

Bases de Datos

Protocolos basados en hora de Entrada



Dr. Diego R. Garcia

**DEPARTAMENTO DE CIENCIAS E
INGENIERÍA DE LA COMPUTACIÓN
UNIVERSIDAD NACIONAL DEL SUR**



Protocolo de estampillas de tiempo

- Cada **transacción T_i** se le asigna una **estampilla de tiempo $ts(T_i)$** (hora de inicio de la transacción)
- Las reglas del protocolo garantizan que **la planificación sea equivalente a la serie producida por el orden de las estampillas**: Si $ts(T_i) < ts(T_j)$ el protocolo asegura que la planificación obtenida será equivalente a la serie T_i seguido de T_j . Si no puede garantizar la serie impuesta por las estampillas hará retroceder a alguna de las transacciones (por alguna de reglas del protocolo).
- Cada **dato Q** tiene una **estampilla de lectura $R-ts(Q)$** y una **estampilla de escritura $W-ts(Q)$** , que corresponden a la estampilla de la transacción mas nueva (estampilla mayor) que leyó y escribió, respectivamente, con éxito el dato Q . Cada vez que una transacción **T_i ejecuta con éxito**:
 - **Read(Q)**, entonces a $R-ts(Q)$ se le asigna el valor mayor entre $R-ts(Q)$ y $ts(T_i)$
 - **Write(Q)**, entonces a $W-ts(Q)$ se le asigna el valor mayor entre $W-ts(Q)$ y $ts(T_i)$

Protocolo de estampillas de tiempo

- **Reglas:** cada vez que una transacción T_i intenta ejecutar una instrucción se verifica:
 - Si T_i ejecuta $\text{Read}(Q)$ y $ts(T_i) < W\text{-}ts(Q)$ entonces T_i retrocede (Ti intenta leer un “valor del futuro” producido por una transacción posterior)
 - Si T_i ejecuta un $\text{Write}(Q)$
 - Si $ts(T_i) < R\text{-}ts(Q)$ entonces T_i retrocede (la escritura “llego tarde”, una transacción posterior a T_i ya leyó Q)
 - Si $ts(T_i) < W\text{-}ts(Q)$ entonces T_i retrocede (una transacción posterior “pisó” el valor de Q)En este mismo caso por regla de Thomas la operación $\text{Write}(Q)$ se puede omitir (no se realiza) y T_i continua su ejecución (siempre y cuando no haya retrocedido por otra regla).
 - En cualquier otro caso la instrucción se realiza con éxito y se actualiza la estampilla correspondiente del dato Q según la instrucción:
 - $\text{Read}(Q) \Rightarrow R\text{-}ts(Q) \leftarrow \text{Mayor} (R\text{-}ts(Q), ts(T_i))$
 - $\text{Write}(Q) \Rightarrow W\text{-}ts(Q) \leftarrow \text{Mayor} (W\text{-}ts(Q), ts(T_i))$

Protocolo de estampillas

Estampillas: $0 < ts(T1) < ts(T2) < ts(T3)$

Inicialmente: $W-ts(X) = R-ts(X) = 0$, con $X = A, B$ o C

T1	T2	T3	Dato	R-ts	W-ts
R(A)			A	$ts(T1)$	0
R(B)			B	$ts(T1)$	0
	W(B)		B	$ts(T1)$	$ts(T2)$
	W(C)		C	0	$ts(T2)$
		R(C)	C	$ts(T3)$	$ts(T2)$
		R(B)	B	$ts(T3)$	$ts(T2)$
W(A)			A	$ts(T1)$	$ts(T1)$
		W(C)	C	$ts(T3)$	$ts(T3)$
		R(A)	A	$ts(T3)$	$ts(T1)$
W(B)					
		W(B)			

• Reglas:

- Si T_i ejecuta $Read(Q)$ y $ts(T_i) < W-ts(Q)$ entonces T_i retrocede
- Si T_i ejecuta un $Write(Q)$
 - Si $ts(T_i) < R-ts(Q)$ entonces T_i retrocede
 - Si $ts(T_i) < W-ts(Q)$ entonces T_i retrocede

→ T3 retrocede en cascada porque leyó el dato A escrito por T1.

