



Příručka administrátora serveru IBM Informix Dynamic Server



Příručka administrátora serveru IBM Informix Dynamic Server

Poznámka

Než použijete tyto informace a popisovaný produkt, přečtěte si informace v části "Poznámky" na stránce B-1.

První vydání (prosinec 2004)

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

Pokud odesíláte IBM informace, udělujete tím IBM nevýluční právo tyto informace používat a distribuovat způsobem, jaký IBM uzná za vhodný, aniž by tím IBM vůči vám vznikl jakýkoli závazek.

© Copyright International Business Machines Corporation 1996, 2004. Všechna práva vyhrazena.

Obsah

Úvod	xix
Obsah příručky	xx
Druhy uživatelů	xx
Softwarové závislosti	xx
Předpoklady pro národní prostředí.	xx
Demonstrační databáze.	xxi
Nové vlastnosti serveru Dynamic Server verze 10.0	xxi
Vylepšení databázového serveru z hlediska zabezpečení a použitelnosti	xxi
Nové vlastnosti předchozích verzí serveru Dynamic Server.	xxiii
Vlastnosti serveru Dynamic Server verze 9.4	xxiii
Vlastnosti serveru Dynamic Server verze 9.3	xxiv
Vlastnosti verze 9.21 serveru Dynamic Server	xxv
Změny uspořádání této příručky ve verzi 9.2	xxvi
Konvence používané v dokumentaci	xxvi
Typografické konvence	xxvi
Vlastnost, produkt a platforma	xxvii
Diagramy syntaxe.	xxvii
Konvence použité v ukázkovém kódu	xxxix
Další dokumentace	xxxii
Příručky pro instalaci.	xxxii
Poznámky online	xxxii
Chybové zprávy systému Informix	xxxiv
Příručky.	xxxv
Nápověda online	xxxv
Usnadnění	xxxv
Sada dokumentace systému IBM Informix Dynamic Server verze 10.0 a sady CSDK verze 2.90	xxxv
Kompatibilita s oborovými standardy	xxxviii
IBM ocení vaší připomínky.	xxxix

Část 1. Databázový server

Kapitola 1. Instalace a konfigurace databázového serveru	1-1
Plánování databázového serveru	1-3
Posouzení priorit	1-3
Posouzení prostředí	1-4
Konfigurace operačního systému	1-4
Konfigurace paměti v systému Windows	1-4
Modifikace parametrů jádra systému UNIX.	1-5
Přidělení diskového prostoru	1-5
Použití velkých bloků	1-5
Vytvoření souborů bloků v systému UNIX.	1-6
Vytvoření oddílů NTFS ve Windows.	1-6
Nastavení oprávnění, vlastnictví a skupiny.	1-7
Vytvoření standardních názvů zařízení (UNIX)	1-7

Nastavení proměnných prostředí	1-8
Nastavení proměnných prostředí GLS	1-9
Nastavení proměnných prostředí v systému UNIX	1-9
Nastavení proměnných prostředí v systému Windows	1-10
Konfigurace propojitelnosti	1-11
Soubor sqlhosts v systému UNIX	1-11
Registr sqlhosts v systému Windows	1-12
Konfigurace propojitelnosti pomocí programu ISA	1-12
Konfigurace databázového serveru	1-12
Příprava konfiguračního souboru ONCONFIG	1-13
Použití funkce Nastavení serveru obslužného programu ISA k vlastnímu nastavení konfigurace.	1-14
Použití obslužného programu IBM Informix Server Administrator k aktualizaci souboru ONCONFIG.	1-14
Použití programu Instance Manager k vytvoření nové instance databázového serveru (Windows)	1-15
Použití programu Instance Manager k přejmenování instance databázového serveru (Windows).	1-15
Konfigurace podpory jazyka Java	1-16
Spuštění a správa databázového serveru	1-16
Spuštění databázového serveru a inicializace diskového prostoru	1-16
Příprava pro automatické spuštění	1-17
Příprava připojení k aplikacím	1-19
Vytvoření paměťových prostorů a bloků	1-19
Podpora velkých bloků	1-20
Nastavení systému zálohování a ukládání	1-20
Provedení rutinních úloh administrace	1-21
Změna režimů databázového serveru	1-21
Zálohování dat a souborů logického protokolu	1-21
Monitorování aktivity	1-22
Kontrola konzistence	1-22
Provádění dalších úloh administrace	1-22
Použití zrcadlení	1-22
Správa stavu protokolování databáze	1-22
Správa logického protokolu	1-22
Správa fyzického protokolu	1-23
Správa sdílené paměti	1-23
Správa virtuálních procesorů	1-23
Správa paralelních databázových dotazů	1-24
Použití replikace dat	1-24
Použití auditu	1-24
Použití distribuovaných dotazů	1-25
Monitorování aktivity databázového serveru	1-25
Alarmy událostí	1-26
Program IBM Informix Server Administrator (ISA)	1-26
Protokol zpráv	1-26
Obslužný program ON-Monitor (UNIX)	1-27
Obslužný program oncheck	1-27
Nástroj onperf (UNIX)	1-28
Obslužný program onstat	1-28
Tabulky SMI	1-28
Systémová konzole	1-28
Nástroje operačního systému UNIX	1-29
Prohlížeč událostí v systému Windows	1-29

Sledování výkonu v systému Windows	1-29
Obslužné programy pro systém Windows	1-29
Kapitola 2. Konfigurační parametry	2-1
Parametry identifikace databázového serveru	2-2
Parametry diskového prostoru.	2-2
Kořenový prostor dbspace	2-2
Zrcadlení kořenového prostoru dbspace	2-3
Další parametry správy prostorů	2-3
Parametry protokolování	2-4
Logický protokol	2-4
Parametry fyzického protokolu	2-5
Parametry odvolávání transakcí a zotavení	2-5
Parametry zálohování a obnovení	2-6
Parametry protokolu zpráv.	2-6
Parametry sdílené paměti	2-6
Velikost přidělené sdílené paměti.	2-7
Přidělený prostor sdílené paměti	2-7
Řízení vyrovnávací sdílené paměti	2-8
Využití rychlé vyrovnávací paměti příkazů SQL	2-9
Parametry podpory rozhodování	2-9
Parametry procesu databázového serveru	2-10
Parametry virtuálních procesorů.	2-10
Časové intervaly	2-11
Parametry obnovení	2-11
Parametry replikace High-Availability Data-Replication	2-11
Parametry alarmů událostí	2-12
Parametry vypisování jádra (systém UNIX)	2-13
Parametry direktiv	2-13
Parametry připojení	2-13
Parametry související se zabezpečením.	2-14
Zvláštní parametry.	2-14
Parametry prověřování (systém UNIX).	2-14
Parametry optických médií	2-14
Parametry systému UNIX	2-15
Monitorování informací o konfiguraci.	2-15
Kapitola 3. Komunikace mezi klientem a serverem	3-1
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	3-5
Připojení podporovaná databázovým serverem.	3-6
Místní připojení	3-7
Připojení prostřednictvím sdílené paměti (systém UNIX).	3-7
Proudové propojení procesů (systém UNIX)	3-8
Připojení prostřednictvím pojmenovaného propojení procesů (systém Windows).	3-8
Připojení prostřednictvím místní zpětné smyčky	3-9

Služby pro podporu komunikace	3-9
Propojovací soubory	3-10
Soubory konfigurace sítě	3-10
Soubory zabezpečení sítě	3-13
Soubor sqlhosts a klíč registru SQLHOSTS	3-15
Informace sqlhosts.	3-18
Informace o propojitelnosti	3-19
Informace o skupinách	3-30
Alternativy k připojení TCP/IP	3-30
Podpora adres IPv6 v serveru Dynamic Server	3-34
Parametry souboru ONCONFIG týkající se propojitelnosti	3-35
Konfigurační parametr DBSERVERNAME	3-36
Konfigurační parametr DBSERVERALIASES	3-36
Konfigurační parametr NETTYPE	3-37
Proměnné prostředí týkající se síťových připojení	3-37
Příklady konfigurací klientu a serveru	3-38
Připojení prostřednictvím sdílené paměti (systém UNIX)	3-38
Připojení prostřednictvím místní zpětné smyčky	3-39
Použití síťového připojení	3-40
Použití více typů připojení	3-41
Přístup k více databázovým serverům	3-42
Použití programu IBM Informix MaxConnect	3-43

Kapitola 4. Inicializace databázového serveru 4-1

Typy inicializace	4-2
Inicializace diskového prostoru	4-2
Jednotlivé kroky inicializace	4-3
Zpracování konfiguračního souboru	4-4
Vytvoření segmentů sdílené paměti	4-5
Inicializace sdílené paměti	4-5
Inicializace diskového prostoru	4-6
Spuštění všech požadovaných virtuálních procesorů	4-6
Provedení potřebných konverzí	4-6
Zahájení rychlé obnovy	4-6
Zahájení kontrolního bodu	4-6
Zaznamenání změn konfigurace	4-7
Vytvoření souboru oncfg_servername.servernum	4-7
Vypuštění dočasných prostorů tblspace	4-7
Nastavení vynucené rezidence, je-li určena	4-7
Návrat řízení uživateli	4-7
Vytvoření databáze sysmaster a příprava tabulek SMI	4-8
Vytvoření databáze sysutils	4-8
Monitorování maximálního počtu uživatelských připojení	4-8
Provozní režimy databázového serveru	4-9
Změna provozních režimů databázového serveru	4-10
Uživatelé oprávnění měnit režimy	4-10
Volby programu ISA pro změnu provozních režimů	4-11
Volby programu ON-Monitor pro změnu režimů (UNIX)	4-11
Volby příkazového řádku pro změnu režimů	4-12

Kapitola 5. Zabezpečení	5-1
Zabezpečení prováděné obslužnými programy serveru před spuštěním v operačním systému UNIX a Linux	5-2
Zakázání kontroly zabezpečení	5-3
Zabezpečení v nedostatečně chráněném prostředí	5-4
Varování a chybové zprávy pro kontrolu zabezpečení prováděnou obslužnými programy	5-4
Oprávnění adresáře INFORMIXDIR	5-5
Uživatelé a vlastnictví skupin pro spuštění obslužných programů serveru Dynamic Server	5-6
Audit událostí v databázi	5-7
Role uživatelů	5-7
Výchozí role	5-7
Omezení zahlcovacích útoků typu DOS (operační systém UNIX)	5-8
Použití příkazů LISTEN_TIMEOUT a MAX_INCOMPLETE_CONNECTIONS	5-8
Použití voleb onmode -wm a onmode -wf ke změně hodnot parametrů	5-9
Šifrování	5-9
Přehled šifrování a voleb šifrování	5-9
Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat	5-11
Použití šifrování na úrovni sloupců	5-23
Ověřovací moduly	5-27
Moduly PAM (Pluggable Authentication Modules) pro systémy spouštěné v operačním systému UNIX nebo Linux	5-27
Podpora ověřování LDAP v operačním systému Windows	5-30
Vývoj aplikací pro moduly ověřování	5-33
Distribuované transakce	5-35
Klientská rozhraní API a moduly na podporu ověřování	5-35
Problémy s kompatibilitou	5-36
Volby zabezpečení připojení replikace Enterprise Replication a replikace HDR	5-37
Zabezpečení externích uživatelských rutin (UDR)	5-38

Část 2. Správa disků, paměti a procesů

Kapitola 6. Virtuální procesory a jednotkové procesy	6-1
Virtuální procesory	6-2
Jednotkové procesy	6-3
Typy virtuálních procesorů	6-4
Výhody virtuálních procesorů	6-6
Obsluha jednotkových procesů virtuálními procesory	6-9
Řídící struktury	6-10
Přepínání kontextu	6-10
Zásobníky	6-12
Fronty	6-13
Objekty mutex	6-15
Třídy virtuálních procesorů	6-16
Virtuální procesory CPU	6-16
Uživatelem definované třídy virtuálních procesorů	6-19
Virtuální procesory Java	6-22
Virtuální procesory diskového vstupu - výstupu	6-22
Síťové virtuální procesory	6-26
Virtuální procesor Communications Support Module	6-33
Šifrovací virtuální procesory	6-33
Optický virtuální procesor	6-34

Virtuální procesor typu audit.	6-34
Virtuální procesor typu miscellaneous	6-34
Kapitola 7. Správa virtuálních procesorů	7-1
Nastavení konfiguračních parametrů virtuálních procesorů	7-1
Nastavení parametrů virtuálního procesoru pomocí textového editoru	7-2
Nastavení parametrů virtuálního procesoru pomocí programu ISA	7-3
Nastavení parametrů virtuálního procesoru pomocí obslužného programu ON-Monitor	7-3
Spuštění a zastavení virtuálních procesorů	7-3
Přidání virtuálních procesorů v režimu online	7-4
Vypuštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů	7-5
Monitorování virtuálních procesorů	7-6
Monitorování virtuálních procesorů pomocí obslužných programů příkazového řádku	7-6
Monitorování virtuálních procesorů pomocí tabulek SMI.	7-8
Kapitola 8. Sdílená paměť	8-1
Sdílená paměť	8-2
Použití sdílené paměti	8-3
Přidělování sdílené paměti	8-4
Velikost sdílené paměti.	8-6
Jaké akce provést, pokud je překročena hodnota parametru SHMTOTAL	8-7
Procesy, které se připojují ke sdílené paměti	8-8
Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)	8-8
Jak se ke sdílené paměti připojují obslužné programy	8-9
Jak se ke sdílené paměti připojují virtuální procesory	8-9
Rezidentní segmenty sdílené paměti	8-12
Rezidentní část sdílené paměti	8-13
Záhlaví sdílené paměti	8-13
Společná oblast vyrovnávacích paměti ve sdílené paměti	8-13
Vyrovnávací paměť logického protokolu	8-15
Vyrovnávací paměť fyzického protokolu	8-16
Vyrovnávací paměť replikace High-Availability Data Replication	8-17
Tabulka zámků.	8-17
Virtuální část sdílené paměti	8-18
Správa virtuální části sdílené paměti	8-18
Součásti virtuální části sdílené paměti	8-19
Rychlá vyrovnávací paměť distribuce dat	8-24
Komunikační část sdílené paměti (systém UNIX)	8-26
Část virtuálních rozšíření sdílené paměti	8-27
Řízení souběžného přístupu	8-27
Objekty mutex sdílené paměti	8-27
Zámky vyrovnávacích paměti ve sdílené paměti	8-27
Přístup jednotkových procesů databázového serveru k sdíleným vyrovnávacím pamětím.	8-28
Fronty FIFO/LRU	8-28
Konfigurace dopředného čtení databázového serveru	8-33
Přístup jednotkových procesů databázového serveru ke stránkám ve vyrovnávací paměti	8-33
Vyprazdňování dat na disk	8-34
Vyprazdňování vyrovnávacích paměti ve společné oblasti	8-34
Přednostní vyprazdňování předobrazů	8-35
Vyprazdňování vyrovnávací paměti fyzického protokolu	8-35

Synchronizace vyprazdňování vyrovnávacích pamětí	8-35
Popis aktivity vyprazdňování	8-36
Vyprazdňování vyrovnávací paměti logického protokolu	8-37
Použití vyrovnávací paměti pro velké objekty.	8-38
Zápis jednoduchých velkých objektů	8-39
Přístup k inteligentním velkým objektům	8-41
Využití paměti na 64bitových platformách	8-42

Kapitola 9. Správa sdílené paměti. 9-1

Nastavení konfiguračních parametrů sdílené paměti operačního systému	9-2
Maximální velikost segmentu sdílené paměti	9-2
Adresa spodní hranice sdílené paměti (systém Windows).	9-3
Semaforey (systém UNIX)	9-4
Nastavení konfiguračních parametrů sdílené paměti databázového serveru	9-4
Nastavení parametrů pro rezidentní sdílenou paměť	9-4
Nastavení parametrů pro virtuální sdílenou paměť	9-5
Nastavení parametrů pro výkon sdílené paměti.	9-6
Nastavení parametrů sdílené paměti pomocí textového editoru	9-6
Nastavení parametrů sdílené paměti pomocí programu ISA	9-6
Nastavení parametrů sdílené paměti pomocí programu ON-Monitor (v systému UNIX)	9-6
Nastavení parametrů rychlé vyrovnávací paměti příkazů jazyka SQL	9-7
Opětovná inicializace sdílené paměti.	9-8
Vypnutí nebo zapnutí rezidence rezidentní sdílené paměti	9-8
Zapínání a vypínání rezidence v režimu online	9-8
Zapnutí a vypnutí rezidence při restartování databázového serveru.	9-9
Přidání segmentu k virtuální části sdílené paměti	9-9
Monitorování sdílené paměti	9-9
Monitorování segmentů sdílené paměti	9-9
Monitorování profilu sdílené paměti a zámků latch	9-10
Monitorování vyrovnávacích pamětí	9-11
Monitorování aktivity společné oblasti vyrovnávacích pamětí	9-13

Kapitola 10. Uložení dat. 10-1

Fyzické a logické paměťové jednotky	10-2
Bloky	10-3
Přidělování disků blokům	10-4
Posuny	10-6
Stránky	10-6
Stránky blobpage	10-7
Stránky sbpage	10-9
Oblasti	10-10
Prostory dbspace	10-11
Řízení umístění ukládaných dat	10-11
Kořenový prostor dbspace	10-13
Dočasné prostory dbspace	10-14
Prostory blobspace	10-15
Prostory sbpace	10-16
Výhody používání prostorů sbpace	10-16
Prostory sbpace a replikace Enterprise Replication	10-16
Metadata, uživatelská data a rezervovaná oblast	10-17

Řízení umístění ukládaných dat	10-17
Paměťové charakteristiky prostorů sbspace	10-19
Úrovně dědičnosti charakteristik prostorů sbspace	10-22
Další informace o prostorech sbspace	10-23
Dočasné prostory sbspace	10-25
Porovnání dočasných a standardních prostorů sbspace.	10-25
Dočasné inteligentní velké objekty	10-26
Prostory extspace.	10-27
Databáze	10-28
Tabulky	10-29
Typy tabulek serveru Dynamic Server	10-30
Standardní trvalé tabulky	10-31
Tabulky typu RAW	10-31
Tabulky typu TEMP	10-32
Vlastnosti typů tabulek	10-32
Dočasné tabulky	10-33
Prostory tblspace	10-36
Maximální počet prostorů tblspace v tabulce	10-36
Prostory tblspace tabulek a indexů	10-37
Prokládání oblastí	10-38
Fragmentace tabulek a uložení dat	10-38
Objem diskového prostoru potřebný k uložení dat	10-40
Velikost kořenového prostoru dbspace	10-40
Objem prostoru požadovaného databázemi	10-42
Pravidla pro rozvržení disků	10-42
Pravidla pro prostory dbspace a bloky	10-42
Pravidla umisťování tabulek	10-44
Ukázková rozvržení disku	10-44
Ukázkové rozvržení upřednostňující výkon	10-45
Ukázkové rozvržení upřednostňující dostupnost.	10-46
Správce logických disků	10-48
Kapitola 11. Správa diskového prostoru	11-1
Přidělení diskového prostoru.	11-3
Určení posunu	11-4
Přidělení prostorů předpřipravených souborů v systému UNIX	11-5
Přidělení diskového prostoru s přímým přístupem v systému UNIX	11-6
Vytvoření symbolických propojení k přímým zařízením (UNIX)	11-6
Přidělení prostoru souborům systému NTFS v systému Windows.	11-7
Přidělení diskového prostoru s přímým přístupem v systému Windows	11-8
Určení názvů paměťových prostorů a bloků	11-9
Určení maximální velikosti bloků	11-9
Určení maximálního počtu bloků a paměťových prostorů	11-9
Zálohování po změně fyzického schématu	11-10
Správa prostorů dbspace	11-10
Vytvoření prostoru dbspace, který používá výchozí velikost stránky.	11-10
Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky	11-14
Vytvoření dočasného prostoru dbspace	11-19
Co dělat, pokud dojde k vyčerpání diskového prostoru	11-20
Přidání bloku do prostoru dbspace nebo blobspace	11-20

Přejmenování prostorů dbspace	11-22
Správa oddílů prostorů dbspace	11-23
Správa prostorů blobspace	11-24
Vytvoření prostoru blobspace	11-24
Příprava prostorů blobspace k ukládání dat typu TEXT a BYTE	11-26
Určení velikosti stránky blobpage	11-26
Správa prostorů sbspace	11-28
Vytvoření prostoru sbspace	11-28
Určení velikosti oblasti metadat prostoru sbspace	11-29
Přidání bloku do prostoru sbspace	11-29
Změna paměťových charakteristik inteligentních velkých objektů	11-30
Vytvoření dočasného prostoru sbspace	11-31
Vypuštění bloku	11-32
Ověření, zda je blok prázdný	11-32
Vypuštění bloku z prostoru dbspace pomocí obslužného programu onspaces	11-33
Vypuštění bloku z prostoru blobspace	11-33
Vypuštění bloku z prostoru sbspace příkazem onspaces	11-33
Vypuštění paměťového prostoru	11-34
Příprava na vypuštění paměťového prostoru	11-34
Vypuštění zrcadlového paměťového prostoru	11-35
Vypuštění paměťového prostoru s obslužným programem onspaces	11-35
Vypuštění prostorů dbspace a blobspace pomocí programu ON-Monitor (UNIX)	11-35
Zálohování po vypuštění paměťového prostoru	11-36
Správa prostorů extspace	11-36
Vytvoření prostoru extspace	11-36
Vypuštění prostoru extspace	11-37
Přeskakování nepřístupných fragmentů	11-37
Použití konfiguračního parametru DATASKIP	11-37
Použití funkce Dataskip obslužného programu onspaces	11-37
Použití obslužného programu onstat ke kontrole stavu funkce dataskip	11-38
Použití příkazu SET DATASKIP jazyka SQL	11-38
Vliv funkce dataskip na transakce	11-38
Jak určit, kdy používat parametr dataskip	11-39
Monitorování využití fragmentace	11-40
Zobrazení databázi	11-40
Použití tabulek SMI	11-40
Použití programu ISA	11-40
Použití programu ON-Monitor (systém UNIX)	11-41
Monitorování využití disku	11-41
Monitorování bloků	11-41
Monitorování prostorů tblspace a oblastí	11-47
Monitorování jednoduchých velkých objektů v prostoru blobspace	11-47
Monitorování prostorů sbspace	11-51
Zavedení dat do tabulky	11-58

Část 3. Protokolování a administrace protokolů

Kapitola 12. Protokolování	12-1
Procesy databázového serveru, které vyžadují protokolování	12-1
Protokolování transakcí	12-3

Protokolování příkazů jazyka SQL a aktivity databázového serveru	12-4
Aktivita, která se vždy protokoluje	12-4
Aktivity protokolované u databázi s protokolováním transakcí	12-5
Aktivita, která se neprotokoluje	12-6
Stav protokolování databáze	12-6
Protokolování transakcí bez vyrovnávací paměti	12-7
Protokolování transakcí s vyrovnávací paměti	12-8
Protokolování transakcí kompatibilní se standardem ANSI	12-8
Databáze bez protokolování	12-8
Databáze s různými stavy používání vyrovnávací paměti pro protokolování	12-9
Protokolování databáze v prostředí X/Open DTP	12-9
Nastavení nebo změny stavu nebo režimu protokolování	12-9
Kapitola 13. Správa režimu protokolování databáze	13-1
Změna režimu protokolování databáze	13-2
Změna režimu protokolování databáze pomocí obslužného programu ondblog	13-3
Změna režimu protokolování pomocí obslužného programu ondblog	13-3
Zrušení změny režimu protokolování pomocí obslužného programu ondblog	13-3
Ukončení protokolování pomocí obslužného příkazu ondblog	13-3
Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ondblog	13-4
Změna režimu protokolování databáze kompatibilní se standardem ANSI	13-4
Změna režimu protokolování databáze pomocí obslužného programu ontape	13-4
Zapnutí protokolování transakcí pomocí obslužného programu ontape	13-4
Ukončení protokolování pomocí obslužného programu ontape	13-5
Změna režimu protokolování pomocí obslužného programu ontape	13-5
Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ontape	13-5
Změna režimu protokolování databáze pomocí programu ISA	13-6
Změna režimu protokolování pomocí programu ON-Monitor (UNIX)	13-6
Změna režimu protokolování tabulek	13-6
Změna tabulky za účelem vypnutí protokolování	13-6
Změna tabulky za účelem zapnutí protokolování	13-6
Monitorování transakcí	13-6
Monitorování režimu protokolování databáze	13-7
Monitorování režimu protokolování u tabulek typu SMI	13-7
Monitorování režimu protokolování pomocí programu ON-Monitor (UNIX)	13-7
Monitorování režimu protokolování pomocí programu ISA	13-7
Kapitola 14. Logický protokol	14-1
Co je logický protokol?	14-1
Umístění souborů logického protokolu	14-2
Identifikace souborů logického protokolu	14-3
Příznaky stavu souborů logického protokolu	14-3
Velikost logického protokolu	14-4
Počet souborů logického protokolu	14-5
Pokyny týkající se výkonu	14-5
Dynamické přidělování protokolů	14-6
Uvolnění souborů logického protokolu	14-7
Akce v případě, že další soubor logického protokolu není volný	14-7
Akce v případě, že následující logický soubor obsahuje poslední kontrolní bod	14-7
Protokolování prostorů blobspace a jednoduché velké objekty	14-8

Přepínání souborů protokolu za účelem aktivace prostorů blobospace	14-8
Zálohování souborů protokolů za účelem uvolnění stránek blobpage	14-9
Zálohování prostorů blobospace po vložení nebo odstranění dat typu TEXT a BYTE	14-9
Protokolování prostorů sbpace a inteligentních velkých objektů	14-9
Používání protokolování prostoru sbpace	14-9
Používání záznamů protokolu inteligentních velkých objektů	14-11
Zabránění dlouhým transakcím při protokolování dat inteligentních velkých objektů	14-12
Proces protokolování	14-12
Protokolování prostorů dbspace	14-12
Protokolování prostorů blobospace	14-12

Kapitola 15. Správa souborů logických protokolů. 15-1

Odhad velikosti a počtu souborů protokolu	15-2
Odhad velikosti protokolu při protokolování inteligentních velkých objektů	15-3
Odhad počtu souborů logického protokolu.	15-3
Zálohování logických souborů	15-4
Zálohování prostorů blobospace	15-4
Zálohování prostorů sbpace	15-4
Prepnutí na následující soubor logického protokolu	15-5
Uvolnění souboru logického protokolu	15-5
Odstranění souboru protokolu se stavem D	15-6
Uvolnění souboru protokolu se stavem U	15-6
Uvolnění souboru protokolu se stavem U-B nebo F	15-6
Uvolnění souboru protokolu se stavem U-C nebo U-C-L	15-7
Uvolnění souboru protokolu se stavem U-B-L	15-7
Monitorování aktivity protokolování	15-8
Monitorování logického protokolu za účelem zjištění jeho zaplnění	15-8
Monitorování dočasných logických protokolů.	15-9
Používání tabulek typu SMI	15-10
Použití programu ON-Monitor (operační systém UNIX)	15-10
Monitorování stavu zálohy protokolů	15-10
Přidělení souborů protokolu	15-10
Dynamické přidávání protokolů	15-10
Ruční přidání souborů logického protokolu	15-12
Vypuštění souborů logického protokolu	15-14
Změna velikosti souborů logického protokolu	15-15
Přesun souboru logického protokolu do jiného prostoru dbspace	15-16
Změna konfiguračních parametrů protokolování	15-17
Použití programu ON-Monitor ke změně parametru LOGFILES (operační systém UNIX)	15-18
Zobrazení záznamů logického protokolu	15-18
Monitorování událostí dynamického přidávání protokolů	15-18
Nastavení horních mezí pro odvolání dlouhých transakcí	15-20
Horní mez dlouhé transakce (LTXHWM)	15-20
Horní mez dlouhé transakce s výlučným přístupem (LTXEHW)	15-21
Úprava velikosti souborů protokolu za účelem zabránění dlouhých transakcí	15-21
Zotavení ze zablokované dlouhé transakce	15-21

Kapitola 16. Fyzické protokolování, kontrolní body a rychlá obnova 16-1

Kritická část	16-2
Fyzické protokolování	16-2

Jak rychlá obnova používá fyzicky protokolované stránky	16-3
Jak zálohování používá fyzicky protokolované stránky	16-3
Aktivity databázového serveru, které jsou fyzicky protokolovány	16-3
Velikost a umístění fyzického protokolu	16-4
Určení umístění fyzického protokolu	16-4
Odhad velikosti fyzického protokolu	16-5
Konfigurace velikosti fyzického protokolu	16-6
Podrobnosti fyzického protokolování	16-7
Kontrolní body	16-8
Úplný kontrolní bod	16-9
Kontrolní bod typu fuzzy	16-9
Události, které vyvolají kontrolní bod typu fuzzy	16-11
Události, které vyvolají úplný kontrolní bod	16-11
Pořadí událostí v kontrolním bodu	16-12
Pokyny týkající se zálohování a obnovy	16-13
Rychlá obnova	16-14
Potřeba rychlé obnovy	16-14
Situace, při kterých se spouští rychlá obnova	16-14
Podrobnosti rychlé obnovy po úplném kontrolním bodu	16-15
Podrobnosti rychlé obnovy po kontrolním bodu typu fuzzy	16-17
Alternativní volby obnovy při rychlém restartování pro operace typu fuzzy	16-21

Kapitola 17. Správa fyzického protokolu 17-1

Změna umístění a velikosti fyzického protokolu	17-1
Příprava na provedení změn	17-2
Kontrola odpovídajícího souvislého prostoru	17-2
Použití textového editoru ke změně umístění a velikosti fyzického protokolu	17-3
Použití obslužného programu onparams ke změně umístění nebo velikosti fyzického protokolu	17-3
Použití programu ON-Monitor ke změně umístění a velikosti fyzického protokolu	17-3
Monitorování aktivity fyzického a logického protokolování	17-4
Vzorový výstup příkazu onstat -l	17-5
Monitorování informací o kontrolních bodech	17-5
Vynucení úplného kontrolního bodu	17-6
Vynucení kontrolního bodu typu fuzzy	17-7
Použití tabulek SMI	17-7

Část 4. Odolnost vůči selhání

Kapitola 18. Zrcadlení 18-1

Zrcadlení	18-1
Výhody zrcadlení	18-2
Požadavky na zrcadlení	18-2
Důsledky nepoužívání zrcadlení	18-2
Data vhodná k zrcadlení	18-3
Alternativy k zrcadlení	18-3
Průběh zrcadlení	18-4
Vytvoření zrcadleného bloku	18-4
Stavové příznaky zrcadlení	18-5
Zotavení	18-5

Činnosti v průběhu zpracování	18-6
Výsledek zastavení zrcadlení	18-7
Struktura zrcadleného bloku	18-7
Kapitola 19. Použití zrcadlení	19-1
Příprava k zrcadlení dat	19-2
Povolení konfiguračního parametru MIRROR	19-2
Změna parametru MIRROR pomocí programu ON-Monitor (UNIX)	19-3
Přidělení diskového prostoru pro zrcadlená data	19-3
Propojení bloků (UNIX)	19-3
Opětovné propojení bloku se zařízením po poruše disku	19-3
Použití zrcadlení	19-4
Zrcadlení kořenového prostoru dbspace během spuštění	19-4
Změna stavu zrcadlení	19-5
Správa zrcadlení	19-5
Spuštění zrcadlení u nezrcadlených paměťových prostorů	19-5
Spuštění zrcadlení nových paměťových prostorů	19-6
Přidání zrcadlených bloků	19-7
Vypnutí zrcadleného bloku	19-7
Obnovení zrcadleného bloku	19-7
Ukončení zrcadlení	19-8
Kapitola 20. Replikace HDR (High-Availability Data Replication)	20-1
Replikace High-Availability Data Replication	20-2
Režimy replikace HDR	20-3
Typy replikovaných dat	20-3
Výhody replikace dat	20-3
Jak pracuje replikace HDR	20-7
Průběh prvotní replikace dat	20-7
Reprodukce aktualizací primárního databázového serveru	20-8
Jednotkové procesy, které zpracovávají replikaci HDR	20-11
Kontrolní body mezi databázovými servery	20-12
Jak je zaznamenávána synchronizace dat	20-12
Selhání replikace HDR	20-13
Definice selhání replikace HDR	20-13
Zjišťování selhání replikace HDR	20-13
Akce, které je třeba provést při selhání replikace HDR	20-14
Co provést v případě selhání replikace HDR	20-14
Přesměrování a propojitelnost pro klienty replikace dat	20-17
Návrh klientů pro přesměrování	20-17
Automatické přesměrování klientů pomocí proměnné prostředí DBPATH	20-18
Směrování klientů pomocí informací o propojitelnosti	20-19
Směrování klientů pomocí proměnné prostředí INFORMIXSERVER	20-22
Přesměrování z aplikace	20-23
Porovnání různých mechanismů přesměrování	20-24
Návrh klientů replikace HDR	20-25
Nastavení režimu uzamykání při přístupu k primárnímu databázovému serveru na čekání	20-25
Návrh klientů k použití sekundárního databázového serveru	20-26
Kapitola 21. Použití replikace HDR	21-1

Plánování replikace HDR	21-2
Konfigurace systému pro replikaci HDR	21-3
Splnění požadavků na hardware a operační systém	21-3
Splnění požadavků na databázi a data	21-3
Splnění požadavků na konfiguraci databázového serveru	21-4
Konfigurace propojitelnosti replikace HDR	21-6
První spuštění replikace HDR	21-7
Zkrácení doby nastavení alternativní metodou zálohování	21-11
Provádění základních úloh administrace	21-11
Změna konfiguračních parametrů databázového serveru	21-11
Zálohování paměťových prostorů a souborů logického protokolu	21-11
Změna režimu protokolování databázi	21-12
Přidání a vypuštění bloků a paměťových prostorů	21-12
Přejmenování bloků	21-12
Uložení stavu bloku na sekundární databázový server	21-13
Použití a změna zrcadlení bloků	21-14
Správa fyzického protokolu	21-14
Správa logického protokolu	21-15
Správa virtuálních procesorů	21-15
Správa sdílené paměti	21-15
Replikace indexu sekundárního databázového serveru	21-15
Změna režimu databázového serveru	21-16
Změna typu databázového serveru	21-18
Monitorování stavu replikace HDR	21-18
Použití obslužných programů příkazového řádku	21-18
Použití tabulek SMI	21-20
Použití programu ON-Monitor (v systému UNIX)	21-20
Obnovení dat po selhání prostředku	21-20
Obnovení po selhání prostředku na primárním databázovém serveru	21-20
Obnovení po selhání prostředku na sekundárním databázovém serveru	21-21
Replikace indexu na sekundární server	21-22
Restartování replikace HDR po selhání	21-22
Restartování po poškození kritických dat	21-23
Restartování bez poškození kritických dat	21-24
Kapitola 22. Kontrola konzistence	22-1
Provádění periodické kontroly konzistence	22-2
Ověření konzistence	22-2
Monitorování nekonzistence dat	22-4
Zachování konzistentních záloh úrovně 0	22-6
Co provést v případě poškození	22-6
Zjištění příznaků poškození	22-6
Oprava poškození indexu	22-6
Oprava chyb vstupu-výstupu bloku	22-7
Shromažďování diagnostických dat	22-8
Chyby vstupu-výstupu bránící v provozu	22-9
Monitorování chyb vstupu - výstupu bránících v provozu v databázovém serveru	22-9
Používání protokolu zpráv ke sledování chyb vstupu-výstupu bránících v provozu	22-10
Používání alarmů událostí k monitorování chyb vstupu-výstupu bránících v provozu	22-10
Používání vypnutého mapování chybných sektorů	22-11

Část 5. Distribuovaná data

Kapitola 23. Protokoly vícefázového potvrzování	23-1
Správci transakcí	23-2
Použití knihovny TP/XA se správcem transakcí	23-2
Použití serveru MTS/XA (Microsoft Transaction Server)	23-3
Použití volně vázaného režimu a provázaného režimu	23-3
Protokol dvoufázového potvrzování	23-4
Kdy se používá protokol dvoufázového potvrzování	23-4
Koncepce dvoufázového potvrzování	23-5
Fáze protokolu dvoufázového potvrzování	23-6
Jak protokol dvoufázového potvrzování ošetřuje selhání	23-6
Optimalizace předpokládaného přerušení	23-7
Nezávislé akce	23-8
Situace, které vyvolávají nezávislé akce	23-8
Možné důsledky nezávislých akcí	23-8
Scénář heuristického odvolání transakce	23-10
Scénář heuristického ukončení transakce	23-13
Monitorování globální transakce	23-15
Chyby protokolu dvoufázového potvrzování	23-15
Dvoufázové potvrzování a záznamy logického protokolu	23-16
Záznamy logického protokolu při potvrzení transakce	23-16
Záznamy logického protokolu zapsané během heuristického odvolání transakce	23-18
Záznamy logického protokolu zapsané během heuristického ukončení transakce	23-19
Konfigurační parametry používané při dvoufázovém potvrzování	23-21
Funkce parametru DEADLOCK_TIMEOUT	23-21
Funkce parametru TXTIMEOUT	23-21
Protokol heterogenního potvrzování	23-21
Brány, které se mohou účastnit heterogenního potvrzování transakcí	23-22
Povolení a zakázání heterogenního potvrzování	23-23
Jak funguje heterogenní potvrzování	23-23
Důsledky selhání heterogenního potvrzování	23-25
Kapitola 24. Ruční obnova při selhání dvoufázového potvrzování	24-1
Určení, zda je požadována ruční obnova	24-1
Určení, zda byla transakce implementována nekonzistentně	24-1
Určení, zda distribuovaná databáze obsahuje nekonzistentní data	24-3
Rozhodnutí, zda je nezbytné k nápravě situace provést akci	24-5
Příklad ruční obnovy	24-6

Část 6. Dodatky a přílohy

Dodatek. Usnadnění	A-1
Poznámky	B-1
Rejstřík	X-1

Úvod

Obsah příručky	xx
Druhy uživatelů	xx
Softwarové závislosti	xx
Předpoklady pro národní prostředí.	xx
Demonstrační databáze.	xxi
Nové vlastnosti serveru Dynamic Server verze 10.0	xxi
Vylepšení databázového serveru z hlediska zabezpečení a použitelnosti	xxi
Nové vlastnosti předchozích verzí serveru Dynamic Server.	xxiii
Vlastnosti serveru Dynamic Server verze 9.4	xxiii
Vylepšení databázového serveru z hlediska použitelnosti	xxiii
Posílení bezpečnosti	xxiv
Vlastnosti serveru Dynamic Server verze 9.3	xxiv
Vylepšení výkonu	xxv
Vylepšení jazyka SQL	xxv
Další významné změny ve verzi 9.3	xxv
Vlastnosti verze 9.21 serveru Dynamic Server	xxv
Změny uspořádání této příručky ve verzi 9.2	xxvi
Konvence používané v dokumentaci	xxvi
Typografické konvence	xxvi
Vlastnost, produkt a platforma	xxvii
Diagramy syntaxe.	xxvii
Jak číst diagramy syntaxe příkazového řádku	xxix
Klíčová slova a interpunkce	xxx
Identifikátory a názvy	xxxi
Konvence použité v ukázkovém kódu	xxxi
Další dokumentace	xxxii
Příručky pro instalaci.	xxxii
Poznámky online	xxxii
Nalezení poznámek online	xxxiii
Názvy souborů poznámek online	xxxiv
Chybové zprávy systému Informix	xxxiv
Příručky.	xxxv
Příručky online	xxxv
Tištěné příručky	xxxv
Nápověda online	xxxv
Usnadnění	xxxv
Sada dokumentace systému IBM Informix Dynamic Server verze 10.0 a sady CSDK verze 2.90	xxxv
Kompatibilita s oborovými standardy	xxxviii
IBM ocení veškeré připomínky.	xxxix

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 s třídami 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 *IBM Informix: Dynamic Server Performance Guide*.

Tato část popisuje členění příručky, skupiny uživatelů, pro které je určena a přidružené softwarové produkty potřebné k používání databázového serveru.

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 *IBM Informix: Úvodní příručka Dynamic Server* databázového serveru.

Softwarové závislosti

Tato příručka je určena uživatelům serveru IBM Informix Dynamic Server verze 10.0.

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 lze použít k instalaci demonstračních databází jsou uloženy v adresáři **\$INFORMIXDIR/bin** v systému UNIX nebo v adresáři **%INFORMIXDIR%\bin** v systému Windows.

Nové vlastnosti serveru Dynamic Server verze 10.0

Následující tabulka poskytuje informace o nových vlastnostech serveru IBM Informix Dynamic Server verze 10.0, které popisuje tato příručka. Popis všech nových vlastností naleznete v příručce *IBM Informix: Úvodní příručka*.

Vylepšení databázového serveru z hlediska zabezpečení a použitelnosti

Verze 10.0 databázového serveru obsahuje nové vlastnosti, které vylepšují funkčnost databázového serveru a usnadňují jeho používání.

Nové vlastnosti	Odkaz
<p>Nová kapitola o zabezpečení obsahuje informace o nových vlastnostech týkajících se zabezpečení. Dále byly informace o modulech pro podporu komunikace (CSM) přesunuty z kapitoly 3 do nové kapitoly o zabezpečení s číslem 5.</p> <p>Kapitoly 5 až 23 předchozích verzí příručky mají nyní čísla 6 až 24.</p>	<p>Kapitola 5, “Zabezpečení”</p>
<p>V systémech UNIX a Linux kontrolují obslužné programy databázového serveru před spuštěním databázového serveru, zda je prostředí zabezpečené.</p>	<p>“Zabezpečení prováděné obslužnými programy serveru před spuštěním v operačním systému UNIX a Linux” na stránce 5-2</p>
<p>Server poskytuje novou výchozí roli, kterou můžete přiřadit jednotlivým uživatelům nebo skupině PUBLIC konkrétní databáze. Výchozí role bude automaticky použita, pokud se uživatel připojí k databázi.</p>	<p>“Role uživatelů” na stránce 5-7</p>
<p>Nové konfigurační parametry umožňují zkrátit časový limit pro neúplná připojení, omezit počet nedokončených požadavků na připojení a omezit tak riziko agresivního útoku typu DoS (Denial of Service).</p>	<p>“Omezení zahlcovacích útoků typu DOS (operační systém UNIX)” na stránce 5-8</p>
<p>Nové volby umožňují implementovat šifrování na úrovni sloupců, šifrování hesel a šifrování přenosu dat.</p>	<p>“Šifrování” na stránce 5-9</p>
<p>Nový konfigurační parametr umožňuje administrátorovi zabezpečení databázového serveru omezit skupinu uživatelů, kteří mohou registrovat externí rutiny.</p>	<p>“Zabezpečení externích uživatelských rutin (UDR)” na stránce 5-38</p>
<p>Nový parametr alarmů událostí umožňuje nastavit program alarmů událostí tak, aby zpracovával všechny události zaznamenávané do protokolu MSGPATH a nikoli jen významné události.</p>	<p>“Parametry alarmů událostí” na stránce 2-12</p>
<p>Funkce přejmenování prostoru dbspace.</p>	<p>“Přejmenování prostorů dbspace” na stránce 11-22</p>
<p>Jednouživatelský režim - nový režim databázového serveru, který lze používat k úlohám administrace s použitím příkazů jazyků SQL a DDL.</p>	<p>“Provozní režimy databázového serveru” na stránce 4-9 a “Změna provozních režimů databázového serveru” na stránce 4-10</p>
<p>Konfigurační parametr DRAUTO umožňuje automatizovat přepnutí databázových serverů replikace HDR, pokud dojde k selhání primárního serveru.</p>	<p>“Akce, které je třeba provést při selhání replikace HDR” na stránce 20-14, “Postup v případě selhání primárního databázového serveru” na stránce 20-15, “Změna režimu databázového serveru” na stránce 21-16, “Selhání kritického prostředku na primárním databázovém serveru” na stránce 21-23 a “Sekundární databázový server je automaticky změněn na standardní databázový server” na stránce 21-27</p>
<p>Funkce externího zálohování a obnovení při nastavování replikace HDR.</p>	<p>“První spuštění replikace HDR” na stránce 21-7</p>

Nové vlastnosti	Odkaz
Dva konfigurační parametry umožňují databázovému serveru při zotavení ve fázi přehrání žurnálu provádět fyzické protokolování v kontrolním bodu typu fuzzy a zkrátit tak dobu potřebnou k zotavení.	“Alternativní volby obnovy při rychlém restartování pro operace typu fuzzy” na stránce 16-21
Konfigurační parametr DS_NONPDQ_QUERY_MEM umožňuje zvětšit množství paměti dostupné k řazení při dotazech jiných než paralelní dotazy (PDQ).	“Velikost virtuální části sdílené paměti” na stránce 8-19 a příručka <i>IBM Informix: Dynamic Server Performance Guide</i>
Funkce ruční nebo automatické replikace indexů replikace HDR do sekundárního serveru, pokud dojde k poškození indexu v sekundárním serveru.	“Replikace indexu na sekundární server” na stránce 21-22
Funkce vytvoření více oddílů tabulky nebo indexu v jediném prostoru dbspace v případě fragmentovaných tabulek, které používají schéma distribuce založené na výrazu nebo typu cyklická obsluha.	“Správa oddílů prostorů dbspace” na stránce 11-23
Nové konfigurační parametry a volby obslužného programu onspace k určení velikosti první oblasti a dalších oblastí prostoru tblspace v prostoru dbspace.	“Určení velikosti první oblasti a dalších oblastí prostoru tblspace ” na stránce 11-12
Funkce k určení velikosti stránky standardního nebo dočasného prostoru dbspace při vytváření prostoru dbspace a konfigurační parametr BUFFERPOOL, pomocí kterého můžete určit společnou oblast vyrovnávacích pamětí. Informace, které byly před verzí 10.0 zadávány pomocí konfiguračních parametrů BUFFERS, LRUS, LRU_MAX_DIRTY a LRU_MIN_DIRTY se nyní zadávají pomocí konfiguračního parametru BUFFERPOOL.	“Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky” na stránce 11-14
Funkce přejmenování instance databázového serveru.	“Použití programu Instance Manager k přejmenování instance databázového serveru (Windows)” na stránce 1-15
Podpora protokolu IPv6.	“Podpora adres IPv6 v serveru Dynamic Server” na stránce 3-34

Nové vlastnosti předchozích verzí serveru Dynamic Server

Vlastnosti serveru Dynamic Server verze 9.4

Následující tabulka poskytuje informace o nových vlastnostech serveru IBM Informix Dynamic Server verze 9.4, které popisuje tato příručka. Popis všech nových vlastností naleznete v příručce *IBM Informix: Úvodní příručka Dynamic Server*.

Vylepšení databázového serveru z hlediska použitelnosti

Verze 9.4 zahrnuje nové vlastnosti, které usnadňují instalaci, používání a správu databázového serveru.

Nové vlastnosti	Odkaz
Schopnost rychleji čistit vyrovnávací paměti díky možnosti zadávat konfigurační parametry front LRU jako desetinná čísla.	“Počet stránek přidávaných do front MLRU” na stránce 8-31.
Schopnost odesílat upozornění na události a jejich závažnost na e-mail a pager.	“Parametry alarmů událostí” na stránce 2-12. “Dodatek C: Alarmy událostí” v příručce Administrator’s Reference
Schopnost používat bloky a oblasti o velikosti až 4 TB.	“Použití velkých bloků” na stránce 1-5. “Vytvoření paměťových prostorů a bloků” na stránce 1-19. “Podpora velkých bloků” na stránce 1-20. “Bloky” na stránce 10-3. “Ukázkové rozvržení upřednostňující výkon” na stránce 10-45. “Ukázková rozvržení disku” na stránce 10-44.
Schopnost používat maximální povolený počet 32766 bloků.	“Určení maximálního počtu bloků a paměťových prostorů” na stránce 11-9.
Schopnost replikace rozšířených typů prostřednictvím replikace HDR.	“Typy replikovaných dat” na stránce 20-3.

Posílení bezpečnosti

Verze 9.4 přidává podporu modulů pro podporu šifrování komunikace (modulů ENCCSM), které umožňují šifrovat datové přenosy v síti. Tato volba poskytuje šifrování veškerých dat pomocí knihovny OpenSSL s mnoha konfigurovatelnými volbami.

Nové vlastnosti	Odkaz
Šifrování datových přenosů v síti.	“Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat” na stránce 5-11

Tento produkt obsahuje software vyvinutý v rámci projektu OpenSSL k použití v sadě OpenSSL Toolkit (<http://www.openssl.org/>).

Vlastnosti serveru Dynamic Server verze 9.3

V serveru IBM Informix Dynamic Server verze 9.3 byly představeny následující vlastnosti.

Vylepšení výkonu

Verze 9.3 obsahuje mnoho nových vlastností, které pomáhají monitorovat a zlepšovat výkon databáze.

Nové vlastnosti	Odkaz
Volba příkazu onstat -g stm .	“Monitorování transakcí” na stránce 13-6
Dynamické přidávání logických protokolů.	Kapitola 14, “Logický protokol”, na stránce 14-1 Kapitola 15, “Správa souborů logických protokolů”, na stránce 15-1

Vylepšení jazyka SQL

Verze 9.3 obsahuje několik nových příkazů jazyka SQL, které usnadňují migraci databází serverů jiných než Informix na server Dynamic Server verze 9.4.

Nové vlastnosti	Odkaz
Konfigurovatelné výchozí režimy uzamykání	“Tabulka zámek” na stránce 8-17

Další významné změny ve verzi 9.3

Následující tabulka uvádí další významné změny této příručky ve verzi 9.3.

Změny příručky	Odkaz
Program IBM Informix Server Administrator (ISA) obsahuje mnoho nových vlastností.	Nápověda online programu ISA.
Použití konfiguračního parametru VPCLASS místo parametrů AFF_NPROCS, AFF_SPROC, NOAGE, NUMAIOVPS a NUMCPUVPS.	Kapitola 7, “Správa virtuálních procesorů”, na stránce 7-1

Vlastnosti verze 9.21 serveru Dynamic Server

Tyto vlastnosti byly představeny v IBM Informix Dynamic Server, verze 9.21.

Vlastnosti	Odkaz
Neprotokolující tabulky (typu RAW).	“Typy tabulek serveru Dynamic Server” na stránce 10-30
Obslužný program onpladm .	“Zavedení dat do tabulky” na stránce 11-58
Vylepšení rychlé vyrovnávací paměti příkazů jazyka SQL: <ul style="list-style-type: none">• nové konfigurační parametry,• nové volby příkazu onstat -g ssc,• nové volby příkazu onmode -W ke změně parametrů rychlé vyrovnávací paměti příkazů jazyka SQL.	“Nastavení parametrů rychlé vyrovnávací paměti příkazů jazyka SQL” na stránce 9-7

Změny uspořádání této příručky ve verzi 9.2

Uspořádání této příručky bylo změněno následovně:

- Informace o konfiguraci databázového serveru a přehledné informace o úlohách administrace se nacházejí v části Kapitola 1, “Instalace a konfigurace databázového serveru”, na stránce 1-1.
- Informace o vícenásobném uložení a vysoké dostupnosti v systému Windows byly přesunuty do příručky *IBM Informix: Příručka pro instalaci*.
- Informace o správě provozních režimů databázového serveru a o inicializaci databázového serveru naleznete v části Kapitola 4, “Inicializace databázového serveru”, na stránce 4-1.
- Informace o fyzickém protokolu a kontrolních bodech naleznete v části Kapitola 16, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 16-1.

Konvence používané v dokumentaci

Tato část popisuje konvence používané v této příručce. Tyto konvence usnadňují vyhledávání informací v této i dalších příručkách dokumentace.

Naleznete zde popis konvencí těchto typů:

- typografické konvence,
- další konvence,
- diagramy syntaxe,
- konvence příkazového řádku,
- konvence kódu příkladů.

Typografické konvence

V této příručce se používají při zavádění nových termínů, popisu zobrazených objektů, popisu syntaxe příkazů atd. následující konvence.

Konvence	Význam
KLÍČOVÉ_SLOVO	Všechny primární prvky příkazů programovacího jazyka (klíčová slova) jsou psána velkými písmeny písma serif.
kurzíva <i>kurzíva</i> <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 <i>tučné písmo</i>	Tučným písmem jsou sázeny součásti programů (například třídy, události či tabulky), proměnné prostředí, názvy souborů a cest a prvky rozhraní (například ikony, položky v nabídce či tlačítka).
bezpatkové písmo <i>bezpatkové písmo</i>	Bezpatkovým písmem je sázen text zobrazený daným produktem a text zadáný uživatelem.
KLÁVESY	Velkými písmeny písma sans serif jsou sázeny klávesy, které by měly být stisknuty uživatelem.

Konvence	Význam
>	Tento symbol označuje položku v nabídce. Například zápis “Zvolte Nástroje> Možnosti ” znamená: Zvolte položku Možnosti v nabídce Nástroje .

Tip: Pokud “zadávaté” posloupnost znaků nebo “provádíte” příkaz, stiskněte po jeho zadání klávesu ENTER. Pokud program požaduje “zadání” textu nebo “stisk” jiné klávesy, není nutné klávesu ENTER stisknout.

Vlastnost, produkt a platforma

Značky vlastností, produktů a platformem 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 Extended Parallel Server

Označuje informace týkající se pouze serveru IBM Informix Extended Parallel Server.

Konec Jen pro Extended Parallel Server

Pouze pro UNIX

Označuje informace týkající se pouze platformy systému UNIX.

Konec Pouze pro UNIX

Jen pro Windows

Označuje informace týkající se pouze prostředí systému 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 (jen pro Linux)

Diagramy syntaxe


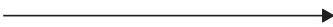




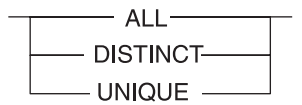
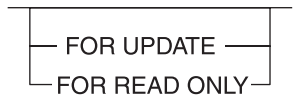
Tato příručka používá k popisu syntaxe příkazů (netýká se příkazů systémové úrovně) a příkazů SQL diagramy syntaxe sestávající z následujících komponent.

Poznámka: Počínaje rokem 2004 mají diagramy syntaxe změněný formát podle standardů IBM.

V diagramech syntaxe popisujících příkazy SQL a příkazy příkazového řádku došlo k těmto změnám:

- Místo svislé čáry na začátku a konci příkazů jsou nyní používány dvojité šipky.
- Místo šipek na začátku a konci segmentů diagramu syntaxe jsou používány svislé čáry.
- Maximální počet opakování smyčky je nyní vyznačen v poznámce pod čarou (dříve byl umístěn v symbolu brány).
- Příkazy syntaxe delší než jeden řádek pokračují na dalším řádku.
- Cesty vztahující se ke konkrétnímu produktu či podmínce jsou vyznačeny v poznámce pod čarou (dříve byly vyznačeny ikonou).

Následující tabulka popisuje komponenty používané v diagramech syntaxe.

Reprezentace komponenty v souborech PDF	Reprezentace komponenty v souborech HTML	Význam
	>>-----	Začátek příkazu.
	----->	Příkaz pokračuje na dalším řádku.
	>-----	Příkaz pokračuje z předchozího řádku.
	-----><	Konec příkazu.
	-----SELECT-----	Povinná položka.
	--+-----+--- '-----LOCAL-----'	Nepovinná položka.
	---+----ALL-----+--- +--DISTINCT-----+ '---UNIQUE-----'	Povinná položka s možností volby. Je nutné zadat právě jednu položku.
	---+-----+--- +--FOR UPDATE-----+ '--FOR READ ONLY--'	Volitelné položky s možností volby jsou zobrazeny pod hlavní čarou a je možné vybrat jednu z nich.

Reprezentace komponenty v souborech PDF	Reprezentace komponenty v souborech HTML	Význam
	<pre> .---NEXT----- -----+-----+ +---PRIOR-----+ '---PREVIOUS----'</pre>	Hodnoty pod hlavní čarou jsou volitelné, jednu z nich můžete určit. Pokud neurčíte položku, použije se hodnota nad čarou jako výchozí.
	<pre> .-----,----- v -----+-----+ +--název_indexu--+ '--název_tabulky--'</pre>	Nepovinné položky. Příkaz může obsahovat několik položek, které musí být odděleny čárkou.
	>>- Odkaz na tabulku -<<	Odkaz na segment syntaxe.
<p>Odkaz na tabulku</p>	<p>Odkaz na tabulku</p> <pre> --+----pohled-----+-- +-----tabulka-----+ '---synonymum-----'</pre>	Segment syntaxe.

Jak číst diagramy syntaxe příkazového řádku

Následující diagramy syntaxe příkazů příkazového řádku používají některé prvky popsané v tabulce v předchozí části.

Vytvoření úlohy bez převodu

```

>>--onpladm create job--job--[-p--project]--[-n--d--device]--[-D--database]-->>
```

```

>>--t--table-->>
```

```

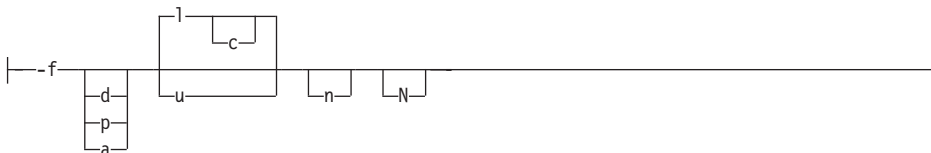
>>--[-S--server]--[-T--target]--| Nastavení režimu spuštění | (1) -->>
```

Poznámky:

1 Další informace naleznete na stránce 17-4.

Druhý řádek tohoto diagramu obsahuje segment s názvem “Nastavení režimu spuštění”, který je podle poznámky pod čarou možné nalézt na stránce 17-4. Tento segment je zakreslen v následujícím diagramu segmentů (diagram používá komponenty označující začátek a konec segmentu).

Nastavení režimu spuštění:



Začátek diagramu je v levém horním rohu. Procházejte diagram směrem doprava a zařaďte do příkazu požadované prvky. V příkazu není možné změnit velká písmena na malá a naopak.

Význam diagramu Vytvoření úlohy bez převodu je následující:

1. Zadejte příkaz **onpladm create job** následovaný názvem úlohy.
2. Název projektu je možné zadat pomocí volby **-p** následované tímto názvem.
3. Dále je nutné zadat tyto volby:
 - **-n**
 - **-d** a název zařízení
 - **-D** a název databáze
 - **-t** a název tabulky
4. Dále je možné přidat libovolný počet následujících voleb:
 - **-S** a název serveru.
 - **-T** a název cílového serveru.
 - Režim spuštění. Nastavení režimu spuštění je podle segmentového diagramu Nastavení režimu spuštění možné provést použitím volby **-f**, volitelnou volbou **d**, **p** či **a** následovanou opět volitelnou volbou **l** nebo **u**.
5. Procházejte diagram až k ukončovacímu členu.

Nyní je diagram kompletní.

Klíčová slova a interpunkce

Klíčová slova jsou slova rezervovaná pro všechny příkazy kromě příkazů systémové úrovně. Klíčová slova v diagramech syntaxe jsou psána velkými písmeny. Pokud používáte klíčové slovo v příkazu, je možné ho psát malými i velkými písmeny. Je však nutné přesně dodržet tvar uvedený v diagramu syntaxe.

Zároveň je v příkazu nutné dodržet uvedenou interpunkci.

Identifikátory a názvy

Proměnné nahrazují identifikátory a názvy v diagramech syntaxe a příkladech. Proměnnou je možné podle kontextu nahradit libovolným názvem, identifikátorem či literálem. Proměnné také nahrazují větší celky diagramu a bývají pro přehlednost podrobně rozepsány v jiném diagramu syntaxe. Proměnné v diagramech syntaxe, příkladech a textu jsou psány *malými písmeny kurzívou*.

Následující diagram syntaxe ilustruje použití proměnných v obecném tvaru příkazu SELECT.

►—SELECT—*název_sloupce*—FROM—*název_tabulky*—►

Při zadávání příkazu SELECT v tomto tvaru je nutné nahradit proměnné *název_sloupce* a *název_tabulky* názvy příslušného sloupce a tabulky.

Konvence použité v ukázkovém kódu

Příklady kódu SQL se vyskytují v celé této příručce. Pokud není uvedeno jinak, není daný kód specifický pro žádný konkrétní aplikační vývojový nástroj 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 SQL kód je nutné pro každý produkt upravit podle příslušných pravidel syntaxe. Například při použití 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).

Tip: 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.

Podrobnější informace o použití příkazů SQL v jednotlivých aplikačních vývojových nástrojích či rozhraních SQL API naleznete v příslušné příručce.

Další dokumentace

Další informace naleznete v následujících typech dokumentace:

- příručky pro instalaci
- poznámky online
- chybové zprávy systému Informix
- příručky
- nápověda online

Příručky pro instalaci

Příručky pro instalaci se nacházejí v adresáři **/doc** produktového disku CD nebo v adresáři **/doc** komprimovaného souboru k produktu, pokud jste soubor stáhli z webového serveru IBM. Příručky pro instalaci můžete také získat ze serveru dokumentace online systému IBM Informix na adrese <http://www-3.ibm.com/software/data/informix/pubs/library/>.

Poznámky online

Následující části popisují soubory online, které doplňují informace uvedené v této příručce. Přečtěte si tyto soubory, než začnete používat produkt IBM Informix. Obsahují důležité informace týkající se produktu a výkonu.

Online soubor	Popis	Formát
Poznámky s obsahem	Soubor poznámek s obsahem poskytuje úplný adresář hypertextových odkazů na poznámky k verzi, soubor známých a opravených chyb a na všechny soubory poznámek k dokumentaci k jednotlivým příručkám.	HTML
Poznámky k dokumentaci	Soubor poznámek k dokumentaci každé příručky obsahuje důležité informace a opravy, které doplňují informace v příručce, případně informace, které se od vydání příručky změnily.	HTML, text
Poznámky k verzi	Soubor poznámek k verzi popisuje rozdíly vlastností oproti předchozím verzím produktu IBM Informix a způsob, jakým tyto rozdíly mohou ovlivnit existující produkty. Pro některé produkty obsahuje tento soubor také informace o všech známých problémech a způsobech jejich odstranění.	HTML, text
Poznámky k počítači	(Pouze pro platformy jiné než Windows) Soubor poznámek k počítači popisuje všechny akce specifické pro platformu, které je zapotřebí provést ke konfiguraci a použití produktů IBM Informix ve vašem počítači.	text
Soubor známých a opravených chyb	Tento soubor popisuje problémy, které byly zjištěny v současné verzi. Také popisuje chyby zjištěné zákazníky, které byly opraveny v současné verzi a ve verzích předchozích.	text

Nalezení poznámek online

Poznámky online jsou k dispozici na webovém serveru dokumentace online systému IBM Informix na adrese <http://www-3.ibm.com/software/data/informix/pubs/library/>. Tyto soubory můžete také nalézt před instalací nebo po instalaci podle následujícího popisu.

Před instalací

Všechny poznámky online se nacházejí v adresáři **/doc** produktového disku CD. Nejsnazší způsob zpřístupnění poznámek k dokumentaci, poznámek k verzi a souboru známých chyb je pomocí hypertextových odkazů v souboru poznámek s obsahem.

Soubor poznámek k počítači a soubor známých a opravených chyb jsou k dispozici pouze v textovém formátu.

Po instalaci

Na platformách systému UNIX se ve výchozím národním prostředí nacházejí poznámky k dokumentaci, poznámky k verzi a poznámky k počítači nacházejí v adresáři **SINFORMIXDIR/release/en_us/0333**.

Dynamic Server

V systému Windows se poznámky k dokumentaci a k verzi nacházejí ve složce **Informix**. Tuto složku zobrazíte klepnutím na tlačítko **Start > Programy > IBM Informix Dynamic Server číslo verze > Documentation Notes** nebo **Release Notes** na hlavním panelu.

Poznámky k počítači se netýkají platformou Windows.

Konec Dynamic Server

Názvy souborů poznámek online

Poznámky online jsou k dispozici v následujících formátech souborů:

Online soubor	Formát souboru	Příklady
Poznámky s obsahem	<i>prod_os_tocnotes_verze.html</i>	ids_win_tocnotes_10.0.html
Poznámky k dokumentaci	<i>prod_nazev_knihy_docnotes_verze.html/txt</i>	ids_hpl_docnotes_10.0.html
Poznámky k verzi	<i>prod_os_relnotes_verze.html/txt</i>	ids_unix_relnotes_10.0.txt
Poznámky k počítači	<i>prod_machine_notes_verze.txt</i>	ids_machine_notes_10.0.txt
Opravené a známé chyby	<i>prod_defects_verze.txt</i>	ids_defects_10.0.txt client_defects_2.90.txt
	<i>ids_win_fixed_and_known_defects_verze.txt</i>	ids_win_fixed_and_known_defects_10.0.txt

Chybové zprávy systému Informix

Tento soubor obsahuje úplný rejstřík chybových zpráv a příslušných nápravných akcí pro produkty Informix příslušných verzí.

Na platformách UNIX použijte k zobrazení chybových zpráv a příslušných nápravných akcí příkaz **finderr**.

Dynamic Server

V systému Windows použijte k zobrazení chybových zpráv a příslušných nápravných akcí obslužný program Informix Error Messages. K zobrazení tohoto obslužného programu klepněte na hlavním panelu na tlačítko **Start > Programy > IBM Informix Dynamic Server verze > Informix Error Messages**.

Konec Dynamic Server

Tyto soubory můžete také zpřístupnit z webového serveru dokumentace online systému IBM Informix <http://www-3.ibm.com/software/data/informix/pubs/library/>.

Příručky

Příručky online

S produkty IBM Informix se dodává disk CD obsahující příručky v elektronickém formátu. Dokumentaci můžete nainstalovat nebo ji zpřístupnit přímo z disku CD. Informace o způsobu instalace, prohlížení a tisku příruček online naleznete na archu s postupem instalace přiloženém k disku CD. Tytéž příručky online můžete získat ze serveru dokumentace online systému IBM Informix na adrese <http://www-3.ibm.com/software/data/informix/pubs/library/>.

Tištěné příručky

Chcete-li si objednat tištěné verze příruček, obraťte se na svého obchodního zástupce nebo navštivte webový server IBM Publications Center Web na adrese <http://www.ibm.com/software/howtobuy/data.html>.

Nápověda online

Nápověda online systému IBM Informix dodávaná ke každému grafickému uživatelskému rozhraní (GUI) zobrazuje informace o těchto rozhraních a o funkcích, které provádějí. K zobrazení nápovědy online použijte funkce nápovědy, které poskytuje každé rozhraní GUI.

Usnadnění

IBM se zavázala vytvářet dokumentaci přístupnou postiženým osobám. Naše knihy jsou dostupné ve formátu HTML, aby byly přístupné prostřednictvím pomocného softwaru, jakým jsou nástroje pro čtení obsahu obrazovky. Diagramy syntaxe v našich příručkách jsou k dispozici ve formátu desítkových čísel oddělených tečkami, což je přístupný formát dostupný pouze pokud používáte nástroj pro čtení obsahu obrazovky. Další informace o formátu desítkových čísel oddělených tečkami naleznete v dodatku Usnadnění.

Sada dokumentace systému IBM Informix Dynamic Server verze 10.0 a sady CSDK verze 2.90

V následujících tabulkách jsou uvedeny příručky, které jsou součástí sady dokumentace serveru IBM Informix Dynamic Server verze 10.0 a sady CSDK verze 2.90. Verze těchto příruček ve formátu PDF a HTML jsou k dispozici na adrese <http://www-3.ibm.com/software/data/informix/pubs/library/>. Tištěné verze těchto příruček lze objednat na webovém serveru IBM Publications Center na adrese <http://www.ibm.com/software/howtobuy/data.html>.

Tabulka 1. Příručky k databázovému serveru

Příručka	Téma
Administrator's Guide	Porozumění, konfigurace a administrace databázového serveru.
Administrator's Reference	Referenční materiál serveru Informix Dynamic Server, například syntax obslužných programů databázového serveru onmode a onstat a popisy konfiguračních parametrů, tabulek sysmasters a souborů logických protokolů.
Backup and Restore Guide	Koncepty a metody potřebné při použití obslužných programů ON-Bar a ontape k zálohování a obnově dat.
DB-Access User's Guide	Použití obslužného programu DB-Access ke zpřístupnění, úpravám a získávání dat z databází Informix.
DataBlade API Function Reference	Funkce rozhraní API DataBlade a část funkcí jazyka ESQL/C podporovaná rozhraním API DataBlade. Pomocí rozhraní API DataBlade API k vývoji aplikací LIBMI a uživatelských rutin v jazyku C, které přistupují k databázím systému Informix.
DataBlade API Programmer's Guide	Rozhraní API DataBlade je rozhraní pro programování aplikací v jazyku C dodávané se serverem Dynamic Server. Rozhraní API DataBlade můžete použít k vývoji klientských a serverových aplikací, které přistupují k datům uloženým v databázích systému Informix.
Database Design and Implementation Guide	Návrh, implementace a správa databází Informix.
Enterprise Replication Guide	Způsob návrhu, implementace a správy systému replikace Enterprise Replication k replikaci dat mezi více databázovými servery.
Soubor chybových zpráv	Příčiny a řešení číslovaných chybových zpráv, které se mohou vyskytnout při práci s produkty IBM Informix.
Úvodní příručka	Popisuje produkty dodávané se serverem IBM Informix Dynamic Server a spolupráci s ostatními produkty IBM. Shrnuje důležité funkce serveru Dynamic Server a nové funkce každé verze.
Guide to SQL: Reference	Informace o databázích systému Informix, datových typech, tabulkách systémového katalogu, proměnných prostředí a ukázkové databázi stores_demo.
Guide to SQL: Syntax	Podrobné popisy syntaxe všech příkazů SQL a SPL systému Informix.
Guide to SQL: Tutorial	Kurz jazyka SQL implementovaného v produktech Informix, popisuje základní koncepty a termíny potřebné k práci s relačními databázemi.
High-Performance Loader User's Guide	Zpřístupnění a použití zavaděče High-Performance Loader (HPL) k zavádění a uvolňování velkého množství dat v databázích systému Informix.
Příručka pro instalaci pro Microsoft Windows	Instrukce k instalaci serveru IBM Informix Dynamic Server v systému Windows.
Příručka pro instalaci pro UNIX a Linux	Instrukce k instalaci serveru IBM Informix Dynamic Server v systému UNIX.
J/Foundation Developer's Guide	Tvorba uživatelských rutin (UDRs) v programovacím jazyku Java pro servery Informix Dynamic Server s třídami J/Foundation.

Tabulka 1. Příručky k databázovému serveru (pokračování)

Příručka	Téma
Large Object Locator DataBlade Module User's Guide	Použití modulu Large Object Locator, základního modulu DataBlade, který může být použit ostatními moduly, které vytvářejí nebo ukládají data velkých objektů. Modul Large Object Locator umožňuje vytvářet jednotné konzistentní rozhraní k velkým objektům a rozšiřuje koncept velkých objektů o data ukládaná mimo databázi.
Migration Guide	Převod a opětovné vracení z nejnovějších verzí databázových serverů Informix. Migrace mezi různými databázovými servery Informix.
Optical Subsystem Guide	Obslužný program Optical Subsystem, který podporuje ukládání dat typu TEXT a BYTE na optický disk.
Performance Guide	Konfigurace a provoz serveru IBM Informix Dynamic Server k dosažení optimálního výkonu.
R-Tree Index User's Guide	Vytváření indexů R-strom na vhodných datových typech, vytváření nových tříd operátorů, které používají přístupovou metodu R-strom a správa databází, které používají sekundární přístupovou metodu R-strom.
SNMP Subagent Guide	Dílčí agent systému IBM Informix, který umožňuje správci sítě používajícímu protokol Simple Network Management Protocol (SNMP) monitorovat stav serverů Informix.
Storage Manager Administrator's Guide	Program Informix Storage Manager (ISM), který spravuje paměťová zařízení a média pro databázový server systému Informix.
Trusted Facility Guide	Možnosti zabezpečeného auditu serveru Dynamic Server včetně vytváření a správy protokolů auditu.
User-Defined Routines and Data Types Developer's Guide	Způsob definování nových datových typů a umožnění uživatelským rutinám (UDR) rozšířit server IBM Informix Dynamic Server.
Virtual-Index Interface Programmer's Guide	Vytváření sekundární přístupové metody (indexu) pomocí rozhraní Virtual-Index Interface (VII) k rozšíření vestavěných schémat serveru IBM Informix Dynamic Server. Typicky se používá společně s modulem DataBlade.
Virtual-Table Interface Programmer's Guide	Vytváření primární přístupové metody pomocí rozhraní Virtual-Table Interface (VTI), aby uživatelé měli k dispozici jednotné rozhraní jazyka SQL k tabulkám systému Informix a k datům, která neodpovídají schématu ukládání serveru Informix Dynamic Server.

Tabulka 2. Příručky klientů a propojitelnosti

Příručka	Téma
Client Products Installation Guide	Instalace sady IBM Informix Client Software Developer's Kit (Client SDK) a produktu IBM Informix Connect do počítačů používajících systémy UNIX, Linux, and Windows.
Embedded SQLJ User's Guide	Použití jazyka Embedded SQLJ systému IBM Informix k vkládání příkazů jazyka SQL do programů v jazyku Java.
ESQL/C Programmer's Manual	Implementace vloženého jazyka SQL pro jazyk C systému IBM Informix.

Tabulka 2. Příručky klientů a propojitelnosti (pokračování)

Příručka	Téma
GLS User's Guide	Funkce Globální podpora jazyků (GLS), která umožňuje rozhraním API a databázovým serverům systému IBM Informix zpracovávat různé jazyky, kulturní konvence a znakové sady.
JDBC Driver Programmer's Guide	Instalace a použití ovladače Informix JDBC Driver ke ke připojení k databázi systému Informix z aplikace psané v jazyku Java nebo z appletu.
.NET Provider Reference Guide	Použití poskytovatele Informix .NET Provider ke zpřístupnění a manipulaci dat v databázích systému Informix klientskými aplikacemi platformy .NET.
ODBC Driver Programmer's Manual	Použití rozhraní API ovladače Informix ODBC Driver ke zpřístupnění databáze systému Informix a k interakci s databázovým serverem systému Informix.
OLE DB Provider Programmer's Guide	Instalace a konfigurace poskytovatele Informix OLE DB Provider ke zpřístupnění dat serveru Informix klientským aplikacím, například aplikacím ActiveX Data Object (ADO) nebo webovým stránkám.
Object Interface for C++ Programmer's Guide	Architektura objektového rozhraní jazyka C++ a úplná reference tříd.

Tabulka 3. Příručky sady DataBlade Developer's Kit

Příručka	Téma
DataBlade Developer's Kit User's Guide	Vývoj a komprimace modulů DataBlade pomocí programů BladeSmith a BladePack.
DataBlade Module Development Overview	Základní orientace ve vývoji modulů DataBlade. Obsahuje příklad ilustrující vývoj modulu DataBlade.
DataBlade Module Installation and Registration Guide	Instalace modulů DataBlade a použití programu BladeManager ke správě modulů DataBlade v databázích systému Informix.

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

Naše společnost ocení jakékoli připomínky k jednotlivým příručkám, případně návrhy na zlepšení v příštích verzích těchto příruček. Tyto připomínky by měly obsahovat:

- název a verzi příručky
- číslo části a strany
- jakékoli komentáře k příručce

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.

Veškeré návrhy jsou vítány.

Část 1. Databázový server

Kapitola 1. Instalace a konfigurace databázového serveru

Plánování databázového serveru	1-3
Posouzení priorit	1-3
Posouzení prostředí	1-4
Konfigurace operačního systému	1-4
Konfigurace paměti v systému Windows	1-4
Modifikace parametrů jádra systému UNIX.	1-5
Přidělení diskového prostoru	1-5
Použití velkých bloků	1-5
Vytvoření souborů bloků v systému UNIX.	1-6
Přímý přístup k disku neboli přístup bez vyrovnávací paměti	1-6
Předpřipravené soubory	1-6
Vytvoření oddílů NTFS ve Windows.	1-6
Nastavení oprávnění, vlastnictví a skupiny	1-7
Vytvoření standardních názvů zařízení (UNIX)	1-7
Nastavení proměnných prostředí	1-8
Nastavení proměnných prostředí GLS	1-9
Nastavení proměnných prostředí v systému UNIX	1-9
Nastavení proměnných prostředí v systému Windows	1-10
Konfigurace propojitelnosti	1-11
Soubor sqlhosts v systému UNIX	1-11
Soubory konfigurace sítě	1-12
Soubory zabezpečení sítě	1-12
Registr sqlhosts v systému Windows	1-12
Konfigurace propojitelnosti pomocí programu ISA	1-12
Konfigurace databázového serveru	1-12
Příprava konfiguračního souboru ONCONFIG	1-13
Vytvoření souboru ONCONFIG v systému UNIX	1-13
Vytvoření souboru ONCONFIG v systému Windows	1-13
Použití funkce Nastavení serveru obslužného programu ISA k vlastnímu nastavení konfigurace.	1-14
Použití obslužného programu IBM Informix Server Administrator k aktualizaci souboru ONCONFIG.	1-14
Použití programu Instance Manager k vytvoření nové instance databázového serveru (Windows)	1-15
Použití programu Instance Manager k přejmenování instance databázového serveru (Windows).	1-15
Konfigurace podpory jazyka Java	1-16
Spuštění a správa databázového serveru	1-16
Spuštění databázového serveru a inicializace diskového prostoru	1-16
Příprava pro automatické spuštění	1-17
Příprava automatického spuštění v systému Windows	1-17
Příprava spouštěcích a ukončovacích skriptů pro systém UNIX	1-17
Příprava připojení k aplikacím	1-19
Vytvoření paměťových prostorů a bloků	1-19
Podpora velkých bloků	1-20
Nastavení systému zálohování a ukládání	1-20
Nastavení obslužného programu ontape	1-20
Nastavení správce paměti a programu ON-Bar	1-20

Provedení rutinních úloh administrace	1-21
Změna režimů databázového serveru	1-21
Zálohování dat a souborů logického protokolu	1-21
Monitorování aktivity	1-22
Kontrola konzistence	1-22
Provádění dalších úloh administrace	1-22
Použití zrcadlení	1-22
Správa stavu protokolování databáze	1-22
Správa logického protokolu	1-22
Správa fyzického protokolu	1-23
Správa sdílené paměti	1-23
Správa virtuálních procesorů	1-23
Správa paralelních databázových dotazů	1-24
Použití replikace dat	1-24
Replikace HDR (High-Availability Data Replication)	1-24
Replikace ER (Enterprise Replication)	1-24
Použití auditu	1-24
Použití distribuovaných dotazů	1-25
Použití globálních transakcí	1-25
Použití správce transakcí	1-25
Monitorování aktivity databázového serveru	1-25
Alarmy událostí	1-26
Program IBM Informix Server Administrator (ISA)	1-26
Protokol zpráv	1-26
Určení cíle pro zprávy protokolu zpráv	1-27
Monitorování protokolu zpráv	1-27
Obslužný program ON-Monitor (UNIX)	1-27
Obslužný program oncheck	1-27
Nástroj onperf (UNIX)	1-28
Obslužný program onstat	1-28
Tabulky SMI	1-28
Systémová konzole	1-28
Nástroje operačního systému UNIX	1-29
Prohlížeč událostí v systému Windows	1-29
Sledování výkonu v systému Windows	1-29
Obslužné programy pro systém Windows	1-29

Obsah kapitoly

Při instalaci verze IBM Informix Dynamic Server 10.0 doporučujeme dodržet instalační pokyny, aby byla správně nastavena oprávnění všech klíčových souborů a adresářů. Pokyny pro instalaci naleznete v částech *IBM Informix: Příručka pro instalaci serveru Dynamic Server pro UNIX a Linux* a *IBM Informix: Příručka pro instalaci serveru Dynamic Server pro Microsoft Windows*.

Po instalaci nové verze databázového serveru je nutné ho konfigurovat.

Konfigurace je souhrnným označením pro nastavení specifických parametrů, která individuálně nastaví databázový server pro konkrétní prostředí zpracování dat: množství a typ dat, počet tabulek, hardware, počet uživatelů a požadavky zabezpečení.

V následujícím seznamu jsou uvedeny základní konfigurační požadavky popsané v této kapitole:

- Plánování databázového serveru.
- Konfigurace operačního systému.
- Přidělení diskového prostoru.
- Nastavení proměnných prostředí.
- Konfigurace údajů o propojitelnosti.
- Příprava konfiguračního souboru ONCONFIG.
- Použití funkce **Nastavení serveru** k vlastnímu nastavení konfigurace.
- Spuštění a správa databázového serveru.
- Monitorování aktivity databázového serveru.

Pokud databázový server spouštíte na počítači se systémem UNIX nebo Linux, obslužné programy serveru kontrolují, zda je prostředí zabezpečené. Další informace naleznete v části “Zabezpečení prováděné obslužnými programy serveru před spuštěním v operačním systému UNIX a Linux” na stránce 5-2.

Plánování databázového serveru

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 platforma UNIX službu Network Information Service (NIS)?
- 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 UNIX

Soubor Poznámky k počítači obsahuje informace o tom, které parametry lze použít u jednotlivých platformách UNIX.

Konec Pouze pro 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 8-42.

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.

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 *IBM Informix: Dynamic Server Performance Guide*.

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

Podrobné informace o úloze parametrů jádra systému UNIX naleznete v části Kapitola 9, “Správa sdílené paměti”, na stránce 9-1.

Přidělení diskového prostoru

Optimální výkon datových tržišť 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 blokům” na stránce 10-4.

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 kořenový blok zároveň velkým blokem, je během inicializace serveru přeskočen krok **onmode -BC 1** a automaticky je inicializován režim **-BC 2**.

Vytvoření souborů bloků v systému UNIX

V systému UNIX můžete data ukládat do bloků, které využívají buď disky bez vyrovnávací paměti (*disky s přímým přístupem*) nebo soubory operačního systému známé též jako soubory *s vyrovnávací pamětí* nebo *předpřipravené*.

Přímý přístup k disku neboli přístup bez vyrovnávací paměti

Systém UNIX poskytuje přímý přístup k disku prostřednictvím znakově orientovaných zařízení (nazývaných známých též *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: Ačkoli se pro dosažení lepšího výkonu doporučuje používat v systémech UNIX přímá disková zařízení, díky poslednímu vývoji může rychlá vyrovnávací paměť vstupu - výstupu pro předpřipravené zápisy dosahovat stejného, ne-li lepšího výkonu. 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 sbspace však musíte uložit do souborů NTFS nebo na fyzickou jednotku nebo logický diskový oddíl. Ke zjednodušení administrace disku doporučujeme používat soubory NTFS. Více informací o souborech NTFS naleznete v dokumentaci systému Windows a v části "Přístup k disku v systému Windows" na stránce 10-4.

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.

Pouze 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. Informace o tom, jak vytvořit skupinu naleznete v příručce *IBM Informix: Příručka pro instalaci serveru Dynamic Server pro UNIX a Linux* nebo v dokumentaci systému UNIX.

Konec Pouze 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 přímým diskovým zařízením doporučujeme použít symbolické odkazy. Použijete-li symbolické odkazy, můžete porušený disk nahradit novým diskem pouhým přiřazením symbolického názvu novému disku.

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

Na adresáři zařízení proveďte příkaz systému UNIX **ls -l**, který prověří existenci zařízení i propojení. 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 musí nastavit příslušné proměnné prostředí, chce-li spouštět a ukončovat databázový server a přistupovat k němu. 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. Proměnnou prostředí **PATH** nastavte na název adresáře **\$INFORMIXDIR/bin** (UNIX) nebo **%INFORMIXDIR%\bin** (Windows).
3. Proměnnou **INFORMIXSERVER** nastavte na název databázového serveru.

Tip: Doporučujeme nastavit proměnné prostředí v příslušném spouštěcím souboru pro soubor se skriptem interpretu příkazů nebo pro systém Windows.

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 *IBM Informix: Dynamic Server Performance Guide*.

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, určuje umístění souboru jvphome/krakatoa.jar , 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.

