



Velké databáze

High Performance

Databases

Jan Géryk, Miroslav Křipač
Vývojový tým IS MU

PV005, Služby počítačových sítí, 14. 12. 2011



Témata přednášky

1. Databázový systém
2. Historie databází
3. Integrita, normální normy
4. Databáze jako služba počítačové sítě
5. Architektury rozsáhlých databází
6. Distribuované databázové systémy



Databázový systém

- Uspořádaná množina dat
- Uložená na paměťovém médiu
- Nástroje pro manipulaci a správu
- Data
- Databázové objekty
 - pohledy, indexy, trigger, funkce, ...



Historie databází

- 50. léta, COBOL
 - hromadné zpracování dat
- 60. léta, síťové a hierarchické db
 - sálové počítače
- 70. léta, relační db, první spec. SQL
 - původně pro koncové uživatele
- 90. léta, objektové db
- Objektově-relační technologie
- Oracle, DB2, MySQL



Relační databáze

- Založená na relačním modelu
- Tabulky
 - dvourozměrné struktury
- Řádky jsou jednotlivé záznamy
- Cizí klíče
 - sloupce, informace o relaci
- Každá konkrétní tabulka realizuje podmnožinu kartézského součinu (relaci)
- Sloupce (atributy)
 - mají datový typ a doménu



Integrita databáze

- Konzistence vůči definovaným pravidlům
 - zajištění - integritní omezení
- Entitní integritní omezení
 - zajišťuje úplnost primárního klíče
- Doménová integritní omezení
 - dodržování datových typů sloupců
- Referenční integritní omezení
 - vazba primárního a cizího klíče



Vztahy mezi tabulkami

- Slouží ke svázání dat mezi tabulkami
 - tři základní vlastnosti
- Stupeň vztahu
 - unární, binární, ...
- Kardinalita vztahu
 - 1:1, 1:N, M:N
- Volitelnost účasti ve vztahu



Normální formy

- proces zjednodušování a optimalizace struktur db tabulek
- 1NF
 - neobsahuje složené hodnoty
- 2NF
 - závislé na celém klíči
- 3NF
 - žádné závislosti mezi neklíčovými atributy
- 4NF, 5NF



Databáze jako služba sítě

- **DB jako mechanismus přístupu k datům**
 - jednotné rozhraní pro efektivní vývoj a provoz malých aplikací
 - pouze jednouživatelský režim (více uživatelů => síťové FS)
- **DB jako služba sítě**
 - služba obsluhující aplikace přes vlastní jednotné rozhraní po síti
 - implementace vlastních síťových protokolů, ne sdílený síťový FS
 - server <-> více klientů = striktně klient – server pohled
(klientem z hlediska databáze může být i aplikační server)
 - služba odstíněná od uživatelů dalšími vrstvami
(aplikační, prezentační vrstvy)



Vlastnosti DB systémů

- **System pro zpracování transakcí splňující:**
 - **Atomičnost**
 - Transakce jsou zpracovány jako celek
 - **Konzistence**
 - Transakce uchovávají databázi v konzistentním stavu
 - **Izolovanost**
 - Jednotlivé operace jsou prováděny izolovaně vůči ostatním trans.
 - **Trvanlivost**
 - Data úspěšně ukončených transakcí musí být uložena trvanlivě



Zpracování transakcí

- Realizace – zámky
 - zamykání na úrovni řádků (ne tabulek, bloků)
 - vše zajišťuje databázový systém
- Kompromis: propustnost x konzistence
 - „read committed“ režim
 - každý příkaz čte co bylo commitnuto
 - ochrana před čtením dočasných dat (verze dat)
 - během transakce může dojít ke změně hodnoty
- Deadlock
 - prevence, **detekce**, obcházení, neřešení
 - kultura programování – pořadí zámků



Architektury rozsáhlých DB

- Principy spojení klienta s DB
 - aplikační rozhraní
 - klientské knihovny
 - spojení, session
- Způsoby zpracování požadavku
 - vyhrazený server
 - sdílený server
- Příklad: Architektura Oracle Database
 - Oracle instance + databáze
 - instance: SGA + procesy na pozadí



Vlastnost: dostupnost

- Ochrana před chybou (uživatel, HW, SW)
 - Uživatel: transakce (rollback), flashback (i DDL), PITR
 - HW: multiplexed redo logs, archive logs
 - SW: distribuované systémy
- Redo log
 - každá změna zapsaná **na redolog disk**
 - před ukončením commit
 - zápis samotných dat asynchronní (vyšší výkon)
- Undo records (undo/rollback segments)
 - zajišťuje verzování (původní data)
 - „before image“ u necommitnutých transakcí



Vlastnost: výkon

- Výkon jedné operace x propustnost celku
- Omezení klasické role OS
- Přístup na disk, velké množství dat
 - vlastní systém cache dat = **sdílená paměť** (pers.)
 - přímý (raw) přístup k disku, asynchronní
- Provádění příkazů
 - prepare, execute, { fetch }, ...
 - optimalizace přístupu k datům na úrovni DB serveru
 - uložení dat – rychlý zápis x rychlý přístup, (indexy, ...)



Distribuované DBS

- Motivace:
 - vysoká dostupnost, transparentní vůči aplikacím
 - navýšení propustnosti, horizontální zvýšení výkonu
- „share-nothing“ clustery
 - standby databáze
 - single point of failure
 - In-memory databáze
- „share-everything“ clustery
 - on-line sdílení dat více instancemi (čtení i zápis)
 - „global cache“ - sdílení na úrovni paměti