

Controle de concorrência

Notas de Aula - referentes a textos
do livro do autor:
Silberschatz (veja referencias)

UTFPR

Curso: Engenharia de Computação
Disciplina: Banco de Dados

Protocolos com Base em Bloqueio(Lock)

- Um meio de garantir a serialização é obrigar que os acessos aos itens de dados seja feito de maneira mutuamente exclusiva; isto é enquanto uma transação acessa a um item de dados, nenhuma outra transação pode modifica-lo. O método mais usado para sua implementação é permitir o acesso a um item de dados somente se ele estiver bloqueado.

Protocolos baseados em bloqueio

- ▮ Um bloqueio é um mecanismo para controlar o acesso simultâneo a um item de dados
- ▮ Os itens de dados podem ser bloqueados em dois modos:
 1. Modo *exclusivo* (X). O item de dados pode ser lido e também escrito. O bloqueio X é solicitado pela instrução lock-X.
 2. Modo *compartilhado* (S). O item de dados só pode ser lido. O bloqueio S é solicitado pela instrução lock-S.
- ▮ As solicitações de bloqueio são feitas ao gerenciador de controle de concorrência. A transação só pode prosseguir após a concessão da solicitação.

Protocolos baseados em bloqueio

□ Matriz de compatibilidade de bloqueio


	S	X
S	true	false
X	false	false

- Uma transação pode receber um bloqueio sobre um item se o bloqueio solicitado for compatível com os bloqueios já mantidos sobre o item por outras transações
- Qualquer quantidade de transações pode manter bloqueios compartilhados sobre um item, mas se qualquer transação mantiver um bloqueio exclusivo sobre um item, nenhuma outra pode manter qualquer bloqueio sobre o item.
- Se um bloqueio não puder ser concedido, a transação solicitante deve esperar até que todos os bloqueios incompatíveis mantidos por outras transações tenham sido liberados. O bloqueio é então concedido.

TRANSAÇÃO 1	TEMPO	TRANSAÇÃO 2
UPDATE LIVRO SET preco = 60 WHERE id_livro = 1;	09:00	UPDATE LIVRO SET preco = 80 WHERE id_livro = 2;
1 row updated.		1 row updated.
UPDATE LIVRO SET preco = 55 WHERE id_livro = 2;	09:10	<i>Transação continua normal, utilizando selects, inserts, updates e até mesmo deletes na mesma ou em outras tabelas.</i>
<i>Nenhuma linha atualizada no momento, pois a transação 1 entra em lock conflict com a transação 2.</i>		
Sessão ainda aguardando	09:20	<i>Transação continua normal, utilizando selects, inserts, updates e até mesmo deletes na mesma ou em outras tabelas.</i>
1 row updated. <i>Liberado o lock, a sessão continua normalmente.</i>	09:35	commit;

Protocolos baseados em bloqueio

- Exemplo de uma transação realizando bloqueio:

 T_2 : lock-S(**A**);
read (**A**); 100
unlock(**A**);
lock-S(**B**);
read (**B**); 200
unlock(**B**);

- O bloqueio acima não é suficiente para garantir a serialização - se **A** e **B** fossem atualizados entre a leitura de **A** e **B**, a soma exibida estaria errada.
display(A+B) 300
- Um protocolo de bloqueio é um conjunto de regras seguidas por todas as transações enquanto solicita e libera bloqueios. Os protocolos de bloqueio restringem o conjunto de schedules possíveis.

