

Aspectos Operacionais: Gerenciamento de Transações



Valéria Times

Gerenciamento de Transações

- ◆ Mono-usuário x Multi-usuário
Um dos critérios de classificação de um SGBD: NÚMERO DE USUÁRIOS que podem acessar o sistema concorrentemente
- ◆ Sistema Multi-usuário ↔ Conceito de Multi-programação
↓
- ◆ Permite processar múltiplos programas (ou transações) ao mesmo tempo

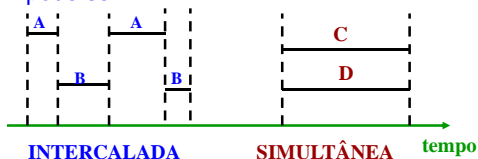
3/5/2012

© CIn/UFPE

2

Gerenciamento de Transações

- ◆ Em um sistema operacional de multi-programação, a execução de um programa pode ser:



A execução de um programa que acessa ou modifica o conteúdo de um BD é chamada TRANSAÇÃO

3/5/2012

© CIn/UFPE

3

Gerenciamento de Transações

- ◆ Quando uma transação é submetida a um SGBD, este deve garantir:
 - Todas as operações na transação são completadas com sucesso e seus efeitos gravados permanentemente no BD
 - A transação não tem efeito sobre o BD ou sobre qualquer outra transação
 - A transação deve manter a base de dados consistente.

3/5/2012

© CIn/UFPE

4

Gerenciamento de Transações

- ◆ Uma transação pode ser interrompida, por algum tipo de falha, depois da execução de algumas operações e antes da execução de todas elas.



Recuperação do Estado Consistente do BD

3/5/2012

© CIn/UFPE

5

Gerenciamento de Transações

OPERAÇÕES:

- ◆ Begin_Transaction: início da execução da transação
- ◆ Read/Write: especifica operações de leitura ou escrita de itens do BD que são executadas como parte da transação
- ◆ End_Transaction: especifica que as operações terminaram e marca o limite final da transação

3/5/2012

© CIn/UFPE

6

Gerenciamento de Transações

- ◆ Commit_Transaction: Sinaliza o final com sucesso de uma transação de modo que qualquer atualização executada pela transação possa ser efetivada no BD
- ◆ Rollback (abort): sinaliza que a transação terminou sem sucesso e o que foi executado deve ser desfeito.

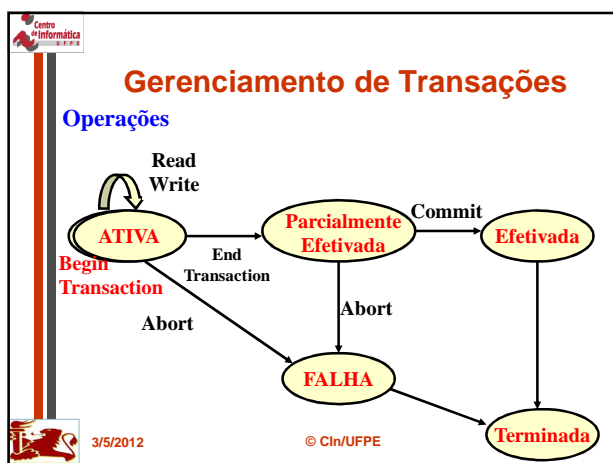
3/5/2012 © Cin/UFPE 7

Gerenciamento de Transações

OPERAÇÕES ADICIONAIS:

- ◆ Undo: similar ao Rollback mas aplicada a uma única operação
- ◆ Redo: especifica que certas operações de uma transação têm que ser refeitas para garantir que todas as operações de uma transação efetivada (committed) foram aplicadas com sucesso ao BD.