Tabulka 1-1. Povinné proměnné prostředí (pokračování)

Proměnná prostředí	Popis
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 nemusejí.</p> <p>Není-li k dispozici proměnná prostředí ONCONFIG, použijte 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>
PATH	<p>Určuje umístění spustitelných souborů.</p> <p>V systému UNIX: \$INFORMIXDIR/bin</p> <p>V systému Windows: %INFORMIXDIR%\bin</p>
TERM	<p>Umožňuje příkazu DB–Access rozpoznat terminál, který používáte a komunikovat s ním. Tato proměnná prostředí není povinná při inicializaci, ale je nutné nastavit ji předtím, než spustíte aplikaci.</p>
TERMCAP TERMINFO INFORMIXTERM	<p>Určuje, zdali 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 jejich hodnoty se systém od systému liší.</p>

Nastavení proměnných prostředí GLS

Následující proměnné prostředí umožňují využívat funkci Global Language Support (GLS). 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

Proměnné prostředí systému UNIX nastavíte 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 přiřadte 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 ESQ/L/C.

```
setenv INFORMIXDIR /ix/informix93
setenv INFORMIXSQLHOSTS /ix/sqlhosts.unified
setenv ONCONFIG onconfig.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 1-13.

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 inicializací databázového serveru není nutné určit všechna možná síťová připojení v souboru nebo registru **sqlhosts**. 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 části “Omezení zahlcovacích útoků typu DOS (operační systém UNIX)” na stránce 5-8.

Soubor **sqlhosts** v systému UNIX

V systému UNIX obsahuje údaje o propojitelnosti soubor **sqlhosts**. Výchozí umístění tohoto souboru je **\$INFORMIXDIR/etc/sqlhosts**. Umístíte-li tyto údaje jinam, musíte nastavit proměnnou prostředí **INFORMIXSQLHOSTS**. Další informace naleznete v části “Soubor **sqlhosts** a klíč registru **SQLHOSTS**” na stránce 3-15.

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. Více informací naleznete v části “Konfigurace propojitelnosti pomocí programu ISA” na stránce 1-12.

Soubory konfigurace sítě

Kromě souborů **sqlhosts** je pro připojení prostřednictvím protokolu TCP/IP nutné upravit též soubory **/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 se řídí bezpečnostními požadavky systému UNIX pro vytváření připojení. Je možné, že administrátor systému UNIX bude muset provést úpravy v souborech **/etc/passwd**, **etc/hosts**, **~/rhosts** a dalších souvisejících souborech.

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. Doporučujeme registr **HKEY_LOCAL_MACHINE** neupravovat.

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.

V programu ISA vyberte **Konfigurace > SQLHOSTS**.

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 4-4.

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 během inicializace nahradí chybějící parametry hodnotami uvedenými v souboru **onconfig.std**.

Vytvoření souboru ONCONFIG v systému UNIX

Během instalace programového vybavení IBM Informix je vytvořena a inicializována 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 je nová instance databázového serveru vytvořena a inicializována v rámci instalace programového vybavení IBM Informix. 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 1-15.

2. Pomocí textového editoru nebo programu ISA upravte konfigurační soubor ONCONFIG.
3. Do proměnné prostředí ONCONFIG vložte název nového souboru ONCONFIG.
4. Vypněte a restartujte databázový server, aby změny nabýly 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 nabýly 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 adresu URL obslužného programu ISA:

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

Pokud jste nenainstalovali nebo nespustili program ISA, nahlédněte do příručky *IBM Informix: Příručka pro instalaci*, 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 nabýly 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: Příručka pro instalaci serveru Dynamic Server pro 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 s třídami J/Foundation umožňuje vyvíjet a spouštět uživatelské rutiny UDR v jazyku Java. Databázový server nakonfigurujete bez jazyka Java a pak ho upravte a přidejte podporu jazyka Java. Konfigurace databázového serveru k podpoře jazyka 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 *IBM Informix: J/Foundation Developer's Guide*.

Spuštění a správa databázového serveru

Po instalaci a konfiguraci databázového serveru musíte provést některé z následujících kroků:

- 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

Pouze pro UNIX

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

Spouštíte-li nový databázový server, použijte příkaz **oninit** s příznakem **-i**, kterým inicializujete diskový prostor a uvedete databázový server do režimu online.

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

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.

Příprava automatického spuštění v systému Windows

Pokud v systému Windows změníte 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 spuštění** 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 v systému UNIX můžete upravit tak, aby automaticky inicializoval databázový server ve chvíli, kdy počítač přejde do víceuživatelského režimu. Můžete upravit také ukončovací skript systému UNIX tak, aby řízeně ukončil databázový server při ukončování operačního 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.

Obslužný program ISA obsahuje příklad spouštěcího a ukončovacího skriptu systému UNIX, který lze individuálně přizpůsobit, v souboru **\$INFORMIXDIR/etc/ids-example.rc**.

Příprava spouštěcího skriptu systému UNIX:

1. Proměnnou prostředí **INFORMIXDIR** nastavte na úplný název cesty k adresáři, ve kterém je nainstalován databázový server.
2. Do proměnné prostředí **PATH** nastavte název adresáře **\$INFORMIXDIR/bin**.
3. Do proměnné prostředí **ONCONFIG** vložte název požadovaného konfiguračního souboru.
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: Příručka pro instalaci*.

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 řízeně ukončit databázový server, jakmile je ukončen systém UNIX, přidejte do ukončovacího skriptu pro systém UNIX příkazy systému UNIX a obslužných programů databázového serveru, aby skript provedl následující kroky.

Příprava ukončovacího skriptu systému UNIX:

1. Do proměnné prostředí **INFORMIXDIR** nastavte úplný název cesty k adresáři, v kterém je nainstalován databázový server.
2. Do proměnné prostředí **PATH** nastavte název adresáře **\$INFORMIXDIR/bin**.
3. Do proměnné prostředí **ONCONFIG** nastavte název požadovaného konfiguračního souboru.
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 ukončovací příkazy 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 jednoho bloku je 4 TB. Maximální povolený počet bloků je 32766. Logický *paměťový prostor* je tvořen jedním nebo několika bloky.

Tip: 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 sbpace:

- 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 tbspace a oblasti, uvádí Kapitola 10, “Uložení dat”, na stránce 10-1. Informace související s přidělováním a správou paměťových prostorů uvádí část “Parametry diskového prostoru” na stránce 2-2 a Kapitola 11, “Správa diskového prostoru”, na stránce 11-1.

Další informace naleznete v příručce *IBM Informix: Dynamic Server Enterprise Replication Guide* a v příručce *IBM Informix: 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, spusťte 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. Jakmile je aktivována podpora velkých bloků, nelze ji zakázat.

Nastavení systému zálohování a ukládání

K zálohování a obnovení dat používejte obslužný program ON-Bar nebo **ontape**. Další informace o nastavení a použití obslužných programů ON-Bar a **ontape** naleznete v části “Parametry zálohování a obnovení” na stránce 2-6 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 pojmenovávání 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. Pokud ale chcete používat sofistikovanější 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. Více informací o konfiguraci a správě paměťových zařízení a médií naleznete v části *IBM Informix: Storage Manager Administrator's Guide* nebo v dokumentaci správce paměti jiného výrobce.

Provedení rutinních úloh administrace

V závislosti na potřebách organizace může být správce serveru odpovědný za provádění periodických úloh popsanych 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 4-9 objasňuje, jak měnit provozní režimy databázového serveru.

Zálohování dat a souborů logického protokolu

Chcete-li mít jistotu, že v případě poruchy budete moci obnovit databáze, doporučujeme časté zálohování paměťových prostorů a logických protokolů. 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 dbospace, 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 1-25 obsahuje popis dostupných informací, pokyny k získání informací a doporučení k jejich používání.

Kontrola konzistence

Doporučujeme provádět občasné kontroly konzistence dat. Další informace o těchto úlohách uvádí Kapitola 22, “Kontrola konzistence”, na stránce 22-1.

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 18, “Zrcadlení”, na stránce 18-1. Pokyny týkající se úloh souvisejících se zrcadlením dat uvádí Kapitola 19, “Použití zrcadlení”, na stránce 19-1.

Důležité: Doporučujeme zrcadlit důležité prostory *dbspace* obsahující 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 10-33 a Kapitola 12, “Protokolování”, na stránce 12-1. Další informace o změnách voleb protokolování uvádí Kapitola 13, “Správa režimu protokolování databáze”, na stránce 13-1.

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 2-4 a Kapitola 14, “Logický protokol”, na stránce 14-1. Pokyny k vytváření a úpravám konfigurace logického protokolu uvádí část Kapitola 15, “Správa souborů logických protokolů”, na stránce 15-1. 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 2-5, Kapitola 16, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 16-1 a Kapitola 17, “Správa fyzického protokolu”, na stránce 17-1.

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 rychlé vyrovnávací paměti příkazů jazyka SQL ke snížení využití paměti a doby potřebné k přípravě dotazů.

Informace o tom, jak databázový server využívá sdílenou paměť, uvádí Kapitola 8, “Sdílená paměť”, na stránce 8-1. Další informace uvádí část “Parametry sdílené paměti” na stránce 2-6 a Kapitola 9, “Správa sdílené paměti”, na stránce 9-1.

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 6, “Virtuální procesory a jednotkové procesy”, na stránce 6-1. Další informace uvádí část “Parametry virtuálních procesorů” na stránce 2-10 a Kapitola 7, “Správa virtuálních procesorů”, na stránce 7-1.

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 *IBM Informix: Dynamic Server Performance Guide*. Informace o konfiguračních parametrech naleznete v části “Parametry podpory rozhodování” na stránce 2-9.

Použití replikace dat

Pojem *replikace dat* označuje proces reprezentace objektů databáze na více než jednom samostatném místě. Server Dynamic Server podporuje dvě metody replikace dat: *replikaci HDR* (High-Availability Data Replication) a *replikaci IBM Informix Enterprise Replication* (ER). Na jednom databázovém serveru můžete kombinovat replikaci HDR a replikaci Enterprise Replication. Další informace naleznete v příručce *IBM Informix: Dynamic Server Enterprise Replication Guide*.

Replikace HDR (High-Availability Data Replication)

Replikace HDR podporuje synchronní replikaci celé databáze na jiný databázový server, čímž poskytuje okamžitou náhradu v případě vážné poruchy počítače. Více informací uvádí část “Parametry replikace High-Availability Data-Replication” na stránce 2-11, Kapitola 20, “Replikace HDR (High-Availability Data Replication)”, na stránce 20-1 a Kapitola 21, “Použití replikace HDR”, na stránce 21-1.

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 (systém UNIX)” na stránce 2-14 a v příručce *IBM Informix: Trusted Facility Guide*.

Použití distribuovaných dotazů

Databázový server umožňuje uživatelům dotazovat a aktualizovat více databází ve více databázových serverech. Tento typ dotazování se nazývá *distribuovaný dotaz*. Databázové servery mohou být spuštěny na jediném hostitelském počítači, v jediné síti nebo k nim můžete přistupovat prostřednictvím 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 23, “Protokoly vícefázového potvrzování”, na stránce 23-1.

Použití správce transakcí

Programy typu *správce transakcí* slouží ke správě terminálů a k obnově dat. Knihovna TP/XA umožňuje komunikaci mezi správcem transakcí jiného výrobce a databázemi IBM Informix v prostředí X/Open. Více informací naleznete v příručce *IBM Informix: TP/XA Programmer's Manual* a v části “Správci transakcí” na stránce 23-2.

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 událostí 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.

Více informací o programu ISA naleznete v kapitole o obslužných programech v příručce *IBM Informix: Dynamic Server Administrator's Reference* a v nápovědě online 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 situací vyžadujících okamžitou pozornost používá databázový server funkci alarmu událostí. Více informací naleznete v části "Alarmy událostí" na stránce 1-26.

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

Doporučujeme jednou nebo dvakrát denně zkontrolovat protokol zpráv a přesvědčit se, že je zajištěn normální průběh zpracování a že jsou události protokolovány podle očekávání. Pomocí příkazu **onstat -m** získáte název protokolu zpráv a 20 nejaktuálnějších záznamů. K zobrazení celého protokolu zpráv použijte textový editor. Použijte příkaz operačního systému (například příkaz systému UNIX **tail -f**), chcete-li zobrazit zprávy tak, jak se objevují.

Monitorujte velikost protokolu zpráv, protože databázový server přidává k tomuto souboru nové záznamy. Protokol dle 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 nepředvidatelný 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 kapitole věnované obslužným programům 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 *IBM Informix: Dynamic Server Performance Guide*.

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 znázorňuje typické obslužné programy k monitorování zdrojů integrované do operačního 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 položku **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, rychlá vyrovnávací paměť, 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\informix</i>.</p> <p>Použití: <code>ixpasswd [-y nové_heslo]</code></p>
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> <p>Obslužný program ixsu je ekvivalentem příkazu Windows 2000 runas. Použití příkazu runas ke spuštění příkazového řádku jako jiný uživatel:</p> <p>Použití: <code>runas /user:uživatelské_jméno cmd</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>

Kapitola 2. Konfigurační parametry

Parametry identifikace databázového serveru	2-2
Parametry diskového prostoru.	2-2
Kořenový prostor dbspace	2-2
Zrcadlení kořenového prostoru dbspace	2-3
Další parametry správy prostorů	2-3
Parametry protokolování	2-4
Logický protokol	2-4
Parametry fyzického protokolu	2-5
Parametry odvolávání transakcí a zotavení.	2-5
Parametry zálohování a obnovení	2-6
Parametry protokolu zpráv.	2-6
Parametry sdílené paměti	2-6
Velikost přidělené sdílené paměti.	2-7
Přidělený prostor sdílené paměti	2-7
Řízení vyrovnávací sdílené paměti	2-8
Využití rychlé vyrovnávací paměti příkazů SQL	2-9
Parametry podpory rozhodování	2-9
Parametry procesu databázového serveru	2-10
Parametry virtuálních procesorů.	2-10
Časové intervaly	2-11
Parametry obnovení	2-11
Parametry replikace High-Availability Data-Replication	2-11
Parametry alarmů událostí	2-12
Parametry vypisování jádra (systém UNIX)	2-13
Parametry direktiv	2-13
Parametry připojení	2-13
Parametry související se zabezpečením.	2-14
Zvláštní parametry.	2-14
Parametry prověřování (systém UNIX).	2-14
Parametry optických médií	2-14
Parametry systému UNIX	2-15
Monitorování informací o konfiguraci.	2-15

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	<p>Určuje alternativní název nebo názvy instance databázového serveru. Maximální počet aliasů je 32.</p> <p>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 6-28.</p> <p>Upozornění: Neměňte konfigurační parametr DBSERVERNAME bez restartování databázového serveru.</p>
DBSERVERNAME	<p>Určuje jedinečný název instance databázového serveru. Konfigurační parametr DBSERVERNAME použijte pro proměnnou prostředí INFORMIXSERVER a v souboru sqlhosts.</p>
SERVERNUM	<p>Určuje jedinečné číslo instance databázového serveru. Databázový server používá parametr SERVERNUM k určení adres segmentů sdílené paměti.</p>

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 10-13.
ROOTOFFSET	Zadáva posun. Informace o tom, kdy je třeba zadat parametr ROOTOFFSET naleznete v části “Určení posunu” na stránce 11-4.
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 11-3.
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 10-40.
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 19, “Použití zrcadlení”, na stránce 19-1.
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 11-4.

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 11-19.
FILLFACTOR	Určuje míru zaplnění stránek indexů při vytváření indexů. Další informace naleznete v příručce <i>IBM Informix: Dynamic Server Performance Guide</i> .
ONDBSPACEDOWN	Určuje, jak má databázový server pracovat ze zakázaným prostorem dbspace, který není kritický pro provoz serveru.
SBSPECENAME	Určuje název výchozího prostoru sbspace. Další informace naleznete v části “Řízení umístění ukládaných dat” na stránce 10-11.
SBSPECETEMP	Určuje název výchozího dočasného prostoru sbspace. Další informace naleznete v části “Dočasné prostory sbspace” na stránce 10-25.
SYSSBSPECENAME	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 zotavení ze selhání systému a podobně. Logické protokolování popisují následující parametry.

Konfigurační parametr	Popis
DYNAMIC_LOGS	Určuje, zda má databázový server přidělovat nové soubory logického protokolu automaticky. Další informace naleznete v části Kapitola 14, “Logický protokol”, na stránce 14-1.
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 8-15.
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 15-2.
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 15-20.
LTXEHWM	Určuje bod, od kterého odvolávaná dlouhá transakce získá výhradní přístup k logickému protokolu.

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 dbspace, do kterého bude ukládán fyzický protokol.
PHYSFILE	Určuje velikost fyzického protokolu.

Parametry odvolávání transakcí a zotavení

Informace o používání následujících parametrů ovlivňujících zotavení a odvolávání transakcí naleznete v příručce *IBM Informix: Dynamic Server Performance Guide*. Informace naleznete také v části Kapitola 16, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 16-1.

Konfigurační parametr	Popis
FAST_RESTART_CKPT_FUZZYLOG	Povoluje vyprazdňování záznamů tabulky DPT (Dirty-Pages Table) do fyzického protokolu v kontrolních bodech typu fuzzy v průběhu zotavení ve fázi přehrávání žurnálu. Tento parametr zkracuje dobu zotavení podobně jako parametr FAST_RESTART_PHYSLOG.
FAST_RESTART_PHYSLOG	Tento parametr umožňuje databázovému serveru provádět fyzické protokolování v kontrolních bodech typu fuzzy v průběhu zotavení ve fázi přehrávání žurnálu.

Parametry zálohování a obnovení

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 1-28.
MSGPATH	Určuje název cesty k souboru protokolu zpráv databázového serveru. Další informace naleznete v části "Protokol zpráv" na stránce 1-26.

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
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 8, “Sdílená paměť”, na stránce 8-1.

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	<p>Určuje informace o společné oblasti vyrovnávacích 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 11-14.</p> <p>Informace, které se dříve zadávaly pomocí konfiguračních parametrů BUFFERS, LRUS, LRU_MAX_DIRTY a LRU_MIN_DIRTY se ve verzi 10.0 databázového serveru určují pomocí konfiguračního parametru BUFFERPOOL.</p>
CKPTINTVL	Určuje maximální časový interval, který smí uplynout před dosažením kontrolního bodu. Další informace naleznete v části Kapitola 16, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 16-1.
DD_HASHMAX	Určuje maximální počet položek každého sektoru hashovací tabulky v rychlé vyrovnávací paměti datového slovníku. Další informace o nastavení parametru DD_HASHMAX naleznete v příručce <i>IBM Informix: Performance Guide</i> .
DD_HASHSIZE	Určuje počet sektorů hashovací tabulky v rychlé vyrovnávací paměti datového slovníku. Další informace o nastavení parametru DD_HASHSIZE naleznete v příručce <i>IBM Informix: Performance Guide</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 zámek dostupných pro uživatelské procesy databázového serveru při zpracování transakcí. Další informace uvádí Kapitola 9, “Správa sdílené paměti”, na stránce 9-1 a příručka <i>IBM Informix: Dynamic Server Performance Guide</i> .
PC_POOLSIZE	Určuje počet rutin UDR (rutin jazyka SPL i externích rutin), které lze uložit v rychlé vyrovnávací paměti rutin URD. Další informace o nastavení parametru PC_POOLSIZE naleznete v příručce <i>IBM Informix: Dynamic Server Performance Guide</i> .
PC_HASHSIZE	Určuje počet sektorů hashovací tabulky v rychlé vyrovnávací paměti rutin UDR. Další informace o nastavení parametru PC_HASHSIZE naleznete v příručce <i>IBM Informix: Dynamic Server Performance Guide</i> .
RESIDENT	Určuje, zda má být vynucena rezidence sdílené paměti. Další informace naleznete v části Kapitola 9, “Správa sdílené paměti”, na stránce 9-1 a v příručce <i>IBM Informix: Dynamic Server Performance Guide</i> .
STACKSIZE	Určuje velikost zásobníku uživatelských jednotkových procesů databázového serveru. Informace týkající se používání zásobníků naleznete v části “Zásobníky” na stránce 8-24.

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 8-34.
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 8-33.

Využití rychlé vyrovnávací paměti příkazů SQL

Ke konfiguraci rychlé vyrovnávací paměti příkazů jazyka SQL použijte následující parametry. Další informace naleznete v části “Nastavení parametrů rychlé vyrovnávací paměti příkazů jazyka SQL” na stránce 9-7.

Konfigurační parametr	Popis
STMT_CACHE	Zapne, povolí nebo zakáže rychlou vyrovnávací paměť příkazů jazyka SQL v paměti. Pokud je parametr zapnut, určuje, zda má rychlá vyrovnávací paměť příkazů jazyka SQL uchovávat analyzované a optimalizované příkazy SQL.
STMT_CACHE_SIZE	Určuje velikost rychlé vyrovnávací paměti příkazů jazyka SQL.
STMT_CACHE_HITS	Určuje počet použití příkazu (odkazů na příkaz), po kterém bude příkaz zcela vložen do rychlé vyrovnávací paměti příkazů jazyka SQL.
STMT_CACHE_NOLIMIT	Řídí, zda mají být příkazy jazyka SQL vkládány do rychlé vyrovnávací paměti příkazů i po překročení velikosti určené hodnotou parametru STMT_CACHE_SIZE.
STMT_CACHE_NUMPOOL	Definuje počet společných oblastí rychlé vyrovnávací paměti 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 *IBM Informix: Dynamic Server Performance Guide*.

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 (SHMVIRTSIZE - 10 MB).
DS_NONPDQ_QUERY_MEM	Umožňuje zvýšit velikost paměti dostupné jiným dotazům, než jsou dotazy PDQ.
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:

- MULTIPROCESSOR
- SINGLE_CPU_VP
- VPCLASS

Pravidla nastavování těchto parametrů naleznete v části “Nastavení konfiguračních parametrů virtuálních procesorů” na stránce 7-1.

Konfigurační parametr	Popis
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. Doporučujeme místo parametrů NOAGE, NUMCPUVPS, NUMAIOVPS, AFF_NPROCS a AFF_SPROCS používat parametr VPCLASS. Další informace naleznete v části “Určení parametru VPCLASS” na stránce 7-2.

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 *IBM Informix: Dynamic Server Performance Guide*.

Konfigurační parametr	Popis
OFF_RECVRY_THREADS	Určuje počet jednotkových procesů obnovení, které server používá při studeném obnovení.
ON_RECVRY_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 20, “Replikace HDR (High-Availability Data Replication)”, na stránce 20-1.

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 událostí 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 událostí.

Konfigurační parametr	Popis
ALRM_ALL_EVENTS	Určuje, zda má být program ALARMPROGRAM spouštěn pro všechny události protokolované v souboru určeném parametrem MSGPATH nebo zda má být spouštěn jen pro určité významné události.
ALARMPROGRAM	Určuje umístění souboru, který bude spuštěn, když dojde ke vzniku alarmu událostí. 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 (systém 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 22-4.

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>IBM Informix: Dynamic Server Performance Guide</i> .
EXT_DIRECTIVES	Povolí externí direktivy. Další informace naleznete v příručce <i>IBM Informix: Dynamic Server Administrator's Reference</i> a v příručce <i>IBM Informix: Dynamic Server Performance Guide</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 části “Omezení zahlcovacích útoků typu DOS (operační systém UNIX)” na stránce 5-8.

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 Kapitola 5, “Zabezpečení”.

Konfigurační parametr	Popis
IFX_EXTEND_ROLE	Umožňuje administrátorovi databázového serveru (DBSA) zabránit neoprávněným uživatelům registrovat uživatelské rutiny (UDR) modulů DataBlade.

Zvláštní parametry

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

Parametry prověřování (systém UNIX)

Parametry prověřování používejte pouze tehdy, pokud použitý databázový server podporuje funkce prověřování. Tyto parametry popisuje příručka *IBM Informix: Trusted Facility 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.
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 rychlé vyrovnávací paměti.
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 1-12. 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 proměnná prostředí ONCONFIG není nastavena, 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 (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	
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

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	3-5
Připojení podporovaná databázovým serverem	3-6
Místní připojení	3-7
Připojení prostřednictvím sdílené paměti (systém UNIX).	3-7
Proudové propojení procesů (systém UNIX)	3-8
Připojení prostřednictvím pojmenovaného propojení procesů (systém Windows).	3-8
Připojení prostřednictvím místní zpětné smyčky	3-9
Služby pro podporu komunikace	3-9
Propojovací soubory	3-10
Soubory konfigurace sítě	3-10
Propojovací soubory sítě TCP/IP	3-10
Více portů TCP/IP	3-12
Propojovací soubory IPX/SPX (systém UNIX)	3-12
Soubory zabezpečení sítě	3-13
Soubor hosts.equiv	3-13
Informace souboru netrc	3-14
Soubor sqlhosts a klíč registru SQLHOSTS	3-15
Soubor sqlhosts (systém UNIX)	3-15
Nástroje k aktualizaci informací SQLHOSTS	3-16
Klíč registru SQLHOSTS (systém Windows)	3-16
Informace sqlhosts.	3-18
Informace o propojitelnosti	3-19
Název databázového serveru	3-19
Pole typu připojení	3-19
Pole názvu hostitele	3-21
Pole názvu služby	3-22
Pole options.	3-24
Informace o skupinách	3-30
Skupina databázových serverů	3-30
Alternativy k připojení TCP/IP	3-30
Adresy IP při připojení prostřednictvím protokolu TCP/IP	3-30
Určení adresy připojení TCP/IP pomocí zástupných znaků	3-31
Čísla portů při připojení prostřednictvím protokolu TCP/IP.	3-34
Podpora adres IPv6 v serveru Dynamic Server	3-34
Parametry souboru ONCONFIG týkající se propojitelnosti	3-35
Konfigurační parametr DBSERVERNAME	3-36
Konfigurační parametr DBSERVERALIASES	3-36
Konfigurační parametr NETTYPE	3-37
Proměnné prostředí týkající se síťových připojení	3-37

Příklady konfigurací klientu a serveru	3-38
Připojení prostřednictvím sdílené paměti (systém UNIX)	3-38
Připojení prostřednictvím místní zpětné smyčky	3-39
Použití síťového připojení	3-40
Položka souboru sqlhosts pro protokol IPX/SPX (systém UNIX)	3-40
Použití více typů připojení	3-41
Přístup k více databázovým serverům	3-42
Použití programu IBM Informix MaxConnect	3-43

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, které 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 knihovnicích 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 pole `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 3-15.

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 3-40.

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. *Radič 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 3-9 a “Soubory zabezpečení sítě” na stránce 3-13.

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.

Tip: 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.

Pokud chcete nakonfigurovat databázový server k podpoře multiplexního připojení, zadejte do souboru ONCONFIG zvláštní parametr NETTYPE s hodnotou sqlmux jako v následujícím příkladu:

```
hodnota sqlmux parametru NETTYPE
```

Pokud chcete nakonfigurovat klienty tak, aby multiplexní připojení používaly automaticky, musí položka souboru **sqlhosts** nebo registru, pomocí které se klient připojuje k databázovému serveru, určovat v poli **options** hodnotu m=1 jako v následujícím příkladu:

```
menlo ontlitcp valley jfk1 m=1
```

Není zapotřebí provádět žádné změny souboru sqlhosts nebo 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 konfiguračním parametru NETTYPE naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix: Dynamic Server Administrator's Reference*. Další informace o souboru **sqlhosts** a registru naleznete v části "Soubor sqlhosts a klíč registru SQLHOSTS" na stránce 3-15.

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 omezit riziko agresivního útoku typu Denial of Service, protože je tak obtížnější zahltit virtuální procesor typu Listener, který zpracovává připojení. Další informace naleznete v části “Omezení zahlcovacích útoků typu DOS (operační systém UNIX)” na stránce 5-8.

Pouze 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 Pouze 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. Definujte 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:

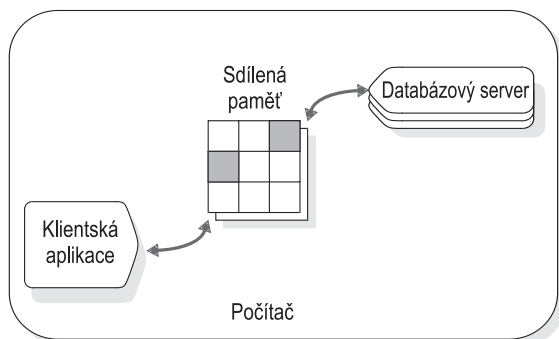
- “Propojovací soubory” na stránce 3-10
- “Informace sqlhosts” na stránce 3-18
- “Parametry souboru ONCONFIG týkající se propojitelnosti” na stránce 3-35
- “Proměnné prostředí týkající se síťových připojení” na stránce 3-37

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 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 připojení

procesů. Příklad připojení prostřednictvím sdílené paměti naleznete v části “Připojení prostřednictvím sdílené paměti (systém UNIX)” na stránce 3-38.

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 8-26. Další informace naleznete také v části “Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)” na stránce 8-8.

Proudové propojení procesů (systém UNIX)

Proudové propojení procesů je funkce meziprocesové komunikace (IPC) systému UNIX, která umožňuje procesům v jediném 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 3-20 a v příručce “Komunikace prostřednictvím sdílená paměti a proudového propojení procesů (systém UNIX)” na stránce 3-22.

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.

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 3-39.

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 3-13. 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). Podrobnosti naleznete v části “Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat” na stránce 5-11.

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. Ten 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 znaků. Tabulka 3-4 na stránce 3-31 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 znaků.
- 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 3-23 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.

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 DBSERVERALIASES. 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/SPX 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, kteří distribuují 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 úplný název domény jako v následujícím příkladu:

```
viking.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 **host.equiv** oba formáty názvu hostitele.

Abyste určili, zda je klient důvěryhodný, spusťte 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 oprávněn 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 obsluhovaného 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 netrc	machine trngpc3 login bruce password im4golf
příkaz CONNECT	CONNECT TO ol_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 3-18.

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	volby
menlo	onipeshm	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**.

Tip: Ke správě informací o propojitelnosti SQLHOSTS použijte program ISA.

Přestože obslužný program **setnet32** umožňuje nastavit databázové servery (pole nettype, hostname, servicename a volby), 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 téměř 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.

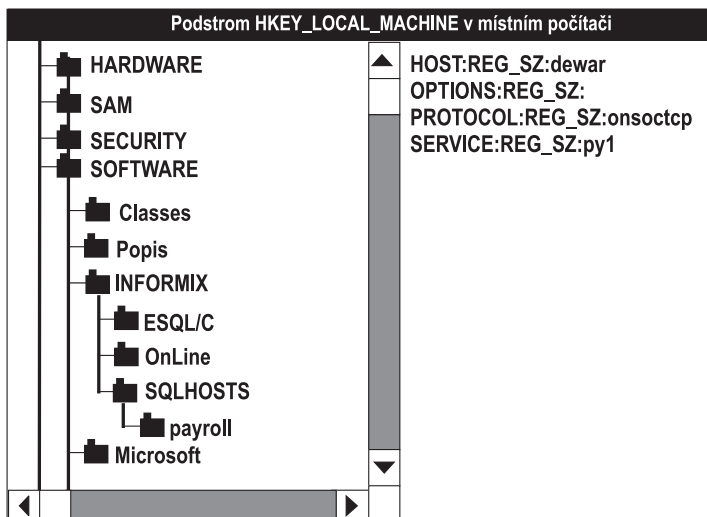
- Jiný počítač v síti, který slouží jako sdílené centrální úložiště informací **sqlhosts** pro několik databázových serverů v síti.

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 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 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 inicializuje, 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**.

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

Pouze pro UNIX

Pokud instalujete produkt IBM Informix Enterprise Gateway s architekturou DRDA do této 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 Pouze pro UNIX

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 3-35.

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 znaků.

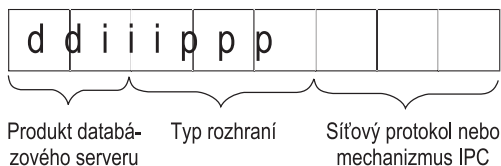
Pouze pro UNIX

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

Konec Pouze 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.



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 s architekturou DRDA

Další informace o architektuře DRDA naleznete v příručce
IBM Informix: Enterprise Gateway with DRDA User Manual.

Typ rozhraní: Prostřední 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

Typ rozhraní

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

Typ protokolu

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 3-43.

Tabulka 3-2. Přehled hodnot pole *nettype*

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

Informace o typech připojení pro použitou platformu naleznete v části “Připojení podporovaná databázovým serverem” na stránce 3-6.

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 znaků.

V následujících částech je popsán způsob, jakým klientské aplikace odvozuji 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 3-10.

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 3-30.

Komunikace prostřednictvím sdílená 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.

Tip: 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.

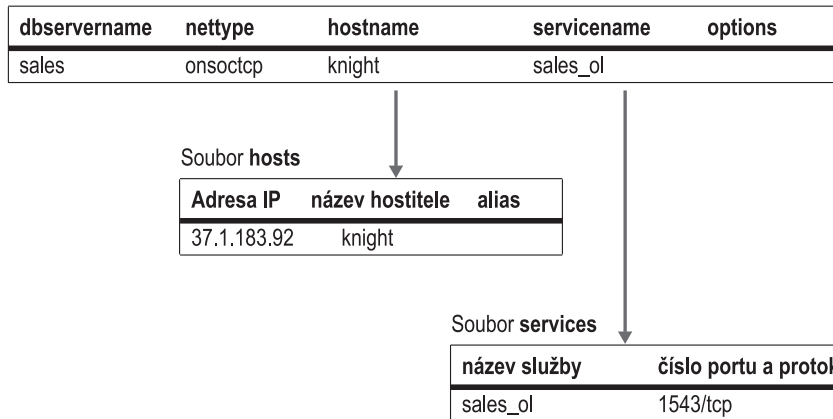
Pole názvu služby

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 znaků.

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 3-34.

Komunikace prostřednictvím pojmenovaného propojení procesů (systém

Windows): Pokud pole PROTOCOL určuje připojení prostřednictvím pojmenovaného propojení procesů (**onipcnmp**), 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 3-25
Přesměrování připojení	c	Strana 3-25
Konec skupiny	e	Strana 3-26
Skupina	g	Strana 3-27
Identifikátor	i	Strana 3-29
Volba keep-alive	k	Strana 3-29
Zabezpečení	s (databázový server) r (klient)	Strana 3-29
Modul pro podporu komunikace	csm	“Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat” na stránce 5-11

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. Pokud změníte volby databázového serveru, je zapotřebí znovu inicializovat 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 znaků.

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

Tip: Je doporučeno používat 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) přeměrovat připojení na záložní databázový server, pokud dojde k selhání primárního serveru.

Následující tabulka popisuje možná nastavení volby přeměrování připojení.

Nastavení	Účinek
c=0	Klientská aplikace se ve výchozím nastavení připojí k první instanci databázového serveru uvedené v souboru sqlhosts 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 přeměrování 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 3-27.

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 části “Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat” na stránce 5-11. Informace o položkách souboru **concsm.cfg** naleznete v části “Konfigurační soubor CSM” na stránce 5-12.

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 3-28.

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:

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

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 Data 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 20-19.

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.3**), hledá databázový server členy skupiny, dokud nenalezne server **asia.3**, takže do skupiny bude zahrnut i server **usa.2**. Protože skupina **peru** nepoužívá volbu konce skupiny, zahrne databázový server do této skupiny všechny členy až do konce souboru.

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.3
asia.1	ontlited	node6	svc8	g=asia
asia.2	onsoctcp	node0	svc1	g=asia
usa.2	ontlispix	node9	sv2	
asia.3	onsoctcp	node1	svc9	g=asia
peru	group	—	—	
peru.1	ontlited	node4	svc4	
peru.2	ontlited	node5	svc5	g=peru
peru.3	ontlited	node7	svc6	
usa.1	onsoctcp	37.1.183.92	sales_ol	k=1, s=0

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 je doporučeno 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.

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	Účinek
r=0	Zakáže vyhledávání informací netrc na straně klientu (nebude možné poskytnout heslo).
r=1	Povolí vyhledávání informací netrc na straně klientu (výchozí nastavení na straně klientu).
s=0	Zakáže na straně databázového serveru vyhledávání v souborech hosts.equiv a rhosts (budou povolena pouze příchozí připojení s použitím hesla).
s=1	Povolí na straně databázového serveru pouze vyhledávání v souboru hosts.equiv .

s=2 Povolí na straně databázového serveru pouze vyhledávání v souboru **rhosts**.

s=3 Povolí na straně databázového serveru vyhledávání v souborech **hosts.equiv** i **rhosts** (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 3-27.

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. Tabulka 3-4 na stránce 3-31 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_ol
sales  ontlitcp  knight      sales_ol
```

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.

Pouze pro UNIX

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

Konec Pouze pro UNIX

Jen pro Windows

Systém 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:

- službu Windows Internet Name Service,
- službu 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 síť 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 (NICs), jak znázorňuje Obrázek 3-5 na stránce 3-33, musí v souboru **hosts** být položka pro každou kartu rozhraní. Soubor **hosts** počítače **texas** může například obsahovat následující položky.

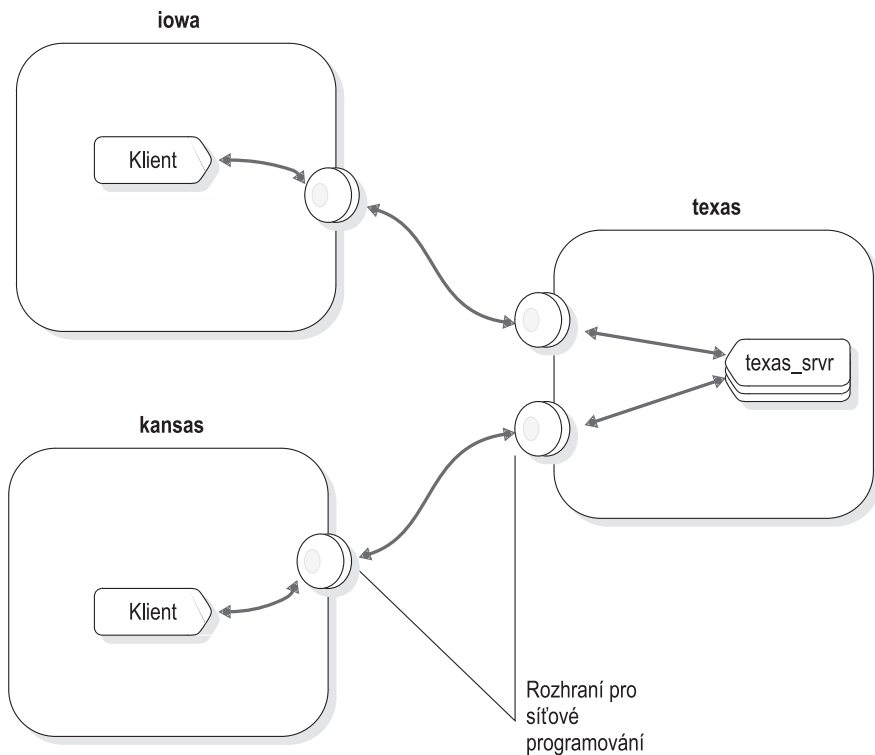
Karta NIC	Adresa IP	Název hostitele
Karta 1	123.45.67.81	texas1
Karta 2	123.45.67.82	texas2

Zástupný znak (*) můžete použít samostatně nebo jako předponu názvu hostitele nebo adresy IP, jak znázorňuje Tabulka 3-5 na stránce 3-33.

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 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

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.

Tip: V zájmu přehlednosti a snadné správy je doporučeno 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	ontlitedp	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 následujících platformách:

Tabulka 3-6.

Platforma	Verze s podporou adres IPv6
Solaris	Solaris 8 a novější
AIX	Server Dynamic Server: AIX 5.2 a novější Sada CSDK 2.90.UC1: AIX 5.1

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

Server Dynamic Server využívá mechanismus dvojí sady softwaru TCP/IP, který umožňuje spolupráci prostředí IPv4 a IPv6. Aby mohl databázový server s podporou adres IPv6 spolupracovat s aplikacemi bez podpory adres IPv6, musí mít nakonfigurovanu také adresu IPv4.

Pokud do hostitele s podporou adres IPv6 instalujete verzi serveru Dynamic Server, která nepodporuje adresy IPv6 (verze starší než 10.00.UC1), je zapotřebí nakonfigurovat pro hostitelský počítač adresu IPv4. Klientské aplikace by se k serveru měly připojovat prostřednictvím adresy IPv4, protože starší databázové servery nepodporují adresy IPv6. Pokud je server Dynamic Server verze 10.0 prostřednictvím souboru SQLHOSTS nakonfigurován tak, aby očekával připojení na adrese IPv6, nebudou starší klientské produkty a aplikace bez podpory adres IPv6 s tímto databázovým serverem schopny komunikovat.

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 inicializujete databázový server, budou při inicializaci 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 snadno změnit prostřednictvím konfiguračního souboru, protože informace o instanci databázového serveru ukládá v klíči DBSERVERNAME registr.

Konec Jen pro Windows

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 soketů TCP/IP pomocí následujícího příkazu:

```
CONNECT TO '@sockets_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`, `ipcstr`, `ipcnmp`, `soctcp`, `tlitcp` a `tlisp`) zadána samostatná položka NETTYPE.

Přestože parametr NETTYPE není požadovaný, je doporučeno 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 6-26. 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 8-8.

Proměnná prostředí **INFORMIXSERVER** umožňuje 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*.

Příklady konfigurací klientu a serveru

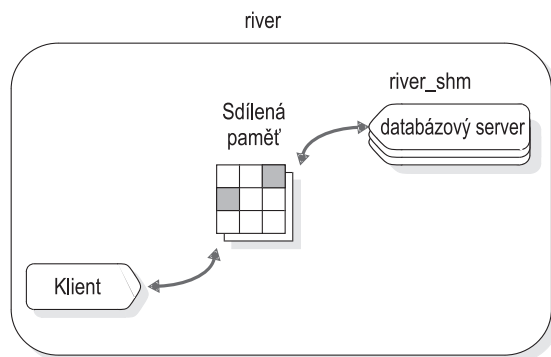
V následujících částech jsou uvedeny správné položky souboru **sqlhosts** a registru SQLHOSTS pro několik připojení klientů k serverům. Jednotlivé příklady představují:

- 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.

Připojení prostřednictvím sdílené paměti (systém UNIX)

Obrázek 3-8 představuje připojení prostřednictvím sdílené paměti 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 3-8.

Tabulka 3-7. Položka informací **sqlhosts**

dbservername	nettype	hostname	servicename
river_shm	onipcshm	river	rivershm

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

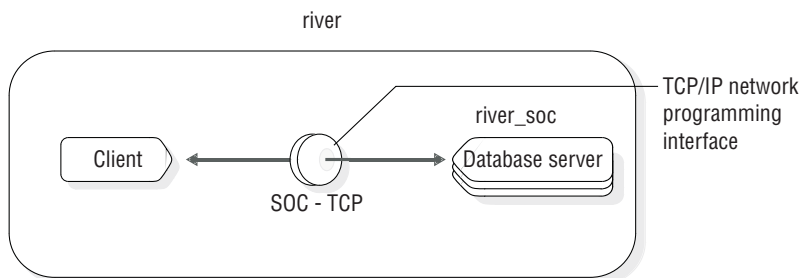
```
CONNECT TO '@river_shm'
```

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 8-8.

Připojení prostřednictvím místní zpětné smyčky

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 3-9.

Tabulka 3-8. Položka **sqlhosts**

dbservername	nettype	hostname	servicename
river_soc	onsoctcp	river	riverol

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 **ontlntcp**, nikoli **onsoctcp**.

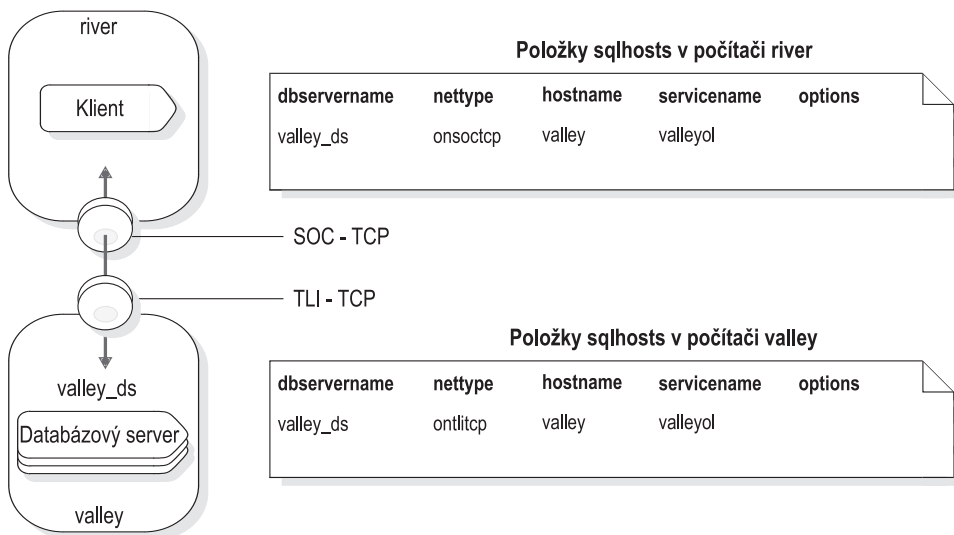
Soubor **ONCONFIG** obsahuje následující řádek:

```
DBSERVERNAME river_soc
```

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

Použití síťového připojení

Obrázek 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.

Pouze 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 Pouze pro UNIX

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

Software IPX/SPX často poskytuje rozhraní TLI. Tabulka 3-9 znázorňuje položky souboru **sqlhosts** v obou počítačích platné pro případ, kdy konfigurace, kterou znázorňuje Obrázek 3-10 na stránce 3-40, používá rozhraní IPX/SPX namísto rozhraní TCP/IP.

Tabulka 3-9. Položka `sqlhosts`

dbservername	nettype	hostname	servicename
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

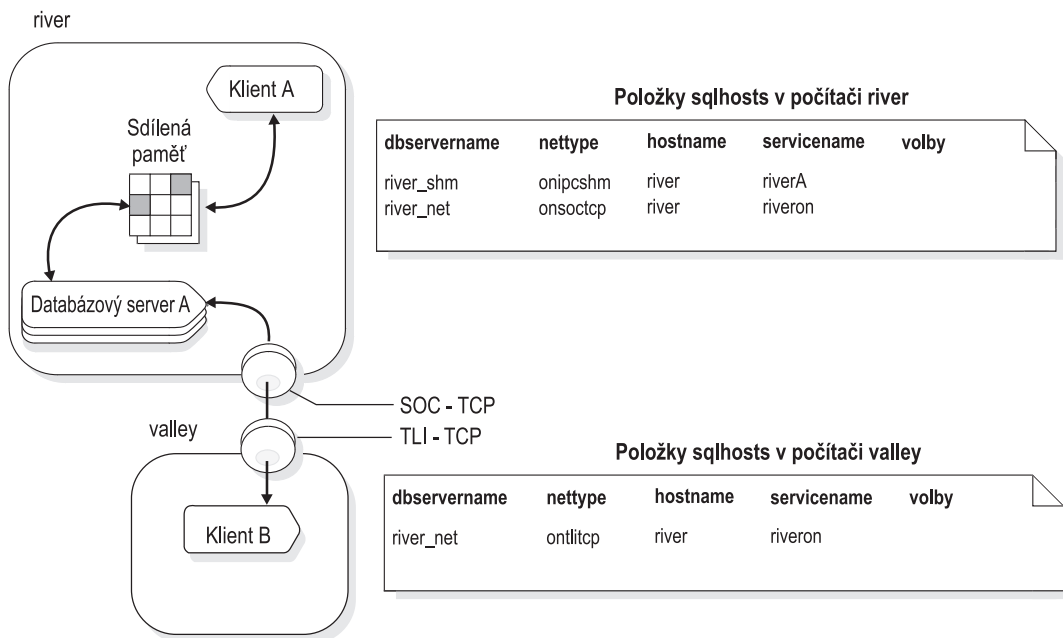
Instance databázového serveru může podporovat více typů připojení. Obrázek 3-11 znázorňuje takovou konfiguraci. Databázový server se nachází v hostiteli **river**. Klient A se k databázovému serveru připojuje prostřednictvím sdílené paměti, protože sdílená paměť je rychlá. Klient B musí použít síťové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

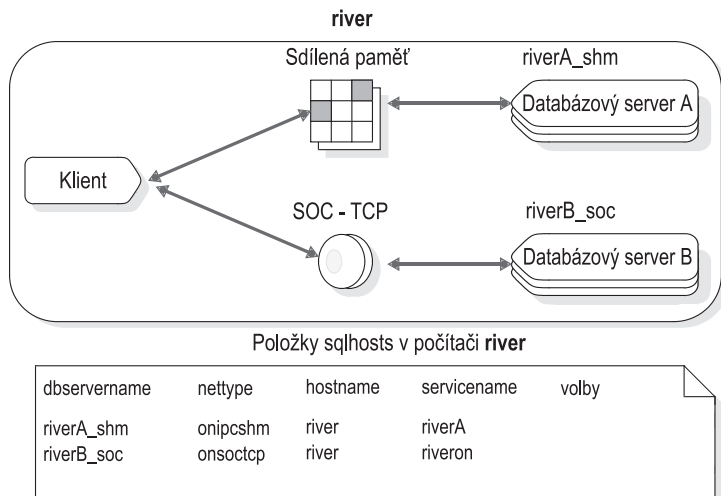
Klient B použije k připojení k databázovému serveru následující příkaz:

```
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 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: Příručka pro instalaci*.)



Obrázek 3-12. Několik databázových serverů v systému UNIX

Konfigurace, kterou znázorňuje 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 **ontliimc** a **onsocimc**. Protokoly **ontliimc** a **onsocimc** 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

Typy inicializace	4-2
Inicializace diskového prostoru	4-2
Jednotlivé kroky inicializace	4-3
Zpracování konfiguračního souboru	4-4
Vytvoření segmentů sdílené paměti	4-5
Inicializace sdílené paměti	4-5
Inicializace diskového prostoru	4-6
Spuštění všech požadovaných virtuálních procesorů	4-6
Provedení potřebných konverzí	4-6
Zahájení rychlé obnovy	4-6
Zahájení kontrolního bodu	4-6
Zaznamenání změn konfigurace	4-7
Vytvoření souboru oncfg_servername.servnum	4-7
Vypuštění dočasných prostorů tblspace	4-7
Nastavení vynucené rezidence, je-li určena	4-7
Návrat řízení uživateli	4-7
Vytvoření databáze sysmaster a příprava tabulek SMI	4-8
Vytvoření databáze sysutils	4-8
Monitorování maximálního počtu uživatelských připojení	4-8
Provozní režimy databázového serveru	4-9
Změna provozních režimů databázového serveru	4-10
Uživatelé oprávnění měnit režimy	4-10
Volby programu ISA pro změnu provozních režimů	4-11
Volby programu ON-Monitor pro změnu režimů (UNIX)	4-11
Volby příkazového řádku pro změnu režimů	4-12
Z režimu offline na klidový	4-12
Z režimu offline do režimu online	4-12
Z režimu offline na jednouživatelský režim	4-13
Z klidového režimu do režimu online	4-13
Nenásilně z režimu online do klidového	4-13
Okamžitě z režimu online do klidového	4-14
Z klidového režimu nebo režimu online na jednouživatelský režim	4-14
Z jednouživatelského režimu do režimu online	4-14
Z jednouživatelského režimu na klidový	4-15
Z kteréhokoliv režimu okamžitě do režimu offline	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.

Inicializace sdílené paměti se liší od inicializace disku ve dvou zásadních rysech:

- Inicializace sdílené paměti nemá vliv na přidělení nebo rozvržení diskového prostoru. Nedochozí při ní ke zničení dat.
- Inicializace sdílené paměti provede rychlou obnovu.

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 informix nebo root . Spusťte příkaz oninit -iy .
Windows	Chcete-li inicializovat databázový server, musíte být členem skupiny Administrators nebo Power Users . <ul style="list-style-type: none">• Databázový server pracuje jako služba. V ovládacím panelu Služby vyberte v poli Parametry pro spuštění službu databázového serveru a jeho typ -iy. Pak klepněte na Start.• Do příkazového řádku vložte příkaz starts dbservername -iy.

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.

K inicializaci databázového serveru doporučujeme nepoužívat příkaz **oninit -iy**, s výjimkou případů, kdy diagnostikujete případný problém.

Konec Jen pro Windows

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 1-16.

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.

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 oncfg_servername.servernum .	Aktualizace souboru oncfg_servername.servernum .
Změna na klidový režim.	Změna na klidový režim.
Vypuštění dočasných prostorů tbspaces (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 sysmaster obsahující tabulky SMI.
	Vytvoření databáze sysutils .
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. 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/atd**. na platformě UNIX a v adresáři **%INFORMIXDIR%\atd**. 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**.

Doporučujeme Vám, aby jste před *každou* inicializací databázového serveru nastavili 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 1-12.

Proces inicializace porovná hodnoty v aktuálním konfiguračním souboru s předchozími hodnotami, uloženými na stránce vyhrazené kořenovému prostoru dbspace, PAGE_CONFIG, jsou-li takové hodnoty k dispozici. Vyskytnou-li se rozdíly, použije databázový server k inicializaci 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 využívaná pro vyrovnávací paměť dat a interní tabulky
- Virtuální část využívaná pro většinu systémových paměťových požadavků a požadavků uživatelských relací

Pouze pro UNIX

- Komunikační část využívaná pro komunikaci typu IPC

Databázový server přidělí tuto část sdílené paměti jen v případě, že konfiguruje připojení typu IPC prostřednictvím sdílené paměti.

Konec Pouze 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 8-18.

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 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. 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, například, může ovlivnit počet a typ procesorů, které vytvářejí připojení. Více informací o virtuálních procesorech naleznete v části “Virtuální procesory” na stránce 6-2.

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 16-14.

Rychlá obnova není provedena během inicializace diskového prostoru, protože inicializovaný disk neobsahuje žádná data k obnovení.

Zahájení kontrolního bodu

Po provedení rychlé obnovy zahájí databázový server úplný kontrolní bod. 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 16-8.

Databázový server se nyní uvede do klidového režimu nebo do režimu online, v závislosti na tom, jak jste spustili proces inicializace.

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 během inicializace diskového prostoru neprovádí.

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/atd`, na platformě UNIX nebo v adresáři `%INFORMIXDIR%\atd`, v systému Windows. Databázový server využívá tento soubor během úplného obnovení systému.

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, zdali 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 10-33.

Tato úloha není provedena během inicializace 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`. 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 přeruší. 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. *Nedoporučujeme* zastavovat databázový server odstraněním virtuálního procesoru nebo jiného procesu databázového serveru. Více informací naleznete v části “Spuštění a zastavení virtuálních procesorů” na stránce 7-3.

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**. Více informací naleznete v části *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.

Tato zpráva pomůže zákazníkům zaznamenat využití licence a zjistit, kdy je nutné zakoupit větší množství licencí. 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. Záhloví **onstat** zobrazí příslušný režim.

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 informix) může přepnout z tohoto režimu na jiný.
Klidový režim	Jsou spuštěny procesy databázového serveru a přidělené zdroje sdílené paměti. Administrátoři využívají tento režim k provedení udržovacích funkcí, které nevyžadují spuštění příkazů jazyka SQL a DDL.	Na databázový server může přistupovat jen administrátor (uživatel informix). Jiní uživatelé mohou zobrazit údaje o stavu databázového serveru, ale nemohou na něj přistupovat.
Jednouživatelský režim	Tento režim je přechodným režimem mezi Klidovým režimem a režimem Online. 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.	Na databázový server může přistupovat jen administrátor (uživatel informix). 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. Uživatelé informix nebo root mohou využít obslužných programů příkazového řádku a změnit celou řadu hodnot parametrů databázového serveru ONCONFIG.

Kromě toho se může databázový server nacházet v jednom z následujících režimů:

- *Režim pouze pro čtení* je využíván sekundárním databázovým serverem v replikačním páru HDR. Aplikace mohou odesílat dotazy na databázový server v režimu pouze pro čtení, ale nemohou zapisovat do databáze určené pouze ke čtení.
- *Režim zotavení* 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**.

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, spusťte 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 příkazNET a nástroj Server Manager. Více informací o těchto metodách naleznete v dokumentaci operačního systému Windows.

Konec Jen pro Windows

Tip: Poté, co změníte režim databázového serveru, proveďte příkaz **onstat** a prověřte aktuální stav serveru.

Uživatelé oprávnění měnit režimy

Pouze pro UNIX

Provozní režim databázového serveru mohou měnit jen uživatelé přihlášení jako uživatel **root** nebo **informix**.

Konec Pouze pro UNIX

Jen pro Windows

Tabulka 4-2 na stránce 4-9 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 spouštějí		X
ISA		X
Ovládací panel Služby	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 online nápovědě k programu ISA.

Akce	Volba programu ISA
Inicializace databázového serveru.	Režim > Online (inicializace disků) nebo Klidový (inicializace disků)
Změna z režimu offline nebo jednouživatelského režimu na klidový režim.	Režim > Klidový
Změna z kteréhokoliv režimu na jednouživatelský režim.	Režim > Jednouživatelský
Změna z režimů offline, klidového nebo jednouživatelského režimu na režim online.	Režim > Online
Nenásilná změna z režimu online do klidového režimu.	Režim > Klidový (nenásilný)
Okamžitá změna z režimu online do klidového režimu.	Režim > Klidový (okamžitý)
Vypne databázový server.	Režim > Offline

Volby programu ON-Monitor pro změnu režimů (UNIX)

Obslužný program ON-Monitor můžete použít ke změně režimu databázového serveru v systému UNIX. Více informací naleznete v části věnované obslužnému programu ON-Monitor v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Akce	Volba nabídky
Inicializace databázového serveru.	Parametry > Inicializace
Změna z režimu offline do klidového režimu.	Režim > Startup
Změna z režimu offline nebo klidového režimu na režim online.	Režim > Online
Nenásilná změna z režimu online na klidový režim.	Režim > Nenásilné vypnutí
Okamžitá změna z režimu online na klidový režim.	Režim > Okamžité vypnutí
Změna na jednovýživatelový režim.	Režim > Jeden uživatel
Vypnutí databázového serveru.	Režim > Přepnout na offline

Volby příkazového řádku pro změnu režimů

Tato část obsahuje příkazy ke změně provozních režimů a informace o tom, jak změny režimů ovlivňují relace uživatelů. Tabulka 4-2 na stránce 4-9 obsahuje popisy všech režimů a znázorňuje, kteří uživatelé mohou přistupovat na databázový server v jednotlivých jeho provozních režimech.

Z režimu offline na klidový

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

Akce

UNIX

- Proveďte příkaz **oninit -s**.

Windows

- V příkazovém řádku použijte příkaz **starts dbservername -s**.

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

Akce

UNIX

- Proveďte příkaz **oninit**.

Windows

- V nástroji Služby vyberte službu databázového serveru a klepněte na příkaz **Start**.
- V příkazovém řádku použijte příkaz **starts dbservername**.

Z režimu offline na jednouživatelský režim

Když přepínáte databázový server z režimu offline do jednouživatelského režimu, 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**.

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**.

Nenásilně z režimu online do klidového

Databázový server převedte nenásilně z režimu online do klidového režimu, chcete-li omezit přístup uživatelů bez přerušení 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 iniciujete změnu režimu, nelze tuto 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**.

Okamžitě z režimu online do klidového

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 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.

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.

Z klidového režimu nebo režimu online na jednouživatelský režim

Přepnete-li databázový server z klidového režimu nebo režimu online na režim jednouživatelský, 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 jednouživatelský režim přepněte, chcete-li provést příkazy jazyka SQL a DLL a zároveň nechcete, aby byli připojeni další uživatelé.

Operační systém

Akce

UNIX nebo Windows

- Proveďte příkaz **onmode -j**.

Z jednouživatelského režimu do režimu online

Když databázový server přepnete z jednouživatelského režimu na režim online, mohou na databázový server přistupovat všichni uživatelé.

Operační systém

Akce

UNIX nebo Windows

- Proveďte příkaz **onmode -m**.

Z jednoruživatelského režimu na klidový

Když přepínáte databázový server z jednoruživatelského 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**.

Z kteréhokoliv režimu okamžitě 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 nebo jednoruživatelském režimu 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 sekund, 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 nebo jednoruživatelském režimu 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**.

Kapitola 5. Zabezpečení

Zabezpečení prováděné obslužnými programy serveru před spuštěním v operačním systému UNIX a Linux	5-2
Zakázání kontroly zabezpečení	5-3
Zabezpečení v nedostatečně chráněném prostředí	5-4
Varování a chybové zprávy pro kontrolu zabezpečení prováděnou obslužnými programy	5-4
Oprávnění adresáře INFORMIXDIR.	5-5
Uživatelé a vlastnictví skupin pro spuštění obslužných programů serveru Dynamic Server.	5-6
Audit událostí v databázi	5-7
Role uživatelů	5-7
Výchozí role	5-7
Omezení zahlcovacích útoků typu DOS (operační systém UNIX)	5-8
Použití příkazů LISTEN_TIMEOUT a MAX_INCOMPLETE_CONNECTIONS	5-8
Použití voleb onmode -wm a onmode -wf ke změně hodnot parametrů	5-9
Šifrování	5-9
Přehled šifrování a voleb šifrování	5-9
Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat	5-11
Konfigurační soubor CSM	5-12
Konfigurace šifrování hesel	5-12
Konfigurace šifrování síťových dat	5-14
Použití šifrování na úrovni sloupců	5-23
Pokyny týkající se paměti	5-25
Šifrování dat sloupce	5-25
Příklad ilustrující jak určit velikost šifrovaného sloupce.	5-25
Příklad ilustrující způsob šifrování sloupce	5-26
Příklad ilustrující dotazování na šifrovaná data	5-27
Ověřovací moduly	5-27
Moduly PAM (Pluggable Authentication Modules) pro systémy spouštěné v operačním systému UNIX nebo Linux	5-27
Podporované platformy	5-28
Název služby modulu PAM	5-28
Režim ověřování	5-29
Požadovaná velikost zásobníku	5-29
Implicitní připojení	5-29
Konfigurace databázového serveru k používání modulu PAM.	5-29
Replikace Enterprise Replication a replikace HDR (High-Availability Data Replication)	5-30
Vývoj aplikací	5-30
Distribuované transakce s modulem PAM, klientská rozhraní API a problémy kompatibility	5-30
Podpora ověřování LDAP v operačním systému Windows	5-30
Instalace a přizpůsobení modulu LDAP Authentication Support Module	5-30
Konfigurace modulu LDAP	5-31
Konfigurace serveru Dynamic Server	5-31
Režim ověřování	5-32
Implicitní připojení	5-32
Replikace Enterprise Replication a replikace HDR (High-Availability Data Replication)	5-32
Vývoj aplikací	5-32

Distribuované transakce s modulem LDAP Authentication Support Module, klientská rozhraní API a problémy kompatibility	5-32
Vývoj aplikací pro moduly ověřování	5-33
Distribuované transakce	5-35
Klientská rozhraní API a moduly na podporu ověřování	5-35
Problémy s kompatibilitou	5-36
Volby zabezpečení připojení replikace Enterprise Replication a replikace HDR	5-37
Zabezpečení externích uživatelských rutin (UDR)	5-38

Obsah kapitoly

Tato kapitola obsahuje informace o těchto funkcích zabezpečení databázového serveru:

- Kontrola prováděná obslužnými programy zajišťující zabezpečené prostředí před spuštěním serveru Dynamic Server v operačním systému UNIX nebo Linux.
- Výchozí role, kterou můžete pro určitou databázi přiřadit jednotlivým uživatelům nebo skupině PUBLIC.
- Konfigurační parametry, které můžete použít ke snížení doby prodlevy nedokončeného připojení a k omezení počtu nedokončených požadavků na připojení, čímž dojde ke snížení rizika nepřátelského útoku typu DOS (denial-of-service).
- Volby šifrování hesel, šifrování přenosu dat a šifrování na úrovni sloupců.
- Modul PAM (Pluggable Authentication Module) pro systémy serveru Dynamic Server spouštěné v operačním systému UNIX nebo Linux.
- Podpora ověřování protokolu LDAP (Lightweight Directory Access Protocol) pro systémy serveru Dynamic Server spouštěné v operačním systému Windows.
- Volba zabezpečení připojení replikace Enterprise Replication a replikace HDR (High-Availability Data Replication).
- Volba zabezpečení, pomocí které můžete zabránit neoprávněným uživatelům v registraci externích rutin.

Zabezpečení prováděné obslužnými programy serveru před spuštěním v operačním systému UNIX a Linux

Při instalaci nové verze databázového serveru doporučujeme dodržet instalační pokyny, aby byla správně nastavena oprávnění všech klíčových souborů a adresářů.

Aby bylo dosaženo zvýšeného zabezpečení, provádějí klíčové obslužné programy serveru kontrolu, pomocí které zjišťují, zda je prostředí zabezpečené. Před spuštěním databázového serveru nesmí být následující nastavení změněna z nastavení vytvořených během instalace:

- Oprávnění k adresáři \$INFORMIXDIR a některým adresářům pod tímto adresářem. U každého adresáře databázový server kontroluje, zda tento adresář existuje, zda jej vlastní uživatel **informix** a správná skupina (jak je uvedeno v části “Oprávnění

adresáře INFORMIXDIR” na stránce 5-5) a zda oprávnění k adresáři nezahrnují oprávnění k zápisu pro skupinu nebo jiné uživatele.

- Oprávnění k souboru ONCONFIG.

Soubor musí patřit skupině DBSA (Database Server Administrator). Pokud skupina DBSA je skupina **informix** (výchozí skupina), musí být soubor ONCONFIG vlastněn uživatelem **informix**. V opačném případě není vlastnictví omezeno. Ostatní uživatelé nesmějí mít k tomuto souboru oprávnění k zápisu.

- Oprávnění k souboru **sqlhosts**.

Ve výchozí konfiguraci je soubor **sqlhosts** umístěn v adresáři **\$INFORMIXDIR/etc/sqlhosts**, vlastníkem by měl být uživatel **informix**, skupina by měla být **informix** nebo skupina DBSA a neměl by být povolen veřejný zápis do souboru. Pokud je soubor určen nastavením proměnné prostředí INFORMIXSQLHOSTS, pak se vlastníkem a skupinou nekontroluje, ale veřejná oprávnění k zápisu nejsou povolena.

- Délky názvů souborů.

Délka názvů souborů **\$INFORMIXDIR/etc/onconfig.std** a **\$INFORMIXDIR/etc/\$ONCONFIG** musí být menší než 256 znaků.

Pokud se ověření kterékoli z těchto podmínek nezdaří, obslužné programy se ukončí a vypíší chybovou zprávu.

Upozornění: Pokud není proměnná prostředí INFORMIXDIR v adresáři **/etc/informix** označena jako zabezpečená, nechte databázový server konfigurační soubory, je-li možné do nich veřejně zapisovat. Další informace naleznete v části “Zakázání kontroly zabezpečení” a “Zabezpečení v nedostatečně chráněném prostředí” na stránce 5-4.

Zakázání kontroly zabezpečení

Ačkoli důrazně doporučujeme, abyste **nikdy** nezakázali kontrolu zabezpečení, můžete částečně zakázat kontrolu zabezpečení pro databázový server uložený v určitém adresáři \$INFORMIXDIR. Pokud chcete zakázat kontrolu zabezpečení, spusíte jako uživatel **root** skript **INFORMIXDIR/etc/informixdir-is-insecure**. Po úspěšném spuštění skriptu se budou při zpracování obslužného programu zobrazovat varovné zprávy, ale program bude pokračovat.

Poznámka: V příkazovém řádku můžete určit hodnotu INFORMIXDIR jako argument skriptu. Nemusíte tedy nastavovat hodnotu INFORMIXDIR v prostředí uživatele **root**.

Skript **informixdir-is-insecure** vytvoří adresář **/etc/informix** (je-li to nutné), který je vlastněn uživatelem **root** a má udělená oprávnění 555. V tomto adresáři vytvoří skript soubor pojmenovaný **server-10.xx.yyy**, který má udělená oprávnění 444. Část **xx** názvu souboru je hlavní číslo verze a část **yyy** je číslo sady oprav, například **server-10.00.UC1**. V tomto souboru jsou uvedeny hodnoty \$INFORMIXDIR, pro které je zakázáno kontrolování zabezpečení.

Poznámka: Formát obsahu souborů **server-10.xx.yyy** se může v budoucích vydáních změnit.

Pokud se rozhodnete zakázat kontrolu zabezpečení, ačkoli je doporučováno, abyste tak nikdy nečinili, měli byste v systému pomocí skriptu **ibmifmx_security.sh** omezit počet programů SUID a SGID.

Zabezpečení v nedostatečně chráněném prostředí

Pokud databázový server hlásí, že prostředí již není zabezpečené a programy se ukončují, může uživatel **root** spuštěním skriptu **\$INFORMIXDIR/etc/make-informixdir-secure** znovu nastavit odpovídající oprávnění k adresářům tak, aby byly znovu zabezpečené.

Ačkoli uživatel **informix** má oprávnění ke spuštění skriptu, nemůže skript problémy opravit, dokud je adresář vlastněn uživatelem **informix**. Chybové zprávy oznamují, co je stále nutné opravit. Skript také udává soubory a adresáře pod adresářem **\$INFORMIXDIR**, které patří neočekávanému vlastníku či skupině nebo mají udělené veřejné oprávnění k zápisu.

Varování a chybové zprávy pro kontrolu zabezpečení prováděnou obslužnými programy

Pokud kontrola zabezpečení, kterou provádějí při spuštění obslužné programy serveru, zjistí problém, vrátí chybovou zprávu. Tyto zprávy jsou vráceny, když není dostupný soubor zpráv a podpora mezinárodních jazyků. Z tohoto důvodu nemají chybové zprávy čísla chyb a nejsou přeloženy.

V případě následujících problémů se zobrazí jedna z níže uvedených chybových zpráv a obslužný program bude pokračovat:

- **INFORMIXDIR or ONCONFIG is too long. Maximum length for \$INFORMIXDIR/etc/\$ONCONFIG is 255 characters.** (Proměnná **INFORMIXDIR** nebo **ONCONFIG** je příliš dlouhá. Maximální délka cesty **\$INFORMIXDIR/etc/\$ONCONFIG** je 255 znaků.)
- **INFORMIXSQLHOSTS is too long. Maximum length is 255 characters.** (Proměnná **INFORMIXSQLHOSTS** je příliš dlouhá. Maximální délka je 255 znaků.)
- **ONCONFIG not set; TBCONFIG set. TBCONFIG will not be supported in future.** (Proměnná **ONCONFIG** není nastavena, proměnná **TBCONFIG** je nastavena. Proměnná **TBCONFIG** nebude v budoucnu podporována.)

V případě následujících problémů se obslužný program ukončí a zobrazí se jedna z níže uvedených chybových zpráv:

- **User informix not found.** (Uživatel **informix** nebyl nalezen.)
- **Group informix not found.** (Skupina **informix** nebyla nalezena.)
- **Could not access logical-file filename.** (Nelze přistupovat k souboru *logical-file filename*.)

- *Logical-file filename* is not owned by user with id *UID*. (Soubor *logical-file filename* není vlastněn uživatelem s identifikátorem *UID*.)
- *Logical-file filename* not owned by group with id *GID*. (Soubor *logical-file filename* není vlastněn skupinou s identifikátorem *GID*.)
- *Logical-file filename* has insecure mode *mode*. (Soubor *logical-file filename* má nedostatečně zabezpečený režim *mode*.)
- Could not access *logical-file filename*. (Nelze přistupovat k souboru *logical-file filename*.)

Následující tabulka definuje proměnné používané ve výše uvedených zprávách.

Proměnná	Popis
<i>filename</i>	Název souboru nebo adresáře
<i>logical-file</i>	ONCONFIG, INFORMIXSQLHOSTS, INFORMIXDIR, nebo INFORMIXDIR/xxx (kde xxx jeden z podadresářů adresáře \$INFORMIXDIR). Pokud je například adresář INFORMIXDIR nastaven na /usr/informix, bude zpráva vypadat následovně: INFORMIXSQLHOSTS /usr/informix/etc/sqlhosts is not owned by the user with id 1234.
<i>mode</i>	Oktalová hodnota oprávnění
<i>UID</i>	Číslo
<i>GID</i>	Číslo

Oprávnění adresáře INFORMIXDIR

Následující tabulka uvádí adresáře \$INFORMIXDIR a jejich odpovídající vlastníky, skupiny a oprávnění.

Podadresář	Vlastník	Skupina	Oprávnění
. (\$INFORMIXDIR)	informix	informix	755
bin	informix	informix	755
lib	informix	informix	755
gls	informix	informix	755
msg	informix	informix	755
etc	informix	DBSA	775
aaodir	informix	AAO	775
dbssodir	informix	DBSSO	775

Další informace o administrátorovi databázového serveru (DBSA), správci analýzy auditu (AAO) a správci zabezpečení databázového systému (DBSOO) naleznete v příručce *IBM Informix: Trusted Facility Guide*.

Uživatelé a vlastnickví skupin pro spuštění obslužných programů serveru Dynamic Server

Následující obslužné programy serveru Dynamic Server mají nastaveno oprávnění SUID **root** a SGID **informix**:

- **onaudit,**
- **onbar_d,**
- **ondblog,**
- **onedcu,**
- **oninit,**
- **onmode,**
- **ON-Monitor,**
- **onshowaudit,**
- **onsmsync,**
- **onsnmp,**
- **onsrvapd,**
- **ontape,**
- **snmpdm.**

Následující obslužné programy serveru Dynamic Server mají nastaveno oprávnění SGID **informix**:

- **oncheck**
- **onedpdu**
- **onload**
- **onlog**
- **onparams**
- **onpload**
- **onspaces**
- **onstat**
- **onunload**
- **xtree**

Audit událostí v databázi

Můžete provádět audit událostí v databázi zaznamenáváním aktivit, které uživatelé provádějí s konkrétními objekty v různých časech. Pomocí těchto informací můžete monitorovat podezřelé používání databáze, zjišťovat pokusy o neoprávněný přístup, posoudit potenciální bezpečnostní rizika, zjišťovat bezohledné uživatele a shromažďovat důkazy o zneužívání databázového serveru.

Podrobnější informace týkající se auditu naleznete v příručce *IBM Informix: Trusted Facility Guide*.

Role uživatelů

Role je klasifikace pracovní úlohy, jako například mzdový účetní nebo vedoucí mzdové účtárny. Každá definovaná role má oprávnění k databázovému objektu, která jsou této roli udělena. K definování role můžete použít příkaz `CREATE ROLE`.

Po vytvoření role můžete pomocí příkazu `GRANT` udělit oprávnění jednomu nebo více uživatelům přidruženým k tomuto názvu role. Příkazy `CREATE ROLE` a `GRANT ROLE` umožňují pro roli vytvořit jednu sadu oprávnění a potom tuto roli udělit mnoha uživatelům namísto toho, abyste udělovali stejnou sadu oprávnění každému uživateli ve skupině.

Když je role uživateli udělena, musí ten, kdo roli udělil nebo ten kdo ji přijal (uživatel), pomocí příkazu `SET ROLE` roli aktivovat. Teprve poté bude mít uživatel oprávnění role.

Další informace o vytváření a používání rolí naleznete v příručce *IBM Informix: Guide to SQL Syntax*.

Výchozí role

Administrátor může pro konkrétní databázi definovat výchozí roli pro přiřazení jednotlivým uživatelům nebo skupině `PUBLIC`. Výchozí role se automaticky použije, když uživatel vytvoří spojení s databází. To umožní uživateli připojit se k databázi bez vydání příkazu `SET ROLE`.

Každý uživatel má libovolná oprávnění, která mu byla udělena individuálně, a rovněž oprávnění výchozí role. Uživatel může přepínat z aktuální individuální roli na výchozí roli pomocí příkazu `SET ROLE DEFAULT` statement.

Pokud jsou uživatelé a skupině `PUBLIC` uděleny různé výchozí role, má přednost výchozí role udělená uživateli. Pokud výchozí role není přiřazena uživateli, má uživatel pouze oprávnění přidělená individuálně a veřejná oprávnění.

Postup definování a udělení oprávnění pro výchozí roli:

1. Vyberte existující roli v aktuální databázi, kterou použijete jako výchozí roli, nebo vytvořte roli, kterou chcete použít jako výchozí roli. Pomocí příkazu `CREATE ROLE název_rol` vytvořte novou roli v aktuální databázi.
2. Pomocí příkazu `GRANT` udělte roli oprávnění.
3. Udělte roli uživateli a nastavte roli jako výchozí roli pro uživatele nebo skupinu `PUBLIC` pomocí syntaxe `GRANT DEFAULT ROLE název_rol TO jméno_uživatele` nebo `GRANT DEFAULT ROLE název_rol TO PUBLIC`.

Pomocí příkazu `REVOKE DEFAULT ROLE` zrušte přidružení výchozí role k uživateli.

Uživatel musí použít příkaz `SET ROLE DEFAULT`, pokud chce změnit jakoukoli jinou aktuální roli na výchozí roli.

Další informace o používání těchto příkazů naleznete v příručce *IBM Informix: Guide to SQL Syntax*.

Omezení zahlcovacích útoků typu DOS (operační systém UNIX)

Za účelem omezení útoků typu DOS (Denial-of-service) spouští server Dynamic Server několik jednotkových procesů typu listener (**listen_authenticate**). Tyto jednotkové procesy ověřují klientské požadavky, zatímco hlavní jednotkový proces typu listener pouze přijímá příchozí požadavky a spouští nové jednotkové procesy pro ověřování.

Pomocí konfiguračního parametru `MAX_INCOMPLETE_CONNECTIONS` můžete konfigurovat počet jednotkových procesů, které provádějí ověřování v každém bodě v čase.

Pomocí konfiguračního parametru `LISTEN_TIMEOUT` můžete nakonfigurovat hodnotu prodlevy pro nedokončená připojení.

K útokům typu DOS může dojít, pokud k připojení k portu rezervovanému pro databázový server použijete externí mechanismus, například program Telnet. Pokud například k připojení k portu rezervovanému pro službu databázového serveru použijete program Telnet, ale neodešlete data a k serveru se prostřednictvím aplikace (například DB-Access) pokusí připojit samostatná relace, bude jednotkový proces typu listener blokován čekáním na informaci z relace Telnet a nebude moci přijmout připojení k aplikaci použité v druhé relaci. Jestliže během čekací doby útočník zahájí distribuovaný útok typu DOS (DDOS) ve smyčce, můžete obdržet zahlcovací útok na připojení, který povede k nízkému výkonu připojení.

Použití příkazů `LISTEN_TIMEOUT` a `MAX_INCOMPLETE_CONNECTIONS`

Pokud chcete snížit riziko nepřátelských zahlcovacích útoků typu DOS, můžete přizpůsobit následující konfigurační parametry:

- `LISTEN_TIMEOUT`. Nastavuje délku časového limitu pro nedokončená připojení. Výchozí časový limit pro nedokončená připojení je 10 sekund.

- `MAX_INCOMPLETE_CONNECTIONS`. Omezuje počet nedokončených žádostí o připojení. Výchozí maximální počet nedokončených připojení je 1024.

Pokud nenastavíte konfigurační parametry `LISTEN_TIMEOUT` a `MAX_INCOMPLETE_CONNECTIONS` a nastane zahlcení způsobené neoprávněnými útoky, může se procesor VP typu listener stát nedostatečně chráněný a nemusí být schopen včas naslouchat platnému požadavku.

Jestliže nastavíte konfigurační parametry `LISTEN_TIMEOUT` a `MAX_INCOMPLETE_CONNECTIONS` a někdo se pokusí proniknout do systému a dosáhne maximálního zadaného limitu, zobrazí se v online protokolu zpráv následující informace oznamující, že je systém vystaven útoku:

```
%d incomplete connection at this time.
System is under attack through invalid clients
on the listener port.
```

V závislosti na schopnosti počítače provozovat jednotkové procesy (v určitém počtu) můžete nastavit konfigurační parametr `MAX_INCOMPLETE_CONNECTIONS` na vyšší hodnotu a v závislosti na provozu sítě můžete nastavit konfigurační parametr `LISTEN_TIMEOUT` na nižší hodnotu, a tak snížit pravděpodobnost, že útok dosáhne maximálního limitu.

Použití voleb `onmode -wm` a `onmode -wf` ke změně hodnot parametrů

Když je databázový server v režimu online, můžete pomocí voleb `onmode -wm` a `onmode -wf` změnit hodnoty konfiguračních parametrů `LISTEN_TIMEOUT` a `MAX_INCOMPLETE_CONNECTIONS`.

Pokud chcete zadat novou hodnotu pro libovolný z těchto konfiguračních parametrů pro aktuální relaci, použijte tuto syntaxi:

```
onmode -wm konfigurační_parametr=hodnota
```

Chcete-li změnit hodnotu libovolného z těchto konfiguračních parametrů v souboru `ONCONFIG`, použijte tuto syntaxi:

```
onmode -wf konfigurační_parametr=hodnota
```

Šifrování

Dynamic Server poskytuje prostředky pro šifrování hesel, přenášených dat a dat v určených sloupcích.

Přehled šifrování a voleb šifrování

Šifrování je proces transformace dat do nesrozumitelného tvaru, aby se tak zabránilo neoprávněnému použití dat. Pokud chcete číst zašifrovaný soubor, musíte mít přístup

k tajnému klíči nebo heslu, které vám umožní tato data dešifrovat. Dešifrovaná data se nazývají *otevřený text*, šifrovaná data jsou označována jako *šifry*. *Šifra* představuje algoritmus šifrování a dešifrování.

Standard DES (Data Encryption Standard) je šifrovací algoritmus navržený k šifrování a dešifrování dat pomocí 8bajtových bloků a 64bitového klíče.

Standard DES3 (Triple DES) je variací standardu DES, ve kterém se pro 192bitový klíč používají tři 64bitové klíče. Standard DES3 nejdříve šifruje otevřený text pomocí prvních 64 bitů klíče. Potom je text šifry pomocí další části klíče dešifrován. Nakonec je výsledný text šifry znovu šifrován pomocí poslední části klíče.

Standard AES (Advanced Encryption Standard) je nový algoritmus používaný vládou Spojených států amerických, který má původní algoritmus nahradit.

Existují dva šifrovací režimy:

- *Blokový režim*, metoda šifrování, při níž jsou zprávy rozděleny do bloků a každý blok je šifrován jako jednotka. Jelikož je každý blok dlouhý nejméně 8 bajtů, umožňuje blokový režim používat v šifrovacím algoritmu 64bitovou aritmetiku.
- *Proudový režim*, metoda šifrování, při níž je šifrován každý bajt jednotlivě. Je obecně považován za slabou formu šifrování.

Standard *Blowfish* je bloková šifra operující s 64bitovými (8bajtovými) bloky dat. Používá klíč proměnné velikosti, ale běžně jsou 128bitové (16bajtové) klíče považovány za dostatečné pro silné šifrování. Standard Blowfish může být používán ve stejných režimech jako standard DES.

Volby serveru Dynamic Server, které můžete použít pro šifrování, jsou uvedeny v následující tabulce:

Volba šifrování	Kdy a proč se používá tato volba
Šifrování hesel	Použijte k šifrování hesel. Další informace naleznete v části “Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat” na stránce 5-11.
Šifrování přenosu dat	Použijte k šifrování dat přenášených sítí. Jsou zahrnuta data přenášená mezi databázovým serverem a klientskými systémy. Další informace naleznete v části “Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat” na stránce 5-11.

Volba šifrování	Kdy a proč se používá tato volba
Šifrování na úrovni sloupce	<p>Použijte k nastavení šifrování hesel pro sloupce obsahující důvěrná data, jako například čísla platebních karet. Pokud nastavíte hesla pro šifrování na úrovni sloupců, budou data v těchto sloupcích uložena v šifrovaném formátu. Pouze uživatelé, kteří budou moci poskytnout tajné heslo, budou moci prohlížet, kopírovat a měnit šifrovaná data.</p> <p>Další informace naleznete v části “Použití šifrování na úrovni sloupců” na stránce 5-23.</p>

Používání modulů pro podporu komunikace (CSM) pro šifrování hesel a přenášených dat

Pomocí modulů CMS můžete umožnit dva různé druhy šifrování:

- Modul Simple Password CSM (SPWDCSM) poskytuje šifrování hesel.
Toto šifrování chrání heslo, když musí být za účelem ověření odesláno mezi klientem a databázovým serverem. Modul SPWDCSM je dostupný na všech platformách.
- Modul Encryption CSM (ENCCSM) umožňuje v rámci sítě šifrovat přenosy dat, včetně distribuovaných dotazů.
Tato volba poskytuje úplné šifrování dat pomocí knihovny openssl s řadou konfigurovatelných voleb. Jako součást šifrovaných dat je odesílán kód ověření zprávy (MAC), aby se zajistila integrita dat. MAC je šifrovaný výběr zprávy. Šifrovací algoritmus používá kód knihovny openssl 0.9.6.

Důležité: Moduly CSM mají následující omezení:

- Nelze používat oba druhy modulů CSM souběžně. Pokud například používáte modul SPWDCSM a rozhodnete se šifrovat síťová data, musíte ze souborů **concsm.cfg** a **sqlhosts** odstranit záznamy pro modul SPWDCSM.
- Při použití multiplexovaného připojení nelze používat ani modul SPWDCSM, ani modul ENCCSM.
- Replikace Enterprise Replication a replikace HDR (High-Availability Data Replication) podporují šifrování, ale nemohou používat připojení konfigurované pomocí modulu CSM.
Pokud chcete používat síťové šifrování s replikací Enterprise Replication, nastavte konfigurační parametry šifrování (další informace naleznete v příručce *IBM Informix: Dynamic Server Enterprise Replication Guide*).
- Šifrovaná připojení a nešifrovaná připojení nelze kombinovat na stejném portu.

Pokud chcete používat šifrování, musíte provést následující činnosti:

1. Do konfiguračního souboru **concsm.cfg** přidejte řádek.

Soubor **concsm.cfg** musí obsahovat záznam pro každý modul CSM (téhož druhu), který používáte. Více informací o souboru **concsm.cfg** naleznete v části “Konfigurační soubor CSM”.

2. Do sloupce **options** souboru nebo registru **sqlhosts** přidejte záznam.

Více informací o určení modulu CSM v souboru nebo registru **sqlhosts** naleznete v části “Volba modulu pro podporu komunikace” na stránce 3-26.

Konfigurační soubor CSM

Pokud používáte modul pro podporu komunikace (CSM), musíte vytvořit soubor **concsm.cfg**. Záznam v souboru je jediný řádek a je omezený 1024 znaky. Po předepsání modulu CSM do souboru **concsm.cfg** jej můžete povolit v parametru voleb v souboru **sqlhosts**, jak bylo popsáno v části “Volba modulu pro podporu komunikace” na stránce 3-26.

Soubor **concsm.cfg** je ve výchozím nastavení uložen v adresáři **\$INFORMIXDIR/etc**. Výchozí adresář pro soubor **concsm.cfg** je **INFORMIXDIR/etc**. Pokud chcete uložit soubor někde jinde, musíte přepsat výchozí umístění prostřednictvím nastavení proměnné prostředí **INFORMIXCONCSMCFG** na úplný název cesty nového umístění. Další informace o nastavení proměnné prostředí **INFORMIXCONCSMCFG** naleznete v příručce *IBM Informix: Guide to SQL Reference*.

Záznamy v souboru **concsm.cfg** musí splňovat následující omezení:

- Následující znaky nesmí být částí názvů cest ke knihovně:
 - = (rovnítko)
 - " (dvojitě uvozovky)
 - , (čárka)
- Nelze používat prázdné znaky, pokud tyto prázdné znaky nejsou částí názvu cesty.

Konfigurace šifrování hesel

Pro šifrování hesel musíte určit knihovny šifrování hesel a volby připojení. Při konfiguraci šifrování hesel přidejte pomocí následující syntaxe řádek k souboru **concsm.cfg**.

```
▶▶název_csm—(—"client"—knihovna_klienta—,—server—knihovna_serveru—"—,—  
knihovna_csm—  
▶—"globální_volby"—,——"volby_připojení"—)—▶▶
```

Volba	Popis						
client=clientlib	<p>Určuje úplný název a cestu ke sdílené knihovně, která je modulem CSM v klientském počítači.</p> <p>Klientské počítače používají tento modul CSM ke komunikaci s databázovým serverem. Knihovna poskytovaná serverem Dynamic Server je knihovna \$INFORMIXDIR/lib/client/csm/libixspw.so.</p>						
<i>volby_připojení</i>	<p>Pole <i>volby_připojení</i> nabývá těchto hodnot:</p> <table border="0"> <thead> <tr> <th>Nastavení</th> <th>Výsledek</th> </tr> </thead> <tbody> <tr> <td>p=1</td> <td>Heslo je pro ověřování povinné.</td> </tr> <tr> <td>p=0</td> <td>Heslo není povinné. Pokud jej klient poskytne, bude heslo šifrováno a použije se pro ověřování.</td> </tr> </tbody> </table> <p>Neznámá volba umístěná do pole <i>volby_připojení</i> bude mít za následek chybu inicializace kontextu.</p> <p>Do pole <i>volby_připojení</i> můžete zadat prázdnou hodnotu, například "". V případě klienta Client SDK předcházejícího verzi 2.3, pokud pole <i>conn_options</i> bude obsahovat hodnotu null, bude výchozí chování p=1. V případě klienta Client SDK verze 2.3 a novější, pokud pole <i>volby_připojení</i> bude obsahovat prázdnou hodnotu, bude výchozí chování p=0.</p>	Nastavení	Výsledek	p=1	Heslo je pro ověřování povinné.	p=0	Heslo není povinné. Pokud jej klient poskytne, bude heslo šifrováno a použije se pro ověřování.
Nastavení	Výsledek						
p=1	Heslo je pro ověřování povinné.						
p=0	Heslo není povinné. Pokud jej klient poskytne, bude heslo šifrováno a použije se pro ověřování.						
<i>knihovna_csm</i>	<p>Úplná cesta a název sdílené knihovny, která je modulem CSM, pokud je modul CSM sdílen jak databázovým serverem, tak klientskými počítači.</p> <p>Knihovna poskytovaná serverem Dynamic Server je knihovna \$INFORMIXDIR/lib/csm/libixspw.so.</p>						
<i>název_csm</i>	<p>Název, který přiřadíte modulu pro podporu komunikace (CSM).</p> <p>Název by mohl být například SPWDCSM.</p>						
<i>global_options</i>	Tato volba (globální volby) se v současnosti nepoužívá.						
server=knihovna_serveru	<p>Určuje úplnou cestu a název ke sdílené knihovně, která je modulem CSM v databázovém serveru.</p> <p>Knihovna poskytovaná serverem Dynamic Server je obvykle instalovaná v následujících adresářích:</p> <ul style="list-style-type: none"> • UNIX: \$INFORMIXDIR/lib/csm/libixspw.so • Windows: %INFORMIXDIR%/bin/libixspw.so 						

Tabulky SMI a nastavení souboru conescm.cfg: Pokud chcete vytvořit tabulky SMI při spuštění databázového serveru (**oninit -i**), nezadávejte do záznamu CSM

databázového serveru v souboru **concsm.cfg** volbu **p=1**. Proces **oninit** nemá heslo pro ID uživatele **informix** nebo **root**. Pokud do souboru **concsm.cfg** zadáte pro databázový server volbu **p=1**, obdržíte následující chybovou zprávu:

```
-5013 CSM: cannot obtain credential:  
authentication error.
```

Jak určit, že heslo je pro modul CSM databázového serveru povinné, když tabulky SMI nejsou dosud vytvořeny:

1. Nezadávejte volbu **p=1** do záznamu souboru **concsm.cfg**.
2. Spusťte databázový server pomocí příkazu **oninit -i** a vytvořte tabulky typu SMI.
3. Vypněte databázový server.
4. Zadejte volbu **p=1** do záznamu modulu CSM databázového serveru v souboru **concsm.cfg**.
5. Příkazem **oninit** spusťte znovu databázový server.

Příklad záznamů souboru concsm.cfg pro šifrování hesel: Následující dva příklady ilustrují dvě alternativy pro parametry, které musíte zadat do souboru **concsm.cfg** při definování modulu SPWDCSM:

```
SPWDCSM  
("client=/usr/informix/lib/client/csm/libixspw.so,  
server=/usr/informix/lib/csm/  
libixspw.so", "", "")
```

```
SPWDCSM("/usr/informix/lib/csm/  
libixspw.so", "", "")
```

Následující příklad ukazuje pole *volby_připojení* nastavené na hodnotu 0, takže není nutné žádné heslo:

```
SPWDCSM("/work/informix/csm/libixspw.so", "", "p=0")
```

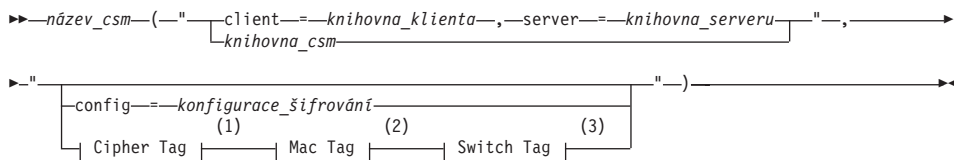
Konfigurace šifrování síťových dat

V případě šifrování síťových dat musíte určit síťové knihovny šifrování a volby šifrování. Můžete určit následující typy voleb šifrování:

- Šifry a režimy, které se mají používat při šifrování.
- Soubory klíčů MAC (Message authentication code).
- Úrovně MAC.
- Frekvence přepínání pro šifry a klíče.

Každá z těchto voleb je popsána dále v této části.

Volby šifrování můžete určit buď pomocí parametrů šifrování v samostatném souboru, nebo pomocí příznaků šifrování v souboru **concsm.cfg**. Pokud chcete konfigurovat síťové šifrování, použijte následující syntaxi k přidání jednoho nebo více řádků do souboru **concsm.cfg**.



Poznámky:

- 1 Další informace naleznete v části “Příznak šifry” na stránce 5-20.
- 2 Další informace naleznete v části “Příznak MAC” na stránce 5-21.
- 3 Další informace naleznete v části “Příznak přepínání” na stránce 5-22.

Volba	Popis
client = <i>knihovna_klienta</i>	Úplný název a cesta ke sdílené knihovně, která je modulem CSM v klientském počítači. Klientské počítače používají tento modul CSM ke komunikaci s databázovým serverem. Knihovna poskytovaná serverem Dynamic Server je knihovna \$INFORMIXDIR/lib/client/csm/iencs09a.so .
config = <i>konfigurace_šifrování</i>	Úplná cesta a název souboru, ve kterém jsou definovány parametry šifrování. Pokud tento soubor neexistuje, použijí se výchozí hodnoty. Není vrácena žádná chyba. Další informace o používání parametrů šifrování naleznete v části “Použití parametrů šifrování” na stránce 5-18.
<i>knihovna_csm</i>	Úplná cesta a název sdílené knihovny, která je modulem CSM, pokud je modul CSM sdílen jak databázovým serverem, tak klientskými počítači. Knihovna poskytovaná serverem Dynamic Server je knihovna \$INFORMIXDIR/lib/csm/iencs09a.so .
<i>název_csm</i>	Název, který přiřadíte modulu pro podporu komunikace. Název by mohl být například SPWDSCSM.
server = <i>knihovna_serveru</i>	Úplný název a cesta ke sdílené knihovně, která je modulem CSM v databázovém serveru. Knihovna poskytovaná serverem Dynamic Server je obvykle instalovaná v následujících adresářích: <ul style="list-style-type: none"> • UNIX: \$INFORMIXDIR/lib/csm/libixspw.so • Windows: %INFORMIXDIR%/bin/iencs09a.so

Šifry a režimy: Musíte určit, které šifry a režimy se budou používat během šifrování. Používaná šifra (a režim) je náhodně vybrána mezi šiframi, které jsou společné pro dané dva servery. Ujistěte se, že všechny servery a klientské počítače, které se účastní

šifrované komunikace, mají společné šifry a režimy. Šifrování je bezpečnější, pokud zahrnete více šifer a režimů, mezi nimiž může databázový server přepínat. Další informace o přepínání mezi šiframi naleznete v části “Frekvence přepínání” na stránce 5-18.

Důležité: Důrazně doporučujeme, abyste nezadávali konkrétní šifry. Z bezpečnostních důvodů by měly být povoleny všechny šifry. Pokud by u některé šifry byla zjištěna slabina, můžete ji vyloučit.

Pomocí volby **allbut** zadejte seznam všech šifer a režimů, které chcete vyloučit. Seznam **allbut** uzavřete do lomených závorek (<>). Seznam může zahrnovat jedinečné, zkrácené záznamy. Například **bf** může představovat **bf1**, **bf2** a **bf3**. Pokud je ovšem zkratka názvem skutečné šifry, bude vyloučena *pouze* tato šifra. Z tohoto důvodu hodnota **des** vyloučí pouze šifru DES, ale hodnota **de** vyloučí šifry **des**, **ede** a **desx**.

Jsou podporovány následující šifry:

Šifra	Popis	Šifra Blowfish	Popis
des	DES (64bitový klíč)	bf1	Blowfish (64bitový klíč)
ede	Triple DES	bf2	Blowfish (128bitový klíč)
desx	Extended DES (128bitový klíč)	bf3	Blowfish (192bitový klíč)

Upozornění: Šifru **desx** je možné používat pouze v režimu **cbc**.

Jsou podporovány následující režimy:

Režim	Popis
ecb	Electronic Code Book
cbc	Cipher Block Chaining
ocb	Cipher Feedback
ofb	Output Feedback

Jelikož je režim **ecb** považován za slabý, je zahrnut pouze na výslovný požadavek. Není zahrnut do seznamu všech režimů - **all** ani do seznamu **allbut**.

Soubory klíčů MAC: Soubory klíčů MAC obsahují šifrovací klíče, které se používají k šifrování zpráv. Všechny databázové servery a klientské počítače, které se účastní šifrování, by měly mít společné soubory klíčů MAC. Další informace o přepínání mezi klíči MAC naleznete v části “Frekvence přepínání” na stránce 5-18.

Výchozí soubor klíčů MAC je vestavěný soubor poskytovaný serverem Dynamic Server. Tento soubor poskytuje omezené ověřování zpráv (určitou kontrolu platnosti obdržené zprávy a určení, zda se jeví, jako by přišla z klienta nebo serveru IBM Informix Dynamic Server). Na místě generovaný soubor klíčů MAC poskytuje přísnější ověření. Soubory klíčů můžete vytvářet pomocí obslužného programu **GenMacKey**.

Každý ze souborů klíčů MAC dostává prioritu a je domluven v čase připojení. Přiřazení priority souborům klíčů MAC je založeno na jejich času vytvoření pomocí obslužného programu **GenMacKey**. Vestavěný soubor klíčů má nejnižší prioritu.

Tip: Pokud neexistují žádné soubory klíčů MAC, použije se jako výchozí vestavěný klíč MAC. Použitím souboru klíčů MAC je ovšem vestavěný klíč MAC zablokován.

Postup vytvoření nového souboru klíčů MAC:

1. Z příkazového řádku spusíte následující příkaz:
GenMacKey -o filename
kde *filename* je cesta a název souboru nového souboru klíčů MAC.
2. Jedním z následujících způsobů aktualizujte konfiguraci centrálního serveru tak, aby zahrnovala umístění nového souboru klíčů MAC:
 - **Pomocí příznaků šifrování:** Upravte odpovídající řádek souboru **conscsm.cfg** tak, že do příznaku **mac** přidáte cestu a název souboru. Pokyny naleznete v části “Příznak MAC” na stránce 5-21.
 - **Pomocí parametrů šifrování:** Upravte soubor parametrů šifrování tak, že pozměníte hodnotu parametru ENCCSM_MACFILES. Pokyny naleznete v části “ENCCSM_MACFILES” na stránce 5-19.
3. Je-li to nutné, odstraňte z konfigurace záznamy o starém souboru klíčů MAC.
4. Distribuuje nový soubor klíčů MAC mezi všechny vhodné počítače.

Úrovně MAC: Úrovně MAC určují typ generování klíčů MAC.

Podporované úrovně generování jsou následující:

- **high** (vysoká). Používá generování SHA1 MAC u všech zpráv.
- **medium** (střední). Používá generování SHA1 MAC u všech zpráv větších než 20 bajtů a u menších zpráv používá sdružování XOR.
- **low** (nízká). Používá sdružování XOR u všech zpráv.
- **off** (vypnutá). Nepoužívá generování MAC.

Úroveň dostává prioritu nejvyšší hodnoty. Pokud například jeden databázový server má povolenou úroveň **high** a **medium** a jiný databázový server má povolenou pouze úroveň **low**, potom pokus o připojení selže. Záznam **off** by se měl používat pouze mezi servery, u kterých je zaručeno, že se jedná o zabezpečené síťové připojení.

Ujistěte se, že všechny servery a klientské počítače, které se účastnily šifrované komunikace, mají společně úroveň MAC.

Frekvence přepínání: Frekvence přepínání definuje, kdy jsou domlouvány šifry a tajné klíče. Čím déle zůstává tajný klíč a šifrovací šifra v používání, tím je pravděpodobnější, že budou šifrovací pravidla narušená útočníkem. Abyste se tomu vyhnuli, doporučují kryptologové v případě dlouhodobých připojení pravidelně měnit tajný klíč a šifru. Výchozí frekvence tohoto domlouvání je jedenkrát za hodinu. Pomocí voleb přepínání můžete nastavit čas v minutách, po kterém bude docházet k domlouvání.

Použití parametrů šifrování: Můžete nakonfigurovat volby šifrování tak, že v souboru nastavíte parametry šifrování. V souboru parametrů šifrování má každá volba následující formát:

název_parametru hodnota

K nastavení voleb šifrování použijte následující parametry:

- ENCCSM_CIPHERS: Šifry, které se mají používat.
- ENCCSM_MAC: Úroveň MAC.
- ENCCSM_MACFILES: Umístění souboru MAC.
- ENCCSM_SWITCH: Frekvence změny šifry a klíče.

Každý z těchto parametrů je popsán dále v této části.

Na hodnoty parametrů se vztahují následující omezení:

- Každý záznam by měl být ve formátu *název_parametru hodnota*, oddělený bílými mezerami.
- V rámci hodnoty nejsou povoleny prázdné znaky.
- Každý parametry by měl mít v konfiguračním souboru jeden záznam. Pokud existuje několik záznamů, je brán v potaz pouze první záznam.
- Pokud v konfiguračním souboru parametr neexistuje, použije se výchozí hodnota.
- Znak následující za znakem komentáře (#) jsou ignorovány, hodnota názvu cesty ovšem ignorována není.

ENCCSM_CIPHERS:

Syntaxe parametru ENCCSM_CIPHERS

```
all|allbut:<seznam šifer a režimů>|šifra:režim{,šifra:režim ...}
```

- all

Určuje zahrnutí všech dostupných šifer a režimů, kromě režimu ECB. Například: ENCCSM_CIPHERS all

- allbut:<seznam šifer a režimů>

Určuje zahrnutí všech šifer a režimů kromě těch, které jsou uvedeny v seznamu. Šifry nebo režimy odděluje čárkou. Například: ENCCSM_CIPHERS allbut:<cbc,bf>

- **šifra:režim**

Určuje šifry a režimy. Páry šifra-režim odděluje čárkou. Například: ENCCSM_CIPHERS des3:cbc,des3:ofb

Výchozí hodnota allbut:<ecb>

Parametr ENCCSM_CIPHERS určuje šifry a režimy, které se mají používat během šifrování. Další informace o šifrách a režimech naleznete v části “Šifry a režimy” na stránce 5-15.

ENCCSM_MAC:

Výchozí hodnota

medium

Rozsah hodnot Jedna nebo více následujících voleb, oddělených čárkami:

- **off** - Generování MAC se nepoužívá.
- **low** - Používá sdružování XOR u všech zpráv.
- **medium** - Používá generování SHA1 MAC u všech zpráv větších než 20 bajtů a u menších zpráv používá sdružování XOR.
- **high** - U všech zpráv používá generování SHA1 MAC.

Například: ENCCSM_MAC medium,high

Parametr ENCCSM_MAC určuje úroveň MAC, která se má používat. Další informace o úrovních MAC naleznete v části “Úrovně MAC” na stránce 5-17.

ENCCSM_MACFILES:

Výchozí hodnota

builtin

Jednotky Názvy cest až 1536 bajtů dlouhé

Rozsah hodnot Jeden nebo více úplných názvů cesty a souboru oddělených čárkami a volitelné klíčové slovo **builtin**. Například: ENCCSM_MACFILES /usr/local/bin/mac1.dat,/usr/local/bin/mac2.dat,builtin

Parametr ENCCSM_MACFILES určuje soubory klíčů MAC, které se mají používat. Další informace naleznete v části “Soubory klíčů MAC” na stránce 5-16.

ENCCSM_SWITCH:

Syntaxe parametru ENCCSM_SWITCH
cipher_switch_time,key_switch_time

- *cipher_switch_time* - Určuje počet minut mezi domlouváním šifer.
- *key_switch_time* - Určuje počet minut mezi domlouváním tajného klíče.

Výchozí hodnota

60,60

Jednotky

Minuty

Rozsah hodnot

Kladná celá čísla

Parametr ENCCSM_SWITCH definuje počet minut mezi domlouváním šifry a klíče. Další informace naleznete v části “Frekvence přepínání” na stránce 5-18.

Příklad souboru parametrů šifrování: Následující příklad znázorňuje soubor parametrů šifrování:

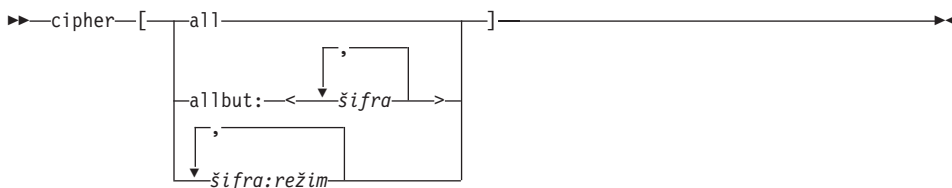
```
ENCCSM_CIPHERS      all
ENCCSM_SWITCH      120,60
ENCCSM_MAC         medium
ENCCSM_MACFILE     /usr/informix/etc/MacKey.dat
```

Následující příklad ilustruje řádek v souboru **concsn.cfg**, který určuje šifrování pomocí souboru parametrů:

