING příkazu onspaces 9-17
- parametry jádra
 - změna 1-4
- plánovač 14-3
 - přehled 26-1, 26-5
- plánování
 - priorita 1-13
- plánování podle priority 1-12
- plánování zdrojů 1-3
- plány dotazů 7-19
- plně kvalifikovaný název domény (FQDN) 3-12
- počáteční konfigurace
 - rozvržení disků 9-33
 - směrnice pro kořenový prostor dbspace 9-11
 - vytvoření paměťových prostorů 1-16
- podpora 32bitových systémů 1-4
- pokyny týkající se velikosti
 - logický protokol 13-3, 14-2, 14-11
- pole beginlg 24-14
- pole curlg 24-14
- pole lchwaits 8-8
- pole logposit 24-14

- pole názvu hostitele
 - definované 3-19
 - několik karet síťového rozhraní 5-27
 - použití adresy IP 3-27
 - pravidla syntaxe 3-19
 - protokol IPX/SPX 3-20
 - sdílená paměť 3-20, 3-21
 - určení adresy pomocí zástupných znaků 3-28
- pole názvu služby v informacích sqlhosts
 - definované 3-20
 - pravidla syntaxe 3-19
 - protokol IPX/SPX 3-21
 - proudová propojení procesů 3-21
 - sdílená paměť 3-21
- pole nettype
 - formát 3-17
 - použití typu rozhraní 3-39
 - přehled hodnot 3-19
 - syntaxe 3-17
- pole RAID 17-3
- pole voleb
 - pravidla syntaxe 3-21
 - přehled 3-21
 - přesměrování připojení 3-23
 - volba CSM 3-23
 - volba identifikátoru 3-26
 - volba keep-alive 3-26
 - volba skupiny 3-26
 - volba velikosti vyrovnávací paměti 3-22
 - volba zabezpečení 3-26
- porucha média
 - obnova po 17-2
 - zjišťování 17-5
- posun
 - rozdělování oddílů 10-4
 - účel 9-5
 - v případě potřeby 10-4
 - zabránění přepsání informací o oddílech 9-5, 10-4
- poškozené tabulky 9-23
- poškození, řešení chyb vstupu - výstupu 23-5
- potvrzování
 - heterogenní 24-19, 24-21
- použití programu onperf 1-24
- pozice v protokolu 24-14
- požadavky
 - konfigurace 1-11
- prázdný soubor
 - vytvoření 10-6
- primární blok 9-3
- primární databázový server 19-3
- priorita
 - diskový vstup - výstup 5-18
 - stárnutí, prevence 5-14
- probuzení jednotkových procesů 5-11
- procedura sysdbclose() 1-17, 1-18
- procedura sysdbopen() 1-17, 1-18
- procesorová afinita
 - definované 5-14
 - použití 5-14
- procesy
 - DSA a architektura dvou procesů 5-7
 - porovnání s jednotkovými procesy 5-6
 - připojení ke sdílené paměti 7-5
- profil
 - statistika 8-8
- program Windows Instance Manager 1-12, 4-2

programy SPL
 konfigurační parametr PC_HASHSIZE 2-6
 konfigurační parametr PC_POOLSIZE 2-6
 mezipaměť
 Viz mezipaměť rutin UDR.
 určení velikosti společné oblasti 7-20
 velikost hashovací tabulky mezipaměti rutin UDR 2-6, 7-20, 8-4

prohledávání
 indexy 7-26
 paralelní zpracování 5-6
 sekvenční prohledávání tabulek 7-26

prohlížeč událostí, Windows NT 1-26

procházení dotazů
 přehled 26-1, 26-11

proměnná prostředí C8BITLEVEL 1-8
 proměnná prostředí CLASSPATH 1-7, 1-9
 proměnná prostředí CLIENT_LOCALE 1-8
 proměnná prostředí DB_LOCALE 1-8
 proměnná prostředí DBLANG 1-8
 proměnná prostředí DBPATH
 použití při automatickém přesměrování 19-24
 skupina databázových serverů 3-25

proměnná prostředí DBSPACETEMP 9-26, 10-17
 proměnná prostředí ESQLMF 1-8
 proměnná prostředí GL_DATE 1-8
 proměnná prostředí GL_DATETIME 1-8
 proměnná prostředí GLS8BITFSYS 1-8
 proměnná prostředí IFX_NODBPROC 1-18
 proměnná prostředí IFX_SESSION_MUX 3-5
 proměnná prostředí INF_ROLE_SEP 4-9

proměnná prostředí INFORMIXDIR
 definovaná 1-7
 spouštěcí skript 1-15
 ukončovací skript 1-15

proměnná prostředí INFORMIXSERVER
 definovaná 1-7
 klientské aplikace 3-25
 použit při přeměrování klientů 19-28

proměnná prostředí INFORMIXSHMBASE 7-6
 proměnná prostředí INFORMIXSQLHOSTS 1-10, 3-13, 3-15
 proměnná prostředí INFORMIXSTACKSIZE 7-19
 proměnná prostředí JVPHOME 1-7, 1-9
 proměnná prostředí LD_LIBRARY_PATH 1-9

proměnná prostředí ONCONFIG
 definovaná 1-7
 nastavení 1-11
 vícenásobné databázové servery 1-15

proměnná prostředí PATH 1-8, 1-15
 spouštěcí skript 1-7, 1-15
 ukončovací skript 1-15

proměnná prostředí SERVER_LOCALE 1-8
 proměnná prostředí TERM 1-8
 proměnná prostředí TERMCAP 1-8
 proměnná prostředí TERMINFO 1-8

proměnné prostředí
 CLASSPATH 1-7, 1-9
 CLIENT_LOCALE 1-8
 DB_LOCALE 1-8
 DBSPACETEMP 10-17
 IFX_SESSION_MUX 3-5
 INFORMIXDIR 1-7, 1-15
 INFORMIXSERVER 1-7
 INFORMIXSHMBASE 7-6
 INFORMIXSQLHOSTS 1-10, 3-13
 JVPHOME 1-7, 1-9
 klientské aplikace 1-7
 konfigurační soubor prostředí 1-8

proměnné prostředí (*pokračování*)
 LD_LIBRARY_PATH 1-9
 nastavení 1-7
 NODBPROC 1-18
 ONCONFIG 1-7, 2-11
 ovlivňující
 multiplexní připojení 3-5
 PATH 1-8, 1-15
 požadované proměnné 1-7
 příklad souboru nastavení 1-9
 SERVER_LOCALE 1-8
 soubor .profile nebo .login 1-8
 soubor dbservername.cmd 1-9
 soubory informix.rc a .informix 1-8
 TERM 1-8
 TERMCAP 1-8
 TERMINFO 1-8

propojení
 vytvoření 10-5

propojitelnost
 ASF 3-4
 konfigurace 1-9
 konfigurační parametry 3-31
 soubor hosts 3-10
 soubor services 3-10
 soubor sqlhosts 1-10
 soubory, definované 3-9
 Windows 1-10

propojovací soubory sítě TCP/IP 3-9

prostor extspace
 vypuštění pomocí obslužného programu onspaces 10-31
 vytvoření 10-30

prostor tblspace
 definovaný 9-28
 dočasný, během inicializace 4-6
 monitorování
 tabulka systabnames 10-38
 obsah tabulky 7-17
 typy obsažených stránek 9-29
 velikost, oblast, pro prostor tblspace tblspace 10-10
 vypustit dočasně 4-6

prostor v předpřipraveném souboru
 definovaný 1-6, 9-4
 porovnání s prostorem s přímým přístupem 9-4
 pro statická data 9-4
 přidělení 10-4
 výkon využívající přímý vstup - výstup 10-17

prostorů dbspace
 definice kořenového prostoru dbspace 9-11
 definovaný 9-9
 dočasně 9-11
 inteligentní velké objekty 9-7
 kořenový 9-11
 monitorování inteligentních velkých objektů 10-40
 názvy 10-7
 omezení, přesun protokolů 14-14
 omezení, přidání protokolů 14-11
 přejmenování 10-19
 přidání
 blok 10-18
 pomocí programu ISA 10-9
 zrcadlený blok 18-5
 souvislost mezi logickými a fyzickými pamětovými
 jednotkami 9-9
 tabulka
 definovaná 7-16
 metadata 7-16

