

15. Възстановяване

Лекционен курс "Бази от данни"

Основни понятия

Възстановяване - възстановяване на БД след грешка в предишно състояние, за което се знае, че е коректно.

Основен принцип – възстановяването се извършва на основата на излишество на информация.

Възстановяване на системата

Системата трябва да бъде подготвена не само за възстановяване при локални, но и при глобални пропадания:

- Локални – засягат само транзакцията, в която са възникнали;
- Глобални – засягат всички активни в момента транзакции.

Глобалните пропадания се разделят на две категории:

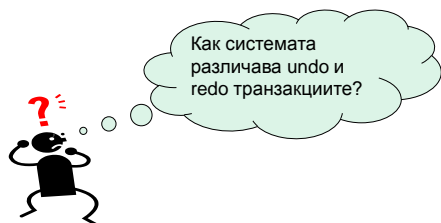
- Системни:
 - Прекъсване на напрежението;
 - Засягат всички активни транзакции – не причиняват физически повреди на БД;
 - Наричат се soft crash;
- Медийни:
 - Повреди върху дисковите устройства;
 - Наричат се hard crash.

При системните откази се губи съдържанието на оперативната памет (буферите на БД).

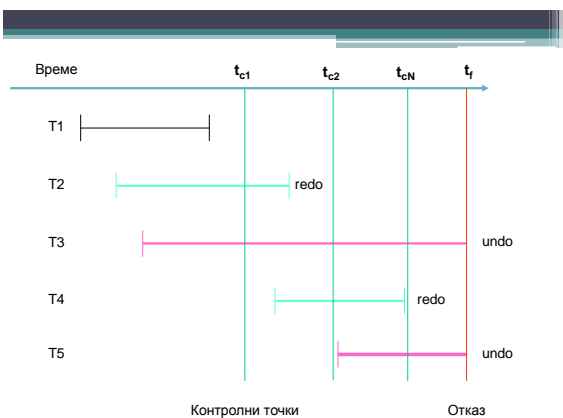
При рестарт системата стартира с два списъка на транзакции – undo и redo.

- Кандидати за undo:
 - Не е ясно състоянието им в момента на пропадането;
 - Такива транзакции не могат да бъдат завършени успешно;
 - При рестарт на системата трябва да бъдат отменени.

- Кандидати за redo:
 - В момента на пропадането транзакциите са завършили успешно;
 - Резултатите се намират в буферите, но не са направени постоянни – т.е. не са записани върху дисковите устройства;
 - При рестарт - системата ги изпълнява автоматично.



- Системата води протокол;
- През определени интервали записва контролни точки (checkpoints);
- Една контролна точка включва:
 - Записи на съдържанието на буферите (force-writing) във вторичната памет;
 - Запис на списък на активните транзакции по време на checkpoint.



Възстановяване на медията

- Този вид възстановяване обикновено включва възстановяване на БД от архив;
- Задължително водене на архивни копия на БД.

Двуфазово предаване за съхранение (two-phase commit)

Протокол за реализиране на commit/rollback концепцията:

- Съществен – транзакциите могат да взаимодействат с различни независими един от друг *resource managers* (RM), напр. – за две различни СУБД (ORACLE, DB2);
- Всеки отделен RM има:
 - Собствено множество от управляеми информационни ресурси;
 - Поддържа собствени протоколи за възстановяване.

Нека имаме транзакция, която променя данни в две СУБД:

- Ако транзакцията завърши успешно – тогава всички промени за ORACLE и DB2 трябва да бъдат потвърдени;
- Ако транзакцията завърши неуспешно – тогава всички промени за ORACLE и DB2 трябва да бъдат отменени;
- Т.е. не трябва да е възможно промените за ORACLE да бъдат потвърдени, а за DB2 – отменени, и обратно – тогава транзакцията вече няма да е атомична.

Не е целесъобразно транзакциите да се натоварват с излишни операции за уведомяване на отделните RMs и гарантиране, че update операциите в отделните БД са завършили успешно.

За тази цел според двуфазовия протокол:

- Транзакциите разпространяват единични COMMIT/ROLLBACK в цялата система;
- Те се прехващат от друг системен компонент, наречен COORDINATOR.

Нека приемем, че транзакцията е обработила данните успешно, така че операцията, която следва, е COMMIT.


Координаторът работи по следния начин:

- Инструктира всеки участник в процеса (RM) да запише направените промени локално;
- Така, каквото и да се случи по-нататък, той ще разполага със списък на активните транзакции и ще може да изпълни COMMIT или ROLLBACK;
- Ако записът е успешен, RM връща отговор на Координатора за успех (OK), иначе отговор за неуспех (NOK);

- Когато Координаторът е получил отговори от всички участници в транзакцията, записва решението си за транзакцията в собствения си журнал:
 - Ако всички са отговорили с OK, решението е COMMIT;
 - Ако някой е отговорил NOK, решението е ROLLBACK.

След това Координаторът информира всеки участник за решението си и всеки участник тогава трябва да изпълни COMMIT или ROLLBACK спрямо указанията.

Тук ако се получи срив, при рестарт на системата ще се гледа решението на Координатора в неговия журнал. Ако такова бъде намерено, то процесът може да продължи; ако такова не бъде намерено – приема се ROLLBACK.



Тук трябва да подчертаем, че ако участниците са в разпределена среда, то срыв от страна на Координатора може да задържи останалите участници да чакат решението му.

Докато трае изчакването всички промени, направени от транзакцията, трябва да бъдат невидими за другите транзакции.
