Recuperação de falhas

Última atualização a *16 de maio de 2020*

Como qualquer sistema informático, os SGBD devem estar preparados para responder a falhas, tendo neste caso como objetivo a recuperação do estado da BD mais próximo do momento em que a falha ocorreu.

Uma falha já abordada é a falha de uma transação, para a qual a solução é o **escalonamento**.

No entanto, há casos mais graves em que está envolvida a perda parcial ou total da base de dados, aos quais apenas os **backups** e **logs de transações** podem responder.

Escalonamento

Apesar de parecer simples quando falamos de transações, quando é mal planeado pode levar a falhas irrecuperáveis.

Um escalonamento diz-se **recuperável** quando nenhuma transação for *commited* até que todas as outras que escrevem elementos lidos por ela tenham sido concluídas.

> Apesar das leituras e as escritas estarem desencontradas, se uma leitura for feita depois da escrita de uma transação que ainda não foi concluída, se a segunda for concluída antes da primeira, esta pode ainda vir a ser cancelada e assim cria-se uma situação **irrecuperãvel** .

Mesmo sendo recuperável, o ideal é que **não tenha *aborts* em cascata**, ou seja, uma transação só pode ler elementos atualizados por transações já concluídas.

> No exemplo anterior, caso a primeira seja abortada, a segunda também o seria, uma vez que tinha lido um elemento escrito pela anterior. O ideal seria a segunda ser executada apenas quando a primeira tivesse sido concluída (*commited*).

Por fim, mesmo que recuperável e sem *aborts* em cascata, é importante que o escalonamento seja **estrito**, ou seja, uma transação só pode ler ou atualizar elementos atualizados por transações já concluídas.

Backups

Consistem em **cópias de segurança** efetuadas regularmente sobre os dados da BD, criando pontos de recuperação para cenários de erros graves no sistema.

Deve ser assumido um **comprimisso** entre a regularidade dos *backups* e a necessidade de manter cópias atualizadas, pois são operações exigentes ao nível dos recursos, mas se forem realizadas demasiado espaçadas no tempo não serão representações fiáveis dos dados.

Transaction logs

Um sistema de logs **regista todas as operações** (incluindo *commits*) realizadas sobre a BD, guardando uma imagem dos dados alterados antes e depois de uma transação, de forma sequencial.

São armazenados de forma repartida entre a memória e o disco.

> São utilizados para fazer...
>
> *Rollback*, que consiste em recuperar o estado antes de uma dada operação ter sido executada *Rollforward*, que consiste em reconstituir as operações realizadas depois de um determinado momento (normalmente para reconstituir transações feitas após um *backup*)

Fluxo das operações

1. As operações são armazenadas no log em memória;
2. Quando é dado o *commit* de uma transação, os dados do log são atualizados em disco;
3. Os dados da BD são escritos em disco.

Recuperação de falhas

Existem várias falhas possíveis sobre um SGBD. É importante ter em conta que todas elas têm **custos** associados, pois obrigam a um maior número de acessos ao disco para atualizar os ficheiros de recuperação, que também vão ocupar espaço adicional, havendo ainda uma sobrecarga do CPU para realizar as operações de manutenção dos dados.

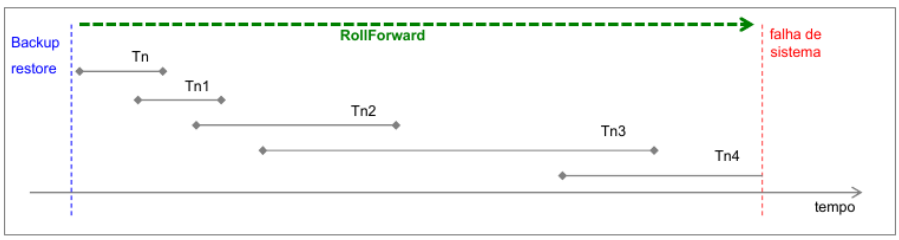
Falha de transação

Uma falha mais simples, bastando fazer *rollback* com base no log de registos.

Falha de disco

Esta é a falha mais grave, sendo necessário reconstruir toda a BD.

A solução passa pelo *restore* do último *backup*, sobre o qual é feito o *rollforward*, que consiste em analisar o log de registos para atualizar a BD desde o momento do *backup* até ao log mais recente.



Apenas Tn4 não é recuperada.

Falha no sistema

Caso hajam falhas no SO ou no SGBD, considera-se que a BD está corrompida e que é necessário regressar ao último estado de integridade conhecido, com recurso ao *rollback*. No entanto, esta tarefa pode ser complicada, pois é difícil detetar o ponto até ao qual devemos desfazer as transações.

Rollback

O ponto até ao qual devemos fazer *rollback* é o ponto em que o log de registos e a BD estão sincronizados.

Podemos optar por fazê-lo através do restauro do último *backup*, fazendo *rollforward* até ao ponto de sincronismo, no entanto esta abordagem é exigente do ponto de vista dos recursos.

A alternativa é criar **check points**: marcas no log que identificam o momento em que os buffers são escritos em disco. Assim, a amplitude dos processos de *rollback* e *forward* é reduzida.