

Transakce, dvoufázový uzamykací protokol, detekce uváznutí.

Thursday, May 30, 2013 8:17 AM

Transakce

Databázové transakce musí splňovat tzv. vlastnosti **ACID**

- **A - Atomicity**
Databázová transakce je jako operace dále nedělitelná (atomická). Provede se buď jako celek, nebo se neprovede vůbec (a daný databázový systém to dá uživateli na vědomí, např. chybovou hláškou).
- **C - Consistency - konzistentnost**
Při a po provedení transakce není porušeno žádné integritní omezení.
- **I - Isolation - izolovanost**
Operace uvnitř transakce jsou skryty před vnějšími operacemi. Vracením transakce (ROLLBACK) není zasažena jiná transakce, jinak i tato musí být vrácena. V důsledku tohoto chování může dojít k tzv. řetězovému vrácení (cascading rollback).
- **D - Durability - trvalost**
Změny, které se provedou jako výsledek úspěšných transakcí, jsou skutečně uloženy v databázi a již nemohou být ztraceny.

Globální vs. lokální

- Lokální transakce probíhá pouze na jediném uzlu.
- Globální (distribuovaná) transakce přesahuje rozsah jednoho uzlu.

Stavy transakce

- Aktivní - od počátku provádění transakce
- Částečně potvrzený - stav po provedení poslední operace transakce
- Chybný - nelze pokračovat v normálním průběhu transakce
- Zrušený - nastane po skončení operace ROLLBACK
- Potvrzený - po úspěšném vykonání COMMIT

Prováděné operace

Pro práci s transakcemi je nutné zavést následující operace:

- BEGIN - začátek transakce
- COMMIT - ukončení transakce a uložení dosažených výsledků do databáze
- ROLLBACK - odvolání změn - není-li definován savepoint, (místo, po které lze provedené změny vrátit zpět) tak návrat do stavu před započítím vykonávání transakce

Optimistické vs. pesimistické zamykání

- U **pesimistického zpracování** se v jeho průběhu změny zaznamenávají do dočasných objektů (například a nejčastěji: do řádků tabulek s příznakem dočasných dat, platných jen po dobu transakce) a teprve po přesunu/změně dat se odznačí příznak dočasnosti a data se stanou platnými. (Tento způsob se dá přibližně připodobnit přepisu souboru, při kterém se nejdříve nová verze souboru nakopíruje pod dočasným jménem a teprve poté se tento soubor přejmenuje za starý a tím ho nahradí.)
- U **optimistického zpracování** se (optimisticky) předpokládá, že při transakci nenastane chyba a nebude třeba ji vrátit zpět (přestože tato možnost je zachována). Měněné záznamy v tabulkách jsou při optimistickém zpracování transakce zapisovány „natvrdo“, současně s tím se však vytváří tzv. rollback log coby seznam SQL příkazů, které dokáží prováděné změny vrátit zpět. V případě, že při transakci dojde k nějaké nezotavitelné chybě, tento log se provede a transakce (aby dodržela pravidlo atomicity) skončí ve výchozím stavu s chybou. Naopak, na konci transakce, při které k žádné takové chybě nedošlo, se rollback log maže.

Žurnály

Jsou záznamy, které uchovávají informace o průběhu transakcí a slouží k zotavení po vzniklé chybě. Žurnály musí být v každém uzlu a obsahují záznamy o historii každé transakce.

Dvoufázový protokol (2PL)

Dvoufázová transakce v první fázi zamyká vše co je potřeba a od prvního odemknutí (druhá fáze) již jen odemyká co měla zamčeno (již žádná operace LOCK). Tedy transakce musí mít všechny objekty uzamčeny předtím, než nějaký objekt odemkne. Dá se dokázat, že pokud jsou všechny transakce v dané množině transakcí dobře formované a dvoufázové, pak každý jejich legální rozvrh je uspořádatelný. Dvoufázový protokol zajišťuje uspořádatelnost, ale ne zotavitelnost ani bezpečnost proti kaskádovému rušení transakcí nebo uváznutí.

Striktní dvoufázový protokol (S2PL)

Problémy 2PL jsou nezotavitelnost a kaskádové rušení transakcí. Tyto nedostatky lze odstranit pomocí striktních dvoufázových protokolů, které uvolňují zámky až po skončení transakce (COMMIT). Zřejmá nevýhoda je omezení paralelismu. 2PL navíc stále nevylučuje možnost deadlocku. Read-lock je tu možné uvolnit kdykoli během 2. fáze, write-lock jen na konci.

SS2PL

Prakticky se dnes používá SS2PL, což je Strong Strict twophase locking, které sice nevylučuje deadlock, ale pokud k němu dojde, tak ho umí automaticky vyřešit. Liší se od S2PL jen tím, že i read-lock uvolňuje až na konci.

Konzervativní dvoufázový protokol (C2PL)

Rozdíl oproti 2PL je ten, že transakce žádá o všechny své zámky, ještě než se začne vykonávat. To sice vede občas k zbytečnému zamykání (nevíme co přesně budeme potřebovat, tak radši zamkneme víc), ale stačí to již k prevenci uváznutí (deadlocku). Pomalý, je třeba vědět předem, co se bude číst/zapisovat, nepoužívá se.

Detekce uváznutí a zotavení

- Uváznutí se detekuje pomocí čekacího grafu
 - Vrcholy jsou transakce T_i
 - Orientovaná hrana $T_i \rightarrow T_j$ značí, že T_i čeká, až T_j odemkne datovou položku
 - Je-li v čekacím grafu cyklus, došlo k uváznutí
- Hledá se takový plán transakcí, aby se co nejmíň kryly a tak, aby byl dodržen princip ACID (hlavně Isolation), když se takový plán povede najít, nazývá se uspořádatelný
- **Well formed transakce** (správně zamykat a odemykat)
- **Když se zjistí uváznutí**
 - Je nutno nalézt obětní transakci a vnutit jí abort (a tím i obnovu dat). Obětuje se obvykle nejmladší transakce, tj. ta, která ještě neudělala mnoho změn
 - Transakce mohou stárnout, bude-li za oběť vybírána vždy nejmladší transakce. Proto je vhodné do kritéria výběru obětí zahrnout i počet transakcí provedených návratů.
 - Která data se ale mají obnovovat?
 - **Totální obnova** transakci úplně zruší, data se vrátí do počátečního stavu, a transakce se restartuje. To může být velmi nákladné
 - Efektivnější je, když se transakce "vrací postupně" do stavu, kdy uváznutí zmizí. Tento postup je ale náročný na evidenci kroků a změn transakcí provedených: metoda kontrolních bodů (**checkpointing**) – konzistentní mezistavy

Zajištění uspořádatelnosti pomocí pořadových čísel transakcí

Transakce vyvolá write $W(x)$:

1. $TSR(x) > TS(t)$: zápis do „později přečtené“ paměti > ROLLBACK
2. $TSW(x) > TS(t)$: zápis do „později přepsané“ paměti > ROLLBACK
3. jinak proved' zápis

Transakce vyvolá read $R(x)$:

4. $TSW(x) > TS(t)$: čtení z „později zapsané“ paměti > ROLLBACK
5. jinak read

Využití časových razítek - time stamp viz stará přednáška 5 nebo link:

<http://www.inf.fu-berlin.de/lehre/WS03/DBSII/unterlagen/dbsII-03-17-DBCC1-6.pdf>

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