```
ENCCSM("usr/informix/lib/cms/iencs09a.so",
"config=/usr/lib/encrypt.txt")
```

Použití příznaků šifrování: Příznaky šifrování můžete použít k určení voleb šifrování v souboru **concsn.cfg**.

Příznak šifry: Příznak **šifry** může zahrnovat volby šifry zobrazené v následujícím diagramu syntaxe.



Volba šifry	Popis
all	Určuje zahrnutí všech dostupných šifer a dostupných režimů. Zde není seznam hodnot. Například: cipher[all],...

Volba šifry	Popis
allbut: <seznam šifer k vyloučení>	<p>Určuje vyloučit všechny šifry kromě těch, které nejsou uvedeny v seznamu.</p> <p>Další informace naleznete v části “Šifry a režimy” na stránce 5-15.</p> <p>Například: cipher[allbut:<ecb,des>],... cipher[allbut:<cbc,bf>]</p>
šifra:režim	<p>Určuje jednu nebo více šifer a režimů. Například: cipher[des:cbc,des:ofb]</p>

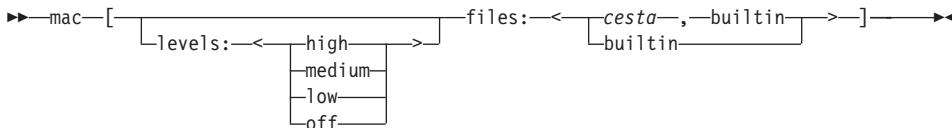
Výchozí hodnota pro pole cipher (šifra) je:

cipher[allbut:<ecb>]

Další informace o šifrách a režimech naleznete v části “Šifry a režimy” na stránce 5-15.

Příznak MAC: Příznak **mac** definuje soubory klíčů MAC a úroveň generování MAC, která se má použít během generování MAC.

Příznak **mac** může zahrnovat volby MAC uvedené v následujícím diagramu syntaxe.

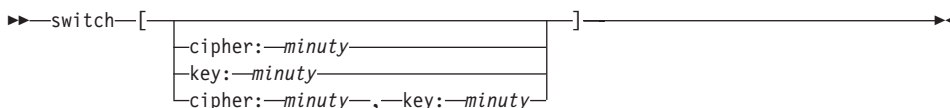


Volby Mac	Popis
levels	<p>Určuje seznam úrovní generování MAC oddělených čárkou, které jsou podporovány připojením.</p> <p>Další informace naleznete v části “Úrovně MAC” na stránce 5-17.</p>

Volby Mac	Popis
files	<p>Určuje seznam úplných názvů cest k souborům klíčů MAC, které jsou odděleny čárkou.</p> <p>Další informace naleznete v části “Soubory klíčů MAC” na stránce 5-16.</p> <p>Například: <code>mac[levels:<high,low>,files: </usr/local/bin/mac1.dat, /usr/local/bin/mac2.dat,builtin]</code></p>

Príznak přepínání: Příznak **přepínání** definuje frekvenci domlouvání šifer a tajných klíčů.

Příznak **přepínání** může zahrnovat volby přepínání uvedené v následujícím diagramu syntaxe.



Volby přepínání	Popis
cipher: -minuty	Určuje čas v minutách mezi domlouváním šifry.
key: -minuty	<p>Určuje čas v minutách mezi domlouváním tajného klíče.</p> <p>Například: <code>switch[cipher:120,key:20]</code></p>

Další informace o přepínání šifer a režimů naleznete v části “Frekvence přepínání” na stránce 5-18.

Příklady používání příznaků šifrování: Následující dva příklady znázorňují alternativy pro příznaky zadávané do souboru **concsm.cfg** za účelem definice šifrování CSM.
 Například:

```
ENCCSM("$INFORMIXDIR/lib/csm/iencs09a.so",
"cipher[allbut:<ecb,bf>"]")
```

Tento konfigurační řetězec určuje, že se mají použít všechny dostupné šifry kromě cifer typu Blowfish a žádná šifra se nemá použít v režimu ECB.

Další příklad:

```
ENCCSM("/$INFORMIXDIR/lib/csm/iencs09a.so",  
"cipher[des:cbc,ede:ofb,desx:cbc],switch[cipher:120,key:15] ")
```

Tento konfigurační řetězec určuje, že se mají pro toto připojení použít kombinace šifry a režimu DES/CBC, EDE/OFB a DESX/CBC a že má být používána šifra přepínána každých 120 minut a každých 15 minut má být domlouván tajný klíč.

Použití šifrování na úrovni sloupců

Pomocí příkazu SET ENCRYPTION PASSWORD a funkce ENCRYPT můžete určit šifrovací heslo a můžete zadat šifrování dat ve sloupcích obsahujících následující datové typy nebo datové typy inteligentních velkých objektů:

- CHAR,
- NCHAR,
- VARCHAR,
- NVARCHAR,
- LVARCHAR,
- BLOB,
- CLOB.

Příkaz SET ENCRYPTION PASSWORD udává *heslo* za účelem utajení dat prostřednictvím vestavěných funkcí, které používají pro šifrování a dešifrování algoritmy DES3 nebo AES. Tyto funkce umožňují databázi ukládat důvěrná data v zašifrovaném formátu.

Možná budete chtít nastavit šifrovací hesla pro sloupce obsahující důvěrná data, jako například čísla kreditních karet. Pokud nastavíte hesla pro šifrování na úrovni sloupců, budou data v těchto sloupcích uložena v šifrovaném formátu. Pouze uživatelé, kteří budou moci poskytnout tajné heslo, budou moci prohlížet, kopírovat a měnit šifrovaná data.

Pokud nastavíte šifrovací heslo, můžete také určit náznak hesla. Jestliže určíte náznak hesla, můžete tento náznak uložit spolu se zašifrovaným heslem, nebo jej uložit do jiného umístění. Heslo musí být dlouhé minimálně 6 bajtů a maximálně 128 bajtů. Heslo použité pro dešifrování se musí shodovat s heslem použitým pro šifrování.

Když nastavíte šifrovací hesla pro sloupcová data, musíte zadat následující typy šifrování:

- **Šifrování na úrovni sloupce.** Všechny hodnoty v určitém sloupci v databázové tabulce jsou šifrovány pomocí stejného hesla (slovo nebo fráze), stejného šifrovacího algoritmu a stejného šifrovacího režimu. U šifrování na úrovni sloupce můžete náznak uložit mimo zašifrovaný sloupec místo toho, abyste jej opakovali v každém řádku.

Tip: Pokud se nepoužívají funkce šifrování, mohli by koncoví uživatelé zadávat nešifrovaná data do sloupců, která jsou určena pro šifrovaná data. Abyste zajistili, že data zadávaná do určitého pole jsou vždy šifrovaná, použijte pohledy a spouštěče typu "instead-of".

- **Šifrování na úrovni buňky** (rovněž nazývané *šifrování na úrovni sloupce v řádce* ašifrování na úrovni sloupce sady). V rámci sloupce šifrovaných dat je použito mnoho různých hesel, šifrovacích algoritmů nebo režimů. Tento typ šifrování může být nezbytná při ochraně osobních údajů.

Hesla a názny, které stanovíte pomocí příkazu SET ENCRYPTION PASSWORD, nejsou uloženy jako otevřený text v žádné tabulce systémového katalogu. Abyste zabránili ostatním uživatelům v přístupu k otevřenému textu šifrovaných dat nebo hesla, musíte se vyhnout akcím, které by mohly kompromitovat utajení hesla:

- Pokud není databáze přístupná pouze prostřednictvím zabezpečené sítě, musíte povolit modul ENCCSM (Encryption Communication Support Module), který bude chránit přenosy dat mezi databázovým serverem a jakýmkoli klientským systémem.
- Nevytvářejte funkční index pomocí funkce DECRYPT_CHAR(). To by způsobilo, že se do databáze uloží data otevřeného textu, což by mařilo účel šifrování.
- Neukládejte hesla do spouštěče nebo do uživatelské rutiny (UDR), které by vystavily heslo veřejnosti. Před aktivací spouštěče, vyvoláním uživatelské rutiny nebo předáním jakéhokoli hesla jako parametru uživatelské rutiny použijte heslo relace.

Při nastavování hesla server Dynamic Server přeneše heslo a jakýkoli náznak do 128bitového klíče, který se používá k šifrování hesla a názny. Hesla a názny nejsou ukládány jako otevřený text. Klíč je na čase založená náhodná hodnota generovaná pro konkrétní instanci. Databázový server inicializuje klíč při spuštění serveru, klíč je zničen, když se databázový server vypne.

Server Dynamic Server zahrnuje šifrovací virtuální procesor Encrypt Virtual Processor. Pokud není v konfiguračním souboru ONCONFIG definovaná volba `encrypt` parametru VPCLASS, spustí databázový server při prvním vyvolání funkce šifrování nebo dešifrování na úrovni sloupce jeden šifrovací virtuální procesor Encrypt VP. Můžete definovat několik šifrovacích virtuálních procesorů Encrypt VP, pokud je nezbytné snížit čas potřebný ke spuštění databázového serveru. Další informace naleznete v části "Šifrovací virtuální procesory" na stránce 6-33 a v kapitole věnované konfiguračním parametrům v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Když je databázový server v režimu online, můžete použít příkaz **onmode -p** k přidání a vypuštění šifrovacích virtuálních procesorů Encrypt VP. Pokud chcete například přidat čtyři šifrovací virtuální procesory Encrypt VP, zadejte:

```
onmode -p 4 encrypt
```

Chcete-li vypustit tři šifrovací virtuální procesory Encrypt VP, zadejte:

```
onmode -p -3 encrypt
```

Další informace naleznete v kapitole věnované obslužnému programu **onmode** v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Pokyny týkající se paměti

Šifrovaná hodnota používá více paměťového prostoru než odpovídající hodnota otevřeného textu. Je to způsobeno tím, že s hodnotou jsou uloženy všechny informace nutné k dešifrování hodnoty, kromě šifrovacího klíče. Z tohoto důvodu se nedoporučuje vkládat nulové bajty do šifrovaného výsledku.

Šifrování dat sloupce

Dříve než nastavíte šifrovací heslo a data zašifrujete, musíte ověřit, že se data vejdou do sloupce.

Postup šifrování sloupce:

1. Vypočtete velikost šifrovaného sloupce. Je-li to nutné, sloupec změňte.
Dva příklady metod pro výpočet velikosti šifrovaného sloupce naleznete v části “Příklad ilustrující jak určit velikost šifrovaného sloupce”.
2. Do svého kódu vložte informaci o šifrovacím hesle. K určení hesla nebo hesla a náznaku použijte příkaz SET ENCRYPTION PASSWORD SQL. Pomocí funkce ENCRYPT_AES() nebo ENCRYPT_TDES() definujte šifrovaná data.
Příklad vkládání hesla do kódu a použití funkce ENCRYPT naleznete v části “Příklad ilustrující způsob šifrování sloupce” na stránce 5-26.

Pomocí funkcí DECRYPT_BINARY() a DECRYPT_CHAR() se dotazujte na šifrovaná data. Příklad dotazování na šifrovaná data naleznete v části “Příklad ilustrující dotazování na šifrovaná data” na stránce 5-27.

V příručce *IBM Informix: Guide to SQL Syntax* naleznete další informace o následujících tématech:

- Příkaz SET ENCRYPTION PASSWORD a syntaxe používaná pro určení hesla a náznaku.
- Funkce ENCRYPT a DECRYPT.

Příklad ilustrující jak určit velikost šifrovaného sloupce

Následující příklad ilustruje, jak lze vypočítat velikost sloupce **Credit Card**:

```
DATA SIZE 16 bytes
  ENCRYPTED DATA SIZE = (DATA SIZE + blocksize8) / blocksize8 *
  blocksize8 = 24 bytes (integer operation)
  OR ENCRYPTED DATA SIZE = (DATA SIZE - DATA SIZE% blocksize8 +
  blocksize8 ) = 24 bytes
  (For ENCRYPT_TDES, round up to (N + 1) * 8 bytes, for example
  13 bytes round up to 16 bytes, 16 bytes to 24 bytes)
  HEADER SIZE = 11 bytes (for Base64 encoding)
  IV SIZE = 8 bytes (fixed size)
  HINT SIZE = 32 bytes (maximum size)
  ENCRYPED HINT SIZE = 40 bytes (maximum size)
```

BASE64 SIZE = ((INPUT DATA SIZE + 2) / 3) * 4
(integer operation)

OR BASE64 SIZE = ((INPUT DATA SIZE + 2) -
(INPUT DATA SIZE + 2) % 3) / 3 * 4

TOTAL SIZE = HEADER SIZE
+ BASE64(IV SIZE + ENCRYPTED DATA SIZE + ENCRYPTED HINT)
= 11 + BASE64(8 + 24 + 40)
= 11 + (72 + 2) / 3 * 4
= 11 + 96 = 107

V předchozím příkladu je vektor inicializace (IV) pseudonáhodná série bajtů, která se používá k vyvolání šifrování při použití některých režimů šifer. Velikost vektoru inicializace je počet náhodných sérií bajtů, pro server Dynamic Server je to 8 bajtů.

Náznak není uložen ve sloupci, celková velikost v předchozím příkladu je 55 bajtů.

Tip: Jiný způsob, jak určit velikost šifrovaného sloupce, představuje následující výpočet:

```
SELECT LENGTH(ENCRYPT_TDES  
("1234567890123456",  
"heslo", "dlouhý...náznak"))  
FROM "informix".systables WHERE tabid = 1
```

Bez náznaku bude výpočet probíhat následovně:

```
SELECT LENGTH(ENCRYPT_TDES("1234567890123456",  
"heslo", ""))  
FROM "informix".systables  
WHERE tabid = 1
```

Důležité: Pokud je velikost sloupce menší než velikost dat vrácená funkcemi ENCRYPT/DECRYPT, budou šifrovaná data oříznuta při vložení a nebude možné tato data dešifrovat (neboť záhlaví bude oznamovat, že by přijatá data měla mít větší délku).

Příklad ilustrující způsob šifrování sloupce

Následující příklad ilustruje použití šifrovacího hesla u sloupce, který obsahuje číslo sociálního pojištění:

```
create table emp  
( name char(40),  
 salary money,  
 ssn lvarchar(64)  
);  
  
set encryption password to "one two three 123";  
insert into emp (ssn) values ("Alice", 50000, encrypt_aes  
( '123-456-7890'));
```



```

insert into emp (ssn) values ("Bob", 65000, encrypt_aes
('213-656-0890'));
select name, salary, decrypt(ssn) from emp where
decrypt(ssn) = '123-456-7890';

```

Příklad ilustrující dotazování na šifrovaná data

Následující příklad ilustruje použití dešifrovací funkce za účelem dotazování na šifrovaná data:

```

select id, decrypt(creditcard, "číslo platební karty je
šifrované") from customer;
or
set encryption password to "číslo platební karty je šifrované";
select id from customer
where decrypt(creditcard) =
"2345678901234567"

```

Ověřovací moduly

V závislosti na platformě můžete použít jeden z následujících ověřovacích modulů:

- Modul PAM (Pluggable Authentication Module) pro systémy serveru Dynamic Server spouštěné v operačním systému UNIX nebo Linux. Tyto moduly umožňují implementovat různé ověřovací moduly pro různé aplikace. Další informace naleznete v níže uvedené části “Moduly PAM (Pluggable Authentication Modules) pro systémy spouštěné v operačním systému UNIX nebo Linux”.
- Modul LDAP (Lightweight Directory Access Protocol) Authentication Support pro operační systém Windows. Modul LDAP Authentication Support Module použijte, pokud budete chtít k ověřování uživatelů využívat server LDAP. Další informace naleznete v části “Podpora ověřování LDAP v operačním systému Windows” na stránce 5-30.

Moduly PAM (Pluggable Authentication Modules) pro systémy spouštěné v operačním systému UNIX nebo Linux

Modul PAM je dobře definovaný rámec pro podporu různých ověřovacích modulů původně vyvinutý společností Sun Microsystems.

Modul PAM umožňuje systémovým administrátorům implementovat různé ověřovací mechanismy pro různé aplikace. Například systém, jakým je přihlašovací program k operačnímu systému UNIX, bude mít jiné potřeby než aplikace, která přistupuje k důvěrným informacím v databázi. Modul PAM umožňuje mnoho takovýchto scénářů v jednom počítači, neboť ověřovací služby jsou připojeny k úrovni aplikace.

Kromě toho, že umožňuje aplikaci vybrat si potřebnou úroveň ověření, povoluje modul PAM také stohování modulů. Mnoho modulů je možné stohovat jeden za druhým a tím povolit aplikaci, aby byla před udělením přístupu ověřována několika způsoby. Modul PAM poskytuje sadu rozhraní API pro podporu ověřování, správu účtu, správu relace a správu hesla.

Systémový administrátor může použití modulů PAM povolit nebo zakázat. Ve výchozím nastavení používá databázový server tradiční mechanismus ověřování systému Informix (který je založen na mechanismu rhosts systému BSD), aby uživatelé nemuseli provádět velké změny.

Postup použití modul PAM se serverem Dynamic Server:

- Databázový server Informix musí být na platformě operačního systému, který podporuje modul PAM.
- Klientské aplikace musí být napsány pomocí dostatečně aktuální verze Client SDK.
- Musíte nakonfigurovat příslušnou službu modulu PAM v operačním systému.
- Musíte vědět, zda služba modulu PAM bude pouze přijímat dané heslo, nebo zda bude používat protokol výzva-odezva (challenge-response), jako například ověřovací server RADIUS.
- Pokud služba modulu PAM bude používat protokol výzva-odezva, musíte změnit své aplikace tak, aby byly schopné pracovat s výzvou a odezvou. Aplikace si musí být vědoma, že modul PAM může vznést několik výzev.
- Musíte zajistit, aby replikace Enterprise Replication a replikace HDR (High-Availability Data Replication) nebyly ovlivňovány ověřováním modulem PAM.
- Musíte změnit záznam serveru v souboru sqlhosts pro klientskou aplikaci i databázový server (pokud jsou v oddělených počítačích nebo v oddělených umístěních v jediném počítači).

Podrobnější informace o těchto změnách naleznete dále.

Podporované platformy

Modul PAM je podporován v operačním systému Solaris a Linux, v 32bitovém i 64bitovém režimu.

V operačním systému HP-UX a AIX je modul PAM podporován pouze v 32bitovém režimu.

Název služby modulu PAM

Název služby modulu PAM identifikuje modul PAM. Tento modul PAM bývá běžně uložen v adresáři **/usr/lib/security** a jeho parametry jsou uvedeny v souboru **/etc/pam.conf**.

V operačním systému Linux může být soubor **/etc/pam.conf** nahrazen adresářem nazvaným **/etc/pam.d**, který obsahuje soubor pro každou službu modulu PAM. Pokud adresář **/etc/pam.d** existuje, bude soubor **/etc/pam.conf** operačním systémem Linux ignorován. Podrobné informace o tomto konfiguračním souboru naleznete v dokumentaci systému.

Režim ověřování

Modul PAM určuje, zda je stačí jednoduché heslo, nebo zda budou požadovány další výzvy. Implementace modulu PAM v serveru Dynamic Server využívá skutečnosti, že při explicitním připojení odesílá heslo k serveru klient. Toto heslo může v případech, kdy se používá jednoduché heslo, splnit požadavky modulu PAM. Pokud režim ověřování zahrnuje odpovědi na výzvy, musí být aplikace na tyto odezvy připravena. Aplikace musí být připravena, že modul PAM může vznést několik výzev.

Požadovaná velikost zásobníku

Funkce PAM zavádí moduly PAM (sdílené knihovny) operačního systému nebo jiného dodavatele do uživatelského jednotkového procesu **informix**. Požadavky na velikost zásobníku těchto modulů PAM nelze předvídat. Například v operačním systému Linux potřebují některé moduly více než 128K prostoru pro zásobníky. Pomocí konfiguračního parametru PAM_STACKSIZE můžete přizpůsobit velikost zásobníku pro moduly PAM. Výchozí hodnota parametru PAM_STACKSIZE je 32 KB.

Nastavte hodnotu PAM_STACKSIZE v souboru ONCONFIG například následovně:

```
PAM_STACKSIZE 64 # Velikost zásobníku potřebná pro moduly PAM  
(K Bajty)
```

V operačním systému Linux je výchozí hodnota 128 KB plus hodnota konfiguračního parametru STACKSIZE.

Implicitní připojení

Systém PAM je systém orientovaný na výzvy, kde je jako odezva na zprávu modulu PAM poskytnuta ověřovací odezva (heslo). V případě implicitních připojení k databázovému serveru se heslo nepoužívá.

Z tohoto důvodu mohou implicitní připojení pracovat při použití modulu PAM pouze v režimu výzvy. Výsledkem implicitních připojení v režimu hesla bude selhání.

Konfigurace databázového serveru k používání modulu PAM

Aby bylo možné konfigurovat server k používání modulu PAM, musí systémový administrátor znát:

- Název modulu PAM.
- Zda modul PAM vznese výzvu kromě přijetí jednoduché kombinace jména uživatele a hesla.

Následující příklad znázorňuje záznam souboru sqlhosts s ilustrativními názvy:

```
Režim ověřování: challenge  
ifxserver2 otlitcp servermc portnum2 volby  
kde volby jsou "s=4, pam_serv=(pam_pass), pamauth=(challenge)"
```

Služba PAM: pam_password (Požaduje pouze heslo)

```
Režim ověřování: password  
ifxserver2 otlitcp servermc portnum2 volby  
kde volby jsou "s=4, pam_serv=(pam_pass), pamauth=(password)"
```

Replikace Enterprise Replication a replikace HDR (High-Availability Data Replication)

Replikace Enterprise Replication a replikace HDR (High-Availability Data Replication) nemohou odpovídat na výzvy, pokud je schéma ověřování vyžaduje. Z tohoto důvodu nesmí být replikace Enterprise Replication a replikace HDR konfigurovány na portu používajícím modul PAM. Více informací o prostředcích pro vyhrazení portů replikaci Enterprise Replication a replikaci HDR naleznete v části “Volby zabezpečení připojení replikace Enterprise Replication a replikace HDR” na stránce 5-37.

Vývoj aplikací

Proces přípravy aplikace k odpovědím na výzvy vznesené modulem PAM je stejný proces, který se používá při odpovědích na výzvy vznesené moduly LDAP Authentication Support Module. Další informace naleznete v části “Vývoj aplikací pro moduly ověřování” na stránce 5-33.

Distribuované transakce s modulem PAM, klientská rozhraní API a problémy kompatibility

Distribuované transakce, rozhraní API, která podporují modul PAM a problémy kompatibility s ostatními produkty a nástroji IBM Informix jsou stejné u modulů PAM a LDAP Authentication Support Module. Další informace naleznete v částech “Distribuované transakce” na stránce 5-35, “Klientská rozhraní API a moduly na podporu ověřování” na stránce 5-35 a “Problémy s kompatibilitou” na stránce 5-36.

Podpora ověřování LDAP v operačním systému Windows

Ověřování LDAP v operačním systému Windows se nastavuje a konfiguruje jako modul PAM (Pluggable Authentication Module), který je používán v operačním systému UNIX a Linux. Pokud chcete, aby uživatelé systému ověřoval server LDAP, použijte modul LDAP Authentication Support Module. Tento modul obsahuje zdrojový kód, který můžete pro vlastní konkrétní modul LDAP Authentication Support Module změnit.

Ověřovací modul je DLL, které je obvykle uloženo v adresáři **INFORMIXDIR%\dbssodir\lib\security**. Parametry modulu jsou uvedeny v souboru **%INFORMIXDIR%\dbssodir\pam.conf**. Zdrojový kód pro plně funkční modul LDAP Authentication Module a příklady požadovaných konfiguračních souborů jsou zahrnuty v adresáři **%INFORMIXDIR%\demo\authentication**.

Modul LDAP Authentication Module poskytuje pouze ověřování jediným modulem. Tento model nepodporuje funkce jako stohování modulů. Systémový administrátor může použití ověřování povolit nebo zakázat.

Instalace a přizpůsobení modulu LDAP Authentication Support Module

Než budete moci použít modul LDAP Authentication Module serveru Dynamic Server k vytvoření vlastního ověřovacího modulu, musíte mít server LDAP a systém LDAP na straně klienta. Příkladem systému LDAP je server IBM Directory Server a server openLDAP.

System LDAP na straně klienta obvykle zahrnuje knihovny LDAP a hlavičkové soubory. Tyto knihovny a hlavičkové soubory jsou požadovány pro kompilaci modulu LDAP.

Postup přizpůsobení modulu:

1. Přizpůsobte soubor **pam_ldap.c**, který je zahrnut do serveru Dynamic Server.
2. Zkompilujte soubor **pam_ldap.c** do knihovny DLL a umístěte ji do zabezpečeného adresáře. Doporučuje se, abyste knihovnu DLL umístili do adresáře **%INFORMIXDIR%\dbssodir\lib**. Můžete ovšem knihovnu DLL umístit do jakéhokoli adresáře, který vyhovuje potřebám na zabezpečení.

Vaše instalace také zahrnuje šablonu konfiguračního souboru **pam_ldap_tmpl** pro modul LDAP. Tento konfigurační soubor obsahuje informace specifické pro pracovní místo (site specific information). Doporučuje se, abyste uložili informace specifické pro pracovní místo do tohoto konfiguračního souboru, protože tento soubor umožňuje jedinému modulu LDAP pracovat v různých nastaveních.

Konfigurace modulu LDAP

Ke konfiguraci modulu LDAP použijte šablonu konfiguračního souboru PAM.

Postup konfigurace modulu LDAP:

1. Zkopírujte soubor šablony do adresáře **%INFORMIXDIR%\dbssodir\etc** a pojmenujte jej **pam.conf**.
2. Přizpůsobte soubor tak, aby vyhovoval místnímu nastavení zabezpečení ochrany dat. Podrobnosti týkající se přizpůsobení souboru naleznete v souboru šablony **pam.conf_tmpl**.

Konfigurace serveru Dynamic Server

Pokud chcete nakonfigurovat server pro použití modulu LDAP Authentication Support Module, upravte soubor **sqlhosts**. Administrátor systému musí znát:

- Název modulu.
- Zda modul vynesou výzvu kromě přijetí jednoduché kombinace jména uživatele a hesla.

Následující příklad znázorňuje záznam souboru **sqlhosts** s popisnými názvy:

Služba PAM: pam_chal

```
Režim ověřování: challenge
ifxserver1 otlitcp servermc portnum1
s=4, pam_serv=(pam_chal), pamauth=(challenge)
```

Služba PAM: pam_password (Požaduje pouze heslo)

```
Režim ověřování: password
ifxserver2 otlitcp servermc portnum2
s=4, pam_serv=(pam_pass), pamauth=(password)
```

Režim ověřování

Modul LDAP Authentication Support Module určuje, zda dostačuje jednoduché heslo, nebo zda jsou požadovány další výzvy. Implementace modulu v serveru Dynamic Server využívá skutečnosti, že při explicitním připojení odesílá heslo k serveru klient. Toto heslo může v případech, kdy se používá jednoduché heslo, být použito ke splnění požadavků modulu LDAP Authentication Support Module. Pokud režim ověřování zahrnuje odpovědi na jednu nebo několik výzev, musí být aplikace schopna na tyto odezvy odpovědět.

Implicitní připojení

Modul LDAP Authentication Support Module je systém orientovaný na výzvy, kde je jako odezva na zprávu modulu LDAP Authentication Support Module poskytnuta ověřovací odezva (heslo). V případě implicitních připojení k databázovému serveru se heslo nepoužívá.

Implicitní připojení mohou pod modulem LDAP Authentication Support Module pracovat pouze v režimu výzvy. Pokus o implicitní připojení v režimu hesla způsobí selhání.

Replikace Enterprise Replication a replikace HDR (High-Availability Data Replication)

Replikace Enterprise Replication a replikace HDR (High-Availability Data Replication) nemohou odpovídat na výzvy, pokud je schéma ověřování vyžaduje. Z tohoto důvodu nesmí být replikace Enterprise Replication a replikace HDR konfigurovány na portu používajícím modul LDAP Authentication Support Module. Více informací o prostředcích pro vyhrazení portů replikaci Enterprise Replication a replikaci HDR naleznete v části “Volby zabezpečení připojení replikace Enterprise Replication a replikace HDR” na stránce 5-37.

Vývoj aplikací

Proces přípravy aplikace k odpovědím na výzvy vznesené modulem LDAP Authentication Support Module je stejný proces, který se používá při odpovědích na výzvy vznesené moduly PAM. Další informace naleznete v části “Vývoj aplikací pro moduly ověřování” na stránce 5-33.

Distribuované transakce s modulem LDAP Authentication Support Module, klientská rozhraní API a problémy kompatibility

Distribuované transakce, rozhraní API, která podporují modul LDAP Authentication Support Module a problémy kompatibility s ostatními produkty a nástroji IBM Informix jsou stejné u modulů PAM a LDAP Authentication Support Module. Další informace naleznete v části “Distribuované transakce” na stránce 5-35, “Klientská rozhraní API a moduly na podporu ověřování” na stránce 5-35 a “Problémy s kompatibilitou” na stránce 5-36.

Vývoj aplikací pro moduly ověřování

Metoda ověřování závisí na instalovaném modulu PAM nebo LDAP Authentication Support Module. Metoda může zahrnovat výzvu a odezvu. Když modul PAM nebo LDAP Authentication Support Module vznese výzvu, nastanou tyto procesy:

1. Databázový server předá výzvu klientovi.
2. Aplikace musí odpovědět na tuto výzvu pomocí funkce zpětného volání, která je poskytována ovladači IBM Informix CSDK a IBM Informix JDBC.
3. Pokud je server, ke kterému je klient připojen, nastaven na výzvy, musí aplikace pomocí ovladače CSDK nebo JDBC registrovat funkci zpětného volání.
4. Když ovladač CSDK nebo JDBC obdrží od serveru výzvu, je tato výzva předána aplikaci pomocí funkce zpětného volání.
5. Aplikace musí odpovědět na výzvu.
6. Ovladač CSDK nebo JDBC předá odpověď databázovému serveru.

Aplikace musí být připravena odpovědět na několik výzev a nemůže předpokládat počet výzev, ani o jaké výzvy se bude jednat.

Funkce zpětného volání by měla vypadat jako následující funkce:

```
mint ifx_pam_callback(mint (*callbackfunc_ptr)(char *challenge,
char *response, mint msg_style))
```

char *challenge

Znaková vyrovnávací paměť, ve které se nachází výzva předávaná serverem. Velikost vyrovnávací paměti je pevně stanovena na 512 a je definovaná parametrem PAM_MAX_MSG_SIZE v souboru **pam_appl.h**.

char *response

Znaková vyrovnávací paměť, do které má být uložena odezva poskytnutá uživatelem. Velikost vyrovnávací paměti je pevně stanovena na 512 a je definovaná parametrem PAM_MAX_RESP_SIZE v souboru **pam_appl.h**.

int msg_style

Obsahuje číslo označující typ zprávy předané serverem. Podle typu odezvy může aplikace ve funkci zpětného volání přijmout odpovídající akce.

Klientská aplikace musí před vytvořením prvního připojení zaregistrovat funkci zpětného volání. Pokud není funkce zpětného volání zaregistrovaná, když bylo vytvořeno první připojení k databázovému serveru a když server odpovídá na výzvu, potom ESQL/C vrátí chybu -1809.

Níže uvedený příklad znázorňuje velmi jednoduchý program, který nejdříve funkci zpětného volání zaregistruje a potom registraci zruší.

```
#include <stdio.h>
#include <security/pam_appl.h>
```

```
static int user_callback(char *challenge, char *response,
```

```

int msg_style);

int main(void)
{
    EXEC SQL char passwd[]="password";
    int retval = 0;

    /* nejdříve je třeba zaregistrovat zpětné volání */
    retval = ifx_pam_callback(user_callback);

    if (retval == -1)
    {
        printf("Při registraci zpětného volání došlo k chybě.\n");
        return (-1);
    }
    else
    {
        EXEC SQL database test; /* úspěšně připojení */
        /* Všimněte si, že se jedná o implicitní připojení. Takže
        * by aplikace měla být připravena odpovídat na výzvy.*/
        printf ("Kód sqlcode při připojení PAM = %d\n", SQLCODE);
    }

    retval = ifx_pam_callback(NULL); /* zrušení registrace funkce
    * zpětného volání */

    if (retval == -1)
    {
        printf("Při registraci zpětného volání došlo k chybě.\n");
        return (-1);
    }
    else
    {
        /* Toto připojení vrací chybu -1809, protože funkce zpětného
        * volání byla neregistrovaným příkazem */
        EXEC SQL database test;
        printf ("Kód sqlcode při připojení = %d\n", SQLCODE);
    }
    return 0;
}

static int user_callback(char *challenge, char *response,
int msg_style)
{
    switch (msg_style)
    {
        /* Pokud styl msg_je PAM_PROMPT_ECHO_OFF, aplikace
        * by neměla opakovat uživatelskou odezvu. */
        case PAM_PROMPT_ECHO_OFF:
        case PAM_PROMPT_ECHO_ON :
            printf("%s: %d:", challenge, msg_style);
            scanf("%.*s", PAM_MAX_RESP_SIZE, response);
            break;
    }
}

```



```

        case PAM_ERROR_MSG:
        case PAM_TEXT_INFO:
        default:
            printf("%s: %d\n", challenge, msg_style);
            break;
    }
    return 0;
}

```

Distribuované transakce

Distribuovaná připojení nemohou odpovídat na výzvy, protože nelze předvídat, kdy bude vytvořeno distribuované připojení. Ověřování u vzdálených serverů musí být prováděno v rámci databáze. Pro správu vzdálených připojení je možné použít novou databázi nazvanou **sysusers**.

Pokud je na vzdálených serverech povolen modul PAM nebo LDAP Authentication Support Module, administrátor systému musí zadat pro každý vzdálený server oprávněné uživatele do tabulky **sysauth** v databázi **sysusers**.

Databáze: **sysusers**

Tabulka: **sysauth**

Tabulka **sysauth** v databázi **sysusers** má následující strukturu:

Sloupec	Struktura
Username	char(32)
Groupname	char(32)
Servers	char(128)
Hosts	char(128)

Tabulka může obsahovat pro jediného uživatele několik řádků, aby bylo možné provádět ověřování u několika serverů a hostitelů.

Následující příklad povolí serveru přijímat distribuované transakce od uživatele **user1** patřícího ke skupině **group1** z databázového serveru **server1** a od hostitele **host1.mycompany.com**:

```

insert into sysauth values ("user1", "group1", "server1",
    "host1.mycompany.com");

```

Klientská rozhraní API a moduly na podporu ověřování

Následující rozhraní API klienta IBM Informix podporují moduly PAM a LDAP Authentication Support Module:

- ESQ/C,
- ODBC,

- JDBC.

Ostatní rozhraní API nepodporují moduly PAM a LDAP Authentication Support Module. Pokud je chcete používat spolu s verzí serveru Dynamic Server, který má povolen modul pro ověřování, připojte se k DBSERVERALIAS, který v souboru **sqlhosts** nemá parametry modulu PAM.

Následující klientská rozhraní API, nástroje a aplikace nepodporují moduly PAM, ani LDAP Authentication Support Module:

- LibC++,
- Libdmi,
- OLEDB,
- aplikace v jazyku VB používající rozhraní ODBC,
- ilogin, dbping a zkušební připojení ODBC,
- ISA.

Problémy s kompatibilitou

Ne všechny produkty a nástroje IBM Informix podporují ověřování PAM nebo LDAP:

- Produkt IBM Informix-4GL nepodporuje přímo ověřování PAM nebo LDAP, protože nemá mechanismus pro identifikaci funkcí zpětného volání. Pokud ovšem produkt IBM Informix-4GL používá správnou verzi sady CSDK, můžete napsat program v jazyce C, který může být volán z produktu IBM Informix-4GL za účelem zpracování protokolu výzva - odezva. Pokud chcete implementovat modul PAM, proveďte migraci na novou verzi sady CSDK, změňte aplikaci tak, aby zaregistrovala zpětnou vazbu, která dokáže zpracovat výzvy a odezvy, a tuto aplikaci znovu zkompilujte.
- Produkty jako Informix SQL protokol výzvy a odpovědi nezpracovávají.
- Obslužné programy DB-Access, **dbexport**, **dbimport**, **dbload** a **dbschema** podporují modul PAM. Pokud obdrží výzvu, předají výzvu uživateli a budou čekat na odezvu. To se opakuje s každou výzvou, kterou modul PAM vznese.
- Obslužné programy pro administraci **onmode**, **onstat** a **oncheck** nepoužívají modul PAM, ale na portu s podporou pro modul PAM pracují, protože tyto nástroje pracují na všech portech serveru Dynamic Server.
- Ostatní obslužné programy serveru nepodporují modul PAM.

Pokud používáte jakékoli nástroje, které nepodporují ověřovací moduly PAM nebo LDAP, připojte se k serveru DBSERVERALIAS, který neobsahuje v souboru SQLHOSTS parametry modulu PAM.

Volby zabezpečení připojení replikace Enterprise Replication a replikace HDR

Můžete zvýšit zabezpečení u připojení replikace Enterprise Replication a replikace HDR (High-Availability Data Replication) pomocí nové konfigurační volby v souboru `INFORMIXSQLHOSTS`.

Do pole volby (páté) v souborech `INFORMIXSQLHOSTS` přidejte položku `s=6`, abyste označili, že odpovídající port přijímá pouze požadavky na připojení replikace Enterprise Replication nebo HDR. Všechny ostatní typy požadavků na připojení budou odmítnuty s chybou číslo -25539, invalid connection type (neplatný typ připojení).

Následuje přehled záznamu souboru `INFORMIXSQLHOSTS`:

```
dbservername nettype hostname servicename s=6
```

Například:

```
ifxer1 oltlitcp mc001 er_port s=6,další_parametry_replikace_ER
```

Když nastavíte hodnotu `s=6`, budou požadavky na připojení replikace ER a HDR ověřovány pomocí nového mechanismu. Systémový administrátor by měl v adresáři `$INFORMIXDIR/etc` vytvořit soubor `hosts.equiv` a přidat do něho názvy uzlů účastnících se replikace Enterprise Replication a HDR (názvy hostitelů, které se nacházejí ve třetím sloupci souboru `INFORMIXSQLHOSTS`), na každý řádek jeden název. Formát tohoto souboru je podobný jako formát souboru `/etc/hosts.equiv` v operačním systému UNIX. Soubor by měl být vlastněn uživatelem `informix`, který patří do skupiny `informix` a oprávnění by měla být omezena tak, mohl tento soubor měnit nanejvýš uživatel `informix` (pomocí oktalových oprávnění, vhodná je jedna z hodnot 644, 640, 444 nebo 440).

V případě konfigurace, kdy replikační servery jsou na stejném počítači, není soubor `$INFORMIXDIR/etc/hosts.equiv` potřeba.

Následující omezení se týkají této volby zabezpečení:

- Pro porty vyhrazené pouze pro replikaci Enterprise Replication nebo HDR by měla být přítomna pouze volba zabezpečení `s=6`. Když je použita volba `s=6`, neměly by se používat žádné jiné volby zabezpečení (`s=0,1,2,3,4`).
- Tato volba je v prostředí databázového serveru specifická pro replikaci Enterprise Replication a HDR, a neměla by se proto používat v klientském prostředí. V případě, že je tato volba nastavena v souboru `SQLHOSTS` a klient se pokusí použít název přidruženého serveru, budou klienti vracet chybu.

Další informace o replikaci HDR obsahuje Kapitola 20, “Replikace HDR (High-Availability Data Replication)”, na stránce 20-1. Další informace o replikaci Enterprise Replication naleznete v příručce *IBM Informix: Dynamic Server Enterprise Replication Guide*.

Zabezpečení externích uživatelských rutin (UDR)

Externí rutiny se sdílenými knihovnamí, které se nacházejí mimo databázový server, mohou představovat bezpečnostní riziko. Externí rutiny zahrnují uživatelské rutiny (UDR) a rutiny v modulech DataBlade. Administrátor databázového serveru (DBSA), ve výchozím nastavení uživatel **informix**, může implementovat bezpečnostní opatření, která určují, kteří uživatelé mohou registrovat externí rutiny. Tak je zabráněno neoprávněným osobám v registraci externích rutin.

Pomocí konfiguračního parametru `IFX_EXTEND_ROLE` můžete omezit schopnost uživatelů registrovat externí rutiny.

Výchozí hodnota konfiguračního parametru `IFX_EXTEND_ROLE` je `Off`. Pokud chcete udělit uživateli oprávnění vytvořit nebo vypustit uživatelskou rutinu, která má klauzuli `EXTERNAL`, nastavte konfigurační parametr `IFX_EXTEND_ROLE` na hodnotu `On`. Když je tento parametr nastaven na hodnotu `On`, je role `EXTEND` funkční a vy můžete uživateli udělit oprávnění vytvořit nebo vypustit externí rutinu, která bude mít klauzuli `EXTERNAL`.

Když určitému uživateli udělíte roli `EXTEND`, tabulka systémového katalogu **sysroleauth** se aktualizuje, aby odrážela novou vestavěnou roli.

Po nastavení konfiguračního parametru `IFX_EXTEND_ROLE` na hodnotu `On` použijte následující syntaxi k udělení oprávnění konkrétním uživatelům nebo k odvolání oprávnění od těchto uživatelů.

- `GRANT extend To uživatelské_jméno`
- `REVOKE extend From uživatelské_jméno`

Pokud přístup k uživatelským rutinám nechcete omezit, nastavte konfigurační parametr na hodnotu `Off`. Když je parametr `IFX_EXTEND_ROLE` nastaven na hodnotu `Off`, není role `EXTEND` funkční a externí rutiny mohou být registrovány libovolným uživatelem.

Další informace naleznete v příručce *IBM Informix: Guide to SQL Syntax*.

Část 2. Správa disků, paměti a procesů

Kapitola 6. Virtuální procesory a jednotkové procesy

Virtuální procesory	6-2
Jednotkové procesy	6-3
Typy virtuálních procesorů	6-4
Výhody virtuálních procesorů	6-6
Sdílení zpracování	6-7
Úspory paměti a zdrojů	6-7
Paralelní zpracování	6-8
Přidávání a vypouštění virtuálních procesorů v režimu online	6-8
Vázání virtuálních procesorů k jednotce CPU	6-9
Obsluha jednotkových procesů virtuálními procesory	6-9
Řídící struktury	6-10
Přepínání kontextu	6-10
Zásobníky	6-12
Fronty	6-13
Fronty připravených procesů	6-13
Fronty spících procesů	6-14
Fronty čekajících procesů	6-15
Objekty mutex	6-15
Třídy virtuálních procesorů	6-16
Virtuální procesory CPU	6-16
Stanovení počtu potřebných virtuálních procesorů CPU	6-16
Spuštění na víceprocesorovém počítači	6-17
Spuštění na jednoprocessorovém počítači	6-17
Přidávání a vypouštění virtuálních procesorů CPU v režimu online	6-18
Jak zabránit stárnutí priority	6-18
Užití procesorové afinity	6-18
Uživatелеm definované třídy virtuálních procesorů	6-19
Stanovení potřebného počtu uživatelem definovaných virtuálních procesorů	6-20
Užití uživatelem definovaných virtuálních procesorů	6-20
Určení uživatelem definovaného virtuálního procesoru	6-21
Přirazení UDR k uživatelem definované třídě virtuálních procesorů	6-21
Přidávání a vypouštění uživatelem definovaných virtuálních procesorů v režimu online	6-22
Virtuální procesory Java	6-22
Virtuální procesory diskového vstupu - výstupu	6-22
Priority vstupu - výstupu	6-23
Vstup - výstup logického protokolu	6-23
Vstup - výstup fyzického protokolu	6-24
Asynchronní vstup - výstup	6-24
Síťové virtuální procesory	6-26
Určení síťových přípojení	6-27
Spouštění jednotkových procesů cyklického dotazování na virtuálních procesorech CPU nebo síťových virtuálních procesorech	6-27
Stanovení počtu síťových virtuálních procesorů	6-28
Určení jednotkových procesů listen a cyklického dotazování pro připojení klienta k serveru	6-28

Spuštění několika jednotkových procesů typu listener	6-31
Virtuální procesor Communications Support Module	6-33
Šifrovací virtuální procesory	6-33
Optický virtuální procesor	6-34
Virtuální procesor typu audit.	6-34
Virtuální procesor typu miscellaneous	6-34

Obsah kapitoly

Tato kapitola popisuje, jak databázový server využívá virtuální procesory a jednotkové procesy v rámci virtuálních procesorů ke zlepšení výkonu. Popisuje typy virtuálních procesorů a způsob, jakým virtuální procesory spouštějí jednotkové procesy.

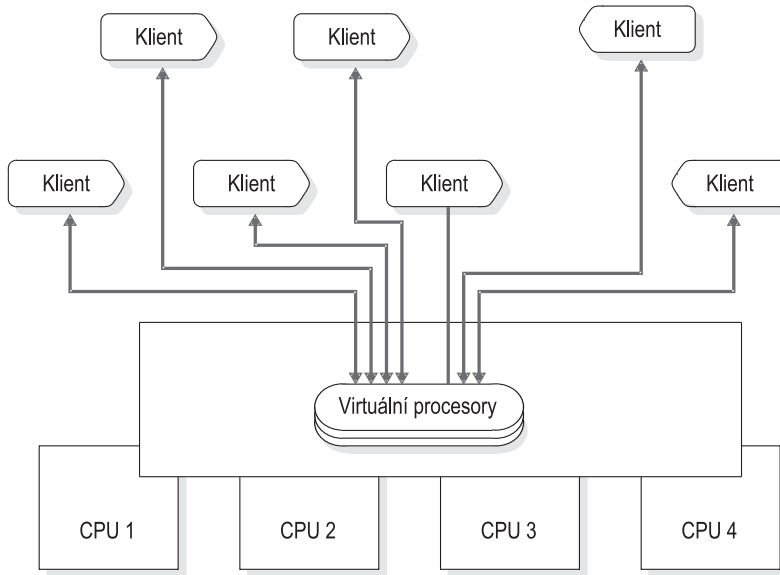
Virtuální procesory

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 6-1 znázorňuje vztah klientských aplikací a virtuálních procesorů. Malý počet virtuálních procesorů obsluhuje podstatně větší počet klientských aplikací nebo dotazů.

Klientské aplikace



Obrázek 6-1. 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

UNIX

Windows

Popis jednotkového procesu

Jednotkový proces je úloha, kterou virtuální procesor plánuje ke zpracování interně.

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 6-8.

Uživatelský jednotkový proces je jednotkový proces databázového serveru, který obsluhuje požadavky klientských aplikací. Uživatelské jednotkové procesy zahrnují jednotkové procesy relací nazývané jednotkové procesy **sqlxec**, 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 zotavení, 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 *IBM Informix: Dynamic Server Performance Guide*.

Typy virtuálních procesorů

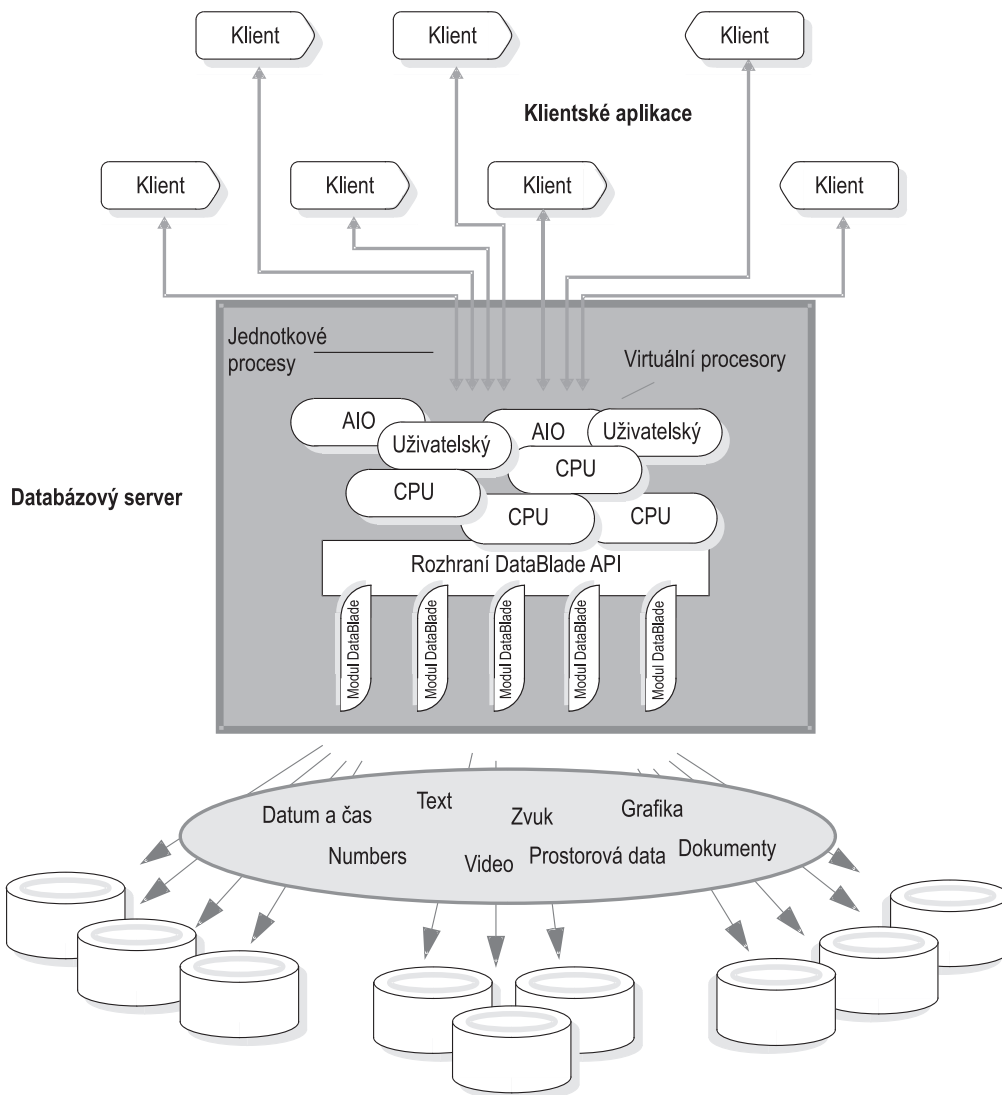
Tabulka 6-1 znázorňuje *třídy* virtuálních procesorů a druhy zpracování, které provádějí. Každá třída virtuálního procesoru je vyhrazena pro zpracování určitého typu jednotkových procesů.

Počet virtuálních procesorů konfigurovaných pro kteroukoli třídu je závislý na počtu dostupných fyzických procesorů (jednotek CPU), velikosti fyzické paměti a používaných databázových aplikacích.

Tabulka 6-1. Třídy virtuálních procesorů

Třída virtuálního procesoru	Kategorie	Účel
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 (UNIX)	Optický podsystém	Zajišťuje vstup - výstup optického disku.
ADM	Administrativní	Zajišťuje administrativní funkce.
ADT	Audit	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>	Uživatelé 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 JVM (Java Virtual Machine).

Obrázek 6-2 znázorňuje hlavní komponenty a rozšiřitelnost databázového serveru.



Obrázek 6-2. 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 6-10.

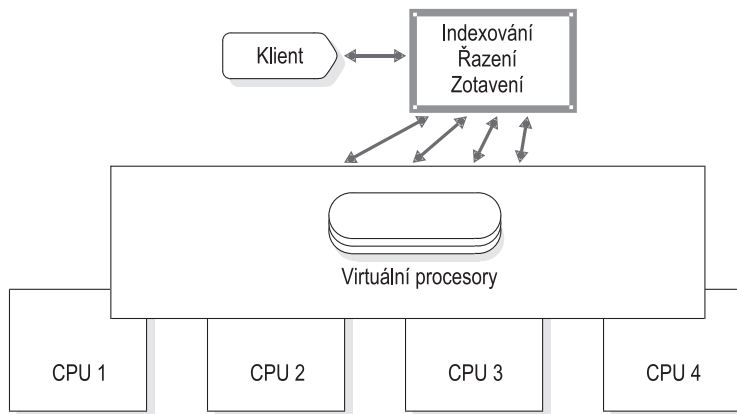
Paralelní zpracování

V následujících případech mohou virtuální procesory třídy CPU spouštět několik jednotkových procesů relací pro jednoho klienta současně:

- Při vytváření indexu.
- Při řazení.
- Při zotavení.
- Při prohledávání.
- Při 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 6-3 znázorňuje paralelní zpracování. Pokud klient zahájí indexování, řazení nebo logické zotavení, 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 6-3. 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 7-4.

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 7-1.

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 spouštěn 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 6-18.

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 (**sqlexec**), 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:

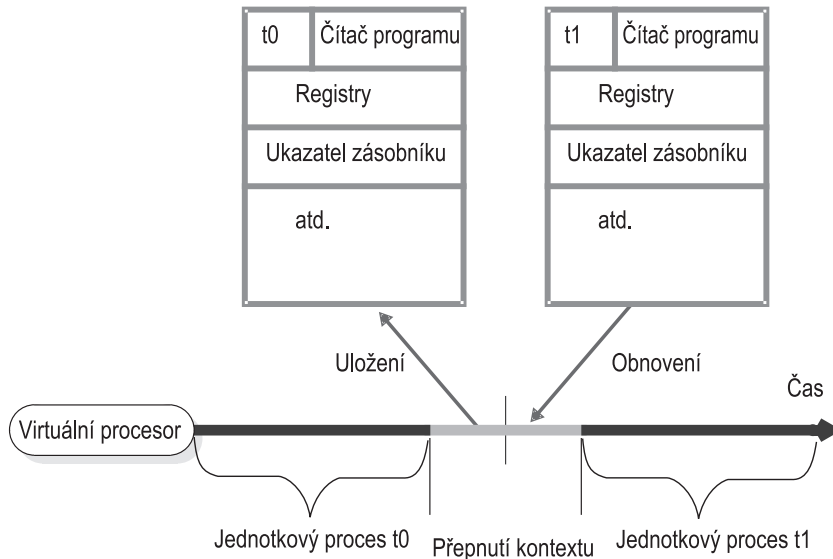
- V předem určeném bodu v kódu.
- Pokud jednotkový proces nelze nadále provádět, dokud nebude splněna určitá podmínka.

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 zámeček 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 6-4 znázorňuje, jak virtuální procesor provede přepnutí kontextu.

Řídicí bloky jednotkových procesů



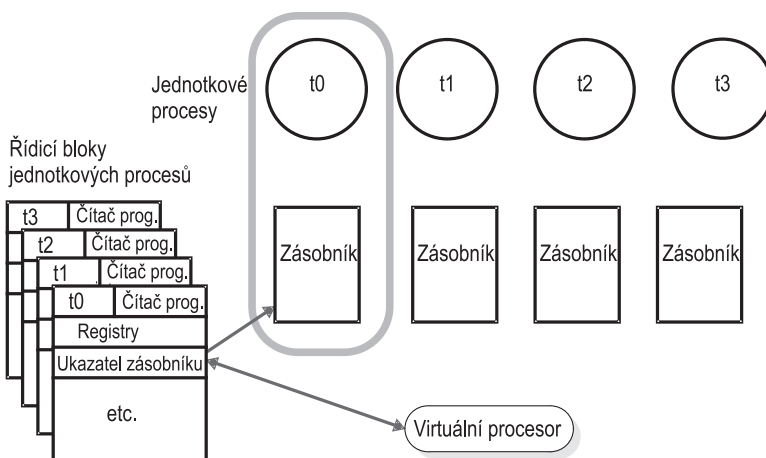
Obrázek 6-4. Přepnutí kontextu: Jak virtuální procesor přepíná z jednoho jednotkového procesu na jiný

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 8-24.

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 6-5 znázorňuje, jak virtuální procesor využívá zásobník k oddělení nesdílených dat jednotkových procesů relací.



Obrázek 6-5. 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 7-6. Další informace o přidávání virtuálních procesorů, zatímco je databázový server v režimu online naleznete v části “Přidání virtuálních procesorů v režimu online” na stránce 7-4.

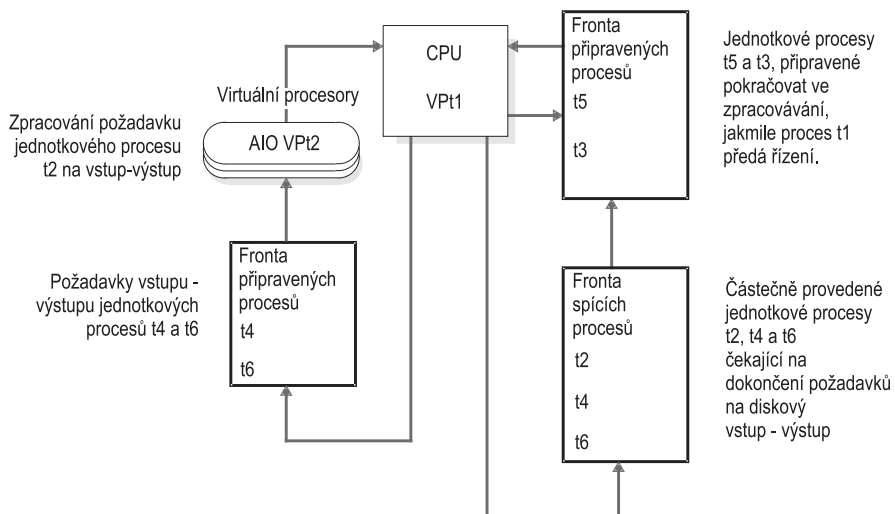
Fronty spících procesů

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 uspány 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 uspán, 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 uspán *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 6-6 znázorňuje řazení jednotkových procesů databázového serveru do fronty k provádění operací vstupu-výstupu databáze.



Obrázek 6-6. 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

Objekty typu mutex (z anglického *mutually* (vzájemně) *exclusive* (výlučný)), označované též jako *zámek* (*latch*) jsou 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 mutex, jakmile je

dostupný, jeho 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 zámek latch" na stránce 9-10.

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.

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 o rozhodnutí, kolik virtuálních procesorů CPU potřebujete, naleznete v části "Spouštění jednotkových procesů cyklického dotazování na virtuálních procesorech CPU nebo síťových virtuálních procesorech" na stránce 6-27.

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: Doporučujeme použít parametr VPCLASS namísto parametrů AFF_SPROC, AFFNPROCS, NOAGE, NUMCPUVPS a NUMAIOVPS. 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 6-21.

Spuštění na víceprocesorovém počítači

Spouštíte-li několik virtuálních procesorů CPU na víceprocesorovém počítači, nastavte parametr MULTIPROCESSOR v souboru ONCONFIG na hodnotu 1. Pokud nastavíte MULTIPROCESSOR na hodnotu 1, bude databázový server provádět uzamykání způsobem vhodným pro víceprocesorový počítač. 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 MULTIPROCESOR na hodnotu 0. Chcete-li databázový server spustit pouze s jedním virtuálním procesorem CPU, nastavte parametr SINGLE_CPU_VP na hodnotu 1.

Nastavení parametru MULTIPROCESOR na hodnotu 0 umožní databázovému serveru obcházet zamykání, které je nezbytné ke spuštění více procesů na víceprocesorovém počítači. Další informace o konfiguračním parametru MULTIPROCESOR 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 7-4.

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 7-4 a “Vypouštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů” na stránce 7-5.

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.

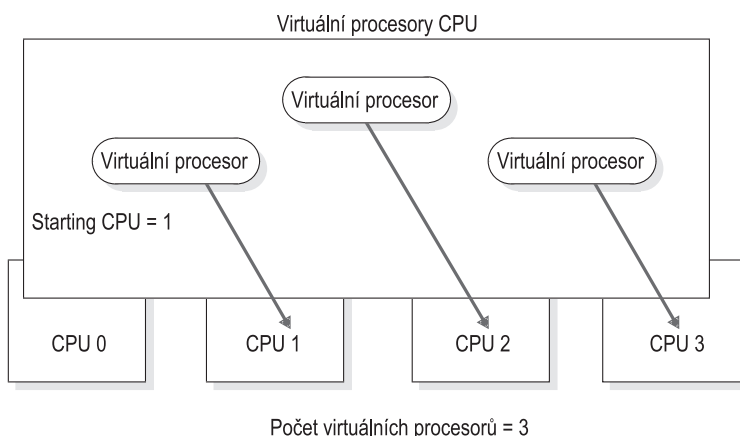
Pokud operační systém umožňuje vypnout stárnutí priority, můžete ho vypnout stanovením hodnoty `noage` pro položku priority v konfiguračním parametru VPCLASS. Další informace naleznete v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

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 6-7 ilustruje koncept procesorové afinity.



Obrázek 6-7. Procesorová afinita

Pouze pro UNIX

Chcete-li zjistit, zda je procesorová afinita podporována na platformě UNIX, nahlédněte do souboru Poznámky k počítači.

Konec Pouze 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 příkladu záznam afinity přiřadí virtuální procesor k jednotkám CPU 1, 2 a 3. K dispozici jsou tři virtuální procesory (určené v *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*.

Uživatelsky definované třídy virtuálních procesorů

Můžete definovat speciální třídy virtuálních procesorů ke spuštění uživatelských rutin nebo modulu DataBlade. Uživatelské rutiny bývají obvykle napsány, aby podporovaly

uživatelé 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). Uživatelem definované třídy virtuálních procesorů se nazývají též *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

Uživatelé 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 7-2 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 **GreaterThanOrEqualTo** a určuje, že volání směřovaná na tuto rutinu mají být provedena uživatelem definovanou třídou VP nazvanou UDR:

```
CREATE FUNCTION GreaterThanOrEqualTo(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 **GreaterThanOrEqualTo** selžou.

Tip: 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 tomu 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 7-4 a “Vypouštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů” na stránce 7-5.

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 úplně 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 *IBM Informix: 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, použije 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.

Pouze pro UNIX

Chcete-li zjistit, zda použitá platforma UNIX podporuje vstup - výstup KAIO, nahlédněte do souboru Poznámky k počítači.

Konec Pouze 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 6-24.

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 rozděljuje podle priorit různé typy diskového vstupu-výstupu, jak znázorňuje Tabulka 6-2.

Tabulka 6-2. 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-výstupu 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. K této třídě virtuálních procesorů nejsou přidružené žádné 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 změnily od posledního *kontrolního bodu*. (Další informace o kontrolních bodech naleznete v části "Kontrolní body" na stránce 16-8.) Na počátku obnovení, ještě před zpracováním transakcí z logického protokolu, použije databázový server fyzický protokol k obnovení předobrazů stránek prostoru dbSPACE, které se změnily 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. K této třídě virtuálních procesorů nejsou přidruženy žádné 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.

Pouze pro UNIX

V serveru Informix je vstup - výstup KAIO implementován tehdy, pokud je databázový server instalován na platformu, která tuto funkci podporuje. Administrátor databázového serveru vstup - výstup KAIO nekonfiguruje. Chcete-li zjistit, zda použitá platforma podporuje KAIO, nahlédněte do souboru poznámek k počítači.

Konec Pouze pro UNIX

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 7-4.

Virtuální procesory AIO nelze vypouštět, když je databázový server v režimu online.

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** umožňuje monitorovat délku front *gfd* pro virtuální procesory AIO. Další informace naleznete v části “Monitorování virtuálních procesorů” na stránce 7-6.

Pokud databázový server implementuje na použité platformě vstup - výstup KAIO a pokud jsou všechny prostory dbspace tvořeny diskovým prostorem s přímým přístupem, bude postačovat jeden virtuální procesor AIO.

Pokud pro některé bloky používáte soubory s použitím vyrovnávací paměti, přestože databázový server implementuje vstup - výstup KAIO, přidejte každému aktivnímu prostoru dbspace, který je tvořen bloky souborů s vyrovnávací pamětí dva virtuální procesory AIO. Pokud na použité platformě není vstup - výstup KAIO implementován, přidejte dva virtuální procesory AIO každému disku, ke kterému databázový server často přistupuje.

Přidejte dostatečný počet virtuálních procesorů AIO, aby server dokázal zpracovávat maximální počet požadavků vstupu-výstupu. Obecně není přidělení příliš mnoha virtuálních procesorů AIO na závadu.

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.

Pouze pro UNIX

- Prostřednictvím sdílené paměti.

Konec Pouze 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 3-19.

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. Podrobný popis tohoto konfiguračního parametru 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 spuštěny 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 síť 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** spouští interní jednotkový proces nazývaný *jednotkový proces typu listener* pro každý název dbservername, který určíte pomocí parametrů DBSERVERNAME a DBSERVERALIASES v souboru ONCONFIG. Chcete-li určit port, na kterém databázový server očekává připojení pro každý z těchto záznamů dbservername, přiřadte mu jedinečnou kombinaci záznamů **hostname** a **service name** v souboru **sqlhosts**. Záznam v souboru nebo registru **sqlhosts**, který uvádí Tabulka 6-3, například způsobí, že databázový server **soc_ol1** spustí jednotkový proces typu listener pro **port1** u hostitele nebo na síťové adrese **myhost**.

Tabulka 6-3. Jednotkový proces typu listener pro každý port, na kterém databázový server očekává připojení

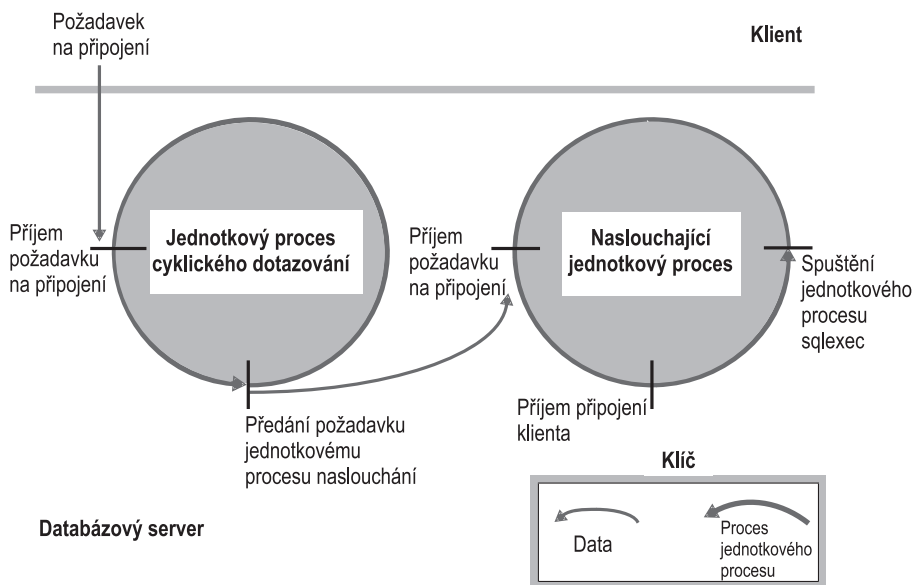
dbservername	nettype	hostname	název služby
soc_ol1	onsoctcp	myhost	port1

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 6-28.

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 6-27 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 6-8 znázorňuje roli jednotkových procesů typu listener a jednotkových procesů cyklického dotazování při vytváření spojení s klientskou aplikací.



Obrázek 6-8. 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

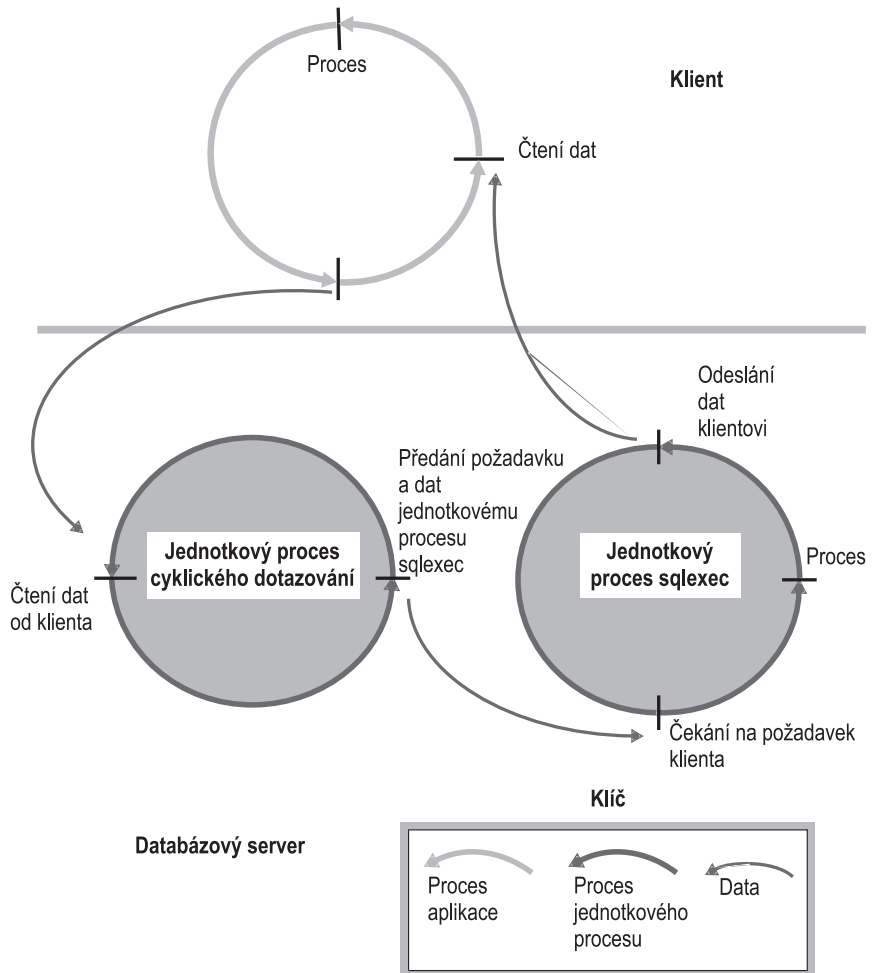
možné, zapisuje jednotkový proces **sqlxec** zpět přímo do klienta bez pomoci jednotkového procesu cyklického dotazování. Jednotkový proces cyklického dotazování obecně načítá data z klienta a jednotkový proces **sqlxec** odesílá tato data ke klientovi.

Pouze pro UNIX

V případě připojení prostřednictvím sdílené paměti umístí jednotkový proces cyklického dotazování zprávu do komunikační části sdílené paměti.

Konec Pouze pro UNIX

Obrázek 6-9 znázorňuje základní úlohy, které provádějí jednotkový proces cyklického dotazování a jednotkový proces **sqlxec** při komunikaci s klientskou aplikací.



Obrázek 6-9. Role jednotkových procesů cyklického dotazování a jednotkových procesů typu sqlexec při komunikaci s klientskou aplikací

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 6-10, například definuje dva další názvy dbservername **soc_ol2** a **soc_ol3** pro instanci databázového serveru označenou jako **soc_ol1**.

```
DBSERVERNAME      soc_ol1
DBSERVERALIASES   soc_ol2,soc_ol3
```

Obrázek 6-10. Definování několika názvů dbservername pro vícenásobná připojení stejného typu

Jakmile určíte další názvy dbservername pro databázový server, musíte určit kombinaci rozhraní a protokolu a port pro každý z nich v souboru nebo registru **sqlhosts**. Každý port je určen jedinečnou kombinací záznamů **hostname** a **servicename**. Záznamy **sqlhosts**, které zobrazuje Tabulka 6-4, například způsobí, že databázový server spustí tři jednotkové procesy typu listener pro kombinaci rozhraní a protokolu **onsoctcp**, jednu pro každý z definovaných portů.

Tabulka 6-4. Záznamy sqlhosts k naslouchání na několika portech pro jedinou kombinaci rozhraní a protokolu

dbservername	nettype	hostname	název služby
soc_ol1	onsoctcp	myhost	port1
soc_ol2	onsoctcp	myhost	port2
soc_ol3	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 6-4, 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 3-10.

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 6-10, záznamy v souboru nebo registru

sqlhosts, které uvádí Tabulka 6-5 na stránce 6-33, způsobí, že databázový server spustí tři jednotkové procesy typu listener pro tutéž kombinaci rozhraní a protokolu (podobně, jak to učinily záznamy, které uvádí Tabulka 6-4). 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 6-5. 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 části “Použití šifrování na úrovni sloupců” na stránce 5-23.

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

Databázový server spustí jeden virtuální procesor třídy audit (ADT), přepnete-li na režim auditu nastavením parametru ADTMODE v souboru ONCONFIG na hodnotu 1. Další informace o auditu databázového serveru naleznete v příručce *IBM Informix: Trusted Facility 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 kilobajtů.

Kapitola 7. Správa virtuálních procesorů

Nastavení konfiguračních parametrů virtuálních procesorů	7-1
Nastavení parametrů virtuálního procesoru pomocí textového editoru	7-2
Určení parametru VPCLASS	7-2
Zakázání stárnutí priority (UNIX)	7-3
Nastavení parametrů virtuálního procesoru pomocí programu ISA	7-3
Nastavení parametrů virtuálního procesoru pomocí obslužného programu ON-Monitor	7-3
Spuštění a zastavení virtuálních procesorů	7-3
Přidání virtuálních procesorů v režimu online	7-4
Přidání virtuálních procesorů v režimu online pomocí obslužného programu onmode	7-4
Přidání virtuálních procesorů v režimu online pomocí obslužného programu ON-Monitor (UNIX)	7-4
Přidání síťových virtuálních procesorů	7-5
Vypuštění virtuálních procesorů třídy CPU a uživatelem definovaných virtuálních procesorů	7-5
Monitorování virtuálních procesorů	7-6
Monitorování virtuálních procesorů pomocí obslužných programů příkazového řádku	7-6
onstat -g ath	7-6
onstat -g glo	7-6
onstat -g ioq	7-7
onstat -g rea	7-8
Monitorování virtuálních procesorů pomocí tabulek SMI.	7-8

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 6, “Virtuální procesory a jednotkové procesy”, na stránce 6-1.

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)

Pouze pro UNIX

- ON-Monitor

Konec Pouze pro UNIX

K zavedení jakýchkoliv změn konfiguračních parametrů je nutné ukončit a restartovat databázový server. Další informace naleznete v části “Opětovná inicializace sdílené paměti” na stránce 9-8.

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 7-1 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 6-16.

Tabulka 7-1. 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 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 (JVP).
VPCLASS	noyield	Určuje virtuální procesor nepředávající řízení.

Určení parametru VPCLASS

Doporučujeme používat parametr VPCLASS místo parametrů NUMCPUVPS, NUMAIOVPS, NOAGE, AFF_SPROC a AFF_NPROCS. Můžete určit až 128-znakový název parametru VPCLASS. 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*, “Uživatelé definované třídy virtuálních procesorů” na stránce 6-19 a “Virtuální procesory Java” na stránce 6-22.

Zakázání stárnutí priority (UNIX)

Chcete-li zakázat proces stárnutí priority na platformách umožňujících tuto funkci, použijte parametr VPCLASS s volbou *noage*.

Doporučené hodnoty těchto parametrů databázového serveru v rámci 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 6-16.

Databázový server umožňuje spustit nanejvýš 1000 virtuálních procesorů.

Jakmile je databázový server v režimu online, můžete v případě potřeby zvýšit výkon spuštěním dalších virtuálních procesorů. 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 7-5.

K ukončení databázového serveru a tudíž i ukončení všech virtuálních procesorů použijte příkaz **onmode -k**. Více informací o použití příkazu **onmode -k** naleznete v kapitole Obslužné programy v části *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 6-21.

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 číslo 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 (UNIX)

Chcete-li k databázovému serveru v režimu online přidat virtuální procesory pomocí programu ON-Monitor, vyberte volbu **Modes> Add-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 7-4.

Přidání síťových virtuálních procesorů

Když přidáváte síťové virtuální procesory, přidáváte jednotkové procesy cyklického dotazování, z nichž každý vyžaduje spuštění vlastního virtuálního procesoru. Pokoušíte-li se přidat jednotkové procesy cyklického dotazování pro protokol, zatímco je databázový server v režimu online, a v parametru NETTYPE jste určili, že jednotkové procesy cyklického dotazování jsou spouštěny na virtuálních procesorech třídy CPU, databázový server nespustí nové jednotkové procesy cyklického dotazování, nejsou-li k dispozici 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 procesů cyklického dotazování pro typ nettype

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 vypustí, například, 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 -číslo.
```

Více informací naleznete v části “Spouštění jednotkových procesů cyklického dotazování na virtuálních procesorech CPU nebo síťových virtuálních procesorech” na stránce 6-27.

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 “Uživatelem definované třídy virtuálních procesorů” na stránce 6-19.

Monitorování virtuálních procesorů

Monitorování virtuálních procesorů slouží ke zjištění, zdali 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 *IBM Informix: Dynamic Server Performance Guide*.

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.

Obrázek 7-1 znázorňuje příklad výstupu této volby.

```

MT global info:
sessions threads vps lngspins
1 15 8 0

```

Virtual processor summary:

class	vps	usercpu	syscpu	total
cpu	3	479.77	190.42	670.18
aio	1	0.83	0.23	1.07
pio	1	0.42	0.10	0.52
lio	1	0.27	0.22	0.48
soc	0	0.00	0.00	0.00
tli	0	0.00	0.00	0.00
shm	0	0.00	0.00	0.00
adm	1	0.10	0.45	0.55
opt	0	0.00	0.00	0.00
msc	1	0.28	0.52	0.80
adt	0	0.00	0.00	0.00
total	8	481.67	191.93	673.60

Individual virtual processors:

vp	pid	class	usercpu	syscpu	total
1	1776	cpu	165.18	40.50	205.68
2	1777	adm	0.10	0.45	0.55
3	1778	cpu	157.83	98.68	256.52
4	1779	cpu	156.75	51.23	207.98
5	1780	lio	0.27	0.22	0.48
6	1781	pio	0.42	0.10	0.52
7	1782	aio	0.83	0.23	1.07
8	1783	msc	0.28	0.52	0.80
		tot	481.67	191.93	673.60

Obrázek 7-1. výstup `onstat -g glo`

onstat -g ioq

Pomocí volby **onstat -g ioq** lze zjistit, zda je nutné přiřadit další virtuální procesory typu AIO. Příkaz **onstat -g ioq** zobrazí délku front vstupu-výstupu ve sloupci **len**, jak znázorňuje Obrázek 7-2.

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.

```
onstat -g ioq
```

AIO I/O queues:

q	name/id	len	maxlen	totalops	dskread	dskwrite	dskcopy
adt	0	0	0	0	0	0	0
msc	0	0	1	12	0	0	0
aio	0	0	4	89	68	0	0
pio	0	0	1	1	0	1	0
lio	0	0	1	17	0	17	0
kio	0	0	0	0	0	0	0
gfd	3	0	3	254	242	12	0
gfd	4	0	17	614	261	353	0

Obrázek 7-2. Výstup `onstat -g ioq` a `onstat -d`

onstat -g rea

K monitorování počtu jednotkových procesů ve frontě připravených procesů použijte volbu **onstat -g rea**. Pokud ve frontě připravených procesů roste počet jednotkových procesů jedné třídy virtuálních procesorů (například třídy CPU), bude možná nutné rozšířit konfiguraci o další virtuální procesory. Obrázek 7-3 znázorňuje výstup volby **onstat -g rea**.