- prostorů dbspace *(pokračování)*
 - tabulka *(pokračování)*
 - vypouštění 10-29
 - tabulka ve sdílené paměti 7-16
 - účel 9-9
 - určení velikosti stránky při vytvoření 10-12
 - velikost stránky, určení 10-12
 - velikosti stránek jiné než výchozí v prostředí replikace HDR 20-4
 - vypouštění
 - blok 10-27
 - obslužný program ON-Monitor 10-30
 - onspaces 10-30
 - přehled 10-29
 - vytvoření 10-9, 10-12
 - dočasně 10-17
 - počátečního prostoru dbspace 9-11
 - pomocí obslužného programu onspaces 10-9
 - vytvoření počátečního prostoru dbspace 1-16
 - zálohování 10-8, 10-9
 - zrcadlení, pokud jsou obsaženy soubory logického protokolu 17-4
- prostory blobspace
 - aktivace 13-7
 - definovaný 9-12
 - názvy 10-21
 - omezení
 - přidání protokolů 14-11
 - přidání pomoci
 - onspaces 10-22
 - statistické údaje o paměti 10-23
 - stránka s mapou volných stránek
 - zaznamenávání stránek blobpage 7-32
 - úlohy protokolování 9-12, 13-7
 - vypouštění pomocí
 - obslužný program ON-Monitor 10-30
 - onspaces 10-30
 - počáteční úlohy 10-29
 - vyrovňovací paměti 7-31
 - vytvoření pomocí
 - onspaces 10-22
 - zálohování 10-8, 10-22
 - zápis dat 7-30
 - zápis dat typu TEXT a BYTE 9-12
 - získání počtu volných stránek 10-35
- prostory sbspace
 - čas posledního přístupu 9-16
 - definovaná 9-12
 - definované 9-13
 - definování replikačního serveru 9-13
 - diskové struktury 9-18
 - dočasně
 - definované 9-19
 - přidávání bloků 9-20, 10-25
 - příklad 10-26
 - příznak LO_CREATE_TEMP 10-27
 - vypouštění bloků 9-20, 10-28
 - vypouštění prostoru sbpace 10-29
 - zálohování a obnovení 9-20
 - dočasný
 - vytvoření 10-26
 - Enterprise Replication 9-13
 - charakteristiky 9-20
 - kontrola konzistence 9-20
 - metadata 7-32, 9-13, 9-16
 - názvy 10-7
 - oblast uživatelských dat 7-32, 9-13, 10-41
- prostory sbspace *(pokračování)*
 - odlehčená verze vstupu - výstupu 7-11
 - odlehčené verze vstupu - výstupu 9-16
 - omezení
 - přidání protokolů 14-11
 - použití příkazu onstat -g smb c 10-42, 10-45
 - protokolování 9-17, 13-7
 - přesunutí prostoru z rezervované oblasti 10-45
 - přidávání bloků 9-18, 10-25
 - přidělení prostoru 9-18
 - přírůstkové zálohy 14-5
 - rezervovaná oblast 9-13
 - režim ukládání do vyrovnávací paměti 9-16
 - rychlá obnova 9-13
 - schopnost obnovy 9-13
 - soubory JAR 1-13
 - statistické údaje pro uživatelem definovaná data 2-3
 - stránky sbpace 9-7
 - ukládání inteligentních velkých objektů 7-32, 9-12
 - určení paměťových charakteristik 9-15, 10-24
 - určení velikosti oblastí metadat 10-25
 - výpočet metadat 9-19
 - vypouštění
 - pomocí obslužného programu onspaces 10-29
 - sloupců typů CLOB a BLOB 10-29
 - vytvoření
 - pomocí obslužného programu onspaces 10-24
 - pomocí programu ISA 10-25
 - zálohování 9-19, 10-24, 10-26
 - zamykání 9-17
 - změna paměťových charakteristik 10-26
 - zrcadlení 10-24
- prostředí
 - konfigurační soubor 1-8
 - řídící aplikace 1-9
- prostředí X/Open
 - prostředí DTP 7-17
- protokol
 - určení 5-22
- protokol dvoufázového potvrzování
 - aktivita účastníka 24-6
 - automatická obnova
 - obnova koordinátora 24-7
 - automatické zotavení 24-7
 - role administrátora 24-7
 - zotavení účastníka 24-7
 - DEADLOCK_TIMEOUT 24-18
 - definice 24-5
 - distribuované dotazy 1-23
 - fáze po rozhodnutí 24-6
 - fáze před potvrzením 24-7
 - heuristické rozhodnutí
 - heuristické odvolání transakce 24-9
 - heuristické ukončení transakce 24-12
 - typy 24-8
 - chybové stránky 24-14
 - identifikátor globální transakce 25-3
 - kdy se používá 24-5
 - konfigurační parametr TXTIMEOUT 24-19
 - konfigurační parametry 24-18
 - nezávislá akce 24-8
 - definice 24-8
 - chyby 24-9
 - výsledky 24-8
 - vyvolávající 24-8
 - obnova koordinátora 24-5
 - optimalizace předpokládaného konce 24-7, 24-8

- protokol dvoufázového potvrzování (*pokračování*)
 - předčasné ukončení globální transakce 25-1
 - přehled 1-23
 - role aktuálního serveru 24-6
 - rozdíl proti heterogennímu potvrzování 24-18
 - vyprázdnění záznamů protokolu 24-15
 - záznamy logického protokolu 24-10
 - zotavení účastníka 24-7
 - zprávy 24-6
- protokol Dynamic Host Configuration Protocol 3-10
- protokol Internet Protocol verze 6 3-30
- protokol IPX/SPX
 - položka sqlhosts 3-40
 - služba, definovaná 3-21
 - v poli servicename 3-21
 - v poli názvu hostitele 3-40
- protokol ontlime 3-19
- protokol potvrzování
 - dvoufázový 24-2, 24-5
 - heterogenní 24-2
- protokol zpráv
 - během inicializace 4-6
 - definovaný 1-23
 - dynamicky přidáné protokoly 14-16
 - fyzická obnova 15-2
 - poškození dat 23-5
 - využití metadat 10-45
 - zobrazení zpráv 1-23
- protokolování
 - aktivita, která se vždy protokoluje 11-3
 - data typu TEXT a BYTE 13-6
 - databáze kompatibilní se standardem ANSI 12-3
 - definované protokolování transakcí 11-3
 - Enterprise Replication 9-13, 11-2
 - fyzické protokolování
 - definice 15-2
 - pokyny týkající se velikost 15-3
 - potlačení v dočasných prostorech dbspace 9-11
 - inteligentní velké objekty 7-32, 9-21, 13-9
 - kdy použít tabulky protokolování 13-8
 - metadat a uživatelských dat 9-20
 - obnovení k bodu v čase 9-24
 - potlačení pro implicitní tabulky 9-11
 - používání protokolování transakcí 11-5
 - proces
 - data prostoru blobspace 13-6, 13-7
 - data prostoru dbspace 13-10
 - procesy databázového serveru vyžadující 11-2
 - prostory sbpace 9-17, 13-7, 13-8
 - prostředí DTP 11-7
 - protokolování transakcí s vyrovnávací pamětí 11-6
 - role v replikaci dat 19-7
 - tabulky
 - souhrn 9-24
 - tabulky typu RAW 9-24
 - tabulky typu STANDARD 9-24
 - typy tabulek 12-5
 - vliv použití vyrovnávací paměti na rychlost zaplnění logického protokolu 13-4
 - zakázání v dočasných tabulkách 12-6
 - změna tabulky za účelem vypnutí 12-5
 - změna tabulky za účelem zapnutí 12-6
 - zobrazení stavu pomocí programu ISA 12-7
- protokolování bez vyrovnávací paměti
 - vyprázdnění vyrovnávací paměti logického protokolu 7-30
- protokolování databáze
 - prostředí DTP 11-7
 - protokolování databáze (*pokračování*)
 - replikace Enterprise Replication 11-2
 - zálohy 10-8
 - protokolování transakcí
 - bez vyrovnávací paměti 11-5
 - definované 11-3, 11-5
 - Enterprise Replication 9-13
 - kdy použít 11-5
 - s vyrovnávací pamětí 11-5
 - protokolování transakcí s vyrovnávací pamětí
 - kdy jsou vyprázdněny 11-6
 - proudové propojení procesů
 - na platformách Linux 3-8
 - na platformách UNIX 3-7
 - pole servicename 3-21
 - výhody a nevýhody 3-7
 - provádění
 - operací jazyka DDL v primárním serveru 19-33
 - provázaný režim 24-5
 - prověřování
 - konfigurační parametry 2-10
 - přehled 1-22
 - režim auditu 5-28
 - průměrná velikost inteligentního velkého objektu 10-25
 - předobraz
 - během rychlé obnovy po kontrolním bodu 15-8
 - definice 15-1, 15-2
 - vyprazdňování 7-27
 - vyrovnávací paměť fyzického protokolu 7-27
 - přejmenování
 - instance databázového serveru 1-13
 - prostoru dbspace 10-19
 - přepínač /usera=xxxx 1-4
 - přepínání
 - mezi jednotkovými procesy 5-10
 - následující soubor protokolu 14-5
 - přepínání kontextu
 - definované 5-8
 - jak pracuje, pokud je řízeno operačním systémem 5-6
 - porovnání operačního systému a provozu s vícenásobnými jednotkovými procesy 5-6
 - přeskakování fragmentů
 - použití vlastností 10-32
 - vliv na transakce 10-33
 - všechny fragmenty 10-31
 - všechny nedostupné 10-32, 10-33
 - vybrané fragmenty 10-33
 - převod, během inicializace 4-5
 - převzetí totožnosti klientu 3-13
 - převzetí totožnosti uživatele 3-13
 - přidání
 - virtuální procesory 5-10
 - přidělení diskového prostoru
 - metadata 10-24
 - oblast 9-8
 - počáteční blok 10-6
 - procedura 10-3
 - prostor v předpřipraveném souboru 10-4
 - prostory sbpace 9-18
 - přehled 1-6
 - sdílená paměť 1-4
 - systém Windows
 - fyzická jednotka 10-7
 - UNIX
 - předpřipravené soubory 10-4
 - soubory s přímým přístupem 10-5

přidělení diskového prostoru (*pokračování*)

