

21103 - Sistemas de Gestão de Bases de Dados
2013-2014
e-fólio B

Resolução e Critérios de Correção

PARA A RESOLUÇÃO DO E-FÓLIO, ACONSELHA-SE QUE LEIA ATENTAMENTE O SEGUINTE:

- 1) O e-fólio é constituído por 3 perguntas. A cotação global é de 3 valores.
- 2) O e-fólio deve ser entregue num único ficheiro PDF, não zipado, com fundo branco, com perguntas numeradas e sem necessidade de rodar o texto para o ler. Penalização de 1 a 3 valores.
- 3) Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.
- 4) O nome do ficheiro deve seguir a normal “eFolioB” + <nº estudante> + <nome estudante com o máximo de 3 palavras>
- 5) Durante a realização do e-fólio, os estudantes devem concentrar-se na resolução do seu trabalho individual, não sendo permitida a colocação de perguntas ao professor ou entre colegas.
- 6) A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.
- 7) A legibilidade, a objectividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas.

Vetor Cotações

1 2 3 pergunta
10 10 10 décimas

1) Dado o seguinte sequenciamento (“schedule”) que envolve as transações T1 e T2:

T1	T2
read(A) write(A)	read(A) read(B)
read(B) write(B)	

Responda à seguinte questão: o sequenciamento é conflito-serializável (“conflict serializable”)? Justifique detalhadamente a sua resposta.

Resposta:

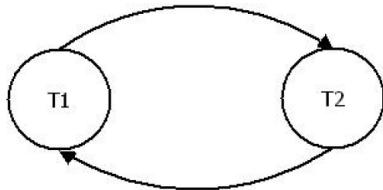
Na construção do grafo de precedências as arestas são montadas a partir das observações das transações que participa, da escala sendo duas transação T_i e T_j haverá uma aresta $T_i \rightarrow T_j$ se forem observadas as seguintes condições:

1. T_i executa $write(A)$ antes de T_j executar $read(A)$
2. T_i executa $read(A)$ antes de T_j executar $write(A)$
3. T_i executa $write(A)$ antes de T_j executar $write(A)$

No caso em questão:

write(A)	read(A)	T1-> T2
read(B)	write(B)	T2->T1

Graficamente obtemos um grafo cíclico:



Conclusão: o sequenciamento que envolve 2 transações **não é conflito-serializável**. Para a sequência (T1 -> T2) temos os conflito $write(A) - read(A)$ e para a sequência (T2 -> T1) encontramos o conflito $write(B) - read(B)$.

Critério de correção:

- (0,5) solução: não é conflito-serializável
- (0,5) justificação detalhada da resposta

2) Para cada um dos seguintes protocolos, descreva aspectos práticos que levam à utilização do protocolo e pelo contrário aspectos que não levam à sua utilização.

Protocolo	A favor	Contra
2-phase locking		
2-phase locking with multiple granularity locking		
The tree protocol		
Timestamp protocol		
Timestamp ordering		
Validation		

Resposta:

Protocolo	A favor	Contra
2-phase locking	- transações de leitura (<i>read only</i>)	- o <i>deadlock</i> deve ser evitado
2-phase locking with multiple granularity locking	- aplicações com mistura de transações leves (que acedem a registos individuais) com transações pesadas (que produzem relatórios)	- o <i>deadlock</i> deve ser evitado
The tree protocol	- acesso aos dados por uma determinada ordem - livre de <i>deadlocks</i>	- ocorrem <i>locks</i> indesejados implicando maiores tempos de espera
Timestamp ordering protocol	- livre de <i>deadlocks</i>	- não adaptado para aplicações com mistura de transações leves com pesadas - risco de <i>roll-back</i>
Validation	- transações de leitura (<i>read only</i>) - ambientes com baixa taxa de conflito, i.e. esquema otimista	- não adaptado para aplicações com mistura de transações leves com pesadas

Critério de correção:

- (0,2) por cada alínea correta (x 5 alíneas)

3) Considere a seguinte sequência do "log" dos algoritmos ARIES, "Algorithms for Recovery and Isolation Exploiting Semantics", para recuperação de uma base de dados. O que é realizado nas fases de Redo() e Undo().