```
Ready threads:
tid    tcb      rstcb  prty   status          vp-class  name
6      536a38  406464  4      ready          3cpu     main_loop()
28     60cfe8  40a124  4      ready          1cpu     onmode_mon
33     672a20  409dc4  2      ready          3cpu     sqlexec
```

Obrázek 7-3. Výstup `onstat -g rea`

Monitorování virtuálních procesorů pomocí tabulek SMI

Informace o právě spuštěných virtuálních procesorech vyhledáte v tabulce **sysvpprof**. Tato tabulka obsahuje následující sloupce.

sloupec	Popis
vpid	Číslo ID virtuálního procesoru.
class	Třída virtuálního procesoru.
usercpu	Spotřebované minuty času uživatelské jednotky CPU.
syscpu	Spotřebované minuty času systémové jednotky CPU.

Kapitola 8. Sdílená paměť

Sdílená paměť	8-2
Použití sdílené paměti	8-3
Přidělování sdílené paměti	8-4
Velikost sdílené paměti	8-6
Jaké akce provést, pokud je překročena hodnota parametru SHMTOTAL	8-7
Procesy, které se připojují ke sdílené paměti	8-8
Jak se klient připojuje ke komunikační části sdílené paměti (systém UNIX)	8-8
Jak se ke sdílené paměti připojují obslužné programy	8-9
Jak se ke sdílené paměti připojují virtuální procesory	8-9
Získání hodnot klíčů segmentů sdílené paměti	8-10
Určení místa připojení prvního segmentu sdílené paměti	8-10
Připojení dalších segmentů sdílené paměti	8-11
Definování adresy dolní hranice sdílené paměti	8-11
Rezidentní segmenty sdílené paměti	8-12
Rezidentní část sdílené paměti	8-13
Záhlaví sdílené paměti	8-13
Společná oblast vyrovnávacích paměti ve sdílené paměti	8-13
Přetečení vyrovnávacích paměti do virtuální části	8-15
Velikost vyrovnávací paměti	8-15
Vyrovnávací paměť logického protokolu	8-15
Vyrovnávací paměť fyzického protokolu	8-16
Vyrovnávací paměť replikace High-Availability Data Replication	8-17
Tabulka zámek	8-17
Virtuální část sdílené paměti	8-18
Správa virtuální části sdílené paměti	8-18
Velikost virtuální části sdílené paměti	8-19
Součásti virtuální části sdílené paměti	8-19
Interní tabulky sdílené paměti	8-20
Velké vyrovnávací paměti	8-23
Data relací	8-23
Data jednotkových procesů	8-23
Rychlá vyrovnávací paměť distribuce dat	8-24
Rychlá vyrovnávací paměť slovníku	8-25
Rychlá vyrovnávací paměť příkazů SQL	8-25
Řadičí paměť	8-25
Rutiny SPL a rychlá vyrovnávací paměť rutin UDR	8-26
Globální společná oblast	8-26
Komunikační část sdílené paměti (systém UNIX)	8-26
Část virtuálních rozšíření sdílené paměti	8-27
Řízení souběžného přístupu	8-27
Objekty mutex sdílené paměti	8-27
Zámky vyrovnávacích paměti ve sdílené paměti	8-27
Typy zámek vyrovnávacích paměti	8-28
Přístup jednotkových procesů databázového serveru k sdíleným vyrovnávacím pamětem	8-28

Fronty FIFO/LRU	8-28
Komponenty front LRU	8-29
Stránky seřazené podle nejdlejší doby nepoužití	8-29
Správa společné oblasti vyrovnávacích pamětí a front LRU	8-29
Počet front LRU, které je třeba nakonfigurovat	8-30
Počet jednotkových procesů čištění, které je třeba přidělit	8-31
Počet stránek přidávaných do front MLRU	8-31
Konec čištění fronty MLRU	8-32
Konfigurace dopředného čtení databázového serveru	8-33
Přístup jednotkových procesů databázového serveru ke stránkám ve vyrovnávací paměti	8-33
Vyprazdňování dat na disk	8-34
Vyprazdňování vyrovnávacích pamětí ve společné oblasti	8-34
Přednostní vyprazdňování předobrazů	8-35
Vyprazdňování vyrovnávací paměti fyzického protokolu	8-35
Synchronizace vyprazdňování vyrovnávacích pamětí	8-35
Popis aktivity vyprazdňování	8-36
Zápis na popředí	8-36
Zápisy front LRU	8-36
Zápisy bloků	8-37
Vyprazdňování vyrovnávací paměti logického protokolu	8-37
Po přípravě nebo dokončení transakce v databázi s protokolováním bez vyrovnávací paměti	8-38
Při ukončení relace, která používá databáze bez protokolování nebo protokolování bez vyrovnávací paměti	8-38
Pokud nastane kontrolní bod	8-38
Pokud byla změněna stránka, která nevyžaduje předobraz v souboru fyzického protokolu	8-38
Použití vyrovnávací paměti pro velké objekty	8-38
Zápis jednoduchých velkých objektů	8-39
Stránky blobpage a sdílená paměť	8-39
Vytváření jednoduchých velkých objektů	8-39
Vytvoření vyrovnávacích pamětí stránek blobpage	8-39
Přístup k inteligentním velkým objektům	8-41
Využití paměti na 64bitových platformách	8-42

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 9, “Správa sdílené paměti”, na stránce 9-1.

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 operací I/O.
- Vysokorychlostní komunikace mezi procesy.

Sdílená paměť umožňuje databázovému serveru snížit celkový objem využité paměti, protože účastníci se procesy, v tomto případě virtuální procesory, nemusejí udržovat soukromé kopie dat uložených ve sdílené paměti.

Sdílená paměť omezuje počet diskových operací I/O, 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 operací I/O se zmenšuje také celková doba provádění požadavků.

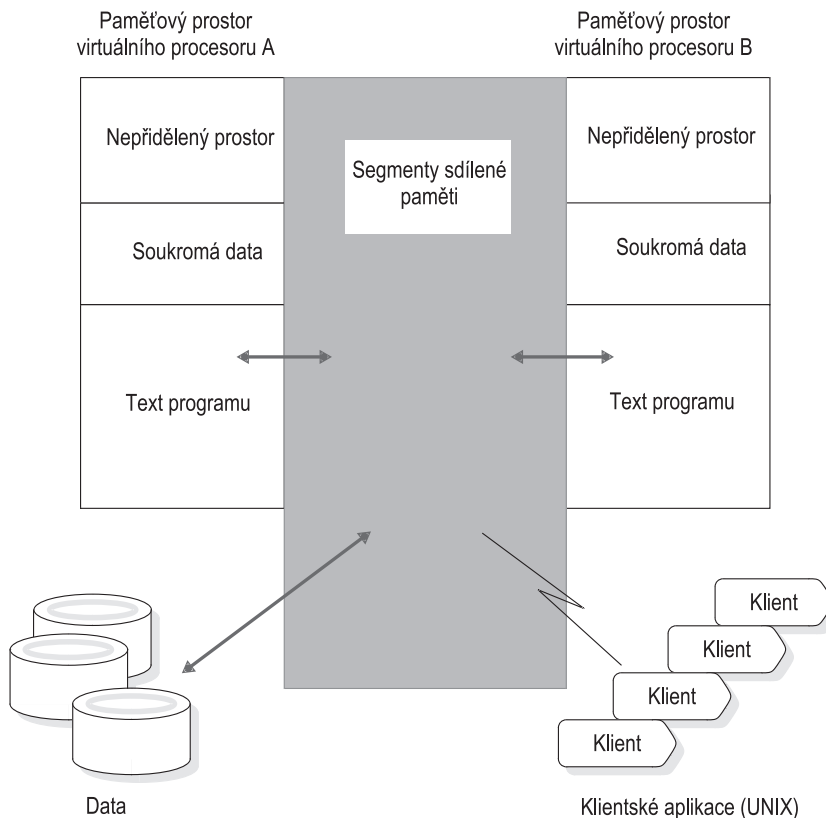
Sdílená paměť představuje nejrychlejší metodu meziprocesové komunikace, protože zapisování a čtení zpráv probíhá rychlostí paměťových přenosů.

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 8-1 znázorňuje schéma sdílené paměti.



Obrázek 8-1. Jak databázový server používá sdílenou paměť

Přidělování sdílené paměti

Databázový server vytváří následující bloky sdílené paměti:

- *Rezidentní část.*
- *Virtuální část.*

Pouze 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.

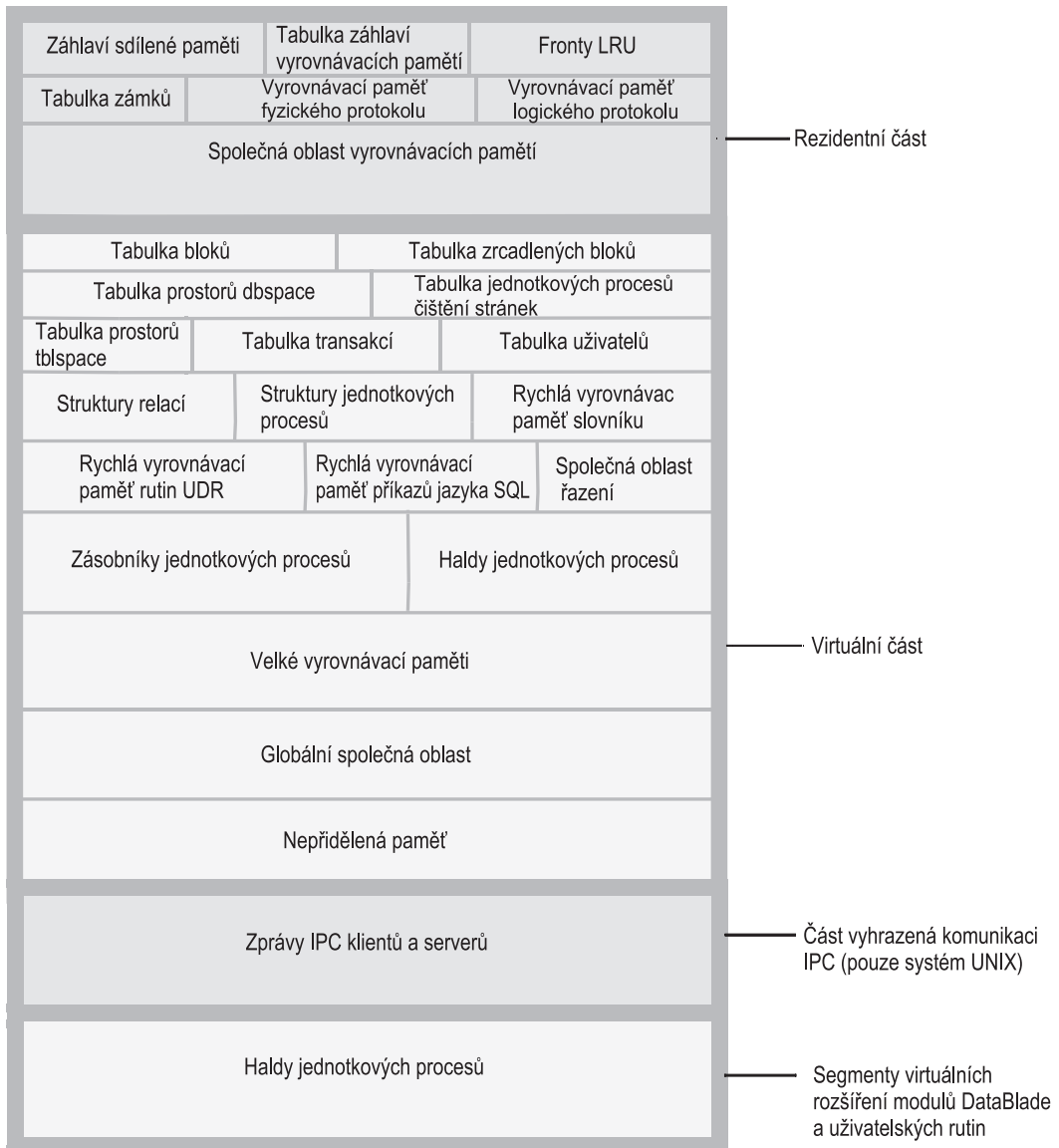
Konec Pouze pro UNIX

- *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 8-2 na stránce 8-6 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, zámků a zámků latch. Virtuální procesory se ke sdílené paměti připojují tehdy, pokud databázový server převedete z režimu offline do klidového režimu, do jednouzivatelského režimu nebo do režimu online. Databázový server používá zámků k řízení souběžného přístupu několika jednotkových procesů ke zdrojům sdílené paměti.



Obrázek 8-2. 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. Informace o způsobu použití obslužného programu **onstat** naleznete v kapitole věnované obslužným programům 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:

Pouze pro UNIX

- Procesy klientských aplikací, které komunikují s databázovým serverem prostřednictvím komunikační části sdílené paměti (**ipcshm**).

Konec Pouze 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 6-26.

Pokud není nastavena proměnná prostředí **INFORMIXSHMBASE**, připojuje se aplikace ke komunikační části paměti na adrese specifické pro platformu. Pokud se klientská aplikace připojuje k jiným segmentům sdílené paměti (nikoliv ke sdílené paměti databázového serveru), může uživatel pomocí proměnné prostředí **INFORMIXSHMBASE** určit adresu, na které se má aplikace připojit ke komunikačním segmentům 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 ke sdílené paměti připojují v průběhu inicializace. 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, spusíte příkaz **onstat -g seg**. Další informace naleznete v částech týkajících se parametru `SHMADD` a společné oblasti vyrovnávacích paměti v příručce *IBM Informix: Dynamic Server Performance Guide*.

Když virtuální procesor žádá operační systém o připojení prvního segmentu sdílené paměti, poskytnete 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: Je doporučeno neměnit 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.

Pouze pro UNIX

- Některé verze systému UNIX vyžadují, aby uživatel jako parametr `SHMBASE` zadal virtuální adresu 0. Adresa 0 sděluje jádru systému UNIX, aby jádro vybralo nejlepší adresu k připojení segmentů sdílené paměti automaticky. Ne všechny architektury systému UNIX však tuto volbu podporují. V některých systémech navíc volba adresy

prováděná jádrem nemusí být optimální.

Konec Pouze 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.

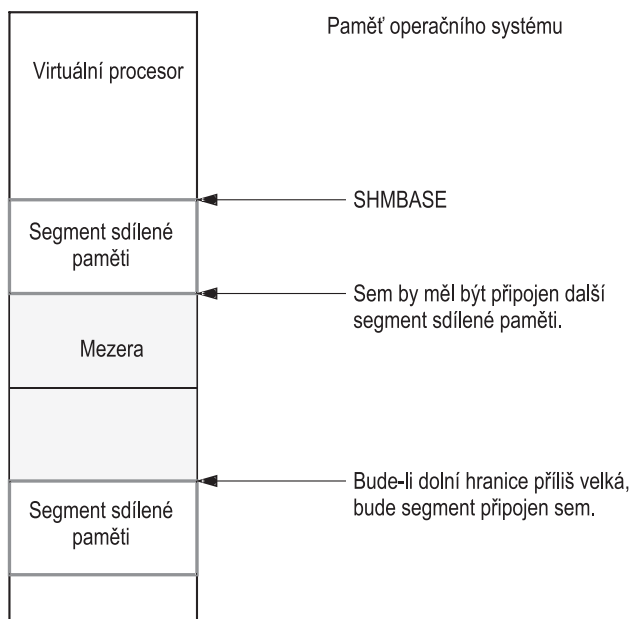
Definování adresy dolní hranice sdílené paměti

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 8-3 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. Popis parametru jádra operačního systému naleznete v části “Adresa spodní hranice sdílené paměti (systém Windows)” na stránce 9-3.



Obrázek 8-3. 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 operací diskového I/O, 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 uzamknete 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 vyrovnávacích pamětí,
- 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 vyrovnávacích pamětí.

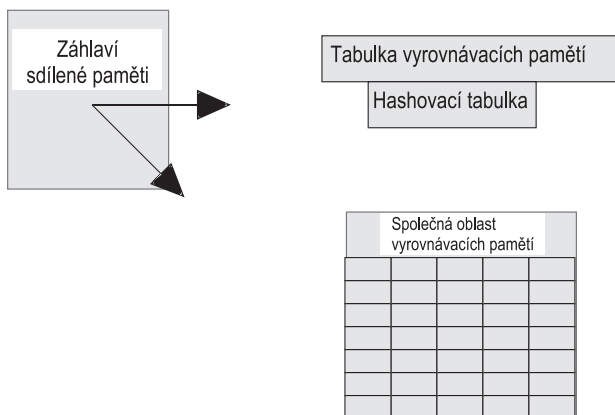
Záhlaví sdílené paměti obsahuje také ukazatele na umístění těchto struktur. Pokud se virtuální procesor poprvé připojuje ke sdílené paměti, načte informace záhlaví sdílené paměti, které ho nasměrují ke všem dalším strukturám.

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 8-4 znázorňuje záhlaví sdílené paměti a společnou oblast vyrovnávacích pamětí.



Obrázek 8-4. 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, přiřadíte 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 *IBM Informix: Dynamic Server Performance Guide*.

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.

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 pamětí. Získávání vyrovnávacích pamětí je řízeno prostřednictvím zámeků latch nazývaných *objekty mutex* a informací o přístupu k zámekům.

Popis činnosti front LRU naleznete v části "Fronty FIFO/LRU" na stránce 8-28. Popis objektů mutex naleznete v části "Objekty mutex" na stránce 6-15.

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 operace I/O 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 8-23 a “Vytvoření vyrovnávacích pamětí stránek blobpage” na stránce 8-39.) Informace o tom, kdy použít soukromé vyrovnávací paměti, naleznete v částech příručky *IBM Informix: Dynamic Server Performance Guide* věnovaných operacím odlehčeného vstupu - výstupu.

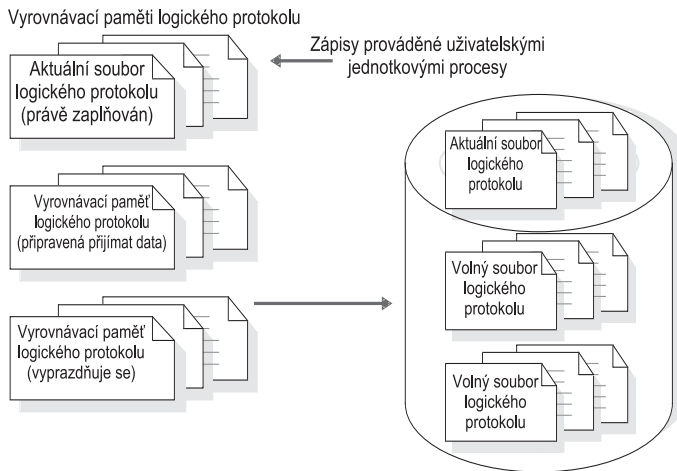
Informace o vyrovnávacích pamětech zobrazuje příkaz **onstat -b**. Informace o příkazu **onstat** naleznete v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Vyrovňovací 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 o úplném kontrolním bodu nebo kontrolním bodu fuzzy.
- 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 8-5.



Obrázek 8-5. 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 8-37.

Velikost vyrovnávacích pamětí logického protokolu určuje parametr LOGBUFF v souboru ONCONFIG. Malé vyrovnávací paměti mohou působit problémy, pokud ukládáte záznamy větší, než je velikost vyrovnávacích pamětí (například data typu TEXT nebo BYTE v prostorech dbspace). Možné hodnoty tohoto konfiguračního 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 8-38.

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

8-35. Informace o monitorování souboru fyzického protokolu naleznete v části “Monitorování aktivity fyzického a logického protokolování” na stránce 17-4.

Velikost vyrovnávacích pamětí fyzického protokolu určuje parametr PHYSBUFF v souboru ONCONFIG. Zápis do vyrovnávací paměti fyzického protokolu zapisuje přesně jednu stránku. Pokud není určená 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. Další informace o tomto konfiguračním parametru naleznete v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Vyrovňovací paměť replikace High-Availability Data Replication

Replikace High-Availability Data Replication (HDR) vyžaduje dvě instance databázového serveru (primární instanci a sekundární instanci), spuštěné ve dvou počítačích. Pokud implementujete v databázovém serveru replikaci HDR, ukládá databázový server záznamy logického protokolu do vyrovnávacích pamětí replikace HDR, než je odešle sekundárnímu databázovému serveru. Velikost vyrovnávací paměti replikace HDR 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 8-15. Informace o tom, jak databázový server používá vyrovnávací paměť replikace HDR, naleznete v části “Jak pracuje replikace HDR” na stránce 20-7.

Tabulka zámků

K vytvoření zámku dojde tehdy, pokud uživatelský jednotkový proces запиše položku do tabulky zámků. Tabulka zámků představuje společnou oblast dostupných zámků. Velikost každého zámku je 44 bajtů. 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).
- Uzamčená stránka a identifikátor rowid.
- Prostor tblspace, který je zamykán.
- 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 zámek určete pomocí konfiguračního parametru LOCKS.

Pokud počet zámek přidělených relacím překročí hodnotu parametru LOCKS, zdvojnásobí databázový server velikost tabulky zámek, a to až patnáctkrát. Maximální hodnota parametru LOCKS je 8000000. Pomocí parametru DEF_TABLE_LOCKMODE nastavte režim zamykání nových tabulek na stránkový nebo řádkový.

Další informace o používání a monitorování zámek naleznete v kapitole o zamykání v příručce *IBM Informix: Dynamic Server Performance Guide* a v příručce *IBM Informix: Guide to SQL Tutorial*. Další informace o použití parametru LOCKS k určení počtu zámek relace naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix: Dynamic Server Administrator's Reference* a v kapitole o vlivech konfigurace na využití paměti v příručce *IBM Informix: Dynamic Server Performance Guide*.

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, rychlé vyrovnávací paměti rutin SPL, rychlé vyrovnávací paměti 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 SHMVIRTFSIZE v souboru ONCONFIG. Velikost segmentů, které budou k virtuální části sdílené paměti přidány později, určete pomocí parametru SHMADD.

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 SHMVIRTFSIZE, SHMADD, DS_TOTAL_MEMORY, DS_TOTAL_SIZE a DS_NONPDQ_QUERY_MEM naleznete v příručkách *IBM Informix: Dynamic Server Performance Guide* a *IBM Informix: Dynamic Server Administrator's Reference*. Další informace také naleznete v části "Přidání segmentu k virtuální části sdílené paměti" na stránce 9-9.

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),
- rychlá vyrovnávací paměť distribuce dat,
- rychlá vyrovnávací paměť slovníku,
- rychlá vyrovnávací paměť rutin SPL,
- rychlá vyrovnávací paměť příkazů 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 tblspace,
- 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 10-6.

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 8-13. Informace o tom, jak monitorovat vyrovnávací paměti, naleznete v části “Monitorování vyrovnávacích pamětí” na stránce 9-11.

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 je povoleno zrcadlení, vytvoří se při inicializaci sdílené paměti 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 zotavení, a příznak, zda je blok částí prostoru blobspace. Informace o monitorování bloků naleznete v části “Monitorování bloků” na stránce 11-41.

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 sbpace, 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 11-41.

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 monitorování aktivity jednotkových procesů čištění stránek naleznete v části věnované příkazu **onstat -F** v kapitole o obslužných programech příručky *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 11-47.

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 *IBM Informix: Dynamic Server Performance Guide* 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 o příkazech jazyka SQL používaných při transakcích naleznete v příručce *IBM Informix: Guide to SQL Tutorial*, v příručce *IBM Informix: Guide to SQL Reference* a v příručce *IBM Informix: Guide to SQL Syntax*.

Pouze 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 Pouze 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 (lehkého 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 operací I/O v příručce *IBM Infix: Dynamic Server Performance Guide*.

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 6-10.

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 6-12. Další informace o způsobu monitorování velikosti zásobníku relace naleznete v části týkající se monitorování relací a jednotkových procesů v příručce *IBM Informix: Dynamic Server Performance Guide*.

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 prostoru v zásobníku, použijte proměnnou prostředí **INFORMIXSTACKSIZE**, nikoliv konfigurační parametr STACKSIZE. Proměnná prostředí **INFORMIXSTACKSIZE** ovlivňuje velikost zásobníku jediného uživatele, proto je méně pravděpodobné, že touto změnou budou ovlivněny nepředpokládané nové klientské aplikace.

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.

Rychlá vyrovnávací paměť 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 disku a uloží statistické údaje do rychlé

vyrovnávací paměti 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 rychlé vyrovnávací paměti distribuce dat, výkon databázového serveru se zvýší. Velikost rychlé vyrovnávací paměti distribuce dat můžete konfigurovat pomocí konfiguračních parametrů DS_HASHSIZE a DS_POOLSIZE. Informace o změně výchozí velikosti rychlé vyrovnávací paměti distribuce dat naleznete v kapitole o dotazech a optimalizátoru dotazů v příručce *IBM Informix: Dynamic Server Performance Guide*.

Rychlá vyrovnávací paměť slovníku

Pokud je v rámci relace spuštěn příkaz jazyka SQL, který požaduje přístup k tabulce systémového katalogu, načte databázový server tabulky systémového katalogu a uloží je ve strukturách, které umožňují účinnější přístup. Tyto struktury se vytvářejí ve virtuální části sdílené paměti a jsou sdílené všemi relacemi. Tyto struktury tvoří rychlou vyrovnávací paměť slovníku.

Velikost rychlé vyrovnávací paměti 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 *IBM Informix: Dynamic Server Performance Guide*.

Rychlá vyrovnávací paměť příkazů SQL

Rychlá vyrovnávací paměť příkazů jazyka SQL omezuje využití paměti a dobu přípravy dotazů. Databázový server používá rychlou vyrovnávací paměť příkazů jazyka SQL k ukládání analyzovaných a optimalizovaných dotazů jazyka SQL spouštěných uživatelem. Pokud uživatel spustí příkaz, který byl již uložen do rychlé vyrovnávací paměti příkazů jazyka SQL, nebude databázový server příkaz analyzovat a optimalizovat znovu, čímž dojde ke zvýšení výkonu.

Další informace naleznete v části “Nastavení parametrů rychlé vyrovnávací paměti příkazů jazyka SQL” na stránce 9-7. Informace o vlivech těchto parametrů na výkon rychlé vyrovnávací paměti příkazů SQL naleznete v příručce *IBM Informix: Dynamic Server Performance Guide*.

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 *IBM Informix: Dynamic Server Performance Guide*.

Rutiny SPL a rychlá vyrovnávací paměť rutin UDR

Databázový server převádí rutiny v jazyku SPL do spustitelného formátu a ukládá je v rychlé vyrovnávací paměti 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 rychlé vyrovnávací paměti rutin UDR.

Velikost rychlé vyrovnávací paměti rutin UDR můžete konfigurovat pomocí konfiguračních parametrů `PC_HASHSIZE` a `PC_POOLSIZE`. Informace o změně výchozí velikosti rychlé vyrovnávací paměti rutin UDR naleznete v kapitole o dotazech a optimalizátoru dotazů v příručce *IBM Informix: Dynamic Server Performance Guide*.

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 **sqlxex** vyzvedávají zprávy z globální společné oblasti a zpracovávají je.

Další informace naleznete v částech o společných oblastech síťových vyrovnávacích paměti a o virtuální části sdílené paměti v příručce *IBM Informix: Dynamic Server Performance Guide*.

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 přidělení při inicializaci 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 není určen parametr **nettype ipcshm**, bude výchozí hodnota očekávaného počtu připojení rovna číslu 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 8-8.

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 a SHMTOTAL podobně jako pro jiné části sdílené paměti platí i pro část virtuálních rozšíření 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 *zámký 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 změny 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 9-10.

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 paměti

Databázový server používá k řízení přístupu k vyrovnávacím pamětem ve sdílené paměti dva typy zámků:

- sdílené zámků,
- výlučné zámků.

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 zámkem), 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 zámkem), 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 rychlé vyrovnávací paměti 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.

K určení informací o společné oblasti vyrovnávacích paměti použijte konfigurační parametr BUFFERPOOL. Parametr určuje i informace o počtu front LRU, které mají být vytvořeny při inicializaci databázového serveru, a hodnoty parametrů **lru_min_dirty** a **lru_max_dirty**, které řídí, jak často mají být vyrovnávací paměti ve sdílené paměti vyprazdňovány na disk.

Poznámka: Informace, které byly dříve určovány pomocí konfiguračních parametrů BUFFERS, LRUS, LRU_MAX_DIRTY a LRU_MIN_DIRTY, jsou ve verzi 10.0 databázového serveru určovány pomocí konfiguračního parametru BUFFERPOOL.

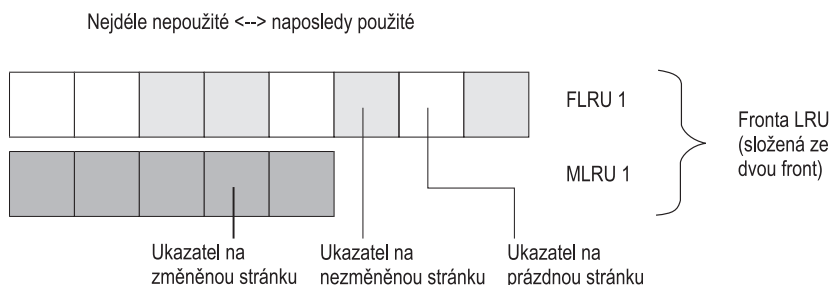
Komponenty front LRU

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 8-6 znázorňuje strukturu front LRU.



Obrázek 8-6. Fronta LRU

Stránky seřazené podle nejdelší doby nepoužití

Pokud databázový server zpracovává požadavky na načtení stránky z disku, musí rozhodnout, kterou stránku v paměti nahradí načítanou stránkou. Databázový server nevybírá 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 nejdelší dobu nepoužita. Databázový server může snadno nalézt nejdéle nepoužité stránky, protože udržuje seznam stránek seřazený sestupně podle doby nepoužívání.

Správa společné oblasti vyrovnávacích 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 paměti ve frontách FLRU je rovnoměrné. Počet vyrovnávacích paměti v každé frontě lze vypočítat vydělením celkového počtu vyrovnávacích paměti počtem front LRU. Počet vyrovnávacích paměti a front LRU je určen konfiguračním parametrem BUFFERPOOL .

Pokud uživatelský jednotkový proces potřebuje získat vyrovnávací paměť, vybere databázový server náhodnou frontu FLRU a použije nejdéle nepoužitou položku v seznamu. Pokud je možné nejdéle 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 vyrovnávacích pamětí” na stránce 9-13.

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 závisejí na počtu jednotek CPU dostupných v počítači. Pokud je počítač vybaven jediným procesorem, nastavte nejprve hodnotu **lrus** v konfiguračním parametru BUFFERPOOL na hodnotu 4. Pokud je počítač vybaven více procesory, použijte k výpočtu následující vzorec:

$$\text{LRUS} = \max(4, (\text{NUMCPUVPS}))$$

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 popsáno v následující části .

Poznámka: Informace, které byly dříve určovány pomocí konfiguračních parametrů BUFFERS, LRUS, LRU_MAX_DIRTY a LRU_MIN_DIRTY, jsou ve verzi 10.0 databázového serveru určovány pomocí konfiguračního parametru BUFFERPOOL.

Počet jednotkových procesů čištění, které je třeba přidělit

Obecně je doporučeno nakonfigurovat jeden jednotkový proces čištění pro každý disk, který je aplikacemi často aktualizován. 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 paměti 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 paměti ve společné oblasti” na stránce 8-34.

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 paměti 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 paměti 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 8-34.

Obrázek 8-7 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: $1000 \times 0.60 = 600$

*Obrázek 8-7. 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 8-8 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: $1000 \times .50 = 500$

*Obrázek 8-8. 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 8-34.

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.

Příklad výstupu příkazu **onstat -p**, který umožňuje monitorovat používání čtení napřed databázovým serverem naleznete v části “Monitorování profilu sdílené paměti a zámek latch” na stránce 9-10 a v kapitole o obslužných programech v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Přístup jednotkových procesů databázového serveru ke stránkám ve vyrovnávací paměti

Databázový server ukládá stránky do vyrovnávací paměti s použitím sdílených zámeků, aby umožnil přistupovat k téže vyrovnávací paměti několika jednotkovým procesům současně. Databázový server používá dvě kategorie zámeků vyrovnávacích pamětí, aby mohl poskytnout tento souběžný přístup a nenarušil přitom izolaci jednotkových procesů. Tyto dvě kategorie přístupu zámeků jsou *sdílený zámek* a *výlučný zámek*. (Další informace naleznete v části “Typy zámeků vyrovnávacích pamětí” na stránce 8-28.)

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 paměti (popisovanou v této části).
- Vyrovnávací paměť fyzického protokolu.

Další informace naleznete v části “Vyprazdňování vyrovnávací paměti fyzického protokolu” na stránce 8-35.

- Vyrovnávací paměť logického protokolu.

Další informace naleznete v části “Vyprazdňování vyrovnávací paměti logického protokolu” na stránce 8-37.

Vyprazdňování vyrovnávacích paměti 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 paměti fyzického protokolu, změněných stránek vyrovnávacích paměti ve sdílené paměti a vyrovnávacích paměti logického protokolu musí být synchronizováno s aktivitou jednotkových procesů čištění stránek podle určitých pravidel, která jsou navržena tak, aby zajistila konzistenci dat.

Vyprazdňování vyrovnávacích paměti ve společné oblasti

K zahájení vyprazdňování vyrovnávacích paměti dochází při splnění jedné z následujících tří podmínek:

- Počet neaktualizovaných vyrovnávacích paměti 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 16-8.)

Přednostní vyprazdňování předobrazů

Nejdůležitější pravidlo vyprazdňování vyrovnávacích paměti stanovuje, že předobrazy změněných stránek musejí být vyprázdněny na disk před vyprázdněním samotných změněných stránek.

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 byl předobraz zapsán do vyrovnávací paměti fyzického protokolu, ale nikoliv do fyzického protokolu na disku, musí být nejprve na disk vyprázdněna vyrovnávací paměť fyzického protokolu a až pak může být na disk vyprázdněna změněná stránka. Tato akce je nezbytná pro funkci rychlé obnovy.

Vyrovnávací paměť fyzického protokolu a fyzický protokol přispívají k udržování fyzické i logické konzistence dat. Informace o fyzickém protokolování, kontrolních bodech a rychlé obnově naleznete v části Kapitola 16, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 16-1.

Následující události způsobují vyprázdnění vyrovnávací paměti fyzického protokolu:

- Došlo k zaplnění aktuální vyrovnávací paměti fyzického protokolu.
- Je zapotřebí vyprázdnit změněnou stránku ve sdílené paměti, ale v aktuální vyrovnávací paměti fyzického protokolu se stále nachází předobraz.
- Dojde k výskytu úplného kontrolního bodu nebo kontrolního bodu typu fuzzy.

Obsah vyrovnávací paměti fyzického protokolu musí být vždy vyprázdněn na disk před vyprázdněním jakékoli datové vyrovnávací paměti. Toto pravidlo je nezbytné pro funkci rychlé obnovy.

Databázový server používá vždy pouze jednu ze dvou vyrovnávacích paměti fyzického protokolu. Tato vyrovnávací paměť je aktuální vyrovnávací paměť fyzického protokolu. Než databázový server začne vyprazdňovat na disk aktuální vyrovnávací paměť fyzického protokolu, nastaví druhou vyrovnávací paměť jako aktuální, aby mohl pokračovat v zapisování i v průběhu vyprazdňování první vyrovnávací paměti.

Synchronizace vyprazdňování vyrovnávacích pamětí

Po inicializaci 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 fronty LRU,
- zápis bloku.

K zobrazení počtu zápisů zaznamenaného databázovým serverem použijte příkaz `onstat -F` popsany v kapitole o obslužných programech příručky *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 vyrovnávacích pamětí” na stránce 9-13.

Zápis na popředí

Kdykoli zapíše vyrovnávací paměť na disk jednotkový proces **sqlxec**, je tento zápis označován jako *zápis na popředí*. K zápisu na popředí dojde, pokud jednotkový proces **sqlxec** prohledává z pověření uživatele fronty LRU, ale nemůže najít žádnou prázdnou nebo nezměněnou stránku. Aby získal prostor, vyprazdňuje jednotkový proces **sqlxec** jednotlivé stránky na disk, aby mohl uložit data načtená z disku. (Další informace naleznete v části “Fronty FIFO/LRU” na stránce 8-28.)

Pokud procesy **sqlxec** 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 **sqlxec**. Databázový server provádí zápisy front LRU jako zápisy na pozadí, ke kterým obvykle dochází, pokud procento neaktualizovaných vyrovnávacích pamětí překročí hodnotu volby `lru_max_dirty` zadanou 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 **sqlxec** 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 **sqlxec**, 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 **sqlxec**. 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 nejučinnějším 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 14, “Logický protokol”, na stránce 14-1.

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.
- Nastane 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 pamětí 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 databázi 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 16-8.

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 sbpace. 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 8-34.

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 nezpřístupňuje 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 blobspace 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 blobspace naleznete v části “Protokolování prostorů blobspace a jednoduché velké objekty” na stránce 14-8.

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

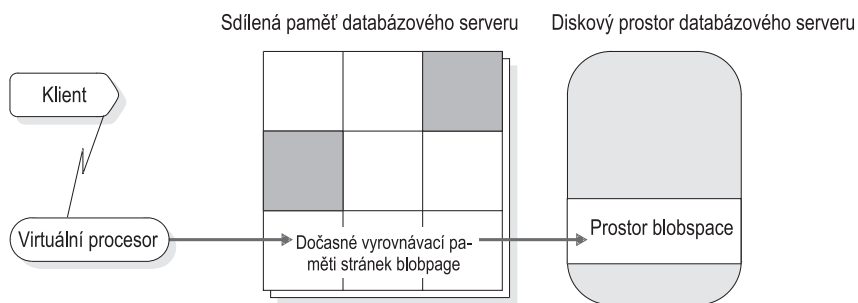
Aby mohl databázový server přijímat od procesu aplikace data jednoduchých velkých objektů, vytváří dvojici vyrovnávacích pamětí prostoru blobspace, jednu ke čtení a jednu k zápisu. Obě paměti mají velikost jedné stránky blobpage prostoru blobspace.

Každý uživatel má k dispozici pouze jednu sadu vyrovnávacích pamětí prostoru blobspace 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 blobspace 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 nezbývají 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 blobspace 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 blobspace 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 blobspace. Obrázek 8-9 znázorňuje proces zápisu jednoduchého velkého objektu do prostoru blobspace.



Obrázek 8-9. Zápis jednoduchého velkého objektu do prostoru blobspace

Stránky blobpage jsou v prostorech blobspace 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 (vložení), 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 paměti 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 paměti, 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 blobspace. 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 10-16.

Protože inteligentní velké objekty procházejí společnou oblastí vyrovnávacích paměti ve sdílené paměti a mohou být protokolovány, je třeba s nimi počítat při přidělování vyrovnávacích paměti. K přidělení vyrovnávacích paměti ve sdílené paměti použijte konfigurační parametr BUFFERPOOL. Obecné pravidlo určuje, že by měl být dostupný dostatek vyrovnávacích paměti 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 paměti pro inteligentní velké objekty naleznete v příručce *IBM Informix: Dynamic Server Performance Guide*.

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 části týkající se obslužného programu **onspaces** v kapitole o obslužných programech 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 *IBM Informix: Dynamic Server Performance Guide*.

Kapitola 9. Správa sdílené paměti

Nastavení konfiguračních parametrů sdílené paměti operačního systému	9-2
Maximální velikost segmentu sdílené paměti	9-2
Použití více než dvou gigabajtů paměti (systém Windows)	9-3
Maximální počet identifikátorů sdílené paměti (systém UNIX)	9-3
Adresa spodní hranice sdílené paměti (systém Windows).	9-3
Semaforey (systém UNIX)	9-4
Nastavení konfiguračních parametrů sdílené paměti databázového serveru	9-4
Nastavení parametrů pro rezidentní sdílenou paměť	9-4
Nastavení parametrů pro virtuální sdílenou paměť	9-5
Nastavení parametrů pro výkon sdílené paměti	9-6
Nastavení parametrů sdílené paměti pomocí textového editoru	9-6
Nastavení parametrů sdílené paměti pomocí programu ISA	9-6
Nastavení parametrů sdílené paměti pomocí programu ON-Monitor (v systému UNIX)	9-6
Nastavení parametrů rychlé vyrovnávací paměti příkazů jazyka SQL	9-7
Opětovná inicializace sdílené paměti.	9-8
Vypnutí nebo zapnutí rezidence rezidentní sdílené paměti	9-8
Zapínání a vypínání rezidence v režimu online	9-8
Zapnutí a vypnutí rezidence při restartování databázového serveru.	9-9
Přidání segmentu k virtuální části sdílené paměti	9-9
Monitorování sdílené paměti	9-9
Monitorování segmentů sdílené paměti	9-9
Monitorování profilu sdílené paměti a zámků latch	9-10
Použití obslužných programů příkazového řádku	9-10
Použití programu IBM Informix Server Administrator	9-10
Použití programu ON-Monitor (v systému UNIX)	9-10
Použití tabulek SMI	9-10
Monitorování vyrovnávacích pamětí	9-11
Použití obslužných programů příkazového řádku	9-11
Použití programu ON-Monitor (v systému UNIX)	9-13
Použití tabulek SMI	9-13
Monitorování aktivity společné oblasti vyrovnávacích pamětí	9-13
Použití obslužných programů příkazového řádku	9-14
Použití tabulek SMI	9-15

Obsah kapitoly

Tato kapitola informuje o tom, jak provádět následující úlohy, které se týkají správy sdílené paměti:

- Nastavení konfiguračních parametrů sdílené paměti.
- Opětovná inicializace 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 *IBM Informix: Dynamic Server Performance Guide*.

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.

Pouze pro UNIX

- Maximální počet identifikátorů semaforů.
- Maximální počet semaforů.
- Maximální počet semaforů na jeden identifikátor.

V systému UNIX soubor Poznámky k počítači obsahuje doporučené hodnoty, které slouží ke konfiguraci zdrojů operačního systému. Při konfiguraci operačního systému použijte těchto doporučených hodnot. 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 Pouze 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), zaokrouhlená na nejbližší násobek 8 kilobajtů.

Databázový server obdrží chybu z 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 dvou gigabajtů paměti (systém Windows)

V systému Windows může databázový server přistupovat k segmentům sdílené paměti větším než dva gigabajty. 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 než dvou gigabajtů:

```
[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 gigabajty, 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 obsluhy 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 8-9.

Maximální množství sdílené paměti, kterou operační systém může přidělit, se vypočítá jako součin počtu identifikátorů sdílené paměti a maximální velikosti segmentu sdílené paměti.

Adresa spodní hranice sdílené paměti (systém Windows)

Výchozí zaváděcí adresa knihoven DLL v systému Windows je 0x10000000. Server Dynamic Server používá jako výchozí adresu počátku sdílené paměti (SHMBASE) adresu 0x0C000000. Sdílená paměť v databázovém serveru se proto může zvětšit až do velikosti 0x10000000, což odpovídá dalším 64 megabajtům.

Jestliže je sdílená paměť při spuštění větší, nebo blízká hodnotě 64 megabajtů, může nastat kolize adres. V tomto případě nastavte vyšší hodnotu parametru SHMBASE v souboru ONCONFIG, například 0x20000000. Chcete-li zjistit velikost sdílené paměti, spusťte příkaz **onstat -g seg** a zobrazte celkovou paměť.

Semafore (systém UNIX)

Činnost databázového serveru vyžaduje jeden semafor systému UNIX pro každý virtuální procesor, jeden pro každého uživatele, který se připojí k databázovému serveru přes sdílenou paměť (protokol **ipeshm**), šest pro obslužné programy databázového programu, a šestnáct pro další 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)

Pouze pro UNIX

- ON-Monitor

Konec Pouze pro UNIX

V systému UNIX může těmito metodami konfigurovat paměť pouze uživatel **root** nebo **informix**. V systému Windows je nutné být členem skupiny Informix Admin.

Nastavení parametrů pro rezidentní sdílenou paměť

Tabulka Tabulka 9-1 obsahuje seznam parametrů v souboru ONCONFIG, které určují konfiguraci společné oblasti vyrovnávacích pamětí 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 9-1. Konfigurace rezidentní části sdílené paměti

Konfigurační parametr	Účel
BUFFERPOOL	Určuje údaje o společné oblasti vyrovnávacích pamětí, které je zapotřebí zadat pro každou jednotlivou velikost stránky, používanou prostorem dbospace.
LOCKS	Určuje počáteční počet zámek databázových objektů, například řádků, hodnot klíčů, stránek a tabulek.
LOGBUFF	Určuje velikost vyrovnávacích pamětí logického protokolu.
PHYSBUFF	Určuje velikost vyrovnávacích pamětí fyzického protokolu.
RESIDENT	Určuje rezidenci rezidentní části sdílené paměti databázového serveru.
SERVERNUM	Určuje jedinečné identifikační číslo databázového serveru v místním hostitelském počítači.
SHMTOTAL	Určuje celkové množství paměti, kterou má používat databázový server.

Nastavení parametrů pro virtuální sdílenou paměť

Tabulka Tabulka 9-2 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 *IBM Informix: Dynamic Server Performance Guide*.

Tabulka 9-2. Konfigurace virtuální části sdílené paměti

Konfigurační parametr	Účel
DS_HASHSIZE	Počet sektorů hashovací tabulky pro seznamy v rychlé vyrovnávací paměti distribuce dat.
DS_POOLSIZE	Maximální počet položek v rychlé vyrovnávací paměti distribuce dat.
PC_HASHSIZE	Určuje počet segmentů hashovací tabulky pro rychlou vyrovnávací paměť rutin UDR a další rychlé vyrovnávací paměti používané databázovým serverem. Další informace o nastavení parametru PC_HASHSIZE naleznete v příručce <i>IBM Informix: Dynamic Server Performance Guide</i> .
PC_POOLSIZE	Určuje počet uživatelských rutin (rutin SPL a externích rutin), které je možno uložit v rychlé vyrovnávací paměti uživatelských rutin. Tento parametr určuje také velikost dalších rychlých vyrovnávacích pamětí databázového serveru, jako je rychlá vyrovnávací paměť názvů typů a rychlá vyrovnávací paměť tříd operátorů. Další informace o nastavení parametru PC_POOLSIZE najdete v příručce <i>IBM Informix: Dynamic Server Performance Guide</i> .
SHMADD	Určuje velikost dynamicky přidávaných segmentů sdílené paměti
SHMTOTAL	Určuje celkové množství paměti, kterou má používat databázový server.
SHMVRTSIZE	Určuje počáteční velikost virtuální části sdílené paměti.
STACKSIZE	Určuje velikost zásobníku uživatelských jednotkových procesů databázového serveru.

Nastavení parametrů pro výkon sdílené paměti

Tabulka Tabulka 9-3 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 9-3. Nastavení voleb výkonu sdílené paměti

Konfigurační parametr	Účel
CKPTINTVL	Určuje maximální počet sekund, které mohou uplynout před tím, než databázový server zkontroluje, zda je nutný kontrolní bod.
CLEANERS	Určuje počet jednotkových procesů čištění stránek, které má databázový server spustit.
RA_PAGES	Určuje počet diskových stránek, které by měl databázový server načíst v předstihu při sekvenčním prohledávání dat nebo indexových záznamů.
RA_THRESHOLD	Určuje počet nezpracovaných stránek paměti, které po načtení způsobí, že databázový server čte z disku v předstihu.

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. Vyhledejte 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 rychlé vyrovnávací paměti 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 a SHMTOTAL mají vliv na rezidentní i virtuální část sdílené paměti.

Chcete-li nastavit konfigurační parametry pro následující volby výkonu sdílené paměti pomocí programu ON-Monitor, vyberte volbu **Parameters > Performance**:

- CKPTINTVL,
- RA_PAGES,
- RA_THRESHOLD.

Nastavení parametrů rychlé vyrovnávací paměti příkazů jazyka SQL

Tabulka 9-4 zobrazuje různé způsoby konfigurace rychlé vyrovnávací paměti příkazů jazyka SQL.

Tabulka 9-4. Konfigurace rychlé vyrovnávací paměti příkazů jazyka SQL

Konfigurační parametr	Účel	Příkaz onmode
STMT_CACHE	Zapne, povolí nebo zakáže rychlou vyrovnávací paměť příkazů jazyka SQL v paměti. Pokud je parametr zapnut, určuje, zda má rychlá vyrovnávací paměť příkazů jazyka SQL uchovávat analyzované a optimalizované příkazy SQL.	onmode -e režim
STMT_CACHE_HITS	Určuje počet použití příkazu (odkazů na příkaz), po kterém bude příkaz zcela vložen do rychlé vyrovnávací paměti příkazů jazyka SQL.	onmode -W STMT_CACHE_HITS
STMT_CACHE_NOLIMIT	Řídí, zda mají být příkazy jazyka SQL vkládány do rychlé vyrovnávací paměti příkazů i po překročení velikosti určené hodnotou parametru STMT_CACHE_SIZE.	onmode -W STMT_CACHE_NOLIMIT
STMT_CACHE_NUMPOOL	Definuje počet společných oblastí rychlé vyrovnávací paměti příkazů jazyka SQL.	Nelze použít.
STMT_CACHE_SIZE	Určuje velikost rychlé vyrovnávací paměti příkazů jazyka SQL.	onmode -W STMT_CACHE_SIZE

Pomocí následujících voleb programu **onstat** lze monitorovat rychlou vyrovnávací paměť příkazů jazyka SQL:

- **onstat -g ssc** (stejně jako **onstat -g cac stmt**)
- **onstat -g ssc all**
- **onstat -g ssc pool**

Další informace o těchto konfiguračních parametrech, volbách **onstat -g** a příkazech **onmode** naleznete v kapitole o obslužných programech v příručce *IBM Informix: Dynamic Server Administrator's Reference*. Další informace o použití rychlé

vyrovnávací paměti 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 *IBM Informix: Dynamic Server Performance Guide*. Podrobnosti o kvalifikačních a identických příkazech najdete v příručce *IBM Informix: Guide to SQL Syntax*.

Opětovná inicializace sdílené paměti

Chcete-li sdílenou paměť znovu inicializovat, 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 “Z kteréhokoliv režimu okamžitě do režimu offline” na stránce 4-15.

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.
- Chcete-li vypnout nebo zapnout rezidenci sdílené paměti při příští opětovné inicializaci sdílené paměti databázového serveru, změňte parametr **RESIDENT** v souboru **ONCONFIG**.

Popis rezidentní části sdílené paměti najdete v části “Rezidentní část sdílené paměti” na stránce 8-13.

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 vrátí zpět do stavu určeného parametrem **RESIDENT** při příští inicializaci sdílené paměti. V systému UNIX může rezidenci vypnout nebo zapnout pouze uživatel **root** nebo **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 `SHMVIRTFSIZE`. Další informace naleznete v kapitole zaměřené na konfigurační parametry 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.

Následující příklad znázorňuje výstup příkazu `onstat -g seg`.

```
Segment Summary:
(resident segments are not locked)
id      key      addr      size      ovhd      class blkused blkfree
300     1381386241 400000    614400    800       R      71      4
301     1381386242 496000    4096000   644       V      322     178
```

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 všech polí, které příkaz **onstat** zobrazuje, najdete v kapitole pojednávající o obslužných programech 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** můžete zobrazit statistiku aktivity databázového serveru a čekajících zámků latch (viz pole **lchwaits**).

Následující příklad znázorňuje tyto statistiky:

```
Profile
dskreads pagreads bufreads %cached dskwrits pagwrits bufwrits %cached
382      400      14438    97.35   381      568      3509    89.14

isamtot  open      start     read      write     rewrite   delete   commit   rollbk
9463     1078     1584     2316     909      162      27      183     1

ovlock   ovuserthread ovbuff    usercpu   syscpu    numckpts  flushes
0        0        0        13.55    13.02    5         18

bufwaits lokwaits lockreqs deadlks  dltouts  ckpwaits  compress  seqscans
14       0        16143   0        0        0         101     68

ixda-RA  idx-RA    da-RA    RA-pgsused lchwaits
5        0        204     148      12
```

Použití programu IBM Informix Server Administrator

Pomocí programu ISA lze získat informace o zámcích latch, zámcích spin a profilech.

Použití programu ON-Monitor (v systému 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 rychlou vyrovnávací pamětí a kolikrát musely jednotkové procesy čekat, než získaly vyrovnávací paměť. Procento zápisů v rychlé vyrovnávací paměti je důležitým měřítkem výkonu. (Informace o tom, jak využít tuto statistiku k vyladění databázového serveru, naleznete v příručce *IBM Informix: Dynamic Server Performance Guide*.) 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**

Příkaz onstat -p: Pomocí příkazu **onstat -p** můžete získat statistiku o zápisech a čteních ukládaných do rychlé vyrovnávací paměti. Následující statistika ukládání do rychlé vyrovnávací paměti se objeví ve čtyřech polích v horním řádku výstupní obrazovky:

- Počet čtení z rychlé vyrovnávací paměti ve sdílené paměti (**bufreads**).
- Procento čtení z rychlé vyrovnávací paměti (**%cached**).
- Počet zápisů do sdílené paměti (**bufwrits**).
- Procento zápisů do rychlé vyrovnávací paměti (**%cached**).
- Informace o obecných stránkách (tj. nestandardních stránkách ve společné oblasti vyrovnávacích pamětí).

Následující příklad ukazuje statistiku čtení a zápisů uložených v rychlé vyrovnávací paměti ve výstupu příkazu **onstat -p**.

```
Profile
dskreads pagreads bufreads %cached dskwrits pagwrits bufwrits %cached
382      400      14438   97.35   381      568      3509   89.14
...
```

Počet čtení a zápisů se může zobrazit jako záporné číslo, pokud počet výskytů překročí 2³² (v závislosti na platformě).

Volba **onstat -p** zobrazí také statistiku (**bufwaits**), která ukazuje, kolikrát musely tyto relace čekat na vyrovnávací paměť.

onstat -B: Provedením příkazu **onstat -B** jsou zobrazeny následující informace o vyrovnávací paměti:

- adresa každé obyčejné vyrovnávací paměti ve sdílené paměti,
- adresa jednotkového procesu, který vyrovnávací paměť právě používá,
- adresa prvního jednotkového procesu, který čeká na každou vyrovnávací paměť,
- informace o definovaných společných oblastech vyrovnávací paměti.

Následující příklad ukazuje výstup příkazu **onstat -B**.

```
Buffers
address  userthread flgs pagenum memaddr  nslots  pgflgs xflgs owner  waitlist
a0bd068  0           806  1:10264 a0ee800 6      52    0     0
a0bd0e8  0           806  3:24099 a0ef000 1       41    0     0
a0bd168  0            6   1:1      a0ef800 88     15   8067c4 0
...
```

```
1 modified, 256 hash buckets, 2048 buffer size
```

onstat -b: Provedením příkazu **onstat -b** jsou pro každou vyrovnávací paměť zobrazeny následující informace:

- adresa každé vyrovnávací paměti, kterou je právě používána jednotkovým procesem,
- čísla 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 definovaných společných oblastech vyrovnávací paměti.

Porovnáním adres uživatelských jednotkových procesů s adresami, které se zobrazí po provedení příkazu **onstat -u**, můžete získat číslo ID relace.

Následující příklad ukazuje výstup příkazu **onstat -b**.

```
Buffers
address  userthread flgs pagenum memaddr  nslots  pgflgs xflgs owner  waitlist
a0f7f28  0           c47  1:14322 a173000 9       11    10    0     0
a0fa2a8  0            47  1:14310 a196800 0        3    10    0     0
a0fac28  0           847  1:14307 a1a0000 74     52    10    0     0
a0fe3a8  0            47  1:14306 a1d7800 0        3    10    0     0
...
a133128  0            c47  1:14328 a525000 2       11    10    0     0
317 modified, 2048 hash buckets, 2048 buffer size
```

Další informace o polích, které zobrazí příkaz **onstat**, najdete v kapitole o obslužných programech 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 vyrovnávacích paměti včetně informací o vyrovnávacích pamětech.

Použití programu ON-Monitor (v systému UNIX)

Chcete-li získat přístup k polím uvedeným na stránce 9-11 pro příkaz **onstat -p** (**bufreads, %cached, bufwrits %cached**), vyberte volbu **Status > Profile**.

Následující příklad ukazuje statistiku čtení a zápisů uložených v rychlé vyrovnávací paměti, 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 rychlé vyrovnávací paměti a celkový počet čekání na rychlá vyrovnávací paměť. Důležité jsou následující řádky.

Řádek	Popis
dskreads	Počet čtení z disku.
bufreads	Počet čtení z vyrovnávací paměti.
dskwrits	Počet zápisů na disk.
bufwrits	Počet zápisů do vyrovnávací paměti.
buffwts	Počet čekání všech jednotkových procesů na vyrovnávací paměť.

Monitorování aktivity společné oblasti vyrovnávacích pamětí

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 *IBM Informix: Dynamic Server Performance Guide*.

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 vyrovnávacích pamětí můžete získat pomocí obslužného programu **onstat**. Volby obslužného programu **onstat** lze také používat z programu ISA.

Další informace o volbách obslužného programu **onstat** najdete v kapitole pojednávající o obslužných programech 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**.

Následující příklad ukazuje některé hodnoty pole **ovbuff** ve výstupu příkazu **onstat -p**:

```
...
ovtbls  ovlock  ovuserthread  ovbuff  usercpu  syscpu  numckpts  flushes
0        0        0             0       13.55    13.02   5         18
...
```

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 8-36.)

Příkaz **onstat -F** zobrazí celkové počty následujících typů zápisu:

- zápis na popředí,
- zápis fronty 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.

Následující příkaz ukazuje výstup příkazu **onstat -F**. Zobrazené údaje informují o tom, kdy a jak jsou vyrovnávací paměti vyprázdněny.

```
...
Fg Writes      LRU Writes      Chunk Writes
0              146             140
```



```

address flusher state data
8067c4 0 I 0 = 0X0
states: Exit Idle Chunk Lru

```

onstat -R: Provedením příkazu **onstat -R** můžete získat informace o počtu vyrovnávacích paměti v každé frontě LRU a počtu a procentu změněných nebo uvolněných vyrovnávacích paměti.

Následující příklad ukazuje výstup příkazu **onstat -R**.

```

8 buffer LRU queue pairs
# f/m length % of pair total
0 f 3 37.5% 8
1 m 5 55.6%
2 f 5 45.5% 11
3 m 6 54.5%
4 f 2 18.2% 11
5 m 9 81.8%
6 f 5 50.0% 10
7 m 5 55.6%
8 F 5 50.0% 10
9 m 5 45.5%
10 f 0 0.0% 10
11 m 10 100.0%
12 f 1 11.1% 9
13 m 8 88.9%
14 f 2 28.6% 7
15 m 5 71.4%
53 dirty, 76 queued, 80 total, 128 hash buckets,
2048 buffer size start clean at 60% (of pair total) dirty,
or 6 buffs dirty, stop at 50%

```

Použití tabulek SMI

Dotazováním na tabulku **sysprofile** můžete získat statistiku typů zápisu, které se nacházejí v následujících řádcích.

Řádek	Popis
fgwrites	Počet zápisů na popředí.
lrurwrites	Počet zápisů front LRU.
chunkwrites	Počet zápisů bloků.

Kapitola 10. Uložení dat

Fyzické a logické paměťové jednotky	10-2
Bloky	10-3
Přidělování disků blokům	10-4
Přístup k disku v systému Windows	10-4
Přístup k disku s vyrovnávací pamětí a bez vyrovnávací pamětí v systému UNIX	10-4
Posuny	10-6
Stránky	10-6
Stránky blobpage	10-7
Stránky sbpage	10-9
Oblasti	10-10
Prostory dbspace	10-11
Řízení umístění ukládaných dat	10-11
Kořenový prostor dbspace	10-13
Dočasné prostory dbspace	10-14
Prostory blobspace	10-15
Prostory sbpace	10-16
Výhody používání prostorů sbpace	10-16
Prostory sbpace a replikace Enterprise Replication	10-16
Metadata, uživatelská data a rezervovaná oblast	10-17
Řízení umístění ukládaných dat	10-17
Paměťové charakteristiky prostorů sbpace	10-19
Velikosti oblastí v prostorech sbpace	10-20
Průměrná velikost inteligentních velkých objektů	10-20
Režim ukládání do vyrovnávací paměti	10-21
Čas posledního přístupu	10-21
Režim uzamykání	10-21
Protokolování	10-22
Úrovně dědičnosti charakteristik prostorů sbpace	10-22
Další informace o prostorech sbpace	10-23
Dočasné prostory sbpace	10-25
Porovnání dočasných a standardních prostorů sbpace	10-25
Dočasné inteligentní velké objekty	10-26
Prostory extspace	10-27
Databáze	10-28
Tabulky	10-29
Typy tabulek serveru Dynamic Server	10-30
Standardní trvalé tabulky	10-31
Tabulky typu RAW	10-31
Tabulky typu TEMP	10-32
Vlastnosti typů tabulek	10-32
Zavedení dat do tabulky	10-32
Rychlá obnova jednotlivých typů tabulek	10-32
Zálohování a obnovení tabulek typu RAW	10-33
Dočasné tabulky	10-33

Uživatelské dočasné tabulky	10-34
Dočasné tabulky vytvářené databázovým serverem	10-35
Prostory tblspace	10-36
Maximální počet prostorů tblspace v tabulce	10-36
Prostory tblspace tabulek a indexů	10-37
Prokládání oblastí	10-38
Fragmentace tabulek a uložení dat	10-38
Objem diskového prostoru potřebný k uložení dat	10-40
Velikost kořenového prostoru dbspace	10-40
Fyzické a logické protokoly	10-41
Dočasné tabulky	10-41
Kritická data	10-42
Další prostor	10-42
Objem prostoru požadovaného databázemi	10-42
Pravidla pro rozvržení disků	10-42
Pravidla pro prostory dbspace a bloky	10-42
Pravidla umisťování tabulek	10-44
Ukázková rozvržení disku	10-44
Ukázkové rozvržení upřednostňující výkon	10-45
Ukázkové rozvržení upřednostňující dostupnost	10-46
Správce logických disků	10-48

Obsah kapitoly

Tato kapitola definuje termíny a vysvětluje koncepty, jejichž znalost je nezbytná k provádění úloh, které popisuje Kapitola 11, “Správa diskového prostoru”, na stránce 11-1. 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:

- bloky,
- stránky,

- oblasti,
- stránky blobpage,
- stránky sbpage.

Databázový server ukládá data do následujících logických jednotek:

- prostory dbspace,
- dočasné prostory dbspace,
- prostory blobspace,
- prostory sbospace,
- dočasné prostory sbospace,
- prostory extspace,
- databáze,
- tabulky,
- prostory tblspace,
- oddíly.

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 10-11).
- Prostory blobspace (viz strana 10-15).
- Prostory sbospace (viz strana 10-16).

Informace o názvech, velikostech a počtech bloků naleznete v části “Určení názvů paměťových prostorů a bloků” na stránce 11-9 a v části “Určení maximální velikosti bloků” na stránce 11-9.

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 18, “Zrcadlení”, na stránce 18-1.

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 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 systému Informix může být v připojeném adresáři systému souborů NFS umístěn pouze tehdy, pokud IBM certifikovala dodavatele tohoto zařízení NFS. Produkty NFS, pomocí kterých můžete připojit paměťový prostor databázového serveru IBM Informix, naleznete v informacích o kompatibilitě produktů na webové stránce produktů IBM Informix.

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 tehdy, 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 11-8.

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 11-7.

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 *znaková speciální zařízení*), použijte databázový server přístup k disku bez vyrovnávací paměti. Při použití diskových zařízení s přímým přístupem dojde ke značnému zvýšení výkonu, protože databázový server má k zařízením přístup pomocí operací přímého vstupu - výstupu. Při použití disku s přímým přístupem jsou také data mezi pamětí databázového serveru a diskem přenášena bez vytváření kopií.

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 znakového speciálního souboru v adresáři **/dev**. V mnoha operačních systémech lze rozlišit znakové speciální soubory a blokové speciální soubory pomocí prvního písmene názvu souboru (typicky *r*). Například soubor **/dev/rsd0f** představuje znakové speciální zařízení, kterému odpovídá blokové speciální zařízení **/dev/sd0f**.

Další informace naleznete v části “Přidělení diskového prostoru s přímým přístupem v systému UNIX” na stránce 11-6.

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 11-5.

Ve výukovém prostředí, kde není výkon důležitý, nebo k uložení statických dat mohou být předpřipravené soubory snáze použitelné. Pokud v systému UNIX musíte používat předpřipravené soubory, uložte do těchto souborů taková data, ke kterým se přistupuje nejméně často. 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 je v systému UNIX doporučeno k dosažení vyššího výkonu používat přímá disková zařízení, mohou díky nedávným pokrokům v oblasti rychlých vyrovnávacích pamětí předpřipravené soubory poskytovat podobný nebo i vyšší výkon. Pokud chcete určit, jak dosáhnout co nejlepšího výkonu zařízení, změřte výkon systému s použitím obou typů zařízení pro prostory dbspace a rozvržení tabulek.

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 rozdělit na menší bloky pomocí *posunů*. Další informace naleznete v části “Pravidla pro rozvržení disků” na stránce 10-42.

Tip: Protože je velikost bloku omezena na 4 TB, můžete se vyhnout dělení disku na oddíly a přidělit celou diskovou jednotku jedinému bloku.

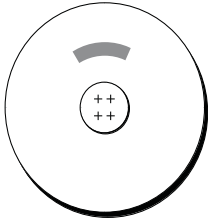
Posun umožňuje určit umístění daného bloku v diskovém oddílu, souboru nebo zařízení. Předpokládejme například, že vytváříte blok o velikosti 1000 KB, který chcete rozdělit do dvou bloků po 500 KB. Počátek prvního bloku můžete označit pomocí posunu s hodnotou 0 KB a počátek druhého bloku pomocí posunu 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 11-6 popisuje, kdy a jak je třeba určit posun.

Stránky

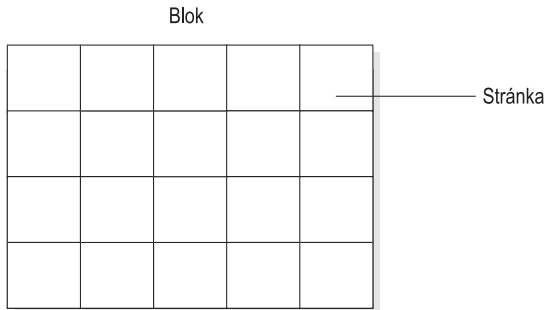
Stránka je fyzická jednotka diskové paměti, kterou databázový server používá ke čtení a zápisu databází systému Informix. Obrázek 10-1 znázorňuje koncept stránky, zde reprezentované tmavým sektorem na disku.



Obrázek 10-1. 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 10-2. Stránka je vždy celá obsažena v jediném bloku, nemůže tedy překračovat hranici dvou bloků.



Obrázek 10-2. Blok logicky dělený do série stránek

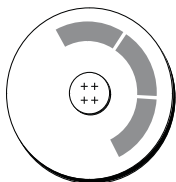
Informace o způsobu, jakým databázový server strukturuje data v rámci stránky, naleznete v kapitole o diskových strukturách a paměti v příručce *IBM Informix: Dynamic Server Administrator's Reference*

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 10-15.

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

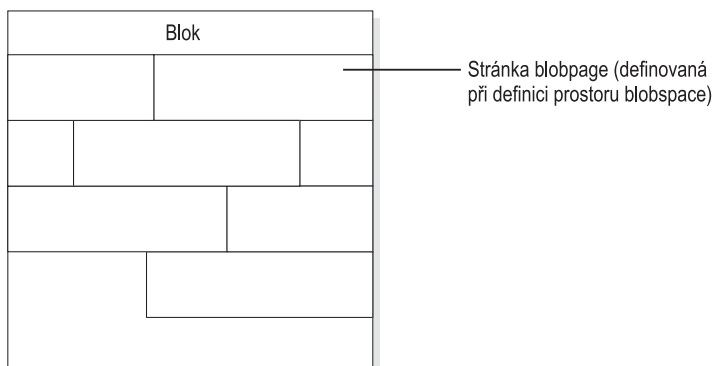
nejlépe odpovídá velikosti ukládaných dat. Obrázek 10-3 znázorňuje koncept stránky blobpage reprezentované několika (třemi) datovými stránkami.



Obrázek 10-3. 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 blobspace 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 blobspace, jak znázorňuje Obrázek 10-4. Stránka blobpage je vždy celá obsažena v jediném bloku a nemůže překračovat hranici dvou bloků.



Obrázek 10-4. Blok v prostoru blobspace logicky dělený do série stránek blobpage

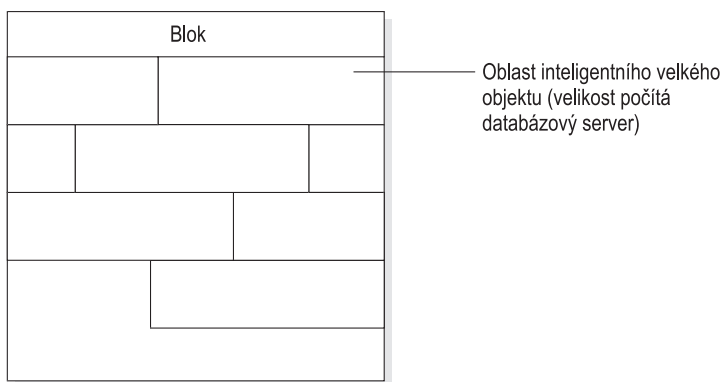
Data jednoduchých velkých objektů nemusíte ukládat v prostoru blobspace, můžete je ukládat i v prostoru dbspace. V případě jednoduchých velkých objektů větších než dvě stránky však bude výkon vyšší, pokud je budete ukládat v prostoru blobspace. Jednoduché velké objekty uložené v prostoru dbspace mohou sdílet jedinou stránku, jednoduché velké objekty uložené v prostoru blobspace stránky nesdílejí.

Informace o tom, jak určit velikost stránky blobpage, naleznete v části “Určení velikosti stránky blobpage” na stránce 11-26.

Stránky sbpage

Stránka sbpage je typ stránky, do které databázový server ukládá v prostoru sbspace inteligentní velké objekty. Popis prostorů sbspace naleznete v části “Prostory sbspace” na stránce 10-16. Na rozdíl od stránek blobpage nelze stránky sbpage konfigurovat. Stránka sbpage 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 sbspace je oblast, zatímco jednotkou přidělování v prostoru blobspace je stránka blobpage. Podobně jako stránky vytvářejí blok, i určitý počet oblastí inteligentních velkých objektů tvoří blok prostoru sbspace, jak znázorňuje Obrázek 10-5. Oblast je vždy celá obsažena v jediném bloku a nemůže překračovat hranici dvou bloků.



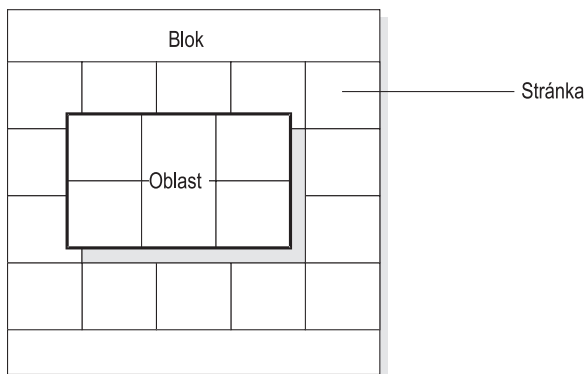
Obrázek 10-5. Blok v prostoru sbspace logicky dělený do série stránek oblastí

Inteligentní velké objekty nelze ukládat do prostorů dbspace a blobspace. Další informace naleznete v části “Prostory sbspace” na stránce 10-16 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 10-20.

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 10-6 znázorňuje koncept oblasti.



Obrázek 10-6. Oblast, která se skládá ze šesti sousedících stránek na přímém diskovém zařízení

Oblast tvoří kolekce sousedících stránek, do kterých jsou ukládána data konkrétní tabulky. (Viz “Tabulky” na stránce 10-29.) 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 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.

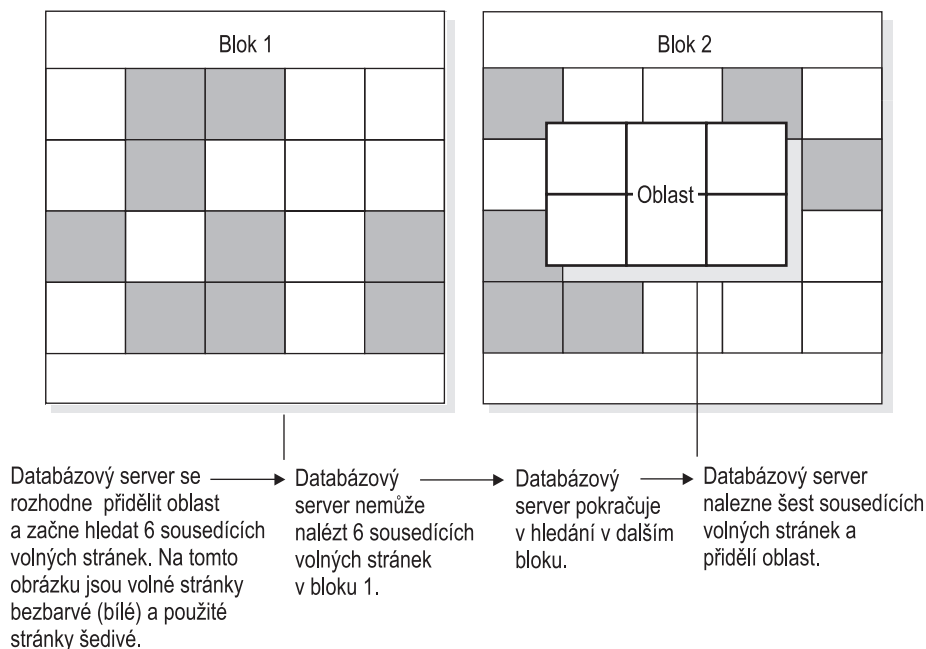
Velikost počáteční oblasti a dalších oblastí určete pomocí příkazů `CREATE TABLE` a `ALTER TABLE`. Další informace naleznete v příručce *IBM Informix: Guide to SQL Syntax* a v části o diskových strukturách v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Pokud vytváříte tabulku se sloupcem datového typu CLOB nebo BLOB, definujete také oblasti prostoru `space`. Další informace naleznete v příručce “Paměťové charakteristiky prostorů `space`” na stránce 10-19.

Obrázek 10-7 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 10-7. Proces přidělování oblasti

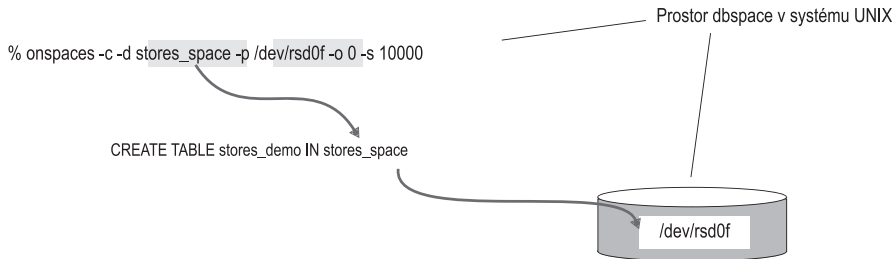
Prostory dbspace

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.

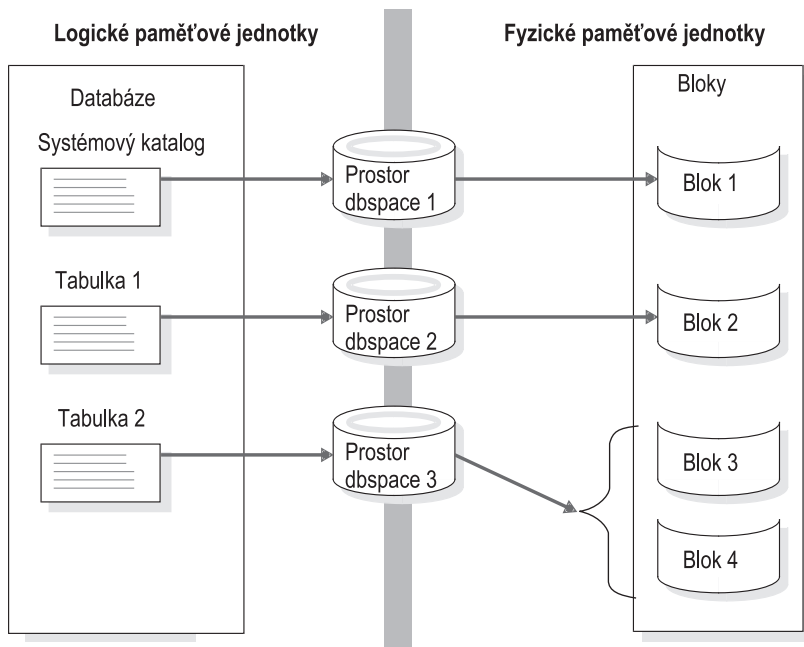
Jak znázorňuje Obrázek 10-8, můžete k řízení umístění databázi a tabulek používat volbu `IN prostor_dbpace` příkazů `CREATE DATABASE` a `CREATE TABLE`. (Další informace naleznete v části “Tabulky” na stránce 10-29.)



Obrázek 10-8. 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 11-10.

Prostor dbspace obsahuje jeden nebo více bloků, jak znázorňuje Obrázek 10-9. 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 11-41.) Pokud prostor dbspace obsahuje více než jeden blok, nelze určit blok, do kterého budou data uložena.



Obrázek 10-9. Prostory dbspace, které propojují logické a fyzické paměťové jednotky

Databázový server používá prostory dbSPACE k ukládání databází a tabulek. (Viz “Tabulky” na stránce 10-29.)

Pokud vytváříte standardní nebo dočasný prostor dbSPACE, můžete určit velikost stránky tohoto prostoru. Velikost stránky nemůžete určit prostorům blobSPACE, prostorům sbSPACE 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 11-14.

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 11-12.

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 11-41.

V části Kapitola 11, “Správa diskového prostoru”, na stránce 11-1 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 sbSPACE.

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 10-29 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 10-40 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 vypuštění 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 logického zotavení, což zvyšuje výkon serveru v kritickém okamžiku zotavování 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 vkládány 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 11-19.

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 10-29).

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 téhož důvodu nezapíše 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. K tomuto prostoru 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 11-47. 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 11, “Správa diskového prostoru”, na stránce 11-1.

Informace o struktuře prostoru blobSPACE naleznete v kapitole o diskových strukturách a paměti v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Prostory sbpace

Prostor sbpace 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 sbpace 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.

Ke čtení a zapisování inteligentních velkých objektů a k vyhledávání v těchto objektech mohou programátoři používat funkce podobné funkcím systémů UNIX a Windows. Server Dynamic Server poskytuje toto rozhraní inteligentních velkých objektů v rozhraní API modulů DataBlade a v programovacím rozhraní ESQ/C.

- Tyto objekty lze zotavit.

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 zámku respektuje pravidla pro úroveň izolace transakcí. Další informace o zamykání a souběžném zpracování naleznete v příručce *IBM Informix: Dynamic Server Performance Guide*.

- 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 ESQ/C. Další informace o funkcích rozhraní DataBlade API naleznete v příručce *IBM Informix: DataBlade API Function Reference*. Další informace o funkcích jazyka ESQ/C naleznete v příručce *IBM Informix: ESQ/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 sbspace přiřazen tomuto prostoru jeden nebo více bloků. První blok prostoru sbspace se však vždy skládá ze tří částí:

- *Oblast metadat.*

Metadata určují klíčové aspekty prostoru sbspace a všech inteligentních velkých objektů uložených v prostoru sbspace a umožňují databázovému serveru manipulovat s inteligentními velkými objekty uloženými v tomto prostoru a provádět jejich zotavení.

- *Oblast uživatelských dat.*

Uživatelská data jsou data inteligentních velkých objektů ukládaná do prostoru sbspace 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.

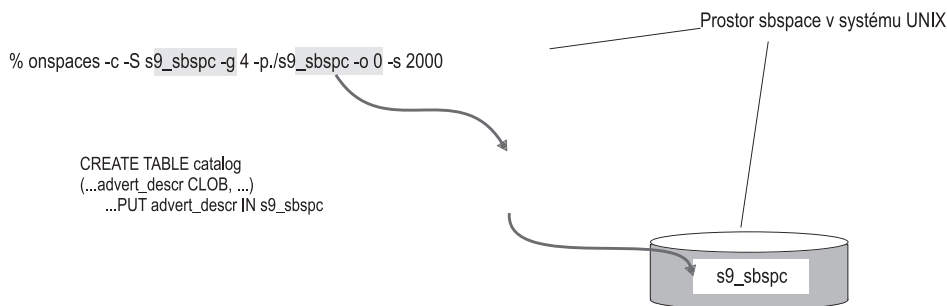
Informace o tom, jak správně přidělit oblasti metadat a uživatelských dat v prostorech sbspace naleznete v části “Určení velikosti oblasti metadat prostoru sbspace” na stránce 11-29 a v příručce *IBM Informix: Dynamic Server Performance Guide*.

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. K 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 části Kapitola 11, “Správa diskového prostoru”, na stránce 11-1.

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 10-10, můžete umístění inteligentních velkých objektů řídit pomocí volby `IN prostor_sbspace` v klauzuli `PUT` příkazu `CREATE TABLE`.



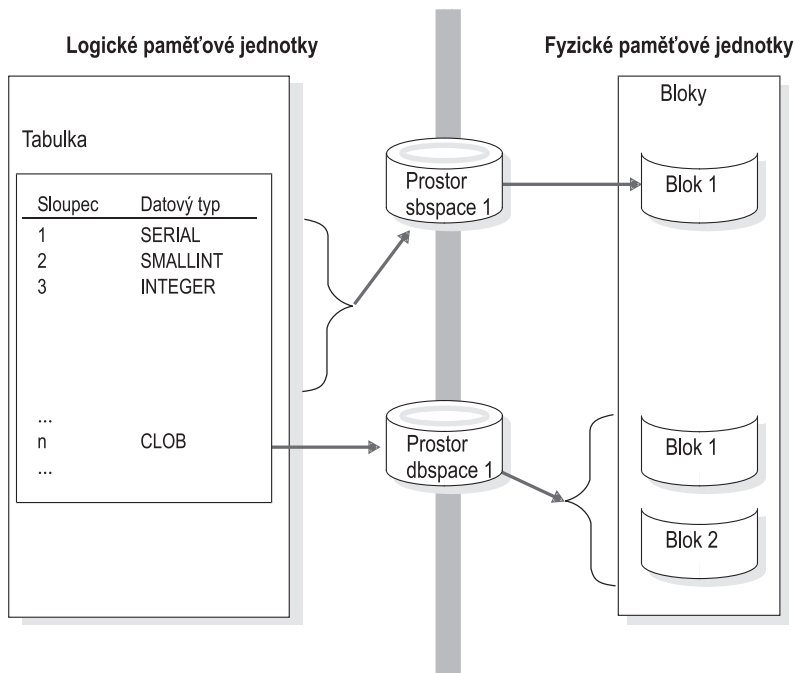
Obrázek 10-10. Řízení umístění inteligentních velkých objektů

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 dbspace nebo blobspace” na stránce 11-20. 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 10-11. Pokud prostor sbspace obsahuje více než jeden blok, nelze určit blok, do kterého budou data uložena.

K 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 *IBM Informix: Dynamic Server Performance Guide*.



Obrázek 10-11. 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 zotavení v případě selhání média. Další informace naleznete v části “Zrcadlení” na stránce 18-1.

Informace o použití příkazu **onspaces** k provádění následujících úloh uvádí Kapitola 11, “Správa diskového prostoru”, na stránce 11-1.

- 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 jazyka 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 *IBM Informix: Dynamic Server Performance Guide*.

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 oblastí metadat, kterou je třeba přidělit, pomocí hodnoty parametru **AVG_LO_SIZE**. Další informace naleznete v části "Určení velikosti oblasti metadat prostoru sbspace" na stránce 11-29.

Režim ukládání do 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 neurčujte 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 *IBM Informix: Dynamic Server Performance Guide*.

Čas posledního přístupu

Při vytváření prostoru sbspace můžete pomocí parametru ACCESSTIME 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 “ACCESSTIME=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 uzamykání

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 zámčích v příručce *IBM Informix: Dynamic Server Performance Guide*.

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 14-9.

Důležité: Pokud používáte protokolující databáze, protokolování prostorů sbspace zapněte. Pokud dojde k selhání vyžadujícímu zotavení protokolu, budou inteligentní velké objekty po zotavení konzistentní se zbytkem databáze.

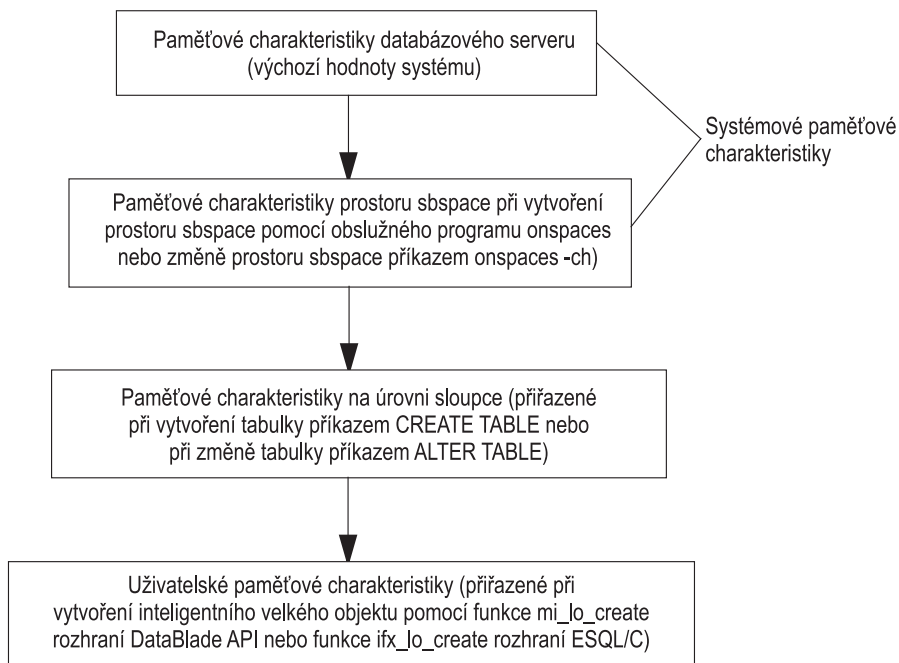
Stav protokolování určete pomocí parametru LOGGING volby příkazu **onspaces -c -Df**. Výchozí nastavení je “LOGGING=off”. Stav protokolování můžete změnit pomocí volby **onspaces -ch -Df**. Stav protokolování můžete také potlačit pomocí klauzule PUT příkazů CREATE TABLE a ALTER TABLE jazyka SQL. Další 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 jazyka 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 8-41.

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 10-12 znázorňuje hierarchii charakteristik inteligentního velkého objektu.



Obrázek 10-12. Hierarchie paměťových charakteristik

Obrázek 10-12 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 10-1 uvádí zdroje informací o různých úlohách, které se týkají používání a správy prostorů sbspace.

Tabulka 10-1. Nalezení informací o úlohách, týkajících se prostorů sbpace

Úkol	Odkaz
Nastavení konfiguračních parametrů inteligentních velkých objektů	Kapitola 9, “Správa sdílené paměti”, na stránce 9-1
Porozumění stránkám sbpace	“Stránky sbpace” na stránce 10-9
Určení charakteristik vstupu - výstupu prostoru sbpace	Volba příkazu onspaces v části “Paměťové charakteristiky prostorů sbpace” na stránce 10-19
Přidělení prostoru prostorům sbpace.	“Vytvoření prostoru sbpace” na stránce 11-28
Přidání bloku k prostoru sbpace.	“Přidání bloku do prostoru sbpace” na stránce 11-29
Definování nebo změna paměťových charakteristik inteligentních velkých objektů.	“Změna paměťových charakteristik inteligentních velkých objektů” na stránce 11-30 Klauzule PUT příkazů CREATE TABLE a ALTER TABLE v příručce <i>IBM Informix: Guide to SQL Syntax</i>
Monitorování prostorů sbpace.	“Monitorování prostorů sbpace” na stránce 11-51 Kapitola o posouzení výkonu tabulek v příručce <i>IBM Informix: Dynamic Server Performance Guide</i>
Nastavení protokolování prostoru sbpace.	“Protokolování prostorů sbpace a inteligentních velkých objektů” na stránce 14-9
Zálohování prostoru sbpace.	“Zálohování prostorů sbpace” na stránce 15-4
Kontrola konzistence prostoru sbpace.	“Ověření metadat” na stránce 22-4
Porozumění struktuře prostoru sbpace.	Kapitola o diskových strukturách v příručce <i>IBM Informix: Administrator's Reference</i>
Použití onspaces 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> <i>IBM Informix: ESQL/C Programmer's Manual</i>
Výpočet velikosti oblasti metadat. zlepšení vstupu - výstupu metadat	Kapitola o výkonu tabulek v příručce <i>IBM Informix: Dynamic Server Performance Guide</i>
Změna paměťových charakteristik	
Porozumění zámekům inteligentních velkých objektů	Kapitola o zamykání v příručce <i>IBM Informix: Dynamic Server Performance Guide</i>
Konfigurace prostorů sbpace pro dočasné velké objekty	Kapitola o vlivu konfigurace na aktivitu vstupu - výstupu v příručce <i>IBM Informix: Dynamic Server Performance Guide</i> .

Dočasné prostory sbpace

Dočasné prostory sbpace používejte k ukládání inteligentních velkých objektů bez protokolování metadat a uživatelských dat. Pokud ukládáte dočasné inteligentní velké objekty do standardního prostoru sbpace, dochází k protokolování metadat. Dočasné prostory sbpace se podobají dočasným prostorům dbpace. K vytvoření dočasného prostoru sbpace použijte příkaz **onspaces -c -S** s volbou **-t**. Další informace naleznete v části “Vytvoření dočasného prostoru sbpace” na stránce 11-31.

Dočasné velké objekty můžete ukládat do standardního prostoru sbpace nebo do dočasného prostoru sbpace.

- Pokud určíte dočasný prostor sbpace v konfiguračním parametru SBSPACETEMP, můžete do něj ukládat dočasné inteligentní velké objekty.
- Pokud určíte standardní prostor sbpace v parametru SBSPACENAME, můžete do něj ukládat trvalé a dočasné inteligentní velké objekty.
- Pokud určíte název dočasného prostoru sbpace v příkazu CREATE TEMP TABLE, můžete do něj ukládat dočasné inteligentní velké objekty.
- Pokud určíte název trvalého prostoru sbpace 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 sbpace, nebudete do tohoto prostoru sbpace moci ukládat *trvalé* inteligentní velké objekty. Bude však možné ukládat do tohoto prostoru dočasné inteligentní velké objekty.

Porovnání dočasných a standardních prostorů sbpace

Tabulka 10-2 porovnává standardní a dočasné prostory sbpace.

Tabulka 10-2. Dočasné a standardní prostory sbpace

Charakteristiky	Standardní prostor sbpace	Dočasný prostor sbpace
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. Vytváření a odstraňování prostoru a přidávání bloků je protokolováno.
Podporuje rychlou obnovu?	Ano	Ne (prostor sbpace je vyprázdněn při restartování databázového serveru) Pokud chcete inicializovat sdílenou paměť a nechcete odstranit dočasné inteligentní velké objekty, použijte příkaz oninit -p . Pokud dočasné velké objekty zachováte, budou se nacházet v neurčitěm stavu.
Podporuje zálohování a obnovení?	Ano	Ne
Lze přidávat a vypouštět bloky?	Ano	Ano
Príslušný 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 10-3 porovnává standardní a dočasné inteligentní velké objekty.

Tabulka 10-3. 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
Podporuje odvolávání?	Ano	Ne
Provádí 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*.

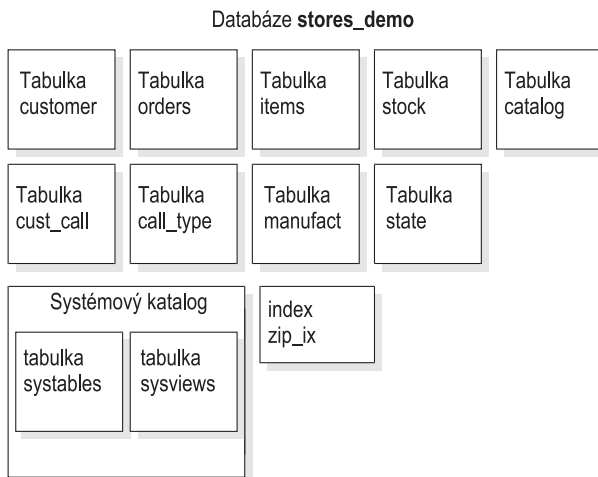
Databáze

Databáze je logická paměťová jednotka, která obsahuje tabulky a indexy. (Další informace naleznete v části “Tabulky” na stránce 10-29.) 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 10-13 znázorňuje tabulky, které jsou součástí databáze stores_demo.



Obrázek 10-13. Databáze stores_demo

Omezení velikosti, která platí pro databáze, souvisí s jejich umístěním v prostorech dbspace. Pokud chcete zaručit, že všechny tabulky konkrétní databáze budou vytvořeny na konkrétním fyzickém zařízení, přiřaďte tomuto zařízení jediný blok a vytvořte 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 11-40.

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 10-10.) 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.

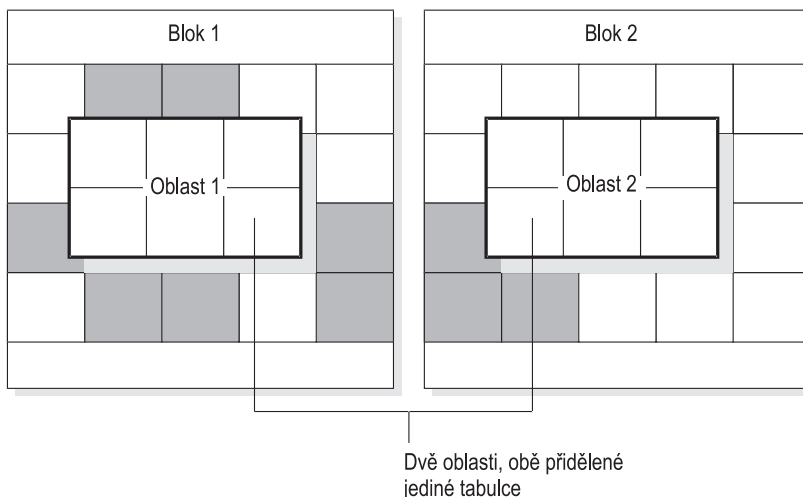
Dále můžete:

- 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 *IBM Informix: Dynamic Server Performance Guide*.

Tabulka složená z oblastí se může rozprostírat přes několik bloků, jak znázorňuje Obrázek 10-14.



Obrázek 10-14. 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 10-42 a v příručce *IBM Informix: Dynamic Server Performance Guide*.

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 10-4 uvádí typy tabulek dostupných v serveru Dynamic Server. Hodnoty příznaku jsou hexadecimální hodnoty každého typu tabulky ve sloupci **flags** tabulky **systables**.

Tabulka 10-4. Typy tabulek serveru Dynamic Server

Charakteristika	STANDARD	RAW	TEMP
Je tabulka trvalá?	Ano	Ano	Ne
Je tabulka protokolovaná?	Ano	Ne	Ano
Podporuje indexy?	Ano	Ne	Ano
Podporuje odvolání transakce?	Ano	Ne	Ano
Podporuje zotavení?	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	Bez příznaků.	0x10	Bez příznaků.

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 zotavení 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 typu RAW nepodporují indexy, referenční omezení ani 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 (zaváděč 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 typu RAW 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 selhání serveru. Dočasné tabulky podporují indexy, omezení a odvolávání transakcí. Dočasné tabulky nelze zotavovat, 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 *IBM Informix: Dynamic Server Performance Guide*.

Další informace naleznete v části “Dočasné tabulky” na stránce 10-33.

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 *IBM Informix: Dynamic Server Performance Guide*. 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

Tabulka 10-5 znázorňuje scénáře rychlé obnovy pro jednotlivé typy tabulek dostupné v serveru Dynamic Server.

Tabulka 10-5. 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.
RAW	Pokud byl od poslední změny tabulky typu RAW dokončen kontrolní bod, bude možné zotavit všechna data. Budou ztracena veškerá vložená data, aktualizace dat a odstranění dat, ke kterým došlo po posledním kontrolním bodu. Nedokončené transakce v tabulce typu RAW nebudou odvolány.

Zálohování a obnovení tabulek typu RAW

Tabulka 10-6 popisuje scénáře zálohování pro jednotlivé typy tabulek dostupné v serveru Dynamic Server.

Tabulka 10-6. Zálohování tabulek serveru Dynamic Server

Typ tabulky	Je zálohování povoleno?
STANDARD	Ano.
TEMP	Ne.
RAW	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.

Tabulka 10-7 znázorňuje scénáře obnovení pro tyto typy tabulek.

Tabulka 10-7. Obnovení tabulek serveru Dynamic Server

Typ tabulky	Je obnovení povoleno?
STANDARD	Ano. Lze použít teplé obnovení, studené obnovení a obnovení k bodu v čase.
TEMP	Ne.
RAW	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.

Přesvědčte 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 dočasné tabulky, 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 dočasné tabulky, 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 dočasné tabulky. 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 dočasné tabulky 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, spusťte 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 11-19.

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. (Viz “Tabulky” na stránce 10-29.)

Maximální počet prostorů tblspace v tabulce

Pro tabulku můžete určit nejvýše 2**20 (nebo 1048576) prostorů tblspace.

Prostory tblspace tabulek a indexů

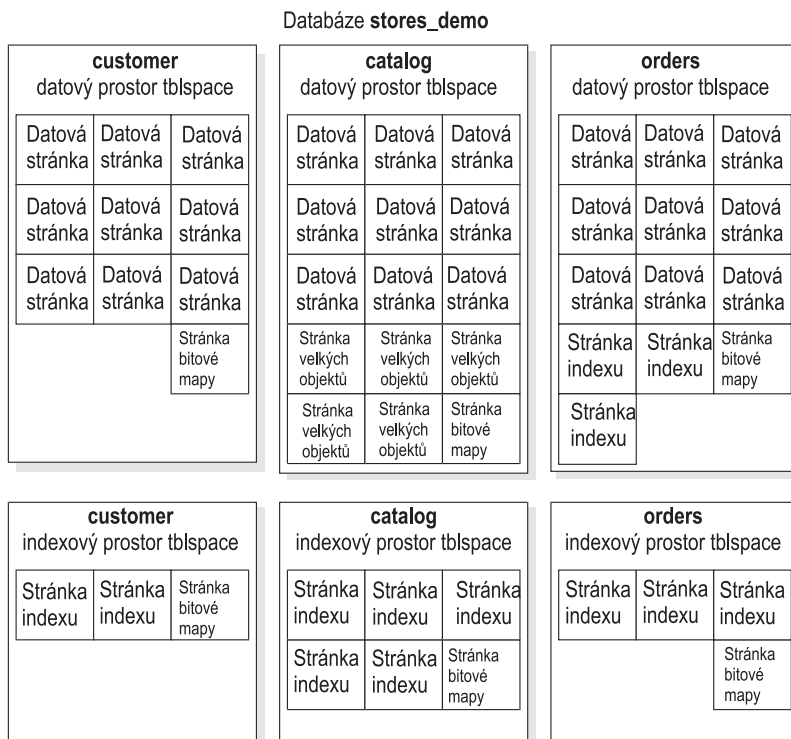
Prostor tblspace tabulky obsahuje následující typy stránek:

- 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 10-15 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.

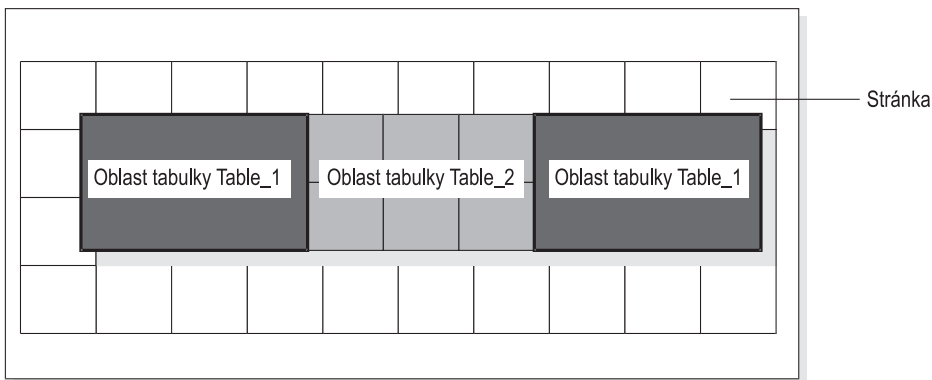


Obrázek 10-15. Ukázkové prostory tblspace v databázi *stores_demo*

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 10-16 znázorňuje tuto situaci v podobě dvou oblastí, které spolu nesousedí a tvoří prostor tblspace tabulky **table_1**, a třetí oblasti, která patří do prostoru tblspace tabulky **table_2**. Oblast tabulky **table_2** se nachází mezi první oblastí tabulky **table_1** 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 *IBM Informix: Dynamic Server Performance Guide*.



Obrázek 10-16. 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.

