Aspectos Operacionais

Recuperação após Falha

Recuperação após Falha (ou Restauração)

- Após uma falha na execução de uma transação, o BD deve ser restaurado para um estado consistente anterior
  - ⇨O Sistema deve manter informações sobre as atualizações do BD em separado (LOG)
- Estratégias:
  - ❶ Perda por falha: reconstrução (REDO)

    Backup ----------> estado consistente mais próximo da falha

Recuperação após Falha

- Estratégias (Cont.)
  - ❷O BD tornou-se inconsistente: reverter mudanças (UNDO)

    BD inconsistente-------> BD consistente
- Duas técnicas de recuperação no caso 2:
- ❶ Atualização adiada: BD atualizado depois do *commit*
  - ⇨antes do *commit*: atualizações gravadas em áreas de trabalho locais
  - ⇨durante o *commit*: atualizações gravadas no log e depois no BD

Recuperação após Falha

Não é necessário Undo e o Redo é necessário em alguns casos: No-Undo/Redo

❷ Atualização imediata: BD atualizado antes do commit

- ⇨ gravação no log antes do BD
- ⇨ recuperação possível

Undo e Redo necessários: Undo/Redo

Recuperação após Falha

Conceitos para recuperação:

- SGBD "Cache": buffer na memória principal sob controle do SGBD
- Diretório da "Cache": controla que itens do BD estão em qual buffer. Quando é requerido um item que não está na "cache", provoca a paginação.
- "Dirty Bit": indica se um item de dado na "cache" foi atualizado (1) ou não (0).
- Imagem antes (ImAn): valor antigo do dado
- Imagem depois (ImDp): valor novo do dado

Recuperação após Falha

Estratégias para liberação de um item de dado modificado da "cache" para o disco:

- Atualização in-place: grava o item no mesmo local do disco, apagando o valor anterior, mantendo apenas uma cópia do dado --> É necessário Log gravado antes do BD
- "Shadowing" (imagem): grava o novo item em uma localização diferente, mantendo pelo menos duas cópias do item de dado --> Não é necessário o Log

Recuperação após Falha

Write-Ahead Logging: na atualização in-place, garante que a ImAn seja gravada no log antes da gravação da ImDp no disco.

Entradas do log:

- Do tipo *Redo*: corresponde a uma operação de gravação (ImDp) --> *write-item*
- Do tipo *Undo*: corresponde ao antigo valor do item (ImAn) --> *read-item*

Recuperação após Falha

Protocolo Write-Ahead Logging para Redo e Undo:

1. A ImAn não pode ser apagada pela sua ImDp no disco, até que todas as entradas do tipo Undo daquela transação (até aquele ponto) tenham sido gravadas no disco.
2. A operação Commit de uma transação não pode ser completada até que todas as entradas do tipo Undo e Redo tenham sido gravadas no disco.

Recuperação após Falha

O SGBD deve manter uma lista de transações processadas com os seguintes estados:

- Ativas: iniciadas mas sem commit
- Acabadas (committed): desde o último checkpoint
- Abortadas desde o último checkpoint

Recuperação após Falha

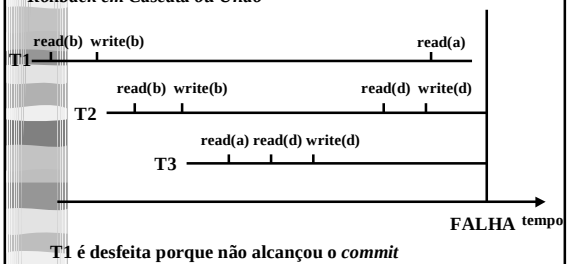
Rollback de Transações (Undo): É necessário se uma falha ocorrer depois do BD ter sido atualizado

- Qualquer item de dado atualizado pela transação deve voltar a seu valor anterior

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.

Recuperação após Falha

Rollback em Cascata ou Undo



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

Recuperação após Falha

Recuperação com Atualização Adiada (RAA)

Ambiente Mono-usuário:

- Protocolo:
  1. Uma transação não pode modificar o BD até seu commit
  2. Uma transação não pode chegar ao commit até que todas as operações de atualização sejam gravadas no log e o log no disco

Recuperação após Falha

- Procedimento:
  - ➪ Usar duas listas de transações: as transações acabadas desde o último checkpoint e as transações ativas
  - ➪ Aplicar a operação Redo para todas as operações write_item das transações acabadas no log, na ordem na qual elas foram gravadas
  - ➪ Recomeçar as transações ativas

Recuperação após Falha

Recuperação com Atualização Adiada (RAA) (cont.)

Redo

- **➪Examinar a entrada no log [write_item, T, X, novo-valor] e colocar o valor do item X no BD como "novo valor" (ImDp)**
- **➪A operação *Redo* é idempotente: Redo(Redo(Redo(x)))=Redo(x)**

T1	T2	Log
read_item(a)	read_item(b)	[start_transaction, T1]
read_item(d)	write_item(b)	[write_item, T1, d, 20]
write_item(d)	read_item(d)	[commit, T1]
	write_item(d)	[start_transaction, T2]
Falha -->		[write_item, T2, b, 10]
		[write_item, T2, d, 25]

A operação *write_item* de T1 é refeita
As entradas no log de T2 são ignoradas

Recuperação após Falha

Recuperação com Atualização Adiada (RAA)

Ambiente Multi-usuário:

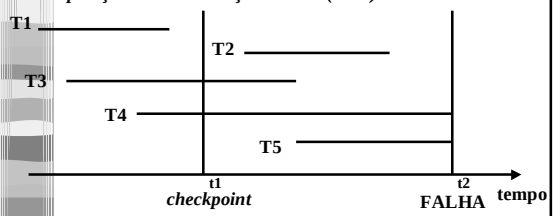
- Protocolo:
  - ❶Depende do protocolo usado no controle de concorrência
  - ❷Assumir protocolo de duas fases com bloqueios antes do início da execução, guardando-os até o commit.