3/5/2012 © Cin/UFPE 8



Gerenciamento de Transações

- ◆ O Log do Sistema:
Utilizado para permitir recuperação de falhas de transações. Também chamado de Journal.
 - Registra todas as operações que afetam valores de itens do BD
 - Guardado em disco
 - Periodicamente sofre backup

3/5/2012 © Cin/UFPE 10

Gerenciamento de Transações

- ◆ Tipos de entradas no Log:
 - [start_transaction, T]
 - [write_item, T, X, old-value, new-value]
 - [read_item, T, X]
 - [commit, T]
 - [abort, T]

3/5/2012 © Cin/UFPE 11

Gerenciamento de Transações

- ◆ Commit Point
 - Quando todas as operações que acessam o BD foram executadas com sucesso e o efeito de todas as operações da transação no BD já foram gravadas no log
- ◆ Checkpoints no Log do Sistema
 - Escritos no log periodicamente quando o sistema grava no disco todas as operações WRITE de transações efetivadas

3/5/2012 © Cin/UFPE 12

Gerenciamento de Transações

- ◆ Checkpoints no Log do Sistema (Cont.)
 - O gerenciamento de recuperação do SGBD deve decidir quais os intervalos nos quais devem ocorrer checkpoints
 - em unidade de tempo
 - em número de transações efetivadas após o último checkpoint

3/5/2012 © CIn/UFPE 13

Gerenciamento de Transações

- ◆ Ações de um checkpoint
 - Suspender temporariamente a execução de transações
 - Forçar a escrita de todas as operações de modificação das transações efetivadas do buffer para o disco
 - Escrever um registro [checkpoint] no log e forçar a escrita do log no disco
 - Voltar à execução de transações

3/5/2012 © CIn/UFPE 14

Gerenciamento de Transações

- ◆ Informações adicionais em um registro de checkpoint
 - Lista das identidades das transações ativas
 - Endereços do primeiro e do último registros no log, para cada transação.

3/5/2012 © CIn/UFPE 15

Gerenciamento de Transações

- ◆ Propriedades desejáveis (ACID) em transações
 - Atomicidade: unidade atômica de processamento
 - Consistência: a correta execução deve levar o BD de um estado consistente para outro estado consistente

3/5/2012 © CIn/UFPE 16

Gerenciamento de Transações

- ◆ Propriedades ACID (Cont.)
 - Isolamento: as atualizações não devem ser tornadas visíveis para outras transações até o commit
 - Durabilidade: sempre que o BD for modificado e essas mudanças forem efetivadas, não podem ser perdidas por causa de falhas subsequentes.

3/5/2012 © CIn/UFPE 17

Gerenciamento de Transações

- ◆ Escalonamento ou História de Transações
 - A ordem de execução das operações de várias transações executando concorrentemente na forma intercalada.
- ◆ **Escalonamento e Recuperabilidade**

Um escalonamento S de 'n' transações T1, T2, ..., Tn é um ordenamento das operações dessas transações, sujeito à restrição de que para cada transação Ti em S, as operações de Ti em S devem aparecer na mesma ordem que ocorrem em Ti

3/5/2012 © CIn/UFPE 18

Gerenciamento de Transações

- ◆ Operações Conflitantes em um escalonamento.
Pertencem a diferentes transações, acessam o mesmo item e uma das duas operações é um `write_item`
- ◆ Escalonamento Recuperável
Nenhuma transação T em S é efetivada até que todas as transações T' que escrevam um item a ser lido por T, tenham sido efetivadas

3/5/2012 © CIn/UFPE 19

Gerenciamento de Transações

- ◆ Rollback em Cascata
Uma transação não efetivada tem que ser cancelada porque leu um item gravado por uma transação que falhou
- ◆ Escalonamento que Evita Rollback em Cascata
Toda transação no escalonamento apenas lê itens que foram escritos por transações efetivadas

3/5/2012 © CIn/UFPE 20

Gerenciamento de Transações

- ◆ Escalonamento Estrito
Nenhuma transação pode ler ou escrever itens que a última transação que o escreveu não tenha ainda sido efetivada ou abortada
- ◆ Serializabilidade de Escalonamentos
Se duas transações são submetidas sem intercalação, então:
 - ◆ Todas as operações da transação T1 (em sequência) são seguidas por todas as operações de T2, ou vice-versa

3/5/2012 © CIn/UFPE 21

Gerenciamento de Transações

- ◆ Teoria da Serializabilidade
Tenta determinar quais escalonamentos são "corretos" e quais não são e desenvolver técnicas que permitam apenas escalonamentos corretos
- ◆ **Escalonamento Serial**
As operações de cada transação são executadas consecutivamente, sem operações intercaladas com as de outras transações