- Windows
 - logická jednotka 10-6
 - soubory systému NTFS 10-6
 - zrcadlená data 18-2
- přidělení disku 1-6
- příkaz ALLOCATE COLLECTION 11-5
- příkaz ALLOCATE DESCRIPTOR 11-5
- příkaz ALLOCATE ROW 11-5
- příkaz ALTER ACCESS_METHOD 11-4
- příkaz ALTER FRAGMENT 11-4
- příkaz ALTER FUNCTION 11-4
- příkaz ALTER INDEX 11-4
- příkaz ALTER PROCEDURE 11-4
- příkaz ALTER ROUTINE 11-4
- příkaz ALTER SEQUENCE 11-4
- příkaz ALTER TABLE 11-4
 - protokolování 11-4
 - připojení ke klientům 3-31
 - změna typu tabulky 9-25
- příkaz BEGIN WORK 11-4
- příkaz CLOSE 11-5
- příkaz CLOSE DATABASE 11-4
- příkaz COMMIT 11-4, 11-5
- příkaz CONNECT 3-31, 11-5
 - příklad 3-32
- příkaz CREATE ACCESS_METHOD 11-4
- příkaz CREATE AGGREGATE 11-4
- příkaz CREATE CAST 11-4
- příkaz CREATE DATABASE 9-9, 11-4
- příkaz CREATE DISTINCT TYPE 11-4
- příkaz CREATE EXTERNAL TABLE 11-4
- příkaz CREATE FUNCTION 5-16, 11-4
- příkaz CREATE FUNCTION FROM 11-4
- příkaz CREATE INDEX 11-4
- příkaz CREATE OPAQUE TYPE 11-4
- příkaz CREATE OPCLASS 11-4
- příkaz CREATE PROCEDURE 11-4
- příkaz CREATE PROCEDURE FROM 11-4
- příkaz CREATE ROLE 11-4
- příkaz CREATE ROUTINE 11-4
- příkaz CREATE ROUTINE FROM 11-4
- příkaz CREATE ROW TYPE 11-4
- příkaz CREATE SEQUENCE 11-4
- příkaz CREATE SCHEMA 11-4
- příkaz CREATE SYNONYM 11-4
- příkaz CREATE TABLE 11-4
 - fragmentace
 - s oddíly 10-20
 - protokolování 11-4
 - připojení ke klientům 3-31
 - určení prostorů dbspace v klauzuli PUT 9-14
 - volba IN prostor_dbspace 9-9
- příkaz CREATE Temporary TABLE 11-4
- příkaz CREATE TRIGGER 11-4
- příkaz CREATE VIEW 11-4
- příkaz CREATE XADATASOURCE 11-4
- příkaz DATABASE 3-31, 11-5
- příkaz DEALLOCATE COLLECTION 11-5
- příkaz DEALLOCATE DESCRIPTOR 11-5
- příkaz DEALLOCATE ROW 11-5
- příkaz DECLARE 11-5
- příkaz DESCRIBE 11-5
- příkaz DISCONNECT 11-5
- příkaz DROP ACCESS_METHOD 11-4
- příkaz DROP AGGREGATE 11-4
- příkaz DROP CAST 11-4

- příkaz DROP DATABASE 11-4
- příkaz DROP FUNCTION 11-4
- příkaz DROP INDEX 11-4
- příkaz DROP OPCLASS 11-4
- příkaz DROP PROCEDURE 11-4
- příkaz DROP ROLE 11-4
- příkaz DROP ROUTINE 11-4
- příkaz DROP ROW TYPE 11-4
- příkaz DROP SEQUENCE 11-4
- příkaz DROP SYNONYM 11-4
- příkaz DROP TABLE 11-4
- příkaz DROP TRIGGER 11-4
- příkaz DROP TYPE 11-4
- příkaz DROP VIEW 11-4
- příkaz DROP XADATASOURCE 11-4
- příkaz DROP XADATASOURCE TYPE 11-4
- příkaz EXECUTE 11-4
- příkaz EXECUTE FUNCTION 11-4
- příkaz EXECUTE IMMEDIATE 11-4
- příkaz EXECUTE PROCEDURE 11-4
- příkaz FETCH 11-5
- příkaz FLUSH 11-4
- příkaz FREE 11-5
- příkaz GET DESCRIPTOR 11-5
- příkaz GET DIAGNOSTICS 11-5
- příkaz GRANT 11-4
- příkaz imcadmin 3-43
- příkaz INFO 11-5
- příkaz LOAD 10-46, 11-4
- příkaz LOCK TABLE 11-5
- příkaz OPEN 11-5
- příkaz OUTPUT
 - protokolování 11-5
- příkaz PREPARE 11-5
- příkaz PUT 11-4
- příkaz RENAME COLUMN 11-4
- příkaz RENAME DATABASE 11-4
- příkaz RENAME INDEX 11-4
- příkaz RENAME SEQUENCE 11-4
- příkaz RENAME TABLE 11-4
- příkaz REVOKE 11-4
- příkaz REVOKE FRAGMENT 11-4
- příkaz SELECT INTO TEMP 11-4
- příkaz SET ... 11-5
- příkaz SET DATASKIP 10-31, 10-32
- příkaz starts dbservername 1-14
- příkaz tail -f 1-24
- příkaz UNLOAD 11-4
- příkaz UNLOCK TABLE 11-5
- příkaz UPDATE STATISTICS 2-3, 7-19, 7-20, 9-24
- příkaz WHENEVER 11-5
- příkazy DELETE 11-4
- příkazy INSERT 11-4
- příkazy jazyka SQL
 - ALTER TABLE 9-8
 - analýza a optimalizace 8-6
 - CREATE TABLE 9-9
 - CREATE TABLE, klauzule PUT 9-14
 - databáze s protokolováním 11-4
 - DECLARE 11-4
 - monitorování 12-6
 - nikdy neprotokolovaný 11-4
 - počáteční připojení k databázovému serveru 1-16
 - použití dočasného diskového prostoru 9-27
 - UPDATE STATISTICS 7-19
 - vytvoření prostoru dbspace 9-14
 - vždy protokolované 11-3

- příkazy SELECT 11-5
- příkazy UPDATE 11-4
- příklady
 - jak je spouštěno čištění 7-25
 - konfigurační parametr DBSERVERALIASES 3-32
 - položky souboru /etc/services 3-10
 - připojení IPX/SPX 3-40
 - připojení prostřednictvím místní zpětné smyčky 3-38
 - připojení prostřednictvím sdílené paměti 3-37
 - připojení TCP/IP 3-39
 - více typů připojení 3-40
- přímá disková zařízení 1-5
- připojení
 - bezpečnostní omezení 3-12
 - databáze a síť 3-4
 - definované 5-23
 - klientských aplikací k serveru 3-5
 - metody 5-22
 - místní zpětná smyčka
 - definovaná 3-8
 - příklad 3-38
 - multiplexní 3-4
 - pole typu 3-17
 - prostřednictvím sdílené paměti, definované 3-6
 - protokol IPX/SPX 3-40
 - síťové, příklad 3-39
 - TCP/IP 3-9, 3-11, 3-12
 - více
 - typů připojení, příklad 3-40
- připojení ke sdílené paměti 7-5
 - obslužné programy databázového serveru 7-6
 - virtuální procesory 7-6
- připojení prostřednictvím pojmenovaného propojení procesů
 - definované 3-5, 3-8
 - podporované platformy 3-5
- připojení prostřednictvím sdílené paměti
 - pole servicename 3-21
 - příklad 3-37
 - virtuální procesor 5-22
 - vyrovnávací paměti zpráv 7-20
 - způsob připojení klientu 7-6
- připojení TCP 5-23
- připojený adresář systému NFS 9-3
- přírůstkové zálohování
 - definované 1-19
- přístup k disku bez vyrovnávací paměti
 - porovnání s přístupem s vyrovnávací pamětí 9-4
 - uložení dat 9-4
- příznak LO_CREATE_LOG 9-21
- příznak LO_CREATE_NOLOG 9-21
- příznak LO_CREATE_TEMP 9-21, 10-27
- příznak LO_LOG 13-9
- příznak LO_NOLOG 13-9

