

Distribuované databáze – koncepce distribuovaného databázového systému, replikace a fragmentace dat, distribuovaná správa transakcí.

Thursday, May 30, 2013 8:20 AM

Distribuovaná DB je množina databází, která je uložena na několika počítačích. Uživatel se může jevit jako jedna velká databáze. V DB neexistuje žádný centrální uzel nebo proces odpovědný za vrcholové řízení funkcí celého systému. Výrazně to zvyšuje odolnost systému proti výpadkům jeho částí. Data i funkce rozděleny mezi více počítačů

Charakteristické vlastnosti

Distribuovaná DB je charakterizována:

- **Transparentnost** – z pohledu klienta se zdá, že data jsou zpracována na jednom serveru v lokální databázi. Uživatel si rozdělení nemusí být vědom.
- **Rozšiřitelnost** – zvýšení výkonu přidáním dalších počítačů.
- **Robustnost** – výpadek jednoho počítače neovlivní funkci ostatních
- Jsou syntakticky shodné příkazy pro lokální i vzdálená data, nespecifikuje se místo uložení dat (řeší to distribuovaný SŘBD)
- **Autonomnost** – s každou lokální bází dat zapojenou do distribuované databáze je možno pracovat nezávisle na ostatních databázích.
- Lokální Db je funkčně samostatná, připojení do jiné části distribuované db se v případě potřeby zřizují dynamicky.
- **Nezávislost na počítačové síti** – jsou podporovány různé typy architektur lokálních i globálních počítačových sítí (LAN, WAN)
- V distribuované databázi mohou být zapojeny počítače i počítačové sítě různých architektur, pro komunikaci se používá jazyk SQL.

Distribuované databáze jsou výhodné kvůli:

- **Lokální autonomie** – odpovídají struktuře decentralizovaných organizací. Data jsou uložena v místě nejčastějšího využití a zpracování – zlevnění provozu
- **Zvýšení výkonu** (rozdělení zátěže na více počítačů)
- **Spolehlivosti** (replikace dat, degradace služeb při výpadku uzlu, přesunutí na jiný uzel)
- **Rozšiřitelnosti**
- **Schopnosti sdílet informace** integrací podnikových zdrojů
- **Agregaci informací** – z více bází dat lze získat informace nového typu.

Naopak Distribuované Db mohou způsobit i pro ně specifické problémy:

- **Složitost** – distribuce db, distribuce zpracování dotazu a jeho optimalizace, složité globální transakční zpracování, distribuce katalogu, paralelismus a uvíznutí, složité zotavování z chyb
- **Cena** (komunikace je navíc)
- **Bezpečnost**
- **Obtížný přechod** – neexistuje automatický konverzní prostředek z centralizovaných DB na DDB

Taxonomie DDBS

- Těsně integrované

- Uživatel vidí data centralizovaná v jediné databázi, DDB je vybudována nad lokálními DB, každé místo má úplnou znalost o datech v celém DDBS a může zpracovávat požadavky používající data z různých míst.

- Semiautonomní

- Lokální DBMS pracují nezávisle a sdílejí svoje lokální data v celé federaci, jen část jejich dat je sdílena.

- Zcela autonomní

- Izolované, lokální DBMS pracují nezávisle a neví o ostatních DBMS, pro vzájemnou komunikaci potřebují softwarovou vrstvu pracující nad jednotlivými DBMS.

• Problémy replikace a fragmentace dat

U fragmentace a replikace bychom měli respektovat hlediska:

- Rozdělit relace do lokálních serverů tak, aby aplikace zatěžovaly servery stejnoměrně.
- Přístupnost a spolehlivost - Replikací zlepšíme spolehlivost a read-only dostupnost.
- Lokality zpracování (maximalizovat lokální).
- Dostupnosti a ceny paměti v jednotlivých uzlech

Replikace dat

Replikační transparentnost je neobtěžovat uživatele skutečností, že pracuje s daty existujícími ve více kopiích = uživatel neví o replikách.

Při replikaci dat se nachází kopie množiny objektů v každém uzlu, ve kterém je využívána. V systému tedy existuje několik kopií každého objektu. Výhodou tohoto řešení je kvalitní dostupnost, rychlý přístup ke každému objektu a menší nároky na komunikaci mezi uzly. Nevýhodou je problém duplicity, díky níž je nutné trvale *zajišťovat konzistenci všech kopií*. Je nutné implementovat systém, který určí správné pořadí provedených operací a při replikaci rozdistribuuje správnou kopii. Také je nutné zabránit současné modifikaci dvou kopií objektu.

Správné pořadí provedených operací je určováno většinou časovými razítky, současné modifikaci dvou kopií jednoho objektu mohou zabránit klasické paralelní synchronizační prostředky (zámky, semaforey apod.)