3/5/2012 © CIn/UFPE 22

Gerenciamento de Transações

- ◆ Escalonamento Não-serial
As operações de uma transação são executadas intercaladas com operações de outra transação
- ◆ Escalonamento Serializável
É equivalente a algum escalonamento serial com as mesmas 'n' transações

3/5/2012 © CIn/UFPE 23

Gerenciamento de Transações

- ◆ Escalonamento de Resultados Equivalentes
Produzem o mesmo resultado final do BD
- ◆ Conflitos Equivalentes
A ordem de quaisquer duas operações conflitantes é a mesma em ambos os escalonamentos

3/5/2012 © CIn/UFPE 24

Gerenciamento de Transações

- ◆ **Conflito Serializável**
Se for conflito equivalente a algum escalonamento serial.
- ◆ **Teste de Conflito Serializabilidade de um escalonamento**
Grafo de Precedência: grafo dirigido $G=(N,E)$ que consiste de um conjunto de nós $N=\{T_1, T_2, \dots, T_n\}$ e um conjunto de lados dirigidos $E=\{e_1, e_2, \dots, e_n\}$

3/5/2012 © CIn/UFPE 25

Gerenciamento de Transações

◆ **Algoritmo**

1. Para cada transação participando de um escalonamento S , criar um nó rotulado T_i no grafo de Precedência
2. Para cada caso em S onde uma transação T_i executa um `read_item(x)` antes de um `write_item(x)` executado por outra transação T_j
crie um lado $(T_i \rightarrow T_j)$ no grafo de precedência

3/5/2012 © CIn/UFPE 26

Gerenciamento de Transações

3. Para cada caso em S onde uma transação T_i executa um `write_item(x)` antes de um `read_item(x)` executado por outra transação T_j
crie um lado $(T_i \rightarrow T_j)$ no grafo de precedência
4. Para cada caso em S onde uma transação T_i executa um `write_item(x)` antes de um `write_item(x)` executado por outra transação T_j
crie um lado $(T_i \rightarrow T_j)$ no grafo de precedência

3/5/2012 © CIn/UFPE 27

Gerenciamento de Transações

5. Um escalonamento é serializável se e somente se o grafo de precedência não tem ciclos

3/5/2012 © CIn/UFPE 28

Gerenciamento de Transações

Esquema A:

T1	T2
<code>Read_item (x);</code> <code>x := x + n;</code> <code>write_item (x);</code> <code>read_item (y);</code> <code>y := y + n;</code> <code>write_item (y);</code>	<code>Read_item(x);</code> <code>x := x+m;</code> <code>write_item(x);</code>

tempo ↓

SERIALIZÁVEL

3/5/2012 © CIn/UFPE 29

Gerenciamento de Transações

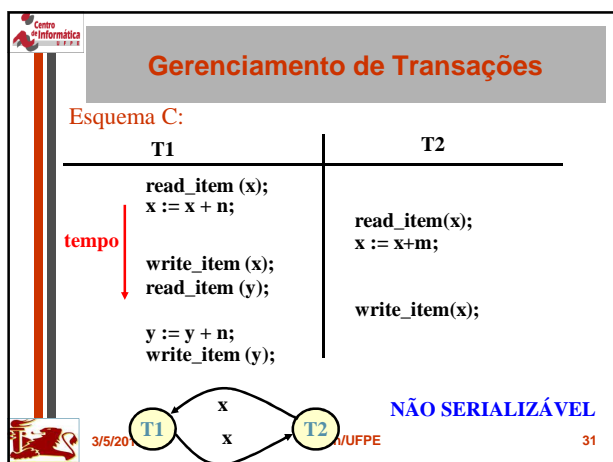
Esquema B:

T1	T2
<code>Read_item (x);</code> <code>x := x + n;</code> <code>write_item (x);</code> <code>read_item (y);</code> <code>y := y + n;</code> <code>write_item (y);</code>	<code>Read_item(x);</code> <code>x := x+m;</code> <code>write_item(x);</code>

tempo ↓

SERIALIZÁVEL

3/5/2012 © CIn/UFPE 30



Gerenciamento de Transações

Na Prática:

- ◆ Fatores como a carga do sistema, tempo de submissão das transações e prioridade dessas transações contribuem na ordenação de operações
- ◆ Praticamente impossível testar a serializabilidade de um escalonamento.

➔ Protocolo

3/5/2012 © CIn/UFPE 32