Recuperação após Falha

- Procedimento:
  - ➪ Usar duas listas de transações: as transações acabadas desde o último checkpoint e as transações ativas
  - ➪ Aplicar a operação Redo para todas as operações write_item das transações acabadas no log, na ordem na qual elas foram gravadas
  - ➪ As transações ativas e não acabadas são canceladas e devem ser re-submetidas.

Recuperação após Falha

Recuperação com Atualização Adiada (RAA)



T1: OK

T2 e T3: operações *write_item* devem ser refeitas

T4 e T5: canceladas e re-submetidas

Recuperação após Falha

Recuperação com Atualização Adiada (RAA)

- Desvantagem:
  - ➪Limita a execução concorrente das transações porque itens ficam bloqueados até o commit das transações.
- Vantagens:
  - ➪Uma transação não grava as modificações no BD até o commit. Logo, uma transação nunca é desfeita por causa de falha

Recuperação após Falha

- Vantagens (Cont.)
  - Uma transação nunca vai ler o valor de um item gravado por outra não acabada, porque os itens estão bloqueados. Logo, não ocorrerá rollback em cascata.

Recuperação após Falha

Recuperação com Atualização Imediata (RAI)

Dois tipos de Algoritmos:

- Undo/No-Redo:

  Se a técnica de recuperação garante que todas as atualizações são gravadas no BD (disco) antes do commit da transação, não é necessário Redo
- Undo/Redo:

  Se as modificações são gravada no BD (disco) depois do commit da transação
  - Caso mais geral e mais complexo

Recuperação após Falha

Recuperação com Atualização Imediata (RAI)

Ambiente Mono-usuário:

- Procedimento:
  - Usar duas listas de transações: as transações acabadas desde o último checkpoint e a transação ativa (máximo 1)
  - Aplicar o procedimento Undo: para desfazer todas as operações write_item da transação ativa no log.

Recuperação após Falha

- Procedimento (Cont.) :
  - Aplicar a operação Redo: para todas as operações write_item das transações acabadas na ordem na qual elas foram gravadas no log
- Undo
  - Desfazer uma operação write_item consiste em examinar sua entrada no log: [write_item, T, X, valor_ant, valor_novo] e colocar o valor do item X no BD como "valor_ant" (ImAn)

Recuperação após Falha

- Undo (Cont.)
  - Para desfazer as operações write_item de uma ou mais transações do log, deve-se proceder na ordem inversa de gravação no log.

Recuperação após Falha

Recuperação com Atualização Imediata (RAI)

Ambiente Multi-usuário:

- Protocolo:
  - Assume-se que o log inclui checkpoints e o protocolo de concorrência como o de duas fases

Recuperação após Falha

- Procedimento:
  - ➪ Usar duas listas de transações: as transações acabadas desde o último checkpoint e as transações ativas
  - ➪ Aplicar a operação Undo para todas as operações write_item das transações ativas, na ordem inversa de suas gravações no log
  - ➪ Aplicar a operação Redo para todas as operações write_item das transações acabadas, na ordem na qual elas foram gravadas no log

Recuperação após Falha

Backup e Recuperação após Falhas Catastróficas

(Exemplo: pane no disco)

- Principal Técnica: backup do BD
  - ➪Cópia periódica do BD e do log em outro meio de armazenamento (fitas)
- É necessário backup do log para não perder as transações efetuadas desde o último checkpoint

Recuperação após Falha

- Um log é inicializado após cada operação de backup. Logo, para recuperar após uma falha no disco:
  - ➪ O BD é recriado a partir do último backup
  - ➪ Os efeitos de todas as transações committed, cujas operações foram gravadas no log, são reconstruídos

Recuperação após Falha

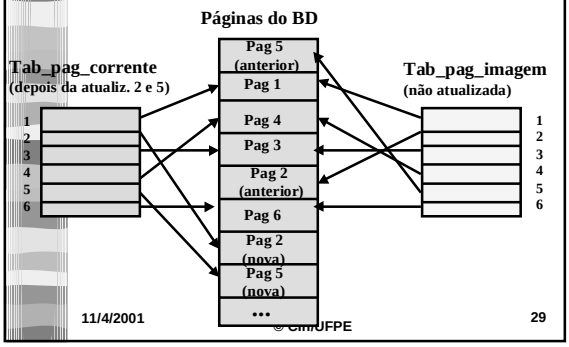
Páginas Imagem (Shadow)

- **Ambiente Mono-usuário: não é necessário o log**
- **Ambiente Multi-usuário: pode-se usar o log se o método de controle de concorrência usar.**
- **Pressupõe-se que o BD seja composto por "n" páginas de tamanho fixo**
- **Uma tabela de páginas com "n" entradas é construída onde pag_tabi = pag_bdi**

Begin_Transaction =>tab_pag_corrente -->tab_pag_imagem

Nunca é modificada durante a transação

Recuperação após Falha



Recuperação após Falha

Páginas Imagem (Shadow)

- Vantagem:
  - ➪ Não há necessidade de Undo ou Redo de operações
- Desvantagens:
  - ➪Páginas atualizadas mudam de localização no disco, impedindo de manter juntas páginas relacionadas

Recuperação após Falha

- Desvantagens (Cont.):
  - Se a tabela de páginas é grande, o tempo para gravar as tabelas de páginas imagem no commit é significativo
  - Garbage Collection (liberação de páginas antigas) é necessário após o commit