→ $ts(T1) < R-ts(B) = ts(T3) \Rightarrow T1$ retrocede

Protocolo de estampillas: regla de Thomas

Estampillas: $0 < ts(T1) < ts(T2) < ts(T3)$

Inicialmente: $W-ts(X) = R-ts(X) = 0$, con $X = A, B$ o C

T1	T2	T3	Dato	R-ts	W-ts
R(A)			A	$ts(T1)$	0
R(B)			B	$ts(T1)$	0
	W(B)		B	$ts(T1)$	$ts(T2)$
	W(C)		C	0	$ts(T2)$
		R(C)	C	$ts(T3)$	$ts(T2)$
W(A)			A	$ts(T1)$	$ts(T1)$
W(B)					
		R(B)	B	$ts(T3)$	$ts(T2)$
		W(C)	C	$ts(T3)$	$ts(T3)$
		R(A)	A	$ts(T3)$	$ts(T1)$
		W(B)	B	$ts(T3)$	$ts(T3)$

• Reglas:

- Si T_i ejecuta $Read(Q)$ y $ts(T_i) < W-ts(Q)$ entonces T_i retrocede
- Si T_i ejecuta un $Write(Q)$
 - Si $ts(T_i) < R-ts(Q)$ entonces T_i retrocede
 - Si $ts(T_i) < W-ts(Q)$ entonces por regla de Thomas $Write(Q)$ se omite

$ts(T1) = R-ts(B)$ pero $ts(T1) < W-ts(B) = ts(T2)$:

Se ignora por la regla de Thomas (no se actualizan estampillas ni los valores)

En el protocolo tradicional, sin considerar regla de thomas, T1 retrocede.

Protocolo de Multiversión

- Cada **transacción T_i** se le asigna una **estampilla de tiempo $ts(T_i)$** (hora de inicio de la transacción)
- El **primer $Write(Q)$ de cada transacción crea una nueva versión del dato Q** . Existe una versión inicial Q_0 para cada dato Q .
- Para cada versión Q_k se almacenan:
 - El valor de la versión Q_k
 - $R-ts(Q_k)$ la estampilla de tiempo mayor de todas las transacciones que leyeron Q_k
 - $W-ts(Q_k)$ la estampilla de tiempo de la transacción que creo la versión Q_k
- Cuando una transacción **T_i ejecuta $Read(Q)$** se selecciona la **versión de Q correspondiente a T_i** , esta es: la versión Q_k tal que $W-ts(Q_k)$ es la estampilla mas grande de todas las versiones de Q que verifica $W-ts(Q_k) \leq Ts(T_i)$ (i.e., la versión anterior mas próxima en el tiempo a T_i)
- **Reglas:** para una transacción T_i , sea Q_k la versión correspondiente a T_i (ver ítem anterior)
 - Si T_i ejecuta $Read(Q)$ entonces $R-ts(Q_k)$ se le asigna *Mayor* ($ts(T_i)$, $R-ts(Q_k)$)
 - Si T_i ejecuta $Write(Q)$
 - Si $ts(T_i) < R-ts(Q_k)$ entonces **T_i retrocede** (la escritura “llego tarde”)
 - Si $ts(T_i) = W-ts(Q_k)$ entonces solo se actualiza el valor de Q_k (es la versión que creo T_i)
 - En otro caso se crea una nueva versión Q_i con $R-ts(Q_i) = W-ts(Q_i) = ts(T_i)$

Protocolo de Multiversión

T1	T2	T3	Vers.	valor	R-ts	W-ts
R(A)			A0	11	$ts(T1)$	0
B:=A						
W(B)			B1	11	$ts(T1)$	$ts(T1)$
	C:=23					
	W(C)		C2	23	$ts(T2)$	$ts(T2)$
C:=A+20						
W(C)			C1	31	$ts(T1)$	$ts(T1)$
		R(A)	A0	11	$ts(T3)$	0
		R(C)	C2	23	$ts(T3)$	$ts(T2)$
		B:=C+10				
		W(B)	B3	33	$ts(T3)$	$ts(T3)$
	R(B)		B1	11	$ts(T2)$	$ts(T1)$
A:=A+10						
W(A)						

- **Reglas:** sea Q_k la versión correspondiente a T_i
 - Si T_i ejecuta $Read(Q)$ entonces $R-ts(Q_k)$ se le asigna $Mayor(ts(T_i), R-ts(Q_k))$
 - Si T_i ejecuta $Write(Q)$
 - Si $ts(T_i) < R-ts(Q_k)$ entonces T_i retrocede
 - Si $ts(T_i) = W-ts(Q_k)$ se actualiza el valor de Q_k
 - En otro caso se crea una nueva versión Q_i con $R-ts(Q_i) = W-ts(Q_i) = ts(T_i)$

Estampillas: $0 < ts(T1) < ts(T2) < ts(T3)$

Versiones iniciales: $\langle A0, valor=11, R-ts=0, W-ts=0 \rangle$

$\langle B0, valor=12, R-ts=0, W-ts=0 \rangle$

$\langle C0, valor=13, R-ts=0, W-ts=0 \rangle$

→ T3 retrocede en cascada porque leyó la versión C2 de T2

→ T2 retrocede en cascada porque leyó la versión B1 de T1

→ A0 es la versión correspondiente de A para T1 y $ts(T1) < R-ts(A0) = ts(T3) \Rightarrow T1$ retrocede