

DBS – BI-SPOL-10

Transakce a jejich vlastnosti - ACID.

Obsah

1	Požadavky na konzistenci databáze	2
2	Transakce	2
3	ACID vlastnosti transakce	2
4	Rozvrhy	3

1 Požadavky na konzistenci databáze

V rámci DBMS (Database Management Service) je potřeba myslet na konzistenci dat a jejich ochranu. Je potřeba myslet na to, že žádná akce by neměla ohrozit integritu celého systému.

- Dva základní požadavky na DBMS:
 - chránit data – ve smyslu odolnosti vůči různým haváriím serveru
 - poskytnout korektní, rychlý a asynchronní přístup většímu množství současně pracujících uživatelů.
- Řešení
 - komponenta řízení souběžného (paralelního) zpracování (concurrency control)
 - komponenta zotavení z chyb (recovery)

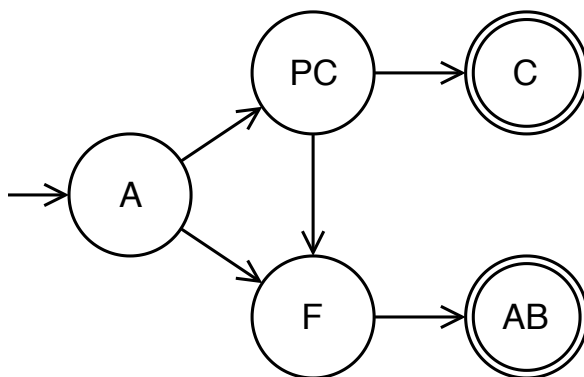
2 Transakce

Vhodná programová jednotka a vhodné mechanismy, které zabezpečí, že po skončení akce (korektním i nekorektním) zůstane databáze konzistentní (platí všechna IO definovaná ve schématu).

- COMMIT - potvrzení
- ROLLBACK - zrušení

Stavový diagram transakce

- aktivní (Active) - od začátku (probíhají DML příkazy)
- částečně potvrzený (Partially Committed) - po provedení poslední operace transakce
- potvrzený (Committed) - po úspěšném zakončení, tj. po potvrzení operace COMMIT
- chybný (Failed) - v normálním průběhu transakce nelze pokračovat
- zrušený (ABorted) - po skončení operace ROLLBACK, tj. uvedení databáze do stavu před započtím transakce



Obrázek 1: Stavový diagram transakce

3 ACID vlastnosti transakce

- atomicita (Atomicity) - transakce musí buď proběhnout celá, nebo vůbec

- konzistence (Consistency) - transformuje databázi z konzistentního stavu do jiného konzistentního stavu
- nezávislost (Independence) - dílčí efekty jedné transakce nejsou viditelné jiným transakcím
- trvanlivost (Durability) - efekty úspěšné transakce jsou trvale uloženy

Žurnál obsahuje sekvenci změnových vektorů $\langle \text{XID}, \text{pageID}, \text{offset}, \text{length}, \text{old data}, \text{new data} \rangle$ Žurnál a přidružená infrastruktura umožňuje implementaci Atomicity a Durability u transakčního zpracování. Informace z transakčního žurnálu se používají pouze pro obnovu databáze po chybě.

4 Rozvrhy

Stanovení pořadí provádění dílčích akcí více transakcí v čase nazveme **rozvrhem**. Rozvrh je korektní, když je v nějakém smyslu ekvivalentní kterémukoliv sériovému rozvrhu.

Rozvrh je uspořádatelný (korektní) pokud nemá precendenční graf kružnici. Rozvrhy jsou ekvivalentní mají-li stejný precendenční graf.

Precendenční graf rozvrhu:

- uzly = jednotlivé transakce rozvrhu
- hrany (orientované)
 - jedna transakce READ(A) před tím než druhá transakce WRITE(A)
 - jedna transakce WRITE(A) před tím než druhá transakce READ(A)
 - poslední WRITE(A) v jedné transakci je před posledním WRITE(A) v druhé transakci.

Paralelní zpracování transakcí:

- Testování uspořádatelnosti
- Uzamykání (LOCK TABLE)

Dvoufázová transakce:

- 1. fáze - uzamyká se, nic neodemyká
- 2. fáze - od prvního odemknutí do konce se už nic nezamyká

Dobře formované transakce:

- transakce zamyká objekt, chce-li k němu přistupovat
- transakce nezamyká objekt, který již zamkla
- transakce neodmyká objekt, který nezamkla
- na konci transakce nezůstane žádný objekt zamčený

Jestliže všechny transakce v dané množině transakcí T jsou:

- dobře formované
- dvoufázové

⇒ pak každý jejich legální rozvrh je uspořádatelný.