R

- referenční omezení 9-24, 11-2
- registr
 - definování vícenásobných síťových adres 5-27
 - změna názvu hostitele 1-27
- registr HKEY_LOCAL_MACHINE 1-10
- registr sqlhosts
 - definovaný 3-14, 3-15
 - proměnná prostředí INFORMIXSQLHOSTS 3-15
 - umístění 3-14
 - v systému Windows 1-10

- relace
 - aktivní prostor tblspace 7-17
 - definované 5-8
 - jednotkové procesy 7-18
 - jednotkové procesy sqlexec 5-3
 - mezipaměť rutin UDR 7-20
 - mezipaměť slovníku 7-19
 - primární jednotkový proces 7-18
 - řídící blok 5-8
 - sdílená paměť 7-18
 - společná oblast sdílené paměti 7-14
 - zámky 7-14
- replikace dat 11-2
 - Viz též* replikace HDR
 - datové typy, které lze replikovat 19-3
 - doba čekání na odpověď 2-8
 - interval vyprazdňování 2-8
 - jak jsou replikovány aktualizace 19-7
 - jak pracuje 19-6
 - moduly DataBlade, současná instalace 19-6
 - návrh klientů k použití
 - primárního databázového serveru 19-31
 - počáteční replikace 19-6
 - popis 19-2
 - použití skupin databázových serverů 3-25
 - přehled 1-21
 - replikace Enterprise Replication 1-22
 - replikace HDR 1-22
 - restartování po selhání 20-13
 - restartování, je-li poškozen index sekundárního serveru 20-13
 - režim 19-3
 - režim pouze pro čtení 4-8
 - role
 - záznamy protokolu 19-7
 - rutiny UDR, současná instalace 19-6
 - typy UDT, současná instalace 19-6
 - velikosti stránek jiné než výchozí v prostředí replikace HDR 20-4
 - vyrovnávací paměť 7-13, 19-7
- replikace Enterprise Replication
 - replikace HDR, současné použití 19-5
- replikace HDR
 - administrace 20-9
 - akce, které je třeba provést, pokud
 - dojde k selhání sekundárního serveru 20-20
 - primární 20-20
 - asynchronní aktualizace 19-8
 - definovaná 19-4
 - Enterprise Replication 19-5
 - konfigurace propojitelnosti 20-3, 20-6
 - konfigurační parametr DRINTERVAL 19-8
 - konfigurační parametr DRTIMEOUT 20-19
 - monitorování stavu 20-17
 - možná selhání 20-18
 - nastavení 20-2
 - typ databázového serveru 20-16
 - návrh klientů k použití
 - sekundárního databázového serveru 19-32
 - obnovení systému po selhání prostředku 20-22, 20-23
 - plánování 20-2
 - požadavky na data 20-3
 - požadavky na hardware a operační systém 20-3
 - provádění operací DDL 19-33
 - přesměrování klientů 19-23
 - DBPATH 19-24
 - INFORMIXSERVER 19-28
 - porovnání metod 19-30

- replikace HDR *(pokračování)*
 - přesměrování klientů *(pokračování)*
 - soubor sqlhosts 19-25
 - z aplikace 19-29
 - restartování po selhání 20-21, 20-24, 20-25
 - restartování po zjištění, že je index porušen 20-23
 - role
 - dočasné prostory dbspace 19-33
 - primárního databázového serveru 19-3
 - sekundárního databázového serveru 19-3
 - ruční přepnutí 20-20, 20-21
 - skripty hdrmkpri a hdrmksec 20-17
 - spuštění 20-6
 - stav bloku na sekundárním 20-11
 - synchronizace 19-11
 - synchronní aktualizace 19-8
 - typy replikovaných dat 19-3
 - volby šifrování 2-10, 20-14
 - výhody 19-4
 - zjišťování selhání 20-19
 - změna režimu databázového serveru 20-15
 - změna typu databázového serveru 20-17
 - ztracené a nalezené transakce 19-9
- rezervovaná oblast
 - definovaná 9-13
 - monitorování velikosti 10-42, 10-45
 - přesunutí prostoru do oblasti metadat 10-45
- rezervovaná stránka PAGE_CONFIG 2-11, 4-5, 4-6
- rezervovaná stránka PAGE_PZERO 4-5
- rezervované stránky
 - ověření 23-3
 - stav bloku na sekundárním serveru 20-11
- rezidentní sdílená paměť
 - definovaná 4-4, 7-9
 - interní tabulky 7-15
- režim
 - administrace 4-8, 4-15
 - administrace uživatelem DBSA 4-9
 - definovaný 4-7
 - do režimu offline z kteréhokoliv režimu 4-14
 - klidový 4-8
 - nenásilně vypnutí 4-12
 - obnova 4-8
 - offline 4-8
 - okamžitě vypnutí 4-13
 - online 4-8
 - opětovná inicializace sdílené paměti 4-11
 - přesun souborů protokolu 14-14
 - přidání souborů protokolu 14-11
 - uvedení do režimu offline 4-14
 - vypnutí 4-8
 - vypuštění souborů protokolu 14-12, 14-13
 - z offline do administrace 4-12
 - z offline do klidového 4-11
 - z offline do online 4-11
 - z administrace do klidového 4-14
 - z administrace do online 4-14
 - z klidového do administrace 4-13
 - z online do administrace 4-13
 - z online do klidového
 - nenásilně 4-12
 - okamžitě 4-13
 - změna 4-9
 - režim administrace 4-8, 4-12, 4-13, 4-14, 4-15
 - pro pár replikace HDR 20-16
 - režim offline
 - definovaný 4-8
 - režim online
 - definovaný 4-8
 - režim pouze pro čtení
 - definovaný 4-8
 - režim velkých bloků 1-5
 - režimy SQLTRACE 26-12
 - rozdělené čtení 17-2, 17-5
 - rozhraní API DataBlade
 - inteligentní velké objekty, přístup 9-19
 - rozhraní DataBlade API
 - inteligentní velké objekty, přístup 9-13
 - velikost inteligentních velkých objektů 9-16
 - rozhraní SQL Administration API 26-1
 - rozhraní Transport-Layer Interface
 - v poli nettype 3-18
 - rozložení zatížení
 - pomocí parametru DBSPACETEMP 9-34
 - pomocí virtuálních procesorů 5-6
 - s cílem maximalizovat výkon 9-33
 - rozšířená uživatelská oprávnění 1-27
 - rozšířené datové typy
 - replikace dat 19-3
 - rozvržení disků
 - izolace tabulek 9-34
 - kompromisy 9-35
 - optimální výkon 9-34
 - správci logických disků 9-39
 - ukázkové rozvržení disků 9-35
 - zrcadlení 9-34
 - ruční obnova
 - postup zjištění, zda je nezbytná 25-1
 - použití identifikátoru GTRID 25-3
 - příklad 25-5
 - rozhodnutí, zda je nutná akce 25-4
 - určení, zda jsou data nekonzistentní 25-2
 - získání informace ze souborů logického protokolu 25-3
 - rychlá obnova 4-5, 4-8, 11-2
 - bez protokolování 15-8
 - definice 15-7
 - definovaná 1-21
 - fáze čištění 14-11
 - inteligentní velké objekty 9-20
 - jak databázový server zjistí, kdy je potřeba 15-7
 - kdy nastane 15-7
 - odvolávání neprovedených transakcí 15-9
 - po kontrolním bodu 15-8, 15-9
 - podrobnosti procesu 15-8
 - prostory sbspace 9-13
 - přehrávání záznamů logického protokolu 15-9
 - přetečení fyzického protokolu 15-7
 - typy tabulek 9-25
 - účel 15-7
 - vliv protokolování s vyrovnávací pamětí 15-7
 - rychlé cyklické dotazování 5-25

Ř

- řazení
 - paralelní zpracování 5-6
 - sdílená paměť 7-20
- řídící blok jednotkového procesu
 - role při přepínání kontextů 7-18
 - vytvoření 7-18
- řídící blok relace 5-8
 - definovaná 7-18
 - sdílená paměť 7-18

- řídící struktury
 - definované 5-8
 - fronty 5-10
 - řídící blok jednotkového procesu 5-8
 - řídící blok relace 5-8
 - zásobníky 5-9