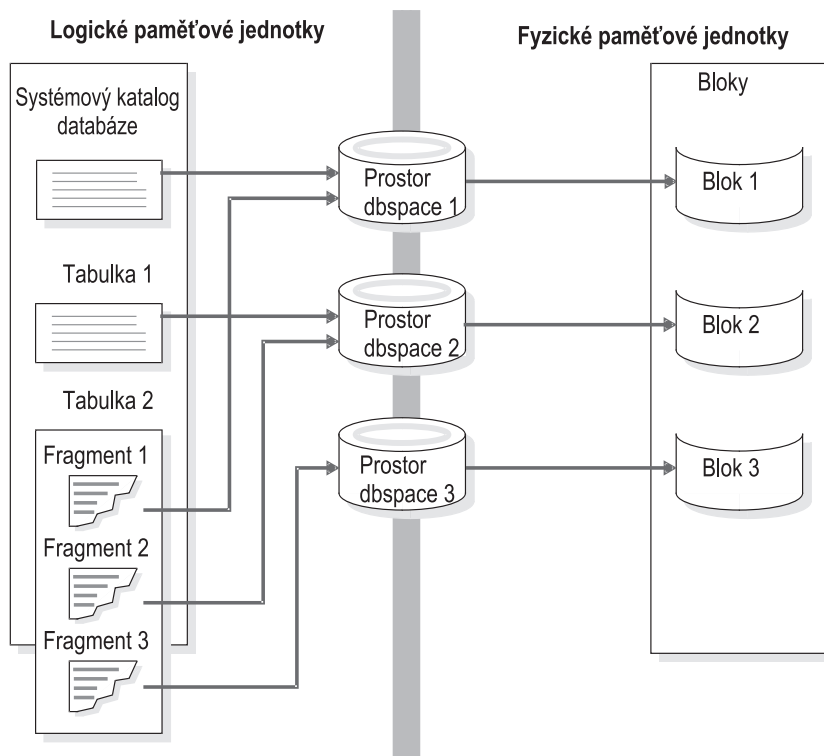
Pokud fragmentujete tabulku, můžete také vytvořit několik oddílů tabulky v jediném prostoru dbspace, jak znázorňuje následující příklad:

```

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 10-17 znázorňuje roli fragmentů při určování umístění dat.



Obrázek 10-17. Prostory dbspace, které propojují logické paměťové jednotky (včetně fragmentů tabulek) a fyzické paměťové jednotky

Informace o prostorech a oddílech uvádí Kapitola 11, “Správa diskového prostoru”, na stránce 11-1.

Další informace o fragmentaci naleznete v příručce *IBM Informix: Database Design and Implementation Guide* a v příručce *IBM Informix: Dynamic Server Performance Guide*.

Objem diskového prostoru potřebný k uložení dat

K určení potřebného diskového prostoru proveďte následující kroky:

Postup určení potřebného diskového prostoru:

1. Vypočítejte 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 veliký 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 vypusíte.

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 15-16 a v části “Změna umístění a velikosti fyzického protokolu” na stránce 17-1.

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. Pokyny ke správnému určení velikosti logického protokolu naleznete v části “Velikost a umístění fyzického protokolu” na stránce 16-4.

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 15-2.

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 velikostí 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 10-33.

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 spouštěná 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 *IBM Informix: Dynamic Server Performance Guide*.

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 *IBM Informix: Dynamic Server Performance Guide*.

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. Podrobnosti naleznete v části “Přidělení diskového prostoru s přímým přístupem v systému UNIX” na stránce 11-6.

Tip: 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 umísťte 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 16-4 a v části “Umístění souborů logického protokolu” na stránce 14-2. Další informace naleznete také v části “Přesun souboru logického protokolu do jiného prostoru dbspace” na stránce 15-16 a v části “Změna umístění a velikosti fyzického protokolu” na stránce 17-1.

- Pokud chcete zvýšit výkon při zálohování a obnovování:
 - Systémové katalogy umístíte 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 umístování 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 umístěte 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 *IBM Informix: Performance Guide*. 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.

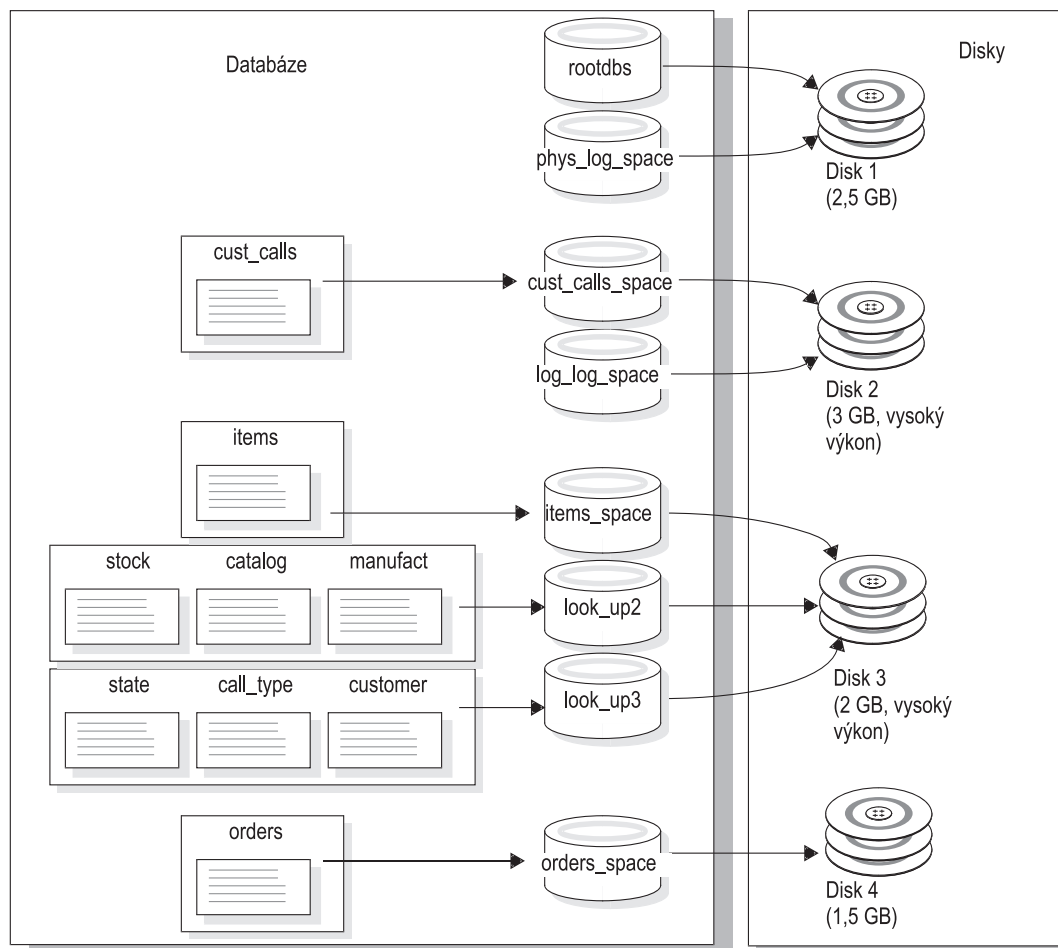
Název tabulky	Maximální velikost	Četnost přístupů
cust_calls	2,5 GB	Nízká
items	0,5 GB	Vysoká
orders	50 MB	Vysoká
customers	50 MB	Nízká
stock	50 MB	Nízká
catalog	50 MB	Nízká
manufact	50 MB	Nízká
state	50 MB	Nízká
call_type	50 MB	Nízká

Ukázkové rozvržení upřednostňující výkon

Obrázek 10-18 na stránce 10-46 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).

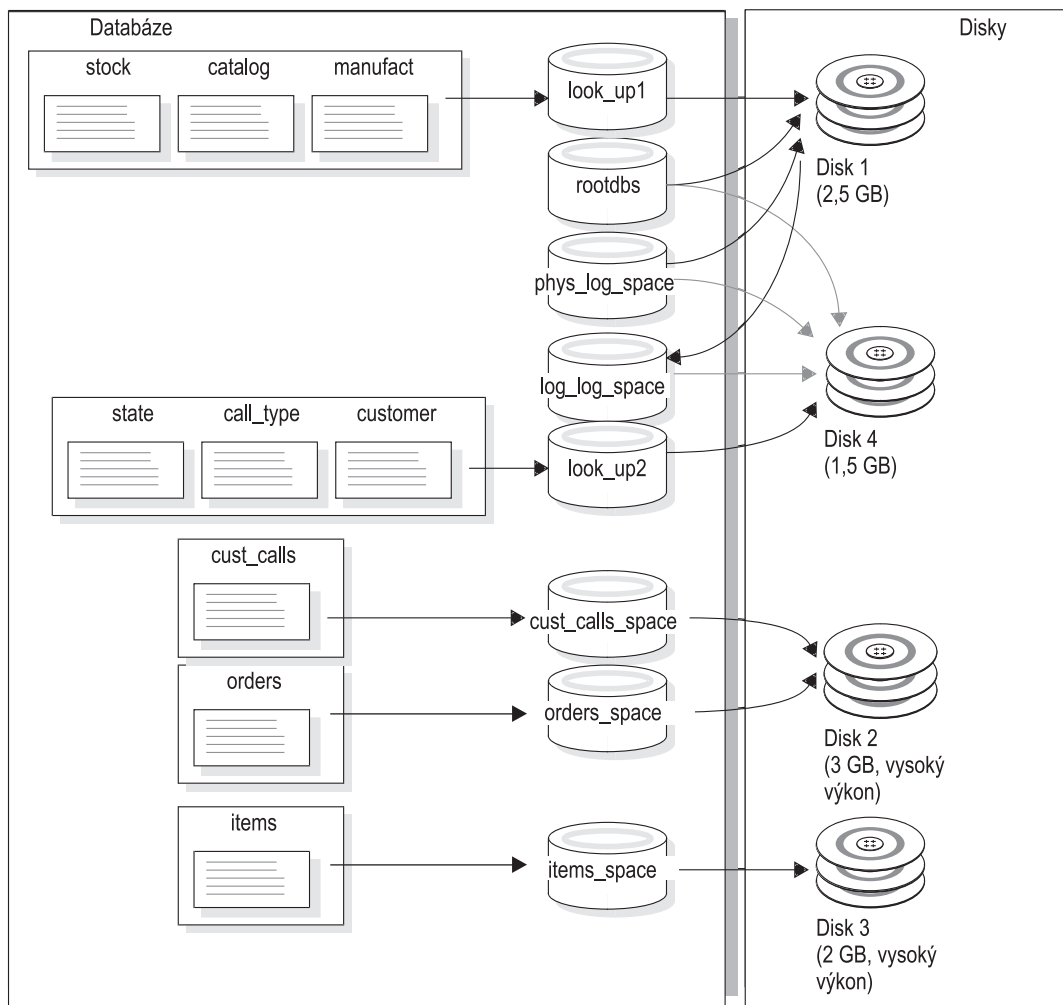


Obrázek 10-18. Rozvržení disků optimalizované na výkon

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 10-19 na stránce 10-47 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 10-19. 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 jí 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 neumísťují přímo na logický disk. Vytvoříte nejprve blok jako logický disk, pak přiřadíte blok prostoru dbspace a nakonec umístíte tabulku do tohoto prostoru dbspace. Další informace naleznete v části “Řízení umístění ukládaných dat” na stránce 10-11.)

Tip: 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 18-3.

Kapitola 11. Správa diskového prostoru

Přidělení diskového prostoru	11-3
Určení posunu	11-4
Určení posunu pro počáteční blok kořenového prostoru dbspace	11-5
Určení posunu pro další bloky	11-5
Použití posunů při vytváření více bloků	11-5
Přidělení prostorů předpřipravených souborů v systému UNIX	11-5
Přidělení diskového prostoru s přímým přístupem v systému UNIX	11-6
Vytvoření symbolických propojení k přímým zařízením (UNIX)	11-6
Přidělení prostoru souborům systému NTFS v systému Windows	11-7
Přidělení diskového prostoru s přímým přístupem v systému Windows	11-8
Určení názvů paměťových prostorů a bloků	11-9
Určení maximální velikosti bloků	11-9
Určení maximálního počtu bloků a paměťových prostorů	11-9
Zálohování po změně fyzického schématu	11-10
Správa prostorů dbspace	11-10
Vytvoření prostoru dbspace, který používá výchozí velikost stránky	11-10
Určení velikosti první oblasti a dalších oblastí prostoru tbspace tbspace	11-12
Vytvoření prostoru dbspace s jinou než výchozí velikostí stránky	11-14
Vytvoření společné oblasti vyrovnávacích paměti pro jinou než výchozí velikost stránky	11-15
Definování velikosti stránky	11-19
Vytvoření dočasného prostoru dbspace	11-19
Co dělat, pokud dojde k vyčerpání diskového prostoru	11-20
Přidání bloku do prostoru dbspace nebo blobspace	11-20
Přidání bloku pomocí programu ON-Monitor (systém UNIX)	11-21
Přejmenování prostorů dbspace	11-22
Další akce, které může být zapotřebí provést po přejmenování prostoru dbspace	11-22
Správa oddílů prostorů dbspace	11-23
Správa prostorů blobspace	11-24
Vytvoření prostoru blobspace	11-24
Příprava prostorů blobspace k ukládání dat typu TEXT a BYTE	11-26
Určení velikosti stránky blobpage	11-26
Určení velikost stránky databázového serveru	11-27
Získání statistických údajů o paměti prostoru blobspace	11-27
Správa prostorů sbospace	11-28
Vytvoření prostoru sbospace	11-28
Určení velikosti oblasti metadat prostoru sbospace	11-29
Přidání bloku do prostoru sbospace	11-29
Změna paměťových charakteristik inteligentních velkých objektů	11-30
Vytvoření dočasného prostoru sbospace	11-31
Vypuštění bloku	11-32
Ověření, zda je blok prázdný	11-32
Vypuštění bloku z prostoru dbspace pomocí obslužného programu onspaces	11-33
Vypuštění bloku z prostoru blobspace	11-33
Vypuštění bloku z prostoru sbospace příkazem onspaces	11-33

Použití možnosti -f (force)	11-33
Odstranění inteligentních velkých objektů, na které neukazují žádné ukazatele	11-34
Vypuštění paměťového prostoru	11-34
Příprava na vypuštění paměťového prostoru	11-34
Vypuštění zrcadlového paměťového prostoru	11-35
Vypuštění paměťového prostoru s obslužným programem onspaces.	11-35
Vypuštění prostorů dbspace a blobspace pomocí programu ON-Monitor (UNIX).	11-35
Zálohování po vypuštění paměťového prostoru	11-36
Správa prostorů extspace	11-36
Vytvoření prostoru extspace	11-36
Vypuštění prostoru extspace	11-37
Přeskakování nepřístupných fragmentů	11-37
Použití konfiguračního parametru DATASKIP	11-37
Použití funkce Dataskip obslužného programu onspaces	11-37
Použití obslužného programu onstat ke kontrole stavu funkce dataskip	11-38
Použití příkazu SET DATASKIP jazyka SQL	11-38
Vliv funkce dataskip na transakce	11-38
Jak určit, kdy používat parametr dataskip	11-39
Určení, kdy přeskakovat vybrané fragmenty	11-39
Určení, kdy přeskočit všechny fragmenty	11-39
Monitorování využití fragmentace.	11-40
Zobrazení databází	11-40
Použití tabulek SMI	11-40
Použití programu ISA	11-40
Použití programu ON-Monitor (systém UNIX)	11-41
Monitorování využití disku	11-41
Monitorování bloků	11-41
onstat -d	11-41
Příkaz onstat -d update	11-42
Příkaz onstat -D	11-43
Příkaz onstat -g iof	11-43
Příkaz oncheck -pr	11-44
Příkaz oncheck -pe	11-45
Použití programu IBM Informix Server Administrator	11-45
Použití programu ON-Monitor (systém UNIX)	11-46
Použití tabulek SMI	11-46
Monitorování prostorů tblspace a oblastí.	11-47
Použití tabulek SMI	11-47
Monitorování jednoduchých velkých objektů v prostoru blobspace	11-47
Příkaz onstat -O	11-47
Určení zaplnění stránek blobpage pomocí příkazu oncheck -pB	11-49
Monitorování využití prostoru blobspace pomocí příkazu oncheck -pe	11-49
Monitorování jednoduchých velkých objektů v prostoru dbspace pomocí příkazu oncheck -pT	11-50
Monitorování prostorů sbspace.	11-51
Použití příkazu onstat -d.	11-53
Použití příkazů oncheck -ce a oncheck -pe	11-54
Použití příkazu oncheck -cs.	11-55
Použití příkazu oncheck -ps.	11-56
Monitorování oblastí metadat a uživatelských dat	11-57
Použití příkazu onstat -g smb c.	11-58

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 10, “Uložení dat”, na stránce 10-1.

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.

Průručka *IBM Informix: Dynamic Server Performance Guide* obsahuje také informace o správě diskových prostorů. Popisuje zejména, jak lze eliminovat prokládané oblasti, uvolnit prostor v prázdné oblasti a zlepšit diskový vstup - výstup.

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 10-4
- “Objem diskového prostoru potřebný k uložení dat” na stránce 10-40
- “Pravidla pro rozvržení disků” na stránce 10-42

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.

Pouze pro UNIX

Přidělíte-li diskový prostor s přímým přístupem v systému UNIX, doporučuje se použít příkaz **ln** systému UNIX a vytvořit propojení mezi názvem znakově orientovaného zařízení a názvem jiného souboru. Další informace o tomto tématu naleznete v části “Vytvoření symbolických propojení k přímým zařízením (UNIX)” na stránce 11-6.

Použití souboru v systému UNIX a základního rozhraní operačního systému pro diskový prostor databázového serveru je také označováno jako použití *předpřipraveného prostoru*.

Konec Pouze 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 pamětí v systému UNIX” na stránce 10-4.

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 terabajty.

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 naleznete v kapitole zaměřené na konfigurační parametry 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 11-10.

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 předpřipraveného souboru `usr/data/my_chunk` diskovému prostoru 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`
3. Vytvořte blok zapsáním souboru `null` pomocí příkazu `cat` do souboru, který bude databázový server používat pro diskový prostor:
`cat /dev/null > my_chunk`
4. Nastavte oprávnění souboru na hodnotu `660` (`rw-rw----`):
`chmod 660 my_chunk`

5. 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
```

6. 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 11-10, “Vytvoření prostoru blobspace” na stránce 11-24 a “Vytvoření prostoru sbpace” na stránce 11-28.

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:

1. Vytvořte a nainstalujte přímé zařízení.

Specifické pokyny jak nainstalovat diskový prostor s přímým přístupem v systému UNIX naleznete v dokumentaci používaného operačního systému a v části “Přístup k disku s vyrovnávací pamětí a bez vyrovnávací paměti v systému UNIX” na stránce 10-4.

2. 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 11-5.

3. Ověřte, že oprávnění znakově orientovaných zařízení operačního systému je nastaveno na hodnotu **crw-rw----**.

4. Vytvořte symbolické propojení mezi názvem znakově orientovaného zařízení a názvem dalšího souboru pomocí příkazu **link** systému UNIX: **ln -s**. Podrobnější informace naleznete v části “Vytvoření symbolických propojení k přímým zařízením (UNIX)” na stránce 11-6.

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 **link** systému UNIX (obvykle **ln**). Ověřte, že

jak zařízení, tak propojení existují, provedením příkazu **ls -l** systému UNIX (**ls -lg** v systému BSD) v adresáři příslušného zařízení. 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 zotavení 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, blobpace a sbpace:

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 vytvořili, pomocí obslužného programu **onspace**. Informace o vytváření prostorů dbspace a blobpace naleznete v části "Vytvoření

prostoru dbspace, který používá výchozí velikost stránky” na stránce 11-10 and
“Vytvoření prostoru blobspace” na stránce 11-24.

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, spusťte 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 **dbspc3**.
`onspaces -a dbspc3 \\.\e: -o 5200 -s 5000`

Určení fyzické 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 **dbspc3**.
`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:

Pouze pro UNIX

- Pokud používáte disk s přímým přístupem v systému UNIX, je doporučeno použít propojenou cestu. (Další informace naleznete v části “Vytvoření symbolických propojení k přímým zařízením (UNIX)” na stránce 11-6.)

Konec Pouze 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 znaků.
- 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 terabajty, 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 zotavení 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 nebo 2.

Zálohování úrovně 0 změněných paměťových prostorů musíte provést, aby byla zajištěna možnost obnovení 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 tblspace **tblspace** 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 11-14.

Informace o vytváření prostoru dbspace pomocí programu ISA naleznete v nápovědě online programu ISA.

Libovolný nově přidáný prostor dbspace a jeho zrcadlo, pokud existuje, jsou okamžitě dostupné. Používáte-li zrcadlení, můžete prostor dbspace zrcadlit při vytvoření. Zrcadlení je okamžitě platné.

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 jednoho uživatele nebo v klidovém režimu.
3. Přidělte diskový prostor pro prostor dbspace podle popisu v části “Přidělení diskového prostoru” na stránce 11-3.
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 11-14, vytváříte-li prostor dbspace s jinou než výchozí velikostí stránky.
5. Nechcete-li určit velikost první oblasti a dalších oblastí pro prostor tbspace **tbspace** v prostoru dbspace, přejděte ke kroku 7.
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 11-12.
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 **dbspce1** s posunem 5000 kB pro primární i zrcadlený blok pomocí diskového prostoru s přímým přístupem v systému UNIX:

```
onspaces -c -d dbspce1 -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 prostorů dbspace pomocí obslužného programu **onspaces** naleznete v kapitole o obslužných programech v příručce “Prostory dbspace” na stránce 10-11.

Vytvoření prostoru dbspace pomocí programu ON-Monitor (systém UNIX):

1. Vyberte volbu **Dbspaces > Create**.
2. Zadejte název nového prostoru dbspace do pole **Dbspace 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 11-14.
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. Zadejte posun do pole **Offset**.
8. Zadejte velikost bloku v kilobajtech do pole **Size**.
9. Zrcadlíte-li tento prostor dbspace, zadejte úplnou cestu k zrcadlenému bloku, jeho velikost a nepovinný posun 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 tbspace tbspace

Zadáním velikosti první oblasti a dalších oblastí můžete snížit počet oblastí prostoru tbspace **tbspace** a četnost situací, ve kterých je zapotřebí umístit oblasti prostoru tbspace **tbspace** 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 oblastí. Pokud neurčíte velikost první oblasti a dalších oblastí pro prostor tbspace **tbspace**, použijte dynamický server existující výchozí velikost oblastí.

Velikost první oblasti a dalších oblastí pro prostor tbspace **tbspace** 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 tbspace **tbspace** 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 dbspace již nelze velikost první a dalších oblastí měnit.

Určení velikosti první oblasti a dalších oblastí:

1. Určete celkový počet stránek potřebných v prostoru tblspace **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 **tblspace**.
2. Vypočtete počet kilobajtů potřebný pro počet stránek. Tento počet závisí na počtu kilobajtů připadajícím na stránku v systému.
3. Zjistěte potřeby správy prostorů v systému tím, že zvážíte, jak je důležité, aby všechny oblasti pro prostor tblspace **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. Příklad: 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.

Typ oblasti	Minimální velikost	Maximální velikost
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 **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 dbpace1 -p /usr/data/dbpace1
-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 HDR 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 musí být zároveň menší než 16 kB.

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 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, abyste tuto úlohu provedli 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 části “Vytvoření prostoru dbSPACE s jinou než výchozí velikostí stránky” na stránce 11-14.

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 11-19.

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 platform 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 tyto 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 tyto 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 řádcích 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 se systémem 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 určovány pomocí konfiguračních parametrů BUFFERS, LRUS, LRU_MAX_DIRTY a LRU_MIN_DIRTY před verzí 10.0, 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.
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 disku z pověření klientské aplikace. Počet vyrovnávacích pamětí, které požaduje databázový server, závisí na těchto aplikacích.</p> <p>Přístupuje-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>Obecně by měl mít prostor vyrovnávacích pamětí rozsah od 20 do 25 procent fyzické paměti. Doporučujeme, abyste po nastavení prostoru vyrovnávacích pamětí (buffers * <i>velikost_systémové_stránky</i>) na 20 procent fyzické paměti vypočítali všechny ostatní parametry sdílené paměti.</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 použití 64bitovou platformu naleznete v souboru Poznámky k verzi. Maximální počet vyrovnávacích pamětí v systému Solaris je 536870912.)</p>
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

pole	Popis	Rozsah hodnot
lru_min_dirty	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. Pokud byla zadána hodnota parametru mimo rozsah povolených hodnot, bude použita výchozí hodnota 50.00.	0 až 100 (jsou povoleny i neceločíselné hodnoty)
lru_max_dirty	Určuje procento změněných stránek ve frontách LRU, po jehož dosažení bude fronta vyčištěna. Pokud byla zadána hodnota parametru mimo rozsah povolených hodnot, bude použita výchozí hodnota 60.00.	0 až 100 (jsou povoleny i neceločíselné hodnoty)

Pokud se databázový server nachází v režimu online, v klidovém režimu nebo v jednoruživatelském režimu, 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)>
```

Příklad:

```
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*.

Pokud potřebujete změnit velikost existující společné oblasti vyrovnávacích pamětí nebo společnou oblast vyrovnávacích pamětí odstranit, bude zapotřebí databázový server restartovat.

Je doporučeno nastavit konfigurační parametr PHYSBUFF na násobek největší použité velikosti stránky. Pokud například používáte stránky o velikosti 16 KB jako největší stránky, můžete nastavit parametr PHYSBUFF na hodnotu 32.

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.

Definování velikosti stránky

Pomocí volby **onspaces -k** můžete nastavit velikost stránky v kilobajtech, a to následovně:

```
onspaces -c -d prostor_dbspace [-t] [-k velikost_stránky]  
-p cesta -o posun -s velikost [-m cesta_posun]
```

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*.

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 11-10.

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. Všechny prostory dbspace určené seznamem **DBSPACETEMP** však musejí mít totožnou velikost stránky.

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.
- Znovu inicializujte 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 10-14.

Co dělat, pokud dojde k vyčerpání diskového prostoru

Pokud je počáteční blok vytvářeného prostoru dbspace předpřipravený soubor systému UNIX nebo soubor systému NTFS v systému Windows, ověří databázový server, zda je dostupný dostatečný diskový prostor k vytvoření počátečního bloku. 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 zbyval 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 sbpace 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. Aby uživatel mohl přidat blok v systému UNIX, musí být přihlášen 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 jednoho uživatele, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.
3. Přidělte bloku diskový prostor podle popisu v části “Přidělení diskového prostoru” na stránce 11-3.
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 také naleznete v části “Co dělat, pokud dojde k vyčerpání diskového prostoru” na stránce 11-20.

5. Po vytvoření bloku je třeba provést zálohování úrovně 0 kořenového prostoru dbspace a prostoru dbspace, blobspace nebo sbpace, 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 obslužných programech 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 sbspace, 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 sbspace můžete přejmenovat:

- Pokud je povolena replikace Enterprise Replication.
- V primárním databázovém serveru replikace High-Availability Data Replication (HDR).

Prostory dbspace a sbspace nemůžete přejmenovávat v sekundárním databázovém serveru replikace HDR a také v případě, že sekundární databázový server replikace HDR je 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 příručce *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 zkompilejte.

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.

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 **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 například 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 *IBM Informix: Dynamic Server Performance Guide* 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 11-20.

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 blobspace diskový prostor, jak je popsáno v části “Přidělení diskového prostoru” na stránce 11-3.
2. Určete optimální velikost stránky blobpage ve svém prostředí.
Pokyny naleznete v části “Určení velikosti stránky blobpage” na stránce 11-26.

Vytvoření prostoru blobspace pomocí obslužného programu onspaces:

1. Aby uživatel mohl vytvořit prostor blobspace v systému UNIX, musí být přihlášen jako uživatel **informix** nebo **root**.
Aby mohl uživatel vytvořit prostor blobspace v systému Windows, musí být členem skupiny **Informix-Admin**.
2. Zajistěte, aby byl databázový server v režimu online, v režimu jednoho uživatele, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.
3. K přidání prostoru blobspace použijte volby obslužného programu **onspaces -c -b**.
 - a. Určete explicitní název cesty k prostoru blobspace. Pokud je prostor blobspace zrcadlený, je třeba zadat název cesty a velikost primárního i zrcadleného bloku.
 - b. Pomocí volby **-o** zadejte posun prostoru blobspace.
 - c. Pomocí volby **-s** určete velikost prostoru blobspace 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 11-26. 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 blobspace nevytvoří a zobrazí chybovou zprávu. Další informace také naleznete v části “Co dělat, pokud dojde k vyčerpání diskového prostoru” na stránce 11-20.

4. Po vytvoření prostoru blobspace musíte provést zálohování úrovně 0 kořenového prostoru dbspace a nového prostoru blobspace.

Následující příklad znázorňuje, jak vytvořit zrcadlený prostor blobspace **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 blobspace je vytvářen v systému UNIX v diskovém 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 prostorů blobspace pomocí obslužného programu onspaces naleznete v kapitole o obslužných programech v příručce *IBM Informix: Administrator's Reference*.

Vytvoření prostoru blobspace pomocí programu ON-Monitor (UNIX):

1. Vyberte volbu **Dbspaces > BLOBSpace**.
2. Zadejte název nového prostoru blobspace do pole **BLOBSpace Name**.
3. Chcete-li vytvořit zrcadlo pro počáteční blok prostoru blobspace, 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 velikost stránky databázového serveru” na stránce 11-27. 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 počátečnímu 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 zotavit 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 14-9.)

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 *IBM Informix: Dynamic Server Performance Guide*.

Pokud tabulka obsahuje více sloupců typu TEXT nebo BYTE a objekty v těchto sloupcích mají různou velikost, uložte tyto sloupce do různých prostorů blobpace tak, aby každý měl vhodnou velikost stránky. Další informace naleznete v části “Tabulky” na stránce 10-29.

Určení velikost 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:

- Spusíte 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 pro UNIX

- V programu ON-Monitor zobrazte systémovou velikost stránky pomocí volby **Parameters > Shared-Memory** nebo **Parameters > Initialize**.

Konec Pouze pro UNIX

Získání statistických údajů o paměti prostoru blobpace

Při určování optimální velikosti stránek blobpage všech prostorů blobpace mohou pomoci následující příkazy obslužných programů databázového serveru:

- **oncheck -pe**
- **oncheck -pB**

Příkaz **oncheck -pe** poskytuje podrobné informace o objektech uložených v prostoru blobpace:

- Úplné informace o vlastnictví (zobrazené ve formátu *databáze:vlastník.tabulka*) o každé tabulce, jejíž data jsou uložena v bloku prostoru blobpace.
- 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 blobpace.

Příkaz **oncheck -pB** zobrazuje následující statistické údaje o každé tabulce databáze:

- Počet stránek blobpage používaných tabulkou nebo databází v každém z prostorů blobpace.
- Průměrné zaplnění stránek blobpage používaných všemi jednoduchými velkými objekty uloženými jako součást tabulky nebo databáze.

Další informace naleznete v části “Monitorování využití prostoru blobpace pomocí příkazu oncheck -pe” na stránce 11-49, v části “Určení zaplnění stránek blobpage pomocí příkazu oncheck -pB” na stránce 11-49 a v části týkající se optimalizace

velikosti stránek blobpage v prostorech blobspace v kapitole o posouzení výkonu tabulek v příručce *IBM Informix: Dynamic Server Performance Guide*.

Správa prostorů sbSPACE

Tato část popisuje, jak vytvořit standardní nebo dočasný prostor sbSPACE, jak monitorovat metadata, jak přidat blok k prostoru sbSPACE a jak změnit paměťové charakteristiky inteligentního velkého objektu.

Vytvoření prostoru sbSPACE

K vytvoření prostoru sbSPACE použijte obslužný program onspaces nebo program ISA.

Vytvoření prostoru sbSPACE pomocí obslužného programu onspaces:

1. Aby uživatel mohl vytvořit prostor sbSPACE v systému UNIX, musí být přihlášen jako uživatel **informix** nebo **root**.
Aby mohl uživatel vytvořit prostor sbSPACE v systému Windows, musí být členem skupiny **Informix-Admin**.
2. Zajistěte, aby byl databázový server v režimu online, v režimu jednoho uživatele, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.
3. Vytvořte prostor sbSPACE pomocí příkazu **onspaces -c -S**.
 - a. Pomocí volby **-p** zadejte název cesty, pomocí volby **-o** zadejte posun a pomocí volby **-s** zadejte velikost prostoru sbSPACE.
 - b. Pokud chcete prostor sbSPACE zrcadlit, zadejte pomocí volby **-m** cestu k zrcadlu a posun zrcadla.
 - c. Pokud chcete použít výchozí paměťové charakteristiky prostoru sbSPACE, volbu **-Df** vynechte.
Pokud chcete určit odlišné paměťové charakteristiky, použijte volbu **-Df**. Další informace naleznete v části “Paměťové charakteristiky prostorů sbSPACE” na stránce 10-19.
 - d. První blok prostoru sbSPACE musí obsahovat oblast metadat.
Výpočet velikosti oblasti metadat přenechat databázovému serveru nebo můžete oblast metadat prostoru sbSPACE určit sami. Další informace naleznete v části “Určení velikosti oblasti metadat prostoru sbSPACE” na stránce 11-29.
4. Po vytvoření prostoru sbSPACE musíte provést zálohování úrovně 0 kořenového prostoru dbSPACE a nového prostoru sbSPACE.
5. Abyste mohli začít ukládat inteligentní velké objekty do tohoto prostoru sbSPACE, určete název tohoto prostoru v konfiguračním parametru SBSPACENAME.
6. K zobrazení informací o prostoru sbSPACE použijte příkazy **onstat -d**, **onstat -g smb s a oncheck -cs, -cS, -ps** nebo **-pS**.
Další informace naleznete v části “Monitorování prostorů sbSPACE” na stránce 11-51.

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 prostorů sbspace a o výchozích volbách pro inteligentní velké objekty naleznete v části týkající se obslužného programu **onspaces** v kapitole o obslužných programech 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 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 oblasti 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 *IBM Informix: Dynamic Server Performance Guide*. Další informace také naleznete v části “Monitorování oblastí metadat a uživatelských dat” na stránce 11-57.

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 k prostoru sbspace pomocí obslužného programu onspaces:

1. Zajistěte, aby byl databázový server v režimu online, v režimu jednoho uživatele, v klidovém režimu nebo ve fázi čištění režimu rychlé obnovy.

2. Vytvořte blok prostoru sbpace 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 sbpace používat oblast metadat v některém z jiných bloků.
3. Po přidání bloku do prostoru sbpace zapiše databázový server záznamy protokolu CHRESERV a CHKADJUP.
4. Proveďte zálohování úrovně 0 kořenového prostoru dbspace a prostoru sbpace.
5. Pomocí příkazů **onstat -d** a **oncheck -pe** monitorujte množství volného prostoru v bloku prostoru sbpace.
Další informace naleznete v části “Monitorování prostorů sbpace” na stránce 11-51.

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 11-20 a v části týkající se obslužného programu onspaces v kapitole o obslužných programech v příručce *IBM Informix: Administrator's Reference*.

Změna paměťových charakteristik inteligentních velkých objektů

Pomocí příkazu **onspaces -ch** můžete změnit následující výchozí paměťové charakteristiky prostoru sbpace:

- 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 10-19 a v části týkající se správy prostorů sbspace v kapitole o posouzení výkonu tabulek v příručce *IBM Informix: Dynamic Server Performance Guide*.

Vytvoření dočasného prostoru sbspace

Podrobné informace a pravidla k určení, kam ukládat dočasné inteligentní velké objekty naleznete v části “Dočasné prostory sbspace” na stránce 10-25. 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:

1. Přidělte dočasnému prostoru sbspace diskový prostor. Podrobnosti naleznete v části “Přidělení diskového prostoru” na stránce 11-3.

Informace o parametru SBSPACETEMP naleznete v kapitole o konfiguračních parametrech v příručce *IBM Informix: Administrator's Reference*.

2. 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**.

Obrázek 11-1 znázorňuje výstup příkazu. Povšimněte si hodnoty 0xa001 v prvním sloupci **flags** a hodnoty N S ve druhém sloupci **flags** dočasného prostoru sbspace.

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 jazyka 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*.

```

Dbspaces
address  number  flags  fchunk  nchunks  flags  owner  name
ab01660  5          0xa001 5       1        N S    informix tempsbsp

Chunks
address  chk/dbs  offset  size  free  bpages  flags  pathname
ab01a50  5        5       0     500   347   347   POS   ./tempsbsp
          Metadata 100     74    100

```

Obrázek 11-1. Výstup příkazu `onstat -d` zobrazující dočasné prostory `sbspace`

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 jedouzivatelském režimu nebo v klidovém režimu	Databázový server v režimu offline
Blok prostoru <code>dbspace</code>	Ano	Ano	Ne
Blok dočasného prostoru <code>dbspace</code>	Ano	Ano	Ne
Blok prostoru <code>blobospace</code>	Ne	Ano	Ne
Blok prostoru <code>sbspace</code> nebo dočasného prostoru <code>sbspace</code>	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:

```
Chunk is not empty.
```

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 programu 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 týkající se obslužných programů v příručce *IBM Informix: Administrator's Reference*.

Další informace o vypuštění bloku z prostoru dbspace příkazem **onspaces** naleznete v kapitole týkající se obslužných programů 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 blobspace, který je popsán v části “Vypuštění bloku z prostoru dbspace pomocí obslužného programu onspaces” na stránce 11-33 s výjimkou toho, že databázový server musí být v klidovém režimu nebo v jednoruživatelském režimu. 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

Následující příklad vypouští blok z prostoru **sbsp3** v systému UNIX. Je určen posun 300 kB. Databázový server musí být při vypouštění bloku z prostoru sbospace nebo dočasněho prostoru sbospace v jednoruživatelském 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 volbě **onspaces -cl** naleznete v kapitole o obslužných programech 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 sbpace, dočasný prostor sbpace 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 jednoho uživatele 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 sbpace je třeba vypustit všechny tabulky, které obsahují sloupec CLOB a BLOB, který odkazuje na objekty uložené v prostoru sbpace. Pro prostor sbpace není potřeba odstranit sloupce, které odkazují na prostor sbpace, tyto sloupce ale musí být nulové; tzn. všechny inteligentní velké objekty nesmí být umístěny v prostoru sbpace.

Tip: 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 19-8.) 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 vypuštění bloku z paměťového prostoru příkazem **onspaces** naleznete v kapitole o obslužných programech 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 části *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 10-27.

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 jedna z následujících zpráv:

```
dataskip is OFF for all dbspaces

dataskip is ON for all dbspaces

dataskip is ON for dbspaces:
  dbspace1    dbspace2    ...
```

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 doporučí 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 selžou:

- 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 přeskokovat 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 přeskokovat fragmenty, určete pomocí nastavení ON *seznam-prostorů-dbspace* seznam prostorů dbspace s fragmenty, které má databázový server přeskokovat.

Určení, kdy přeskočit všechny fragmenty

Nastavení konfiguračního parametru DATASKIP na hodnotu ALL způsobí, že databázový server bude přeskokovat 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 pro jediného uživatele, je důležité, aby data byla rovnoměrně rozložena. Abyste mohli monitorovat využití disků fragmenty, musíte monitorovat prostory tblspace databázového serveru, protože diskovou paměťovou jednotkou fragmentu je prostor tblspace. (Informace o tom, jak monitorovat distribuci dat fragmentované tabulky naleznete v části “Monitorování prostorů tblspace a oblastí” na stránce 11-47.)

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 11-41.)

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 11-41.

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 dokáže 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**. Podrobnosti naleznete v části “Příkaz onstat -d update” na stránce 11-42.

Obrázek 11-2 znázorňuje ukázkový výstup příkazu **onstat -d**. Sloupec **flags** v části **chunk** poskytuje 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 zotavení nebo zda se jedná o nový blok.

Důležité: Než bude možné aktivovat zrcadlení, musí být nejprve provedeno zálohování úrovně 0. Totéž je třeba provést i po vypnutí zrcadlení.

```
Dbspaces
address  number  flags    fchunk  nchunks  flags    owner  name
40c980   1         0x1     1        1         N        informix rootdbs
40c9c4   2         0x1     2        1         N        informix fstdbs
40ca08   3         0x11    3        1         N B      informix fstblob
  3 active, 2047 maximum
```

Note: For BLOB chunks, the number of free pages shown is out of date.
Run 'onstat -d update' for current stats.

```
Chunks
address  chk/dbs  offset  size  free  bpages  flags  pathname
40c224   1 1 0     20000 14001          PO-   /home/server/root_chunk
40c2bc   2 2 0     2000  1659          PO-   /home/server/fst_chunk
40c354   3 3 0     12500 ~6250  6250  POB   /home/server/blob_chunk
  3 active, 2047 maximum
```

Obrázek 11-2. Výstup příkazu **onstat -d**

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. Obrázek 11-3 znázorňuje prostor blobSPACE s názvem **fstblob**, který obsahuje jediný blok zvaný **blob_chunk**.

```

Dbspaces
address number flags fchunk nchunks flags owner name
a7317d8 1 0x1 1 1 N informix rootdbs
40c9c4 2 0x1 2 1 N informix fstdbs
40ca08 3 0x11 3 1 N B informix fstblob
3 active, 2047 maximum

```

Waiting for server to update BLOB chunk statistics:

```

Chunks
address chk/dbs offset size free bpages flags pathname
40c224 1 1 0 2000 14001 P0- /home/server/root_chunk
40c2bc 2 2 0 2000 1659 P0- /home/server/fst_chunk
40c354 3 3 0 12500 ~6237 6250 P0B /home/server/blob_chunk
3 active, 2047 maximum

```

Obrázek 11-3. Výstup příkazu `onstat -d update s` informacemi o prostorech 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**).

Obrázek 11-4 znázorňuje ukázkový výstup.

```

Dbspaces
address number flags fchunk nchunks flags owner name
40d100 1 1 1 1 N informix rootdbs
40d144 2 2 2 1 M informix cookedspace
40d188 3 10 3 1 N B informix cookedblob
3 active, 2047 maximum

Chunks
address chk/dbs offset page Rd page Wr pathname
40c274 1 1 0 146 4 /home/server/root_chunk
40c30c 2 2 0 1 0 /home/server/test_chunk
40c8fc 2 2 0 36 0 /home/server/test_mirr
40c3a4 3 3 0 4 0 /home/server/blob_chunk
3 active, 2047 maximum

```

Obrázek 11-4. Výstup příkazu `onstat -D`

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. Obrázek 11-5 znázorňuje ukázkový výstup.

```

...
AIO global files:
gfd pathname          totalops  dskread  dskwrite  io/s
  3 raw_chunk          38808    27241    11567    6.7
  4 cooked_chk1        7925     5660     2265    1.4
  5 cooked_chk2        3729     2622     1107    0.6

```

Obrázek 11-5. Výstup příkazu `onstat -g iof`

Příkaz `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 11-6 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        07/28/2000 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 11-6. Výstup příkazu `oncheck -pr` s informacemi o prostorech `dbspace` a blocích

Příkaz oncheck -pe

Pokud chcete zjistit fyzické rozvržení dat v bloku, spusíte příkaz **oncheck -pe**. Uvedeny jsou prostory dbspace, blobspace a sbspace. Obrázek 11-7 na stránce 11-45 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: 08/08/2000

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 11-7. 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 sbspace.

- 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
Status > Spaces	Zobrazuje informace o stavu paměťových prostorů a bloků.
Dbspaces > Create	Vytvoří prostor dbspace.
Dbspaces > BLOBSpace	Vytvoří prostor blobspace.
Dbspaces > Mirror	Přidá nebo vypustí zrcadlení paměťového prostoru.
Dbspaces > Info	Zobrazuje informace o paměťových prostorech.
Dbspaces > Add Chunk	Přidá blok do paměťového prostoru.
Dbspaces > dataSkip	Spustí nebo zastaví funkci dataskip.
Dbspaces > Chunk	Přidá blok do prostoru dbspace nebo blobspace.
Dbspaces > Drop	Vypustí prostor dbspace nebo blobspace.
Dbspaces > Status	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.
dbsnum	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 zotavení bloku.
mis_offline	Určuje, zda je zrcadlený blok vypnutý.
mis_recovering	Určuje, zda právě probíhá zotavení 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 **sysabnames** získáte informace o všech prostorech **tblspace**. Tabulka **sysabnames** 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 rychlé vyrovnávací paměti podsystému **Optical Subsystem**. Obrázek 11-8 znázorňuje příklad výstupu tohoto příkazu. Zobrazované souhrnné údaje se v každé relaci přičítají. Databázový server opět nastaví souhrnné údaje na hodnotu 0 pouze při spuštění příkazu **onstat -z**.

```
Subsystem not available

Optical StageBlob Cache
System Cache Totals:                               System Blob Totals:
Size Alloc Avail. Number Kbytes Number Kbytes
0 0 0 0 0 0 0

User Cache Totals:                               User Blob Totals:
SID User      Size Number Kbytes Number Kbytes
No sessions currently using the cache
```

Obrázek 11-8. Výstup příkazu **onstat -O**

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 rychlé vyrovnávací paměti, aniž by došlo k přetečení.
kbytes	Počet kilobajtů jednoduchých velkých objektů, které databázový server úspěšně uložil do rychlé vyrovnávací paměti, 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, rychlá vyrovnávací paměť 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 rychlé vyrovnávací paměti, aniž by došlo k přetečení.

kbytes	Počet kilobajtů jednoduchých velkých objektů, které databázový server uložil do rychlé vyrovnávací paměti, aniž by došlo k přetečení.
number	Počet jednoduchých velkých objektů, které databázový server zapsal do prostoru blobpace k dočasnému ukládání velkých objektů.
kbytes	Velikost jednoduchých velkých objektů (v kilobajtech), které databázový server zapsal do prostoru blobpace 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 blobpace.

Spusíte 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 blobpace v kapitole o posouzení výkonu tabulek v příručce *IBM Informix: Dynamic Server Performance Guide*.

Monitorování využití prostoru blobpace pomocí příkazu **oncheck -pe**

Příkaz **oncheck -pe** poskytuje informace o využití prostoru blobpace:

- 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 11-9 znázorňuje ukázkou výstupu příkazu **oncheck -pe**.

```
BLOBSpace Usage Report: fstblob          Owner: informix  Created: 03/01/99
  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 11-9. Výstup příkazu **oncheck -pe** znázorňující využití prostoru blobpace

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 11-10 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
<hr/>					
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 11-10. 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 sbpace v systému:</p> <ul style="list-style-type: none"> • Název, příznaky a vlastníka prostoru sbpace. • Stav protokolování. • Průměrnou velikost inteligentního velkého objektu. • Velikost první oblasti, velikost dalších oblastí a minimální velikost oblasti. • Maximální přístupovou dobu vstupu - výstupu. • Režim uzamykání.
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 11-58.</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>Viz kapitoly “Použití příkazů oncheck -ce a oncheck -pe” na stránce 11-54 a “Monitorování oblastí metadat a uživatelských dat” na stránce 11-57.</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” na stránce 11-53.</p>
oncheck -cs oncheck -ps	<p>Ověří oblasti metadat prostorů sbpace a zobrazí informace o nich. Další informace naleznete v části “Použití příkazu oncheck -cs” na stránce 11-55 a v části “Použití příkazu oncheck -ps” na stránce 11-56.</p>

Příkaz	Popis
oncheck -cS	Zobrazí informace o oblastech inteligentních velkých objektů a o oblastech uživatelských dat v prostorech sbpace.
oncheck -pS	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>IBM Informix: Dynamic Server Performance Guide</i> .

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.

Obrázek 11-11 znázorňuje ve výstupu příkazu **onstat -d** informace o prostoru **rootdbs** a o prostoru sbpace **s9_sbsp**. Příznaky **8000** a **S** ve sloupci **Flags** označují prostor sbpace. První řádek každého bloku sbpace zobrazuje informace o celém prostoru sbpace a o oblasti uživatelských dat. Druhý řádek zobrazuje informace o oblasti metadat. V tomto příkladu je ve sloupci **Size** zobrazen celkový počet stránek prostoru **s9_sbsp** jako 1000. Oblast uživatelských dat tvoří 842 stránek a 726 stránek je volných. Oblast metadat tvoří 105 stránek a 60 stránek je volných.

Pokud chcete zjistit celkové množství využitého místa, spusťte příkaz **oncheck -pe**. Další informace naleznete v části “Použití příkazů **oncheck -ce** a **oncheck -pe**” na stránce 11-54.

```

Dbspaces
address number flags fchunk nchunks flags owner name
a1b01d8 1 1 1 1 N informix rootdbs
a1b0658 2 8000 2 1 N S informix s9_sbsp
2 active, 2047 maximum

Chunks
address chk/dbs offset size free bpages flags pathname
a1b0320 1 1 0 75000 64588 PO- /ix/ids9.2/root_chunk
a1b04f8 2 2 0 1000 726 842 POS /ix/ids9.2/./s9_sbsp
Metadata 105 60 105
2 active, 2047 maximum

```

Obrázek 11-11. Výstup příkazu **onstat -d s** informacemi o prostorech sbpace

Příkaz **onstat -d** neregistruje stránku sbpage 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 sbspace, 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 sbspace. Výstup volby **-pe** poskytuje následující informace o využití prostorů sbspace:

- 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 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.

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 v oblasti uživatelských dat.

Tip: Příkaz **oncheck -pe** poskytuje informace o využití prostorů sbspace, přičemž jednotkou jsou stránky databázového serveru, nikoli stránky sbpage.

Obrázek 11-12 znázorňuje ukázkový výstup. V tomto příkladu se v prostoru sbspace s názvem **s9_sbsp** 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 11-12. Výstup příkazu `oncheck -pe` znázorňující využití prostoru `sbspace`

Použití příkazu `oncheck -cs`

Příkazy `oncheck -cs` a `oncheck -Cs` ověřují oblast metadat prostoru `sbspace`. Obrázek 11-13 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 11-13 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'.L0_hdr_partn	0x200004	21	11
s9_sbspc:'informix'.L0_ud_free	0x200005	2	3

Obrázek 11-13. Výstup příkazu `oncheck -cs`

Použití příkazu `oncheck -ps`

Příkaz `oncheck -ps` ověří a zobrazí informace o oblastech metadat v oddílech prostoru sbspace. Obrázek 11-14 na stránce 11-57 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, spusíte 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, přidělte 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 Jun 25 14:14:33 1999
```

Obrázek 11-14. 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 11-54.
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 11-29.

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 oblasti metadat prostoru sbpace” na stránce 11-29.

Použití příkazu onstat -g smb c

Pomocí příkazu **onstat -g smb c** můžete monitorovat množství volného prostoru ve všech blocích prostorů sbpace a velikosti oblasti uživatelských dat, oblasti metadat a vyhrazené oblasti ve stránkách. Obrázek 11-15 znázorňuje situaci, kdy blok číslo 2 prostoru **sbpace1** obsahuje 2253 použitých stránek (**usr pgs**) a 2245 volných stránek (**free pg**). Posun počáteční stránky první oblasti uživatelských dat **Ud1** je 53 a počet stránek oblasti je 1126. Oblast metadat **Md** začíná počáteční stránkou s posunem 1179 a obsahuje 194 stránek. Rezervovaná oblast **Ud2** začíná na stránce s posunem 1373 a obsahuje 1127 stránek.

Chunk Summary:

```

sbnm 2 chunk 2
chunk:  address  flags  offset  size  orig fr  usr pgs  free pg
        303cf2a8  F-----  0      2500  2253    2253    2245
        path: /usr11/myname/sbpace1

        start pg  npages
Ud1  :    53      1126
Md   :   1179     194
Ud2  :   1373    1127

```

Obrázek 11-15. Výstup příkazu onstat -g smb c

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 ESQ/L/C	Ano	Ano	<i>IBM Informix: ESQ/L/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 12. Protokolování

Procesy databázového serveru, které vyžadují protokolování	12-1
Protokolování transakcí	12-3
Protokolování příkazů jazyka SQL a aktivity databázového serveru	12-4
Aktivita, která se vždy protokoluje	12-4
Aktivity protokolované u databázi s protokolováním transakcí	12-5
Aktivita, která se neprotokoluje	12-6
Stav protokolování databáze	12-6
Protokolování transakcí bez vyrovnávací paměti	12-7
Protokolování transakcí s vyrovnávací paměti	12-8
Protokolování transakcí kompatibilní se standardem ANSI	12-8
Databáze bez protokolování	12-8
Databáze s různými stavy používání vyrovnávací paměti pro protokolování	12-9
Protokolování databáze v prostředí X/Open DTP	12-9
Nastavení nebo změny stavu nebo režimu protokolování	12-9

Obsah kapitoly

Tato kapitola popisuje protokolování databázi 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 paměti nebo zda tabulka používá protokolování.

Pokud chcete změnit stav protokolování databáze, naleznete další informace v části “Nastavení nebo změny stavu nebo režimu protokolování” na stránce 12-9.

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 zotavení dat nebo dokud se administrátor nerozhodne, že již nejsou pro zotavení dat dále potřeba. Další informace o logických protokolech uvádí Kapitola 14, “Logický protokol”, na stránce 14-1.

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 12-3.
- **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 *fyzicky 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 “Podrobnosti rychlé obnovy po úplném kontrolním bodu” na stránce 16-15.)
- **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 sežene před odstraněním podřízených řádků. Další informace o dědičnosti tabulek naleznete v příručce

IBM Informix: Database Design and Implementation Guide. Další informace o omezeních pomoci 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 23-16.
- **Replikace HDR.** Replikace HDR (High-Availability Data Replication) používá záznamy logického protokolu k udržování konzistence dat na dvou různých 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 “Jak pracuje replikace HDR” na stránce 20-7.
- **Replikace 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 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 obsahuje část “Stav protokolování databáze” na stránce 12-6 a Kapitola 13, “Správa režimu protokolování databáze”, na stránce 13-1.

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 části “Aktivita, která se vždy protokuluje” na stránce 12-4 a v části “Protokolování databáze v prostředí X/Open DTP” na stránce 12-9.

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 10-30.

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:

- Příkazy DDL jazyka SQL.
- 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.

Příkazy DDL jazyka SQL uvádí Tabulka 12-1.

Tabulka 12-1. Příkazy DDL jazyka SQL

ALTER ACCESS METHOD	CREATE ROUTINE	DROP TABLE
ALTER FRAGMENT	CREATE ROW TYPE	DROP TRIGGER
ALTER FUNCTION	CREATE SCHEMA	DROP TYPE
ALTER INDEX	CREATE SYNONYM	DROP VIEW
ALTER OPAQUE TYPE	CREATE TABLE	GRANT
ALTER PROCEDURE	CREATE TEMPORARY TABLE	GRANT FRAGMENT
ALTER ROUTINE	CREATE TRIGGER	RENAME COLUMN
ALTER TABLE	CREATE VIEW	RENAME DATABASE
CREATE ACCESS METHOD	DROP ACCESS METHOD	RENAME INDEX
CREATE AGGREGATE	DROP AGGREGATE	RENAME TABLE
CREATE CAST	DROP CAST	REVOKE
CREATE DATABASE	DROP DATABASE	REVOKE FRAGMENT
CREATE DISTINCT TYPE	DROP FUNCTION	ROLLBACK WORK
CREATE EXTERNAL TABLE	DROP INDEX	START VIOLATIONS TABLE
CREATE FUNCTION	DROP OPCLASS	TRUNCATE TABLE
CREATE INDEX	DROP PROCEDURE	STOP VIOLATIONS TABLE
CREATE OPAQUE TYPE	DROP ROLE	UPDATE STATISTICS
CREATE OPCLASS	DROP ROUTINE	
CREATE PROCEDURE	DROP ROW TYPE	
CREATE ROLE	DROP SYNONYM	

Aktivita 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 protokuluje” na stránce 12-4. 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 10-41.

Stav protokolování databáze

Abyste mohli využít všechny vlastnosti uvedené v části “Procesy databázového serveru, které vyžadují protokolování” na stránce 12-1, 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 13-7. Stav protokolování transakce označuje libovolný z následujících typů protokolování:

- Protokolování transakcí bez vyrovnávací paměti.
- Protokolování transakcí s vyrovnávací pamětí.
- Protokolování transakcí kompatibilní se standardem ANSI.
- Bez protokolování.

Všechny záznamy logického protokolu procházejí vyrovnávací pamětí 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 pamětí a pro protokolování transakcí bez vyrovnávací paměti. Další informace znázorňuje Obrázek 8-1 na stránce 8-4 a jsou také uvedeny v části “Vyprazdňování vyrovnávací paměti logického protokolu” na stránce 8-37.

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 disk, dokud nenastane jedna z níže uvedených situací:

- Vyrovnávací paměť je zaplněna.
- Potvrzení v databázi s protokolováním bez vyrovnávací paměti vyprázdní vyrovnávací paměť.
- Nastane 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 12-4. Obvykle je vhodné vypnout protokolování databáze, když zavádíte data nebo pokud provádíte pouze dotazy.

Pokud vám vyhovuje zdroj, ze kterého můžete provádět zotavení, 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 paměti *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 paměti, 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 23-2.

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 13, “Správa režimu protokolování databáze”, na stránce 13-1. Běžní koncoví uživatelé nemohou měnit stav protokolování databáze.

Pokud databáze nepoužívá protokolování, nemusíte zvažovat, zda je vhodnější protokolování s vyrovnávací paměti 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 paměti (avšak ne na protokolování kompatibilní se standardem ANSI), a naopak z protokolování s vyrovnávací paměti 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 13. Správa režimu protokolování databáze

Změna režimu protokolování databáze	13-2
Změna režimu protokolování databáze pomocí obslužného programu ondblog	13-3
Změna režimu protokolování pomocí obslužného programu ondblog	13-3
Zrušení změny režimu protokolování pomocí obslužného programu ondblog	13-3
Ukončení protokolování pomocí obslužného příkazu ondblog	13-3
Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ondblog	13-4
Změna režimu protokolování databáze kompatibilní se standardem ANSI	13-4
Změna režimu protokolování databáze pomocí obslužného programu ontape	13-4
Zapnutí protokolování transakcí pomocí obslužného programu ontape	13-4
Ukončení protokolování pomocí obslužného programu ontape	13-5
Změna režimu protokolování pomocí obslužného programu ontape	13-5
Zajištění kompatibility databáze se standardem ANSI pomocí obslužného programu ontape	13-5
Změna režimu protokolování databáze pomocí programu ISA	13-6
Změna režimu protokolování pomocí programu ON-Monitor (UNIX)	13-6
Změna režimu protokolování tabulek	13-6
Změna tabulky za účelem vypnutí protokolování	13-6
Změna tabulky za účelem zapnutí protokolování	13-6
Monitorování transakcí	13-6
Monitorování režimu protokolování databáze	13-7
Monitorování režimu protokolování u tabulek typu SMI	13-7
Monitorování režimu protokolování pomocí programu ON-Monitor (UNIX)	13-7
Monitorování režimu protokolování pomocí programu ISA	13-7

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 pamětí).
- 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 12, “Protokolování”, na stránce 12-1. Informace o tom, jak zjistit aktuální režim protokolování databáze, uvádí “Monitorování režimu protokolování databáze” na stránce 13-7.

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 jednoho uživatele nebo v klidovém režimu.

Informace o programu ON-Bar a obslužném programu **ontape** naleznete v příručce *IBM Informix: Backup and Restore Guide*.

Tabulka 13-1 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 13-1. Přejechy režimu protokolování

Převést z:	Převést na režim bez protokolování	Převést na protokolování bez vyrovnávací paměti	Převést na protokolování s vyrovnávací pamětí	Převést na kompatibilní s ANSI
Režim 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 pamětí	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 zotavení 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 13-7. Potom se pokuste změnit režim protokolování znovu.
- Jakmile zvolíte protokolování s vyrovnávací pamětí nebo protokolování bez vyrovnávací paměti, může aplikace pomocí příkazu jazyka 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
```

Pamatujte, že 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 no log -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 13-2.

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.

Tip: 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.

Tip: Poté, co změníte režim protokolování na kompatibilní se standardem *ANSI*, není možné ho snadno změnit zpět. Pokud chcete změnit režim protokolování databází kompatibilních se standardem *ANSI*, musíte uvolnit data z paměti, znovu vytvořit databázi s novým režimem protokolování a znovu zavést data. Podrobnější informace naleznete v části “Změna režimu protokolování databáze kompatibilní se standardem ANSI” na stránce 13-4.

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 programu ISA a v části “Změna režimu protokolování databáze pomocí obslužného programu ondblog” na stránce 13-3.

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 10-30.

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, provedte zálohu úrovně 0, pokud musíte být schopni tuto tabulku obnovit.

Monitorování transakcí

Tato část obsahuje odkazy na možné způsoby monitorování transakcí.

Příkaz	Popis	Odkaz
onstat -x	Monitorování transakcí.	“Monitorování globální transakce” na stránce 23-15
onstat -g sql	Monitorování příkazů jazyka SQL uvedené v seznamu podle ID relace a databáze.	Monitorování výkonu v příručce <i>IBM Informix: Dynamic Server Performance Guide</i>
onstat -g stm	Monitorování využití paměti připravenými příkazy jazyka SQL.	Využití paměti v příručce <i>IBM Informix: Dynamic Server Performance Guide</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 13-7.

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 14. Logický protokol

Co je logický protokol?	14-1
Umístění souborů logického protokolu	14-2
Identifikace souborů logického protokolu	14-3
Příznaky stavu souborů logického protokolu	14-3
Velikost logického protokolu	14-4
Počet souborů logického protokolu	14-5
Pokyny týkající se výkonu	14-5
Dynamické přidělování protokolů	14-6
Uvolnění souborů logického protokolu	14-7
Akce v případě, že další soubor logického protokolu není volný	14-7
Akce v případě, že následující logický soubor obsahuje poslední kontrolní bod	14-7
Protokolování prostorů blobspace a jednoduché velké objekty	14-8
Přepínání souborů protokolu za účelem aktivace prostorů blobspace	14-8
Zálohování souborů protokolů za účelem uvolnění stránek blobpage	14-9
Zálohování prostorů blobspace po vložení nebo odstranění dat typu TEXT a BYTE	14-9
Protokolování prostorů sbpace a inteligentních velkých objektů	14-9
Používání protokolování prostoru sbpace	14-9
Kdy zvolit protokolování inteligentních velkých objektů	14-10
Protokolování aktualizovaných inteligentních velkých objektů	14-10
Zapnutí a vypnutí protokolování prostoru sbpace	14-10
Používání záznamů protokolu inteligentních velkých objektů	14-11
Zabránění dlouhým transakcím při protokolování dat inteligentních velkých objektů	14-12
Proces protokolování	14-12
Protokolování prostorů dbspace	14-12
Protokolování prostorů blobspace	14-12

Obsah kapitoly

Informace, které uvádí Kapitola 12, “Protokolování”, na stránce 12-1 a tato kapitola, vám pomohou porozumět způsobu, jakým databázový server používá logický protokol. Další informace týkající se používání úloh logického protokolu uvádí Kapitola 15, “Správa souborů logických protokolů”, na stránce 15-1 a Kapitola 13, “Správa režimu protokolování databáze”, na stránce 13-1.

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 14-2.
- Monitorování stavu souboru logického protokolu.
Další informace naleznete v části “Identifikace souborů logického protokolu” na stránce 14-3.
- Přidělení vhodného objemu diskového prostoru pro logický protokol.
Další informace naleznete v části “Velikost logického protokolu” na stránce 14-4.
- Přidělení dodatečných souborů protokolu v případě nutnosti.
Další informace naleznete v části “Přidělení souborů protokolu” na stránce 15-10.
- Zálohování souborů logického protokolu na média.
Další informace naleznete v části “Zálohování logických souborů” na stránce 15-4 a “Uvolnění souborů logického protokolu” na stránce 14-7.
- Správa protokolování prostorů blobspace a sbpace
Další informace naleznete v části “Protokolování prostorů blobspace a jednoduché velké objekty” na stránce 14-8 a “Protokolování prostorů sbpace a inteligentních velkých objektů” na stránce 14-9.

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 15-16.

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 pásce.

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 14-1 ilustruje vztah mezi čísly protokolu a jedinečnými identifikačními čísly. Protokol 7 je vložen za protokol 5 a používá se poprvé v druhém oběhu.

Tabulka 14-1. 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 14-2 uvádí možné kombinace příznaků stavů protokolu.

Tabulka 14-2. 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 zotavení. (Soubor protokolu je uvolněn v případě, že již pro zotavení není dále nutný).
U-B---L	Protokol je zálohovaný, ale stále potřebný pro zotavení. 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 15-8.

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 zotavení 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 logické zotavení.
- Používejte větší soubory protokolu, pokud do protokolů zapisuje zároveň více uživatelů.

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 blobpace. Další informace naleznete v části “Zálohování souborů protokolů za účelem uvolnění stránek blobpage” na stránce 14-9.

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*.)

- Kontrolní body

Kontrolní body značně blokují uživatelské zpracování. Pokud jsou soubory protokolu zálohovány a uvolňovány častěji, objevují se také kontrolní body častěji.

- 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 15-12.

- 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 запиše 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 15-10.

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. Předtím, 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.

Zadejte příkaz **onmode -c**, aby se provedl úplný kontrolní bod a soubor logického protokolu se uvolnil. Další informace naleznete v části “Vynucení úplného kontrolního bodu” na stránce 17-6.

- Soubor obsahuje otevřenou transakci.

Otevřená transakce je dlouhá transakce, popisovaná v části “Nastavení horních mezí pro odvolání dlouhých transakcí” na stránce 15-20.

- 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 úplný kontrolní bod, pokud se záznam předchozího kontrolního bodu nebo nejstarší aktualizovaný záznam dosud nevyprázdňeny 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 *IBM Informix: Dynamic Server Performance Guide*.

Přepínání souborů protokolu za účelem aktivace prostorů blobspace

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 15-5.

Zálohování souborů protokolů za účelem uvolnění stránek blobpage

Když odstraníte data uložena 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 10-16, 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 sbospace

Pokud je protokolován prostor sbospace, databázový server se zpomalí a logické protokoly budou rychle zaplňovány. Pokud chcete používat protokolování prostorů sbospace, 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 15-3.

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 zotavení inteligentních velkých objektů.

Další informace naleznete v části “Zálohování prostorů sbspace” na stránce 15-4 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 zotavení 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 nepříliš často nebo pokud zotavení dat není životně důležité.

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 **onspaces** 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
onspaces -ch -Df LOGGING=ON	Zapne nebo vypne protokolování existujícího prostoru sbspace.	“Změna paměťových charakteristik inteligentních velkých objektů” na stránce 11-30 <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 10-22 <i>IBM Informix: Guide to SQL Syntax</i>
Funkce mi_lo_create rozhraní DataBlade API	Vypne protokolování inteligentního velkého objektu při jeho počátečním zavedení.	<i>IBM Informix: DataBlade API Function Reference</i>
Funkce mi_lo_alter rozhraní DataBlade API	Zapne protokolování po dokončení zavedení.	<i>IBM Informix: DataBlade API Function Reference</i>
Funkce ifx_lo_create jazyka ESQ/C	Vypne protokolování inteligentního velkého objektu při jeho počátečním zavedení.	<i>IBM Informix: ESQ/C Programmer's Manual</i>
Funkce ifx_lo_alter jazyka ESQ/C	Zapne protokolování po dokončení zavedení.	<i>IBM Informix: ESQ/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.

Tip: 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 запиše 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 запиše ji zpět na disk.

Protokolování prostorů blobspace

Databázový server protokoluje data prostoru blobspace, data ovšem neprochází 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 15. Správa souborů logických protokolů

Odhad velikosti a počtu souborů protokolu	15-2
Odhad velikosti protokolu při protokolování inteligentních velkých objektů	15-3
Odhad počtu souborů logického protokolu.	15-3
Zálohování logických souborů	15-4
Zálohování prostorů blobspace	15-4
Zálohování prostorů sbspace.	15-4
Přepnutí na následující soubor logického protokolu	15-5
Uvolnění souboru logického protokolu.	15-5
Odstranění souboru protokolu se stavem D	15-6
Uvolnění souboru protokolu se stavem U	15-6
Uvolnění souboru protokolu se stavem U-B nebo F	15-6
Uvolnění souboru protokolu se stavem U-C nebo U-C-L	15-7
Uvolnění souboru protokolu se stavem U-B-L	15-7
Monitorování aktivity protokolování	15-8
Monitorování logického protokolu za účelem zjištění jeho zaplnění	15-8
onstat -l	15-8
oncheck -pr	15-9
Monitorování dočasných logických protokolů.	15-9
Používání tabulek typu SMI	15-10
Použití programu ON-Monitor (operační systém UNIX)	15-10
Monitorování stavu zálohy protokolů	15-10
Přidělení souborů protokolu	15-10
Dynamické přidávání protokolů	15-10
Velikost a počet dynamicky přidávaných souborů protokolu	15-11
Umístění dynamicky připojovaných souborů protokolu	15-12
Ruční přidání souborů logického protokolu	15-12
Vypuštění souborů logického protokolu	15-14
Změna velikosti souborů logického protokolu	15-15
Přesun souboru logického protokolu do jiného prostoru dbspace	15-16
Změna konfiguračních parametrů protokolování	15-17
Použití programu ON-Monitor ke změně parametru LOGFILES (operační systém UNIX)	15-18
Zobrazení záznamů logického protokolu	15-18
Monitorování událostí dynamického přidávání protokolů	15-18
Nastavení horních mezí pro odvolání dlouhých transakcí	15-20
Horní mez dlouhé transakce (LTXHWM)	15-20
Horní mez dlouhé transakce s vylučným přístupem (LTXEHWL)	15-21
Úprava velikosti souborů protokolu za účelem zabránění dlouhých transakcí	15-21
Zotavení ze zablokované dlouhé transakce	15-21

Obsah kapitoly

Soubory logických protokolů musíte spravovat i v případě, že žádná z databází nepoužívá protokolování transakcí. Základní informace o logických protokolech uvádí Kapitola 14, “Logický protokol”, na stránce 14-1.

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 15-2.
- Pokud nechcete používat výchozí hodnoty, změňte konfigurační parametry LOGSIZE a LOGBUFF.
Další informace naleznete v části “Změna konfiguračních parametrů protokolování” na stránce 15-17.
- Přidejte odhadovaný počet souborů logického protokolu.
Další informace naleznete v části “Přidělení souborů protokolu” na stránce 15-10.

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.

Odhad velikosti a počtu souborů protokolu

Velikost a počet souborů protokolu nastavíte pomocí konfiguračního parametru LOGSIZE. Je velmi obtížné předvídat, kolik prostoru logického protokolu bude systém databázového serveru vyžadovat, dokud nebude v plném provozu.

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 15-12.

Následující výraz poskytuje doporučenou *minimální* konfiguraci celkového prostoru protokolu v kilobajtech:

$$\text{LOGSIZE} = ((\text{připojení} * \text{max_počet_řádků}) * \text{velikost_řádku}) / 1024) / \text{LOGFILES}$$

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_pocet_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 sbspace, budou vždy protokolována metadata, i když inteligentní velké objekty protokolovány nejsou. Jestliže ukládáte inteligentní velké objekty do dočasných prostorů sbspace, neprovádí se žádné protokolování.

Odhad počtu souborů logického protokolu

Parametr LOGFILES poskytuje počet souborů logického protokolu při inicializaci systému. Pokud mají všechny soubory logického protokolu stejnou velikost, můžete vypočítat celkový prostor přidělený souborům logického protokolu následujícím způsobem:

$$\text{celkový prostor logického protokolu} = \text{LOGFILES} * \text{LOGSIZE}$$

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 15-8.

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 zotavení 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.

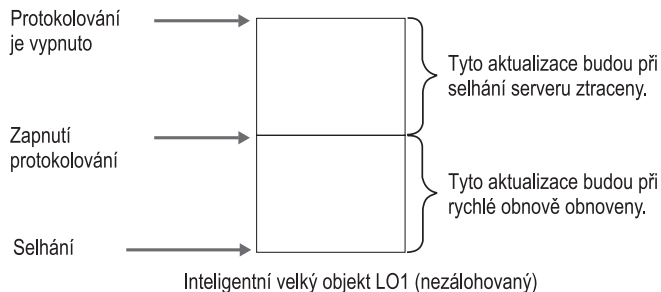
Upozornění: Pokud neprovedete zálohu těchto prostorů blobspace a logických protokolů, nebudete možná schopni obnovit data v prostoru blobspace. Jestliže budete se zálohováním čekat až na okamžik, kdy bude prostor blobspace poškozen, nebude moci databázový server získat přístup k prostoru blobspace a 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 15-1 znázorňuje, co se stane, pokud zapnete protokolování prostoru sbspace 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 16-14.



Obrázek 15-1. Zapnutí zálohování prostoru sbspace

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, spusťte 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 14-3 a “Monitorování aktivity protokolování” na stránce 15-8.

Tip: 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**), změni 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 15-7.

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 “Okamžitě z režimu online do klidového” na stránce 4-14. Všechny aktivní transakce budou odvolány.
2. Jelikož soubor protokolu se stavem **U-B** může obsahovat nejstarší aktualizaci, musíte pomocí příkazu **onmode -c** vynutit úplný kontrolní bod.

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.

Tip: 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 15-6 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

Pokud chcete vynutit kontrolní bod pomocí programu ON-Monitor, vyberte volbu **Force-Ckpt**.

Monitorování aktivity protokolování

Pomocí monitorování souborů logického protokolu můžete určit celkový dostupný prostor (ve všech souborech) a stav souboru (například, zda byl souboru protokolu již zálohován). Další informace týkající se monitorování vyrovnávacích pamětí logického protokolu naleznete v části “Monitorování aktivity fyzického a logického protokolování” na stránce 17-4.

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

Obrazovka obslužného programu **onstat -l** se skládá z následujících tří částí: informace fyzického protokolu, informace logického protokolu (všeobecné) a informace o jednotlivých souborech logického protokolu.

Části obsahující informace o každém souboru logického protokolu zobrazují 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.

Pokud chcete získat výstup, který zobrazuje Obrázek 15-2, postupujte takto:

- Pomocí příkazu **onparams -a -i** vložte protokol 7 za protokol 1.
- Pomocí příkazu **onmode -l** učinite z protokolu 7 aktuální soubor protokolu.
- Pomocí příkazu **onparams -a -i** vložte protokol 8 za protokol 7.

Číslo souborů protokolu (pole ve sloupci **number**) nemusí jít za sebou, pokud vypustíte některé protokoly z prostřední části seznamu nebo pokud databázový server dynamicky

přidá soubory protokolu. Další informace o příkazu **onstat -l** naleznete v kapitole týkající se obslužných programů v příručce *IBM Informix: Administrator's Reference*.

```
...
address  number  flags   uniqid  begin    size    used    %used
a32eaf0  1        U-B---- 7       1004ef   750     750     100.00
a32eb30  7        U---C-L 8       1027d2   750     464     61.87
a328a88  8        A----- 0       102d72   750     0       0.00
a337718  2        U-B---- 2       1007dd   750     750     100.00
a337750  3        U-B---- 3       100acb   750     750     100.00
a337788  4        U-B---- 4       1007dd   750     750     100.00
a3377c0  5        U-B---- 5       1010a7   750     750     100.00
a3377f8  6        U-B---- 6       101395   750     643     85.73
...
```

Obrázek 15-2. Výstup příkazu `onstat -l` zobrazující stav souborů logického protokolu

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 kapitole týkající se obslužných programů v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Rezervované stránky kontrolního bodu je možné zobrazit pomocí příkazu **oncheck -pr**. Obrázek 15-3 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 15-3. Výstup příkazu `oncheck -pr` obsahující informace o souboru logického protokolu

Monitorování dočasných logických protokolů

Databázový server používá *dočasné logické protokoly* k přehraní transakcí během teplého obnovení, protože trvalé protokoly nejsou v tomto okamžiku dostupné. Když se přehraní 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**.

Používání tabulek typu 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 (operační 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 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 logickém zotavení.

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 15-18 a v části “Nastavení horních mezí pro odvolání dlouhých transakcí” na stránce 15-20.

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 15-18.

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.

Postup přidání souboru logického protokolu pomocí příkazu **onparams**:

1. Přihlaste se do operačního systému UNIX jako uživatel **informix** nebo **root** nebo se přihlaste do operačního systému Windows jako člen skupiny **Informix-Admin**.
2. Zajistěte, aby byl databázový server online, v režimu jednoho uživatele nebo v klidovém režimu, případně 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 **onparams** k přidání souboru logického protokolu naleznete v kapitole týkající se obslužných programů v příručce *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 (operační systém UNIX):

1. Řiďte se pokyny pro přidání souboru protokolu v části “Ruční přidání souborů logického protokolu” na stránce 15-12 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 **Dbospace Name** zadejte název prostoru dbospace, 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 dbospace, 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 dbospace, které obsahují soubory protokolu, a kořenový prostor dbospace, 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 online, v režimu jednoho uživatele nebo v klidovém režimu.
2. Pomocí následujícího příkazu vypustíte soubor logického protokolu, jehož číslo souboru protokolu je 21:

```
onparams -d -l 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 zotavení 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 týkající se obslužných programů v příručce *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 15-18.

Postup vypuštění souboru logického protokolu pomocí monitoru ON_Monitor (operační systém UNIX):

1. Zajistěte, aby byl databázový server online, v režimu jednoho uživatele 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.

Tip: 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 velikostí 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 15-12.
- Zvětšíte hodnotu LOGSIZE v souboru ONCONFIG, pokud chcete, aby databá zový server vytvořil větší soubory protokolu.
Další informace naleznete v části “Změna konfiguračních parametrů protokolování” na stránce 15-17.

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 14-2. Další informace užitečné při hledání umístění souborů logického protokolu naleznete v části “Monitorování aktivity protokolování” na stránce 15-8. 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 jednoho uživatele nebo 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 15-12.
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 15-5.
4. Pomocí příkazu **onmode -l** přepněte na nový aktuální soubor protokolu.
Další informace naleznete v části “Přepnutí na následující soubor logického protokolu” na stránce 15-5.
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 15-14.
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 naleznete v kapitole zaměřené na konfigurační parametry 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	32 KB	Hodnota parametru LOGSIZE
LOGFILES	3 soubory	6 souborů	32 767 souborů
LOGSIZE	1500 KB v operačním 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 neinicilizujete diskový prostor.

Postup změny konfiguračních parametrů logického protokolu v souboru ONCONFIG:

1. Uvedte databázový server do režimu offline nebo do klidového režimu nebo režimu jednoho uživatele.
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. 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.
 - b. Znovu inicializujte prostor na disku databázového serveru.
Další informace naleznete v části "Inicilizace 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 (operační 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.

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**. Další informace o používání obslužného programu **onlog** naleznete v kapitole týkající se obslužných programů 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 15-1). Když je alarm spuštěn, запиše 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 provádět příkaz **onstat -d**, aby vyhledal odpovídající prostor a provedl příkaz

onparams -i s umístěním, ve kterém se nachází dostatek prostoru. Chcete-li přidat nový protokol přímo za aktuální soubor protokolu, musíte použít volbu **-i**.

Tabulka 15-1. 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 (vložení) 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>transakční_adresy</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 15-2 obsahuje akce, které databázový server podnikne při jednotlivých nastaveních konfiguračního parametru DYNAMIC_LOGS.

Tabulka 15-2. 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	Nepřiděluje 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> .	Ne	Ne	Ne

Nastavení horních mezí pro odvolání dlouhých transakcí

Databázový server používá konfigurační parametry LTXHWM a LTXEHWK 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 LTXEHWK 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 vynahradit přidělením dalšího prostoru protokolu. Další informace o konfiguračních parametrech LTXHWM a LTXEHWK, 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 přerušeny. Pro běžný provoz databázového serveru se doporučuje 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 vyčerpat 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. Podrobnější informace naleznete v části “Zotavení ze zablokované dlouhé transakce” na stránce 15-21.

Zotavení ze 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 přerušte.

Jak přerušit 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 zotavení ze zablokování dlouhou transakcí:

1. Přidejte více prostoru na disku nebo další disk, dokud se transakce úspěšně neodvolá.
2. Proveďte zotavení do bodu v čase před zahájením dlouhé transakce nebo dostatečně brzy na to, aby databázový server mohl transakci odvolat.
3. Z instance databázového serveru vypusíte přebytečné soubory protokolu, prostory dbspace a bloky.
4. Proveďte zálohu úrovně 0, abyste uvolnili prostor logického protokolu.

Kapitola 16. Fyzické protokolování, kontrolní body a rychlá obnova

Kritická část	16-2
Fyzické protokolování	16-2
Jak rychlá obnova používá fyzicky protokolované stránky	16-3
Jak zálohování používá fyzicky protokolované stránky	16-3
Aktivity databázového serveru, které jsou fyzicky protokolovány	16-3
Zprávy fyzického zotavení	16-4
Fyzické protokolování a jednoduché velké objekty	16-4
Fyzické protokolování a inteligentní velké objekty	16-4
Velikost a umístění fyzického protokolu	16-4
Určení umístění fyzického protokolu	16-4
Odhad velikosti fyzického protokolu	16-5
Konfigurace velikosti fyzického protokolu	16-6
Přetečení fyzického protokolu v případě, že se mnoho uživatelů nachází v kritických částech	16-6
Vliv kontrolních bodů na velikost fyzického protokolu	16-6
Přetečení fyzického protokolu po vypnutí protokolování transakcí	16-7
Podrobnosti fyzického protokolování	16-7
Kontrolní body	16-8
Úplný kontrolní bod	16-9
Kontrolní bod typu fuzzy	16-9
Operace typu fuzzy	16-9
Předběžné protokolování a rychlá obnova	16-10
Zlepšení výkonu pomocí kontrolních bodů typu fuzzy	16-10
Údlosti, které vyvolají kontrolní bod typu fuzzy	16-11
Údlosti, které vyvolají úplný kontrolní bod	16-11
Pořadí událostí v kontrolním bodu	16-12
Uživatelské jednotkové procesy nemohou vstoupit do kritické části	16-12
Vyrovnávací paměť logického protokolu je vyprázdněna do souboru logického protokolu na disk	16-12
Vyrovnávací paměť fyzického protokolu je vyprázdněna do fyzického protokolu na disk	16-12
Změněné stránky ve společné oblasti vyrovnávacích pamětí jsou vyprázdněny na disk	16-13
Jednotkový proces kontrolního bodu zapíše záznam kontrolního bodu	16-13
Fyzický protokol je logicky vyprázdněn	16-13
Pokyny týkající se zálohování a obnovy	16-13
Rychlá obnova	16-14
Potřeba rychlé obnovy	16-14
Situace, při kterých se spouští rychlá obnova	16-14
Rychlá obnova a protokolování s vyrovnávací pamětí	16-15
Možné přetečení fyzického protokolu během rychlé obnovy	16-15
Rychlá obnova a žádné protokolování	16-15
Podrobnosti rychlé obnovy po úplném kontrolním bodu	16-15
Návrat na stav posledního kontrolního bodu	16-16
Nalezení záznamu kontrolního bodu v logickém protokolu	16-16
Přehrání záznamů logického protokolu	16-16
Odvolení nedokončených transakcí	16-17

Podrobnosti rychlé obnovy po kontrolním bodu typu fuzzy	16-17
Návrat do stavu posledního kontrolního bodu pro operace jiného typu než fuzzy	16-18
Vyhledání nejstarší aktualizace v logickém protokolu	16-19
Použití záznamů protokolu pro operace typu fuzzy	16-20
Přehrání záznamů logického protokolu	16-20
Odvolení nedokončených transakcí	16-21
Alternativní volby obnovy při rychlém restartování pro operace typu fuzzy	16-21

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 disku, na kterých 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 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 disku.

Další informace o úlohách správy a monitorování kontrolních bodů fyzických protokolů uvádí Kapitola 17, “Správa fyzického protokolu”, na stránce 17-1.

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 paměti 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 nevyprázdni 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.

Databázový server vyprázdni fyzický protokol v každém kontrolním bodu (kromě výjimečných okolností, které objasňuje část “Konfigurace velikosti fyzického protokolu” na stránce 16-6). Další informace o kontrolních bodech naleznete v části “Kontrolní body” na stránce 16-8.

Jak rychlá obnova používá fyzicky protokolované stránky

Po selhání použije databázový server předobrazy stránek změněných operacemi jiného typu než fuzzy k obnovení těchto stránek na disku na jejich stav v 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 16-14.

Jak zálohování používá fyzicky protokolované stránky

Když provádíte zálohování, databázový server vytváří kontrolní bod a používá fyzický protokol k nalezení změněných stránek. V případě zálohy úrovně 0 zálohuje databázový server všechny stránky na disku. V případě zálohy úrovně 1 nebo 2 zálohuje databázový server pouze změněné stránky. Podrobnější informace naleznete v příručce *IBM Informix: Backup and Restore Guide*.

Aktivity databázového serveru, které jsou fyzicky protokolovány

V případě provedení více změn před následujícím kontrolním bodem se protokoluje do fyzického protokolu pouze první předobraz.

Databázový server ukládá předobraz do fyzického protokolu pouze do následujícího kontrolního bodu. Chcete-li řídit objem dat, které protokoluje databázový server, můžete vyladit konfigurační parametr intervalu kontrolního bodu CKPTINTVL.

Níže uvedené změny stránek prostoru dbospace *nejsou* fyzicky protokolovány:

- Stránky, které nemají platnou adresu databázového serveru.
K této situaci obvykle dochází, když byla stránka používána jiným databázovým serverem nebo tabulkou, která byla vypuštěna.
- Stránky, které databázový server nepoužíval a které jsou umístěny v prostoru dbospace, kde nebyla od posledního kontrolního bodu vypuštěna žádná tabulka.
- Stránky pro operace typu fuzzy.

Důležité: Databázový server již do fyzického protokolu neprotokoluje předobrazy pro operace typu fuzzy. Tyto aktualizace jsou stále zaznamenávány do logického protokolu. Definici operací typu fuzzy naleznete v části “Operace typu fuzzy” na stránce 16-9.

Zprávy fyzického zotavení

Po zahájení rychlé obnovy zapíše databázový server do protokolu následující zprávu:

Physical recovery started at page *blok:posun*.

Po dokončení rychlé obnovy zapíše databázový server do protokolu následující zprávu:

Physical recovery complete: *počet pages examined*, *počet pages restored*.

Pokud je hodnota *number of pages examined* (počet přezkoušených stránek) mnohem větší než hodnota *number of pages restored* (počet obnovených stránek), zvětšete velikost společné oblasti vyrovnávacích pamětí, aby se snížil počet duplikovaných předobrazů. Další informace naleznete v dodatku v příručce *IBM Informix: Administrator's Reference*.

Fyzické protokolování a jednoduché velké objekty

Stránky databázového serveru ve fyzickém protokolu mohou být libovolné stránky databázového serveru, včetně jednoduchých velkých objektů v prostorech *tblspace*. Dokonce režijní stránky (jako například stránky seznamu volných stránek v bloku) jsou kopírovány do fyzického protokolu, než jsou data na stránce modifikována a vyprázdněna na disk.

Režijní stránky také zahrnují stránky mapy volných stránek a bitové mapy prostoru *blobspace*. Stránky *blobpage* prostoru *blobspace* nejsou protokolovány do fyzického protokolu. Další informace o protokolování prostorů *blobspace* naleznete v části “Protokolování prostorů *blobspace* a jednoduché velké objekty” na stránce 14-8.

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 *sbspace*” na stránce 10-16.

Velikost a umístění fyzického protokolu

Tato část popisuje, jak konfigurovat velikost a umístění fyzického protokolu.

Určení umístění fyzického protokolu

Když databázový server inicializuje prostor na disku, umístí soubory logického protokolu a fyzický protokol do kořenového prostoru *dbspace*. Nad tímto počátečním umístěním nemáte žádnou kontrolu. Chcete-li zlepšit výkon (zvláště snížit počet zápisu do kořenového prostoru *dbspace* a minimalizovat kolize požadavků na disky) můžete přesunout fyzický protokol do jiného prostoru *dbspace*, nejlépe na disk, který neobsahuje aktivní tabulky ani soubory logického protokolu.

Fyzický protokol je umístěn v prostoru *dbspace* určeném pomocí parametru *PHYSDBS* souboru *ONCONFIG*. (Další informace o parametru *PHYSDBS* naleznete v kapitole zaměřené na konfigurační parametry v příručce *IBM Informix: Administrator's Reference*.) Parametr *PHYSDBS* změňte pouze tehdy, pokud chcete přesunout fyzický

protokol z kořenového prostoru dbspace. (Další informace naleznete v části “Změna umístění a velikosti fyzického protokolu” na stránce 17-1.)

Protože je fyzický protokol životně důležitý, doporučujeme, abyste prováděli zrcadlení prostoru dbspace, ve kterém je fyzický protokol obsažen.

Odhad velikosti fyzického protokolu

Pokud jsou součástí zatížení databázového serveru intenzivní aktualizace, můžete vypočítat velikost fyzického protokolu pomocí následujícího vzorce, kde hodnota PHYSFILE se bude rovnat velikosti fyzického protokolu.

PHYSFILE =
(*uživatelé * max_stránek_protokolu_na_kritickou_sekci * 4 * velikost_stránky*) / 1024

Proměnné ve vzorci

Popis

uživatelé

Maximální počet souběžných uživatelských jednotkových procesů, jehož odhad získáte spuštěním příkazu **onstat -u** během špičkového zatížení. Poslední řádek výstupu příkazu **onstat -u** obsahuje maximální počet souběžných uživatelských jednotkových procesů.

Pokud nastavíte parametr NETTYPE, sečtěte hodnoty zadané v poli *users* pro každý parametr NETTYPE v souboru ONCONFIG.

max_stránek_protokolu_na_kritickou_sekci

Maximální počet stránek, které databázový server může fyzicky protokolovat během kritické sekce. Doplňte jednu z níže uvedených hodnot:

5, pokud nepoužíváte indexy R-stromu

10, pokud používáte indexy R-stromu

4

Nezbytný činitel, neboť následující část vzorce představuje pouze 25 procent fyzického protokolu:

*uživatelé *
max_stránek_protokolu_na_kritickou_sekci*

pagesize

Velikost systémové stránky v bajtech, kterou můžete získat spuštěním příkazu **oncheck -pr**

1024

Nezbytný dělitel, jelikož parametr PHYSFILE zadáváte v kilobajtech

Tento vzorec je založen na tom, kolik prostoru pro fyzické protokolování databázový server potřebuje v nejhorším případě. Tento scénář nastává, pokud se vyskytne kontrolní bod a protokol je z 75 procent zaplněn. Vypočtená velikost může být příliš malá nebo velká pro skutečné pracovní zatížení nebo konfiguraci.

Pokud používáte jednoduché velké objekty v prostoru dbspace v databázi bez protokolování, nahraďte hodnotu max_stránek_protokolu_na_kritickou_sekci velikostí nejčastěji se vyskytujícího jednoduchého velkého objektu.

Další informace o monitorování a ladění fyzického protokolu naleznete v kapitole zaměřené na vlivy konfigurace na využití vstupu - výstupu v příručce *IBM Informix: Dynamic Server Performance Guide*.

Konfigurace velikosti fyzického protokolu

Jelikož kontrolní bod logicky vyprázdní fyzický protokol, když je ze 75 procent zaplněný, není pravděpodobné, že by se protokol zaplnil ze 100 procent předtím, než se dokončí kontrolní bod. Pokud chcete ještě více zajistit, aby se fyzický protokol nezaplnil během kontrolního bodu, proveďte následující činnosti:

- Nakonfigurujte databázový server v souladu s pokyny pro velikost fyzického protokolu a souborů logického protokolu.
- Pečlivě vyladte velikost fyzického protokolu jeho monitorováním během provozní aktivity.

Kontrolní body typu fuzzy zabráňují fyzickému protokolu v příliš rychlém zaplnění, pokud aplikace provádějí intenzivní aktualizace. (Další informace naleznete v části “Kontrolní bod typu fuzzy” na stránce 16-9.) Fyzický protokol se ovšem přesto může zcela zaplnit, jak je popsáno v následujících částech.

Přetečení fyzického protokolu v případě, že se mnoho uživatelů nachází v kritických částech

Kontrolní bod nemůže nastat, pokud se nějaký jednotkový proces nachází v kritické části. (Další informace naleznete v části “Kritická část” na stránce 16-2.) K tomuto scénáři může dojít, když jednotkový proces, který vytváří kontrolní bod, také odvolává dlouhou transakci.

Vliv kontrolních bodů na velikost fyzického protokolu

Kontrolní body typu fuzzy zabráňují fyzickému protokolu v příliš rychlém zaplnění, pokud aplikace provádějí intenzivní aktualizace. Pokud aplikace vyžadují méně intenzivní aktualizace nebo pokud jsou aktualizace seskupeny na týchž stránkách, můžete snížit velikost fyzického protokolu. Rovněž pokud zamýšlíte používat zaplnění fyzického protokolu ke spouštění kontrolních bodů, můžete snížit velikost fyzického protokolu.

Pokud zvětšíte interval mezi kontrolními body nebo očekáváte zvýšenou aktivitu, zvažte zvětšení velikosti fyzického protokolu. Další informace naleznete v kapitole týkající se vlivu konfigurace na aktivitu vstupu - výstupu v příručce *IBM Informix: Dynamic Server Performance Guide*.

Přetečení fyzického protokolu po vypnutí protokolování transakcí

Fyzický protokol může přetéct, pokud používáte 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 tyto jednoduché velké objekty, může být každá část jednoduchého velkého objektu, kterou databázový server ukládá na disk, protokolována samostatně, což umožňuje jednotkovému procesu vystoupit z kritických částí kódu mezi každou částí jednoduchého velkého objektu. Pokud je ovšem protokolování vypnuto, musí databázový server provádět všechny operace na jednoduchých velkých objektech v jedné kritické části. Jestliže je jednoduchý velký objekt skutečně velký a fyzický protokol malý, může v tomto scénáři dojít k zaplnění fyzického protokolu. Jestliže tato situace nastane, odešle databázový server do protokolu zpráv následující zprávu:

```
Physical log file overflow
```

Databázový server se potom vypne. Navrhované nápravné akce naleznete v této zprávě zapsané do protokolu zpráv.

Podrobnosti fyzického protokolování

Tato část podrobně popisuje fyzické protokolování. Tato část má pouze informativní charakter. Abyste byli schopni spravovat fyzický protokol, není nezbytně nutné, abyste zde uváděným informacím porozuměli.

Databázový server provádí fyzické protokolování v šesti krocích, které jsou uvedeny v následující tabulce.

Kroky fyzického protokolování prováděné databázovým serverem	Podrobný popis
Databázový server přečte datovou stránku z disku do vyrovnávací paměti stránky sdílené paměti (pokud zde již datová stránka není).	Když relace vyžaduje řádek, databázový server identifikuje stránku, na které je uložen tento řádek a pokusí se vyhledat stránku ve společné oblasti vyrovnávacích pamětí sdílené paměti databázového serveru. Jestliže stránka dosud není ve sdílené paměti, je načtena z disku do rezidentní části sdílené paměti databázového serveru.
Databázový server zkopíruje nezměněnou stránku do vyrovnávací paměti fyzického protokolu.	Pokud je předobraz změněné stránky uložen ve vyrovnávací paměti fyzického protokolu, bude nakonec vyprázdněna z vyrovnávací paměti fyzického protokolu do fyzického protokolu na disk. Předobraz stránky hraje důležitou úlohu při obnovení dat a rychlé obnově. Další informace naleznete v části "Vyrovnávací paměť fyzického protokolu" na stránce 8-16.

Kroky fyzického protokolování prováděné databázovým serverem	Podrobný popis
Databázový server zanesle změny do vyrovnávací paměti stránky, jakmile aplikace změnila data.	Databázový server zanesle změny do dat ve vyrovnávací paměti ve sdílené paměti. Data z aplikace jsou předávána do databázového serveru. Poté, co je kopie nezměněné datové stránky uložena ve vyrovnávací paměti fyzického protokolu, jsou nová data zapsána do již získané vyrovnávací paměti stránky.
Databázový server vyprázdní vyrovnávací paměť fyzického protokolu do fyzického protokolu na disk.	Databázový server vyprázdní vyrovnávací paměť fyzického protokolu předtím, než vyprázdní datovou vyrovnávací paměť, aby zajistil, že bude nezměněná stránka dostupná, dokud nebude změněná stránka zkopírována na disk. Po dosažení kontrolního bodu již není předobraz stránky potřebný. Další informace naleznete v části "Vyprazdňování vyrovnávací paměti fyzického protokolu" na stránce 8-35.
Databázový server vyprázdní vyrovnávací paměť stránky a zapíše ji zpátky na disk.	Po vyprázdnění vyrovnávací paměti fyzického protokolu je na disk vyprázdněna vyrovnávací paměť ve sdílené paměti (například během kontrolního bodu) a datová stránka je zapsána na disk. Během kontrolního bodu typu fuzzy jsou vyprazdňovány na disk pouze stránky jiného typu než fuzzy. Informace o podmínkách, které vedou k vyprázdnění vyrovnávací paměti stránky naleznete v části "Vyprazdňování dat na disk" na stránce 8-34.
Když nastane kontrolní bod, vyprázdní databázový server vyrovnávací paměť fyzického protokolu do fyzického protokolu na disku a pak vyprázdní fyzický protokol.	<p>Kontrolní bod může nastat v jakémkoli bodu procesu fyzického protokolování. Databázový server provádí dva typy kontrolních bodů: <i>úplný</i> a <i>fuzzy</i>. Další informace naleznete v části "Kontrolní body".</p> <p>Databázový server spravuje fyzický protokol jako cyklický soubor, ve kterém se neustále přepisují nepotřebná data. Procedura kontrolního bodu uvolní fyzický protokol tak, že opětovně nastaví ukazatel ve fyzickém protokolu, který ukazuje na začátek následující skupiny požadovaných předobrazů.</p>

Kontrolní body

Databázový server provádí dva typy kontrolních bodů: úplné kontrolní body (také nazývané jako *synchronizační* kontrolní body) a kontrolní body typu fuzzy. Termín *kontrolní bod* se označuje bod v provozu databázového serveru, kdy jsou stránky na disku synchronizovány se stránkami ve vyrovnávací paměti sdílené paměti. Výchozím typem kontrolního bodu je typ fuzzy.

Databázový server generuje alespoň jeden kontrolní bod pro každý rozsah prostoru logického protokolu, aby zaručil, že bude mít k dispozici kontrolní bod, od kterého může začít rychlá obnova.

Ačkoli databázový server provádí kontrolní body automaticky, můžete ručně zahájit vytvoření kontrolního bodu nebo můžete řídit, jak často databázový server kontroluje, zda je potřeba kontrolní bod. Interval kontrolních bodů můžete zadat v konfiguračním parametru CKPTINTVL.

Pokud chcete snížit objem práce požadované při kontrolním bodu, použijte konfigurační parametr BUFFERPOOL ke snížení hodnot `lru_max_dirty` a `lru_max_dirty`. Pokud má databázový server například několik instancí s velmi velkou společnou oblastí vyrovnávací paměti, nastavte hodnotu `lru_max_dirty` na méně než 1, abyste snížili dobu požadovanou pro kontrolní bod.

Poznámka: Informace, které byly před verzí 10.0 zadávány pomocí konfiguračních parametrů BUFFERS, LRUS, LRU_MAX_DIRTY a LRU_MIN_DIRTY se nyní zadávají pomocí konfiguračního parametru BUFFERPOOL.

Další informace o konfiguračních parametrech CKPTINTVL a BUFFERPOOL naleznete v kapitole zabývající se konfiguračními parametry v příručce *IBM Informix: Administrator's Reference*. Další informace o parametrech monitorování a ladění kontrolních bodů naleznete v příručce *IBM Informix: Dynamic Server Performance Guide*.

Úplný kontrolní bod

Při *úplném kontrolním bodu* vyprazdňuje databázový server všechny změněné stránky ze společné oblasti vyrovnávacích pamětí sdílené paměti na disk. Po dokončení úplného kontrolního bodu jsou všechny fyzické operace dokončené, fronta MLRU je prázdná a tento stav databázového serveru je označován jako fyzicky konzistentní.

Kontrolní bod typu fuzzy

Při *kontrolním bodu typu fuzzy* databázový server nevyprazdňuje změněné stránky ve společné oblasti vyrovnávacích pamětí sdílené paměti pro určité typy operací, nazvané *operace typu fuzzy*. Po dokončení kontrolního bodu typu fuzzy nemusí být stránky navzájem konzistentní, jelikož databázový server nevyprázdnil všechny datové stránky na disk. Kontrolní bod typu fuzzy se dokončí mnohem rychleji než úplný kontrolní bod a snižuje objem fyzického protokolování při vysoké aktivitě aktualizací. Je-li to nutné, provede databázový server úplný kontrolní bod, aby zajistil fyzickou konzistenci všech dat na disku.

Operace typu fuzzy

Následující operace pro vestavěné datové typy jsou označovány jako operace typu *fuzzy*:

- vložení,
- aktualizace,
- odstranění.

Následující uvedené operace *nejsou typu fuzzy*:

- Vložení, aktualizace a odstranění týkající se řádků obsahujících datové typy definované uživatelem, inteligentní velké objekty (datové typy CLOB a BLOB) nebo jednoduché velké objekty (datové typy TEXT a BYTE).
- Změny a zavádění tabulek.
- Operace vytvářející nebo měnící indexy (B-stromy, R-stromy nebo indexy definované uživatelem).

Databázový server vyprázdní na disk *všechny* datové stránky změněné operacemi jiného typu než fuzzy během kontrolního bodu typu fuzzy stejným způsobem jako při úplném kontrolním bodu.

Důležité: V primárních a sekundárních serverech páru replikace HDR (High-Availability Data Replication) jsou kontrolní body typu fuzzy zakázány.

Předběžné protokolování a rychlá obnova

Kontrolní body typu fuzzy používají pro rychlou obnovu předběžné protokolování. *Předběžné protokolování* znamená, že záznamy logického protokolu reprezentující změny v datech typu fuzzy musí být na disku uloženy dříve, než změněná data nahradí předchozí verzi dat na disku. Rychlá obnova začíná nejstarší aktualizací dosud nevyprázdněnou na disk, nikoli předchozím kontrolním bodem.

Důsledkem kontrolních bodů typu fuzzy jsou poněkud delší časy přehrání protokolu u rychlé obnovy. Databázový server příležitostně provádí úplný kontrolní bod, aby zabránil ztrátě starých záznamů logického protokolu.

Zlepšení výkonu pomocí kontrolních bodů typu fuzzy

Kontrolní body typu fuzzy jsou mnohem rychlejší než úplné kontrolní body a zlepšují průchodnost transakcí. Jelikož databázový server neprotokoluje operace typu fuzzy do fyzického protokolu, fyzický protokol se nezaplňuje tak rychle a kontrolní body se vyskytují méně často. Pokud například vkládáte a aktualizujete množství dat, kontrolní body se vyskytují méně často a jsou kratší.

Databázový server přeskočí úplný kontrolní bod, pokud jsou všechna data fyzicky konzistentní v okamžiku vypršení intervalu kontrolního bodu. Kontrolní bod typu fuzzy bude přeskočen pouze tehdy, pokud nebyly od posledního kontrolního bodu aktualizovány žádné stránky.

Chcete-li zlepšit průchodnost transakcí, zvětšete pomocí konfiguračního parametru BUFFERPOOL hodnoty `lru_max_dirty` a `lru_max_dirty`. Neměňte ovšem rozestup mezi hodnotami `lru_max_dirty` a `lru_max_dirty`.

Další informace o možnostech zlepšení výkonu kontrolních bodů naleznete v kapitole týkající se vlivu konfigurace na vstup - výstup v příručce *IBM Informix: Dynamic Server Performance Guide*.

Události, které vyvolají kontrolní bod typu fuzzy

Když databázový server provádí kontrolní bod automaticky, jedná se obvykle o kontrolní bod typu fuzzy. Kontrolní bod typu fuzzy je vyvolán jakoukoli z následujících podmínek:

- Interval kontrolních bodů určený v konfiguračním parametru CKPTINTVL vypršel a od posledního kontrolního bodu došlo nejméně k jedné změně.
- Fyzický protokol na disku se zaplnil ze 75 procent.
- Databázový server zjistil, že následující soubor logického protokolu, který se stane aktuálním, obsahuje záznam nejnovějšího kontrolního bodu.
- Byly provedeny určité administrativní úlohy, jako například přidání bloku nebo prostoru dbspace.

Události, které vyvolají úplný kontrolní bod

V následujících situacích provádí databázový server úplný kontrolní bod, aby zajistil fyzickou konzistenci všech dat na disku:

- Když vydáte příkaz **onmode -ky** k vypnutí databázového serveru.
- Když vynutíte kontrolní bod pomocí příkazu **onmode -c** nebo programu ISA.
Další informace naleznete v části “Vynucení úplného kontrolního bodu” na stránce 17-6.
- Když převedete databázový server na novější verzi nebo jej vrátíte na předchozí verzi.
- Když provedete zálohování nebo obnovu pomocí programu ON-Bar nebo obslužného programu **ontape**.

Nástroje pro zálohování provádějí úplný kontrolní bod automaticky, aby byla zajištěna fyzická konzistence všech dat před jejich zapsáním na zálohovací média.

- Na konci rychlé obnovy nebo úplného zotavení.
- Pokud databázový server hodlá přepnout na následující volný protokol a protokol, který následuje za tímto volným protokolem, obsahuje nejstarší aktualizaci.

Předpokládejme například, že soubory logického protokolu mají stav uvedený v následujícím seznamu. Databázový server vynutí úplný kontrolní bod po přepnutí na soubor logického protokolu 3, pokud bude soubor logického protokolu 4 obsahovat nejstarší aktualizaci. Úplný kontrolní bod posune nejstarší aktualizaci do souboru logického protokolu 3.

ID protokolu Stav souboru logického protokolu

1	U-B----
2	U---C--
3	F
4	U-B---L

Databázový server provede úplný kontrolní bod, aby předešel problémům s rychlou obnovou starých záznamů protokolu.

Následující části poskytují seznam situací, ve kterých byste měli vyvolat úplný kontrolní bod.

Pořadí událostí v kontrolním bodu

Následující informace vás seznámí s přehledem hlavních událostí, ke kterým dojde, pokud je požadován kontrolní bod. Jsou zde rovněž uvedeny rozdíly mezi úplným kontrolním bodem a kontrolním bodem typu fuzzy:

1. Databázový server zabrání uživatelským jednotkovým procesům ve vstupu do kritických částí.
2. Vyrovnávací paměť logického protokolu je vyprázdněna do aktuálního souboru logického protokolu na disk.
3. Databázový server vyprázdní vyrovnávací paměť fyzického protokolu do fyzického protokolu.
4. V případě kontrolního bodu typu fuzzy vyprázdní databázový server na disk stránky změněné operacemi jiného typu než fuzzy.
V případě úplného kontrolního bodu vyprázdní databázový server na disk všechny změněné stránky.
5. Databázový server zapíše záznam kontrolního bodu do vyrovnávací paměti logického protokolu.
6. Fyzický protokol je logicky vyprázdněn. (Aktuální záznamy mohou být přepsány).

Uživatelské jednotkové procesy nemohou vstoupit do kritické části

Tento krok je stejný pro úplný kontrolní bod i pro kontrolní bod typu fuzzy. Po vyžádání kontrolního bodu nemohou uživatelské jednotkové procesy vstoupit do části kódu, které jsou považované za kritické části. Uživatelské jednotkové procesy, které se nacházejí v kritické části kódu, mají povoleno pokračovat ve zpracování do konce kritické části.

Vyrovnávací paměť logického protokolu je vyprázdněna do souboru logického protokolu na disk

Tento krok je stejný pro úplný kontrolní bod i pro kontrolní bod typu fuzzy. Dále je vyrovnávací paměť logického protokolu vyprázdněna do souboru logického protokolu na disk.

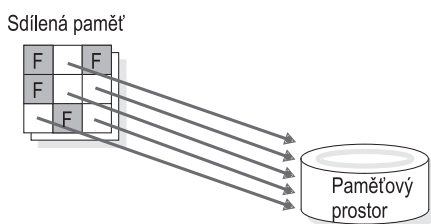
Vyrovnávací paměť fyzického protokolu je vyprázdněna do fyzického protokolu na disk

V případě kontrolního bodu typu fuzzy i úplného kontrolního bodu vyprázdní databázový server vyrovnávací paměť fyzického protokolu do fyzického protokolu na disk. V případě úplného kontrolního bodu obsahuje vyrovnávací paměť fyzického protokolu všechny změněné stránky, zatímco v případě kontrolních bodů typu fuzzy

obsahuje vyrovnávací paměť fyzického protokolu stránky, které byly změněny operacemi typu fuzzy. Časová značka vyprázdnění fyzického protokolu je uložena ve sdílené paměti.

Změněné stránky ve společné oblasti vyrovnávacích pamětí jsou vyprázdňeny na disk

V případě kontrolního bodu typu fuzzy vyprázdní databázový server stránky změněné operacemi jiného typu než fuzzy ze společné oblasti vyrovnávacích pamětí na disk. Databázový server vyprázdní stránky změněné operacemi typu fuzzy (vlození, odstranění, aktualizace) na disk. Obrázek 16-1 ilustruje způsob, jakým databázový server zapisuje na disk pouze stránky jiného typu než fuzzy. Šedivé čtverce označené znakem **F** představují stránky typu fuzzy.



Obrázek 16-1. Selektivní zápis změněných stránek ze sdílené paměti na disk

Při úplném kontrolním bodu vyprázdní databázový server všechny změněné stránky z společné oblasti vyrovnávacích pamětí ve sdílené paměti na disk.

Jednotkový proces kontrolního bodu запиše záznam kontrolního bodu

Tento krok je stejný pro úplný kontrolní bod i pro kontrolní bod typu fuzzy. Záznam o *dokončení kontrolního bodu* je zapsán do vyrovnávací paměti logického protokolu poté, co byly změněné stránky zapsány na disk.

V případě kontrolního bodu typu fuzzy se do vyrovnávací paměti logického protokolu запиše záznam DPT. Další informace naleznete v kapitole zaměřené na typy záznamů logického protokolu v příručce *IBM Informix: Administrator's Reference*.

Fyzický protokol je logicky vyprázdněn

Tento krok je stejný pro úplný kontrolní bod i pro kontrolní bod typu fuzzy. Poté, co je záznam o *dokončení kontrolního bodu* zapsán na disk, dojde k logickému vyprázdnění fyzického protokolu, což znamená, že aktuální záznamy ve fyzickém protokolu mohou být přepsány.

Pokyny týkající se zálohování a obnovy

Při zálohování provádí databázový server úplný kontrolní bod a vyprázdní na disk všechny změněné stránky, včetně stránek s operacemi jiného typu než fuzzy. Během obnovy databázový server znovu použije všechny záznamy logického protokolu.

Důležité: Protože logický protokol obsahuje záznamy operací jiného typu než fuzzy dosud nezapsané na disk, musíte logické protokoly pravidelně zálohovat.

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 funkce zajišťující odolnost proti selhání, kterou databázový server spustí pokaždé, kdy přechází z režimu offline do klidového režimu, do jednonoživatelského režimu nebo do režimu 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.

Průběh rychlé obnovy se liší podle toho, zda předchozí kontrolní bod byl úplný kontrolní bod nebo kontrolní bod typu fuzzy. Tato část obsahuje informace o rychlé obnově po obou druzích kontrolních bodů. Další informace naleznete v částech “Podrobnosti rychlé obnovy po úplném kontrolním bodu” na stránce 16-15 a “Podrobnosti rychlé obnovy po kontrolním bodu typu fuzzy” na stránce 16-17.

Potřeba rychlé obnovy

Rychlá obnova obnovuje databázový server do stavu fyzické a logické konzistence po jakémkoli 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

Pokaždé, když administrátor uvádí databázový server z režimu offline do klidového režimu, jednonoživatelského režimu nebo režimu online, kontroluje databázový server, zda není nutná rychlá obnova.

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 12-8), 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 přetéct. 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 nenovějšího kontrolního bodu. Všechny změny provedené od databáze od posledního kontrolního bodu jsou ztraceny. Všechny operace typu fuzzy (vlození, odstranění, aktualizace) dosud nevyprázdněné na disk jsou také ztraceny.

Podrobnosti rychlé obnovy po úplném kontrolním bodu

Tato část popisuje rychlou obnovu po úplném kontrolním bodu. (Další informace rovněž naleznete v části “Podrobnosti rychlé obnovy po kontrolním bodu typu fuzzy” na stránce 16-17.)

Rychlá obnova vrací databázový server do stavu konzistence jako část inicializace sdílené paměti. Konzistentní stav znamená, že všechny potvrzené transakce jsou obnoveny a všechny nepotvrzené transakce jsou odvolány.

Rychlé obnovy je dosaženo v následujících dvou stupních:

- Databázový server použije fyzický protokol k návratu na poslední známý bod *fyzické konzistence*, na nejnovější kontrolní bod.
- Databázový server použije soubory logického protokolu k návratu do stavu *logické konzistence* přehráním všech potvrzených transakcí, které se vyskytly od posledního kontrolního bodu, a k odvolání všech transakcí, které zůstaly nedokončené.

Rychlá obnova probíhá těmito kroky:

1. Použije data ve fyzickém protokolu k návratu všech stránek na disku do stavu v době před nejnovějším kontrolním bodem.
2. Vyhledá záznam nejnovějšího kontrolního bodu v souborech logického protokolu.
3. Přehraje všechny záznamy logického protokolu zapsané po záznamu nejnovějšího protokolu.
4. Odvolá transakce, které nemají přidružení k záznamu COMMIT nebo BEGCOM v logickém protokolu.

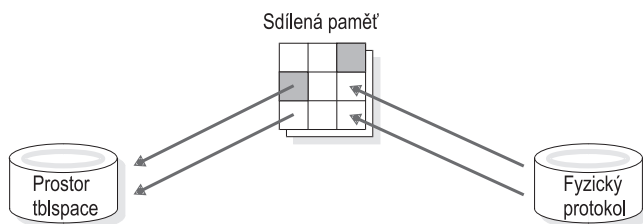
Databázový server zapíše záznam BEGCOM, když je transakce potvrzena. Další informace naleznete v kapitole týkající se záznamů logického protokolu v příručce *IBM Informix: Administrator's Reference*.

Následující odstavce popisují každý uvedený krok podrobně.

Návrat na stav posledního kontrolního bodu

Aby bylo možné provést první krok, vrátí všechny stránky na disku do stavu v okamžiku posledního kontrolního bodu, zapíše databázový server předobrazy uložené ve fyzickém protokolu do sdílené paměti a potom zpátky na disk. Každý předobraz ve fyzickém protokolu obsahuje adresu stránky, která byla aktualizována 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. Tento krok ilustruje Obrázek 16-2.

Databázový server je nyní fyzicky konzistentní.



Obrázek 16-2. Zápis všech zbývajících předobrazů ve fyzickém protokolu na disk

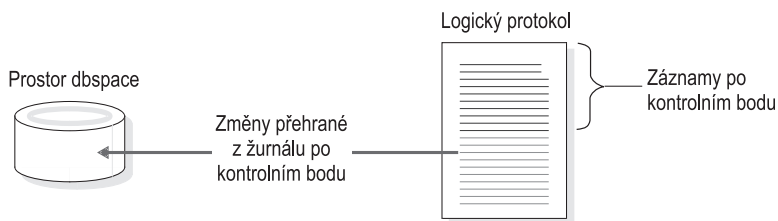
Nalezení záznamu kontrolního bodu v logickém protokolu

Ve druhém kroku databázový server vyhledá adresu záznamu posledního kontrolního bodu v logickém protokolu. Záznam posledního kontrolního bodu se zaručeně nachází v logickém protokolu na disku.

Přehrání záznamů logického protokolu

Ve třetím kroku rychlé obnovy dochází k přehrání záznamů logického protokolu, které byly zapsány po záznamu posledního kontrolního bodu. Touto akcí se reprodukuje

databázi všechny změny od okamžiku posledního kontrolního bodu až do bodu, kdy došlo k nekontrolovanému vypnutí. Tento krok ilustruje Obrázek 16-3.



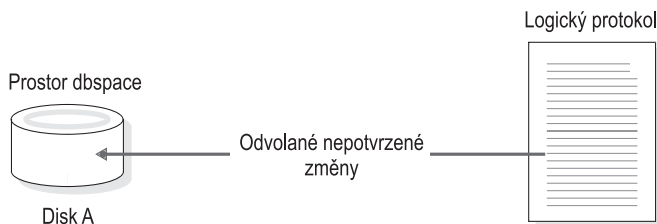
Obrázek 16-3. Přehrání záznamů logického protokolu zapsaných od posledního kontrolního bodu

Odvolání nedokončených transakcí

Posledním krokem v rychlé obnově je odvolání všech záznamů logického protokolu pro transakce, které nebyly v okamžiku selhání systému potvrzeny. Všechny databáze jsou logicky konzistentní, neboť všechny potvrzené transakce byly přehrány a všechny nepotvrzené transakce byly odvolány.

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 23-6.

Protože jedna nebo více transakcí mohou přesahovat přes několik kontrolních bodů, aniž by byly potvrzeny, může procedura odvolání transakce zpětně pročitat logický protokol za záznamem posledního kontrolního bodu. 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 16-4 znázorňuje proceduru odvolání transakce. Po dokončení rychlé obnovy se databázový server vrátí do režimu klidu, režimu jednoho uživatele nebo do režimu online.



Obrázek 16-4. Odvolání všech nedokončených transakcí

Podrobnosti rychlé obnovy po kontrolním bodu typu fuzzy

Tato část popisuje výchozí metodu rychlé obnovy, která nastane po kontrolním bodu typu fuzzy. Informace o dvou konfiguračních parametrech, které můžete použít ke snížení doby obnovy tak, že povolíte databázovému serveru provádět fyzické

protokolování u kontrolních bodů typu fuzzy při obnově během fáze přehrání protokolu, naleznete v části “Alternativní volby obnovy při rychlém restartování pro operace typu fuzzy” na stránce 16-21.

Rychlé obnovy je dosaženo v následujících stupních:

- Databázový server použije fyzický protokol k návratu na poslední kontrolní bod. Databázový server *nemusi* být v tomto bodu rychlé obnovy fyzicky konzistentní, protože operace typu fuzzy neprovádějí fyzické protokolování předobrazu stránek.
- Databázový server zpracovává záznamy logického protokolu počínaje nejstarší aktualizací, která dosud nebyla vyprázdněna na disk, spíše než od předchozího kontrolního bodu.
- Databázový server použije soubory logického protokolu k návratu do stavu *logické konzistence* přehráním všech potvrzených transakcí, které se vyskytly od posledního kontrolního bodu, a k odvolání všech transakcí, které zůstaly nedokončené.

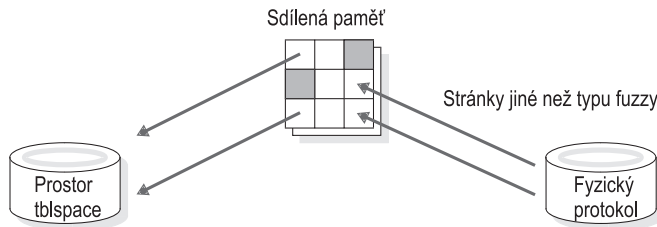
Následující kroky popisují tyto stupně podrobně:

1. Použije data ve fyzickém protokolu k vrácení stránek na disku pro operace jiného typu než fuzzy do stavu v okamžiku nejnovějšího kontrolního bodu.
2. Vyhledá nejstarší aktualizaci v logickém protokolu, která dosud není vyprázdněna na disk.
3. Použije záznamy protokolu pro operace typu fuzzy, které nastaly před nejnovějším kontrolním bodem.
4. Přehraje všechny záznamy logického protokolu zapsané po záznamu nejnovějšího protokolu.
5. Odvolá transakce, které v logickém protokolu nemají přidružení k záznamu COMMIT nebo BEGIN.

Ačkoli rychlá obnova po kontrolním bodu typu fuzzy trvá déle než po úplném kontrolním bodu, můžete ji optimalizovat. Podrobnější informace naleznete v příručce *IBM Informix: Dynamic Server Performance Guide*

Návrat do stavu posledního kontrolního bodu pro operace jiného typu než fuzzy

Aby provedl první krok a vrátil všechny stránky na disku pro operace jiného typu než fuzzy do stavu v okamžiku nejnovějšího kontrolního bodu, zapíše databázový server předobrazy uložené ve fyzickém protokolu do sdílené paměti a potom zpátky na disk. Každý předobraz ve fyzickém protokolu obsahuje adresu stránky, která byla aktualizována 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. Tento krok ilustruje Obrázek 16-5.

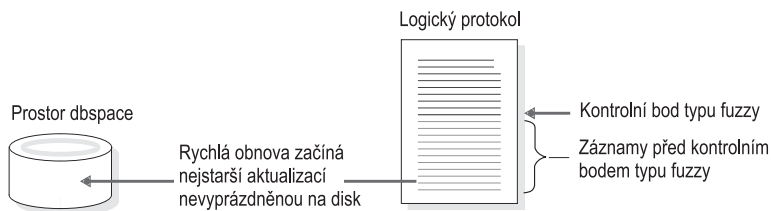


Obrázek 16-5. Zápis předobrazů pro operace jiného typu než fuzzy ve fyzickém protokolu na disk

Stránky, na kterých se vyskytly operace typu fuzzy, nejsou fyzicky konzistentní, protože databázový server neprovádí fyzické protokolování předobrazů. Pokud je nejnovější kontrolní bod kontrolní bod typu fuzzy, změněné stránky pro operace typu fuzzy nebyly vyprázdněny na disk. Disk prostoru dbspace obsahuje předobraz každé stránky. K vrácení změn u těchto stránek před kontrolní bod typu fuzzy používá databázový server logický protokol, jak je popsáno v následujícím kroku.

Vyhledání nejstarší aktualizace v logickém protokolu

Databázový server již dále nezačíná rychlou obnovu na záznamu nejnovějšího kontrolního bodu. V tomto kroku vyhledá databázový server v logickém protokolu záznam nejstarší aktualizace, který nebyl vyprázdněn na disk během nejnovějšího kontrolního kroku. K vyhledání záznamu nejstarší aktualizace používá databázový server pořadová čísla protokolu (LSN). Obrázek 16-6 ukazuje, že se nejstarší aktualizace v logickém protokolu vyskytuje před několika kontrolními body a že jsou použity všechny záznamy protokolu.



Obrázek 16-6. Vyhledání nejstarší aktualizace v logickém protokolu

Čísla LSN nemohou jít dále zpět v protokolu než o jeden soubor logického protokolu a dva kontrolní body. V každém kontrolním bodu typu fuzzy zapiše databázový server stránky s časovou značkou dřívější než je nejstarší číslo LSN a číslo LSN posune dopředu.

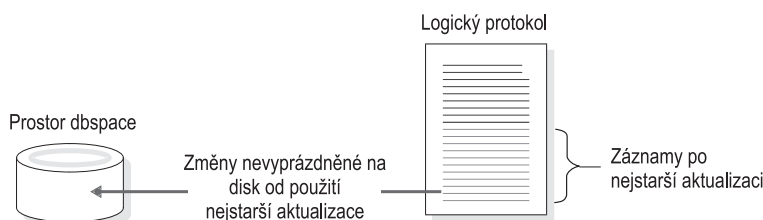
Není možné uvolnit logický soubor, který obsahuje záznam nejstarší aktualizace, dokud nejsou změny zaznamenány na disk. Databázový server automaticky provede úplný kontrolní bod, aby přešel problémům s rychlou obnovou velmi starých záznamů protokolu.

Použití záznamů protokolu pro operace typu fuzzy

Ve druhém kroku zpracuje databázový server záznamy protokolu pro operace typu fuzzy, které se vyskytly po nejstarší aktualizaci a před posledním kontrolním bodem. Záznamy protokolu představující změny dat musí být na disk předtím, než změněná data hradí předchozí verzi na disku.

Záznamy protokolu pro operace typu fuzzy se selektivně provádějí znovu v závislosti na tom, zda již byla aktualizace aplikována na stránku. Je-li časová značka v záznamu logického protokolu novější než časová značka na stránce na disku, použije databázový server tento záznam. V opačném případě databázový server tento záznam přeskočí.

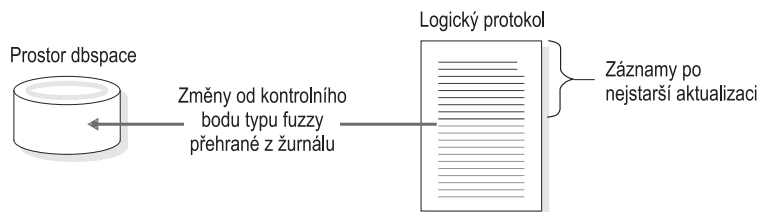
Způsob, jakým databázový server zpracovává pouze záznamy typu fuzzy před kontrolním bodem ilustruje Obrázek 16-7.



Obrázek 16-7. Použití záznamů protokolu pro operace typu fuzzy

Přehrání záznamů logického protokolu

Ve třetím kroku databázový server zpracuje všechny záznamy logického protokolu následující po posledním kontrolním bodu. Rychlá obnova přehraje záznamy logického protokolu, které byly zapsány po záznamu nejnovějšího kontrolního bodu. Touto akcí se databázi reprodukuje všechny změny od okamžiku posledního kontrolního bodu až do bodu, kdy došlo k nekontrolovanému vypnutí. Obrázek 16-8 ilustruje přehrání všech záznamů po kontrolním bodu typu fuzzy.

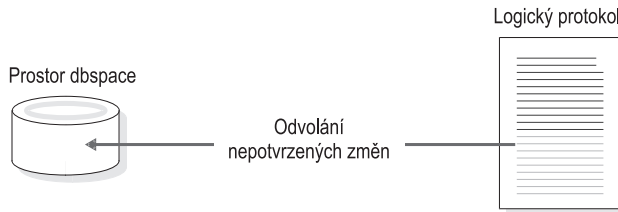


Obrázek 16-8. Přehrání záznamů logického protokolu zapsaných od nejnovějšího kontrolního bodu typu fuzzy

Odvolání nedokončených transakcí

Posledním krokem v rychlé obnově je odvolání všech záznamů logického protokolu pro transakce, které nebyly v okamžiku selhání systému potvrzeny. Tato procedura odvolání transakcí zajistí, že budou databáze zanechány v konzistentním stavu.

Jelikož jedna nebo více transakcí mohou přesahovat přes několik kontrolních bodů, aniž by byly potvrzeny, může procedura odvolání transakce zpětně pročítat logický protokol za záznamem posledního kontrolního bodu. 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 16-9 ilustruje proceduru odvolání transakcí. Po dokončení rychlé obnovy databázový server přejde do režimu klidu, režimu jednoho uživatele nebo do režimu online.



Obrázek 16-9. Odvolání všech nedokončených transakcí

Alternativní volby obnovy při rychlém restartování pro operace typu fuzzy

V situacích, kdy potřebujete výkon obnovy, který bude rychlejší než výchozí, výše popsané operace obnovy, můžete použít konfigurační parametry `FAST_RESTART_PHYSLOG` a `FAST_RESTART_CKPT_FUZZYLOG`. Tyto parametry umožňují databázovému serveru provádět fyzické protokolování kontrolních bodů typu fuzzy během fáze přehrání protokolu, a tak snížit čas nutný na obnovu.

Když je databázový server v režimu online:

- Nastavte konfigurační parametr `FAST_RESTART_PHYSLOG` na hodnotu 1, abyste povolili fyzické protokolování. Používejte tento parametr pouze tehdy, pokud je společná oblast vyrovnávacích paměti alespoň o 25 procent větší než velikost vyrovnávací paměti fyzického protokolu. společná oblast vyrovnávacích paměti musí být dostatečně velká, aby byla schopna uchovat fyzický protokol, stránky protokolu a další stránky přečtené během obnovy.

Pokud není společná oblast vyrovnávacích paměti správně nakonfigurovaná, sníží se výkon rychlé obnovy.

- Nastavte konfigurační parametr `FAST_RESTART_CKPT_FUZZYLOG` na hodnotu 1, abyste povolili vyprázdnění záznamů aktualizovaných stránek do fyzického protokolu během v okamžiku kontrolního bodu.

Při použití kontrolních bodů typu fuzzy můžete použít jeden nebo oba dva konfigurační parametry. Výchozí hodnota pro oba parametry je 0 (vypnuto).

Parametry `FAST_RESTART_PHYSLOG` a `FAST_RESTART_CKPT_FUZZYLOG` zlepšují výkon obnovy pouze tehdy, pokud tyto parametry povolíte, když je databázový server spuštěn. Tyto parametry nemají vliv, pokud je povolíte, když je databázový server vypnutý.

Toto mimořádné fyzické protokolování, ke kterému dochází, když databázový server použije parametr `FAST_RESTART_PHYSLOG`, ovlivňuje výkon výkon rozhraní (runtime). Jestliže nechcete obětovat výkon rozhraní nebo pokud nechcete zvětšovat velikost vyrovnávací paměti, použijte k částečnému snížení doby obnovy parametr `FAST_RESTART_CKPT_FUZZYLOG`.

Parametr `FAST_RESTART_PHYSLOG` vstupuje v platnost okamžitě po povolení. Pokud ovšem databázový server selže předtím, než se provede následující kontrolní bod, rychlá obnova nedosáhne maximálního výkonu, protože databázový server nezaprotokoloval v intervalech mezi kontrolními body všechny aktualizace typu fuzzy.

Parametr `FAST_RESTART_CKPT_FUZZYLOG` vstupuje v platnost v okamžiku kontrolního bodu, který nastane po povolení tohoto parametru. Pokud celkový počet nevyprázdněných, aktualizovaných stránek typu fuzzy přesahuje 20 procent celkového prostoru fyzického protokolu, stránky nebudou zapsány do fyzického protokolu.

Po povolení parametru `FAST_RESTART_PHYSLOG` jeho nastavením na hodnotu 1 můžete zahájit rychlou obnovu pomocí obslužného programu **oninit**, aniž byste zadávali jakékoli volby.

Kapitola 17. Správa fyzického protokolu

Změna umístění a velikosti fyzického protokolu	17-1
Příprava na provedení změn	17-2
Kontrola odpovídajícího souvislého prostoru	17-2
Použití textového editoru ke změně umístění a velikosti fyzického protokolu	17-3
Použití obslužného programu onparams ke změně umístění nebo velikosti fyzického protokolu	17-3
Použití programu ON-Monitor ke změně umístění a velikosti fyzického protokolu	17-3
Monitorování aktivity fyzického a logického protokolování	17-4
Vzorový výstup příkazu onstat -l	17-5
Monitorování informací o kontrolních bodech	17-5
Vynucení úplného kontrolního bodu	17-6
Vynucení kontrolního bodu typu fuzzy	17-7
Použití tabulek SMI	17-7

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 16, “Fyzické protokolování, kontrolní body a rychlá obnova”, na stránce 16-1.

Změna umístění a velikosti fyzického protokolu

Umístění a velikost fyzického protokolu můžete měnit několika způsoby:

- Úpravou souboru ONCONFIG pomocí textového editoru nebo programu ISA.
- Pomocí obslužného programu onparams v příkazovém řádku.

Pouze pro UNIX

- Pomocí programu ON-Monitor.

Konec Pouze pro UNIX

Chcete-li provádět změny, přihlaste se do operačního systému UNIX jako uživatel informix nebo root nebo se přihlaste do operačního systému Windows jako člen skupiny **Informix-Admin**. Následující části popisují každou z uvedených metod.

Chcete-li aktivovat změny velikosti nebo umístění fyzického protokolu ihned po jejich provedení, vypněte databázový server a restartujte jej. Automatické vypnutí a restartování databázového serveru provede obslužný program **onparams**.

Okamžitě po restartování databázového serveru vytvořte zálohu úrovně 0. Tato záloha paměťových prostorů je kritická pro zotavení databázového serveru.

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 16-4. Informace o určování velikosti fyzického protokolu naleznete v části “Velikost a umístění fyzického protokolu” na stránce 16-4. Další informace o fyzickém protokolu naleznete v části “Monitorování aktivity fyzického a logického protokolování” na stránce 17-4.

Příprava na provedení změn

Prostor přidělený pro fyzický protokol musí souvislý. K velmi závažným chybám sdílené paměti dochází, když se pokusíte vypnout a restartovat databázový server v těchto situacích:

- Pokud přesunete fyzický protokol do prostoru dbspace bez odpovídajícího souvislého prostoru.
- Pokud zvětšíte velikost protokolu nad velikost dostupného souvislého prostoru, nastane velmi závažná chyba při pokusu o restartování databázového serveru s novými hodnotami.

Jestliže se vyskytne tato chyba, změňte velikost fyzického protokolu nebo zvolte jiný prostor dbspace s odpovídajícím souvislým prostorem a potom databázový server vypněte a restartujte.

Kontrola odpovídajícího souvislého prostoru

Pomocí volby **oncheck -pe** můžete zjistit, zda je dostupný odpovídající souvislý prostor. Další informace o používání voleb **oncheck -ce** a **-pe** ke kontrole seznamu volných stránek v bloku naleznete v kapitole zaměřené na obslužné programy v příručce *IBM Informix: Administrator's Reference*.

Další informace naleznete v části “Monitorování bloků” na stránce 11-41.

Použití textového editoru ke změně umístění a velikosti fyzického protokolu

Můžete změnit umístění a velikost fyzického protokolu úpravou souboru ONCONFIG, zatímco je databázový server v režimu online.

Parametr	Popis
PHYSFILE	Určuje velikost fyzického protokolu v kilobajtech.
PHYSDBS	Přesune fyzický protokol do určeného prostoru dbspace.

Změny nejsou platné, dokud nevypnete databázový server a nerestartujete jej. Potom okamžitě vytvořte zálohu úrovně 0, abyste zajistili úplnou dostupnost mechanismu zotavení.

Další informace o konfiguračních parametrech PHYSFILE a PHYSDBS naleznete v kapitole týkající se konfiguračních parametrů 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, spusíte následující příkaz poté, co přivedete databázový server do klidového režimu nebo do jednonoživatelského režimu.

```
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. Nová velikost fyzického protokolu je 400 kilobajtů a protokol bude umístěn v prostoru dbspace označeném **dbspace6**. Příkaz také znovu inicializuje sdílenou paměť pomocí volby **-y**, takže změna se uplatní okamžitě, jak je uvedeno:

```
onparams -p -s 400 -d dbspace6 -y
```

Po vypnutí a restartování databázového serveru vytvořte zálohu úrovně 0, abyste zajistili úplnou dostupnost mechanismu zotavení.

Další informace o používání obslužného programu onparams ke změně fyzického protokolu naleznete v kapitole zaměřené na obslužné programy v příručce *IBM Informix: Administrator's Reference*.

Použití programu ON-Monitor ke změně umístění a velikosti fyzického protokolu

Chcete-li změnit velikost, umístění v prostoru dbspace nebo obojí, vyberte položku **Parameters > Physical-Log**.

Potom vytvořte ihned zálohu úrovně 0, aby byly dostupné všechny mechanismy zotavení.

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 *IBM Informix: Performance Guide*.

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).

Obslužný program	Příkaz	Dodatečné informace
		<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> Logický	Klepněte na volbu Přejít na následující soubor protokolu .

Vzorový výstup příkazu onstat -l

Obrázek 17-1 ukazuje vzorový výstup volby **onstat -l**, která zobrazuje informace o vyrovnávacích pamětech fyzického a logického protokolu.

```
Physical Logging
Buffer bufused  bufsize  numpages  numwrits  pages/io
P-2  0          16        110       10        11.00
      phybegin  physize   phypos    phyused   %used
      10003f    500      233      0         0.00

Logical Logging
Buffer bufused  bufsize  numrecs  numpages  numwrits  recs/pages  pages/io
L-1  0          16       3075     162       75        19.0       2.2

...
```

Obrázek 17-1. Výstup příkazu *onstat -l*, který zobrazuje informace o vyrovnávacích pamětech fyzického a logického protokolu.

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 *IBM Informix: Dynamic Server Performance Guide*.

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, nezapíše 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 zabráňuje během kontrolního bodu uživatelským jednotkovým procesům ve vstupu do kritické části.
ON-Monitor (UNIX)	Status > Profile	Pole Checkpoints a Check Waits zobrazují stejné informace jako pole numckpts a ckpwaits v příkazu onstat -p .

Vynucení úplného kontrolního bodu

Pokud chcete vynutit úplný kontrolní bod, spusťte jeden z následujících příkazů.

Obslužný program	Příkaz	Dodatečné informace
Příkazový řádek nebo program ISA	onmode -c	Žádné.
ISA	Protokoly > Logický	Klepněte na volbu Vynutit kontrolní bod .
ON-Monitor (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.

Úplný kontrolní bod byste měli vynutit v jakékoli z následujících situací:

- Chcete-li uvolnit soubor logického protokolu, který obsahuje záznam nejnovějšího kontrolního bodu a který je zálohovaný, ale nebyl dosud uvolněn (u příkazu **onstat -l**: stav **U-B-L** nebo **U-B**).
- 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 se restartuje po inicializaci databázového serveru).

- Pokud se kontrolní bod nevyskytnul 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 jejich četnost na určitou dobu 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. Protože zavádění tabulek používají rychlou vyrovnávací paměť, vyčistěte rychlou vyrovnávací paměť vynucením kontrolního bodu.

Vynucení kontrolního bodu typu fuzzy

Kontrolní bod typu fuzzy vynutíte spuštěním příkazu **onmode -c fuzzy**.

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
plgpagewrites	Počet stránek zapsaných do vyrovnávací paměti fyzického protokolu.
plgwrites	Počet zápisů z vyrovnávací paměti fyzického protokolu do souboru fyzického protokolu.
llgreccs	Počet záznamů zapsaných do vyrovnávací paměti logického protokolu.
llgpagewrites	Počet stránek zapsaných do vyrovnávací paměti logického protokolu.
llgwrites	Počet zápisu 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 .

Část 4. Odolnost vůči selhání

Kapitola 18. Zrcadlení

Zrcadlení	18-1
Výhody zrcadlení	18-2
Požadavky na zrcadlení	18-2
Důsledky nepoužívání zrcadlení	18-2
Data vhodná k zrcadlení	18-3
Alternativy k zrcadlení	18-3
Správci logických disků	18-4
Hardwarové zrcadlení	18-4
Externí zálohování a obnovení	18-4
Průběh zrcadlení	18-4
Vytvoření zrcadleného bloku	18-4
Stavové příznaky zrcadlení	18-5
Zotavení	18-5
Činnosti v průběhu zpracování	18-6
Diskové zápisy do zrcadlených bloků	18-6
Diskové čtení ze zrcadlených bloků	18-6
Zjišťování poruch média	18-6
Zotavení bloku	18-7
Výsledek zastavení zrcadlení	18-7
Struktura zrcadleného bloku	18-7

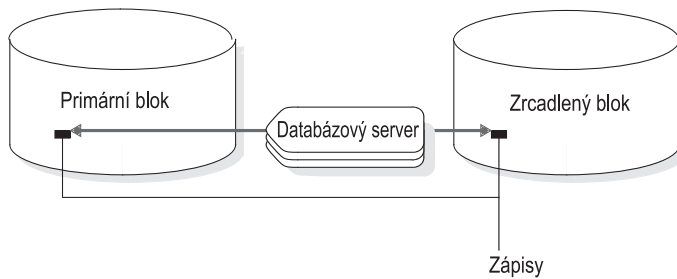
Obsah kapitoly

Tato kapitola popisuje funkci zrcadlení databázového serveru. Další pokyny pro provádění úloh zrcadlení uvádí Kapitola 19, “Použití zrcadlení”, na stránce 19-1.

Zrcadlení

Zrcadlení je strategie, která páruje *primární blok* jednoho definovaného prostoru dspace, blobspace nebo sbpace se stejně velkým *zrcadleným blokem*.

Každý zápis do primárního bloku je automaticky doprovázen identickým zápisem do zrcadleného bloku. Tento pojem ilustruje Obrázek 18-1. 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 18-1. 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 zotavení 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 zrcadlící 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 18-6.

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 zotavením prostoru 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 18-6.) Zrcadlení databázového serveru navíc zvyšuje výkon čtení pomocí rozděleného čtení. (Další informace naleznete v části “Diskové čtení ze zrcadlených bloků” na stránce 18-6.) 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 19, "Použití zrcadlení", na stránce 19-1.

Vytvoření zrcadleného bloku

Zadáte-li zrcadlený blok, databázový server kopíruje všechna data z primárního bloku do zrcadleného bloku. Tento proces kopírování se nazývá *zotavení*. Zrcadlení je spuštěno, jakmile je zotavení dokončeno.

Procedura zotavení, 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 obnovení 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 obnovení 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. Jakmile zálohování začalo, není možné zkopírovat nové údaje, které se musí objevit v prvním bloku záložní pásky prostoru dbspace.

Více informací o vytváření zrcadlených bloků najdete v části Kapitola 19, “Použití zrcadlení”, na stránce 19-1.

Stavové příznaky zrcadlení

Prostory dbspace, blobspace a sbospace 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 zotavování.

Popis těchto stavových příznaků bloků najdete v popisu volby **onstat -d** v kapitole o obslužných programech v příručce *IBM Informix: Administrator's Reference*. Další údaje, jak zobrazit tyto stavové příznaky, najdete v části “Monitorování využití disku” na stránce 11-41.

Zotavení

Když databázový server zotaví zrcadlený blok, provede stejnou obnovovací proceduru, jako když zrcadlení začíná. Proces zotavení zrcadlení kopíruje data z existujícího funkčního bloku do nového opraveného bloku do té doby, než jsou bloky identické.

Při zahájení zotavení databázový server uvede nefunkční blok do režimu zotavení a kopíruje data z funkčního bloku do obnovovaného bloku. Když je zotavení dokončeno, blok automaticky získá stav online. Stejný postup se provádí při zotavení primárního bloku i při zotavení zrcadleného bloku.

Tip: Během procesu zotavení 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 19-7.

Činnosti v průběhu zpracování

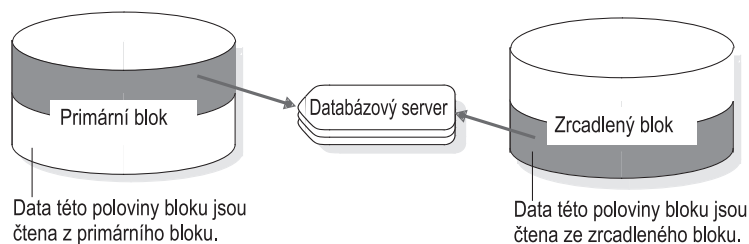
V této části se projednávají některé detaily diskového vstupu - výstupu pro zrcadlené bloky, a to, jak se databázový server u těchto bloků vypořádá s poruchou média.

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 18-2 ilustruje rozdělené čtení.



Obrázek 18-2. 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 18-5.

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.

Jakmile administrátor zotaví nefunkční blok a vrátí ho do režimu online, rozdělí se opět čtení mezi primární a zrcadlené bloky a zápisy se budou opět provádět do obou bloků.

Zotavení bloku

Databázový server používá asynchronní vstup - výstup, aby minimalizoval dobu, která je zapotřebí k zotavení 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 sbpace 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 11-41, 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 19. Použití zrcadlení

Příprava k zrcadlení dat	19-2
Povolení konfiguračního parametru MIRROR	19-2
Změna parametru MIRROR pomocí programu ON-Monitor (UNIX)	19-3
Přidělení diskového prostoru pro zrcadlená data	19-3
Propojení bloků (UNIX)	19-3
Opětovné propojení bloku se zařízením po poruše disku	19-3
Použití zrcadlení	19-4
Zrcadlení kořenového prostoru dbspace během spuštění	19-4
Změna stavu zrcadlení	19-5
Správa zrcadlení	19-5
Spuštění zrcadlení u nezrcadlených paměťových prostorů	19-5
Spuštění zrcadlení nezrcadlených prostorů dbspace pomocí programu onspaces	19-5
Spuštění zrcadlení pomocí programu ISA	19-6
Spuštění zrcadlení nezrcadlených prostorů dbspace pomocí programu ON-Monitor (UNIX)	19-6
Spuštění zrcadlení nových paměťových prostorů	19-6
Spuštění zrcadlení nových prostorů pomocí programu onspaces	19-6
Spuštění zrcadlení nových prostorů pomocí programu ISA	19-6
Spuštění zrcadlení nových prostorů dbspace pomocí programu ON-Monitor (UNIX)	19-6
Přidání zrcadlených bloků	19-7
Přidání zrcadlených bloků pomocí programu onspaces	19-7
Přidání zrcadlených bloků pomocí programu ISA	19-7
Přidání zrcadlených bloků pomocí programu ON-Monitor (UNIX)	19-7
Vypnutí zrcadleného bloku	19-7
Vypnutí zrcadlených bloků pomocí obslužného programu onspaces	19-7
Vypnutí zrcadlených bloků pomocí programu ON-Monitor (systém UNIX)	19-7
Obnovení zrcadleného bloku	19-7
Obnovení zrcadleného bloku pomocí programu onspaces	19-8
Obnovení zrcadleného bloku pomocí programu ISA	19-8
Obnovení zrcadleného bloku pomocí programu ON-Monitor (UNIX)	19-8
Ukončení zrcadlení	19-8
Ukončení zrcadlení pomocí programu onspaces	19-8
Ukončení zrcadlení pomocí programu ON-Monitor (systém UNIX)	19-8
Ukončení zrcadlení pomocí programu ISA	19-8

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. Uveďte databázový server do režimu offline a povolte zrcadlení.
Další informace naleznete v části “Povolení konfiguračního parametru MIRROR” na stránce 19-2.
2. Vraťte databázový server zpět do režimu online.
3. Přidělte 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 19-3.
4. Zvolte prostory dbspace, blobspace a sbspace, 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 19-4.

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 sbspace. Další informace naleznete v části “Použití zrcadlení” na stránce 19-4.

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.

Chcete-li v databázovém serveru povolit zrcadlení, musíte nastavit parametr MIRROR v souboru **ONCONFIG** na hodnotu 1.. Výchozí hodnota parametru MIRROR je 0, to znamená, 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 (UNIX)

Chcete-li povolit zrcadlení, zvolte možnost **Parameters > Initialize**. V poli označeném **Mirror** zadejte hodnotu **Y**. Stisknutím klávesy **ESC** změny uložte.

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 11-3.

Diskový prostor pro zrcadlený blok vždy přidělte 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)

Použijte příkaz `link` operačního systému UNIX (**ln**) k propojení skutečných souborů nebo přímých zařízení zrcadlených bloků s názvy cest k zrcadlům. 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

V operačním systému UNIX je možné opětovně propojit blok se souborem nebo přímým zařízením na jiném disku, jestliže přestane pracovat disk, na němž se nachází skutečný zrcadlený soubor nebo přímá data. Tato akce umožňuje obnovit zrcadlený blok před tím, než je disk, který selhal, znovu uveden do činnosti. Následující příklad ukazuje typické příkazy operačního systému UNIX, které se používají pro opětovné propojení.

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, blobospace nebo sbospace 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 18-4.

Důležité: Vždy je nutné spustit zrcadlení celého prostoru dbspace, blobospace nebo sbospace. Databázový server nedovoluje v prostorech dbspace, blobospace nebo sbospace 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, blobospace, nebo sbospace.

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.

Jestliže nezadáte název cesty a posun zrcadla, ale chcete spustit zrcadlení kořenového prostoru dbSPACE, musíte po inicializaci databázového serveru změnit stav zrcadlení kořenového prostoru dbSPACE.

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 11-41.

Správa zrcadlení

Ke správě zrcadlení je možné použít obslužný program **onspaces**. V operačním systému UNIX je možné ke správě zrcadlení použít také program ON-Monitor. Úplný popis syntaxe programu **onspaces** naleznete v kapitole o obslužných programech 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 **Paměť > Bloky**.
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 sbpace.

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 sbpace**.
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 sbspace 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 zrcadlená bloků pomocí programu ISA:

1. Vyberte **Storage > Chunks**.
2. Vyberte jméno prostoru dbspace a klepněte na volbu **Add Chunk**.
3. Vložte cestu a posun zrcadleného bloku.

Přidání zrcadlených bloků pomocí programu ON-Monitor (UNIX)

Volba programu ON-Monitor **Dbspaces > Add-chunk** zobrazí pole, do nichž je třeba zadat cestu, posun a velikost primárního bloku a cestu 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 19-3.)

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 sbospace databázový server okamžitě uvolní zrcadlené bloky tohoto prostoru. Tyto bloky jsou ihned k dispozici k opětovnému přidělení jiným paměťovým prostorům.

Ukončit zrcadlení mohou jen uživatelé **informix** a **root** systému UNIX, nebo členové skupiny **Informix-Admin** systému Windows.

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 **Paměť > Bloky**.
2. Vyberte název prostoru dbspace a klepněte na volbu **Zastavit zrcadlení**.

Kapitola 20. Replikace HDR (High-Availability Data Replication)

Replikace High-Availability Data Replication	20-2
Režimy replikace HDR	20-3
Typy replikovaných dat	20-3
Výhody replikace dat	20-3
Primární a sekundární databázové servery	20-4
Porovnání replikace HDR a zrcadlení	20-5
Porovnání replikace HDR a dvoufázového potvrzování	20-6
Replikace HDR a replikace Enterprise Replication	20-6
Jak pracuje replikace HDR	20-7
Průběh prvotní replikace dat	20-7
Reprodukce aktualizací primárního databázového serveru	20-8
Jak jsou odesílány záznamy protokolu	20-8
Vyrovnávací paměti replikace HDR	20-8
Kdy jsou záznamy odesílány	20-9
Synchronní aktualizace	20-9
Asynchronní aktualizace	20-9
Jednotkové procesy, které zpracovávají replikaci HDR	20-11
Kontrolní body mezi databázovými servery	20-12
Jak je zaznamenávána synchronizace dat	20-12
Selhání replikace HDR	20-13
Definice selhání replikace HDR	20-13
Zjišťování selhání replikace HDR	20-13
Akce, které je třeba provést při selhání replikace HDR	20-14
Co provést v případě selhání replikace HDR	20-14
Postup v případě selhání sekundárního databázového serveru	20-14
Postup v případě selhání primárního databázového serveru	20-15
Přesměrování a propojitelnost pro klienty replikace dat	20-17
Návrh klientů pro přesměrování	20-17
Automatické přesměrování klientů pomocí proměnné prostředí DBPATH	20-18
Jak pracuje metoda přesměrování pomocí proměnné prostředí DBPATH	20-18
Co by měli provést administrátoři	20-19
Co by měli provést uživatelé	20-19
Směrování klientů pomocí informací o propojitelnosti	20-19
Jak pracuje přesměrování pomocí informací o propojitelnosti	20-19
Změna informací o propojitelnosti	20-20
Připojení k databázovému serveru	20-22
Směrování klientů pomocí proměnné prostředí INFORMIXSERVER	20-22
Jak pracuje přesměrování pomocí proměnné prostředí INFORMIXSERVER	20-22
Co by měli provést administrátoři	20-22
Co by měli provést uživatelé	20-23
Přesměrování z aplikace	20-23
Porovnání různých mechanismů přesměrování	20-24
Návrh klientů replikace HDR	20-25

Nastavení režimu uzamykání při přístupu k primárnímu databázovému serveru na čekání	20-25
Návrh klientů k použití sekundárního databázového serveru	20-26
Žádné příkazy ke změně dat	20-26
Úroveň uzamčení a izolace.	20-27
Použití dočasných prostorů dbSPACE k řazení a pro dočasné tabulky	20-27

Obsah kapitoly

Replikace dat označuje proces reprezentace objektů databáze na více než jednom samostatném místě. Funkce *High-Availability Data Replication* (HDR) umožňuje synchronní replikaci dat serveru Dynamic Server. Replikaci HDR použijte tehdy, pokud potřebujete okamžitě použitelnou zálohu.

V této kapitole jsou popsána následující témata:

- Co je replikace HDR.
- Jak replikace HDR pracuje.
- Jak replikace se replikace HDR vypořádává se selháním.
- Jak lze přesměrovat klient, aby se připojil k druhému databázovému serveru v páru replikace HDR (také nazývanému *replikační pár*).
- Jaká posouzení návrhu je třeba provést v případě aplikací, které se připojují k sekundárnímu databázovému serveru.

Kapitola 21, “Použití replikace HDR”, na stránce 21-1 obsahuje pokyny, jak provádět úlohy administrace spojené s používáním replikace HDR.

Tip: Pokud chcete používat asynchronní replikaci dat, naleznete pokyny v příručce *IBM Informix: Dynamic Server Enterprise Replication Guide*.

Replikace High-Availability Data Replication

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.

Databázový server implementuje téměř transparentní replikaci dat celých databázových serverů. Všechna data spravovaná jedním databázovým serverem jsou replikována a dynamicky aktualizována v jiném databázovém serveru, který se často nachází na geograficky vzdáleném místě. Replikace HDR poskytuje způsob, jak spravovat záložní kopii celého databázového serveru, ke které mohou aplikace rychle získat přístup v případě závažného selhání.

Důležité: Replikace HDR nepodporuje šifrování síťových přenosů v průběhu replikace.

Režimy replikace HDR

Následující tabulka znázorňuje režimy replikace HDR.

Tabulka 20-1.

Režim replikace HDR	Popis
Standardní režim	Není součástí systému replikace HDR.
Primární režim	Primární režim systému HDR. Data lze aktualizovat.
Sekundární režim	Sekundární režim systému HDR. Data nelze aktualizovat, lze je však číst.

Databázový server spuštěný ve standardním režimu nebo databázový server 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 4-9.

Typy replikovaných dat

Replikace HDR replikuje data v prostorech dbspace a sbspace. Replikace HDR nereplikuje data prostorů blobspace.

Replikace HDR 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 sbspace nebo v jiné tabulce téhož databázového serveru. Aby mohla být replikována data prostoru sbspace, musí být prostor sbspace protokolován.

Replikace HDR 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 20-7.

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 nemusejí 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.

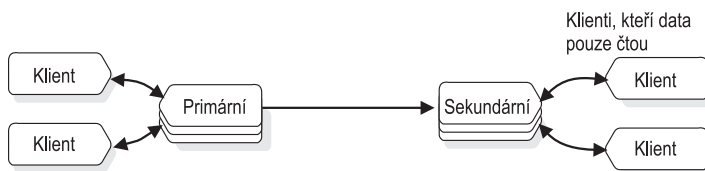
Replikaci dat by bylo možné implementovat v klientské aplikaci pomocí explicitního určení umístění dat, která mají být aktualizována. Tento způsob replikace dat je však náročný, není odolný vůči chybám a znesnadňuje správu. Koncept replikace dat je naopak často spojován s *transparentní replikací*. Replikace vestavěná do databázového serveru (nikoli do aplikace) je transparentní, databázový server sám vyhledává a spravuje repliky dat.

Primární a sekundární databázové servery

Pokud konfigurujete dvojici databázových serverů k použití replikace HDR, bude jeden databázový server nazýván *primární* databázový server a druhý server bude nazýván *sekundární* databázový server. (V tomto kontextu bude databázový server, který nepoužívá replikaci HDR, nazýván *standardní* databázový server.)

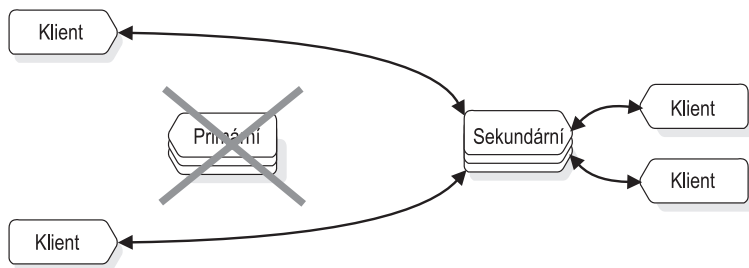
Při normálním provozu se klienti mohou připojovat k primárnímu databázovému serveru a používat ho jako běžný databázový server. Klienti mohou při normálním provozu používat také sekundární databázový server, ale pouze ke čtení dat. Sekundární databázový server nedovoluje klientským aplikacím data aktualizovat.

Jak znázorňuje Obrázek 20-1, bude sekundární databázový server dynamicky aktualizován, jakmile dojde ke změně dat spravovaných primárním databázovým serverem.



Obrázek 20-1. Primární a sekundární databázový server v páru replikace HDR

Pokud jeden ze serverů v páru replikace HDR selže, jak znázorňuje Obrázek 20-2, budete moci přesměrovat klienty používající tento databázový server na druhý databázový server v páru.



Obrázek 20-2. Databázové servery v páru replikace HDR a klienti po selhání

Pokud primární databázový server selže, budete moci změnit sekundární databázový server na standardní databázový server, aby mohl přijímat od klientů aktualizace.

Replikace HDR má následující vlastnosti:

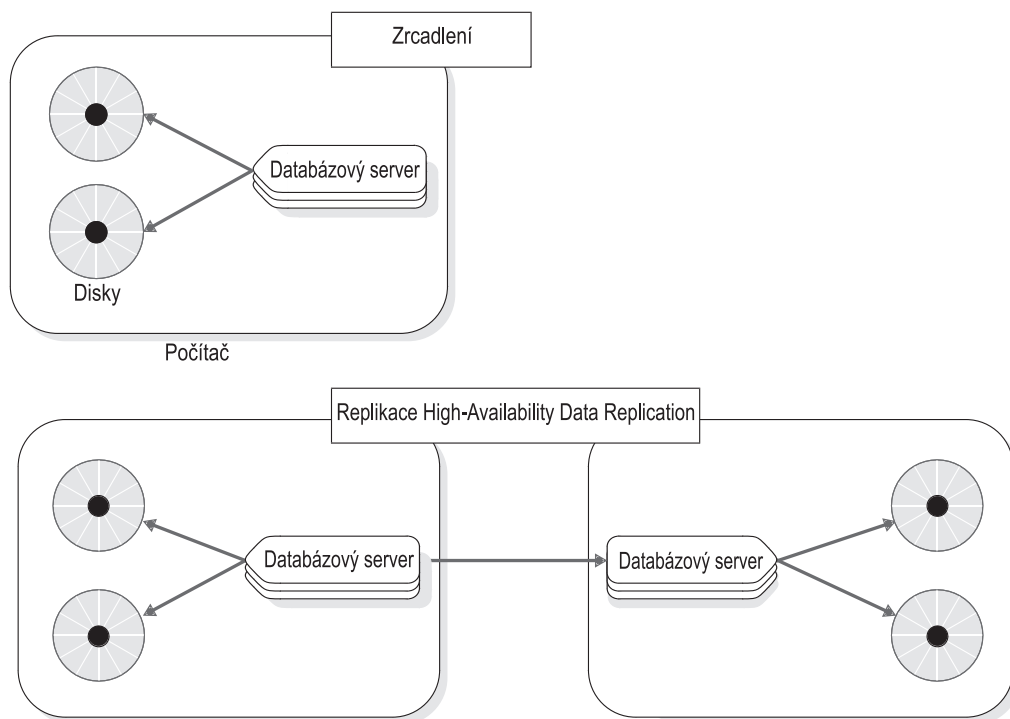
- Umožňuje rychlé zotavení po selhání jednoho z databázových serverů.
- Umožňuje rozkládat zatížení mezi oba databázové servery.

Porovnání replikace HDR a zrcadlení

Jak replikace HDR, tak i zrcadlení jsou transparentní způsoby zvýšení odolnosti databázového serveru proti selhání. Jak však znázorňuje Obrázek 20-3, mezi oběma funkcemi jsou značné rozdíly.

Zrcadlení, které popisuje část “Zrcadlení” na stránce 18-1, 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 HDR však všechna data spravovaná databázovým serverem (nikoli pouze určené prostory dbspace) replikuje ve zcela odděleném počítači. Protože replikace HDR 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 20-3. Porovnání zrcadlení a replikace HDR

Porovnání replikace HDR a dvoufázového potvrzování

Protokol dvoufázového potvrzování, který podrobně popisuje Kapitola 23, “Protokoly vícefázového potvrzování”, na stránce 23-1, 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 HDR.

Replikace HDR a replikace Enterprise Replication

K vytvoření robustního systému replikace můžete použít kombinaci replikace HDR a replikace Enterprise Replication. Replikace HDR 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 HDR a replikace Enterprise Replication, bude k systému replikace Enterprise Replication připojen pouze primární server replikace

HDR. Sekundární server replikace HDR se nebude účastnit replikace Enterprise Replication, pokud nedojde k selhání primárního serveru.

Další informace naleznete v příručce *IBM Informix: Dynamic Server Enterprise Replication Guide*.

Jak pracuje replikace HDR

Tato část popisuje mechanismy, které databázový server používá k provádění replikace HDR. Pokyny, jak nastavit, spustit a spravovat systém replikace HDR uvádí Kapitola 21, “Použití replikace HDR”, na stránce 21-1 a postup inicializace páru replikace HDR pomocí externího zálohování a obnovení naleznete v příručce *IBM Informix: Backup and Restore Guide*.

Průběh prvotní replikace dat

Databázový server použije zálohy paměťových prostorů a logického protokolu (uložené na pásku nebo na disk), aby provedl počáteční replikaci dat primárního databázového serveru do sekundárního databázového serveru.

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.

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 21-7. Příručka *IBM Informix: Backup and Restore Guide* popisuje, jak inicializovat replikaci pomocí obslužného programu ON-Bar.

Počáteční inicializaci replikace HDR je třeba provést s použitím zálohy paměťových prostorů. K inicializaci nelze použít nástroje k migraci dat (například **onload** nebo **onunload**), protože aby replikace HDR mohla pracovat správně, musejí být rozvržení fyzických stránek jednotlivých tabulek v obou databázových serverech totožná.

Jakmile začne replikace HDR 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

Replikace HDR reprodukuje aktualizace primárního databázového serveru v sekundárním databázovém serveru tak, že primární databázový server okamžitě odesílá všechny vytvářené záznamy logického protokolu do sekundárního databázového serveru. Sekundární databázový server přijímá záznamy logického protokolu generované primárním databázovým serverem a používá je k aktualizaci vlastních prostorů dbspace.

Důležité: Databázový server nemůže replikovat aktualizace do databázi, které nepoužívají protokolování transakcí.

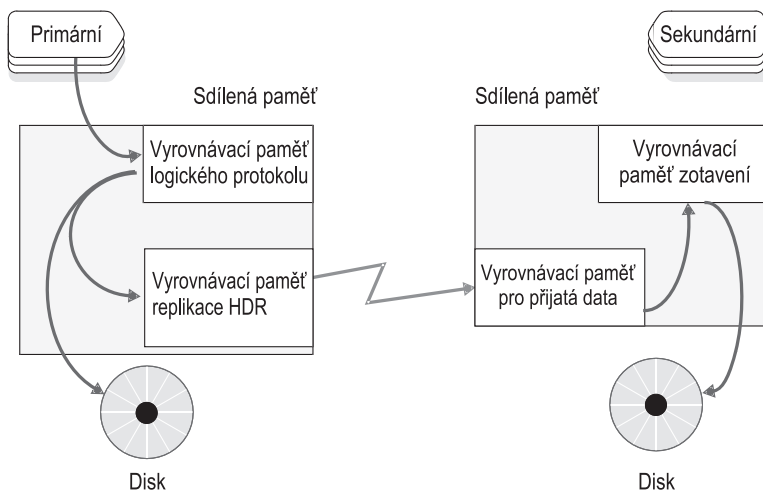
Jak jsou odesílány záznamy protokolu

Jak znázorňuje Obrázek 20-4 na stránce 20-9, 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.

Sekundární databázový server 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 logického zotavení.

Vyrovnávací paměti replikace HDR

Vyrovnávací paměti replikace HDR 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. Vyrovnávací paměti replikace HDR mají stejnou velikost, jako vyrovnávací paměť logického protokolu. Tento koncept znázorňuje Obrázek 20-4.



Obrázek 20-4. Jak primární databázový server odesílá záznamy logického protokolu do sekundárního databázového serveru

Kdy jsou záznamy odesílány

Primární databázový server odesílá obsah vyrovnávací paměti replikace HDR do sekundárního databázového serveru *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*.

Synchronní aktualizace

Pokud nastavíte parametr DRINTERVAL na hodnotu -1, bude replikace HDR probíhat *synchronně*. Jakmile databázový server zapíše obsah vyrovnávací paměti logického protokolu do vyrovnávací paměti replikace HDR, odešle tyto záznamy z vyrovnávací paměti replikace HDR do sekundárního databázového serveru. K vyprázdnění vyrovnávací paměti logického protokolu v primárním databázovém serveru dojde až po přijetí potvrzení o přijetí záznamů protokolu od sekundárního databázového serveru.

Pokud dojde k selhání při synchronní aktualizaci, žádné transakce potvrzené v primárním databázovém serveru nezůstanou v sekundárním databázovém serveru nepotvrzené ani částečně potvrzené.

Asynchronní aktualizace

Pokud nastavíte parametr DRINTERVAL na jinou hodnotu než -1, bude replikace dat probíhat *asynchronně*. Primární databázový server po zkopírování obsahu vyrovnávací paměti logického protokolu do vyrovnávací paměti replikace HDR vyprázdní

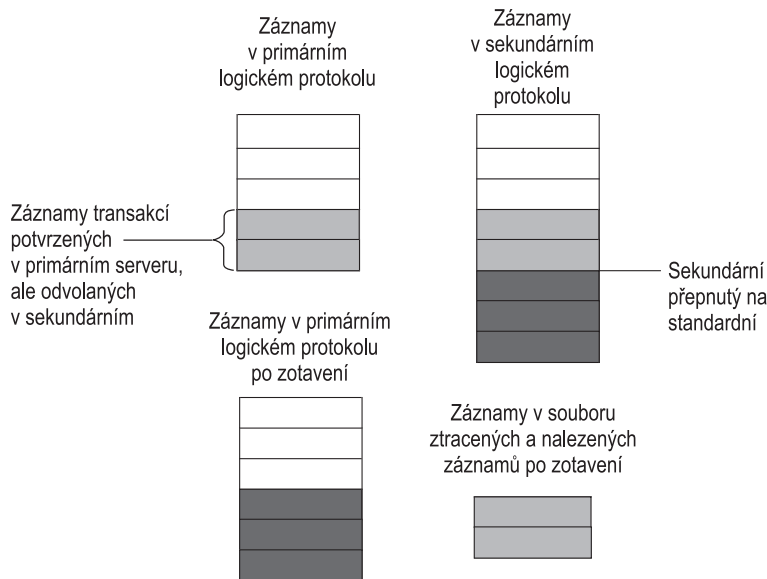
vyrovnávací paměť logického protokolu. Nezávisle na této akci odešle primární databázový server obsah vyrovnávací paměti replikace HDR prostřednictvím sítě, pokud nastane jeden z následujících stavů:

- 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 zotavení primárního databázového serveru všechny záznamy logického protokolu týkající se ztracených transakcí do souboru, jehož název určuje konfigurační parametr DRLOSTFOUND. Tento proces znázorňuje Obrázek 20-5.



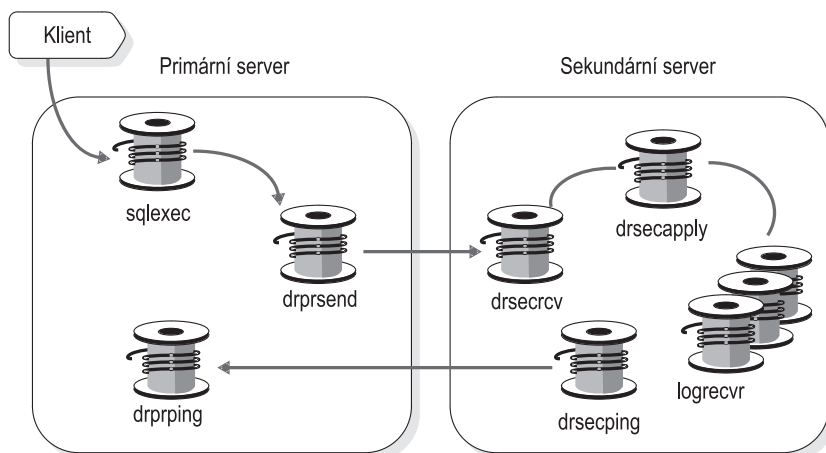
Obrázek 20-5. Použití souboru ztracených a nalezených transakcí

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, použijte 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 zpracovávají replikaci HDR

Databázový server spouští k podpoře replikace dat specializované jednotkové procesy. Jak znázorňuje Obrázek 20-6, jednotkový proces primárního databázového serveru nazývaný **drprsend** odesílá obsah vyrovnávací paměti HDR prostřednictvím sítě do jednotkového procesu sekundárního databázového serveru nazývaného **drsecrecv**.



Obrázek 20-6. Jednotkové procesy, které spravují replikaci dat

Jednotkový proces sekundárního databázového serveru zvaný **drsecapply** kopíruje obsah přijímací vyrovnávací paměti do vyrovnávací paměti zotavení. Jeden nebo více jednotkových procesů **logrecvr** provádí logické zotavení pomocí obsahu vyrovnávací paměti zotavení a používá záznamy logického protokolu k aktualizaci prostorů dbspace spravovaných databázovým serverem. Počet použitých jednotkových procesů **logrecvr** určuje konfigurační parametr `OFF_RECVRY_THREADS`.

Nakonec databázový server spouští k provozu replikace HDR také jednotkové procesy **drprping** a **drsecping**, které zodpovídají za odesílání a přijímání zpráv, které udávají, zda jsou oba databázové servery propojené.

Kontrolní body mezi databázovými servery

Kontrolní body mezi databázovými servery v replikačním páru jsou vždy synchronní, nezávisle na hodnotě parametru `DRINTERVAL`. (Viz “Kontrolní body” na stránce 16-8.) 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 20-13.

Tip: Synchronizujte čas v operačních systémech obou databázových serverů v páru a nastavte v obou serverech parametr `DRTIMEOUT` na shodnou hodnotu.

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.

Databázové servery interně používají tyto informace k synchronizaci replikace dat.

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.
- Přerušení 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.

Tip: 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, запиše 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 20-17.

- 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 20-14 a v části “Postup v případě selhání primárního databázového serveru” na stránce 20-15.

Pokud přesměrujete 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 přesměrování klientů používajících sekundární databázový server na primární databázový server použijte jakoukoli z metod popsanych v části “Přesměrování a propojitelnost pro klienty replikace dat” na stránce 20-17. Pokud přesměrujete 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 popsanych v části “Restartování po selhání sekundárního databázového serveru” na stránce 21-25.

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 logického zotavení. 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 přesměrovat klienty, naleznete v části “Přesměrování a propojitelnost pro klienty replikace dat” na stránce 20-17.

Automatické přepnutí má v porovnání s ručním přepnutím následující výhody:

- Klienti přesměrovaní 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 21-27 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 21-7. Proto pravděpodobně nebudete chtít používat automatické přepnutí, pokud síť není zcela stabilní.

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 21-18.

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 21-26.

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 21-25.

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. (Též může nastat situace, kdy nebudete chtít přesměrovat klienty. Pokud například předpokládáte, že databázové servery budou brzy opět v provozu, nemusí být přesměrování klientů vhodné.)

Databázový server není vybaven transparentním mechanismem přesměrování požadavků klientů na jiné databázové servery v replikačním páru, tuto akci však můžete automatizovat z prostředí aplikace, jak popisuje část “Přesměrování z aplikace” na stránce 20-23. 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 strategii 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 neobsahují, 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 20-20, pole nedostupného databázového serveru (s výjimkou pole **dbservername**) se změní tak, aby směřovala k druhému databázovému serveru v replikačním páru.

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 přípojovací 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 přípojovací 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.

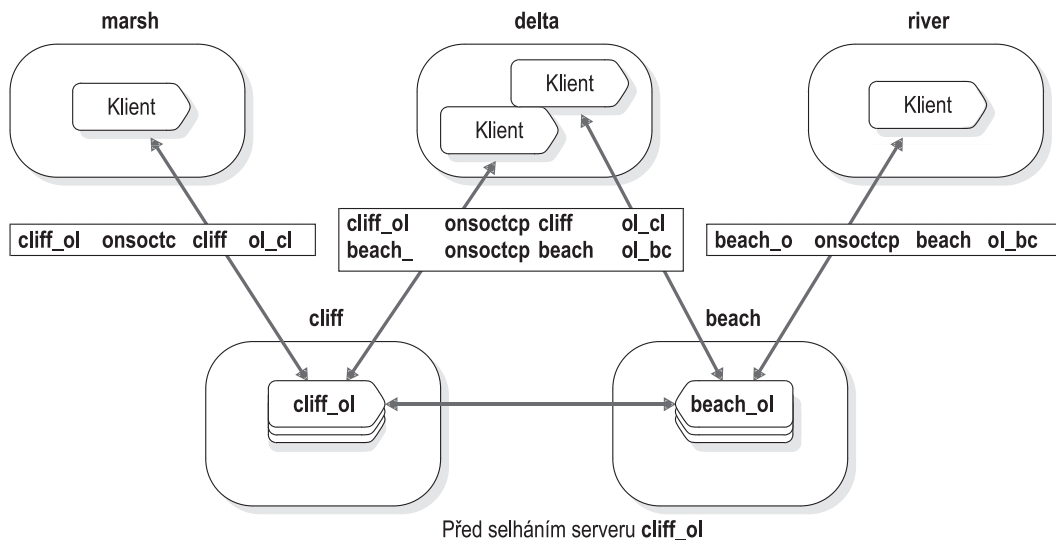
Další informace uvádí část “Konfigurace propojitelnosti replikace HDR” na stránce 21-6 a Kapitola 3, “Komunikace mezi klientem a serverem”, na stránce 3-1.

Změna informací o propojitelnost na klientském počítači:

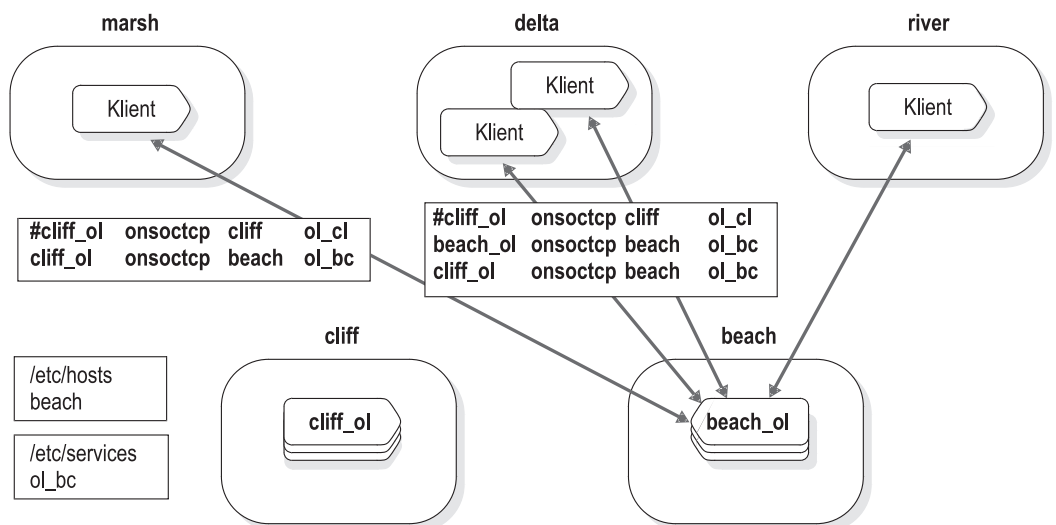
1. Uzavřete záznam databázového serveru s poruchou do poznámky.
2. Vložte záznam, kterým v poli **servername** určíte název dbservername databázového serveru, který selhal, a v polích **nettype**, **hostname**, a **servicename** určíte údaje databázového serveru, k němuž přesměrujete klienty.
3. Pokud dojde k selhání, použijte v souboru nebo registru **sqlhosts** k přesměrování aplikací na jiný databázový server následující volby:
 - a. “Volba přesměrování připojení” na stránce 3-25
 - b. “Volba konce skupiny” na stránce 3-26
 - c. “Volba skupiny” na stránce 3-27

Obrázek 20-7 na stránce 20-21 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.



Po selhání serveru `cliff_ol`



Obrázek 20-7. 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 musí obsahovat záznam pro název **hostname** počítače, na kterém je spuštěný databázový server, k němuž přeměrováváte klienty.

2. Soubor `/etc/services` v systému UNIX nebo soubor `services` v systému Windows musí obsahovat záznam názvu `servicename` databázového serveru, k němuž přesměrováváte klienty.

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.

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 přeměrovat klienty pomocí proměnné prostředí **INFORMIXSERVER**.

Přeměrování 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 20-8 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 20-8. 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_01";
```

V tomto příkladu odkazuje název **superstores@cliff_01** 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 20-2 shrnuje rozdíly mezi třemi mechanismy přesměrování.

Tabulka 20-2. 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 .
Musi 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 replikace HDR

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_btsend**. Jednotkový proces aplikace, který je spuštěn, může pokračovat ve zpracování. Jednotkový proces **dr_btsend** 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, selžou.
- Úrovně 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 **sqlwarn6** 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 **sqlwarn6** 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 **logrecovery**, které provádějí logické zotavení, 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 10-14 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 11-10.
- 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.

Kapitola 21. Použití replikace HDR

Plánování replikace HDR	21-2
Konfigurace systému pro replikaci HDR	21-3
Splnění požadavků na hardware a operační systém	21-3
Splnění požadavků na databázi a data	21-3
Splnění požadavků na konfiguraci databázového serveru	21-4
Verze databázového serveru	21-4
Konfigurace paměťových prostorů a bloků	21-4
Použití jiné než výchozí velikosti stránky v prostředí replikace HDR.	21-5
Zrcadlení	21-5
Konfigurace fyzického protokolu	21-5
Pásková zařízení pro zálohování prostorů dbspace a logického protokolu	21-5
Konfigurace logického protokolu	21-6
Konfigurační parametry replikace HDR	21-6
Konfigurace propojitelnosti replikace HDR	21-6
První spuštění replikace HDR	21-7
Zkrácení doby nastavení alternativní metodou zálohování.	21-11
Provádění základních úloh administrace	21-11
Změna konfiguračních parametrů databázového serveru	21-11
Zálohování paměťových prostorů a souborů logického protokolu	21-11
Změna režimu protokolování databázi	21-12
Přidání a vypuštění bloků a paměťových prostorů	21-12
Přejmenování bloků	21-12
Uložení stavu bloku na sekundární databázový server	21-13
Použití a změna zrcadlení bloků	21-14
Správa fyzického protokolu.	21-14
Správa logického protokolu.	21-15
Správa virtuálních procesorů	21-15
Správa sdílené paměti	21-15
Replikace indexu sekundárního databázového serveru	21-15
Změna režimu databázového serveru	21-16
Změna typu databázového serveru.	21-18
Monitorování stavu replikace HDR	21-18
Použití obslužných programů příkazového řádku	21-18
onstat -g dri	21-19
oncheck -pr	21-19
Použití tabulek SMI	21-20
Použití programu ON-Monitor (v systému UNIX)	21-20
Obnovení dat po selhání prostředku	21-20
Obnovení po selhání prostředku na primárním databázovém serveru	21-20
Obnovení po selhání prostředku na sekundárním databázovém serveru	21-21
Replikace indexu na sekundární server	21-22
Restartování replikace HDR po selhání	21-22
Restartování po poškození kritických dat.	21-23
Selhání kritického prostředku na primárním databázovém serveru	21-23

Selhání kritického prostředku na sekundárním databázovém serveru	21-24
Selhání kritického prostředku na obou databázových serverech	21-24
Restartování bez poškození kritických dat	21-24
Restartování po chybě sítě	21-25
Restartování po selhání sekundárního databázového serveru	21-25
Restartování po selhání primárního databázového serveru	21-25

Obsah kapitoly

Tato kapitola popisuje způsob plánování, konfigurace, spuštění a sledování replikace dat HDR (HDR) pro server Dynamic Server a způsob obnovy 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 20, “Replikace HDR (High-Availability Data Replication)”, na stránce 20-1 vysvětluje, co je replikace HDR, jak pracuje a jaké jsou možnosti návrhu klientských aplikací v prostředí replikace HDR.

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 13, “Správa režimu protokolování databáze”, na stránce 13-1.
- 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 20-17 a v části “Návrh klientů k použití sekundárního databázového serveru” na stránce 20-26.
- 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 20-17.

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 13-4.

- **Data musí být uložena v prostorech dbspace nebo sbpace.**

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 sbpace, jsou replikovány. Prostory sbpace musí být protokolovány. Uživatelské typy (UDT) jsou replikovány, pokud neukládají data mimo řádky do souborů operačního systému. Datové typy, které ukládají data mimo řádky, jsou replikovány, jestliže jsou data uložena v prostoru sbpace nebo v jiné tabulce téhož databázovém 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.

Doporučuje se, aby konfigurace obsahovala alespoň jeden dočasný prostor dbspace. Další informace naleznete v části “Použití dočasných prostorů dbspace k řazení a pro dočasné tabulky” na stránce 20-27.

Pouze 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 11-6.

Důležité: Pokud pro názvy cest bloků nepoužíváte symbolická propojení, nelze název bloku snadno změnit. Další informace naleznete v části “Přejmenování bloků” na stránce 21-12.

Konec Pouze pro UNIX

Následující parametry souboru ONCONFIG musí mít na obou databázových serverech stejnou hodnotu:

- 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 souboru ONCONFIG musí mít na obou databázových serverech stejnou hodnotu:

- 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 obou databázových serverech:

- 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 **ontlitcp**, **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 **onipeshm**, **onipestr** nebo **onipcnmp**.

Další informace o přesměrování klientů a změně propojitelnosti naleznete v části “Přesměrování a propojitelnost pro klienty replikace dat” na stránce 20-17.

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 21-1 na stránce 21-9 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.

Poznámka: 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 spuštění 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 logické zotavení 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 logickému zotavení 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 21-1. Postup prvního spuštění replikace HDR

Krok	Na primárním serveru	Na sekundárním serveru
1.	<p>Nainstalujte uživatelské rutiny, uživatelské typy (UDT) a moduly DataBlade.</p> <p>Zaregistrujte uživatelské rutiny, uživatelské typy a moduly DataBlade.</p>	<p>Nainstalujte uživatelské rutiny, uživatelské typy a moduly DataBlade.</p>
2.	<p>Příkaz ontape</p> <p>ontape -s -L 0</p> <p>Příkaz ON-Bar</p> <p>onbar -b -L 0</p> <p>Zprávy do protokolu zpráv</p> <p>Archivace úrovně 0 v prostoru rootdbs byla zahájena.</p> <p>Archivace v prostoru rootdbs byla dokončena.</p>	
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 zotavení Bylo spuštěno fyzické obnovení prostoru rootdbs. Bylo dokončeno fyzické obnovení prostoru rootdbs.</p>

Tabulka 21-1. Postup prvního spuštění replikace HDR (pokračování)

Krok	Na primárním serveru	Na sekundárním serveru
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 logické zotavení. (Jinak musí být proveden krok 5).</p>
	<p>Zprávy do protokolu zpráv</p> <p>DR: Primární server je 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 je připojen DR: Probíhá zotavení z disku po selhání. Počet pracovních jednotkových procesů zotavení, které budou spuštěny, je <i>n</i>. Bylo spuštěno logické zotavení Spustit logické zotavení - spustit protokol <i>n</i>, ukončit protokol? Počáteční umístění protokolu - <i>n 0xnnnnn</i>DR: Sekundární je server schopný provozu</p>
6.	<p>Zprávy do protokolu zpráv</p> <p>DR: Primární server byl připojen DR: Primární server je schopný provozu</p>	<p>Příkaz ontape</p> <p>ontape -l</p> <p>Příkaz programu ON-Bar</p> <p>onbar -r -l</p> <p>Zprávy do protokolu zpráv</p> <p>DR: Sekundární server byl připojen DR: Probíhá zotavení z disku po selhání. Počet pracovních jednotkových procesů zotavení, které budou spuštěny, je <i>n</i>. Bylo spuštěno logické zotavení Spustit logické zotavení - 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 obnovení 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 po spuštění replikace HDR v systému.

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 21-4.) Ostatní konfigurační parametry serveru Dynamic Server mohou být nastaveny na rozdílné hodnoty.

Provádění změn v souborech ONCONFIG:

1. Převedte 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 21-1 na stránce 21-9 a Tabulka 21-1 na stránce 21-9 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 21-7.

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 11-3.

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 přidáný na sekundární databázový server musíte provést zotavení zrcadla. Další informace naleznete v části “Obnovení zrcadleného bloku” na stránce 19-7.

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 19-8.

Zrcadlený blok můžete vypnout nebo zotavit 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. Parametry PHYSDBS a PHYSFILE v sekundárním souboru ONCONFIG však nejsou aktualizovány. Tyto parametry musíte v souboru ONCONFIG změnit ručně. Další informace o provedení těchto změn naleznete v části “Změna konfiguračních parametrů databázového serveru” na stránce 21-11.

Další informace o změně velikosti nebo umístění fyzického protokolu uvádí Kapitola 17, “Správa fyzického protokolu”, na stránce 17-1.

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 15, “Správa souborů logických protokolů”, na stránce 15-1. 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 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 změníte 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 21-11.

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, přenesete volba **onmode -d idxauto** pouze tento jeden dotčený fragment, zatímco volba **onmode -d index** přenesete 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 inicializován 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.

Změna režimu databázového serveru

Chcete-li změnit režim databázového serveru, spusíte obslužný program **onmode** v příkazovém řádku nebo v programu ISA. Další informace o obslužném programu **onmode** naleznete v kapitole o obslužných programech v příručce *IBM Informix: Dynamic Server Administrator's Reference*.

Tabulka 21-2 shrnuje důsledky změny režimu primárního databázového serveru.

Tabulka 21-2. Změny režimu primárního databázového serveru

Na primárním serveru	Na sekundárním serveru	Restartování replikace HDR
<p>Libovolný režim do režimu offline.</p> <p>(onmode -k)</p>	<p>Sekundární server zobrazí: DR: Chyba příjmu.</p> <p>Replikace HDR je vypnutá.</p> <p>Režim zůstane jen pro čtení.</p> <p>Je-li parametr DRAUTO nastaven na hodnotu 0 (OFF), zůstane server v režimu jen pro čtení.</p> <p>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.)</p>	<p>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:</p> <ul style="list-style-type: none"> • “Sekundární databázový server nebyl změněn na standardní databázový server” na stránce 21-25 • “Sekundární databázový server je automaticky změněn na standardní databázový server” na stránce 21-27
<p>Do režimu online, do klidového režimu nebo do jednouživatelského režimu.</p> <p>(onmode -s / onmode -u)</p> <p>(onmode -j)</p>	<p>Sekundární server nepřijímá chyby.</p> <p>Replikace HDR zůstane zapnutá.</p> <p>Režim zůstane jen pro čtení.</p>	<p>Použijte příkaz onmode -m na primárním serveru.</p>

Tabulka 21-3 shrnuje důsledky změn režimu sekundárního databázového serveru.

Tabulka 21-3. Změny režimu sekundárního databázového serveru

Na sekundárním serveru	Na primárním serveru	Restartování replikace HDR
<p>Jen pro čtení v režimu offline</p> <p>(onmode -k)</p>	<p>Primární server zobrazí: DR: Chyba příjmu.</p> <p>Replikace HDR je vypnutá.</p>	<p>Postupujte stejně jako při selhání sekundárního serveru. Postupujte podle pokynů v části “Restartování po selhání sekundárního databázového serveru” na stránce 21-25.</p>

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 serverem (**ServerB**) nachází v režimu online, spusíte v systému UNIX skript **hdrmkpri.sh** nebo v systému Windows skript **hdrmkpri.bat**. Server **ServerB** je nyní primárním databázovým serverem.
3. Na serveru **ServerA** spusíte v systému UNIX skript **hdrmksec.sh** nebo v systému Windows skript **hdrmksec.bat**. Server **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, spusíte 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.

Obrázek 21-1 znázorňuje příklad výstupu. Tento příklad znázorňuje primární databázový server párování se sekundárním databázovým serverem, který má parametr DBSERVERNAME nastaven na hodnotu **beach_ol**. Replikace HDR byla spuštěna.

```
Data Replication:
Type           State           Paired server      Last DR CKPT (id/pg)
primary        off             beach_ol           4/741

DRINTERVAL    30
DRTIMEOUT     300
DRLOSTFOUND   /usr/informix/etc/dr.lostfound
```

Obrázek 21-1. Výstup příkazu `onstat -g dri`

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 21-2.

```
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 21-2. 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 (v systému UNIX)

Pomocí příkazu **Status > Replication** můžete zobrazit informace o replikaci HDR. Tato volba zobrazí stejné informace jako volba **onstat -g dri**.

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 21-4 na stránce 21-21 uvádí přehled různých scénářů pro obnovení dat v případě selhání prostředku na primárním databázovém serveru. Podstatné jsou následující skutečnosti:

- Pokud jsou bloky zrcadleny, můžete zotavení provést stejným způsobem, jako v případě standardního databázového serveru, který používá zrcadlení.

- Pokud bloky nejsou zrcadleny, závisí postup při zotavení 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 21-23.

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 21-4. Scénáře pro selhání prostředků na primárním databázovém serveru

Server replikace HDR	Důležité prostředky	Bloky jsou zrcadleny	Dopad selhání a postup při obnovení prostředků
Primární	Ano	Ne	Primární databázový server selže. Postupujte podle pokynů v části “Restartování po poškození kritických dat” na stránce 21-23.
Primární	Ano	Ano	Primární databázový server zůstane v režimu online. Postupujte podle pokynů v části “Obnovení zrcadleného bloku” na stránce 19-7.
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 19-7.

Obnovení po selhání prostředku na sekundárním databázovém serveru

Replikace HDR: Tabulka 21-5 uvádí přehled různých scénářů pro obnovení dat v případě selhání prostředku na sekundárním databázovém serveru. Podstatné jsou následující skutečnosti:

- Pokud jsou bloky zrcadleny, můžete zotavení 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 21-23.) 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 21-5. Scénáře pro selhání prostředků na sekundárním databázovém serveru

Server 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 21-23.
Sekundární	Ano	Ano	Sekundární databázový server zůstane online v režimu jen pro čtení. Postupujte podle pokynů v části “Obnovení zrcadleného bloku” na stránce 19-7.
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 21-23.
Sekundární	Ne	Ano	Sekundární databázový server zůstane online v režimu jen pro čtení. Postupujte podle pokynů v části “Obnovení zrcadleného bloku” na stránce 19-7.

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 DRIDX AUTO. Hodnotu konfiguračního parametru DRIDX AUTO 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 20-13. 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 21-6 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 logické zotavení primárního databázového serveru ze souborů logického protokolu na disku sekundárního databázového serveru. Nelze-li logické zotavení 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 zotavení souborů logického protokolu z pásky. Tato zpráva je uložena v protokolu zpráv. Po zotavení všech požadovaných souborů logického protokolu z pásky budou obnoveny všechny zbývající soubory logického protokolu na sekundárním disku.

Tabulka 21-6. 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>
2.	Příkaz programu ON-Bar onbar -r -p Příkaz ontape ontape -p	
3.	Obslužný program onmode onmode -d primary <i>název_sek</i>	
4.	Příkaz ontape ontape -l	

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 21-7.

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 21-7.) 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 sekund 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 21-7. Tento postup předpokládá, že jste od selhání sekundárního databázového serveru zálohovali soubory logického protokolu na primárním databázovém serveru podle potřeby.

Tabulka 21-7. 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 zotavení z pásky po selhání
2.		Příkaz ontape ontape -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 změnil jste sekundární databázový server na standardní databázový server, postupujte podle pokynů, které uvádí Tabulka 21-9.

Tabulka 21-8. Postup restartování po změně sekundárního databázového serveru na standardní

Krok	Na primárním databázovém serveru	Na sekundárním databázovém serveru
1.		<p>onmode -s</p> <p>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 přesměrovány na primární databázový server. Další informace naleznete v části "Přesměrování a propojitelnost pro klienty replikace dat" na stránce 20-17.</p>
2.		<p>onmode -d secondary <i>název_prim</i></p>
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 zálohovali 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 obnovení záznamů logického protokolu z pásky, proveďte tento krok.</p> <p>Příkaz ontape</p> <p>ontape -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 20-15), postupujte podle pokynů v následující tabulce.

Tabulka 21-9. Postup restartování po automatické změně sekundárního databázového serveru na standardní

Krok	Na primárním databázovém serveru	Na sekundárním databázovém serveru
1.	<p>% onit</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 inicializaci 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 zálohovali 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 20-17.</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>
2.	<p>Budete-li vyzváni k obnovení záznamů logického protokolu z pásky, proveďte tento krok.</p> <p>příkaz ontape</p> <p>% ontape -l</p>	

Kapitola 22. Kontrola konzistence

Provádění periodické kontroly konzistence	22-2
Ověření konzistence	22-2
Ověření tabulek systémového katalogu.	22-3
Ověření datových stránek.	22-3
Ověření oblastí	22-3
Ověření indexů.	22-3
Ověření logických protokolů.	22-3
Ověření rezervovaných stránek	22-3
Ověření metadat	22-4
Monitorování nekonzistence dat.	22-4
Čtení chyb aserce v protokolu zpráv a souborech výpisu	22-4
Ověření dat v tabulkách a tabulkových prostorech	22-5
Zachování konzistentních záloh úrovně 0	22-6
Co provést v případě poškození.	22-6
Zjištění příznaků poškození	22-6
Oprava poškození indexu.	22-6
Oprava chyb vstupu-výstupu bloku	22-7
Shromažďování diagnostických dat	22-8
Chyby vstupu-výstupu bránící v provozu	22-9
Monitorování chyb vstupu - výstupu bránících v provozu v databázovém serveru.	22-9
Používání protokolu zpráv ke sledování chyb vstupu-výstupu bránících v provozu	22-10
Používání alarmů událostí k monitorování chyb vstupu-výstupu bránících v provozu.	22-10
Používání vypnutého mapování chybných sektorů	22-11

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

Pokud chcete dosáhnout maximálního užítku z kontroly konzistence a zajistit integritu zálohování prostoru dbspace, doporučujeme 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. Doporučujeme provést tuto kontrolu před vytvořením zálohy úrovně 0.

Jako součást kontroly konzistence spusíte příkazy, které uvádí Tabulka 22-1.

Tabulka 22-1. 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 neznamenaají 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 **sysstable**. Například při ověřování tabulek systémového katalogu pro databázi, která nemá definovaná synonyma v žádné její tabulce, se může zobrazit následující upozornění:

```
WARNING: No sysstable 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 zpráva 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 22-1 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 22-1. Forma chyb aserce v protokolu zpráv

Řádek **See Also:** obsahuje jeden nebo více následujících názvů souborů:

- **af.xxx**
- **shmem.xxx**

Pouze pro UNIX

- **gcore.xxx**
- **gcore.xxx**
- **/název cesty/core**

Konec Pouze 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.

Pouze pro UNIX

V operačním systému UNIX obsahuje soubor **gcore.xxx** snímek stavu virtuálního procesu databázového serveru, na kterém byl v okamžiku výskytu chyby spuštěn jednotkový proces, ale jen pokud je hodnota parametru DUMPGCORE v souboru ONCONFIG nastavena na 1 a pokud operační systém podporuje obslužný program **gcore**. Soubor **core** obsahuje snímek stavu virtuálního procesu databázového serveru, na kterém byl v okamžiku výskytu chyby spuštěn jednotkový proces, ale jen pokud je hodnota parametru DUMPCORE v souboru ONCONFIG nastavena na 1. Název cesty pro soubor **core** je adresář, ze kterého byl databázový server naposledy spuštěn.

Konec Pouze 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 22-2, pokud kontroly nezaznamenají žádné chyby. Zachovejte tuto zálohu paměťového prostoru a všechny následující záložní pásy logických protokolů do doby, než provedete další kontrolu konzistence. Doporučujeme provést kontrolu konzistence 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 22-2.
- 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.

Obrázek 22-2 uvádí příklad, ve kterém má stav bloku 2 hodnotu vypnuto.