Jsou dva základní způsoby zacházení s replikami:

1. *Pouze pro čtení (master-slave)* - změny (zápis) může provádět jen master, repliky jsou read-only a periodicky se synchronizují s masterem; repliky tak mají lehce zastaralá, nekonzistentní data (tzv. eventual consistency, nakonec se dojde do konzistentního stavu)
2. *Pro transakční zpracování (multi-master)* - rovnocennější, změny možné provádět v replikách, je proto nutná obousměrná synchronizace, mohou tak vznikat konflikty, které se řeší různě (priority, časová razítka, manuálně, vlastní procedura)

Jiný zdroj:

Replikace = uchování kopií relací v různých uzlech.

výhody: porucha v jednom uzlu neznemožní přístup k jeho lokálním relacím, data v lokálních bázích jsou připravena k použití okamžitě, bez nutnosti přenosu

nevýhody: ztížení aktualizace, všechny kopie musí být aktualizovány současně a v průběhu aktualizace je nutno uzamknout aktualizovaná data ve všech uzlech sítě.

Proto se replikují data, ke kterým je potřebný rychlý přístup a která nejsou často aktualizována, příp. která jsou pro systém mimořádně důležitá (číselníky, registry ap.)

[http://barborka.vsb.cz/prednasky/presentations/2006-DAIS-vp-Databazove_a_informacni_systemy/Prezentace/Dais_7.pdf]

Fragmentace dat

Fragmentační transparentnost = uživatelův dotaz je specifikován na celou relaci, ale musí být vykonán na jejím fragmentu = uživatel neví o fragmentech.

Fragmentace dat se dělí na:

- Horizontální – dle selekční podmínky rozdělíme tabulku na 2 části horizontálním řezem, tedy např. na knihy s ISBN nižším než 122 a na knihy s ISBN větším nebo rovným 122.
- Odvozená horizontální – dochází k rozdělení tabulky na 2 a více částí horizontálním řezem, v tomto případě však je fragmentace založena na jiné relaci. Např. DODAVATELE a KNIHY. DODAVATELE rozdělíme do fragmentů a na základě těchto fragmentů provedeme fragmentaci v tabulce KNIHY, která je spojena s DODAVATELE relací.

- Vertikální – rozdělení tabulky podle sloupců na 2 a více částí. V jedné skupině jsou jedny sloupce, v druhé jiné.
- Smíšená – tabulku např. rozdělíme dle sloupců (vertikální) a následně v jednotlivých částech provedeme horizontální fragmentaci (a nebo obráceně).

Jiný zdroj:

Fragmentace = rozložení relace na části (fragments), které jsou umístěny v různých uzlech sítě. Může jít o horizontální fragmentaci, kdy se v různých uzlech ukládají části relace rozložené do skupin řádků, nebo o vertikální fragmentaci, kdy se v uzlech ukládají různé projekce relace. Fragmentace se provádí tak, aby bylo možno z fragmentů získat původní relaci standardními operacemi nad relační databází (sjednocením nebo spojením).

kat	jmeno			
400				
449				
456				

kat	jmeno	plat		

Distribuovaná správa transakcí

- Pro provedení transakční změny dat ve více databázích najednou
 - K tomu se využívá **dvoufázový commit** (2PC)
 - Jde o protokol řešení distribuovaných transakcí
 - Řídící uzel (koordinátor) řídí průběh celé distribuované transakce
 - Ostatní zainteresované uzly poslouchají
- Fáze PREPARE**
 - **Koordinátor** zašle zprávu PREPARE všem uzlům účastnících se distrib. transakce
 - Koordinátor v této fázi nic nedělá, čeká jen na výsledky od ostatních uzlů
 - **Každý podřízený uzel** se pokusí provést fázi PREPARE
 - Neprovádí vlastní commit, transakce je pořád neukončená, změněné řádky zůstávají stále zamčené
 - Jen se zajistí, aby aktuální stav rozpracované transakce byl schopný přežít výpadek
 - Každý uzel odpoví zpět koordinátorovi PREPARED (pokud vše ok) nebo ABORT (pokud nastal problém)
 - Fáze COMMIT**
 - Pokud obdržel koordinátor od všech uzlů PREPARED, provede potvrzení svých lokálních změn (normální commit)
 - A všem uzlům zašle zprávu COMMIT
 - Pokud je aspoň 1 výsledek ABORT, provede ROLLBACK svých změn (standardní COMMIT) + všem uzlům zašle zprávu ROLLBACK
- Protokol je imunní vůči výpadku sítě/uzlu v libovolné fázi
 - Koordinátor si po celou dobu zpracování globální transakce udržuje data o stavu transakce a všech podtransakcích
 - Je tedy schopen transakci dokončit např. při obnovení spojení se vzdáleným uzlem

From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>>