S

sada IBM Informix Client Software Developer's Kit 24-2

sdílená data 7-2

sdílená paměť

- část 7-3

- data jednotkového procesu 7-18

- definovaná 7-2

- globální společná oblast 7-20

- haldy 7-19

- hashovací tabulka vyrovnávacích pamětí 7-16

- hodnota klíče 7-7

- identifikátor 7-7

- identifikátor segmentu 7-7

- inicializace 4-1, 4-3

- inicializace a restartování 4-5

- inteligentní velké objekty 7-32, 9-18

- interní tabulky 7-15

- izolace jednotkových procesů a vyrovnávací paměti zámek 7-21

- kommunikace 3-20, 3-21, 7-20

- konfigurace 7-4, 7-14, 8-2

- konfigurační parametr SERVERNUM 7-7

- konfigurační parametr SHMADD 7-14

- konfigurační parametr SHMBASE 2-5, 7-7

- konfigurační parametr SHMTOTAL 2-5, 7-4

- konfigurační parametr SHMVIRTSIZE 2-5, 7-14

- konfigurační parametr STACKSIZE 7-19

- konfigurační parametry 2-6

- kontrolní body 15-5

- kopírování do souboru 8-7

- kritické části 15-2

- mezipaměť distribuce dat 7-19

- mezipaměť pro příkazy SQL 2-6, 7-4, 7-19, 8-6

- mezipaměť slovníku 7-19, 8-5

- mezipaměť uživatelských rutin 7-20

- meziprocesová komunikace 7-2

- monitorování

- ISA 8-5

- obslužný program onstat 8-7

- nastavení 8-6

- nerezidentní část 8-4

- objekty mutex 7-21

- obsah 7-21

- pokyny pro semaforey 8-3

- problém s adresou dolní hranice 7-8

- první segment 7-7

- přehled 1-21

- přídělení 1-4, 4-4, 7-14

- přídělování vyrovnávacích pamětí 7-10

- připojení 5-23, 7-5

- připojení dalších segmentů 7-7, 7-8, 8-5

- připojování obslužných programů 7-6

- rezidentní část

- definovaná 7-9

- ISA 8-5

- obslužný program ON-Monitor 8-5

- textový editor 8-5

- vytvoření 4-4

- řazení 7-20

- řídící blok jednotkového procesu 7-18

sdílená paměť (*pokračování*)

- řídící blok relace 7-18

- segmenty operačního systému 7-4

- společná oblast vyrovnávací paměti 7-10

- společné oblasti 7-14

- stránky blobpage 7-30

- synchronizace vyprazdňování vyrovnávacích pamětí 7-27

- tabulka bloků 7-16

- tabulka jednotkových procesů čištění stránek 7-17

- tabulka prostorů dbspace 7-16

- tabulka prostorů tblspace 7-17

- tabulka transakcí 7-17

- tabulka uživatelů 7-18

- tabulka vyrovnávacích pamětí 7-15

- tabulka zrcadlených bloků 7-16

- tabulky 7-15

- účel 7-2

- velikost

- virtuální část 7-14

- zobrazená programem onstat 7-4

- virtuální část 7-14, 7-15, 8-5, 8-7

- ISA 8-5

- vliv parametrů operačního systému 8-2

- volby výkonu 7-2, 8-5

- vyrovnávací paměť 10-14

- vyrovnávací paměť fyzického protokolu 7-12

- vyrovnávací paměť logického protokolu 7-11

- vyrovnávací paměť replikace dat 19-7

- záhlaví 7-8, 7-10

- zámky vyrovnávacích pamětí 7-21

- zásobníky 7-18

- změna uložení pomocí obslužného programu onmode 8-7

sdílené zámky

- definované 7-22

- sekundární databázový server 19-3, 20-11

sekundární server RS

- alarmy událostí a zprávy 21-8

Sekundární server RS

- spuštění 21-4

sekundární server SD

- alarmy událostí a zprávy 22-6

selhání systému

- vliv na databázi 15-7

- semaforey, parametry systému UNIX 8-3

server Apache 4-10

- servery NIS, vliv na soubory /etc/hosts a /etc/services 3-10

síť

- karty rozhraní

- použití 5-27

- konfigurační soubory 1-10

- soubory zabezpečení 1-10

síťová komunikace

- implementace 5-23

- prostřednictvím protokolu IPX/SPX 3-20, 3-21, 3-40

- prostřednictvím protokolu TCP/IP 3-19, 3-20

- typy 5-22

síťové protokoly 5-22

síťové virtuální procesory

- definované 5-22

- jednotkové procesy cyklického dotazování 5-22

- kolik 5-23

síťový protokol

- definovaný 3-2

skript hdrmkpri 20-17

skript hdrmksec 20-17

skript ids-example.rc 1-15

skupina
 bargroup 1-6
 databázový server 3-23, 3-24, 3-25, 3-27
 paralelní zpracování 5-6
skupina bargroup 1-6
skupina databázových serverů 3-23
skupina Informix-Admin
 správa souboru protokolu 14-1
 vlastnictví souboru 1-6
 změna provozních režimů 4-10
skupiny databázových serverů
 klíč registru 3-24
 vytvoření s použitím klíče registru SQLHOSTS 3-24
 vytvoření v souboru sqlhosts 3-23
sledování výkonu v systému Windows 1-26
služba
 protokol IPX/SPX 3-21
služba Network Information Service 3-10
služba Windows Internet Name Service 3-10, 3-28
služby pro podporu komunikace
 definované 3-8
 utajení a integrity zpráv 3-8
smíšený výsledek transakce 24-11
sokety 3-5
 v poli nettype 3-18
souběžný přístup
 řízení 7-21
soubor .informix 1-8
soubor .rhosts 3-12
soubor /etc/hosts 19-27
soubor /etc/services 19-27
soubor boot.ini 1-4
soubor core 23-4
soubor dbservername.cmd 1-9, 1-12
soubor hosts 1-10, 3-9, 3-10
soubor hosts.equiv 3-12
soubor chyby aserce
 af.xxx 23-4
 gcore.xxx 23-4
 seznam 23-3
 shmem.xxx 23-4
soubor informix.rc 1-8
soubor JAR 1-13
soubor logického protokolu
 alarmy událostí 14-16, 14-17
 číslo souboru protokolu 13-2
 definovaný 11-1, 13-1
 dočasný 14-8
 dynamické přidělování
 definice 14-9
 monitorování 14-16, 14-17
 umístění souborů 14-10
 velikost 14-10
 velikost souboru 14-10
 jedinečné identifikační číslo 13-2, 13-3
 kontrolní bod pro uvolnění 16-4
 minimální a maximální velikost 13-3
 následky neuvolnění 13-5
 nelze přidat k prostoru dbspace s jinou než výchozí velikostí
 stránky 14-14
 nelze přidat k prostoru blobspace nebo sbspace 14-11
 nelze přidat k prostoru dbspace s jinou než výchozí velikostí
 stránky 14-11
 odhad potřebného počtu 13-4, 14-4
 odstranění 14-12
 používání tabulek SMI 14-9
 pozice v protokolu 24-14
soubor logického protokolu (*pokračování*)
 přepínání 13-7, 14-5
 přepnutí za účelem aktivace prostorů blobspace 13-7
 přesun do jiného prostoru dbspace 14-14
 při rychlé obnově 15-9
 při rychlé obnově po kontrolním bodu 15-8
 přidání souboru protokolu
 pomocí programu ON-Monitor 14-12
 pomocí příkazu onparams 14-11
 přidělení diskového prostoru pro 13-3
 role v rychlé obnově 15-9
 ruční přidání souborů protokolu 14-11
soubor logického protokolu
 změna velikosti 14-12
stav
 A 14-6, 14-12, 14-13
 B 14-6, 14-7
 C 14-7
 D 14-6, 14-13
 definovaný 13-3
 F 14-13
 L 14-7
 U 14-6, 14-7, 14-13
umístění 13-2, 14-10
uvolnění souborů 13-5, 14-6, 14-7
virtuální procesor LIO 5-18
vstup - výstup 5-18
vypuštění souboru protokolu
 pomocí programu ON-Monitor 14-13
 pomocí příkazu onparams 14-12
záloha
 cíle 14-4
 přidání souborů protokolu 14-12
 uvolnění odstraněných stránek blobpage 13-7
 vliv na výkon 13-4
 změna fyzického schématu 10-8
 změna velikosti 14-13
 zrcadlení prostoru dbspace, který obsahuje soubor 17-4
soubor netrc
 definovaný 3-12
 volby zabezpečení v informacích sqlhosts 3-26
soubor oncfg_servername.servernum 4-6
soubor onconfig.demo 1-11
soubor onconfig.std
 informace o společné oblasti vyrovnávacích pamětí 10-13
soubor services 1-10, 3-10
soubor setenv.cmd 1-9
soubor shmem
 chyby aserce 23-4
soubor sqlhosts
 definování vícenásobných síťových adres 5-27
 definovaný 3-23
 pole dbservername 3-17, 3-25
 pole názvu hostitele 3-19
 pole názvu služby 3-20, 5-23
 pole typu připojení 3-17
 pole voleb 3-20
 příklad místní zpětné smyčky 3-38
 příklad sdílené paměti 3-37
 příklad síťového připojení 3-39
 příklad více typů připojení 3-39
 standardní čísla portů IANA 3-16
 úprava pomocí programu ISA 1-10
 určení síťových jednotkových procesů cyklického
 dotazování 5-22
 volba CSM 3-21
 volba keep-alive 3-26

