Recuperação de Falhas em SGBDD

Aluno: Antônio Ezequiel de Mendonça
daem@cin.ufpe.br
Orientadora: Ana Carolina Brandão Salgado
acs@cin.ufpe.br
Centro de Informática (CIn)
Pós-Graduação em Ciência da Computação
Universidade Federal de Pernambuco (UFPE)

Recuperabilidade

- A recuperação de transações que falharam significa que o BD será restaurado para o estado de consistência mais recente, exatamente como antes do início da transação que estava executando quando ocorreu da falha.

Recuperabilidade

Duas situações podem ser consideradas:

1) **Falha catastrófica** → Restaura uma cópia anterior do Banco de Dados e reconstrói um estado mais atual de acordo com as ultimas entradas de LOG armazenadas;

2) **Falha não-catastrófica** → a estratégica é reverter quaisquer mudanças que causaram inconsistência desfazendo algumas operações.

Recuperabilidade

Conceitos de Recuperação

- O buffering de páginas de disco (blocos) do banco de dados no cache de memória principal do SGBD.
- O caching de dos blocos sobre o controle do SGBD, independente do sistema operacional.

Recuperabilidade

Conceitos de Recuperação

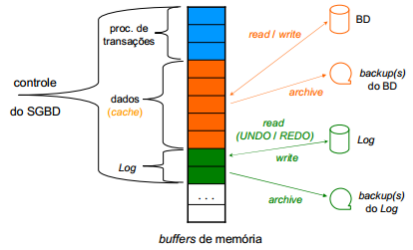
- Um diretório de cache é usado para manter os itens que estão nos buffers. Uma tabela com as entradas: <nome do item, localização do buffer >
- O Banco requisita ações para um mesmo item, verifica no diretório, se o item esta na cache. Caso não, então o item deve ser localizado no disco e copiado para dentro do cache, provoca a paginação.
- Substituir o buffer - FIFO (first in, first out), DBMIN.

Recuperabilidade

Conceitos de Recuperação

- Associado com cada item no cache existe um bit sujo (Dirty Bit), que pode ser incluído na entrada do diretório, para indicar se o item foi ou não modificado.
- 0 - não foi modificado; 1 - modificado.
- Ao esvaziar o buffer, os dados com bit igual a 1 são gravados no disco.
- Bit preso-solto, caso a página esteja presa, 1;

Recuperabilidade



Recuperabilidade

Conceitos de Recuperação

- Em geral o item antigo e chamado de **before** image (BFIM), e o novo valor obtido depois da modificação e chamado de **after** image (AFIM).

Recuperabilidade

Conceitos de Recuperação

Duas estratégias são usadas para um item de dado voltar para o disco:

- 1º **In -place updating** – escreve o item de dado no mesmo espaço, ou seja, sobrescreve o valor antigo do item no disco. Gravando no Log.
- 2º **Shadowing** – escreve um novo item em uma diferente localização no disco, múltiplas cópias do item de dado podem ser mantidas. Log não estritamente necessário.

Recuperabilidade

REDO/ UNDO

- Estratégias

  Perda por falha: reconstrução (REDO)

  Backup (Log) ----------> estado consistente mais próximo da falha.

Recuperabilidade

REDO/ UNDO

- Estratégias

  O BD tornou-se inconsistente: reverter mudanças (UNDO)

  Banco de dados inconsistente -------> Banco de dados consistente.

Recuperabilidade

Abordagem Adiantado em LOG

- O mecanismo de recuperação deve garantir que a BFIM do item de dados seja registrada em uma entrada de LOG antes que a BFIM seja sobrescrita pela AFIM.

Recuperabilidade

Abordagem roubado(steal)/não roubado(no-steal)

- Especifica quando uma página do BD poderá ser gravada em disco a partir do cache.
- Se uma página em cache atualizada por uma transação não puder ser gravada antes que a transação se efetive, ela será chamada de abordagem não-roubada, caso contrário chamada roubada. Não-roubada dispensa Undo.
- Gerenciado de cache precisa de um frame buffer, o gerenciador de buffer substitui uma página existente, mas que a transação não foi confirmada.