```
Dbspaces
address number flags fchunk nchunks flags owner name
40c980 1 1 1 1 N informix rootdbs
40c9c4 2 1 2 1 N informix fstdbs
2 active, 8192 maximum

Chunks
address chk/dbs offset size free bpages flags pathname
40c224 1 1 0 20000 14001 PO- /home/server/root_chunk
40c2bc 2 2 0 2000 1659 PD- /home/server/fst_chunk
2 active, 8192 maximum
```

Obrázek 22-2. Výstup příkazu onstat -d

Protokol zpráv navíc 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 19-7.

Není-li poškozený blok zrcadlený a obsahuje-li soubory logického protokolu, fyzický protokol nebo kořenový paměťový prostor dbspace, přeruší databázový server okamžitě činnost. 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 zjistíte, 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 znovu inicializovat, pokud blok označený jako vypnutý neobsahuje kritická data (soubory logického protokolu, fyzický protokol nebo kořenový prostor dbspace).

Shromažďování 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** pošlete pracovníkům technické podpory systému IBM Informix e-mailem na adresu tmail@us.ibm.com. Soubor **af.xxx** často obsahuje informace, které tým technické podpory potřebuje k vyřešení potíží.

Jaká diagnostická data má databázový server zachovat, pokud dojde k chybě aserce nebo pokud bude server ukončen nestandardním způsobem, určuje několik parametrů v souboru ONCONFIG:

Pouze pro UNIX

- DUMPDIR,
- DUMPSHMEM,
- DUMPCNT,
- DUMPCORE,
- DUMPGCORE.

Konec Pouze 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.

Tip: 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. Zaprvé, 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.
- Otevírání 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:

- pomocí protokolu zpráv,
- pomocí alarmů událostí.

Používání protokolu zpráv ke sledování 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, popsáném 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 musí ukončit činnost.	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.

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_dbpace'
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 ukončí činnost. 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 23. Protokoly vícefázového potvrzování

Správci transakcí	23-2
Použití knihovny TP/XA se správcem transakcí	23-2
Použití serveru MTS/XA (Microsoft Transaction Server)	23-3
Použití volně vázaného režimu a provázaného režimu	23-3
Protokol dvoufázového potvrzování	23-4
Kdy se používá protokol dvoufázového potvrzování	23-4
Koncepte dvoufázového potvrzování	23-5
Fáze protokolu dvoufázového potvrzování.	23-6
Fáze před potvrzením	23-6
Fáze po rozhodnutí	23-6
Jak protokol dvoufázového potvrzování ošetřuje selhání	23-6
Typy selhání ošetřené automatickým zotavením	23-6
Role administrátora v automatickém zotavení	23-7
Mechanismus automatického zotavení při selhání koordinátora	23-7
Mechanismus automatického zotavení při selhání účastníka	23-7
Optimalizace předpokládaného přerušení	23-7
Nezávislé akce	23-8
Situace, které vyvolávají nezávislé akce	23-8
Možné důsledky nezávislých akcí	23-8
Nezávislé akce, které umožní úspěšné dokončení transakcí	23-8
Nezávislé akce, které vyústí v chybový stav	23-9
Nezávislé akce, jejichž důsledkem je heuristické rozhodnutí	23-9
Scénář heuristického odvolání transakce	23-10
Podmínky, jejichž výsledkem je heuristické odvolání transakce	23-10
Výsledky heuristického odvolání transakce	23-11
Scénář heuristického ukončení transakce	23-13
Kdy provést heuristické ukončení transakce	23-13
Jak používat příkaz onmode -Z.	23-14
Akce při heuristickém ukončení transakce	23-14
Monitorování globální transakce	23-15
Chyby protokolu dvoufázového potvrzování.	23-15
Dvoufázové potvrzování a záznamy logického protokolu	23-16
Záznamy logického protokolu při potvrzení transakce	23-16
Záznamy logického protokolu zapsané během heuristického odvolání transakce	23-18
Záznamy logického protokolu zapsané během heuristického ukončení transakce	23-19
Konfigurační parametry používané při dvoufázovém potvrzování	23-21
Funkce parametru DEADLOCK_TIMEOUT	23-21
Funkce parametru TXTIMEOUT	23-21
Protokol heterogenního potvrzování	23-21
Brány, které se mohou účastnit heterogenního potvrzování transakcí	23-22
Povolení a zakázání heterogenního potvrzování	23-23
Jak funguje heterogenní potvrzování	23-23
Fáze před potvrzením	23-24
Fáze potvrzení brány.	23-24

Optimalizace heterogenního potvrzování	23-24
Důsledky selhání heterogenního potvrzování	23-25
Selhání koordinačního databázového serveru	23-25
Selhání účastníka	23-25
Interpretace chybových zpráv heterogenního potvrzování	23-26

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.

Další informace o ruční obnově při selhání dvoufázového potvrzování uvádí Kapitola 24, “Ruční obnova při selhání dvoufázového potvrzování”, na stránce 24-1.

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 aplikaci,
- správcem zdrojů (databázovým serverem Informix),
- správcem transakcí (softwarem 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 a 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 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 Tuxedo pracuje v operačním systému UNIX i v operačním systému 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

Chcete-li získat úplný seznam podporovaných správců transakcí, obraťte se na svého obchodního zástupce.

Protokol dvoufázového potvrzování

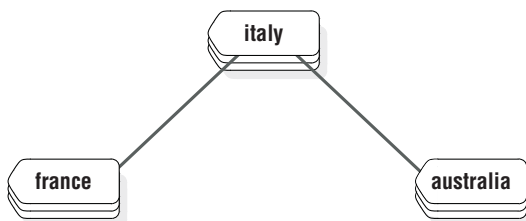
Protokol dvoufázového potvrzování poskytuje automatický mechanismus obnovy v případě selhání systému nebo média během vykonávání transakce. Protokol dvoufázového potvrzování zajistí, že všechny zúčastněné databázové servery obdrží a implementují totožnou akci (potvrzení transakce nebo její odvolání), 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**, and **france**, jsou propojeny způsobem, který ilustruje Obrázek 23-1.



Obrázek 23-1. Propojené databázové servery

Pokud spustíte příkaz, který uvádí Obrázek 23-2, 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 23-2. 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 by globální transakce měla být potvrzena nebo přerušena.

Protokol dvoufázového potvrzování vždy přiřazuje roli koordinátora *aktuálnímu* databázovému serveru. Roli koordinátora nelze v rámci jedné transakce změnit. Ve vzorové transakci, kterou uvádí Obrázek 23-2 na stránce 23-5, je koordinátorem server **italy**. Pokud změníte 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 23-15.

- 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 23-1 na stránce 23-4 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 23-16.

Fáze protokolu dvoufázového potvrzení

Při transakci s dvoufázovým potvrzováním odesílá koordinátor všechny instrukce změn dat (například vložení) všem účastníkům. Potom koordinátor spustí protokol dvoufázového potvrzení. Protokol dvoufázového potvrzení 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 jakýkoli účastník oznámí, že *není* připraven potvrdit svou větev transakce (nebo pokud neodpovídá), rozhodne se koordinátor globální transakci přerušit.

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 potvrzení ošetřuje selhání

Protokol dvoufázového potvrzení je navržen k ošetření selhání systému a médií takovým způsobem, aby byla integrita dat zachována ve všech účastnických databázových serverech. Protokol dvoufázového potvrzení provádí v případě selhání *automatické zotavení*.

Typy selhání ošetřené automatickým zotavením

Následující události mohou způsobit ukončení nebo zastavení koordinačního nebo účastnického jednotkového procesu, a tudíž vyžadují automatické zotavení:

- Selhání systému koordinátora.
- Selhání systému účastníka.
- Selhání sítě.

- Ukončení koordinačního jednotkového procesu administrátorem.
- Ukončení účastnického jednotkového procesu administrátorem.

Role administrátora v automatickém zotavení

Jediná úloha, kterou má administrátor v automatickém zotavení, 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. Toto rozhodnutí je součástí optimalizace předpokládaného přerušení. (Další informace naleznete v části “Optimalizace předpokládaného přerušení” na stránce 23-7.)

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.

Optimalizace předpokládaného přerušení

Optimalizací předpokládaného přerušení se rozumí způsob, jakým protokol dvoufázového potvrzování řeší odvolání transakce (přerušení).

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 potvrzování akce, která se uskuteční nezávisle na protokolu dvoufázového potvrzování. Nezávislé akce mohou, avšak zároveň nemusejí působit v protikladu k akcím, které určuje protokol dvoufázového potvrzování. Pokud akce *působí* v protikladu k protokolu dvoufázového potvrzování, bude výsledkem akce chyba nebo *heuristické rozhodnutí*. Heuristická rozhodnutí mohou mít za následek nekonzistenci databáze a vyžadují ruční obnovu po nezdařeném dvoufázovém potvrzování. Ruční obnova je velmi složitá administrativní procedura, které byste se měli vyhnout. (Další informace o procesu ruční obnovy uvádí Kapitola 24, “Ruční obnova při selhání dvoufázového potvrzování”, na stránce 24-1.)

Situace, které vyvolávají nezávislé akce

Nezávislé akce jsou během protokolu dvoufázového potvrzování vzácné, ale mohou se vyskytnout v následujících situacích:

- Část práce účastníka se vyvine v chybu dlouhé transakce a je odvolána.
- Administrátor odstraní účastnický jednotkový proces během fáze protokolu po rozhodnutí pomocí příkazu **onmode -z**.
- Administrátor během fáze protokolu po rozhodnutí odstraní transakci účastníka (část práce) příkazem **onmode -Z**.
- Administrátor odstraní globální transakci na koordinačním databázovém serveru pomocí příkazu **onmode -z** nebo **onmode -Z poté**, co koordinátor vydal rozhodnutí o potvrzení a dozvěděl se 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 potvrzování. Nezávislé akce mohou přinášet tyto možné výsledky:

- Úspěšné dokončení protokolu dvoufázového potvrzování.
- 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 administrátor databázového serveru spustíte příkaz , **onmode -z** (odstranění koordinačního jednotkového procesu) nebo **onmode -Z** (odstranění globální transakce) poté, co koordinátor vydá závěrečné rozhodnutí o *potvrzení* transakce, odstraníte veškeré informace o transakci ze sdílené paměti v koordinačním databázovém 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, uposlechne koordinační jednotkový proces požadavek na ukončení, ale předtím zapíše do protokolu zpráv databázového serveru chybu číslo -716. Tato akce je považována za chybu, protože protokol dvoufázového potvrzování byl vynuceně přerušen a koordinátor proto nemohl určit, zda je databáze konzistentní.

Odstranění globální transakce na koordinačním databázovém serveru není považováno za heuristické rozhodnutí, jeho důsledkem však může být 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 23-10)
- Heuristické ukončení transakce (popsané v části “Scénář heuristického ukončení transakce” na stránce 23-13)

Pokud dojde k heuristickému odvolání transakce nebo k heuristickému ukončení transakce, budete pravděpodobně muset provést ruční obnovu, což je složitý a časově náročný proces. Abyste se mohli heuristickým rozhodnutím vyhnout, musíte jim zcela porozumět. Při spuštění 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 spustí příkaz **onmode -z id_relace**, aby odstranil jednotkový proces databázového serveru, který provádí část práce za 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 vylučným 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 příkazem `onmode -z` odstraníte účastnický jednotkový proces, uvolníte všechny zámky a zdroje sdílené paměti zadržované účastnickým jednotkovým procesem, ale 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 zapíše do vyrovnávací paměti logického protokolu zprávu ENDTRANS. (Záznam transakce je odstraněn z tabulky transakcí.)
4. Koordinátor zapíše 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. Number and names of servers rolled back.
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 24-1.)

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 funkcí automatického zotavení. Pokud spustíte příkaz **onmode -Z**, budete pak sami zodpovědní 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 vylučným 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 odstraníte 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 naleznete v kapitole týkající se obslužných programů v příručce *IBM Informix: Administrator's Reference*.

Parametr *adresa* získáte z výstupu příkazu **onstat -x**. Další informace o příkazu **onstat -x** naleznete v kapitole zaměřené na obslužné programy v příručce *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 *tyid*

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, nevytvorí 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. Obrázek 23-3 ilustruje heuristické odvolání dlouhé transakce na adrese a733748.

Průznak H na třetí pozici pole **flags** znamená heuristické odvolání transakce a průznak G na páté pozici znamená globální transakci. Průznak na druhé pozici označuje režim, ve kterém je transakce spuštěna (průznak L označuje volně vázaný režim a průznak 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. Tento záznam je nachází v protokolu 20 na stránce s posunem 2 (třetí stránka v protokolu) od bajtu s posunem 0x218. 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.

```
IBM Informix Dynamic Server      Version 9.40.U      -- Online (LONGTX)

Transactions
address  flags  userthread  locks  beginlg  curlog  logposit  isol  retrys  coord
a733018  A----  a701018    0      0        20     0x11a0    COMMIT  0
a7331e4  A----  a701638    0      0         0     0x0      COMMIT  0
a7333b0  A----  a701c58    0      0         0     0x0      COMMIT  0
a73357c  A----  a702278    0      0         0     0x0      COMMIT  0
a733748  --H-G  0          0      17        20     0x2218    COMMIT  0
a733ae0  A----  a7034d8    0      0         0     0x0      COMMIT  0
  6 active, 128 total, 10 maximum concurrent
```

Obrázek 23-3. Výstup příkazu **onstat -x**

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 *IBM Informix: Performance Guide*. Popis polí zobrazovaných příkazem **onstat -x** naleznete v kapitole zaměřené na obslužné programy v příručce *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 23-11. 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 23-11. 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 23-9.

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 24, “Ruční obnova při selhání dvoufázového potvrzování”, na stránce 24-1.)

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*.

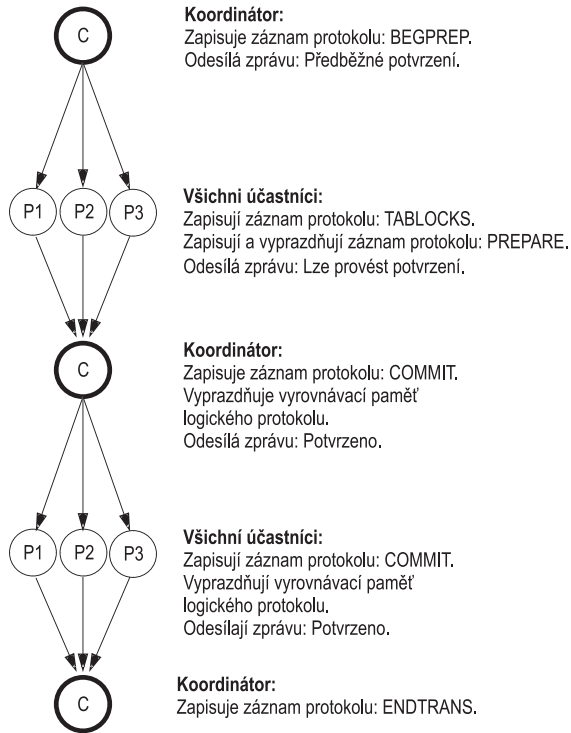
V této části je popsáno pořadí záznamů logického protokolu, které se zapisují během následujících scénářů databázového serveru:

- Transakce je potvrzena.
- Část práce je heuristicky odvolána.
- Část práce je heuristicky ukončena.

Záznamy logického protokolu při potvrzení transakce

Obrázek 23-4 ilustruje pořadí zápisu záznamů logického protokolu během úspěšného protokolu dvoufázového potvrzování, který vyústí v potvrzenou transakci.

Zahájení protokolu



Ukončení protokolu

Obrázek 23-4. Záznamy logického protokolu zapsané během potvrzené transakce

Některé záznamy logického protokolu musí být vyprázdněny z logického protokolu okamžitě, zatímco vyprázdnění jiných není životně důležité.

Koordinátorův záznam 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.

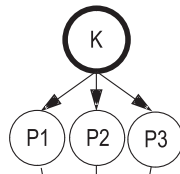
Účastníci, které uvádí Obrázek 23-4 na stránce 23-17, musí okamžitě vyprázdnit záznamy logického protokolu PREPARE a COMMIT. Vyprázdnění záznamu PREPARE zajišťuje, že v případě poruchy hostitelského počítače účastníka bude rychlá obnova schopná zjistit, že je tento účastník částí globální transakce. 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 23-5 na stránce 23-19 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 23-4 na stránce 23-17. 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 23-10.

Zahájení protokolu



Koordinátor:

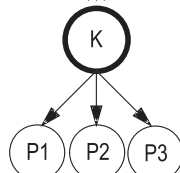
Zapíše záznam protokolu: BEGPREP.
Odesílá zprávu: *Předběžné potvrzení*.

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.
Zapíše záznam protokolu: HEURTX.
Zapíše záznam protokolu: ROLLBACK.
Do protokolu zpráv je zapsána zpráva.



Koordinátor:

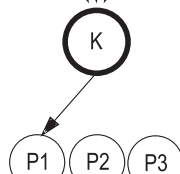
Zapíše záznam protokolu: COMMIT.
Vyprazdňuje záznam protokolu.
Odesílá zprávu: *Potvrzeno*.

Účastník 1:

Odesílá zprávu: *Transakce byla heuristicky odvolána. Nelze potvrdit*.

Účastníci 2 a 3:

Zápis a vyprázdnění záznamu protokolu: COMMIT.
Odesílají zprávu: *Potvrzeno*.

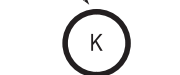


Koordinátor:

Zapíše zprávu do protokolu zpráv (-698).
Odesílá zprávu účastníkovi 1: *Konec transakce*.

Účastník 1:

Zapíše záznam protokolu: ENDTRANS.
Odesílá zprávu: *Transakce byla ukončena*.



Koordinátor:

Zapíše záznam protokolu: ENDTRANS.
Vrací uživateli chybovou zprávu: *Chyba -698*.

Ukončení protokolu

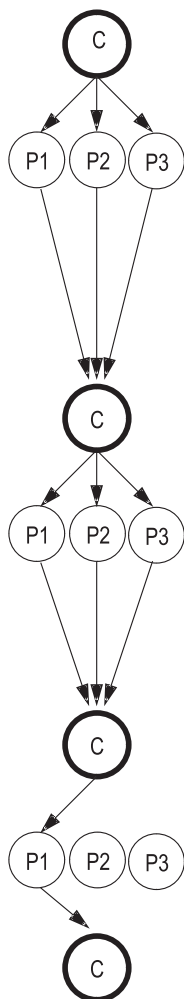
Obrázek 23-5. Záznamy logického protokolu zapsané během heuristického odvolání transakce

Záznamy logického protokolu zapsané během heuristického ukončení transakce

Obrázek 23-6 na stránce 23-20 ilustruje pořadí zápisu záznamů logického protokolu během heuristického ukončení transakce. Tato událost je vždy důsledkem odstranění transakce na účastnickém databázovém serveru, které provedl administrátor (další informace naleznete v části týkající se příkazu **onmode** v kapitole o obslužných

programech v příručce *IBM Informix: Administrator's Reference*) poté, co účastník odeslal zprávu `can commit` (lze potvrdit). Obrázek 23-6 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 23-13.

Zahájení protokolu



Koordinátor:

Zapíše záznam protokolu: BEGPREP.
Odesílá zprávu: *Předběžné potvrzení*.

Všichni účastníci:

Zapíší záznam protokolu: TABLOCKS.
Zapíší a vyprazdňují záznam protokolu: PREPARE.
Odesílají zprávu: *Lze provést potvrzení*.

Prostředí účastníka P1:

Transakce byla přerušena.
Zapíše záznam protokolu: ROLLBACK.
Zapíše záznam protokolu: ENDTRANS.
Do protokolu zpráv je zapsána zpráva.

Koordinátor:

Zapíše záznam protokolu: COMMIT.
Vyprazdňuje vyrovnávací paměť logického protokolu.
Odesílá zprávu: *Potvrzeno*.

Účastník 1:

Vrací chybovou zprávu.

Účastníci 2 a 3:

Zapíší 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.

Participant 1:

Odesílá koordinátorovi zprávu o neznámém stavu transakce.

Koordinátor:

Předpokládá, že *neznámý stav* znamená *potvrzení*.
Zapíše záznam protokolu: ENDTRANS.

Ukončení protokolu

Obrázek 23-6. Záznamy logického protokolu zapsané během heuristického ukončení transakce

Konfigurační parametry používané při dvoufázovém 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 přeruší 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 23-6.

Protokol heterogenního potvrzování

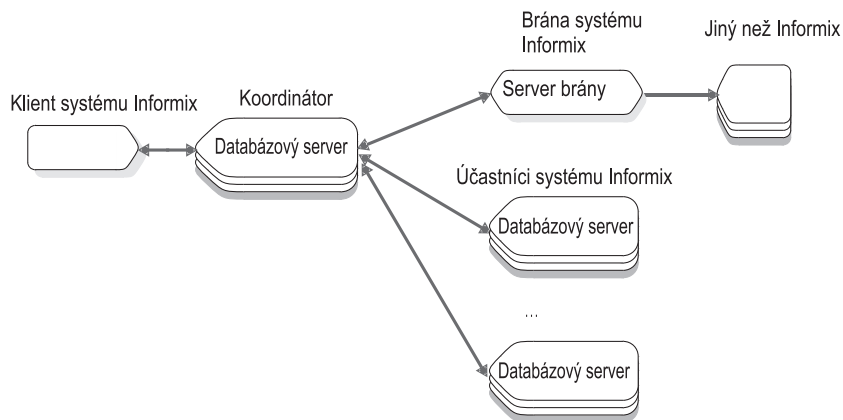
Termín heterogenní prostředí, používaný v souvislosti s databázovými servery Informix, označuje skupinu databázových serverů, z nichž alespoň jeden není databázový server Informix. Heterogenní potvrzování zajišťuje, že se distribuované transakce v heterogenním prostředí řídí principem "všechno nebo nic".

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 Gateway.

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 23-7.



Obrázek 23-7. Konfigurace, která vyžaduje heterogenní potvrzování distribuovaných transakcí.

Brány, které se mohou účastnit heterogenního potvrzování 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 potvrzování.

Tabulka 23-1. Brány a odpovídající databázové servery/heterogenní potvrzení transakcí

Brána	Databázové servery
IBM Informix Enterprise Gateway s architekturou 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 potvrzení
Zakázán	Ano	dvoufázové potvrzení
Povolen	Ne	dvoufázové potvrzení
Povolen	Ano	heterogenní potvrzení

Jak funguje heterogenní potvrzení

Protokol heterogenního potvrzení je pozměněnou verzí standardního protokolu dvoufázového potvrzení. Fáze po rozhodnutí v protokolu heterogenního potvrzení je stejná jako fáze po rozhodnutí v protokolu dvoufázového potvrzení. Fáze před potvrzením obsahuje malé změny, a dále je k protokolu heterogenního potvrzení 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 23-6.

Fáze před potvrzením

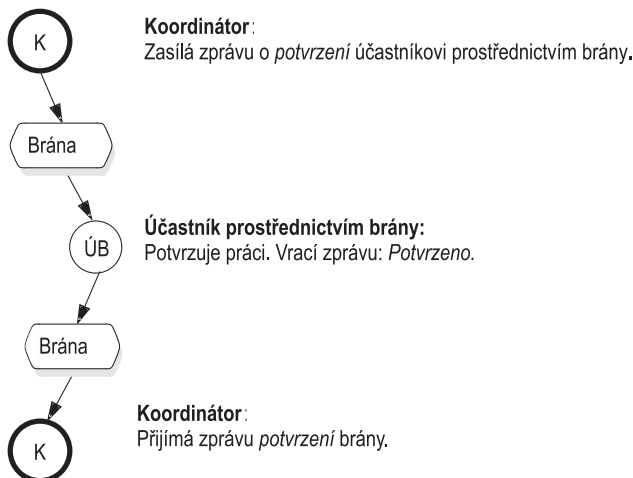
Koordinátor nařídí každému účastníkovi (kromě účastníka brány), aby se připravil na potvrzení transakce.

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 23-8 na stránce 23-24.

Zahájení fáze potvrzení prostřednictvím brány



Konec fáze potvrzení prostřednictvím brány

Obrázek 23-8. Fáze heterogenního potvrzování, jejímž výsledkem je potvrzená transakce

Pokud brána selže při potvrzení transakce, koordinátor odvolá celou transakci, tak jak to ilustruje Obrázek 23-8.

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 přerušena, 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 přerušeny. Toto přerušení 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 запиše 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 запиše 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 Possible inconsistent data at the target DBMS název due to an aborted commit.
```

Navíc odešle databázový server do protokolu zpráv tuto zprávu:

```
Data source accessed using gateway název might be in an inconsistent state.
```

Selhání účastníka se neomezuje pouze na selhání databázového serveru nebo brány. Selhání komunikačního propojení mezi koordinátorem a branou 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 23-25.

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 24. Ruční obnova při selhání dvoufázového potvrzování

Určení, zda je požadována ruční obnova	24-1
Určení, zda byla transakce implementována nekonzistentně	24-1
Předčasné odstranění globální transakce	24-2
Heuristické ukončení transakce	24-2
Heuristické odvolání transakce	24-2
Určení, zda distribuovaná databáze obsahuje nekonzistentní data	24-3
Získání informací z logického protokolu	24-3
Získání identifikátoru globální transakce	24-4
Rozhodnutí, zda je nezbytné k nápravě situace provést akci	24-5
Příklad ruční obnovy	24-6

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 23-8.) 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

Vaším prvním úkolem je určit, zda byla transakce v důsledku nezávislé akce implementována nekonzistentně.

Předčasné odstranění globální transakce

Pokud jste spuštěním příkazu **onmode -z** na koordinátorovi odstranili globální transakci, může být transakce implementována nekonzistentně. (Vysvětlení okolností vzniku této situace naleznete v části “Nezávislé akce, které vyústí v chybový stav” na stránce 23-9.) Zjištění nekonzistentní transakce provedete nejdříve přezkoumáním protokolu zpráv databázového serveru pro koordinátora. Hledejte následující chybovou zprávu:

```
-716 Possible inconsistent transaction.  
Unknown servers are seznam_názvů_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 23-13.) 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 запиše 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. Zrekonstruuje 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** získáte 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 24-4.
3. Z protokolu koordinátora získáte 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 pročtete logický protokol a vyhledejte záznam PREPARE obsahující identifikátor GTRID přidružený k této transakci a získáte 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 24-1 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 ..... ~.
00006b16 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 00000000 00000000 00000000 .....
00000000 00000001 6a756469 74685f73 ..... judith_s
6f630000 736f6374 63700000 oc..soct cp..

```

Obrázek 24-1. 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 24-2, s identifikátorem záznamu BEGPREP, který ukazuje Obrázek 24-1.

```

c7064 184 PREPARE 4 0 c7038 chrisw
000000b8 00000044 00000004 000c7038 .....D .....p8
00005cd6 00000002 63687269 73770000 ..... chrisw..
00000000 00000000 00000069 00087eeb ..... .i..~.
00006b16 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 24-2. 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 vy se rozhodnete, 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 popsanych v následujících částech můžete určit, která data transakce aktualizovala a které záznamy byly ovlivněny.

Když budete provádět rozhodnutí, 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	20	BEGIN	2	1	0	08/27/91 10:56:57	
3482		nhowe					
1802c	32	HINSERT	2	0	18018	1000018	102
4							
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      20          BEGIN      2          1    0      08/27/91
    10:57:07 3483      pault

1602c      32          HINSERT    2          0    16018  1000018  102
    4

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 **sysables**.
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 hodnotu rowid 102 hexadecimálně (258 desítkově). Kompenzační transakce je tedy následující:

```
DELETE FROM t WHERE rowid = 258
```

Část 6. Dodatky a přílohy

Dodatek. Usnadnění

Diagramy syntaxe ve verzi HTML této příručky jsou k dispozici ve formátu syntaxe desítkových čísel oddělených tečkami, což je přístupný formát dostupný pouze tehdy, pokud používáte nástroj pro čtení obsahu obrazovky.

Diagramy syntaxe ve formátu desítkových čísel oddělených tečkami

Ve formátu desítkových čísel oddělených tečkami se každý prvek syntaxe zapisuje na samostatný řádek. Pokud se dva nebo více prvků syntaxe vyskytuje vždy společně (nebo se společně nevyskytují nikdy), mohou se tyto prvky vyskytnout na jediném řádku, protože je lze považovat za jediný složený prvek syntaxe.

Každý řádek začíná desítkovým číslem odděleným tečkami, například 3, 3.1 nebo 3.1.1. Aby byla tato čísla čtena správně, přesvědčete se, zda je nástroj pro čtení obsahu obrazovky nastaven tak, aby četl interpunkci. Všechny prvky syntaxe, které jsou uvozeny stejnými desítkovými čísly oddělenými tečkami (například všechny prvky syntaxe uvozené číslem 3.1) představují vzájemně se vylučující alternativy. Pokud slyšíte řádky 3.1 USERID a 3.1 SYSTEMID, může syntax obsahovat buď prvek USERID nebo prvek SYSTEMID, ne však oba zároveň.

Úroveň číslování desítkovými čísly oddělenými tečkami vyjadřuje úroveň vnoření. Pokud je například za prvkem uvozeným desítkovým číslem 3 následuje řada prvků syntaxe uvozených desítkovými čísly 3.1, jsou všechny prvky syntaxe uvozené čísly 3.1 podřízené prvku syntaxe uvozenému číslem 3.

Kromě desítkových čísel oddělených tečkami se k doplnění informací o prvcích syntaxe určitá slova a symboly. Občas se tato slova nebo symboly mohou vyskytnout na začátku samotného prvku. K usnadnění identifikace předchází slovo nebo symbolu znak zpětné lomítko (\), pokud je toto slovo nebo symbol součástí prvku syntaxe. Symbol * může být použit za desítkovým číslem odděleným tečkami a označuje možnost opakovat prvek syntaxe. Například prvek syntaxe *FILE s desítkovým číslem 3 je čten jako 3 * FILE. Formát 3* FILE označuje, že prvek syntaxe FILE se opakuje. Formát 3* * FILE označuje, že se prvek syntaxe * FILE opakuje.

Znaky jako je například čárka, které se používají k oddělování řetězce prvků syntaxe, se v syntaxi zobrazují těsně před oddělovanými položkami. Tyto znaky se mohou vyskytnout na téže řádce jako oddělovaná položka nebo na řádce uvozeném stejným desítkovým číslem oddělenými tečkami jako příslušné položky. Řádek může také obsahovat jiný symbol, který poskytuje informace o prvcích syntaxe. Například řádky 5.1*, 5.1 LASTRUN, a 5.1 DELETE znamenají, že pokud použijete více než jeden prvek syntaxe LASTRUN a DELETE, musejí být prvky odděleny čárkou. Není-li určen oddělovač, předpokládejte, že prvky syntaxe odděluje znak mezera.

Pokud prvku syntaxe předchází symbol %, znamená to, že jde o odkaz definovaný jinde. Řetězec následující za symbolem % je názvem fragmentu syntaxe, nikoliv literálem. Například řádek 2.1 %OP1 představuje odkaz na samostatný fragment syntaxe OP1.

Společně s desítkovými čísly oddělenými tečkami se používají následující slova a symboly:

- ? Určuje nepovinný prvek syntaxe. Desítková čísla oddělená tečkami následovaná symbolem ? označují, že všechny prvky syntaxe uvozené odpovídajícími desítkovými čísly a jakékoliv podřízené prvky syntaxe jsou nepovinné. Pokud je daným desítkovým číslem uvozen jen jediný prvek syntaxe, symbol ? je zobrazen na téže řádce jako prvek syntaxe (například 5? NOTIFY). Pokud je daným desítkovým číslem oddělenými tečkami uvozen více než jeden prvek syntaxe, symbol ? je zobrazen na samostatném řádku a nepovinné prvky syntaxe za ním následují. Pokud například slyšíte řádky 5 ?, 5 NOTIFY a 5 UPDATE, znamená to, že prvky syntaxe NOTIFY a UPDATE jsou nepovinné, můžete tedy zvolit jeden z nich nebo žádný. Symbol ? je ekvivalentní průchozí čáře v čárovém diagramu.
- ! Určuje výchozí prvek syntaxe. Desítkové číslo oddělené tečkami následované symbolem ! a prvkem syntaxe označuje, že prvek syntaxe je výchozí volbou všech prvků syntaxe, které sdílejí totéž desítkové číslo. Pouze jediný z několika prvků syntaxe uvozených daným desítkovým číslem může být označen symbolem ! . Například pokud slyšíte řádky 2? FILE, 2.1! (KEEP) a 2.1 (DELETE), znamená to, že (KEEP) je výchozí volbou klíčového slova FILE . Pokud v tomto příkladu použijete klíčové slovo FILE, ale nezadáte volbu, použije se výchozí volba KEEP. Výchozí volba se také použije pro nejbližší vyšší desítkové číslo oddělené tečkami. Pokud v tomto příkladu vynecháte klíčové slovo FILE , použije se výchozí volba FILE(KEEP). Pokud však uslyšíte řádky 2? FILE, 2.1, 2.1.1! (KEEP) a 2.1.1 (DELETE), pak se výchozí volba KEEP použije pouze pro nejbližší vyšší desítkové číslo oddělené tečkami 2.1 (ke kterému není přidruženo klíčové slovo) a nepoužije se pro klíčové slovo 2? FILE. Pokud je vynecháno klíčové slovo FILE, nepoužije se nic.
- * Určuje, že se může prvek syntaxe opakovat v libovolném počtu nebo nemusí být zadán. Desítkové číslo oddělené tečkami následované symbolem * označuje, že tento prvek syntaxe může být použit vícekrát nebo nemusí být použit, je tedy nepovinný a lze ho opakovat. Pokud například slyšíte řádek 5.1* datová-oblast, znamená to, že můžete zadat více datových oblastí nebo nemusíte zadat žádnou. Pokud slyšíte řádky 3* , 3 HOST a 3 STATE, znamená to, že můžete zadat klíčové slovo HOST, STATE, obě zároveň nebo ani jedno z nich.

Poznámky:

1. Pokud je k desítkovému číslu oddělenému tečkami připojen znak hvězdička (*) a jde o jedinou položku uvozenou tímto desítkovým číslem, můžete tuto položku několikrát opakovat.
 2. Pokud je k desítkovému číslu oddělenému tečkami připojen znak hvězdička a tímto desítkovým číslem je uvozeno několik položek, můžete použít více než jednu položku z tohoto seznamu, ale každou z nich můžete použít nejvýše jednou. V předchozím příkladu by bylo možné zadat HOST STATE, ale nikoliv HOST HOST.
 3. Symbol * je ekvivalentní zpětné čáře v čárovém diagramu syntaxe.
- +** Určuje, že prvek syntaxe musí být uveden nejméně jednou. Desítkové číslo oddělené tečkami následované symbolem + označuje, že tento prvek syntaxe musí být uveden nejméně jednou. Pokud například slyšíte řádek 6.1+ datová-oblast, musíte zapsat alespoň jednu datovou oblast. Pokud slyšíte řádky 2+, 2 HOST a 2 STATE, znamená to, že musíte zadat položku HOST, STATE nebo obě. Stejně jako v případě symbolu * můžete opakovat položku pouze tehdy, jedná-li se o jedinou položku uvozenou příslušným desítkovým číslem odděleným tečkami. Symbol + je stejně jako symbol * ekvivalentní zpětné čáře v čárovém diagramu syntaxe.

Poznámky

IBM nemusí nabízet produkty, služby nebo vlastnosti zmiňované v tomto dokumentu ve všech zemích. Informace o produktech a službách, které jsou momentálně ve vaší zemi dostupné, můžete získat od zástupce společnosti IBM pro vaši oblast. Žádný z odkazů na produkty, programové vybavení nebo služby není zamýšlen jako tvrzení, že lze použít pouze tyto produkty, programové vybavení nebo služby společnosti IBM. Jako náhrada mohou být použity libovolné funkčně ekvivalentní produkty, programové vybavení nebo služby, které neporušují žádné intelektuální vlastnické právo společnosti IBM. Za operace prováděné produkty, programy nebo službami, které nepochází od společnosti IBM, nese zodpovědnost uživatel.

Společnost IBM může mít patenty nebo podané žádosti o patent, které zahrnují předmět tohoto dokumentu. Vlastnictví tohoto dokumentu vám nedává žádná práva k těmto patentům. Písemné žádosti o licenci můžete posílat na adresu:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Pokud máte zájem o licenci v zemi s dvoubajtovou 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

Následující odstavec se netýká Velké Británie ani kterékoliv jiné země, kde taková opatření odporují místním zákonům: SPOLEČNOST INTERNATIONAL BUSINESS MACHINES CORPORATION TUTO PUBLIKACI POSKYTUJE TAK, JAK JE, BEZ JAKÝCHKOLIV ZÁRUK, VYJÁDŘENÝCH NEBO ODVOZENÝCH, VČETNĚ, MIMO JINÉ, ODVOZENÝCH ZÁRUK PRODEJNOSTI NEBO VHODNOSTI PRO DANÝ ÚČEL. Toto prohlášení se na vás nemusí vztahovat, pokud váš stát nedovoluje zřeknutí se výslovné a předpokládané záruky v některých transakcích.

Tato publikace může obsahovat technické nepřesnosti nebo typografické chyby. Informace zde uvedené jsou pravidelně aktualizovány a v příštích vydáních této publikace již budou tyto změny zahrnuty. IBM může kdykoliv bez upozornění zdokonalovat nebo měnit produkty a programy popsané na těchto stránkách.

Jakékoliv odkazy v těchto informacích na webové stránky jiných společností než IBM jsou poskytovány pouze pro větší pohodlí uživatele a nemohou být žádným způsobem vykládány jako schválení těchto webových stránek společností IBM. Materiály obsažené na takových webových stránkách nejsou součástí materiálů tohoto produktu společnosti IBM a mohou být používány pouze na vlastní riziko.

IBM může, pokud to považuje za vhodné, používat nebo distribuovat libovolné informace, které jí poskytnete, aniž by tím vznikl jakýkoliv závazek IBM vůči vám.

Držitelé licence tohoto programu, kteří si přejí mít přístup i k informacím o tomto programu 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
U.S.A.

Informace tohoto typu mohou být za odpovídajících podmínek dostupné. V některých případech může být požadováno zaplacení poplatku.

Program popsáný v tomto dokumentu a všechny materiály s ním související, které podléhají licenci, jsou dodávány IBM v souladu s textem smlouvy IBM Customer Agreement, IBM International Program License Agreement nebo smlouvy ekvivalentní, uzavřené mezi zákazníkem a IBM.

Všechny zde uvedené informace o výkonu byly zjištěny v řízeném prostředí. Výsledky zjištěné v jiném provozním 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 mohla být také odhadnuta extrapolací. Skutečné výsledky se mohou lišit. Uživatelé tohoto dokumentu by měli příslušné údaje ověřit ve vlastním 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 pro koncové zákazníky 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 jmen 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. Tyto vzorové programy můžete kopírovat, měnit a distribuovat 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 za veškerých podmínek. IBM proto nemůže zaručit spolehlivost, možnost opravy či funkčnost těchto programů ani tyto záruky nelze nijak vyvozovat. Tyto vzorové programy můžete kopírovat, měnit a distribuovat 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 společnosti) (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 prohlížíte tyto informace v souboru, nemusí se fotografie a barevné ilustrace zobrazit.

Ochranné známky

AIX; DB2; DB2 Universal Database; Distributed Relational Database Architecture; NUMA-Q; OS/2, OS/390 a OS/400; IBM Informix®; C-ISAM®, Foundation.2000™; IBM Informix® 4GL; IBM Informix® DataBlade® Module; Client SDK™; Cloudscape™; Cloudsync™; IBM Informix® Connect; IBM Informix® Driver for JDBC; Dynamic Connect™; IBM Informix® Dynamic Scalable Architecture™ (DSA); IBM Informix® Dynamic Server™; IBM Informix® Enterprise Gateway Manager (Enterprise Gateway Manager); IBM Informix® Extended Parallel Server™; i.Financial Services™; J/Foundation™; MaxConnect™; Object Translator™; Red Brick™; IBM Informix® SE; IBM Informix® SQL; InformiXML™; RedBack®; SystemBuilder™; U2™; UniData®; UniVerse®; wintegrate® jsou ochranné známky nebo registrované ochranné známky společnosti International Business Machines Corporation.

Java a všechna ochranná loga a známky založené na známce Java jsou ochranné známky nebo registrované ochranné známky Sun Microsystems, Inc. ve Spojených státech a případně v dalších jiných zemích.

Windows, Windows NT a Excel jsou ochranné známky nebo registrované ochranné známky Microsoft Corporation ve Spojených státech a případně v dalších jiných zemích.

UNIX je registrovaná ochranná známka v USA a případně v dalších jiných zemích, jejíž licenci poskytuje výhradně společnost X/Open Company Limited.

Další jména společností, produktů nebo služeb použitých v této publikaci mohou být ochrannými známkami jiných společností.

Rejstřík

Speciální znaky

(*), hvězdička

Viz též zástupný znak.

zástupný znak v poli názvu hostitele 3-32

Čísla

32bitová platforma

a společná oblast vyrovnávacích pamětí 11-17

64bitové adresování

definované 8-42

maximální počet vyrovnávacích pamětí 8-15

podpora databázového serveru 1-4

společná oblast vyrovnávacích pamětí 8-15

využití pamětí 8-42

64bitové platformy

a společná oblast vyrovnávacích pamětí 11-17

A

ADM.

Viz Administrační třída virtuálních procesorů.

administrační třída virtuálních procesorů 6-14

administrátor databáze

Viz úlohy administrace.

adresa IP

použití v poli názvu hostitele 3-30

zjištění 3-31

adresář

NFS 10-4

adresář \$INFORMIXDIR

oprávnění 5-2

adresář INFORMIXDIR/bin xxi

adresy IPv4 3-34

adresy IPv6 3-34

ADT.

Viz Virtuální procesor typu audit.

agregace paketů 3-44

agregační funkce

paralelní zpracování 6-8

AIO

virtuální procesory 6-26

aktualizace dat

tabulky typu RAW 10-32

tabulky typu STANDARD 10-31

alarm událostí

definovaný 1-26, 2-12

dynamicky přidané protokoly 15-18, 15-20

aliasy

Viz DBSERVERALIASES.

aplikace

klient

Viz klientská aplikace.

aplikace DSS

Viz dotaz pro podporu rozhodování.

aplikace OLTP

Viz paralelní databázový dotaz

asynchronní vstup - výstup

definice 6-24

jádra (KAIO) 6-23

asynchronní vstup - výstup jádra

definice 6-25

neprotokolující diskový vstup - výstup 6-23

atomický datový typ.

Viz vestavěný datový typ.

automatické

spuštění databázového serveru 1-17

vypnutí databázového serveru 1-18

automatické zotavení, dvoufázové potvrzování 23-6

B

běžné vyrovnávací paměti

událostí, které způsobují vyprázdnění 8-34

binární velký objekt

Viz datový typ BLOB.

blok

činnost během zotavení zrcadlení 18-5

definovaný 1-19

chyby vstupu-výstupu během zpracování 22-7

kontrola stavu 11-41, 11-43, 22-7

maximální počet 1-19, 10-3, 11-9

maximální velikost 1-5, 1-19, 10-3, 11-9

monitorování 11-41, 11-43, 22-7

název při přidělení jako přímé zařízení 10-5

oblast 10-10

obnovení vypnutého bloku 19-5

počáteční přidělení 11-7

podpora velkých bloků 1-20

pravidla pro rozvržení disků 10-42

propojení k názvu cesty 1-7

propojení s cestou 11-6, 11-7

propojení s názvem cesty 19-3

překročení maximální velikosti při použití správce

LVM 10-48

přidání k

prostoru sbpace 11-29

přidání pomoci

programu ON-Monitor 11-21

přidat k

zrcadlený prostor dbspace 19-7

- blok (*pokračování*)
 - přidávání
 - pomocí programu ISA nebo obslužného programu
onspaces 11-20
 - přidávání k
 - prostoru dbspace 11-20
 - prostoru sbspace 10-24, 11-29
 - seznam volných stránek
 - monitorování 17-2
 - spojení a rozdělení stránek 12-4
 - tabulka bloků, definovaná 8-21
 - uložení stavu na sekundární server 21-13
 - určení oblastí metadat 11-30
 - vypuštění z
 - blobpace 11-33
 - prostoru dbspace 11-33
 - sbspace 11-33
 - zálohování 11-21
 - zápis
 - monitorování 9-14
- blokové zařízení 10-5
- bloky
 - koncepty 10-3
 - zápis
 - kontrolní body 8-37
- Blowfish 5-10
- brána IBM Informix Gateway při heterogenním
potvrzování 23-22
- BUFSIZE
 - Viz stránka.*
- C**
 - centrální registr
 - informace sqlhosts 3-17
 - cron
 - obslužný program 1-29
 - CSM
 - Viz modul pro podporu komunikace.*
- Č**
 - čítač programu a data jednotkových procesů 8-24
 - čtení napřed
 - definované 8-33
 - konfigurační parametr RA_PAGES 8-33
 - konfigurační parametr RA_THRESHOLD 8-33
 - případy použití 8-33
- D**
 - další oblast
 - velikost 10-10
 - data
 - odhad diskového prostoru 10-42
 - šifrování přenosu 5-11
 - data o uživateli
 - vytváření 3-15
 - data typu TEXT a BYTE
 - Viz též jednoduchý velký objekt.*
 - absence komprese 11-59
 - monitorování v prostoru dbspace 11-50
 - zápis do prostorů blobpace 8-39
 - zavádění 11-58
 - databáze
 - Viz též zotavení.*
 - asynchronní vstup - výstup 6-25
 - definovaná 10-28
 - fragmentace 10-29
 - kompatibilní s ANSI 12-9
 - ladění
 - Viz ladění výkonu.*
 - monitorování 11-40, 13-7
 - odhadnutí velikosti 10-42
 - omezení velikosti 10-28
 - sysutils 4-8
 - účel 10-28
 - umístění 10-28
 - zobrazení stavu protokolování 13-7
 - databáze bez protokolování
 - podporované typy tabulek 10-30
 - příkazy jazyka SQL
 - nikdy neprotokolované 12-6
 - vždy protokolované 12-4
 - tabulky typu RAW 10-31
 - databáze kompatibilní se standardem ANSI
 - obslužný program ondblog 13-4
 - obslužný program ontape 13-5
 - změna režimu protokolování 13-4
 - databáze s protokolováním
 - podporované typy tabulek 10-30
 - příkazy jazyka SQL
 - nikdy neprotokolované 12-6
 - vždy protokolované 12-4, 12-5
 - tabulky typu RAW 10-31
 - tabulky typu STANDARD 10-31
 - databáze stores_demo xxi
 - databáze superstores_demo xxi
 - databáze sysmaster
 - Viz též tabulka SMI*
 - tabulka SMI 1-28
 - vytvoření 4-8
 - databáze sysutils
 - vytvoření 4-8
 - databázový server
 - 32bitové a 64bitové verze 1-4
 - modul pro podporu komunikace hesel (PCMS) 5-14
 - plánování podle priorit 1-15
 - požadavky vytvoření databáze 12-9
 - protokol zpráv 1-26
 - Připojení prostřednictvím SQL API 1-19
 - s vícenásobnými jednotkovými procesy 6-3
 - skupiny 3-27, 3-28, 3-30, 5-6

- databázový server *(pokračování)*
 - Enterprise Replication 3-27
 - replikace HDR 3-27
 - spuštění 1-16
 - vzdálený 3-14
- datové soubory
 - Viz* protokolování.
- datový blok.
 - Viz* stránka.
- datový typ
 - CLOB a BLOB 10-17, 16-10
 - operace jiného typu než fuzzy 16-10
 - operace typu fuzzy 16-9
 - replikovaný replikací HDR 20-3
 - uživatelský 10-17
- datový typ BLOB 10-16
 - Viz* inteligentní velký objekt.
- datový typ BYTE
 - Viz* data typu TEXT a BYTE.
- datový typ CLOB
 - Viz* inteligentní velký objekt.
- datový typ TEXT
 - Viz* data typu TEXT a BYTE.
- DDL.
 - Viz* jazyk Data Definition Language.
- DES
 - Viz* standard DES (Data Encryption Standard)
- DES3
 - Viz* standard DES3 (Triple Data Encryption Standard)
- diagnostická data
 - nastavení parametrů 22-8
 - požadavky na diskový prostor 22-8
 - sběr 22-8
- diagnostické zprávy
 - Viz* protokol zpráv.
- diagramy syntaxe
 - čtení pomocí nástroje pro čtení obsahu obrazovky A-1
 - klíčová slova xxx
 - konvence xxvii
 - proměnné xxxi
- diagramy syntaxe ve formátu desítkových čísel oddělených tečkami A-1
- disková stránka
 - čtení napřed 8-33
 - předobrazy ve fyzickém protokolu 8-35
- diskový prostor
 - blok, maximální velikost 1-19, 11-9
 - cesty k blokům, posuny 11-4
 - definice 11-3
 - inicializace 1-16, 4-2, 4-4, 4-6
 - monitorování pomocí programu ISA 11-45
 - oddíly uprostřed disku 10-44
 - odhadnutí velikosti 10-40
 - požadavky 10-42
 - pravidla pro rozvržení 10-42
- diskový prostor *(pokračování)*
 - přídělení
 - diskový prostor s přímým přístupem 11-6
 - prostor předpřipraveného souboru 11-5
 - v systému Windows 11-8
 - sledování využití prostorem tblspace 10-36
 - ukládání dat typu TEXT a BYTE 11-26
 - vytvoření odkazu na cestu bloku 1-7
 - vytvoření propojení k názvu cesty bloku 1-7
 - vytvoření propojení s cestou k bloku 11-6, 11-7
- diskový prostor s přímým přístupem
 - definovaný 10-5
 - přídělení v systému UNIX 10-4, 11-6
 - přídělení v systému Windows 10-4
 - znakové speciální rozhraní 10-5
- diskový vstup - výstup
 - asynchronní vstup - výstup jádra 6-23
 - čtení ze zrcadlených bloků 18-6
 - fronty 6-26
 - fyzický protokol 6-23
 - chyby během zpracování 22-7
 - inteligentní velké objekty 10-16
 - logický protokol 6-23
 - priority 6-23
 - přímý vstup - výstup 10-5
 - role sdílené paměti při omezování 8-3
 - třídy virtuálních procesorů 6-22
 - vstup - výstup prostřednictvím operačního systému 10-5
 - vyrovnávací paměti 11-17
 - zázpisy do zrcadlených bloků 18-6
- distribuovaná transakce
 - protokol dvoufázového potvrzování 23-4
 - určení, zda byla implementována nekonzistentně 24-3
- distribuované databáze 1-25
- distribuované dotazy
 - definované 1-25
 - dvoufázové potvrzování 1-25
 - nastavení sqlhosts 1-11
- distribuované transakce 12-3
 - protokolování bez vyrovnávací paměti 12-9
- distribuované zpracování 23-2
- dlouhá transakce
 - definovaná 14-6
 - důsledky 14-7
 - dvoufázové potvrzování 23-8, 23-11, 23-13
 - heuristické odvolání transakce 23-15
 - prevence 1-22
- dočasná tabulka
 - během inicializace 4-6
 - Enterprise Replication 10-31
 - inteligentní velký objekt, dočasný 10-25
 - obnovení 10-32
 - záloha 10-33
- dočasné logické protokoly 15-9

- dočasné prostory dbspace
 - definované 11-19
 - objem požadovaný pro dočasné tabulky 20-27
 - vytvoření 11-19
 - dočasné prostory sbspace
 - přidávání bloků 11-31
 - vypouštění bloků 11-31
 - vytváření 11-31
 - dočasný inteligentní velký objekt
 - definovaný 10-26
 - vytvoření 10-26
 - dočasný prostor dbspace
 - objem požadovaný pro dočasné tabulky 10-41
 - parametr DBSPACETEMP 10-34
 - replikace dat 20-26, 21-4
 - dočasný prostor sbspace
 - definovaný 10-25
 - charakteristiky 10-25
 - přidávání bloků 10-24
 - příklad 11-31
 - příznak LO_CREATE_TEMP 11-31
 - vypouštění bloků 10-26, 11-31
 - vypuštění prostoru sbspace 11-34
 - zálohování a obnovení 10-26
 - dokumentace, typy xxxii
 - poznámky k počítači xxxiii
 - příručky online xxxv
 - tištěné příručky xxxv
 - doména
 - systému Windows
 - definovaná 3-3
 - důvěryhodná 3-4
 - radič 3-3
 - spuštění jako určitý uživatel 1-30
 - uživatelské účty 3-4
 - doporučení
 - kontrola konzistence 22-2
 - přidělování diskového prostoru 10-6
 - zrcadlení fyzického protokolu 16-5
 - dostupnost
 - jako cíl efektivního rozvržení disků 10-42
 - kritická data 10-11
 - ukázkové rozvržení disků 10-44
 - dotaz pro podporu rozhodování
 - Viz též* paralelní databázový dotaz
 - konfigurační parametr DS_MAX_QUERIES 2-10
 - DS_POOLSIZ configuration parameter 9-5
 - DTP
 - Viz* distribuované transakce
 - důvěryhodná
 - doména 3-4
 - důvěryhodná doména 23-7
 - důvěryhodné
 - databázové servery 3-4
 - domény systému Windows 3-4
 - důvěryhodné (*pokračování*)
 - klienty 3-4
 - Dynamic Server
 - služba 1-17
 - dynamické přidělování protokolu
 - alarmy událostí a zprávy 15-18
 - definice 15-10
 - přehled 1-23
 - umístění souborů 15-12
 - velikost souboru protokolu 15-11
 - dynamické přidělování zámek 8-18
- ## E
- e-mail
 - oznamování alarmů událostí 2-12
 - Enterprise Replication
 - použití skupin databázových serverů 3-27
 - prostory sbspace 10-16
 - tabulky typu RAW 10-31
 - tabulky typu STANDARD 10-31
 - tabulky typu TEMP 10-32
 - určení prostorů sbspace 10-16
 - ESQL/C
 - přístup k inteligentním velkým objektům 10-24
 - externí prostor
 - Viz* prostor extspace
 - externí rutiny
 - zabezpečení 5-38
 - externí zálohování
 - a obnovení
 - použití při nastavení replikace HDR 21-7
- ## F
- FAT.
 - Viz* Tabulka FAT.
 - fáze po rozhodnutí 23-6
 - fáze před potvrzením 23-6
 - fragment
 - monitorování
 - požadavků na vstup - výstup 11-43
 - překočení
 - vybrané fragmenty 11-39
 - přeskočení
 - nedostupné fragmenty 11-39
 - tabulky a indexy 10-29
 - fragmentace
 - Viz* fragment
 - fragmentování
 - na více disků 10-44
 - fragmenty
 - monitorování
 - využití disku 11-47
 - přeskakování
 - nepřístupných fragmentů 11-37
 - pomocí parametru DATASKIP 11-38

- fronta čekajících procesů
 - definice 6-15
 - zámky vyrovnávacích pamětí 8-28
- fronta připravených procesů
 - definovaná 6-13
 - přesunutí jednotkového procesu 6-14, 20-8
- fronta spících procesů, definovaná 6-14
- fronty
 - čekajících procesů 6-15
 - definované 6-13
 - diskový vstup - výstup 6-22
 - připravených procesů 6-14
 - spánek 6-15
- fronty FIFO/LRU
 - definované 8-28
 - určení informací 8-28
- fronty FLRU
 - Viz též* fronty LRU
 - definované 8-29
 - způsob výběru databázovým serverem 8-30
- fronty LRU
 - definované 8-28
 - fronty FLRU 8-29
 - fronty MLRU 8-29
 - konfigurace více front 8-30
 - správa společné oblasti vyrovnávacích pamětí 8-29
 - stránky seřazené sestupně podle doby nepoužívání 8-29
 - určení informací 8-28
- fronty MLRU
 - Viz též* fronty LRU
 - definované 8-29
 - konec čištění 8-32
 - omezení počtu stránek 8-31
 - ukončení čištění stránek 8-32
 - umisťování vyrovnávacích pamětí 8-30
- funkce ENCRYPT 5-23
- funkce ifx_lo_copy 10-26
- funkce ifx_lo_specset_flags 10-26, 11-31
- funkce mi_lo_copy() 10-26
- funkce mi_lo_specset_flags() 10-26, 11-31
- funkce Nastavení serveru 1-14
 - vlastní nastavení konfigurace 1-14
- fyzická konzistence
 - definice 16-15, 16-18
 - rychlá obnova 16-15
- fyzické paměťové jednotky, seznam 10-2
- fyzické protokolování
 - datová vyrovnávací paměť 16-8
 - definice 16-3
 - inteligentní velké objekty 16-4
 - kontrolní body typu fuzzy 16-8
 - lehká připojení 11-34
 - podrobnosti o protokolování 16-4
 - protokolovaná aktivita 16-3
 - rychlá obnova 16-6
- fyzické protokolování (*pokračování*)
 - stránky blobpage prostoru blobspace 16-8
 - zálohy 16-4
- fyzický protokol
 - jak vyprázdnit 16-8
 - kde umístěn 16-3
 - máte-li jiné než výchozí velikosti stránek 11-15
 - monitorování 17-4
 - obsah předobrazu 16-7
 - opětovná inicializace sdílené paměti 17-2
 - pokyny týkající se velikost 16-6
 - přehled 1-23
 - přetečení během rychlé obnovy 16-15
 - role v rychlé obnově 16-7, 16-14, 16-16, 16-18
 - souvislý prostor 17-2
 - virtuální procesory 6-24
 - virtuální procesory vstupu - výstupu 17-2
 - vstup - výstup, virtuální procesory 6-24
 - vyprázdnění vyrovnávací paměti 16-8
 - vyrovnávací paměť 8-16, 16-7
 - zajištění jeho nezaplnění 16-8
 - zálohování 11-10
 - zaplnit se 16-6
 - změna velikosti a umístění
 - důvody 17-2
 - možné metody 17-1
 - omezení 17-3
 - pomocí programu ON-Monitor 17-3
 - pomocí příkazu onparams 17-2
 - pomocí textového editoru 17-3
 - zprávy fyzického zotavení 16-4
 - zvětšení velikosti 11-15, 16-6, 17-4

G

- gcore
 - obslužný program 2-13
 - soubor 22-4
- globální podpora jazyků xx, 1-9
- globální společná oblast, definovaná 8-26
- GLS
 - Viz* Global Language Support
- GTRID 24-4

H

- haldy 8-24
- hardwarové zrcadlení 18-4
- hashovací tabulka 8-21
- HDR
 - Viz* replikace High-Availability Data Replication
- heslo
 - změna pro uživatele informix 1-30
- heterogenní potvrzování 23-21, 23-23, 23-27
- heuristické odvolání transakce
 - definice 23-10
 - dlouhá transakce 23-15

- heuristické odvolání transakce (*pokračování*)
 - ilustrace a záznamy protokolu 23-18
 - indikace výskytu odvolání transakce 24-2
 - monitorování, příkaz onstat -x 23-15
 - výsledkem (čeho) 23-11
- heuristické rozhodnutí
 - nezávislá akce 23-8
 - typy 23-10
- heuristické ukončení transakce
 - definice 23-13
 - ilustrace a záznamy protokolu 23-20
 - kdy je nezbytné 23-13
 - určení vlivu na transakci 24-2
 - vracené zprávy 23-14
 - výsledkem (čeho) 23-14
- hodnota klíče
 - sdílené paměti 8-10
- hodnota parametru lru_max_dirty 8-28, 8-31, 8-32, 11-18
 - příklad použití 8-32
- hodnota parametru lru_min_dirty 8-28, 8-32, 11-18
 - ve vztahu k čištění stránek 8-33
- hodnota parametru lrus 11-17
- hodnota pole buffers 11-16, 11-17
- hodnota STUDIO 21-11
- HPL
 - Viz* zavaděč High Performance Loader
- hvězdička (*)
 - Viz též* zástupný znak.
 - zástupný znak v poli názvu hostitele 3-32

CH

- chyba aserce
 - během kontroly konzistence 22-4
 - během zpracování požadavku uživatele 22-5
 - definovaná 22-1
 - formát protokolu zpráv 22-4
 - poškození dat 22-6
 - zjištění příčiny 22-8
- chyba média
 - zjišťování 18-6
- chyba vstupu-výstupu bránící v provozu
 - kdy nastane 22-9
 - monitorování
 - protokol zpráv 22-10
 - sledování pomocí
 - alarmů událostí 22-10
- chybová zpráva
 - chyby vstupu-výstupu na bloku 22-7
 - protokol dvoufázového potvrzování 23-15
 - při kontrole zabezpečení prováděné obslužnými programy 5-4
- chybové zprávy xxxiv
- chyby vstupu-výstupu bránící v provozu
 - destruktivní a nedestruktivní 22-9

- IBM Informix Server Administrator
 - funkce Nastavení serveru 1-14
 - konfigurace sqlhosts 1-12
 - monitorování diskové paměti 11-45
 - monitorování zámek latch a zámek spin 9-10
 - obnovení vypnutého bloku 19-8
 - použití obslužného programu onmode 21-16
 - provádění příkazů obslužného programu 9-14
 - přehled 1-26
 - přidání zrcadlených bloků 19-7
 - přidávání bloků 11-20, 11-29
 - režimy databázového serveru 4-11
 - sdílená paměť 9-6
 - spustit zrcadlení 19-6
 - spuštění 1-14
 - stav protokolování databáze 13-6
 - ukončení zrcadlení 19-8
 - úpravy konfigurace 2-15
 - úpravy souboru sqlhosts 1-12
 - uživatelská oprávnění 4-11
 - virtuální procesory 7-3
 - vytváření
 - prostorů sbspace 11-29
 - vytvoření
 - prostorů dbspace 1-19
 - prostory dbspace 11-10
 - vytvoření prostorů dbspace 1-19
 - změna provozních režimů 4-11
 - změna režimů databázového serveru 4-11
 - zobrazení stavu protokolování 13-7
 - zobrazení zpráv 1-26
- IBM Informix Storage Manager 1-20
- ID
 - v logickém protokolu 14-3
- ID databázového serveru 2-2
- ID protokolu 14-3
- identifikátor globální transakce 24-4
- identifikátor segmentu 8-10
- IDS 10.0
 - vlastnosti xxi
- IDS 9.21
 - vlastnosti xxv
- IDS 9.3
 - vlastnosti xxiv
- IDS 9.4
 - vlastnosti xxiii
- imc jako část pole určující protokol 3-20
- index
 - fragmentace 10-29
 - ověření 22-3
 - paralelní vytváření 6-8
 - prostor tblspace 10-37
 - R-strom 16-5

- index B-stromu
 - jednotkový proces čištění 8-23
 - index R-stromu
 - protokolování 16-5
 - INFORMIXDIR
 - adresáře a odpovídající oprávnění 5-5
 - INFORMIXSERVER 1-8
 - inicializace
 - databáze sysmaster 4-8
 - databáze sysutils 4-8
 - diskového prostoru 1-16, 4-2, 4-4
 - diskový prostor 4-6
 - dočasné prostory tablespaces 4-7
 - konfigurační soubory 4-4, 4-7
 - kontrolní bod 4-6, 4-8
 - kroky 4-3
 - maximální počet připojení 4-8
 - návrat řízení uživateli 4-7
 - protokol zpráv 4-7
 - převod interních souborů 4-6
 - rychlá obnova 4-6
 - sdílená paměť 4-4
 - sdílené paměti 4-2
 - soubor oncfg_servername.servernum 4-7
 - tabulky SMI 4-8
 - virtuální procesory 4-6
 - vynucená rezidence 4-7
 - instalace
 - klíč registru sqlhosts 3-16
 - MaxConnect 3-44
 - Windows 1-6
 - Instance Manager 1-13, 1-15
 - instmgr.exe 1-13
 - integrita předávaných zpráv 3-9
 - integrita, datová.
 - Viz kontrola konzistence.*
 - inteligentní velké objekty
 - výpočet velikosti oblasti 11-29
 - inteligentní velký objekt
 - AVG_LO_SIZE 10-20, 11-29
 - čas posledního přístupu 10-21
 - definovaný 10-16
 - fyzické protokolování 16-4
 - lehké operace vstupu - výstupu 10-21
 - metadata 10-17
 - oblasti a bloky 10-10
 - odhadnutá velikost 11-30
 - odlehčený vstup-výstup 8-15
 - odstranění dat CLOB nebo BLOB 11-34
 - odstraňování, nulový počet odkazů 11-34
 - paměť 10-24
 - paměťové charakteristiky 10-17, 10-20, 11-30
 - použití protokolování 14-10
 - použití replikace Enterprise Replication 10-16
 - inteligentní velký objekt (*pokračování*)
 - použití vyrovnávací paměti
 - doporučení 8-38
 - režim 8-41
 - prostory dbspace 10-9
 - protokolování 10-22, 14-9
 - přístup v aplikacích 10-24
 - příznak LO_CREATE_TEMP 10-26
 - statistické údaje pro uživatelem definovaná data 2-4
 - stránky sbpage 10-9
 - ukládání do vyrovnávací paměti 10-20
 - určení datových typů 10-17
 - určení velikostí oblasti metadat 11-57
 - uživatelská data 10-17
 - vlastnosti vstup-výstup 10-17
 - vyhledávání dat 10-16
 - výpočet průměrné velikosti velkého objektu 10-20
 - výpočet velikosti oblasti 10-10
 - vyrovnávací paměti ve sdílené paměti 8-41
 - vytvoření 10-18, 14-11
 - zamykání 10-16, 10-21
 - zamykání rozsahu bajtů 8-17
 - zapnutí nebo vypnutí protokolování 14-9
 - IPC
 - Viz meziprocesová komunikace.*
 - ISA
 - Viz IBM Informix Server Administrator.*
 - ISM
 - Viz IBM Informix Storage Manager.*
 - ixpasswd.exe 1-30
 - izolace tabulek 10-44
- ## J
- J/Foundation
 - instalační adresář ovladače JDBC 1-8
 - konfigurace 1-16
 - nastavení proměnné CLASSPATH 1-8
 - proměnné prostředí 1-10
 - Java Development Kit
 - soubor krakatoa.jar 1-8
 - jazyk DDL
 - příkazy 20-27
 - jazyk DDL (Data definition language)
 - příkazy 12-4
 - jazyk ESQ/C
 - přístup k inteligentním velkým objektům 10-16
 - jazyk Structured Query Language
 - Viz příkazy jazyka SQL.*
 - JDK
 - Viz Java Development Kit.*
 - jednoduchý velký objekt
 - Viz též data typu TEXT a BYTE.*
 - deskriptor 8-40
 - ilustrace paměťového prostoru blobspace 8-40
 - vyrovnávací paměti 8-40

- jednoduchý velký objekt (*pokračování*)
 - vytvoření v prostoru blobspace 8-39
 - zápis na disk 8-39
 - jednoprocesorový počítač 2-11, 6-18
 - jednotkové procesy čištění stránek
 - definované 8-34
 - monitorování 8-22
 - role při zápisu bloků 8-37
 - upozorňované při zápisech na popředí 8-36
 - vyprazdňování běžných vyrovnávací paměti 8-34
 - vyprazdňování společné oblasti vyrovnávacích paměti 8-34
 - jednotkové procesy předávající řízení
 - definované 6-13
 - fronta připravených procesů 6-13
 - podmínky 6-10
 - přepínání mezi jednotkovými procesy 6-10
 - v předem stanoveném okamžiku 6-11
 - jednotkové procesy typu listener 5-8
 - definice 6-28
 - přidávání 6-31
 - více karet rozhraní 6-33
 - jednotkový proces
 - asynchronní vstup - výstup jádra 6-25
 - definovaný 6-4
 - haldy 8-24
 - interní 6-4, 6-16
 - jak pracují virtuální procesory 6-25
 - jednotkové procesy čištění stránek 6-3, 6-4, 8-22
 - klientské aplikace 6-4
 - kontext 6-10
 - migrace 6-13
 - několik současných 6-10
 - onmode 6-4
 - plánování a synchronizace 6-10
 - primární relace 8-23
 - probuzení 6-11
 - prohledávání B-stromu 6-4
 - předání řízení 6-10
 - předávání řízení 6-10
 - přepínání mezi jednotkovými procesy 6-10
 - přístup ke sdíleným vyrovnávacím pamětím 8-28
 - přístup ke zdrojům 6-10
 - relace 6-12, 8-23
 - řídící blok 6-10
 - řízení souběžného přístupu 8-27
 - spící 6-14
 - uživatelský 6-4, 8-23
 - vztah k procesu 6-4, 6-16
 - zásobníky 8-24
 - zotavení 6-4
 - zrcadlení 6-4
 - jednotkový proces cyklického dotazování
 - definice 6-28
 - fronty zpráv 6-30
 - jednotkový proces cyklického dotazování (*pokračování*)
 - kolik 6-27
 - konfigurační parametr DBSERVERNAME 6-28
 - na virtuálních procesorech CPU nebo síťových 6-27
 - počet 6-28
 - položka nettype 6-27
 - připojení 6-27
 - více procesů pro protokol 6-27
 - jednotkový proces KAIO 6-25
 - jednotkový proces splexec
 - klientská aplikace 6-29
 - role při spojení klienta/serveru 6-29
 - uživatelský jednotkový proces 6-4
 - jednotkový proces vstupu - výstupu KAIO 6-23
 - jednouživatelský režim 4-9, 4-13, 4-14
 - pro pár replikace HDR 21-17
 - jiná než výchozí velikost stránky
 - vytvoření společné oblasti vyrovnávacích pamětí pro 11-15
- ## K
- klíč registru sqlhosts
 - definovaný 3-17
 - proměnná prostředí INFORMIXSQLHOSTS 3-17
 - klíč shmkey
 - definovaný 8-10
 - připojení dalších segmentů 8-11
 - klíčová slova
 - v diagramech syntaxe xxx
 - klíčová slova MODE ANSI
 - stav protokolování databáze 12-8
 - klíčové slovo ALL
 - parametr DATASKIP 11-39
 - klíčové slovo DEFAULT
 - v příkazu SET DATASKIP 11-38
 - klidový režim 4-9
 - klient-server
 - konfigurace
 - jednotkové procesy typu listener a cyklického dotazování 6-28
 - místní zpětná smyčka 3-9
 - sdílená paměť 3-7
 - příklad konfigurace
 - místní zpětná smyčka 3-39
 - prostřednictvím protokolu IPX/SPX 3-40
 - sdílená paměť 3-38
 - síťové připojení 3-40
 - více typů připojení 3-41
 - vícenásobné databázové servery 3-42
 - vícenásobné uložení 3-42
 - klientská aplikace
 - definovaná 3-2
 - globální transakce 23-2
 - konfigurace propojitelnosti 1-11, 1-19, 3-10
 - konfigurace prostředí 1-9

klientská aplikace (*pokračování*)

- konfigurace velikosti zásobníku 8-24
- multiplexní připojení 3-5
- podporovaná připojení 3-6
- pole názvu hostitele 3-21
- pole typu připojení 3-19
- pole voleb 3-24
- položky sqlhosts 3-15, 3-18, 3-32
- použití replikace dat 20-3
- proměnná prostředí ONCONFIG 1-9
- přesměrování při replikaci dat 20-14, 20-17
- připojení k hostiteli 3-15
- připojení k primárnímu nebo sekundárnímu serveru 20-4
- připojení ke sdílené paměti 8-8
- připojení prostřednictvím místní zpětné smyčky 3-9
- připojení prostřednictvím sdílené paměti 3-8
- reakce na selhání replikace HDR 20-14
- síťové domény systému 3-4
- soubory zabezpečení sítě 3-13
- určení adresy pomocí zástupných znaků 3-34
- určení názvu dbservername 3-36
- vzdálení hostitelé 3-13
- zahájení relace 8-23
- knihovna TP/XA 8-22
- kód SQL xxxi
- kód, ukázka, konvence xxxi
- kolize požadavků
 - Viz kolize požadavků na přístup k disku.*
- kolize požadavků na přístup k disku
 - omezení 10-42
- kompatibilita
 - s oborovými standardy xxxviii
- komunikační prostředky
 - klienta a databázového serveru
 - Viz propojitelnost.*
 - sdílená paměť
 - definovaná 8-26
 - velikost 8-26
 - způsob připojení klientu 8-8
- komunikační protokol TCP/IP
 - číslo naslouchajícího portu 3-34
 - pole názvu hostitele 3-21
 - pole názvu služby 3-22
 - použití
 - adresa IP sítě Internet 3-34
 - číslo naslouchajícího portu TCP 3-34
 - protokol IPX/SPX 3-40
 - soubor hosts.equiv 3-12
 - více portů 3-12
 - zástupné znaky 3-31
- konfigurace
 - J/Foundation 1-16
 - monitorování 2-15
 - odhad potřebného diskového prostoru 10-42
 - plánování databázového serveru 1-3

konfigurace (*pokračování*)

- použití funkce Nastavení serveru 1-14
- požadavky 1-3
- prostředí databázového serveru 1-2
- více portů 3-12
- Windows 1-4
- konfigurace disku 1-4
- konfigurace paměťových zařízení 1-20
- konfigurace parametru MIRROR 2-3
- konfigurace parametru ROOTNAME 2-3
- konfigurace parametru TBLTBLFIRST 2-3
- konfigurační parametr
 - Viz též názvy jednotlivých parametrů.*
 - ADTERR 2-14
 - ADTMODE 2-14
 - ADTPATH 2-14
 - ADTSIZE 2-14
 - AFF_NPROCS 7-2
 - AFF_SPROC 6-17, 7-2
 - ALARMPROGRAM 1-26, 2-12
 - ALRM_ALL_EVENTS 2-12
 - BUFFERPOOL 2-8, 8-14, 11-15
 - CDR_QDATA_SBSpace 10-16
 - CKPTINTVL 2-8, 16-11
 - CLEANERS 2-9
 - CONSOLE 1-28, 2-6
 - DATASKIP 2-10, 11-37
 - DBSERVERALIASES 2-2, 3-36
 - DBSERVERNAME 2-2, 3-19, 3-36
 - DBSPACETEMP 2-4, 10-43, 11-19
 - DD_HASHMAX 2-8
 - DD_HASHSIZE 2-8
 - DEADLOCK_TIMEOUT 2-11, 23-21
 - DEF_TABLE_LOCKMODE 2-8, 8-18
 - diagnostických informací 22-8
 - DIRECTIVES 2-13
 - DRAUTO 2-12, 20-14, 20-15
 - DRIDXAUTO 2-12, 21-15, 21-16
 - DRINTERVAL, 2-12
 - DRLOSTFOUND 2-12, 20-10
 - DRTIMEOUT 2-12, 21-15
 - DS_HASHSIZE 8-25, 9-5
 - DS_MAX_QUERIES 2-10
 - DS_MAX_SCANS 2-10
 - DS_NONPDQ_QUERY_MEM 2-10, 8-19
 - DS_POOLSIZE 8-25, 9-5
 - DS_TOTAL_MEMORY 2-10, 8-19
 - DUMPCNT 2-13, 22-8
 - DUMPCORE 2-13, 22-8
 - DUMPDIR 2-13, 22-4, 22-8
 - DUMPGCORE 2-13, 22-5, 22-8
 - DUMPSHMEM 2-13, 22-5, 22-8
 - DYNAMIC_LOGS 2-5, 15-17, 15-19
 - EXT_DIRECTIVES 2-13
 - FAST_RESTART_CKPT_FUZZYLOG 2-6, 16-21

konfigurační parametr *(pokračování)*

FAST_RESTART_PHYSLOG 2-6, 16-21
FILLFACTOR 2-4
HETERO_COMMIT 2-11, 23-23
IFX_EXTEND_ROLE 2-14, 5-38
LISTEN_TIMEOUT 1-11, 2-13, 5-8
LOCKS 2-8
LOGBUFF 2-5, 8-16
LOGFILES 2-5
LOGSIZE 2-5
LTAPEBLK 2-6, 21-6
LTAPESIZE 2-6, 21-6
LTXEHWM 2-5, 15-17, 15-21, 23-10
LTXHWM 2-5, 15-17, 15-20, 23-10
MAX_INCOMPLETE_CONNECTIONS 1-11, 2-13, 5-8
MAX_PDQPRIORITY 2-10
MIRROR 2-3, 19-2
MIRROROFFSET 2-3, 10-14, 11-5
MIRRORPATH 2-3, 10-14
MSGPATH 1-27, 2-6
MULTIPROCESSOR 6-17
MULTIPROCESSOR 2-11
nastavení pomoci textového editoru 9-6
NETTYPE 2-11, 3-37
NUMCPUVPS 6-17
OFF_RECVRY_THREADS 2-11, 20-12
ON_RECVRY_THREADS 2-11
ONDBSPACEDOWN 2-4
OPCACHEMAX 2-14, 11-48
OPTCOMPIND 2-10
parametr DYNAMIC_LOGS 15-13
PC_HASHSIZE 2-8, 8-26, 9-5
PC_POOLSIZE 2-8, 8-26, 9-5
PHYSBUFF 2-5, 8-17
PHYSDBS 2-5
PHYSFILE 2-5, 16-5, 17-3
PLOG_OVERFLOW_PATH 16-15
počáteční blok kořenového prostoru dbspace 10-13
RA_PAGES 2-9, 8-33
RA_THRESHOLD 2-9, 8-33
RESIDENT 2-8, 9-8
ROOTNAME 2-3, 10-14
ROOTOFFSET 2-3, 10-13, 11-5
ROOTPATH 2-3
ROOTSIZE. 2-3
SBSPACENAME 2-4, 10-18, 10-25, 11-28
SBSPACETEMP 2-4, 10-25, 11-31
inteligentní velké objekty 11-28
sdílená paměť 9-2
SERVERNUM 2-2, 8-9, 8-10
SHMADD 2-7, 8-19
SHMBASE 2-7, 8-9, 8-10, 9-4
SHMTOTAL 2-7, 8-7
SHMVRTSIZE 2-7, 8-19
SINGLE_CPU_VP 2-11, 6-17

konfigurační parametr *(pokračování)*

STACKSIZE 2-8, 8-24
STAGEBLOB 2-14
STMT_CACHE 2-9, 9-7
STMT_CACHE_HITS 2-9, 9-7
STMT_CACHE_NOLIMIT 2-9, 9-7
STMT_CACHE_NUMPOOL 2-9, 9-7
STMT_CACHE_SIZE 2-9, 9-7
SYSSBPACENAME 2-4
TAPEBLK 2-6, 21-6
TAPEDEV 2-6, 21-11
TAPESIZE 2-6, 21-6
TBLTBLFIRST 2-3, 10-14, 11-12
TBLTBLNEXT 2-3, 3-34, 10-14, 11-12
TXTIMEOUT 2-11, 23-14, 23-21
USEOSTIME 2-11
VPCLASS 2-11, 6-22, 7-2
konfigurační parametr ADTERR 2-14
konfigurační parametr ADTMODE 2-14, 6-34
konfigurační parametr ADTPATH 2-14
konfigurační parametr ADTSIZE 2-14
konfigurační parametr AFF_NPROCS 7-2
konfigurační parametr AFF_SPROC 6-17, 7-2
konfigurační parametr ALARMPROGRAM 1-26, 2-12
konfigurační parametr ALRM_ALL_EVENTS 2-12
konfigurační parametr BUFFERPOOL 2-8, 8-14, 8-28,
11-15, 16-9, 16-10
a inteligentní velké objekty 8-41
konfigurační parametr CDR_DBSPACE
povolení a zakázání 15-10
přidání souborů protokolu 15-13, 15-19
úprava hodnoty 15-17
konfigurační parametr CDR_QDATA_SBSpace 10-16
konfigurační parametr CKPTINTVL
definovaný 2-8
vyvolání kontrolních bodů 16-11
konfigurační parametr CLEANERS 2-9
účel 8-22
konfigurační parametr CONSOLE 1-28, 2-6
konfigurační parametr DATASKIP 2-10
definovaný 11-37
klíčové slovo ALL 11-39
konfigurační parametr DBSERVERALIASES 1-8, 2-2
definovaný 3-36
příklad 3-36
soubor sqlhosts 3-19
více typů připojení 3-41
konfigurační parametr DBSERVERNAME 1-8, 2-2
definovaný 3-36
pravidla syntaxe 3-19
přidružený protokol 6-27
soubor sqlhosts 3-19
virtuální procesor pro jednotkový proces cyklického
dotazování 6-27
konfigurační parametr DBSPACETEMP 2-4, 10-34, 11-19

konfigurační parametr DBSPACETEMP *(pokračování)*
definování dočasných tabulek 10-43
rozložení zatížení 10-43

konfigurační parametr DD_HASHMAX 2-8

konfigurační parametr DD_HASHSIZE 2-8

konfigurační parametr DEADLOCK_TIMEOUT 2-11, 23-21
protokol dvoufázového potvrzování 23-21

konfigurační parametr DEF_TABLE_LOCKMODE 2-8, 8-18

konfigurační parametr DIRECTIVES 2-13

konfigurační parametr DRAUTO 2-12
role při zotavení ze selhání replikace dat 20-14, 20-15
role v zotavení po selhání replikace dat 21-23

konfigurační parametr DRIDXAUTO 2-12, 21-15, 21-16

konfigurační parametr DRINTERVAL 2-12
nastavení pro
asynchronní aktualizaci 20-10
synchronní aktualizaci 20-9

konfigurační parametr DRLOSTFOUND 2-12, 20-10

konfigurační parametr DRTIMEOUT 2-12, 21-15
zjištění selhání replikace dat 20-13

konfigurační parametr DS_HASHSIZE 8-25, 9-5

konfigurační parametr DS_MAX_QUERIES 2-10

konfigurační parametr DS_MAX_SCANS 2-10

konfigurační parametr DS_NONPDQ_QUERY_MEM 2-10, 8-19

konfigurační parametr DS_POOLSIZ 8-25

konfigurační parametr DS_TOTAL_MEMORY 2-10, 8-19

konfigurační parametr DUMPCNT 2-13, 22-8

konfigurační parametr DUMPCORE 2-13, 22-8

konfigurační parametr DUMPDIR 2-13, 22-4, 22-8

konfigurační parametr DUMPGCORE 2-13, 22-5, 22-8

konfigurační parametr DUMPSHMEM 2-13, 22-5, 22-8

konfigurační parametr DYNAMIC_LOGS 2-5

konfigurační parametr EXT_DIRECTIVES 2-13

konfigurační parametr
FAST_RESTART_CKPT_FUZZYLOG 2-6, 16-21

konfigurační parametr FAST_RESTART_PHYSLOG 2-6, 16-21

konfigurační parametr FILLFACTOR 2-4

konfigurační parametr HETERO_COMMIT 2-11, 23-23

konfigurační parametr IFX_EXTEND_ROLE 2-14, 5-38

konfigurační parametr LISTEN_TIMEOUT 1-11, 2-13, 5-8

konfigurační parametr LOCKS 2-8

konfigurační parametr LOGBUFF 2-5, 15-17
inteligentní velké objekty 8-41
vyrovnávací paměti logického protokolu 8-16

konfigurační parametr LOGFILES 2-5, 15-17
nastavení velikosti logického protokolu 15-3
úprava hodnoty ONCONFIG 15-17

konfigurační parametr LOGSIZE 2-5, 15-17
přidání souborů protokolu 15-13
velikost logického protokolu 15-2
změna 15-17

konfigurační parametr LOGSIZE *(pokračování)*
zvětšení velikosti protokolu 15-15

konfigurační parametr LTAPEBLK 2-6, 21-6

konfigurační parametr LTAPEDEV
konfigurační parametr
LTAPEDEV 2-6

konfigurační parametr LTAPESIZ 2-6, 21-6

konfigurační parametr LTXEHWM 2-5, 23-10
definice 15-17
role v heuristickém odvolání transakce 23-10
zabránění dlouhým transakcím 15-21

konfigurační parametr LTXHWM 2-5
definice 15-17
role v heuristickém odvolání transakce 23-10
zabránění dlouhým transakcím 15-20

konfigurační parametr
MAX_INCOMPLETE_CONNECTIONS 1-11, 2-13, 5-8

konfigurační parametr MAX_PDQPRIORITY 2-10

konfigurační parametr MIRROR
počáteční konfigurační hodnota 19-2
změna 19-2

konfigurační parametr MIRROROFFSET 2-3, 10-14
v případě potřeby 11-5

konfigurační parametr MIRRORPATH 2-3, 10-14

konfigurační parametr MSGPATH 1-27, 2-6

konfigurační parametr MULTIPROCESSOR 2-11, 6-17, 7-2

konfigurační parametr NETTYPE 2-11, 3-37, 7-2
jednotkové procesy cyklického dotazování 6-27
multiplexní připojení 3-5
role při určení protokolu 6-27
účel 3-37
vícenásobné síťové adresy 6-33
záznam třídy VP 6-27

konfigurační parametr NUMCPUVPS 6-17

konfigurační parametr OFF_RECVRY_THREADS 2-11, 20-12

konfigurační parametr ON_RECVRY_THREADS 2-11

konfigurační parametr ONDBSPACEDOWN 2-4

konfigurační parametr OPCACHEMAX 2-14, 11-48
konfigurace paměti 11-48

konfigurační parametr OPTCOMPIND 2-10

konfigurační parametr PC_HASHSIZE 2-8, 8-26, 9-5

konfigurační parametr PC_POOLSIZ 2-8, 8-26, 9-5

konfigurační parametr PHYSBUFF 2-5, 8-17

konfigurační parametr PHYSDBS 2-5, 17-3
změna velikosti a umístění 16-5

konfigurační parametr PHYSFILE 2-5, 16-5, 17-3

konfigurační parametr PLOG_OVERFLOW_PATH 16-15

konfigurační parametr RA_PAGES 2-9, 8-33

konfigurační parametr RA_THRESHOLD 2-9, 8-33

konfigurační parametr RESIDENT 2-8
během inicializace 4-7
nastavení pomocí obslužného programu onmode 9-8

konfigurační parametr ROOTNAME 10-14

konfigurační parametr ROOTOFFSET 2-3, 10-13, 11-5

- konfigurační parametr ROOTPATH 2-3, 10-13
- konfigurační parametr ROOTSIZE 2-3
- konfigurační parametr SBSPACENAME 2-4, 10-18, 10-25, 11-28
- konfigurační parametr SBSPACETEMP 2-4, 10-25, 11-31
- konfigurační parametr SERVERNUM 2-2
 - definovaný 8-9, 8-10
 - použití 8-9
- konfigurační parametr SHMADD 2-7, 8-19
- konfigurační parametr SHMBASE
 - definovaný 2-7, 8-10
 - nastavení vyšší hodnoty 9-4
 - připojení prvního segmentu sdílené paměti 8-9
- konfigurační parametr SHMTOTAL 2-7, 8-7
- konfigurační parametr SHMVIRTSIZE 2-7, 8-19
- konfigurační parametr SINGLE_CPU_VP 2-11, 6-17, 7-2
 - jednoprocesorový počítač 6-18
- konfigurační parametr STACKSIZE 2-8
 - definovaný 8-24
 - změna velikosti zásobníku 8-24
- konfigurační parametr STAGEBLOB 2-14
- konfigurační parametr STMT_CACHE 2-9, 9-7
- konfigurační parametr STMT_CACHE_HITS 2-9, 9-7
- konfigurační parametr STMT_CACHE_NOLIMIT 2-9, 9-7
- konfigurační parametr STMT_CACHE_NUMPOOL 2-9, 9-7
- konfigurační parametr STMT_CACHE_SIZE 2-9, 9-7
- konfigurační parametr SYSSBSPACENAME 2-4
- konfigurační parametr TAPEBLK 2-6, 21-6
- konfigurační parametr TAPEDEV 2-6, 21-11
- konfigurační parametr TAPESIZE 2-6, 21-6
- konfigurační parametr TBLTBLFIRST 10-14, 11-12
- konfigurační parametr TBLTBLNEXT 2-3, 3-34, 10-14, 11-12
- konfigurační parametr TXTIMEOUT 2-11, 23-14, 23-21
 - definice 23-21
 - onmode -Z 23-14
 - protokol dvoufázového potvrzování 23-21
- konfigurační parametr USEOSTIME 2-11
- konfigurační parametr VPCLASS 2-11, 6-22, 7-2
 - konfigurace virtuálních procesorů CPU 6-16
 - třída JVP 6-22
 - uživatelé definované VP 6-21
- konfigurační parametry Java 1-16
- konfigurační soubor
 - neprovádění změn v souboru onconfig.std 1-13
 - onconfig.std 1-13
 - propojitelnost 3-10
 - sítě 1-12
- konfigurační soubor komunikace
 - Viz konfigurační soubor ONCONFIG.
- konfigurační soubor ONCONFIG
 - během inicializace 1-13, 1-14, 4-4
 - definovaný 2-15
 - parametry 1-13, 2-15
- konfigurační soubor ONCONFIG (*pokračování*)
 - parametry jazyka Java 1-16
 - propojitelnost 1-13
 - připojení prostřednictvím sdílené paměti 3-38
 - příprava 1-13
 - úprava 1-16
 - vícenásobné uložení 3-42
 - zobrazení pomocí příkazu onstat -c 2-15
- kontakt xxxix
- kontrola konzistence
 - data a režie 22-2
 - indexy 22-3
 - logické protokoly 22-3
 - monitorování konzistence dat 22-4
 - oblast 22-3
 - ověření metadat 22-4
 - periodické úlohy 22-2
 - poškození dat 22-6
 - poškození indexu 22-6
 - prostory sbospace 10-24
 - přehled 1-22, 22-1
 - tabulky systémového katalogu 22-3
- kontrolní bod
 - definovaný 16-8
 - krok při inicializaci sdílené paměti 4-6, 4-8
 - lehká připojení 11-35
 - maximální počet připojení 4-8
 - monitorování aktivity 17-5
 - pokyny týkající se zálohování 16-13
 - princip 16-12
 - replikace dat 21-13
 - role v rychlé obnově 16-16
 - soubor logického protokolu 16-11
 - typu fuzzy 16-9
 - události, které ho vyvolají 16-11
 - úplný 16-9
 - vy nucený 21-13
 - vyprazdňování běžných vyrovnávací paměti 8-34
 - vyrovnávací paměť fyzického protokolu 8-35, 16-8, 16-11
 - vyrovnávací paměť logického protokolu 8-15
 - vyvolávající 4-6
 - zázpisy bloků 8-37
- kontrolní bod typu fuzzy
 - během inicializace 4-8
 - definovaný 16-8, 16-9
 - fyzické protokolování 16-6, 16-8
 - nejstarší aktualizace 14-7
 - princip 16-12
 - rychlá obnova 16-18, 16-21
 - vy nucení 17-7
 - vyprázdnění fyzického protokolu 16-13
 - vyprázdnění společné oblasti vyrovnávacích paměti 16-10, 16-13
 - vyrovnávací paměť fyzického protokolu 8-35

- kontrolní bod typu fuzzy (*pokračování*)
 - vyrovnávací paměť logického protokolu 8-15
 - konvence
 - diagramy syntaxe xxvii
 - dokumentace xxvi
 - příkazový řádek xxix
 - typografické xxvi
 - ukázkový kód xxxi
 - zápis syntaxe xxvii
 - konvence používané v dokumentaci xxvi
 - konvence příkazového řádku
 - jak číst xxix
 - ukázkový diagram xxix
 - konvence ukázkového kódu xxxi
 - konzistence dat
 - jak lze dosáhnout 16-2
 - monitorování 22-4
 - ověření 22-2
 - příznaky poškození 22-6
 - rychlá obnova 16-14
 - konzole
 - zprávy 1-28
 - koordinující databázový server 23-5
 - kořenový prostor dbspace
 - definovaný 10-13
 - dočasné tabulky 10-32, 10-36
 - umístění souborů logického protokolu 14-2
 - výchozí umístění 10-14, 10-34
 - výpočet velikosti 10-40
 - zrcadlení 19-4
 - kritická část kódu
 - definice 16-2
 - kontrolní body 16-12
 - kritické prostory dbspace
 - paměť 10-11
 - zrcadlení 10-43, 10-47
- L**
- ladění výkonu
 - logický protokol, nové umístění 17-2
 - objem protokolovaných dat 16-3
 - pravidla pro rozvržení disků 10-42
 - přesun fyzického protokolu 17-2
 - rozkládání dat na více disků 10-48
 - správce logických disků 10-48
 - ukázkové rozvržení disků optimalizované na výkon 10-46
 - velikost logického protokolu 14-5, 17-2
 - zápisy front LRU 8-36
 - zápisy na popředí 8-36
 - lehká připojení
 - fyzické protokolování 11-35
 - tabulky typu RAW 10-31
 - tabulky typu STANDARD 10-31
 - lehké operace vstupu - výstupu 10-21
 - lehké prohledávání 8-23
 - licencování uživatelé, maximální povolený počet 4-4, 4-8
 - logická konzistence, definice 16-15, 16-18
 - logické paměťové jednotky
 - seznam 10-3
 - logické zotavení, replikace dat 20-8
 - logický protokol
 - Viz též* vyrovnávací paměť logického protokolu; soubor logického protokolu.
 - definovaný 8-15, 12-2, 14-1
 - dynamické přidělování 1-23
 - globální transakce 23-3, 23-15
 - konfigurační parametry 15-17
 - monitorování za účelem zjištění zaplnění pomocí obslužného programu onstat 15-8
 - obslužný program onlog 15-18
 - ověření 22-3
 - pokyny týkající se velikosti 14-4, 15-2
 - pokyny týkající se výkonu 14-5
 - pozice v protokolu 23-15
 - protokolování inteligentních velkých objektů 15-3
 - typy záznamů 8-15
 - úlohy administrace 1-22, 14-8
 - záloha
 - definovaná 1-21
 - harmonogram 1-21
 - kontrolní body 16-13
 - záznam
 - procesy databázového serveru vyžadující 12-2
 - protokol dvoufázového potvrzování 23-5, 23-16, 23-17
 - příkazy jazyka SQL, které generují 12-5
- LVM.
- Viz* správce logických disků.
- M**
- mapování chybného sektoru, absence 22-11
 - mapování chybných sektorů 22-11
 - MaxConnect
 - imc jako část pole určující protokol 3-20
 - instalace 3-44
 - monitorování 3-44
 - protokol onsoncime 3-21
 - protokol ontlime 3-21
 - maximální
 - počet bloků 11-9
 - počet paměťových prostorů 11-9
 - maximální počet
 - uživatelských připojení 4-4, 4-8
 - maximum
 - velikost bloku 11-9
 - mechanismus obnovy dat
 - rychlá obnova 16-14
 - mechanizmy obnovy dat
 - inteligentní velké objekty 10-16

- metadata
 - bloky 11-30
 - definovaná 8-41, 10-17
 - dočasný prostor sbspace 10-25
 - ověření 22-4
 - protokolování 16-4
 - protokolování prostoru sbspace 14-9
 - přesunutí prostoru z rezervované oblasti 11-57
 - přídělení 11-28
 - tabulka prostorů dbspace 8-21
 - určení správné velikosti 10-20, 11-29
 - výpočet oblasti 10-24, 11-29
 - vypuštění bloků prostoru sbspace 11-33
 - vytvoření 10-20
- meziprocesová komunikace
 - sdílená paměť 8-3
 - v poli nettype 3-20
- Microsoft Transaction Server 23-3
- místní zpětná smyčka
 - omezení 3-9
 - příklad 3-39
 - připojení 3-9, 6-27
- modul ENCCSM
 - moduly CSM, šifrování 5-11
- modul LDAP
 - distribuované transakce 5-32, 5-35
 - klientská rozhraní 5-32, 5-35
 - konfigurace 5-31
 - konfigurace serveru Dynamic Server 5-31
 - ověřování 5-32
 - problémy s kompatibilitou 5-36
 - s replikací dat 5-32
 - vývoj aplikací 5-32, 5-33
- modul LDAP Authentication Support Module v operačním systému Windows 5-30
- modul PAM (Pluggable Authentication Module) 5-27
 - definice 5-27
 - distribuované transakce 5-35
 - implicitní připojení 5-29
 - klientská rozhraní API 5-35
 - konfigurace serveru 5-29
 - podporované platformy 5-28
 - požadovaná velikost zásobníku 5-29
 - problémy s kompatibilitou 5-36
 - vývoj aplikací 5-33
- modul Password Communication Support Module 5-14
- modul pro podporu komunikace
 - konfigurační soubor 3-9
 - pole voleb informací sqlhosts 3-26
 - zabezpečení sítě 3-9
- modul pro podporu komunikace (CMS) 5-11
 - konfigurační soubor 5-12
 - záznam souboru conccsm.cfg 5-14, 5-22
- modul pro podporu komunikačních prostředků
 - virtuální procesor 6-33
- modul SPWDCSM (Simple Password Communication Support Module)
 - konfigurační soubor CSM 5-14
- moduly DataBlade
 - omezení přístupu pro registraci uživatelských rutin 5-38
 - virtuální procesory 6-17
- monitorování
 - aktivita uživatele 23-15
 - databázového serveru 1-25
 - globální transakce 23-15
 - licencování uživatelé 4-8
 - licencovaných uživatelů 4-4
 - oblast 11-47
 - oblastí metadat a uživatelských dat 11-57
 - počet uživatelů 16-5
 - prostor sbspace 11-52
 - prostory sbspace 10-24, 11-53
 - prostory tblspace 11-47
 - rychlá vyrovnávací paměť pro příkazy SQL 9-7
 - uživatelských připojení 4-4, 4-8
 - velikost paměti 9-4
 - využití programu MaxConnect 3-44
 - zámky 23-15
 - zámky spin 9-10
- monitorování databázového serveru 1-28
 - aktivita společné oblasti vyrovnávacích pamětí 9-13
 - bloky 11-41
 - databáze 11-40, 13-7
 - délka front diskového vstupu - výstupu 6-26
 - fronty diskového vstupu - výstupu 6-26
 - globální transakce 1-25
 - hodnoty konfiguračních parametrů 2-15
 - jednoduchý velké objekty v prostorech dbspace 11-47, 11-50
 - kontrolní body 17-5
 - oblastí 11-47
 - paměť prostoru blobspace 11-27
 - pomocí monitoru ON-Monitor 1-27
 - pomocí programu oncheck 1-27
 - použití programu onstat 1-28
 - používání tabulek SMI 1-28
 - profil aktivity 9-10
 - sdílená paměť 9-9
 - soubor fyzického protokolu 17-4
 - soubory protokolu 15-8
 - stav protokolování 13-7
 - stav replikace dat 21-18
 - uživatelské jednotkové procesy 8-23
 - virtuální procesory 7-6
 - vyrovnávací paměť fyzického protokolu 8-17, 17-4
 - vyrovnávací paměti 9-11
 - vyrovnávací paměti logického protokolu 17-4
 - využití disku fragmenty 11-43, 11-47
 - zámky latch 9-9, 9-10
- MTS/XA 23-3

multiplexní připojení 3-5
mutex
definice 6-15
synchronizace 6-16

N

nápověda xxxv
nápověda online xxxv
národní prostředí xx
výchozí xxi
národní prostředí en_us.8859-1 xxi
nastavení
programu ON-Bar 1-20
programu ontape 1-20
proměnné prostředí 1-10
nástroj pro čtení obsahu obrazovky
čtení diagramů syntaxe A-1
nástroj setnet32 1-12
nástroje k monitorování
sledování výkonu v systému Windows 1-29
UNIX 1-29
název
paměťové prostory 11-9
název databázového serveru
Viz konfigurační parametr DBSERVERNAME.
název hostitele
změna 1-30
neaktualizovaná vyrovnávací paměť, definovaná 8-34
nejstarší aktualizace, uvolnění souboru logického
protokolu 14-4
několik karet síťového rozhraní 6-32
několik souběžných jednotkových procesů 6-10
nekonzistence, rozpoznání 22-2
nesprávně pracující uživatelská rutina 6-20
New Technology File System 1-6
ntchname.exe 1-30

O

objekty mutex
definované 8-27
použití 8-27
oblast
definovaná 10-10
ověření 22-3
počáteční velikost 10-10, 10-20
přidělování 10-10
sbspace 10-9, 10-20
související klíčové koncepty 10-10
struktura 10-10, 10-20
účel 10-10
velikost další oblasti 10-10, 10-20
velikost pro prostor tbspace tbspace 11-12
velikost v prostorech sbspace 10-9, 10-20
velikost, další oblasti 11-12
velikost, počáteční 11-12
oblast (*pokračování*)
vztah k bloku 10-10
způsob přidělování databázovým serverem 10-11
oblast uživatelských dat, prostor sbspace 8-41
oblasti
monitorování 11-47
monitorování pomocí tabulky sysextents 11-47
obnovení
tabulky typu RAW 10-31
tabulky typu STANDARD 10-31
typy tabulek 10-33
obnovení dat
Viz obnovení.
obnovení k určitému datu
protokolování 10-31
oborové standardy, kompatibilita s xxxviii
obsah svazku 11-4
obslužný program
Viz též obslužný program oncheck; obslužný program
ondblog; obslužný program oninit; obslužný program
onlog; obslužný program onmode; obslužný program
onparams; obslužný program onspaces; obslužný
program onstat; obslužný program ontape; program
ON-Monitor.
cron 1-29
chkenv 1-10
iostat 1-29
ISA 11-40
onstat
volba -d 11-53
připojení ke sdílené paměti 8-8
ps 1-29
sar 1-29
UNIX 1-29
vmstat 1-29
obslužný program archecker
přehled 1-21
obslužný program convert systému Windows 1-7
obslužný program dbload 11-58
obslužný program chkenv 1-10
obslužný program ixsu.exe 1-30
obslužný program ondblog
databáze kompatibilní se standardem ANSI 13-6
změna režimu protokolování
ISA 13-3
ondblog 13-2
obslužný program oncheck
informace o stránkách blobpage 11-49
kontrola konzistence 22-3
monitorování oblastí metadat a uživatelských dat 11-55
monitorování prostorů sbspace 11-28, 11-52, 11-54
ověření
indexy 22-3
metadata 22-2, 22-4
oblast 22-2, 22-3

- obslužný program oncheck *(pokračování)*
 - ověření *(pokračování)*
 - protokoly a rezervované stránky 22-2
 - rezervované stránky 22-2
 - stránky dat 22-2
 - tabulky systémového katalogu 22-3
 - volba -cc 22-2
 - volba -cD 22-3
 - volba -ce 22-2
 - volba -cI 22-2, 22-3
 - volba -cr 22-2
 - volba -cR 22-2
 - volba -cs 11-52, 11-55, 22-4
 - volba -pB 11-49
 - volba -pe 11-45, 11-52, 11-54
 - volba -pr 2-15, 15-9, 16-5
 - volba -ps 11-56, 22-2
 - volba -pS 11-56
 - získání informací
 - bloky 11-45
 - konfigurace 11-49
 - logické protokoly 15-9
 - oblasti a fragmentace 11-47
 - prostory blobspace 11-44, 11-54
 - prostory tblspace 11-47
- obslužný program oninit
 - dočasné tabulky 10-36
 - spuštění databázového serveru 4-2
 - spuštění databázového serveru. 4-2
 - volba -p 4-7, 10-26
- obslužný program onlog
 - rekonstrukce globální transakce 24-3
 - zobrazení záznamů protokolu 15-18
- obslužný program onmode
 - c option 9-8
 - konfigurace rychlé vyrovnávací paměti příkazů jazyka SQL 9-7
 - nastavení
 - typ databázového serveru 21-16
 - odstranění
 - relace 24-2
 - transakce 23-8, 23-14
 - účastnický jednotkový proces 23-10
 - parametry -W 9-7
 - použití v programu ISA 21-16
 - přepínání souborů logického protokolu 15-5
 - přidání segmentu 9-9
 - uvolnění souboru logického protokolu 15-7
 - uživatelský jednotkový proces obsluhující požadavky 6-4
 - volba -a 9-9
 - volba -d idxauto 21-16
 - volba -e 9-7
 - volba -O 22-9
 - volba -W 9-7
- obslužný program onmode *(pokračování)*
 - vynucení úplného kontrolního bodu 11-34, 17-6, 21-13
 - vypuštění virtuálních procesorů CPU 7-5
 - změna uložení sdílené paměti 9-8
- obslužný program onmode -d index 21-16
- obslužný program onparams
 - přidání souboru logického protokolu 15-13
 - vypuštění souboru logického protokolu 15-14
 - změna fyzického protokolu
 - umístění 17-3
 - velikost 17-3
- obslužný program onperf 1-28
- obslužný program onpladm 11-58
- obslužný program onsocimc 3-21
- obslužný program onspaces
 - definovaný 1-19
 - možnost -d 11-35
 - obnovení vypnutého bloku 19-8
 - přidávání bloků prostoru sbospace 11-29
 - přidávání zrcadlených bloků 11-29
 - ukončení zrcadlení 19-8
 - volba -a 11-29, 19-7
 - volba -c -b 11-25
 - volba -c -d 11-11
 - volba -c -S 11-28
 - volba -c -t 11-20
 - volba -c -x 11-36
 - volba -cl 11-34
 - volba -d 11-28
 - volba -f 11-37
 - volba -g 11-25
 - volba -ch 10-20, 10-23, 11-30, 21-13
 - volba -k 11-19
 - volba -s 10-17
 - volba -t 10-43
 - volba -U 11-30
 - volby -Df 10-20, 11-33
 - vypnutí bloku 19-7
 - vypuštění bloků prostoru sbospace 11-33
 - vytváření prostorů sbospace 11-28
 - vytvoření dočasného prostoru sbospace 10-25
 - změna parametru DATASKIP 11-37
 - změna stavu bloku 21-13
- obslužný program onstat
 - aktualizace statistických údajů o stránkách
 - blobpage 11-42
 - monitorování
 - blobspace 11-41
 - fyzický protokol 17-4
 - konfigurace 2-15
 - profil databázového serveru 9-10
 - příkazy jazyka SQL 13-7
 - replikace dat 21-19
 - rychlá vyrovnávací paměť pro příkazy SQL 9-7
 - sdílená paměť 9-9

- obslužný program onstat (*pokračování*)
 - monitorování (*pokračování*)
 - soubory logického protokolu 15-13, 17-4
 - společná oblast vyrovnávacích pamětí 9-14, 9-15
 - stav bloku 11-41
 - transakce 7-6
 - virtuální procesory 7-6
 - vyrovnávací paměť logického protokolu 15-8, 17-4
 - využití vyrovnávací paměti 9-11, 9-12, 9-14
 - zámky latch 9-9
 - zatížení fragmentů 11-43
 - profilování uživatelské aktivity 23-15
 - přehled 1-28
 - příznaky dočasného prostoru sbpace 11-31
 - virtuální procesory CPU 6-4
 - volba -c 2-15
 - volba -d 11-42, 11-52, 11-53
 - volba -d update 11-42
 - volba -g ath 7-6
 - volba -g cac 9-7
 - volba -g glo 7-6
 - volba -g imc 3-44
 - volba -g ioq 7-7
 - volba -g rea 7-8
 - volba -g seg 9-4
 - volba -g smb c 11-52, 11-58
 - volba -g sql 13-7
 - volba -g stm 13-7
 - volba -k 23-15
 - volba -l 15-13
 - volba -m 1-27
 - volba -p 9-10
 - volba -s 9-10
 - volba -u 6-4, 16-5, 23-15
 - volba -x 13-7, 23-14
 - volba aktualizace -d 1-27
 - volby -g ssc 9-7
 - zaznamenání
 - globální transakce 23-15
 - zámky 23-15
 - zobrazení
 - zpráv 1-27
- obslužný program ontape
 - alternativní metoda zálohování 21-11
 - nastavení 1-20
 - zálohování souborů logického protokolu 14-7
 - změna stavu protokolování databáze 13-4
- oddíly
 - uvedené v příkazu ALTER FRAGMENT 11-23
 - vytváření a používání 11-23
 - vytváření v již vytvořené tabulce nebo indexu 11-24
 - vytváření ve fragmentované tabulce nebo indexu 11-23
- odkládání paměti na disk 8-12
- odlehčené procesy 6-3
- odlehčený vstup-výstup 8-15
- odložená kontrola omezení 12-2
- odolnost vůči selhání
 - replikace dat 20-5
 - rychlá obnova 16-14
- odstranění
 - dat inteligentního velkého objektu 11-34
 - data inteligentních velkých objektů 11-35
 - soubory protokolu 15-14
 - tabulky typu STANDARD 10-31
- odstraňování
 - tabulky typu RAW 10-31
- odvolání transakce
 - heuristické, monitorování 23-15
 - inteligentní velké objekty 10-27
 - protokolované příkazy jazyka SQL 12-5
 - rychlá obnova 16-17, 16-21
 - tabulky typu RAW 10-31
 - tabulky typu STANDARD 10-27
- omezení
 - kontrola, odložená 12-2
 - počet bloků 11-9
 - procesorů JVP 6-22
 - uživatелеm definované VP 6-20
 - velikost bloku 11-9
 - VP CPU 6-16
- ON-Bar
 - nastavení 1-20
- ON-Monitor
 - kopírování konfigurace 2-15
 - monitorování databázového serveru 1-27
 - nastavení parametrů
 - sdílená paměť 9-6
 - virtuální procesory 7-3
 - obnovení bloku 19-8
 - přidání
 - soubor logického protokolu 15-14
 - přidávání
 - bloků 11-21
 - zrcadlené bloky 19-7
 - spuštění zrcadlení 19-6
 - vypnutí bloku 19-7
 - vypouštění paměťových prostorů 11-34, 11-35
 - vypuštění souboru logického protokolu 15-15
 - vytváření prostorů blobspace 11-26
 - vytvoreni prostorů dbspace 11-12
 - změna režimů databázového serveru 4-11
- operace jiného typu než fuzzy
 - definice 16-10
 - vyprázdnění společné oblasti vyrovnávacích pamětí 16-10
 - vyprázdnění vyrovnávací paměti 16-12
 - vyprázdnění vyrovnávacích pamětí 16-13
- operace typu fuzzy
 - alternativní volby rychlé obnovy 16-21
 - definice 16-9

- operace typu fuzzy (*pokračování*)
 - společná oblast vyrovnávacích pamětí 16-13
 - operační systém
 - 32bitová a 64bitová verze 9-2
 - nástroje 1-29
 - parametry 9-2
 - operační systém UNIX
 - nastavení proměnných prostředí 1-9
 - příkaz link 1-7, 11-6
 - spouštěcí skript 1-17
 - ukončovací skript 1-17
 - změna jádra 1-5
 - oprávnění
 - nastavení 1-7
 - optický virtuální procesor 6-34
 - optimalizace předpokládaného přerušení 23-7
 - ověření
 - datové stránky 22-3
 - indexů 22-4
 - logické protokoly 22-3
 - metadat 22-4
 - oblast 22-3
 - rezervované stránky 22-3
 - tabulky systémového katalogu 22-3
 - ověřování
 - definované 3-9
 - moduly 5-27
 - výchozí metoda 3-9
 - ovladač IBM Informix ODBC 23-3
 - ovladač JDBC 1-8
 - ovladač ODBC 23-3
 - ovladač síťového protokolu
 - definovaný 3-2
- ## P
- PAM
 - Viz modul Pluggable Authentication Module*
 - paměť
 - Viz též sdílená paměť.*
 - 64bitové platformy 8-42
 - přidání segmentu 9-9
 - paměť uložených procedur
 - Viz vyrovnávací paměť rutin UDR.*
 - paměťová zařízení
 - nastavení 1-20
 - paměťové charakteristiky
 - hierarchie 10-22
 - inteligentní velké objekty 10-20, 11-30
 - paměťové prostory 1-20
 - prostory sbospace 10-20
 - příkaz onspaces -Df 11-28
 - příkaz onspaces -ch 10-20, 11-30
 - paměťové jednotky 10-2
 - paměťový prostor
 - harmonogram zálohování 1-21
 - paměťový prostor (*pokračování*)
 - spuštění zrcadlení 19-5
 - ukončení zrcadlení 19-8
 - uložení oddílů NTFS 1-6
 - paměťový prostor v logickém disku nebo jednotce
 - definovaný 1-19
 - paralelní databázový dotaz
 - dotaz pro podporu rozhodování
 - Viz konfigurační parametr DS_MAX_QUERIES.*
 - konfigurační parametr DS_MAX_QUERIES 2-10
 - konfigurační parametr DS_NONPDQ_QUERY_MEM 2-10, 8-19
 - konfigurační parametr DS_TOTAL_MEMORY 2-10, 8-19
 - konfigurační parametr MAX_PDQPRIORITY 2-10
 - paralelní zpracování
 - virtuální procesory 6-8
 - parametr AVG_LO_SIZE 10-20, 11-29
 - parametr BUFFERING volby příkazu onspaces -c -Df 10-21
 - parametr LOGGING příkazu onspaces 10-22
 - parametr šifrování ENCCSM_CIPHERS 5-18
 - parametr šifrování ENCCSM_MAC 5-18
 - parametr šifrování ENCCSM_MACFILES 5-18
 - parametr šifrování ENCCSM_SWITCH 5-18
 - parametry jádra
 - změna 1-5
 - parametry souboru ONCONFIG
 - Viz konfigurační parametr.*
 - PDQ
 - Viz paralelní databázový dotaz*
 - plán dotazu 8-25
 - plánování
 - priorita 1-15
 - plánování podle priority 1-15
 - plánování zdrojů 1-4
 - počáteční konfigurace
 - rozvržení disků 10-42
 - směrnice pro kořenový prostor dbspace 10-13
 - vytvoření paměťových prostorů 1-19
 - podpora 32bitových systémů 1-4
 - pokyny týkající se velikost
 - fyzický protokol 16-6
 - logický protokol 15-12
 - pokyny týkající se velikosti
 - logický protokol 14-4, 15-2
 - pole beginlg 23-15
 - pole curlog 23-15
 - pole lchwaits 9-10
 - pole logposit 23-15
 - pole názvu hostitele
 - definované 3-21
 - několik karet síťového rozhraní 6-33
 - použití adresy IP 3-30
 - pravidla syntaxe 3-21
 - protokol IPX/SPX 3-22

- pole názvu hostitele (*pokračování*)
 - sdílená paměť 3-22, 3-24
 - určení adresy pomocí zástupných znaků 3-31
- pole názvu služby v informacích sqlhosts
 - definované 3-22
 - pravidla syntaxe 3-21
 - protokol IPX/SPX 3-23
 - proudová propojení procesů 3-23
 - sdílená paměť 3-23
- pole nettype
 - formát 3-19
 - použití typu rozhraní 3-40
 - přehled hodnot 3-21
 - syntaxe 3-19
- pole RAID 18-4
- pole voleb
 - konec skupiny 3-27
 - pravidla syntaxe 3-24
 - přehled 3-24
 - přesměrování připojení 3-26
 - volba CSM 3-26
 - volba identifikátoru 3-29
 - volba keep-alive 3-29
 - volba skupiny 3-29
 - volba velikosti vyrovnávací paměti 3-25
 - volba zabezpečení 3-29
- postížení, zraková
 - čtení diagramů syntaxe A-1
- posun
 - rozdělování oddílů 11-4
 - účel 10-6
 - v případě potřeby 11-4
 - zabránění přepsání informací o oddílech 10-6, 11-4
- poškození, řešení chyb vstupu-výstupu 22-6
- potvrzování
 - heterogenní 23-21, 23-23
- použití programu onperf 1-28
- pozice v protokolu 23-15
- poznámky k dokumentaci xxxiii
- poznámky k počítači, xxxiii
- poznámky k verzi xxxiii
- poznámky online xxxii, xxxiii
- poznámky s obsahem xxxiii
- požadavky
 - konfigurace 1-12
- požadavky na systém
 - databáze xx
 - software xx
- prázdný soubor
 - vytvoření 11-7
- primární blok 10-4
- primární databázový server 20-4
- priorita
 - diskový vstup - výstup 6-23
 - stárnutí, prevence 6-18
- probuzení jednotkových procesů 6-14
- procesorová afinita
 - definice 6-18
 - použití 6-18
- procesy
 - DSA a architektura dvou procesů 6-9
 - porovnání s jednotkovými procesy 6-7
 - připojení ke sdílené paměti 8-8
- procesy s vícenásobnými jednotkovými procesy
 - zdroje operačního systému 6-7
- profil
 - statistika 9-10
- program Instance Manager 1-15
- program MaxConnect
 - agregace paketů 3-44
 - definovaný 3-43
 - příkaz imcadmin 3-44
- program ON-Monitor
 - nastavení parametrů
 - sdílená paměť 9-6
- program Windows Instance Manager 1-15, 4-3
- prohledávání
 - indexů 8-33
 - paralelní zpracování 6-8
 - sekvenční prohledávání tabulek 8-33
- prohlížeč událostí, Windows NT 1-29
- proměnná prostředí
 - CLASSPATH 1-8, 1-10
 - CLIENT_LOCALE 1-9
 - DB_LOCALE 1-9
 - DBSPACETEMP 11-19
 - INFORMIXCONCSMCFG 5-12
 - INFORMIXDIR 1-8, 1-18
 - INFORMIXSERVER 1-8
 - INFORMIXSHMBASE 8-8
 - INFORMIXSQLHOSTS 1-11, 3-15
 - JVPHOME 1-8, 1-10
 - klientské aplikace 1-9
 - konfigurační soubor prostředí 1-9
 - LD_LIBRARY_PATH 1-10
 - nastavení 1-8
 - ONCONFIG 1-9, 2-15
 - PATH 1-9, 1-18
 - požadované proměnné 1-8
 - příklad souboru nastavení 1-10
 - SERVER_LOCALE 1-9
 - soubor .profile nebo .login 1-10
 - soubor dbservername.cmd 1-11
 - soubory informix.rc a .informix 1-10
 - TERM 1-9
 - TERMCAP 1-9
 - TERMINFO 1-9
- proměnná prostředí C8BITLEVEL 1-9
- proměnná prostředí CLASSPATH 1-8, 1-10
- proměnná prostředí CLIENT_LOCALE 1-9

- proměnná prostředí DB_LOCALE 1-9
- proměnná prostředí DBLANG 1-9
- proměnná prostředí DBPATH
 - použití při automatickém přesměrování 20-18
 - skupina databázových serverů 3-28
- proměnná prostředí DBSPACETEMP 10-34, 11-19
- proměnná prostředí ESQLMF 1-9
- proměnná prostředí GL_DATE 1-9
- proměnná prostředí GL_DATETIME 1-9
- proměnná prostředí GLS8BITFSYS 1-9
- proměnná prostředí INFORMIXCONSCMCFG 5-12
- proměnná prostředí INFORMIXDIR
 - definovaná 1-8
 - spouštěcí skript 1-18
 - ukončovací skript 1-18
- proměnná prostředí INFORMIXSERVER
 - definovaná 1-8
 - klientské aplikace 3-28
 - použití při přeměrování klientů 20-22
- proměnná prostředí INFORMIXSHMBASE 8-8
- proměnná prostředí INFORMIXSQLHOSTS 1-11, 3-15, 3-17
- proměnná prostředí INFORMIXSTACKSIZE 8-24
- proměnná prostředí JVPHOME 1-8, 1-10
- proměnná prostředí LD_LIBRARY_PATH 1-10
- proměnná prostředí ONCONFIG
 - definovaná 1-9
 - nastavení 1-13
 - vícenásobné databázové servery 1-18
- proměnná prostředí PATH 1-9, 1-18
 - spouštěcí skript 1-8, 1-18
 - ukončovací skript 1-18
- proměnná prostředí SERVER_LOCALE 1-9
- proměnná prostředí TERM 1-9
- proměnná prostředí TERMCAP 1-9
- proměnná prostředí TERMINFO 1-9
- proměnné prostředí xxvi
- proměnné, v diagramech syntaxe xxxi
- propojení
 - vytvoření 11-6, 11-7
- propojitelnost
 - ASF 3-4
 - konfigurační parametry 3-35
 - soubor hosts 3-11
 - soubor services 3-11
 - soubor sqlhosts 1-11
 - soubory, definované 3-10
 - Windows 1-12
- propojovací soubory sítě TCP/IP 3-10
- prostor blobspace
 - aktivace 14-8
 - definovaný 10-15
 - názvy 11-24
 - omezení
 - přidání protokolů 15-12
- prostor blobspace (*pokračování*)
 - přidání pomoci
 - obslužného programu onspaces 11-25
 - statistické údaje o paměti 11-27
 - stránka s mapou volných stránek
 - zaznamenávání stránek blobpage 8-40
 - úlohy protokolování 10-15, 14-8
 - vypuštění
 - počáteční úlohy 11-34
 - vypuštění pomoci
 - programu ON-Monitor 11-35
 - vypuštění z
 - onspaces 11-35
 - vyrovnávací paměti 8-40
 - vytvoření pomoci
 - obslužného programu onspaces 11-25
 - zálohování 11-10, 11-25
 - zápis dat 8-39
 - zápis dat typu TEXT a BYTE 10-15
 - získání počtu volných stránek 11-42
- prostor dbspace
 - definice kořenového prostoru dbspace 10-13
 - definovaný 10-11
 - dočasný 10-14
 - inteligentní velké objekty 10-9
 - kořenový 10-14
 - monitorování inteligentních velkých objektů 11-50
 - názvy 11-9
 - omezení, přesun protokolů 15-16
 - omezení, přidání protokolů 15-12
 - přejmenování 11-22
 - přidání
 - pomoci programu ISA 11-10
 - přidávání
 - blok 11-20
 - zrcadlený blok 19-7
 - souvislost mezi logickými a fyzickými paměťovými jednotkami 10-11
 - tabulka
 - definovaná 8-21
 - metadata 8-21
 - vypouštění 11-35
 - tabulka ve sdílené paměti 8-21
 - účel 10-11
 - určení velikosti stránky při vytvoření 11-14
 - velikost stránky, určení 11-14
 - velikosti stránek jiné než výchozí v prostředí replikace HDR 21-5
 - vypouštění
 - blok 11-32
 - ON-Monitor 11-35
 - onspaces 11-35
 - přehled 11-34
 - vytvoření 11-10, 11-14
 - dočasného 11-19

- prostor dbspace (*pokračování*)
 - vytvoření (*pokračování*)
 - počátečního prostoru dbspace 10-13
 - použití obslužného programu onspaces 11-11
 - vytvoření počátečního prostoru dbspace 1-19
 - zálohování 11-10, 11-11
 - zrcadlení, pokud jsou obsaženy soubory logického protokolu 18-4
- prostor extspace
 - vypuštění pomocí obslužného programu onspaces 11-37
 - vytvoření 11-36
- prostor sbospace
 - definování replikačního serveru 10-16
 - definovaný 10-16
 - dočasný
 - příklad 11-31
 - příznak LO_CREATE_TEMP 11-31
 - vytvoření 11-31
 - Enterprise Replication 10-16
 - metadata 10-17
 - názvy 11-9
 - oblast uživatelských dat 10-17, 11-51
 - odlehčený vstup - výstup 8-15
 - omezení
 - přidání protokolů 15-12
 - použití příkazu onstat -g smb c 11-52, 11-58
 - protokolování 14-9
 - přesunutí prostoru z rezervované oblasti 11-57
 - rezervovaná oblast 10-17
 - rychlé zotavení 10-16
 - schopnost zotavení 10-16
 - soubory JAR 1-16
 - statistické údaje pro uživatelem definovaná data 2-4
 - stránky sbpage 10-9
 - ukládání inteligentních velkých objektů 8-41, 10-16
 - určení paměťových charakteristik 11-28
 - určení velikosti oblasti metadat 11-29
 - vytvoření
 - pomocí obslužného programu onspaces 11-28
 - pomocí programu ISA 11-29
 - zálohování 11-28, 11-30
 - změna paměťových charakteristik 11-30
 - zrcadlení 11-28
- prostor tblspace
 - definovaný 10-36
 - dočasné, během inicializace 4-7
 - obsah tabulky 8-22
 - typy obsažených stránek 10-37
 - velikost, oblast, pro prostor tblspace tblspace 11-12
 - vypustit dočasné 4-7
- prostor v předpřipraveném souboru
 - definovaný 1-6, 10-5
 - porovnání s prostorem s přímým přístupem 10-5
 - pro statická data 10-5
 - přidělení 11-5
- prostory tblspace
 - monitorování
 - tabulka systabnames 11-47
- prostředí
 - konfigurační soubor 1-10
 - řídící aplikace 1-11
- prostředí X/Open
 - prostředí DTP 8-22
- protokol
 - určení 6-27
- protokol dvoufázového potvrzování
 - aktivita účastníka 23-6
 - automatické zotavení 23-7
 - role administrátora 23-7
 - zotavení koordinátora 23-6
 - zotavení účastníka 23-7
 - definice 23-4
 - distribované dotazy 1-25
 - fáze po rozhodnutí 23-6
 - fáze před potvrzením 23-7
 - heuristické rozhodnutí
 - heuristické odvolání transakce 23-9
 - heuristické ukončení transakce 23-13
 - typy 23-8
 - chybové stránky 23-15
 - identifikátor globální transakce 24-4
 - kdy se používá 23-4
 - konfigurační parametr TXTIMEOUT 23-21
 - konfigurační parametry 23-21
 - nezávislá akce 23-8
 - definice 23-8
 - chyby 23-9
 - výsledky 23-8
 - vyvolávající 23-8
 - optimalizace předpokládaného přerušení 23-7
 - parametr DEADLOCK_TIMEOUT 23-21
 - předčasné odstranění globální transakce 24-2
 - přehled 1-25
 - role aktuálního serveru 23-5
 - rozdíl proti heterogennímu potvrzování 23-21
 - vyprázdnění záznamů protokolu 23-17
 - záznamy logického protokolu 23-10
 - zotavení koordinátora 23-4
 - zotavení účastníka 23-7
 - zprávy 23-5
- protokol Dynamic Host Configuration Protocol 3-12
- protokol Internet Protocol verze 6 3-34
- protokol IPX/SPX
 - položka sqlhosts 3-40
 - služba, definovaná 3-23
 - v poli názvu hostitele 3-40
 - v poli servicename 3-23
- protokol ontlime 3-21
- protokol potvrzování
 - dvoufázový 23-2, 23-4

- protokol potvrzování (*pokračování*)
 - heterogení 23-2
- protokol zpráv
 - během inicializace 4-7
 - definovaný 1-26
 - dynamicky přidávané protokoly 15-19
 - fyzické zotavení 16-4
 - poškození dat 22-6
 - využití metadat 11-57
 - zobrazení zpráv 1-26
- protokolování
 - Viz též* stav protokolování databáze.
 - aktivita, která se vždy protokoluje 12-4
 - data typu TEXT a BYTE 14-8
 - databáze kompatibilní se standardem ANSI 13-4
 - definované protokolování transakcí 12-3
 - Enterprise Replication 10-16
 - fyzické protokolování
 - definice 16-2
 - pokyny týkající se velikost 16-4
 - potlačení v dočasných prostorech dbspace 10-14
 - proces 16-7
 - indexy R-stromu 16-5
 - inteligentní velké objekty 8-41, 10-27, 14-11
 - kdy použít tabulky protokolování 14-10
 - metadat a uživatelských dat 10-26
 - obnovení k bodu v čase 10-31
 - potlačení pro implicitní tabulky 10-14
 - používání protokolování transakcí 12-6
 - proces
 - data prostoru blobspace 14-8, 14-9
 - data prostoru dbspace 14-12
 - procesy databázového serveru vyžadující 12-2
 - prostory sbpace 10-22, 14-9
 - prostředí DTP 12-9
 - protokolování transakcí s vyrovnávací pamětí 12-7
 - replikace Enterprise Replication 12-3
 - role v replikaci dat 20-8
 - tabulky
 - souhrn 10-30
 - tabulky typu RAW 10-31
 - tabulky typu STANDARD 10-31
 - typy tabulek 13-6
 - vliv použití vyrovnávací paměti na rychlost zaplnění logického protokolu 14-5
 - změna tabulky za účelem vypnutí 13-6
 - změna tabulky za účelem zapnutí 13-6
 - zobrazení stavu pomocí programu ISA 13-7
- protokolování bez vyrovnávací paměti
 - Viz též* protokolování.
 - vyprázdnění vyrovnávací paměti logického protokolu 8-38
- protokolování databáze
 - prostředí DTP 12-9
 - replikace Enterprise Replication 12-3
- protokolování databáze (*pokračování*)
 - zálohy 11-10
- protokolování transakcí
 - Viz též* protokolování
 - bez vyrovnávací paměti 12-7
 - definice 12-3, 12-6
 - kdy použít 12-6
 - replikace Enterprise Replication 10-16
 - s vyrovnávací pamětí 12-7
- protokolování transakcí kompatibilní se standardem ANSI.
 - Viz* protokolování.
- protokolování transakcí s vyrovnávací pamětí
 - Viz též* protokolování.
 - kdy jsou vyprázdněny 12-8
- proudové propojení procesů
 - pole servicename 3-23
 - výhody a nevýhody 3-8
- provádění
 - operaci jazyka DDL v primárním serveru 20-27
- provázaný režim 23-5
- prověřování
 - definice 5-7
 - konfigurační parametry 2-14
 - přehled 1-24
 - režim auditu 6-34
- průměrná velikost inteligentního velkého objektu 11-29
- předobraz
 - definice 16-2
 - obsah 16-16, 16-18
 - vyrovnávací paměť fyzického protokolu 8-35
- předobrazy
 - vyprazdňování 8-35
- přehrání protokolu, rychlá obnova 16-17, 16-20
- přejmenování
 - instance databázového serveru 1-15
 - prostoru dbspace 11-22
- přenosy dat v síti
 - šifrování 5-11
- přepínání
 - mezi jednotkové procesy 6-12
 - následující soubor protokolu 15-5
- přepínání kontextu
 - definované 6-10
 - jak pracuje, pokud je řízeno operačním systémem 6-8
 - porovnání operačního systému a provozu s vícenásobnými jednotkovými procesy 6-7
- přeskakování fragmentů
 - použití vlastností 11-37
 - vliv na transakce 11-39
- přeskočení fragmentů
 - všechny fragmenty 11-37
 - všechny nedostupné 11-39
 - vybrané fragmenty 11-39
- převod, během inicializace 4-6
- převzetí totožnosti klientu 3-15

- převzetí totožnosti uživatele 3-15
- přidání
 - virtuálních procesorů 6-13
- přidělení diskového prostoru
 - metadata 11-28
 - počáteční blok 11-7
 - prostor v předpřipraveném souboru 11-5
 - přehled 1-6
 - sdílená paměť 1-5
 - UNIX
 - předpřipravené soubory 11-5
 - soubory s přímým přístupem 11-6
 - Windows
 - fyzická jednotka 11-8
 - logická jednotka 11-8
 - soubory systému NTFS 11-7
- Přidělení diskového prostoru
 - procedura 11-3
- přidělení disku 1-6
- přidělování diskového prostoru
 - oblast 10-10
 - prostory sbpace 10-24
- Přidělování diskového prostoru
 - zrcadlená data 19-3
- příkaz ALLOCATE COLLECTION 12-6
- příkaz ALLOCATE DESCRIPTOR 12-6
- příkaz ALLOCATE ROW 12-6
- příkaz ALTER ACCESS METHOD 12-5
- příkaz ALTER FRAGMENT 12-5
- příkaz ALTER FUNCTION 12-5
- příkaz ALTER INDEX 12-5
- příkaz ALTER OPAQUE 12-5
- příkaz ALTER PROCEDURE 12-5
- příkaz ALTER ROUTINE 12-5
- příkaz ALTER TABLE 12-5
 - protokolování 12-5
 - připojení ke klientům 3-36
 - změna typu tabulky 10-32
- příkaz BEGIN WORK 12-6
- příkaz CLOSE 12-6
- příkaz COMMIT 12-6, 12-7
- příkaz CONNECT 3-36, 12-6
 - příklad 3-37
- příkaz CREATE ACCESS METHOD 12-5
- příkaz CREATE AGGREGATE 12-5
- příkaz CREATE CAST 12-5
- příkaz CREATE DATABASE 10-11, 12-5
- příkaz CREATE DISTINCT TYPE 12-5
- příkaz CREATE EXTERNAL TABLE 12-5
- příkaz CREATE FUNCTION 6-21, 12-5
- příkaz CREATE INDEX 12-5
- příkaz CREATE OPAQUE TYPE 12-5
- příkaz CREATE OPCLASS 12-5
- příkaz CREATE PROCEDURE 12-5
- příkaz CREATE ROLE 5-7, 12-5
- příkaz CREATE ROUTINE 12-5
- příkaz CREATE ROW TYPE 12-5
- příkaz CREATE SCHEMA 12-5
- příkaz CREATE SYNONYM 12-5
- příkaz CREATE TABLE 12-5
 - fragmentace
 - s oddíly 11-23
 - protokolování 12-5
 - připojení ke klientům 3-36
 - určení prostorů sbpace v klauzuli PUT 10-18
 - volba IN prostor_dbspace 10-11
- příkaz CREATE TEMPORARY TABLE 12-5
- příkaz CREATE TRIGGER 12-5
- příkaz CREATE VIEW 12-5
- příkaz DATABASE 3-36, 12-6
- příkaz DEALLOCATE COLLECTION 12-6
- příkaz DEALLOCATE DESCRIPTOR 12-6
- příkaz DEALLOCATE ROW 12-6
- příkaz DECLARE 12-6
- příkaz DELETE 12-5
- příkaz DESCRIBE 12-6
- příkaz DISCONNECT 12-6
- příkaz DROP ACCESS METHOD 12-5
- příkaz DROP AGGREGATE 12-5
- příkaz DROP CAST 12-5
- příkaz DROP DATABASE 12-5
- příkaz DROP FUNCTION 12-5
- příkaz DROP INDEX 12-5
- příkaz DROP OPCLASS 12-5
- příkaz DROP PROCEDURE 12-5
- příkaz DROP ROLE 12-5
- příkaz DROP ROUTINE 12-5
- příkaz DROP ROW TYPE 12-5
- příkaz DROP SYNONYM 12-5
- příkaz DROP TABLE 12-5
- příkaz DROP TRIGGER 12-5
- příkaz DROP TYPE 12-5
- příkaz DROP VIEW 12-5
- příkaz EXECUTE 12-6
- příkaz EXECUTE FUNCTION 12-6
- příkaz EXECUTE IMMEDIATE 12-6
- příkaz EXECUTE PROCEDURE 12-6
- příkaz FETCH 12-6
- příkaz FLUSH 12-5
- příkaz FREE 12-6
- příkaz GET DESCRIPTOR 12-6
- příkaz GET DIAGNOSTICS 12-6
- příkaz GRANT 12-5
 - při přidělení oprávnění uživatelům DataBlade 5-38
- příkaz GRANT DEFAULT ROLE 5-8
- příkaz GRANT FRAGMENT 12-5
- příkaz GRANT ROLE 5-7
- příkaz imcadmin 3-44
- příkaz INFO 12-6
- příkaz INSERT 12-5

- příkaz jazyka SQL
 - nikdy neprotokolovaný 12-6
- příkaz LOAD 11-58, 12-5
- příkaz LOCK TABLE 12-6
- příkaz OPEN 12-6
- příkaz OUTPUT
 - protokolování 12-6
- příkaz PREPARE 12-6
- příkaz PUT 12-5
- příkaz RENAME COLUMN 12-5
- příkaz RENAME DATABASE 12-5
- příkaz RENAME INDEX 12-5
- příkaz RENAME TABLE 12-5
- příkaz REVOKE 12-5
 - při přidělení oprávnění uživatelům DataBlade 5-38
- příkaz REVOKE DEFAULT ROLE 5-8
- příkaz REVOKE FRAGMENT 12-5
- příkaz ROLLBACK WORK 12-5
- příkaz SELECT 12-6
- příkaz SELECT INTO TEMP 12-5
- příkaz SET 5-7
- příkaz SET ... 12-6
- příkaz SET DATASKIP 11-37, 11-38
- příkaz SET ENCRYPTION PASSWORD 5-23
- příkaz SET ROLE DEFAULT 5-7, 5-8
- příkaz SQL
 - ALTER TABLE 10-10
 - analýza a optimalizace 9-7
 - CREATE TABLE 10-11
 - CREATE TABLE, klauzule PUT 10-18
 - databáze s protokolováním 12-6
 - DECLARE 12-5
 - monitorování 13-6
 - počáteční připojení k databázovému serveru 1-19
 - použití dočasného diskového prostoru 10-34
 - UPDATE STATISTICS 8-25
 - vytvoření prostoru dbspace 10-18
 - vždy protokolovaná 12-4
- příkaz START VIOLATIONS TABLE 12-5
- příkaz starts dbservername 1-17
- příkaz STOP VIOLATIONS TABLE 12-5
- příkaz tail -f 1-27
- příkaz TRUNCATE TABLE 12-5
- příkaz UNLOAD 12-5
- příkaz UNLOCK TABLE 12-6
- příkaz UPDATE 12-5
- příkaz UPDATE STATISTICS 2-4, 8-25, 10-31, 12-5
- příkaz WHENEVER 12-6
- příkazy jazyka SQL
 - CREATE ROLE 5-7
 - GRANT DEFAULT ROLE 5-8
 - GRANT ROLE 5-7
 - nové vlastnosti xxv
 - REVOKE DEFAULT ROLE 5-8
 - SET ROLE 5-7
- příkazy jazyka SQL (*pokračování*)
 - SET ROLE DEFAULT 5-7, 5-8
- příklad
 - jak je spouštěno čištění 8-32
 - konfiguračního parametru DBSERVERALIASES 3-36
 - položky souboru /etc/services 3-11
 - připojení IPX/SPX 3-40
 - připojení prostřednictvím místní zpětné smyčky 3-39
 - připojení prostřednictvím sdílené paměti 3-38
 - připojení TCP/IP 3-40
 - více typů připojení 3-41
- přímá disková zařízení 1-6
- připojení
 - bezpečnostní omezení 3-13
 - databáze a síť 3-5
 - definice 6-28
 - klientských aplikací k serveru 3-6
 - metody 6-26
 - místní zpětná smyčka
 - definovaná 3-9
 - příklad 3-39
 - multiplexní 3-5
 - pole typu 3-19
 - prostřednictvím sdílené paměti, definované 3-7
 - protokol IPX/SPX 3-40
 - síťové, příklad 3-40
 - TCP/IP 3-10, 3-12, 3-13
 - více
 - typů připojení, příklad 3-41
- připojení ke sdílené paměti 8-8
 - obslužné programy databázového serveru 8-9
 - virtuální procesory 8-9
- připojení prostřednictvím pojmenovaného propojení procesů
 - definované 3-6, 3-8
 - podporované platformy 3-6
- připojení prostřednictvím sdílené paměti
 - pole servicename 3-23
 - příklad 3-38
 - virtuální procesor 6-26
 - vyrovnávací paměti zpráv 8-26
 - způsob připojení klientu 8-8
- připojení TCP 6-27
- připojený adresář systému NFS 10-4
- příručky online xxxv
- příručky pro instalaci xxxii
- přírůstkové zálohování
 - definované 1-21
- přístup k disku bez vyrovnávací paměti
 - porovnání s přístupem s vyrovnávací paměti 10-5
 - uložení dat 10-5
- příznak LO_CREATE_LOG 10-27
- příznak LO_CREATE_NOLOG 10-27
- příznak LO_CREATE_TEMP 10-27, 11-31
- příznak LO_LOG 14-11
- příznak LO_NOLOG 14-11

příznak MAC 5-21
příznak přepínání 5-22
PSWDCSM
Viz šifrování hesla.

R

RAID

Viz pole Redundant Array of Inexpensive Disks.

referenční omezení 10-31, 12-2

registr

definování vícenásobných síťových adres 6-33

změna názvu hostitele 1-30

registr HKEY_LOCAL_MACHINE 1-12

registr sqlhosts

definovaný 3-16

umístění 3-16

v systému Windows 1-12

relace

aktivní prostor tblspace 8-22

definovaná 6-10

jednotkové procesy 8-23

jednotkové procesy sqlxec 6-4

primární jednotkový proces 8-23

rychlá vyrovnávací paměť rutin UDR 8-26

rychlá vyrovnávací paměť slovníku 8-25

řídící blok 6-10

sdílená paměť 8-23

společná oblast sdílené paměti 8-18

zámky 8-18

replikace dat 12-3

Viz též replikace High-Availability Data Replication

datové typy, které lze replikovat 20-3

doba čekání na odpověď 2-12

interval vyprazdňování 2-12

použití skupin databázových serverů 3-27

přehled 1-24

replikace Enterprise Replication 1-24

replikace High-Availability Data Replication 1-24

restartování po selhání 21-15

restartování, je-li poškozen index sekundárního

serveru 21-15

režim 20-3

režim pouze pro čtení 4-9

velikosti stránek jiné než výchozí v prostředí replikace

HDR 21-5

volba zabezpečení 5-37

vyrovnávací paměť 8-17, 20-8

replikace Enterprise Replication 5-37

protokolování databáze 12-3

replikace HDR, současné použití 20-6

replikace HDR

administrace 21-11

konfigurace propojitelnosti 21-4, 21-6

monitorování stavu 21-18

nastavení 21-3

replikace HDR (pokračování)

typ databázového serveru 21-18

obnovení systému po selhání prostředku 21-20, 21-21

plánování 21-2

požadavky na data 21-3

požadavky na hardware a operační systém 21-3

replikace Enterprise Replication 20-6

restartování po selhání 21-22, 21-25

restartování po zjištění, že je index porušen 21-22

skripty hdrmkpri a hdrmksec 21-18

spuštění 21-7

stav bloku na sekundárním 21-13

změna režimu databázového serveru 21-16

změna typu databázového serveru 21-18

replikace High-Availability Data Replication

Viz též replikace dat.

akce, které je třeba provést, pokud

dojde k selhání primárního serveru 20-15

dojde k selhání sekundárního serveru 20-14

asynchronní aktualizace 20-10

definovaná 20-3

jak jsou replikovány aktualizace 20-8

jak pracuje 20-7

konfigurační parametr DRINTERVAL 20-9, 20-10

konfigurační parametr DRTIMEOUT 20-13

moduly DataBlade, současná instalace 20-7

možná selhání 20-13

návrh klientů

primární databázový server 20-25

návrh klientů k použití

sekundárního databázového serveru 20-26

počáteční replikace 20-7

provádění operací DDL 20-27

přesměrování klientů

DBPATH 20-18

INFORMIXSERVER 20-22

porovnání metod 20-24

soubor sqlhosts 20-19

z aplikace 20-23

restartování po selhání 20-17

role

dočasný prostorů dbspace 20-27

primárního databázového serveru 20-4

sekundárního databázového serveru 20-4

záznamy protokolu 20-8

ruční přepnutí 20-15, 20-17

rutiny UDR, současná instalace 20-7

synchronizace 20-12

synchronní aktualizace 20-9

typy replikovaných dat 20-3

typy UDT, současná instalace 20-7

výhody 20-3

zjišťování selhání 20-13

ztracené a nalezené transakce 20-10

- replikační server.
 - Viz* replikace dat.
- rezervovaná oblast
 - definovaná 10-17
 - monitorování velikosti 11-52, 11-58
 - přesunutí prostoru do oblasti metadat 11-57
- rezervovaná stránka PAGE_CONFIG 2-15, 4-6, 4-7
- rezervovaná stránka PAGE_PZERO 4-6
- rezervované stránky
 - ověření 22-3
 - stav bloku na sekundárním serveru 21-13
- rezidentní sdílená paměť
 - definovaná 4-5, 8-12, 8-13
 - interní tabulky 8-20
- režim
 - definovaný 4-9
 - do režimu offline z kteréhokoliv režimu 4-15
 - jednouživatelský 4-9, 4-14
 - klidový 4-9
 - nenásilné vypnutí 4-13
 - offline 4-9
 - okamžité vypnutí 4-14
 - online 4-9
 - opětovná inicializace sdílené paměti 4-12
 - přesun souborů protokolu 15-16
 - přidání souborů protokolu 15-13
 - server replikace HDR 20-3
 - uvedení do režimu offline 4-15
 - vypnutí 4-10
 - vypuštění souborů protokolu 15-14, 15-15
 - z klidového do jednouživatelského 4-14
 - z offline do jednouživatelského 4-13
 - z offline do klidového 4-12
 - z offline do online 4-12
 - z online do jednouživatelského 4-14
 - z online do klidového
 - nenásilně 4-13
 - okamžitě 4-14
 - změna 4-10
 - zotavení 4-9
- režim offline
 - definovaný 4-9
- režim online
 - definovaný 4-9
- režim pouze pro čtení
 - definovaný 4-9
- režim velkých bloků 1-5
- role
 - definice 5-8
 - výchozí 5-7
 - vytvoření 5-8
- rozdělené čtení 18-2, 18-6
- rozhraní API DataBlade
 - inteligentní velké objekty, přístup 10-16, 10-24
 - velikost inteligentních velkých objektů 10-20
- rozhraní pro monitorování systému
 - Viz* tabulka SMI.
- rozhraní Transport-Layer Interface
 - v poli nettype 3-20
- rozložení zatížení
 - pomocí parametru DBSPACETEMP 10-43
 - pomocí virtuálních procesorů 6-7
 - s cílem maximalizovat výkon 10-42
- rozšířená uživatelská oprávnění 1-30
- rozšířené datové typy
 - replikace HDR 20-3
- rozšiřující virtuální procesor
 - Viz* uživatelský virtuální procesor
- rozvržení disků
 - izolace tabulek 10-44
 - kompromisy 10-44
 - optimální výkon 10-42
 - správci logických disků 10-48
 - ukázkové rozvržení disků 10-44
 - zrcadlení 10-43
- ruční obnova
 - postup zjištění, zda je nezbytná 24-1
 - použití identifikátoru GTRID 24-4
 - příklad 24-6
 - rozhodnutí, zda je nutná akce 24-5
 - určení, zda jsou data nekonzistentní 24-3
 - získání informace ze souborů logického protokolu 24-3
- rutina SPL
 - konfigurační parametr PC_HASHSIZE 2-8
 - konfigurační parametr PC_POOLSIZE 2-8
 - rychlá vyrovnávací paměť
 - Viz* rychlá vyrovnávací paměť rutin UDR.
 - velikost hashovací tabulky rychlé vyrovnávací paměti rutin UDR 2-8, 9-5
- rutiny v jazyku SPL
 - určení velikosti společné oblasti 8-26
 - velikost hashovací tabulky rychlé vyrovnávací paměti rutin UDR 8-26
- rychlá obnova 4-6, 4-9, 12-2
 - alternativní volby pro operace typu fuzzy 16-21
 - bez protokolování 16-15
 - definovaná 1-23, 16-14
 - fáze čištění 15-13
 - inteligentní velké objekty 10-26
 - jak databázový server zjistí, kdy je potřeba 16-14
 - její účel 16-14
 - kdy nastane 16-14
 - podrobnosti procesu 16-15, 16-18
 - prostory sbospace 10-16
 - přetečení fyzického protokolu 16-15
 - typy tabulek 10-32
 - vliv protokolování s vyrovnávací pamětí 16-15
- rychlá vyrovnávací paměť
 - Viz též* vyrovnávací paměť.
 - distribuce dat 8-25, 9-5

- rychlá vyrovnávací paměť *(pokračování)*
 - monitorování vyrovnávacích pamětí ve sdílené paměti 9-11, 9-13
 - optického podsystému 11-47
- rychlá vyrovnávací paměť pro příkazy SQL
 - konfigurační parametry 2-9
 - povolení 9-7
 - určení velikosti 9-7
- rychlá vyrovnávací paměť rutin SPL
 - velikost hashovací tabulky 2-8, 9-5
 - velikost společné oblasti 9-5
- rychlá vyrovnávací paměť distribuce dat 8-25, 9-5
- rychlá vyrovnávací paměť optického podsystému
 - ID klientského uživatele 3-25, 11-48
 - ID uživatelské relace 11-48
 - kilobajty zapsaných dat typu TEXT a BYTE 11-48
 - počet zapisovaných objektů 2-14, 11-48, 11-49
 - přídělování 11-49
 - velikost 11-48
- rychlá vyrovnávací paměť pro příkazy SQL
 - definovaná 8-25
 - konfigurace 9-7
 - monitorování 9-7
 - přehled 2-9
 - umístění ve sdílené paměti 8-19
 - určení velikosti 2-9
- rychlá vyrovnávací paměť rutin UDR 8-26
- rychlá vyrovnávací paměť slovníku 8-25

Ř

řazení

- sdílená paměť 8-25

řídící blok jednotkového procesu

- role při přepínání kontextů 8-24
- vytvoření 8-23

řídící blok relace 6-10

- definovaný 8-23
- sdílená paměť 8-23

řídící struktury

- definované 6-10
- fronty 6-13
- řídící blok jednotkového procesu 6-10
- řídící blok relace 6-10
- zásobníky 6-12

S

sada dokumentace systému Informix Dynamic Server xxxv

sada dokumentace všech příruček xxxv

sada IBM Informix Client Software Developer's Kit 23-3

sbspace

- čas posledního přístupu 10-21
- diskové struktury 10-24
- dočasný
 - definovaný 10-25
 - přidávání bloků 10-26, 11-29

sbspace *(pokračování)*

- dočasný *(pokračování)*
 - vypouštění bloků 10-26, 11-33
 - vypuštění prostoru sbpace 11-34
 - zálohování a obnovení 10-26
- charakteristiky 10-25
- kontrola konzistence 10-25
- lehké operace vstupu - výstupu 10-21
- metadata 8-41, 10-20
- oblast uživatelských dat 8-41
- protokolování 10-22
- přidávání bloků 10-24, 11-29
- přidělení prostoru 10-24
- přírůstkové zálohy 15-4
- režim ukládání do vyrovnávací paměti 10-21
- určení paměťových charakteristik 10-20
- výpočet metadat 10-24
- vypouštění
 - pomocí obslužného programu onspaces 11-34
 - sloupců typů CLOB a BLOB 11-34
- zálohování 10-24
- zamykání 10-21

sdílená data 8-3

sdílená paměť

- část 8-4
- data jednotkového procesu 8-23
- definovaná 8-2
- globální společná oblast 8-26
- haldy 8-24
- hashovací tabulka vyrovnávacích pamětí 8-21
- hodnota klíče 8-9, 8-10
- identifikátor 8-9
- identifikátor segmentu 8-10
- inicializace 4-2, 4-4, 4-6
- inteligentní velké objekty 8-41, 10-24
- interní tabulky 8-20
- izolace jednotkových procesů a vyrovnávací paměti
 - zámků 8-27
- komunikace 3-22, 3-24
- komunikační prostředky 8-26
- konfigurace 8-6, 8-19, 9-2
- konfigurační parametr SERVERNUM 8-9
- konfigurační parametr SHMADD 8-19
- konfigurační parametr SHMBASE 2-7, 8-9
- konfigurační parametr SHMTOTAL 2-7, 8-7
- konfigurační parametr SHMVRTSIZE 2-7, 8-19
- konfigurační parametr STACKSIZE 8-24
- konfigurační parametry 2-8
- kontrolní bod 16-8
- kopírování do souboru 9-9
- kritické části 16-2
- meziprocesová komunikace 8-3
- monitorování
 - ISA 9-6
 - onstat 9-9

sdílená paměť *(pokračování)*

- nerезидентní část 9-5
- objekty mutex 8-27
- obsah 8-27
- pokyny pro semaforey 9-4
- problém s adresou dolní hranice 8-11
- první segment 8-9
- přehled 1-23
- přídělení 1-5, 4-5
- přídělování 8-18
- přídělování vyrovnávacích pamětí 8-14
- připojení 6-27, 8-8
- připojení dalších segmentů 8-9, 8-11, 9-6
- připojování obslužných programů 8-9
- резидентní část
 - definovaná 8-13
 - ON-Monitor 9-6
 - program ISA 9-6
 - textový editor 9-6
 - vytvoření 4-5
- rychlá vyrovnávací paměť distribuce dat 8-25
- rychlá vyrovnávací paměť pro příkazy SQL 2-9, 8-6, 8-25, 9-7
- rychlá vyrovnávací paměť slovníku 8-25, 9-6
- rychlá vyrovnávací paměť uživatelských rutin 8-26
- řazení 8-25
- řídící blok jednotkového procesu 8-23
- řídící blok relace 8-23
- segmenty operačního systému 8-6
- společná oblast vyrovnávacích pamětí 8-13, 16-7
- společné oblasti 8-18
- stránky blobpage 8-39
- synchronizace vyprazdňování vyrovnávacích pamětí 8-34
- tabulka bloků 8-21
- tabulka jednotkových procesů čištění stránek 8-22
- tabulka prostorů dbspace 8-21
- tabulka prostorů tbspace 8-22
- tabulka transakcí 8-22
- tabulka uživatelů 8-23
- tabulka vyrovnávacích pamětí 8-20
- tabulka zrcadlených bloků 8-21
- tabulky 8-20
- účel 8-2
- velikost
 - virtuální část 8-19
 - zobrazená programem onstat 8-7
- virtuální část 8-18, 8-19, 9-6, 9-9
 - program ISA 9-6
- vliv parametrů operačního systému 9-2
- volby výkonu 8-3, 9-7
- vyrovnávací paměť 11-17
- vyrovnávací paměť fyzického protokolu, 8-16
- vyrovnávací paměť logického protokolu 8-15
- vyrovnávací paměť replikace dat 20-8

sdílená paměť *(pokračování)*

- záhlaví 8-11, 8-13
- zámky vyrovnávacích pamětí 8-27
- zásobníky 8-24
- změna uložení pomocí obslužného programu
 - onmode 9-8
 - znovu inicializovat 9-8
- sdílená paměť, protokol ipcshm a komunikační část 8-26
- sdílené zámky
 - definované 8-28
- segment
 - Viz blok.
- segment datového typu
 - Viz diskový prostor.
- segment syntaxe xxx
- sekundární databázový server 20-4, 21-13
- selhání médií
 - zotavení po 18-2
- selhání systému
 - vliv na databázi 16-14
- semafor, parametry systému UNIX 9-4
- server Apache 4-11
- server LDAP 5-30
- servery NIS, vliv na soubory /etc/hosts a /etc/services 3-11
- seznam volných stránek
 - Viz blok, seznam volných stránek.
- síť
 - karty rozhraní
 - použití 6-33
 - konfigurační soubory 1-12
 - soubory zabezpečení 1-12
 - šifrování dat 5-22
- síťová komunikace
 - implementace 6-28
 - prostřednictvím protokolu IPX/SPX 3-22, 3-23, 3-40
 - prostřednictvím protokolu TCP/IP 3-21, 3-22
 - typy 6-26
- síťové protokoly 6-27
- síťové virtuální procesory
 - definice 6-26
 - jednotkové procesy cyklického dotazování 6-27
 - počet 6-28
- síťový protokol
 - definovaný 3-2
- skript hdrmkpri 21-18
- skript hdrmksec 21-18
- skript ids-example.rc 1-18
- skupina
 - bargroup 1-7
 - databázový server 5-6
 - databázových serverů 3-27, 3-28, 3-30
 - paralelní zpracování 6-8
- skupina bargroup 1-7
- skupina databázových serverů 3-27

- skupina Informix-Admin
 - správa souboru protokolu 15-2
 - vlastnictví souboru 1-7
 - změna provozních režimů 4-11
- sledování výkonu v systému Windows 1-29
- služba
 - v síti IPX/SPX 3-23
- služba Network Information Service 3-11
- služba Windows Internet Name Service 3-12, 3-31
- služby pro podporu komunikace
 - definované 3-9
 - utajení a integrity zpráv 3-9
- smíšený výsledek transakce 23-12
- softwarové závislosti xx
- sokety 3-6
 - v poli nettype 3-20
- souběžný přístup
 - řízení 8-27
- soubor
 - core 22-4
 - hosts 1-12
 - konfigurace 1-13
 - konfigurace propojitelnosti 3-10
 - konfigurace sítě 1-12
 - NTFS 10-4
 - oncfg_servername.servernum 4-7
 - ONCONFIG 1-13, 4-4
 - oprávnění 11-5
 - passwd 1-12
 - předpřipravený 1-6
 - services 1-12
 - soubor hosts.equiv 3-13
 - sqlhosts, 1-11
 - vlastnosti JVP 1-16
 - zabezpečení sítě 1-12, 3-13
- soubor .informix 1-10
- soubor .rhosts 3-14
- soubor /etc/hosts 20-21
- soubor /etc/services 20-22
- soubor conccsm.cfg 5-11, 5-12
 - umístění 5-12
 - vytváření tabulek SMI 5-14
 - záznam pro šifrování hesel 5-14
 - záznam pro šifrování síťových dat 5-22
- soubor core 22-4
- soubor dbservername.cmd 1-11, 1-14
- soubor hosts 1-12, 3-10, 3-11
- soubor hosts.equiv 3-13
- soubor chyby aserce
 - af.xxx 22-5
 - gcore.xxx 22-5
 - seznam 22-4
 - shmem.xxx 22-5
- soubor informix.rc 1-10
- soubor JAR 1-16
- soubor logického protokolu
 - Viz též* soubor logického protokolu.
 - alarmy událostí 15-18, 15-20
 - číslo souboru protokolu 14-3
 - definovaný 12-2, 14-1
 - dočasný 15-9
 - dynamické přidělování
 - definice 15-10
 - monitorování 15-18, 15-20
 - umístění 15-12
 - velikost 15-11
 - velikost souboru 15-11
 - jedinečné identifikační číslo 14-3
 - minimální a maximální velikost 14-4
 - následky neuvolnění 14-7
 - nelze přidat k prostoru blobospace nebo sbospace 15-12
 - nelze přidat k prostoru dbospace s jinou než výchozí velikostí stránky 15-12, 15-16
 - odhad potřebného počtu 14-5, 15-3
 - odstranění 15-14
 - používání tabulek typu SMI 15-10
 - pozice v protokolu 23-15
 - přepínání 14-8, 15-5
 - přepnutí za účelem aktivace prostorů blobospace 14-8
 - přesun do jiného prostoru dbospace 15-16
 - přidání souboru protokolu
 - pomocí programu ON-Monitor 15-14
 - pomocí příkazu onparams 15-13
 - přidělení diskového prostoru pro 14-4
 - role v rychlé obnově 16-16, 16-17, 16-20
 - ruční přidání souborů protokolu 15-12
- soubor logického protokolu
 - role v rychlé obnově 16-21
 - změna velikosti 15-13
- stav
 - A 15-5, 15-13, 15-15
 - B 15-6, 15-7
 - C 15-7
 - D 15-6, 15-15
 - definovaný 14-3
 - F 15-15
 - L 15-7
 - U 15-6, 15-7, 15-15
- umístění 14-2, 15-12
- úplný kontrolní bod pro uvolnění 17-6
- uvolnění souborů 14-7, 15-6, 15-7
- virtuální procesor LIO 6-22, 6-23
- vstup - výstup 6-22, 6-23
- vypuštění souboru protokolu
 - pomocí programu ON-Monitor 15-15
 - pomocí příkazu onparams 15-14
- záloha
 - cíle 15-4
 - přidání souborů protokolu 15-13
 - uvolnění odstraněných stránek blobpage 14-9

- soubor logického protokolu (*pokračování*)
 - záloha (*pokračování*)
 - vliv na výkon 14-5
 - změna fyzického schématu 11-10
 - změna velikosti 15-15
 - zrcadlení prostoru dbspace, který obsahuje soubor 18-4
- soubor netrc
 - definovaný 3-14
 - volby zabezpečení v informacích sqlhosts 3-29
- soubor oncfg_servername.servernum 4-7
- soubor onconfig.demo 1-13
- soubor onconfig.std
 - informace o společné oblasti vyrovnávacích paměti 11-15
- soubor services 1-12, 3-11
- soubor setenv.cmd 1-10
- soubor shmem
 - chyby aserce 22-4
- soubor sqlhosts
 - definování vícenásobných síťových adres 6-33
 - definovaný 3-26
 - multiplexní připojení 3-5
 - pole dbservername 3-19, 3-28
 - pole názvu hostitele 3-21
 - pole názvu služby 3-22, 6-28
 - pole typu připojení 3-19
 - pole voleb 3-22
 - příklad místní zpětné smyčky 3-39
 - příklad sdílené paměti 3-38
 - příklad síťového připojení 3-40
 - příklad více typů připojení 3-40
 - úprava pomocí programu ISA 1-12
 - určení síťových jednotkových procesů cyklického dotazování 6-27
 - volba CSM 3-24
 - volba identifikátoru 3-27
 - volba keep-alive 3-29
 - volba konce skupiny 3-27
 - volba skupiny 3-29
 - volby zabezpečení 3-29
 - záznamy pro více karet rozhraní 6-33
- soubor šablony onconfig.std 1-9, 1-13
- soubor vlastností JVP 1-16
- soubor známých a opravených chyb xxxiii
- souborový server NetWare 3-22
- souborový vstup - výstup
 - Viz* diskový vstup - výstup
- soubory klíčů MAC 5-16
- soubory NTFS
 - převodění 1-6
- soubory operačního systému.
 - Viz* prostor v předpřipraveném souboru.
- soubory systému NTFS 10-4
- souvislý
 - prostor pro fyzický protokol 17-2
- spící jednotkové procesy
 - spící navždy 6-14
 - typy 6-14
- spojení
 - paralelní zpracování 6-8
- společná oblast vyrovnávacích pamětí
 - 64bitové adresování 8-15
 - čtení napřed 8-33
 - definovaná 8-13
 - inteligentní velké objekty 8-41
 - kontrolní bod typu fuzzy 16-9, 16-10
 - minimální požadavky 8-14
 - monitorování aktivity 9-13
 - obcházení v případě dat prostoru blobspace 8-39
 - obsah 8-13
 - pro 32bitové platformy 11-17
 - pro 64bitové platformy 11-17
 - pro jiné než výchozí velikosti stránek 11-15
 - správa front LRU 8-29
 - synchronizace vyprazdňování vyrovnávacích pamětí 8-35
 - úplný kontrolní bod 16-9
 - velikost vyrovnávací paměti 8-15
 - vytvoření pro jiné než výchozí velikosti stránek 11-14
 - vytvoření pro jiné než výchozí velikosti stránky 11-15
- spouštěcí skript 1-17
- správa disku 1-6
- správce logických disků
 - definovaný 10-48, 18-4
- správce paměti
 - ISM 1-20
 - role v systému ON-Bar 1-20
- správce transakcí
 - MTS/XA 23-2, 23-3
 - TP/XA 1-25, 23-2
 - účel 23-2
- správce zdrojů 23-2
- spuštění databázového serveru
 - automaticky 1-18
 - inicializace diskového prostoru 1-16
 - pomocí oninit 4-2
- SPWDCSM 5-11
- standard AES
 - Viz* standard AES (Advanced Encryption Standard)
- standard AES (Advanced Encryption Standard) 5-10
- standard DES (Data Encryption Standard) 5-10
 - Viz* šifrování
- standard DES3 (Triple Data Encryption Standard) 5-10
- standardní databázový server 20-4
- statistické údaje distribuce dat 8-25
- statistické údaje o paměti
 - prostory blobspace 11-27
- statistické údaje.
 - Viz* obslužný program onstat.

stav

- soubor protokolu
 - aktuální 15-6
 - kontrolní bod 15-15
 - odstraněný 15-7, 15-15
 - použitý 15-6
 - používaný 2-9, 15-15
 - přidaný 15-6, 15-7, 15-15
 - volný 15-6, 15-15
 - zálohovaný 15-7, 15-15

stav protokolování databáze

- definice 12-6
- kompatibilní se standardem ANSI 12-8
 - změna režimu 13-4
- nastavení 12-3
- protokolování bez vyrovnávací paměti 12-7
- protokolování s vyrovnávací paměti 12-8
- přehled 1-22
- ukončení protokolování
 - pomocí obslužného programu ondblog 13-3
 - pomocí obslužného programu ontape 13-5
- zajištění kompatibility se standardem ANSI
 - pomocí obslužného programu ondblog 13-4
 - pomocí obslužného programu ontape 13-5
- zapnutí protokolování pomocí obslužného programu ontape 13-4
- změna
 - pomocí obslužného programu ontape 13-4
 - pomocí programu ISA 13-6
 - pomocí programu ON-Monitor 13-6
 - používání obslužného programu ondblog 13-3
- změna stavu použití vyrovnávací paměti
 - pomocí obslužného programu ondblog 13-3
 - pomocí obslužného programu ontape 13-5
 - pomocí příkazu SET LOG 12-9
- změny povoleny 13-2
- zrušení protokolování pomocí obslužného programu ondblog 13-3

stránka

- definovaná 10-6
- určení velikosti pro standardní nebo dočasný prostor dbspace 11-14
- určení velikosti stránky databázového serveru 11-27
- vztah k bloku 10-7

stránka blobpage

- definovaná 10-7
- fyzické protokolování 16-4
- statistické údaje o paměti 11-27
- uvolnění odstraněných stránek 14-9
- vztah k bloku 10-8
- zápisy, které obcházejí sdílenou paměť 8-39

stránka sbpage

- definovaná 10-9
- doporučení velikosti 10-9

stránky

- naposledy použité 8-30
- nejdéle nepoužívané 8-29
- stránky bitových map
 - prostor tbspace 10-37
 - prostor tbspace indexu 10-37
- stránky blobpage
 - doporučení velikosti 11-26
 - statistické údaje o paměti 11-27
 - zaplnění
 - určení 11-49
 - zobrazení příkazem oncheck -pB 11-49
- synchronizační kontrolní bod
 - Viz kontrolní bod.
- systémová konzole 1-28
- systémový časovač 6-14

Š

šifra

- definice 5-10
- šifra Blowfish 5-16
 - v šifrování 5-16
- šifrovací karta 5-20
- šifrovací virtuální procesor 6-33
- šifrování
 - dat ve sloupci 5-23, 5-25
 - data sloupce 5-25
 - datových přenosů 5-11
 - definice 5-10
 - hesla 5-11
 - na úrovni sloupců 5-23, 5-25
 - přenosů dat v síti 5-11
 - režimy 5-10
 - virtuální procesory 6-33
 - volby serveru Dynamic Server 5-10
- šifrování dat 5-10
 - data v určených sloupcích 5-10
 - hesla 5-10
- šifrování hesel 5-11
 - inicializace databázového serveru 5-12
 - konfigurační soubor CSM 5-12, 5-14

T

tabulka

- definovaná 10-29
- dočasná
 - odhad diskového prostoru 10-41
- fragmentace 10-31
- oblast 10-10, 10-29
- oddíly uprostřed disků 10-46
- posouzení potřebné paměti 10-42
- pravidla pro rozvržení disků 10-44
- standardní 10-31
- vypouštění 11-34
- vytvoření s datovými typy CLOB a BLOB 10-24

- tabulka FAT
 - oddíly 1-6
- tabulka jednotkových procesů čištění stránek
 - definovaná 8-22
 - počet položek 8-22
- tabulka přeměrování 11-4
- tabulka SMI
 - během inicializace 4-8
 - concsn.cfg 5-14
 - monitorování
 - bloky 11-40
 - databáze 9-10, 13-7
 - databázi 11-40
 - informace ve vyrovnávací paměti 9-13
 - kontrolní body 17-7
 - objekty mutex 9-10
 - oblast 11-47
 - použití vyrovnávací paměti protokolu 17-7
 - prostorů tblspace 11-47
 - replikace dat 21-20
 - sdílená paměť 9-10
 - soubory logického protokolu 15-10
 - společná oblast vyrovnávacích paměti 9-11, 17-7
 - typy zápisu 9-15
 - virtuální procesory 7-8
 - využití vyrovnávací paměti 11-46
 - zámky latch 9-10
 - použití k monitorování databázového serveru 1-28
 - přerušení vytváření tabulky 4-8
 - příprava během inicializace 4-8
 - sysexents 11-47
 - systabnames 11-47
- tabulka sysdistrib 1-28, 8-25
- tabulka syslogs 15-10
- tabulka sysprofile 9-10
- tabulka sysroleauth 5-38
- tabulka systables
 - hodnoty příznaků 10-30
- tabulka sysvpprof 7-8
- tabulka transakcí
 - definovaná 8-22
 - zaznamenávání pomocí příkazu onstat 8-22
- tabulka typu RAW 1-22
 - obnovení 10-33
 - přehled 10-31
 - rychlé zotavení 10-33
 - vlastnosti 8-33
 - zálohování 10-33
 - zálohování a obnovení 10-33
 - změna 13-6
- tabulka typu STANDARD
 - obnovení 10-32, 10-33
 - povolena v databázi s protokolováním 1-22
 - rychlé zotavení 10-33
 - vlastností 10-31
- tabulka typu STANDARD (*pokračování*)
 - zálohování 10-33
- tabulka typu TEMP 1-22
 - obnovení 10-33
- tabulka uživatelů
 - definovaná 8-23
 - maximální počet položek 8-23
- tabulka zámků
 - definovaná 8-17
 - konfigurace 8-18
 - obsah 8-17
- tabulky
 - dočasné
 - vyčištění při restartu 10-36
 - izolace často používaných 10-44
 - tabulky bez protokolování 10-30
 - tabulky systémového katalogu
 - optimální uložení 10-46
 - ověření 22-3
 - rychlá vyrovnávací paměť slovníku 8-25
 - tabulka sysdistrib 8-25
 - umístění 10-28
- terminálové rozhraní 1-9
- textový editor
 - nastavení konfiguračních parametrů
 - sdílená paměť 9-6
 - virtuální procesory 7-2
 - výkon 9-6
 - vytvoření souboru ONCONFIG 1-13, 7-2
- tištěné příručky xxxv
- TLI
 - Viz* rozhraní Transport-Layer Interface.
- TP/XA 1-25, 23-2
- transakce
 - globální
 - definice 23-2, 23-5
 - identifikační číslo, GTRID 24-4
 - určení, zda byla implemenována nekonzistentně 24-1
 - zaznamenání 13-7, 23-15
 - monitorování 13-7, 23-5, 23-15
 - provázané 23-3
 - smíšený výsledek 23-12
 - tabulky typu RAW 10-31
 - volně vázané 23-3
- třída VP v konfiguračním parametru NETTYPE 6-27
- třídění
 - paralelní zpracování 6-8
- třídy
 - virtuální procesor 6-4
- tučné písmo xxvi
- typografické konvence xxvi
- typy tabulek
 - hodnoty příznaků v tabulce systables 10-31
 - obnovení 10-33
 - RAW 10-30

- typy tabulek (*pokračování*)
 - rychlé zotavení 10-33
 - s přímým přístupem (RAW) 13-6
 - souhrn 10-31
 - STANDARD 10-30, 13-6
 - v protokolujících databázích 1-22
 - vlastnosti 10-33
 - zálohování před převodem 11-10
- typy zápisů
 - zápis na popředí 8-36
 - zápisy bloků 8-37
 - zápisy front LRU 8-36
- typy zápisů vyrovnávacích paměti 8-36

U

- účastnický databázový server 23-6
 - automatické zotavení 23-6
- ukončovací skript 1-17
- úlohy administrace
 - kontrola konzistence 22-2
 - přiřazování paměti 10-11
 - rutinní úlohy 1-21
- uložená procedura
 - Viz* rutina SPL.
- uložení
 - Viz* vícenásobné uložení.
- uložení dat
 - Viz též* diskový prostor.
 - koncepty 10-2
 - maximální
 - velikost bloku 11-4
 - maximální blok
 - velikost 11-9
 - maximální počet bloků 11-9
 - maximální počet paměťových prostorů 11-9
 - řízení 10-11, 10-17
- úplný kontrolní bod
 - definice 16-9
 - nejstarší aktualizace 16-11
 - poslední dostupný protokol 14-7
 - události, které ho vyvolají 16-11
 - vynucení 17-6
 - vyprázdnění fyzického protokolu 16-13
 - vyprázdnění společné oblasti vyrovnávacích pamětí 16-10, 16-13
- upozornění
 - soubory v systémech NIS 3-11
 - výstup příkazu oncheck -cc 22-3
- úpravy
 - soubor ONCONFIG 1-14
 - údaje sqlhosts
 - UNIX 1-12
- určení adresy pomoci zástupných znaků
 - klientská aplikace 3-32
 - pole názvu hostitele 3-34
- určení adresy pomoci zástupných znaků (*pokračování*)
 - příklad 3-34
- úroveň izolace neaktualizované čtení 20-27
- úrovně, zálohování
 - definované 1-21
 - prostory sbospace 15-4
- usnadnění xxxv
 - diagramy syntaxe ve formátu desítkových čísel oddělených tečkami A-1
 - diagramy syntaxe, čtení pomocí nástroje pro čtení obsahu obrazovky A-1
- utajení předávaných zpráv 3-9
- uvolnění souborů protokolu 15-6, 15-7
- uživatel informix
 - správa souboru protokolu 15-2
 - změna hesla 1-30
- uživatелеm definovaný virtuální procesor
 - kolik 6-20
 - použití 6-20
 - účel 6-20
- uživatelská data
 - definovaná 10-17
- uživatelská rutina
 - externí 5-38
 - konfigurace rychlé vyrovnávací paměti 8-26
 - nesprávně pracující 6-21
 - paralelní zpracování 6-8
 - psaná v jazyku Java 1-16
 - registrace 5-38
 - umístění sdílené paměti 6-20
 - virtuální procesor nepředávající řízení 6-8
 - virtuální procesory 6-17
- uživatelské jednotkové procesy
 - monitorování 8-23
 - zaznamenávání 8-23
- uživatelské připojení
 - monitorování 4-8
- uživatelské rutiny
 - rychlá vyrovnávací paměť 8-26
- uživatelské účty a domény systému Windows 3-3
- uživatelské virtuální procesory
 - spouštění rutin UDR 8-26
- uživatelský jednotkový proces
 - definovaný 6-4
 - kritické části 16-2
 - získání vyrovnávací paměti 8-30

V

- varovná zpráva
 - pro kontrolu zabezpečení prováděnou obslužnými programy 5-4
- vázání virtuálních procesorů třídy CPU 6-9
- velikost počáteční oblasti 10-10
- velikost stránky jiná než výchozí
 - v prostředí replikace HDR 21-5

- velikost stránky jiné než výchozí
 - pro prostor dbspace 11-14
- velikost vyrovnávací paměti
 - volba 3-25
- velké vyrovnávací paměti, definované 8-23
- vestavěný datový typ
 - operace typu fuzzy 16-9
 - replikace HDR 20-3
- větev transakce 23-5
- více typů připojení
 - Viz též* připojení
 - informace sqlhosts 3-36
 - příklad 3-41
- vícenásobné jednotkové procesy
 - definované 6-3
- vícenásobné uložení
 - příklad 3-42
- víceprocesorový počítač
 - konfigurační parametr MULTIPROCESSOR 6-17
 - procesorová afinita 6-9
- virtuální část
 - sdílená paměť
 - část virtuálních rozšíření 8-4
 - globální společná oblast 8-26
 - konfigurace 8-19
 - konfigurační parametr SHMVRTSIZE 8-19
 - obsah 8-19
 - přidání segmentu 9-9
 - rychlá vyrovnávací paměť pro příkazy SQL 8-25
 - rychlá vyrovnávací paměť rutin UDR 8-26
 - zásobníky 8-24
 - sdílené paměti
 - rychlá vyrovnávací paměť distribuce dat 8-25
- virtuální procesor
 - během inicializace 4-6
 - definovaný 6-2
 - diskový vstup - výstup 6-22
 - fronta připravených procesů 6-13
 - jednotkové procesy cyklického dotazování 6-27
 - koordinace přístupu ke zdrojům 6-8
 - monitorování
 - obslužné programy onstat 7-6
 - pomocí programu ISA 7-3
 - nastavení konfiguračních parametrů 7-1
 - nepředávající řízení 6-21
 - obsluhující jednotkové procesy 6-10
 - paralelní zpracování 6-8
 - použití zásobníků 6-12
 - proces s vícenásobnými jednotkovými procesy 6-6
 - přehled 6-2
 - přepínání kontextu 6-10
 - přesouvání jednotkových procesů 6-3
 - přidání a vypuštění
 - ON-Monitor 7-3
 - pomocí programu ISA 7-3
 - virtuální procesor (*pokračování*)
 - připojení ke sdílené paměti 8-9
 - přístup ke sdílené paměti 8-5
 - rozšíření.
 - Viz* Uživatelem definovaný virtuální procesor.
 - sdílení zpracování 6-7
 - síť 6-26
 - správa 1-23
 - šifrování 6-33
 - třída ADM 6-14
 - třída ADT 6-34
 - třída AIO 6-25, 8-9
 - třída CPU 6-8
 - třída LIO 6-23
 - třída MSC 6-5
 - třída OPT 6-34
 - třída PIO 6-13, 6-24
 - třídy 6-8, 6-16
 - uživatelem definovaná třída 6-20
 - uživatelská rutina 6-5
 - uživatelské rutiny napsané v jazyku Java 6-22
 - vázání k jednotkám CPU 6-9
 - vstup - výstup fyzického protokolu 6-24
 - vstup - výstup logického protokolu 6-23
 - výhody 6-6
 - vypuštění CPU v režimu online 7-3
 - vypuštění virtuálního procesoru CPU v režimu online 7-5
 - virtuální procesor auditu 6-34
 - virtuální procesor bezpečný při použití více jednotkových procesů 6-5
 - virtuální procesor CPU
 - definice 6-16
 - jednoprocesorový počítač 6-17
 - jednotkové procesy cyklického dotazování 6-27, 6-28
 - konfigurace parametru AFFNPROCS 6-17
 - konfigurační parametr AFF_SPROC 6-17
 - moduly DataBlade 6-17
 - omezení 6-20
 - počet 6-16
 - prováděné typy jednotkových procesů 6-16
 - přidávání a vypouštění v režimu online 6-18, 6-22
 - uživatelská rutina 6-17
 - vázání 6-9
 - víceprocesorový počítač 6-17
 - virtuální procesor jazyka Java 6-22
 - virtuální procesor nepředávající řízení 6-21
 - virtuální procesor třídy miscellaneous 6-34
 - virtuální procesory CPU
 - jednotkové procesy 6-3
 - virtuální procesory LIO 6-23
 - virtuální procesory PIO
 - definice 6-24
 - kolik 6-26

- vkládání dat
 - tabulky typu RAW 10-31
 - tabulky typu STANDARD 10-31
- vlastnosti
 - vylepšení jazyka SQL xxv
 - vlastnosti serveru IDS 10.0 xxi
 - vlastnosti serveru IDS 9.21 xxv
 - vlastnosti serveru IDS 9.3 xxiv
 - vlastnosti serveru IDS 9.4 xxiii
 - volba identifikátoru 3-29
 - volba keep-alive 3-29
 - volba konce skupiny 3-26
 - volba přeměrování připojení 3-25
 - volba zabezpečení 5-37
 - volně vázaný režim 23-3
 - vrstvení odstranění 12-3
- vstup - výstup.
 - Viz* diskový vstup - výstup.
- výchozí
 - konfigurační soubor 1-13, 4-5
 - role
 - použití 5-7
 - udělení oprávnění 5-7
 - vytvoření 5-7
- výchozí národní prostředí xxi
- výkon
 - čtení napřed 8-33
 - frekvence vyprazdňování vyrovnávacích pamětí 8-31
 - funkce předávání řízení 6-8
 - nástroj k monitorování 1-29
 - parametry, nastavení
 - pomocí programu ON-Monitor 9-7
 - posouzení VP CPU 6-16
 - přepínání kontextu řízení virtuálním procesorem 6-11
 - rezidentní sdílená paměť 8-12
 - sběr dat 1-28
 - sdílená paměť 3-8, 6-8, 8-3
- vylepšení jazyka SQL xxv
- vylepšení použitelnosti
 - verze 10.0 xxi
 - verze 9.4 xxiii
- vylepšení výkonu
 - verze 9.3 xxv
- výlučný zámek
 - vyrovnávací paměť 8-28
- výměna disků za chodu 18-4
- vynucená rezidence
 - nastavení 4-7
- vynucení kontrolního bodu
 - Viz* kontrolní bod, vynucený
- výpis do souboru core
 - Viz* též DUMP CNT; DUMP DIR; DUMP CORE; DUMP SHMEM.
 - kdy je užitečný 22-8
 - obsah souboru gcore.xxx 22-5
- vypnutí
 - automaticky 1-18
 - nenásilné 4-13
 - okamžité 4-14, 4-15
 - režim
 - definovaný 4-10
 - uvedení do režimu offline 4-15
- výpočet velikosti
 - inteligentní velké objekty 10-20
 - kořenový prostor dbspace 10-40
 - metadat 11-29
 - metadata 10-24
 - stránek blobpage 11-26
 - velikost stránky 11-27
- vyprazdňování
 - předobrazů 8-35
 - vyrovnávací paměť replikace dat, maximální interval 2-12
 - vyrovnávacích pamětí 8-34
- vypuštění
 - bloku z prostoru dbspace 11-32
 - paměťové prostory 11-34
 - prostoru extspace 11-37
 - prostorů sbpace 11-33
 - tabulky v prostorech dbspaces 11-35
- vyrovnávací paměť
 - aktualizovaná 8-34
 - aktuální úroveň přístupu zámku 8-20
 - čekající jednotkové procesy 8-20
 - inteligentní velké objekty 8-41, 10-21
 - jak jednotkový proces
 - přistupuje ke stránce ve vyrovnávací paměti 8-33
 - maximální počet na 64bitové platformě 8-15
 - monitorování statistik a použití 9-11
 - replikace dat 8-17, 20-8
 - sdílené zámky 8-28
 - souběžný přístup 8-33
 - synchronizace vyprazdňování 8-35
 - tabulka vyrovnávacích pamětí, definovaná 8-20
 - typy zámků 8-28
 - velké vyrovnávací paměti 8-23
 - výlučný režim 8-28
 - vyprazdňování 8-34
 - vyrovnávací paměť fyzického protokolu 8-16
 - vyrovnávací paměť logického protokolu 8-15
 - vyrovnávací paměť stránek blobpage 8-40
- vyrovnávací paměť fyzického protokolu
 - definovaná 8-17
 - konfigurační parametr PHYSBUFF 8-17
 - kontrolní body 8-35
 - monitorování 17-4
 - počet 8-17
 - protokolování prostoru dbspace 16-7
 - události, které způsobují vyprázdnění 8-35
 - vyprazdňování 8-17

- vyrovnávací paměť logického protokolu
 - definovaná 8-15
 - kontrolní body 8-15
 - monitorování 17-4
 - vyprazdňování
 - bez předobrazu 8-38
 - databáze bez protokolování 8-38
 - definované 8-37
 - pokud nastal kontrolní bod 8-38
 - protokolování bez vyrovnávací paměti 8-38
 - synchronizace 8-34
 - vyrovnávací paměť logického protokolu 8-37
 - záznamy logického protokolu 8-37
- vyrovnávací paměť pro přijatá data 20-8
- vyrovnávací paměti
 - naposledy použité 8-30
 - neaktualizované 8-34
 - nejdéle nepoužívané 8-29
 - synchronizace vyprazdňování 8-34
 - typy zápisů při vyprazdňování 8-36
 - způsob, jakým jednotkový proces získá paměť 8-30
- vyrovnávacích pamětí
 - vyprazdňování 8-34
- vytváření
 - inteligentních velkých objektů 11-29
- vytvoření
 - dočasných prostorů dbspace 11-19
 - inteligentní velké objekty 10-18
 - prostory blobspace 11-25
 - prostory dbspace 11-10
 - prostory sbpace 11-28
 - tabulek s datovými typy CLOB a BLOB. 10-24
- vzdálení
 - hostitelé a klienty 3-13
- vzdálený
 - klient 3-13
- vzorec
 - velikost fyzického protokolu 16-5
 - velikost logického protokolu 15-2

W

- Windows
 - automatické spuštění 1-17
 - konfigurace paměti 1-4
 - nastavení proměnných prostředí 1-10
 - nastavení propojitelnosti 1-12
 - obslužný program ipasswd 1-30
 - obslužný program ixsu 1-30
 - obslužný program ntchname 1-30
 - ovládací panel proměnných prostředí 1-10
 - prohlížeč událostí 1-29
 - převod na NTFS 1-7
 - přidělení diskového prostoru s přímým přístupem 11-8
 - větší sdílená paměť 1-5

- Windows *(pokračování)*
 - záležitosti týkající se sdílené paměti 9-3

X

- X/Open
 - prostředí DTP 12-9

Z

- zabezpečení
 - audit události v databázi 5-7
 - kontrola prováděná obslužnými programy serveru před spuštěním v operačním systému UNIX 5-2, 5-4
 - modul PAM (Pluggable Authentication Module) 5-27
 - ochrana před zahlcovacími útoky typu DOS (denial-of-service) 5-8
 - oprávnění adresáře INFORMIXDIR 5-5
 - pomocí modulů pro podporu komunikace (CMS) 5-11
 - použití šifrování na úrovni sloupce 5-23
 - pro externí rutiny 5-38
 - pro uživatelské rutiny DataBlade 5-38
 - prostřednictvím modulu LDAP Authentication Support Module 5-30
 - prostřednictvím rolí 5-7
 - rizika komunikace prostřednictvím sdílené paměti 3-8
 - soubory pro síť 1-12
 - volby
 - sqlhosts 3-29
 - volby šifrování 5-10
 - vylepšení ve verzi 10.0 xxi
 - zakázání kontroly prováděné obslužnými programy serveru 5-3
- zabezpečení sítě
 - soubor .netrc 3-14
 - soubor hosts.equiv 3-13
 - soubory 3-13
- zahlcovací útoky typu DOS (Denial-of-service) 5-8
- záloha úrovně 0
 - kontrola konzistence 22-6
- zálohování
 - bloky 11-21
 - data typu TEXT a BYTE 10-15
 - definované 1-21
 - fyzický protokol 11-10
 - kontrolní body 16-13
 - odstraněné soubory protokolu 15-14
 - ověření 1-21
 - prostory blobspace 11-25
 - prostory dbspace 11-11
 - prostory sbpace 10-24, 11-28, 15-4
 - převod typu tabulky 11-10
 - přidání souborů protokolu 15-13
 - soubory protokolu 11-10, 15-4
 - strategie 1-3
 - tabulky typu RAW 10-31, 10-33
 - tabulky typu STANDARD 10-31

- zálohování (*pokračování*)
 - uvolnění souboru protokolu 15-6
 - změna protokolování databáze 11-10
 - zmenšení velikosti 10-14
- zálohování dat.
 - Viz* zálohování.
- zálohování serveru za plného provozu.
 - Viz* replikace dat.
- zámek
 - fronta čekajících procesů 6-15
- zámek latch
 - statistika monitorování 9-10
- zámky
 - definované 8-27
 - dynamické přidělování 8-18
 - fronta čekajících procesů 6-15
 - počáteční počet 8-18
 - příkaz onstat -k 23-15
 - typy 8-28
- zámky latch
 - Viz* objekty mutex.
- zámky spin
 - monitorování 9-10
- zamykání
 - inteligentní velké objekty 10-16
 - prostory sbpace 10-21
 - úroveň izolace neaktualizované čtení 20-27
- zamykání rozsahu bajtů 8-17, 10-21
- zápis na popředí
 - definovaný 8-36
 - monitorování 8-36, 9-14
 - předobraz 8-35
- zápisy front LRU
 - definované 8-36
 - monitorování 9-14
 - provádění 8-36
 - spuštění 8-36
- zařízení
 - NFS 10-4
 - v případě, že jsou potřeba posuny 11-4
 - znakové speciální 10-5
- zařízení systému UNIX
 - vlastnictví, oprávnění
 - předpřipravené soubory 11-5
 - znakově orientované 11-6
 - zobrazení odkazů k názvu cesty 1-7
- zařízení v systému UNIX
 - zobrazení propojení s cestou 11-6
- zasílání zpráv na pager
 - oznamování alarmů událostí 2-12
- zásobník
 - a řídicí blok jednotkového procesu 6-12
 - definovaný 6-12, 8-24
 - jednotkového procesu 8-24
 - konfigurační parametr STACKSIZE 8-24
- zásobník (*pokračování*)
 - proměnná prostředí INFORMIXSTACKSIZE 8-24
 - ukazatel 6-12
 - velikost 2-8, 8-24
- zaváděcí soubor
 - Viz* spouštěcí skript
- zavaděč High Performance Loader 10-32
- zavádění dat
 - expresní režim 10-32
 - metody 11-58
 - obslužné programy 11-58
- závislosti, softwaru xx
- záznam protokolu FREE_RE 11-58
- záznam protokolu CHKADJUP 11-30, 11-58
- záznam protokolu CHRESERV 11-30
- změna
 - režim protokolování, databáze ANSI 13-4
 - stav bloku 21-13
 - typ databázového serveru, replikace HDR 21-18
- znaková sada ISO 8859-1 xxi
- znakově orientovaná zařízení 1-6
- znakový velký objekt
 - Viz* inteligentní velký objekt.
- zotavení
 - paralelní zpracování 6-8
 - po selhání média 18-2
 - protokol dvoufázového potvrzování 23-4
 - režim
 - definovaný 4-9
 - tabulky typu RAW 10-31
 - tabulky typu STANDARD 10-31
- zpracování transakcí online
 - Viz* paralelní databázový dotaz
- zprávy fyzického zotavení 16-4
- znaková postižení
 - čtení diagramů syntaxe A-1
- zrcadlení
 - alternativy 18-3
 - během spuštění systému 19-4
 - během zpracování 18-6
 - bloků v replikaci HDR 21-14
 - činnost během zpracování 18-6
 - činnost zotavení 18-5
 - definovaný 18-1
 - doporučené rozvržení disků 10-43
 - hardware 18-4
 - když zrcadlení skončí 18-7
 - když zrcadlení začíná 18-4
 - náklady 18-2
 - obnovení bloku 19-5
 - povolení 19-2
 - požadované kroky 19-2
 - přehled 1-22
 - rozdělené čtení 18-6
 - síťové omezení 18-2

zrcadlení (*pokračování*)
spuštění 19-2, 19-4, 19-6, 19-7
stavový příznak 18-5
tabulka bloků 8-21
udržování souborů logického protokolu v prostoru
 dbspace 18-4
ukončení 19-8
výhody 18-2
výměna disků za chodu 18-4
vytvoření zrcadlených bloků 19-4
zastavení 19-8
zjišťování chyb média 18-6
změna stavu bloku 19-5

zrcadlený blok
 disková čtení 18-6
 diskové zápisy 18-6
 obnovení 19-5
 přidávání 19-7
 struktura 18-7
 vytvoření 19-4
 změna stavu 19-5
 zotavení 18-5, 18-7

zrcadlený prostor dbspace
 kořenový prostor dbspace 10-13
 vytvoření 11-7

Ž

žurnál předobrazu
Viz fyzický protokol.



Vytištěno v Dánsku společností IBM Danmark A/S.

G210-1606-00



Spine information:

IBM Informix Verze 10.0

Příručka administrátora serveru IBM Informix Dynamic Server

