Integrantes: Andrew Lucas Guedes de Souza, Felipe Borges de Souza Chaves e João Pedro Mota Jardim

O QUE SÃO E O QUE REPRESENTAM AS EXPRESSÕES (definições/conceitos) E O QUE ELES REPRESENTAM NA RECUPERAÇÃO DE BANCO DE DADOS?

- LOG

Log é um arquivo gravado em disco que contém detalhes sobre o estado do sistema e quais forams os estados anteriores dele. o **Log** somente é afetado por erros de discos ou falhas no SO mas está livre de falhas no SGBD. Geralmente é periodicamente limpo ou transferido para um sistema de armazenamento como as fitas magnéticas

O *Log* também é utilizado para garantir a durabilidade do banco. Caso o sistema seja interrompido (desligado), ele pode ser utilizado para recuperar o estado do sistema no momento da interrupção.

ATUALIZAÇÃO POSTERGADA

A técnica de recuperação baseada na atualização postergada adia qualquer atualização no banco de dados até que a transação complete sua execução com êxito e atinja seu ponto de confirmação (abordagem no-steal).

Durante a execução da transação, as atualizações são registradas somente no log e nos buffers do cache. Se uma transação falha antes de atingir seu ponto de confirmação, não há necessidade de desfazer nenhuma operação, porque a transação não afetou, de modo algum, o banco de dados armazenado em disco.

Depois que a transação atinge seu ponto de confirmação e o log é forçosamente gravado no disco, as atualizações são registradas no banco de dados (registro adiantado em log). Na prática, é inviável devido ao tamanho do cache necessário para manipular grandes transações.

A técnica é conhecida como algoritmo de recuperação NOUNDO/REDO (NÃO-DESFAZER/REFAZER).

Não é necessário desfazer porque o banco de dados não é atualizado antes do commit. Pode ser necessário refazer se o sistema falhar depois do commit, porém antes que todas as suas alterações tenham sido registradas no banco de dados. As operações da transação são refeitas a partir das entradas do log.

Procedimento para recuperação no momento de uma falha:

- Todas as operações de escrita (write_item) das transações confirmadas devem ser refeitas, a
 partir do log, na ordem em que foram gravadas, utilizando o procedimento REDO;
- As transações que estavam ativas e não chegaram ao commit são efetivamente canceladas e devem ser resubmetidas;
- Procedimento REDO: Examinar as entradas de log do tipo [write_item, T, X, novo_valor] e ajustar o valor do item X no banco de dados para novo valor;

A operação REDO pode se tornar mais eficiente. Quando um item de dado X for atualizado mais de uma vez por transações confirmadas desde o último checkpoint, só é necessário refazer a última atualização de X a partir do log (as outras atualizações seriam sobrescritas pelo último REDO).

Nesse caso, o processo inicia-se a partir do final do log.

Sempre que um item for refeito, ele é acrescentado a uma lista de itens refeitos. Antes de refazer a operação aplicada a um item, a lista é verificada: se o item aparecer na lista, não é novamente refeito, uma vez que seu último valor já foi recuperado

ATUALIZAÇÃO IMEDIATA

A técnica de recuperação baseada em atualização imediata faz com que o banco de dados seja ou não "imediatamente" atualizado, quando uma transação emite um comando de atualização, sem ter que esperar que a mesma atinja seu ponto de confirmação. O processo de registro adiado em log (atualizar o log após a atualização do banco) deve ser usado para permitir recuperação em caso de falha.

Existem duas categorias de algoritmos, a saber:

Algoritmo de recuperação UNDO/NO-REDO

 Não há necessidade de refazer operações se a técnica de recuperação garantir que todas as atualizações de uma transação sejam registradas no banco de dados em disco antes do commit da transação.

Algoritmo de recuperação UNDO/REDO

• Pode ser necessário desfazer e refazer operações se a técnica de recuperação permitir que a transação chegue ao seu ponto de confirmação antes que todas as alterações sejam gravadas no banco de dados.

O sistema deve manter duas listas de transações: uma lista de transações confirmadas desde o último checkpoint (lista de commit) e uma lista de transações ativas (lista ativa).

Procedimento para recuperação no momento de uma falha:

- Todas as operações de escrita (write_item) das transações ativas devem ser desfeitas, a partir do log na ordem inversa em que foram gravadas, utilizando o procedimento UNDO.
- Todas as operações de escrita (write_item) das transações confirmadas devem ser refeitas, a partir do log na ordem em que foram gravadas, utilizando o procedimento REDO.

Procedimento UNDO:

• Examinar as entradas de log do tipo [write_item, T, X, valor_antigo, novo_valor] e ajustar o valor do item X no banco de dados para valor antigo.

Procedimento REDO:

• Examinar as entradas de log do tipo [write_item, T, X, valor_antigo, novo_valor] e ajustar o valor do item X no banco de dados para novo valor.

- CHECKPOINT

O processo de *CHECKPOINT* verifica quais as transações comitadas já foram armazenadas com sucesso no arquivo de *Transaction Log* e pede para que o *SQL* Server armazene as páginas de dados destas transações, que estão em memória, no arquivo de dados. Esse processo é necessário pois, se a transação fosse realizada diretamente no arquivo de dados, caso ocorra algo como o uma queda de energia durante um processo de update, pode ocorrer do banco de dados ficar corrompido ou inconsistente, pois apenas parte do update seria realizado. Para garantir que isso não aconteça, o *SQL* escreve primeiro no arquivo de *Transaction Log* e o processo de *CHECKPOINT* acontece em determinados intervalos de tempo levando essas alterações para o banco.

O **SQL** Server realiza o **CHECKPOINT** nos seguintes casos:

- Ao ser executado o comando de T-SQL *CHECKPOINT*;
- Quando o arquivo de Transaction Log estiver com mais de 70% de seu espaço preenchido e o banco de dados estiver com o Recovery Model Simple;
- De acordo com o valor (em minutos) definido na propriedade Recovery Interval da instância. O padrão dessa propriedade é que um *CHECKPOINT* ocorra a cada 1 minuto;
- Se o SQL Server for desligado através do comando SHUTDOWN, sem a cláusula NOWAIT;