Recuperabilidade

Abordagem forçado(force)/não-forçado(no-force)

- Se todas as páginas atualizadas por uma transação forem imediatamente escritas no disco quando a transação se efetivar, ela será chamada de abordagem forçada, caso contrário será não-forçada.
- REDO nunca será necessária no forçada.
- Os bancos usam roubada/não-forçada.

Recuperabilidade

Vantagens roubada(steal)/não-forçada(no-force)

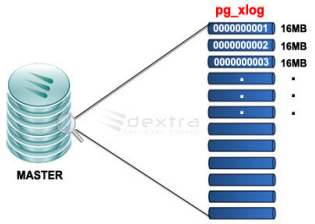
- Roubada →Evita a necessidade de um espaço buffer muito grande, para armazenar todas as páginas.
- Não-forçada →Uma página atualizada de uma transação confirmada ainda pode estar no buffer quando outra transação precisar usar, eliminando os gastos E/S, para páginas muito atualizadas.

Recuperabilidade

Wal (logging write-ahead)

- As entradas no Log precisam ser gravadas permanentemente no disco antes das mudanças serem aplicadas ao banco de dados.
- Fornece a atomicidade e durabilidade.
- Trabalha com UNDO e REDO

Recuperabilidade



No PostgreSQL esses logs são denominados **WAL** (*Write Ahead Logs)* e possuem por padrão 16MB.

Em um intervalo de tempo configurado ou através de comando SQL, as transações que sofreram COMMIT são transferidas do WAL para o arquivo de dados e os logs são reciclados, essa operação é conhecida como ***CHECKPOINT***.

Recuperabilidade

- Para facilitar o processo de recuperação, o DBMS mantém uma lista para cada tipo transação:
  1. ativas → aquelas que tenham começado mas ainda não foram comitadas.
  2. transações já comitadas.
  3. transações abortadas.

Todas com seus respectivos checkpoints.

Recuperabilidade

Checkpoints no LOG de sistema

- Um registro que é escrito periodicamente dentro do LOG. O ponto em que o sistema grava no disco, todos os buffers do SGBD que tiverem sido modificados.
- Transações que tiverem entradas [commit, T] no LOG, antes do [chekpoint], não necessitarão ter suas operações WRITE refeitas, no caso de falha, pois todas as suas atualizações foram registradas em disco durante o checkpoint.

Recuperabilidade

Checkpoints no LOG de sistema

- Como o SGBD deve decidir o momento de submeter um checkpoint?
- **Resp.:** o momento de realização de um checkpoint pode ser decidido por intervalo de tempo (n minutos) ou por número de transações efetivadas desde o último checkpoint.

Recuperabilidade

Checkpoints no LOG de sistema

O que consiste a ação de submissão do checkpoint?

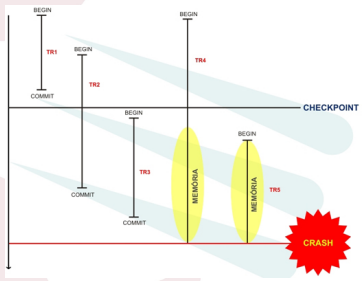
- 1 – Suspender a execução das transações temporariamente;
- 2 – Gravar no disco todos os buffers da memória principal que tenham sido alterados;
- 3 – Escrever um registro de [checkpoint] no LOG e forçar a gravação do LOG no disco;
- 4 – Reassumir a execução das transações.

Recuperabilidade

Checkpoints no Log de sistema

- O passo **2** pode atrasar o processamento da transação por causa do passo **1**.
- Para reduzir este atraso, uma técnica chamada **fuzzy checkpoint** pode ser usada.
- Transações continuem após um registro de checkpoint ser gravado no LOG, sem esperar o passo **2** terminar. Ao termino do passo **2**, a gravação do LOG é feita no disco.

