

PEP 1 Taller de Bases de Datos
Semestre 2-2021 - Departamento de Ingeniería Informática

Nombre: Benjamin Alberto Jorquera Jorquera (Estudiante 2)
Rut: 19.182.719-8

Integrantes:
Estudiante 1: Nicolas Conejeros P. 20.131.229-9
Estudiante 3: Bryan Salas V. 19.316.410-2

1. Para el esquema de ejecución mostrado en la siguiente figura se pide completar las tablas de WTS y RTS para determinar e indicar cuáles transacciones abortan su ejecución para el protocolo de timestamps y versiones múltiples. También se pide el protocolo de 2 fases.

- (1.a)** TS(T1) = 3; TS(T2) = 4; TS(T3) = 1; TS(T4) = 2; (grupál)
(1.b) TS(T1) = 1; TS(T2) = 4; TS(T3) = 3; TS(T4) = 2; (estudiante 1)
(1.c) TS(T1) = 3; TS(T2) = 2; TS(T3) = 1; TS(T4) = 4; (estudiante 2)
(1.d) TS(T1) = 2; TS(T2) = 1; TS(T3) = 4; TS(T4) = 3; (estudiante 3)

T1	T2	T3	T4
	Read(B)		
Read(A)			
			Read(B)
		Read(C)	
	Write(B)		
Write(A)			
			Read(A)
Read(B)			
	Read(A)		
Write(B)			
		Read(A)	
	Read(C)		
			Write(A)

	Write(A)		
		Read(B)	
Write(C)			
			Write(B)

1.1 Tabla para WTS(Q) y RTS(Q) protocolo de Timestamps

Indicar si es respuesta para 1.a, 1.b, 1.c o 1.d

1.1.a:

W(A)	0						3						4
W(B)	0					4							
W(C)	0												
R(A)	0		3					T4*		4	T3*		
R(B)	0	4		4					T1*				
R(C)	0				1							4	

1.1.c:

W(A)	0						3					4	
W(B)	0					T2*				T1*			4
W(C)	0												
R(A)	0		3					4			T3*		
R(B)	0	2		4					4				
R(C)	0				1								

1.2 Tabla para WTS(Q) y RTS(Q) protocolo de versiones múltiples

Indicar si es respuesta para 1.a, 1.b, 1.c o 1.d

Versión de Q (incluya una nueva fila en la tabla para cada nueva versión de A, B o C)



(debe quedar claro en cual versión (fila) se realiza la escritura o lectura)

1.3 En la siguiente tabla escriba el esquema de ejecución para el protocolo de 2 fases. Indicar si es respuesta para 1.a, 1.b, 1.c o 1.d

1.3.a:

T1	T2	T3	T4
		RLock(A)	
		RLock(B)	
		RLock(C)	
		Read(C)	
		Read(A)	
		Read(B)	
		UnLock(A)	
		UnLock(B)	WLock(A)
		UnLock(C)	WLock(B)
			Read(B)
			Read(A)
			Write(A)
			Write(B)
			UnLock(A)
WLock(A)			UnLock(B)
WLock(B)			
WLock(C)			
Read(A)			
Write(A)			
Read(B)			

Write(B)			
Write(C)			
UnLock(A)			
UnLock(B)	WLock(A)		
UnLock(C)	WLock(B)		
	RLock(C)		
	Write(B)		
	Read(A)		
	Read(C)		
	Write(A)		
	UnLock(A)		
	UnLock(B)		
	UnLock(C)		

1.3.c:

T1	T2	T3	T4
		RLock(A)	
		RLock(B)	
		RLock(C)	
		Read(C)	
		Read(A)	
		Read(B)	
		UnLock(A)	
	WLock(A)	UnLock(B)	
	WLock(B)	UnLock(C)	

	RLock(C)		
	Write(B)		
	Read(A)		
	Read(C)		
	Write(A)		
	UnLock(A)		
WLock(A)	UnLock(B)		
WLock(B)	UnLock(C)		
WLock(C)			
Read(A)			
Write(A)			
Read(B)			
Write(B)			
Write(C)			
UnLock(A)			
UnLock(B)			WLock(A)
UnLock(C)			WLock(B)
			Read(B)
			Read(A)
			Write(A)
			Write(B)
			UnLock(A)
			UnLock(B)

1.4. Considerando las tres respuestas dadas, fundamente las razones por las cuales hay protocolos menos eficientes que otros. Para esto se debe identificar el protocolo que resulta ser el más eficiente, explicar las razones de la mejor eficiencia, y luego se deben fundamentar las razones de la menor eficiencia de los restantes protocolos respecto del primero.

Si se evalúa la concurrencia de una base de datos, el protocolo más eficiente es el de versiones múltiples, debido a que el esquema de ejecución presenta menos transacciones abortadas que el protocolo de timestamps, y el paralelismo es más eficiente que el esquema serial del protocolo de bloqueo de dos fases, por lo tanto, se puede concluir lo siguiente:

- A menor cantidad de transacciones abortadas en un esquema de ejecución, mayor la cantidad de estados consistentes.
- Dos o más transacciones pueden ejecutarse en paralelo en distintos procesadores o máquinas, aumentando la eficiencia del esquema.
- Abortar una transacción es costoso en tiempo y espacio, ya que algunas las operaciones deben re-ejecutarse.
- Un esquema secuencial operando sobre una base de datos con muchos registros demora en ejecutarse, es decir, tienen menor rendimiento.
- La existencia de múltiples versiones o tuplas de los registros puede salvar a una transacción de ser abortada.
- El protocolo de bloqueo de dos fases puede producir deadlocks.

2. Para el protocolo de recuperación de fallas utilizando un log o bitácora de tipo “postergar actualizaciones”, (1) asumiendo que los cambios son reflejados instantáneamente en el log y que por lo mismo las escrituras que provocan re-ejecución alcanzan a ser registradas en el log, y (2) asumiendo que las transacciones que son abortadas vuelven a ser re-ejecutadas en orden secuencial completo considerando el orden en que son abortadas, se pide escribir en la siguiente tabla el contenido del log al finalizar la ejecución de todas las transacciones.

Indicar si es respuesta para el caso 1.a, 1.b, 1.c o 1.d

2.1.a:

Log	
	T3, Start
	T4, Start
	T1, Start
	T2, Start
	T2, Write(B)
	T1, Write(A) *

T2, Write(A)
T2, Commit
T4, Start
T4, Write(A)
T4, Write(B)
T4, Commit
T1, Start
T1, Write(A)
T1, Write(B)
T1, Write(C)
T1, Commit
T3, Start
T3, Commit

2.1.c:

Log

T3, Start
T2, Start
T1, Start
T4, Start
T2, Write(B) *
T1, Write(A)
T1, Write(B) *
T4, Write(A)
T4, Write(B)
T4, Commit
T2, Start

T2, Write(B)
T2, Write(A)
T2, Commit
T1, Start
T1, Write(A)
T1, Write(B)
T1, Write(C)
T1, Commit
T3, Start
T3, Commit