- soubor sqlhosts (*pokračování*)
 - volba multiplexního připojení 3-4
 - volba skupiny 3-23, 3-26
 - volby zabezpečení 3-26
 - záznamy pro více karet rozhraní 5-27
- soubor šablony onconfig.std 1-7, 1-11
- soubor vlastností JVP 1-13
- souborový server NetWare 3-20
- soubory
 - core 23-4
 - hosts 1-10
 - konfigurace 1-11
 - konfigurace propojitelnosti 3-9
 - konfigurace sítě 1-10
 - NTFS 9-3
 - oncfg_servername.servernum 4-6
 - ONCONFIG 1-11, 4-3, 4-4
 - oprávnění 10-5
 - passwd 1-10
 - předpřipravený 1-6
 - services 1-10
 - soubor hosts.equiv 3-12
 - sqlhosts 1-10
 - vlastnosti JVP 1-13
 - zabezpečení sítě 1-10, 3-12
- soubory systému NTFS 9-3
 - převedení 1-6
- souvislý
 - prostor pro fyzický protokol 16-2
- spící jednotkové procesy
 - spící navždy 5-11
 - typy 5-11
- spojení
 - paralelní zpracování 5-6
- společné oblasti vyrovnávací paměti
 - 64bitové adresování 7-11
 - čtení napřed 7-26
 - definovaný 7-10
 - inteligentní velké objekty 7-32
 - minimální požadavky 7-10
 - monitorování aktivity 8-10
 - obcházení v případě dat prostoru blobspace 7-30
 - obsah 7-10
 - odstranění 10-16
 - pro 32bitové platformy 10-14
 - pro 64bitové platformy 10-14
 - pro jiné než výchozí velikosti stránek 10-12
 - správa front LRU 7-23
 - synchronizace vyprazdňování vyrovnávacích paměti 7-28
 - velikost vyrovnávací paměti 7-11
 - vytvoření pro jiné než výchozí velikosti stránek 10-12
 - vytvoření pro jiné než výchozí velikosti stránky 10-13
 - změna velikosti 10-16
- spouštěcí skript 1-14, 1-15
- správa disku 1-6
- správce logických disků
 - definovaný 9-38, 17-3
- správce paměti
 - ISM 1-17
 - role v systému ON-Bar 1-17
- správce transakcí
 - MTS/XA 24-2
 - TP/XA 1-23, 24-2
 - účel 24-2
- správce zdrojů 24-2
- správci zdrojů 24-3
- spuštění databázového serveru
 - automaticky 1-15
 - inicializace diskového prostoru 1-14
 - pomocí oninit 4-1
- SQL Administration API 26-3, 26-10
- standard IANA 3-16
- standardní databázový server 19-3
- statistické údaje distribuce dat 7-19
- statistické údaje o paměti
 - prostory blobspace 10-23
- stav
 - soubor protokolu
 - aktuální 14-6
 - kontrolní bod 14-13
 - odstraněný 14-7, 14-13
 - použitý 2-6, 14-6, 14-13
 - přidaný 14-6, 14-7, 14-13
 - volný 14-6, 14-13
 - zálohovaný 14-7, 14-13
 - stav protokolování databáze
 - definice 11-5
 - kompatibilní se standardem ANSI 11-6
 - změna režimu 12-3
 - nastavení 11-3
 - protokolování bez vyrovnávací paměti 11-5
 - protokolování s vyrovnávací pamětí 11-6
 - přehled 1-20
 - ukončení protokolování
 - pomocí obslužného programu ontape 12-4
 - používání obslužného programu ondblog 12-3
 - zajištění kompatibility se standardem ANSI
 - pomocí obslužného programu ontape 12-5
 - používání obslužného programu ondblog 12-3
 - zapnutí protokolování pomocí obslužného programu ontape 12-4
 - změna
 - pomocí obslužného programu ontape 12-4
 - pomocí programu ISA 12-5
 - pomocí programu ON-Monitor 12-5
 - používání obslužného programu ondblog 12-3
 - změna stavu použití vyrovnávací paměti
 - pomocí obslužného programu ontape 12-4
 - pomocí příkazu SET LOG 11-7
 - používání obslužného programu ondblog 12-3
 - změny povoleny 12-2
 - zrušení protokolování pomocí obslužného programu ondblog 12-3
- stránka
 - definovaná 9-5
 - naposledy použitá 7-23
 - nejdéle nepoužívaná 7-23
 - určení velikosti pro standardní nebo dočasný prostor
 - dbspace 10-12
 - určení velikosti stránky databázového serveru 10-23
 - vztah k bloku 9-6
- stránka blobpage
 - definovaná 9-6
 - doporučení velikosti 10-23
 - fyzické protokolování 15-3
 - statistické údaje o paměti 10-23
 - uvolnění odstraněných stránek 13-7
 - vztah k bloku 9-6
 - zázpisy, které obcházejí sdílenou paměť 7-30
 - zaplnění
 - určení 10-40
 - zobrazení příkazem oncheck -pB 10-40
- stránka sbpage
 - definovaná 9-7

- stránka sbpage *(pokračování)*
 - doporučení velikosti 9-7
- stránky bitových map
 - prostor tblspace 9-29
 - prostor tblspace indexu 9-29
- syssqltrace table 26-11
- systém Windows
 - obslužný program ixpasswd 1-26
 - obslužný program ntchname 1-27
 - prohlížeč událostí 1-26
- systémová konzole 1-25
- systémový časovač 5-11

Š

- šifrovací virtuální procesor 5-27
- šifrování
 - virtuální procesory 5-27
- Šifrování
 - v HDR 20-14
- šifrování dat
 - v HDR 20-14

T

- tabulka
 - definovaná 9-22
 - dočasná
 - odhad diskového prostoru 9-33
 - vyčištění při restartu 9-28
 - fragmentace 9-24
 - izolace často používaných 9-34
 - oblast 9-8, 9-22
 - oddíly uprostřed disků 9-37
 - posouzení potřebné paměti 9-33
 - poškozená 9-23
 - pravidla pro rozvržení disků 9-34
 - standardní 9-24
 - vypouštění 10-29
 - vytvoření s datovými typy CLOB a BLOB 9-19
- tabulka command_history 26-1, 26-3
- tabulka FAT
 - oddíly 1-6
- tabulka jednotkových procesů čištění stránek
 - definovaná 7-17
 - počet položek 7-17
- tabulka PH_ALERT 26-3
- tabulka PH_GROUP 26-3
- tabulka PH_RUN 26-3
- tabulka ph_task 26-5, 26-6
- tabulka PH_TASK 26-3, 26-6
- tabulka PH_THRESHOLD 26-3
- tabulka přesměrování 10-4
- tabulka sysdistrib 1-25, 7-19
- tabulka syslogs 14-9
- tabulka sysprofile 8-8
- tabulka systables
 - hodnoty příznaků 9-24
- tabulka sysvpprof 6-6
- tabulka transakcí
 - definovaná 7-17
 - zaznamenávání pomocí příkazu onstat 7-17
- tabulka typu RAW 1-20
 - obnovení 9-26
 - přehled 9-24
 - rychlá obnova 9-26

- tabulka typu RAW *(pokračování)*
 - vlastnosti 7-26
 - zálohování 9-26
 - zálohování a obnovení 9-26
 - změna 12-5
- tabulka typu STANDARD
 - obnovení 9-25, 9-26
 - povolena v databázi s protokolováním 1-20
 - rychlá obnova 9-25
 - vlastnosti 9-24
 - záloha 9-26
- tabulka typu TEMP 1-20
 - obnovení 9-26
- tabulka uživatelů
 - definovaná 7-18
 - maximální počet položek 7-18
- tabulka zámek
 - definovaná 7-13
 - konfigurace 7-13
 - obsah 7-13
- tabulky bez protokolování 9-24
- tabulky SMI
 - během inicializace 4-7
 - monitorování
 - bloky 10-34
 - databáze 8-8, 10-34, 12-6
 - informace ve vyrovnávací paměti 8-10
 - kontrolní body 16-5
 - objekty mutex 8-8
 - oblastí 10-38
 - použití vyrovnávací paměti protokolu 16-5
 - prostory tblspace 10-38
 - replikace dat 20-18
 - sdílená paměť 8-8
 - soubory logického protokolu 14-9
 - společná oblast vyrovnávací paměti 8-8, 16-5
 - typy zápisu 8-11
 - virtuální procesory 6-6
 - využití vyrovnávací paměti 10-38
 - zámky latch 8-8
 - použití k monitorování databázového serveru 1-25
 - příprava během inicializace 4-7
 - sysextents 10-38
 - tabulka systabnames 10-38
- tabulky systémového katalogu
 - mezipaměť slovníku 7-19
 - optimální uložení 9-37
 - ověření 23-2
 - tabulka sysdistrib 7-19
 - umístění 9-21
- tělesně postižení A-1
- terminálové rozhraní 1-8
- textový editor
 - nastavení konfiguračních parametrů
 - sdílená paměť 8-5
 - virtuální procesory 6-2
 - výkon 8-5
 - vytvoření souboru ONCONFIG 1-11, 6-1
- TP/XA 1-23, 7-17, 24-2
- transakce
 - globální
 - definice 24-2, 24-5
 - identifikační číslo, GTRID 25-3
 - určení, zda byla implementována nekonzistentně 25-1
 - zaznamenání 12-6, 24-13
 - monitorování 12-6, 24-6, 24-13
 - provázané 24-4