Recuperabilidade

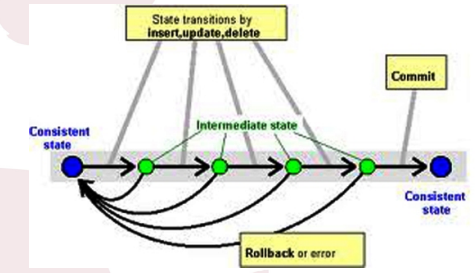


Recuperabilidade

Rollback de Transação

- Caso uma transação falhe durante uma alteração no BD, é realizado um rollback.
- Os itens de dados que tenham sido mudados por uma transação devem ser retornados aos seus valores anteriores a modificação.
- Os registros no LOG do sistema são utilizadas para recuperar o valor antigo.

Recuperabilidade



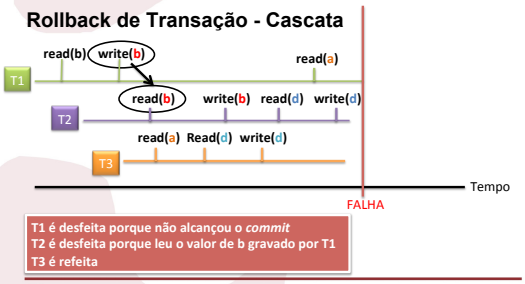
Recuperabilidade

Rollback de Transação

- **Rollback em Cascata**: Se uma transação T é desfeita e uma transação S leu algum dado atualizado por T, S também tem que ser desfeita e assim por diante.
- Bloqueio em duas fases básico.
- Complexo e demorado.
- Normalmente não é utilizado pelos SGBDs.

Recuperabilidade

Rollback de Transação - Cascata



T1 é desfeita porque não alcançou o *commit*
T2 é desfeita porque leu o valor de b gravado por T1
T3 é refeita

Recuperabilidade

Duas técnicas principais são utilizadas:

- **Update adiado**

As atualização no BD só serão feitas quando a transação atinja seu ponto commit.

Antes do ponto commit, todas atualizações das transações são gravadas em uma copia local (buffer) e no LOG, depois gravada no disco.

Caso a transação falhe antes de alcançar seu ponto commit, ela não terá afetado o estado do BD.

Recuperabilidade

Update Adiado

- O BD nunca é atualizado no disco até que a transação seja efetivada, desta forma nunca será necessária qualquer operação UNDO (desfazer).
- Existirão apenas entradas do tipo REDO no Log.
- Utilizamos o algoritmo de recuperação NO-UNDO/ REDO
- Operações Idempotentes

Recuperabilidade

- Por que só utiliza-se o REDO?

Resp.: Refazer (REDO) deve ser usado em situações em que o sistema falhar depois que uma transação for efetivada, sendo que antes que todas as suas mudanças sejam registradas no disco.

Nesse caso, as operações da transação serão refeitas a partir das entradas do LOG.

Recuperabilidade

- **Update imediato**
- O BD é atualizado pelas operações de uma transação antes do seu ponto commit. As operações são gravadas no Log em disco por escrita forçada antes de serem aplicadas ao BD.
- Caso a transação falhe depois da gravação, algumas mudanças no BD devem ser realizadas. Uma coleção de buffers (DBMS cache) é mantida.
- Entradas no Log UNDO(BFIm), utiliza o steal.
- Undo e Redo necessários.

Recuperabilidade

Shadow (Sombra)

- A paginação Shadow considera que o BD e composto por um número de páginas de tamanho fixo (ou bloco de discos) para processo de recuperação.
- Um catálogo com n entradas é construído no qual a i-esima entrada aponta para a i-esima pagina do BD em disco. Se não for muito grande o catálogo será mantido na memória principal.

Recuperabilidade

Shadow (Sombra)

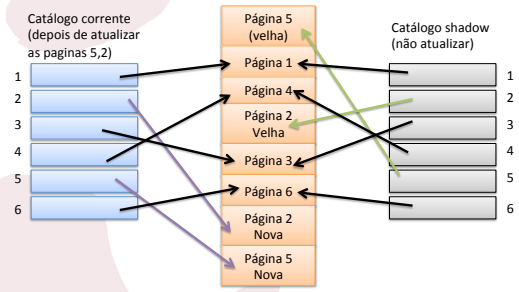
- Transação se inicia, o catálogo corrente cujas entradas apontam para os mais recentes ou correntes páginas em disco é copiado em um shadow (sombra), o qual é salvo no disco, enquanto o catálogo corrente é usado pela transação.
- Durante a execução da transação, o catálogo shadow nunca e modificado.