Sejam A e B duas contas que são acessadas pelas transações T_1 e T_2 . A transação T_1 transfere 50 reais da conta A para a conta B e tem forma: Onde

T1	Passo-a-Passo	T2	Passo-a-Passo
lock-X(B);	Bloqueia exclusivamente(B);	lock-S(A)	Bloqueia de maneira compartilhada (A);
read(B);	Lê(B);	read(A);	Lê(A);
B:=B-50;	Retira R\$ 50,00 de (B)	unlock(A);	Desbloqueia (A)
write(B);	Escreve o valor atual de (B)	lock-S(B);	Bloqueia de maneira compartilhada (B);
unlock(B);	Desbloqueia (B)	read(B);	Lê(B);
lock-X(A);	Bloqueia exclusivamente(A);	unlock(B);	Desbloqueia (B);
read(A);	Lê(A);		
A:=A+50;	Adiciona R\$ 50,00 a (A)		
write(A);	Escreve o valor atual de (A)		
unlock(A);	Desbloqueia (A)		

Se essas duas transações são executadas serialmente, na ordem T_1 , T_2 ou T_2 , T_1 a transação T_2 mostrará o valor de 300 reais.

Armadilhas dos protocolos baseados em bloqueio

- Considere o schedule parcial

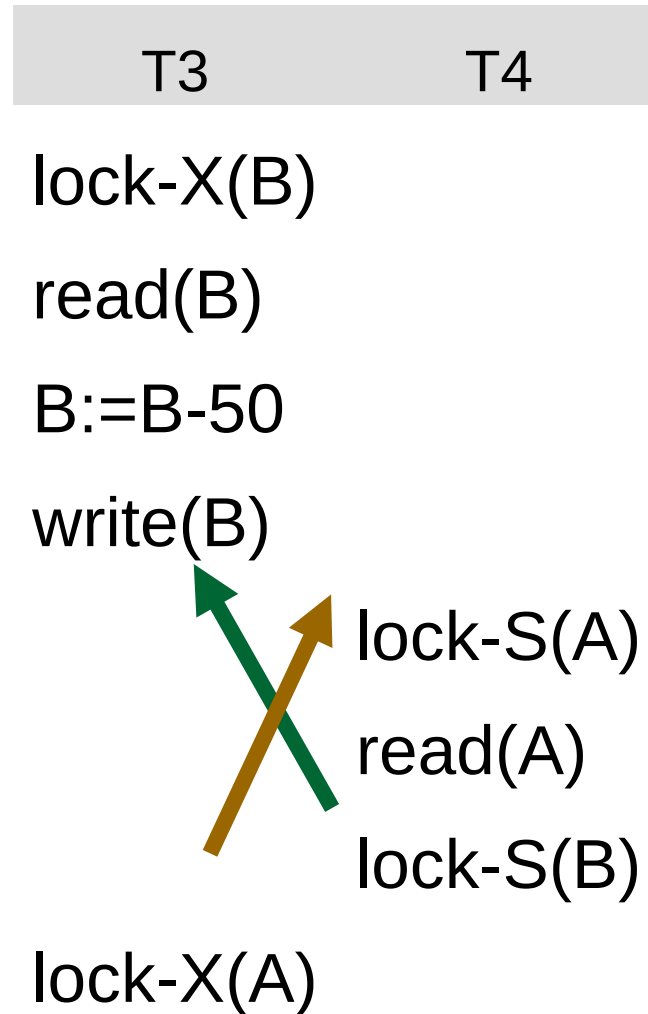
T_3	T_4
lock-X(B) read(B) $B := B - 50$ write(B)	
	lock-S(A) read(A) lock-S(B)
lock-X(A)	

- Nem T_3 nem T_4 podem ter progresso - a execução de lock-S(B) faz com que T_4 espere que T_3 libere seu bloqueio sobre B , enquanto a execução de lock-X(A) faz com que T_3 espere que T_4 libere seu bloqueio sobre A .
- Essa situação é chamada de impasse.
- Para lidar com um impasse, um dentre T_3 ou T_4 precisa ser revertido e seus bloqueios liberados.

Impasse-Deadlock

- Infelizmente o uso do bloqueio pode causar situações indesejáveis. Considere a escala parcial abaixo. Já que T3 mantém um bloqueio exclusivo sobre B, e T4 solicita um bloqueio compartilhado em B, T4 espera que T3 libere B. Analogamente, como T4 mantém um bloqueio compartilhado em A, e T3 está solicitando um bloqueio exclusivo em A, T3 está esperando que T4 libere A. Assim, chegamos a um situação em que nenhuma dessas transações pode processar em sua forma normal. Essa situação é chamada de **DEADLOCK**(impasse). Quando um deadlock ocorre, o sistema precisa desfazer uma das duas transações.