- transakce (*pokračování*)
 - smíšený výsledek 24-11
 - tabulky typu RAW 9-24
 - volně vázané 24-4
- trasování jazyka SQL
 - globální zakázání 26-15
 - povolení 26-15, 26-16
 - přehled 26-11
 - příklad informací o trasování 26-13
 - režim 26-14
 - režimy 26-11, 26-12
 - určení informací o 26-14
 - úroveň 26-14
 - zakázání pro relaci 26-15
- třída VP v konfiguračním parametru NETTYPE 5-22
- třídy
 - virtuální procesor 5-3
- typy tabulek
 - hodnoty příznaků v tabulce systables 9-24
 - obnovení 9-26
 - RAW 9-24, 12-5
 - rychlá obnova 9-25
 - souhrn 9-24
 - STANDARD 9-24, 12-6
 - v protokolujících databázích 1-20
 - vlastnosti 9-26
 - zálohování před převodem 10-8
- typy zápisů
 - zápis front LRU 7-28
 - zápis na popředí 7-28
 - zápisy bloků 7-29
- typy zápisů vyrovnávacích pamětí 7-28

U

- účastnický databázový server 24-6
 - automatická obnova 24-7
- ukončovací skript 1-15
- úloha
 - nastavení 26-6
- úlohy administrace
 - kontrola konzistence 23-2
 - přiřazování paměti 9-9
 - rutinní úlohy 1-19
- uložení dat
 - koncepty 9-2
 - maximální blok
 - velikost 10-4, 10-8
 - maximální počet bloků 10-8
 - maximální počet paměťových prostorů 10-8
 - řízení 9-9, 9-14
- upozornění
 - soubory v systémech NIS 3-10
 - výstup příkazu oncheck -cc 23-2
- úpravy
 - soubor ONCONFIG 1-12
 - údaje sqlhosts
 - UNIX 1-10
- určení adresy pomocí zástupných znaků
 - klientská aplikace 3-28
 - pole názvu hostitele 3-30
 - příklad 3-30
- úroveň izolace neaktualizované čtení 19-32
- úrovně, zálohování
 - definované 1-19
 - prostory sbpace 14-5
- usnadnění přístupu A-1

- usnadnění přístupu (*pokračování*)
 - klávesnice A-1
 - klávesové zkratky A-1
- utajení předávaných zpráv 3-8
- uvolnění souborů protokolu 14-6, 14-7
- uživatel informix
 - správa souboru protokolu 14-1
 - změna hesla 1-26
- uživatelská data
 - definovaná 9-13
 - vytvoření 3-13
- uživatelská připojení
 - monitorování 4-7
- uživatelské programy
 - konfigurace mezipaměti 7-20
 - mezipaměť 7-20
 - nesprávně pracující 5-16
 - paralelní zpracování 5-6
 - psaná v jazyku Java 1-13
 - umístění sdílené paměti 5-16
 - virtuální procesor nepředávající řízení 5-6
 - virtuální procesory 5-13
- uživatelské účty a domény systému Windows 3-3
- uživatelské virtuální procesory
 - kolik 5-15
 - použití 5-16
 - spouštění rutin UDR 7-20
 - účel 5-16
- uživatelský jednotkový proces
 - definované 5-3
 - kritické části 15-2
 - monitorování 7-18
 - zaznamenání 7-18
 - získání vyrovnávací paměti 7-23

V

- vázání virtuálních procesorů třídy CPU 5-7
- velikost počáteční oblasti 9-8
- velikost vyrovnávací paměti
 - volba 3-22
- velké vyrovnávací paměti, definované 7-18
- vestavěné datové typy
 - replikace dat 19-3
- větev transakce 24-6
- více typů připojení
 - příklad 3-40
 - sqlhosts 3-32
- vícenásobné jednotkové procesy
 - definované 5-2
 - zdroje operačního systému 5-6
- vícenásobné uložení
 - příklad 3-41
- víceprocesorový počítač
 - konfigurační parametr MULTIPROCESSOR 5-13
 - procesorová afinita 5-7
- virtuální část
 - sdílená paměť
 - část virtuálních rozšíření 7-3
 - globální společná oblast 7-20
 - konfigurace 7-15
 - konfigurační parametr SHMVIRTSIZE 7-14
 - mezipaměť distribuce dat 7-19
 - mezipaměť pro příkazy SQL 7-19
 - mezipaměť rutin UDR 7-20
 - obsah 7-15
 - přidání segmentu 8-7

- virtuální část (*pokračování*)
 - sdílená paměť (*pokračování*)
 - zásobníky 7-18
- virtuální procesor auditu 5-28
- virtuální procesor bezpečný při použití více jednotkových procesů 5-4
- virtuální procesor CPU
 - definované 5-12
 - jednoprocesorový počítač 5-13
 - jednotkové procesy 5-2
 - jednotkové procesy cyklického dotazování 5-22, 5-23
 - kolik 5-12
 - konfigurace parametru AFFNPROCS 5-13
 - konfigurační parametr AFF_SPROC 5-13
 - moduly DataBlade 5-13
 - omezení 5-16
 - prováděné typy jednotkových procesů 5-12
 - přidávání a vypouštění v režimu online 5-14
 - přidávání a vypouštění v režimu online 5-17
 - uživatelská rutina 5-13
 - vázání 5-7
 - víceprocesorový počítač 5-13
- virtuální procesor jazyka Java 5-17
- virtuální procesor nepředávající řízení 5-16
- virtuální procesor typu miscellaneous 5-28
- virtuální procesory
 - během inicializace 4-5
 - definované 5-2
 - diskový vstup - výstup 5-17
 - fronta připravených procesů 5-10
 - jednotkové procesy cyklického dotazování 5-23
 - koordinace přístupu ke zdrojům 5-6
 - monitorování
 - ISA 6-2
 - obslužné programy onstat 6-5
 - nastavení konfiguračních parametrů 6-1
 - nepředávající řízení 5-16
 - obsluhující jednotkové procesy 5-8
 - paralelní zpracování 5-7
 - použití zásobníků 5-10
 - proces s vícenásobnými jednotkovými procesy 5-5
 - přehled 5-2
 - přepínání kontextu 5-8
 - přesouvání jednotkových procesů 5-2
 - přidání a vypouštění
 - obslužný program ON-Monitor 6-3
 - pomocí programu ISA 6-2
 - připojení ke sdílené paměti 7-6
 - přístup ke sdílené paměti 7-3
 - rozšíření.
 - Viz* Uživatelem definovaný virtuální procesor.
 - sdílení zpracování 5-6
 - síť 5-22
 - správa 1-21
 - šifrování 5-27
 - třída ADM 5-11
 - třída ADT 5-28
 - třída AIO 5-20, 7-6
 - třída CPU 5-6
 - třída LIO 5-18
 - třída MSC 5-4
 - třída OPT 5-28
 - třída PIO 5-10, 5-19
 - třídy 5-6, 5-12
 - uživatelem definovaná třída 5-15
 - uživatelská rutina 5-4
 - uživatelské rutiny napsané v jazyku Java 5-17
- virtuální procesory (*pokračování*)
 - vázání k jednotkám CPU 5-7
 - vstup - výstup fyzického protokolu 5-19
 - vstup - výstup logického protokolu 5-18
 - výhody 5-5
 - vypuštění CPU v režimu online 6-3, 6-4
- virtuální procesory AIO
 - automatické zvyšování a snižování 5-21
 - kolik 5-21
- virtuální procesory LIO 5-18
- virtuální procesory PIO
 - definované 5-19
- vkládání dat
 - tabulky typu RAW 9-24
 - tabulky typu STANDARD 9-24
- vlastnosti relace
 - automatická konfigurace 1-17, 1-18
- volba identifikátoru 3-26
- volba keep-alive 3-26
- volba konce skupiny 3-23
- volba přeměrování připojení 3-22
- volně vázaný režim 24-4
- vrstvení odstranění 11-2
- výchozí
 - konfigurační soubor 1-11, 4-4
- výkon
 - čtení napřed 7-26
 - frekvence vyprazdňování vyrovnávacích pamětí 7-24
 - funkce předávání řízení 5-6
 - nástroj k monitorování 1-26
 - parametry, nastavení
 - pomocí programu ON-Monitor 8-5
 - posouzení VP CPU 5-12
 - přepínání kontextu řízené virtuálním procesorem 5-9
 - rezidentní sdílená paměť 7-9
 - sběr dat 1-25
 - sdílená paměť 3-7, 5-6, 7-2
- výlučný zámek
 - vyrovnávací paměť 7-22
- výměna disků za chodu 17-3
- vynucená rezidence
 - nastavení 4-6
- výpis do souboru core
 - kdy je užitečný 23-7
 - obsah souboru gcore.xxx 23-4
- vypnutí
 - automaticky 1-15
 - nenásilně 4-12
 - okamžitě 4-13, 4-14
 - režim
 - definovaný 4-8
 - vedení do režimu offline 4-14
- výpočet velikosti
 - inteligentní velké objekty 9-16
 - kořenový prostor dbspace 9-32
 - metadata 9-19, 10-25
 - stránky blobpage 10-23
 - velikost stránky 10-23
- vyprazdňování
 - předobrazů 7-27
 - vyrovnávací paměť replikace dat, maximální interval 2-8
 - vyrovnávací paměti 7-26
- vypuštění
 - bloku z prostoru dbspace 10-27
 - paměťové prostory 10-29
 - prostoru extspace 10-31
 - prostory sbospace 10-28