Recuperabilidade

Shadow (Sombra)

- A operação escrever_item for executada, uma nova cópia da página modificada do BD será escrita, mas a cópia antiga dessa página não será sobrescrita. Para recuperar basta livrar-se das páginas modificadas e descartar o catálogo corrente.
- Desvantagem é que as páginas em atualização mudam de localização no disco, tornando difícil manter paginas juntas.
- Em caso de diretório grande, temos overhead significativo.

Recuperabilidade



Recuperabilidade

- **O método de recuperação ARIES**
- Faz uso das abordagens roubada(steal)/não-forçada(no-force) para gravação e é baseado em três conceitos:
- 1) registro adiantado em Log;
- 2) repetição de histórico durante o refazer;
- 3) mudanças do Log durante o desfazer.
- Usado pela IBM.

Recuperabilidade

Recuperação em Falhas Catastróficas

- O Banco de dados será restaurado para o estado de consistência mais recente, exatamente como antes do momento da falha.
- As técnicas vistas se aplicam a falhas não catastróficas.
- Para manipular falhas catastróficas, como por exemplo quebras de disco, a principal técnica usada nestes casos e a de **backup do banco de dados.**

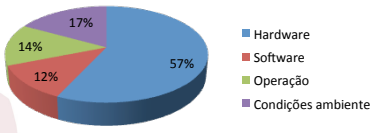
Recuperabilidade

Recuperação em Falhas Catastróficas

- Todo o banco de dados e seu Log são periodicamente copiados para um meio de armazenamento relativamente barato, como por exemplo, fitas magnéticas.

Recuperabilidade

Falhas em sistemas de Banco de Dados



Fonte: IBM, universidade de Stanford.

Recup. Distribuída

Características de Sistemas confiáveis

O que é prevenção de falhas?

É o processo que estabelece os mecanismos para evitar a ocorrência.

O que é detecção de falhas?

É a capacidade de detectar erros.

O que é Latência de erro?

É a diferença de tempo entre o início de um evento e o momento em que seus efeitos tornam-se perceptíveis.

Recup. Distribuída

Características de Sistemas confiáveis

O que é prevenção de falhas?

É o processo que estabelece os mecanismos para evitar a ocorrência.

O que é detecção de falhas?

É a capacidade de detectar erros.

O que é Latência de erro?

É a diferença de tempo entre o início de um evento e o momento em que seus efeitos tornam-se perceptíveis.

Recup. Distribuída

Visão Geral

- Processo bastante complicado.
- Difícil determinar se um site está Parado.
- Site X envia uma mensagem para site Y, e não obtém resposta de Y, possíveis causas?
- A mensagem de X não foi entregue a Y (falha de comunicação).
- Site Y Parado.
- Y está rodando, mas a resposta não chegou a X.

Recup. Distribuída

Visão Geral

- Para manter a atomicidade de uma transação é preciso uma mecanismo de recuperação em dois níveis. Gerenciamento de recuperação global ou coordenador e os gerenciadores de recuperação local (Logs).
- Coordenador segue o protocolo de confirmação em duas fases.

Recup. Distribuída

Visão Geral

- Mensagens adicionais são necessárias.
- Problema da confirmação distribuída. Atualização não pode ser confirmada, se ela altera dados em vários sites, até ter certeza que as mudanças em cada site não será perdida.
- Os dados do Log da alteração local em cada site precisa ter sido feita no disco.

Recup. Distribuída

Tipos de Falha

- Falhas de transações
- Falhas de sites
- Falhas de mídia
- Falhas de comunicação

Recup. Distribuída

Transação distribuída de Dados

- Fases do protocolo de confirmação:

1. Cada site participante sinalizam ao coordenador que a transação local foi concluída.
2. O coordenador envia uma mensagem de preparação para confirmação.
3. Cada site que receber a mensagem forçará a gravação do Log e as informações de recuperação no Log.
4. Os sites enviarão um sinal de Pronto para confirmação ou ok ao coordenador.