DEADLOCK



O protocolo de bloqueio em duas fases

- ▢ **Esse é um protocolo que garante schedules seriáveis por conflito.**
- ▢ **Fase 1: Fase de crescimento (expansão)**
 - ▢ **transação pode obter bloqueios**
 - ▢ **transação não pode liberar bloqueios**
- ▢ **Fase 2: Fase de encurtamento (encolhimento)**
 - ▢ **transação pode liberar bloqueios**
 - ▢ **transação não pode obter bloqueios**
- ▢ **O protocolo garante a serialização. Pode ser provado que as transações podem ser seriadas na ordem de seus pontos de bloqueio (ou seja, o ponto onde uma transação adquiriu seu bloqueio final).**

Protocolo de bloqueio em Duas Fases

- Inicialmente uma transação está em fase de expansão. A transação adquire os bloqueios de que precisa. Tão logo a transação libera um bloqueio, ela entra em fase de encolhimento e não poderá solicitar novos bloqueios.

Protocolo de bloqueio em Duas Fases

- Por exemplo, as transações T3 e T4 (slide 8) têm duas fases. Por outro lado, as transações T1 e T2 (slide 7) não tem duas fases. Note que as instruções de desbloqueios não precisam aparecer no final da transação.
- Por exemplo, no caso da transação T3, podemos colocar a instrução **unlock(B)** logo após a instrução **lock-X(A)**, e ainda assim manter a propriedade do bloqueio em duas fases.

Gerenciador do Controle de Concorrência

Protocolo
com base em
bloqueio
duas-fases
garante a
serialização.

T1	T2	Gerenciador de Controle de Concorrência
lock-X(A)		grant lock-X(A), T1
read(A)		
write(A)		
	lock-X(A)	blocked (wait), T2
	read(A)	
	write(A)	
lock-X(B)		grant lock-X(B), T1
read(B)		
write(B)		
unlock(A,B)		revoke lock-X(A), T1 revoke lock-X(B), T1
	lock-X(B)	grant lock-X(A), T2 grant lock-X(B), T2
	read(B)	
	write(B)	

Protocolo de bloqueio em 2 fases

T1	T2
Lock-X(Aplic); Read(Aplic); Aplic.Saldo = Aplic.Saldo – 500; Write(Aplic); Lock-X(Conta); Unlock(Aplic); // Inicia 2º fase Read(Conta);	
	Lock-S(Conta);
Conta.Saldo = Conta.Saldo + 500; Write(Conta); Unlock(Conta);	Bloqueada
	Read (Conta); Lock-S(Aplic); Unlock(Conta); //Inicia 2º fase Read(Aplic); Print(Conta.Saldo+Aplic.Saldo); Unlock(Aplic);

Bloqueio em duas fases pode causar deadlock:

T1	T2
Lock-X(Aplic); Read(Aplic); Aplic.Saldo = Aplic.Saldo – 500;	
	Lock-S(Conta); Read(Conta); Lock-S(Aplic);
Write(Aplic); Lock-X(Conta);	Bloqueada
Bloqueada	Bloqueada
Não executa: Unlock (Aplic); Read(Conta); Conta.Saldo = Conta.Saldo + 500; Write(Conta); Unlock(Conta);	Não executa: Unlock(Conta); Read(Aplic); Print(Conta.Saldo + Aplic.Saldo); Unlock(Aplic);

Protocolo de Bloqueio em duas fases severo

- Todos os bloqueios exclusivos devem ser mantidos até o final da transação.
- Evita que outras transacoes leiam dados em transição

Referencias

SILBERSCHATZ, Abraham; KORTH, Henry F;
SUDARSHAM, S. Sistema de Banco de
Dados. 3ª ed. São Paulo, MAKRON Books,
1999.