- vypuštění (*pokračování*)
 - tabulky v prostorech dbspaces 10-29
- vyrovnávací paměť fyzického protokolu
 - definovaná 7-12
 - konfigurační parametr PHYSBUFF 7-12
 - kontrolní body 7-27
 - monitorování 16-2
 - počet 7-12
 - události, které způsobují vyprázdnění 7-27
 - vyprazdňování 7-12
- vyrovnávací paměť logického protokolu
 - definovaná 7-11
 - kontrolní body 7-11
 - monitorování 16-2
 - vyprazdňování
 - bez předobrazu 7-30
 - databáze bez protokolování 7-30
 - definované 7-29
 - pokud nastal kontrolní bod 7-30
 - protokolování bez vyrovnávací paměti 7-30
 - synchronizace 7-27
 - vyrovnávací paměť logického protokolu 7-29
 - záznamy logického protokolu 7-29
- vyrovnávací paměť pro přijatá data 19-7
- vyrovnávací paměti
 - aktualizovaná 7-26
 - aktuální úroveň přístupu zámku 7-16
 - čekající jednotkové procesy 7-16
 - inteligentní velké objekty 7-32, 9-16
 - jak jednotkový proces
 - přístupuje ke stránce ve vyrovnávací paměti 7-26
 - získá paměť 7-23
 - maximální počet na 64bitové platformě 7-11
 - monitorování statistik a použití 8-8
 - naposledy použita 7-23
 - neaktualizované 7-26
 - nejdéle nepoužívané 7-23
 - replikace dat 7-13, 19-7
 - sdílené zámky 7-22
 - souběžný přístup 7-26
 - synchronizace vyprazdňování 7-27, 7-28
 - tabulka, definovaná 7-16
 - typy zámků 7-21
 - typy zápisů při vyprazdňování 7-28
 - velké vyrovnávací paměti 7-18
 - výlučný režim 7-22
 - vyprazdňování 7-26
 - vyrovnávací paměť fyzického protokolu 7-12
 - vyrovnávací paměť logického protokolu 7-11
 - vyrovnávací paměť stránek blobpage 7-31
- vytvoření
 - dočasné prostory dbspace 10-17
 - inteligentní velké objekty 9-14, 10-25
 - prostorů dbspace 10-9
 - prostory blobspace 10-22
 - prostory sbospace 10-24
 - tabulek s datovými typy CLOB a BLOB. 9-19
- vzdálení
 - hostitelé a klienty 3-12
 - klient 3-12
- vzorec
 - velikost logického protokolu 14-2

W

- Windows
 - automatické spuštění 1-14

- Windows (*pokračování*)
 - konfigurace paměti 1-4
 - maximální adresový prostor 1-4
 - nastavení proměnných prostředí 1-9
 - nastavení propojitelnosti 1-10
 - obslužný program ixsu 1-27
 - ovládací panel proměnných prostředí 1-9
 - převod na NTFS 1-6
 - přídělení diskového prostoru s přímým přístupem 10-6
 - větší sdílená paměť 1-4

X

- X/Open
 - prostředí DTP 11-7
 - standardy rozhraní XA 24-3

Z

- zabezpečení
 - konfigurační parametry šifrování HDR 2-10
 - možnosti šifrování HDR 20-14
 - rizika komunikace prostřednictvím sdílené paměti 3-7
 - soubory pro síť 1-10
 - volby
 - sqlhosts 3-26
- zabezpečení sítě
 - soubor .netrc 3-12
 - soubor hosts.equiv 3-12
 - soubory 3-12
- záloha úrovně 0
 - kontrola konzistence 23-5
- zálohy
 - bloky 10-18
 - data typu TEXT a BYTE 9-12
 - definované 1-19
 - fyzický protokol 10-8
 - kontrolní body 15-6
 - odstraněné soubory protokolu 14-12
 - ověření 1-19
 - prostorů dbspace 10-9
 - prostory blobspace 10-22
 - prostory sbospace 9-19, 10-24, 14-5
 - převod typu tabulky 10-8
 - přidání souborů protokolu 14-12
 - soubory protokolu 10-8, 14-4
 - strategie 1-3
 - tabulky typu RAW 9-24, 9-26
 - tabulky typu STANDARD 9-24
 - uvolnění souboru protokolu 14-6
 - změna protokolování databáze 10-8
 - zmenšení velikosti 9-11
- zámek latch
 - Viz též* objekty mutex.
 - fronta čekajících procesů 5-11
 - statistika monitorování 8-8
- zámky
 - definované 7-21
 - dynamické přidělování 7-13
 - fronta čekajících procesů 5-11
 - počáteční počet 7-13
 - příkaz onstat -k 24-14
 - typy 7-21
- zámky spin
 - monitorování 8-8

- zamykání
 - inteligentní velké objekty 9-13
 - prostory sbspace 9-17
 - úroveň izolace neaktualizované čtení 19-32
- zamykání rozsahu bajtů 7-13, 9-17
- zápis front LRU
 - definovaný 7-28
 - provádění 7-28
 - spuštění 7-29
- zápis na popředí
 - definovaný 7-28
 - monitorování 7-28, 8-11
 - předobraz 7-27
- zápisy front LRU
 - monitorování 8-11
- zařízení
 - NFS 9-3
 - v případě, že jsou potřeba posuny 10-4
 - znakově orientované 9-4
- zařízení systému UNIX
 - vlastnictví, oprávnění
 - předpřipravené soubory 10-5
 - znakově orientované 10-5
 - zobrazení propojení s názvem cesty 1-6, 10-5
- zásady cílové doby obnovy (RTO) 14-2
- zásady cílového bodu pro obnovu (RPO) 14-3
- zasílání zpráv na pager
 - oznamování alarmů událostí 2-8
- zásobník
 - a řídicí blok jednotkového procesu 5-10
 - definované 5-9
 - definovaný 7-18
 - jednotkového procesu 7-19
 - konfigurační parametr STACKSIZE 7-19
 - proměnná prostředí INFORMIXSTACKSIZE 7-19
 - ukazatel 5-10
 - velikost 2-6, 7-19
- zavaděč High Performance Loader 9-25
- zavádění dat
 - expresní režim 9-25
 - metody 10-46
 - obslužné programy 10-46
- záznam protokolu FREE_RE 10-45
- záznam protokolu CHKADJUP 10-26, 10-45
- záznam protokolu CHRESERV 10-26
- zdroje dat
 - vyhovující specifikaci XA 24-3
- zdroje dat typu XA 24-3
- změna
 - režim protokolování, databáze ANSI 12-3
 - stav bloku 20-11
 - typ databázového serveru, replikace HDR 20-17
- znakově orientovaná zařízení 1-5
- zprávy fyzické obnovy 15-2
- zrcadlení
 - alternativy 17-3
 - během spuštění systému 18-4
 - během zpracování 17-5
 - bloků v replikaci HDR 20-11
 - činnost během zpracování 17-5
 - činnost obnovy 17-4
 - definovaný 17-1
 - doporučené rozvržení disků 9-34
 - hardware 17-3
 - když zrcadlení skončí 17-6
 - když zrcadlení začíná 17-4
 - náklady 17-2
- zrcadlení (*pokračování*)
 - obnovení bloku 18-4
 - povolení 18-2
 - požadované kroky 18-1
 - přehled 1-20
 - rozdělené čtení 17-5
 - síťové omezení 17-1
 - spuštění 18-1, 18-3, 18-5, 18-6
 - stavový příznak 17-4
 - tabulka bloků 7-16
 - udržování souborů logického protokolu v prostoru dbspace 17-4
 - ukončení 18-7
 - výhody 17-2
 - výměna disků za chodu 17-3
 - vytvoření zrcadlených bloků 18-3
 - zastavení 18-7
 - zjišťování chyb média 17-5
 - změna stavu bloku 18-4
- zrcadlený blok
 - disková čtení 17-5
 - diskové zápisy 17-5
 - obnova 17-4, 17-5, 18-4
 - přidání 18-5
 - struktura 17-6
 - vytvoření 18-3
 - změna stavu 18-4
- zrcadlený prostor dbspace
 - kořenový prostor dbspace 9-11
 - vytvoření 10-6



Vytištěno v Dánsku společností IBM Danmark A/S.

G229-1392-00



Spine information:

IBM Informix Verze 11.1

Příručka administrátora serveru IBM Informix Dynamic Server