Recup. Distribuída

Transação distribuída de Dados

- Fases do protocolo de confirmação (continuação):

5. Se a gravação forçada em disco falhar, o participante (site) enviará um sinal de não estou pronto (não ok).
6. Se o coordenador não receber uma mensagem em um certo tempo, assume não ok.
7. Se todos participantes responderem ok e o voto do coordenador for ok, então a transação será bem sucedida, e será enviado uma mensagem confirmação aos sites.

Recup. Distribuída

Transação distribuída de Dados

- Fases do protocolo de confirmação(C2F):

8. Com a gravação das informações locais no Log foi bem sucedida, então a recuperação poderá ser efetuada.
9. Se o coordenador ou algum dos participantes sinalizar como não ok, o coordenador enviará um mensagem de reverter (UNDO) - Global-abort a cada participante.
10. Utilizando as informações do Log, o UNDO é realizado.
11. Abort unilateral – participante tem permissão de abortar.

Recup. Distribuída

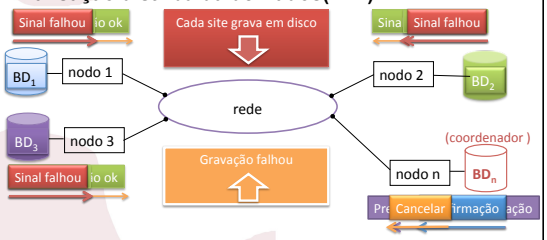
Transação distribuída de Dados

- Fases do protocolo de confirmação(C2F):

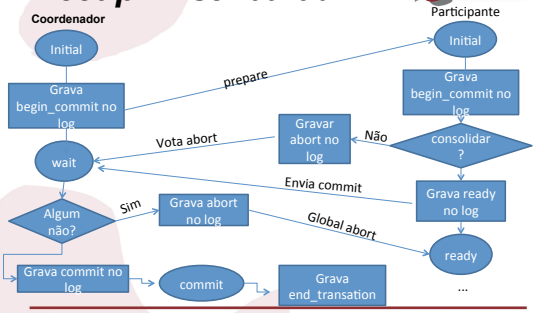
12. O participante não pode mudar seu voto, após ter efetuado.
13. São usados *Timers* para controlar o tempo de espera entre os participantes e o coordenador.

Recup. Distribuída

Transação distribuída de Dados(C2F)



Recup. Distribuída



Recup. Distribuída

Variação do C2F

- Duas variações foram propostas para melhorar o C2F, com isso:

1. Reduzindo o número de mensagens transmitidas entre o coordenador e os participantes.
2. Reduzir o número de vezes que o log é gravado.

Chamados ação de abortar presumida e consolidação presumida

Recup. Distribuída

Protocolo de término e Recuperação para C2F

- Servem os intervalos de tempo limite para o coordenador e os processos participantes
- O método de tratamento depende do sincronismo das falhas e dos tipos das falhas.
- O coordenador tem tempo limite para commit, wait (espera resposta dos participantes), abort.
- Os participantes tem tempo limite para initial(espera uma mensagem prepara), ready(espera uma decisão global).

Recup. Distribuída

Protocolo de consolidação C3F (três fases)

- Semelhante ao protocolo C2F, com a adição do comando PRECOMMIT.
- O coordenador adiciona um espera ao PRECOMMIT.
- É um protocolo em que todos os estados são síncronos dentro de uma única transação de estado.

Bibliografia

- Ramakrishnan, Raghu, *Sistemas de Gerenciamento de banco de dados*, McGrawHill, São Paulo, 2008.
- Ozsu, M. Tomer, *principles of distributed database systems 3rd edition*, Springer, London, 2011.
- Navate, Shamkant B., Sistemas de Banco de Dados, Pearson, São Paulo, 2010.
- IBM. Disponível em: <http://www.ibm.com>. Acesso em: 20 Abril. 2012, 16:30:00.