LSN	LOG
00	Begin checkpoint
10	End checkpoint
20	T1 writes P1
30	T2 writes P2
40	T3 writes P3
50	T2 commit
60	T3 writes P2
70	T2 end
80	T1 writes P5
90	T3 abort
	CRASH, RESTART

Resposta:

O "Algorithms for Recovery and Isolation Exploiting Semantics", ARIES, distingue-se pela utilização de um *Log* com "Log Sequence Number" (LSN), uma "Dirty Page Table" (DPT) e uma "Transaction Table" (TT).

A "Dirty Page Table" (DPT) - contém uma entrada para cada página suja no buffer e o LSN correspondente à primeira atualização dessa página.

"Transaction Table" (TT)- contém uma entrada para cada transação ativa, com informações sobre o ID de transação, o LSN do registo de *log* mais recente para a transação e o status da transação.

O ARIES distinguem-se três fases: análise, *redo* e *undo*.

(0) Na fase prévia de análise, são criadas as DPT e TT. O *log* é varrido desde o *checkpoint* até ao ponto de *crash*. Na DPT é registado o primeiro LSN para cada página e a TT o último LSN para cada transação.

Dirty Page Table		Transaction Table		
Page	first LSN	Trans	last LSN	status
P1	20	T1	80	live
P2	30	T2	70	commit, end
P3	40	T3	60	abort
P5	80			

O ARIES também regista os *rollbacks* das transacções através dos "Compensation Log Records" (CLR) i.e. que compensam as alterações de transacções incompletas ou abortadas.

Assim no *log* existem comandos de *write*, *commit*, *end*, *abort*, CLR (Compensation Log Record).

(1) O *Redo* inicia com o menor LSN da DPT, LSN=20. O *log* é varrido desde LSN 20 até ao ponto de *crash*, voltando a aplicar as alterações (*writes* ou CLR) do *log*.

LSN 20 Redo write P1
 LSN 30 Redo write P2
 LSN 40 Redo write P3
 LSN 50 No action
 LSN 60 Redo write P2
 LSN 70 No action
 LSN 80 Redo write P5
 LSN 90 No action

(2) O *Undo* é iniciado no maior LSN da TT, LSN=80, o *log* é varrido do fim para o princípio. A instrução de CLR é acrescentada ao *log* para cada ação de desmontagem.

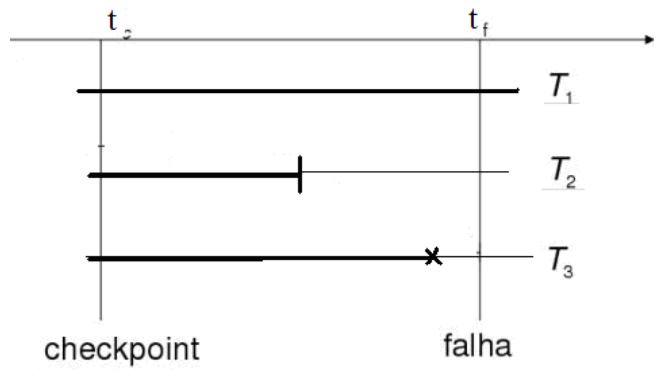
LSN	LOG	prev LSN
00'	begin checkpoint	--
10	end checkpoint	--
20	T1 writes P1	--
30	T2 writes P2	--
40	T3 writes P3	--
50	T2 commit	30
60	T3 writes P2	40
70	T2 end	50
80	T1 writes P5	20
90	T3 abort	60
	crash, restart	
100	CLR T1 LSN 80	80
110	CLR T3 LSN 60	60
120	CLR T3 LSN 40	40
130	T3 end	120
140	CLR T1 LSN 20	20
150	T1 end	140

Em conclusão para cada transação:

T1 fez *rollback (redo, undo)*

T2 fez *commit (redo)*

T3 fez *rollback (redo, undo)*



Critérios de correção:

- (0,5) aplicação detalhada ARIES
- (0,5) comentários conclusivos