

PEP 1 Taller de Bases de Datos

Semestre 1-2022 - Departamento de Ingeniería Informática

Universidad de Santiago

Para el esquema de ejecución mostrado en la siguiente figura se pide completar las tablas de WTS y RTS para determinar cuáles transacciones abortan su ejecución para el protocolo de timestamps y versiones múltiples. Se debe considerar que cuando una transacción aborta su ejecución, esto puede provocar el rollback de otras transacciones relacionadas.

$TS(T1) = 2$; $TS(T2) = 1$; $TS(T3) = 4$; $TS(T4) = 3$;

T1	T2	T3	T4
	Read(B)		
Read(A)			
Read(B)			
		Write(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. Tabla para WTS(Q) y RTS(Q) protocolo de Timestamps

[illegible]

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

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

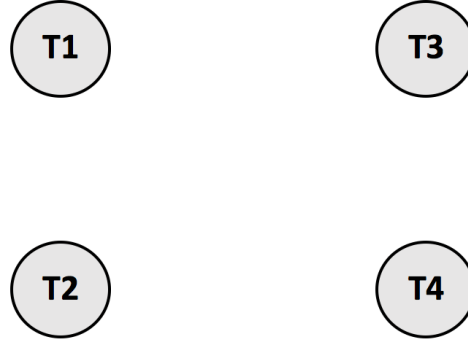
[illegible]

3. 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 bajo el protocolo de timestamps.

TS(T1) = 2; TS(T2) = 1; TS(T3) = 4; TS(T4) = 3;

[illegible]

4. Se pide dibujar el grafo de verificación de serialidad de transacciones “reads antes de writes”



- Si T_j lee el valor de un ítem A , que fue escrito por T_i , entonces se crea un arco $T_i \rightarrow T_j$.
- Si T_i lee el valor de un ítem A y a continuación T_j lo modifica entonces existe un arco $T_i \rightarrow T_j$.

-
- Si T_i intenta ejecutar $\text{read}(Q)$
 - Si $T_s(i) < W_{TS}(Q)$ no se ejecuta la operación y T_i se aborta.
 - Si $T_s(i) > W_{TS}(Q)$ se ejecuta la operación y $R_{TS}(Q) = \max\{R_{TS}(Q), T_s(i)\}$.
 - Si T_i intenta un $\text{write}(Q)$
 - Si $T_s(i) < R_{TS}(Q)$ no se ejecuta la operación y T_i se aborta.
 - Si $T_s(i) < W_{TS}(Q)$ no se ejecuta la operación y T_i continúa.
 - Si no, sí se ejecuta la operación y $W_{TS}(Q) = T_s(i)$.

-
- Si T_i intenta un $\text{read}(Q)$, entonces se retorna el valor de Q_k y $R_{TS}(Q_s) = \max\{R_{TS}(Q_k), T_s(i)\}$.
 - Si T_i intenta un $\text{write}(Q)$:
 - Si $T_s(i) < R_{TS}(Q_k)$, T_i se aborta.
 - Si no, se crea una nueva versión de Q con el valor escrito por T_i , con W_{TS} y R_{TS} iguales a $T_s(i)$.