

ELEMENTOS DE BASES DE DATOS

Segundo Cuatrimestre 2015

**Protocolos de Control de Concurrencia
basados en Estampillas**

Protocolos basados en hora de entrada

- Los **protocolos basados en hora de entrada - PBDHE** (*time-stamp*) resuelven los conflictos realizando un esquema de ordenamiento según la *hora de entrada o de inicio de cada transacción*.
- Si a T_i se le asignó una hora de entrada $ts(T_i)$ e ingresa una nueva transacción T_j luego: $ts(T_i) < ts(T_j)$.
 - Los PBHE manejan ejecuciones concurrentes de manera que **la estampilla de tiempo de cada transacción determine el orden de serializabilidad**.
- Además de **cada dato Q** se mantienen **dos etiquetas de tiempo**:
 - **R-ts (Q)**, la *mayor estampilla de tiempo de una transacción que ejecutó exitosamente un Read (Q)*.
 - **W-ts (Q)**, la *mayor estampilla de tiempo de una transacción que ejecutó exitosamente un Write (Q)*.

PBHE – Controles

T_i requiere un **Read(Q)**, el control consiste:

- Si $ts(T_i) < W-ts(Q)$ entonces T_i desea leer un valor de Q que ya fue sobrescrito por una transacción más joven. La operación Read es rechazada y T_i *retrocede*
- Si $ts(T_i) \geq W-ts(Q)$ entonces la operación Read se ejecuta **exitosamente**. El valor de $R-ts(Q)$ será el **MAX ($R-ts(Q)$, $ts(T_i)$)**

T_i requiere un **Write(Q)**, control:

- Si $ts(T_i) < R-ts(Q)$ entonces el valor de Q que T_i está produciendo ya fue requerido. La operación Write es rechazada y T_i *retrocede*.
- Si $ts(T_i) < W-ts(Q)$ entonces T_i está intentando escribir un valor ya obsoleto de Q. La operación Write es rechazada y T_i *retrocede*.
- En otro caso, la operación Write se ejecuta exitosamente y $W-ts(Q)$ toma el valor $ts(T_i)$.

| t | T1 | T2 | T3 | T4 | dato | val | Rts | Wts | |
|----|----------|----------|----------|----------|---------------------------------------|-----|----------------------|----------------------|----------------------|
| 1 | | | | C = 31 | C | 31 | | | |
| 2 | | | | write(C) | C | 31 | 0 | ts (T ₄) | |
| 3 | | | read(B) | | B | 20 | ts (T ₃) | 0 | |
| 4 | | | B=B+2 | | B | 22 | | | |
| 5 | read(A) | | | | A | 10 | ts (T ₁) | 0 | |
| 6 | | | C=32 | | C | 32 | | | |
| 7 | | | write(C) | | T3 retrocede (ts(T3) < Wts(C)) B=20 | | | | |
| 8 | A=A+3 | | | | A | 13 | | | |
| 9 | write(A) | | | | A | 13 | ts (T ₁) | ts (T ₁) | |
| 10 | | A=15 | | | A | 15 | | | |
| 11 | | B=A | | | B | 15 | | | |
| 12 | | write(A) | | | A | 15 | ts (T ₁) | ts (T ₂) | |
| 13 | | | | | read(A) | A | 15 | ts (T ₄) | ts (T ₂) |
| 14 | | | | | | | | | |
| 15 | | write(B) | | | | B | 15 | 0 | ts (T ₂) |
| 16 | | | | | read(C) | C | 31 | ts (T ₄) | ts (T ₄) |

Suponga que los **valores iniciales** de los datos son: **A=10, B=20, C=30**, que la **modificación es inmediata** y que las estampillas de tiempo son: **Ts(T1) < Ts(T2) < Ts(T3) < Ts(T4)**. Todas las transacciones realizan su commit luego del instante 22 si no han sido retrocedidas.

Complete la siguiente tabla: Deberá mostrarse como se actualizan las estampillas de tiempo y los valores de los datos.

En caso de producirse un retroceso por la violación del protocolo indique:

- en que punto se produce,
- que regla del protocolo a fallado
- que transacciones están en conflicto,
- cuales transacciones retroceden.

Indique además los valores finales de los datos.

| t | T1 | T2 | T3 | T4 | dato | val | Rts | Wts | |
|----|----------|----------|----|----|--|---------------------|---------------------|---------------------|---------------------|
| 8 | A=A+3 | | | | A | 13 | | | |
| 9 | write(A) | | | | A | 13 | ts(T ₁) | ts(T ₁) | |
| 10 | | A=15 | | | A | 15 | | | |
| 11 | | B=A | | | B | 15 | | | |
| 12 | | write(A) | | | A | 15 | ts(T ₁) | ts(T ₂) | |
| 13 | | | | | read(A) | A | 15 | ts(T ₄) | ts(T ₂) |
| 14 | | | | | | | | | |
| 15 | | write(B) | | | B | 15 | 0 | ts(T ₂) | |
| 16 | | | | | read(C) | C | 31 | ts(T ₄) | ts(T ₄) |
| 17 | C=A+10 | | | | C | 23 | | | |
| 18 | write(C) | | | | T1 retrocede (ts(T ₁) < Rts(C)) (sola) | | | | |
| 19 | | | | | | | | | |
| 20 | | | | | | | | | |
| 21 | | | | | | | | | |
| 22 | | read(A) | | A | 15 | ts(T ₄) | ts(T ₂) | | |

Indique además los valores finales de los datos.

A = 15

B = 15

C = 31

Regla de Escritura de Thomas

Supongamos que T_i realiza un **Write(Q)**. Control:

- Si $ts(T_i) < R-ts(Q)$ entonces el valor de Q que T_i está produciendo fue requerido previamente y el sistema asumió que nunca se produciría. Por lo tanto, la operación Write es rechazada y la transacción T_i *retrocede*.
- Si $ts(T_i) < W-ts(Q)$ entonces T_i está intentando escribir un valor obsoleto de Q. Por lo tanto, esta operación puede ser ignorada.
- En otro caso, la operación Write se ejecuta exitosamente y $W-ts(Q)$ toma el valor $ts(T_i)$.

| t | T1 | T2 | T3 | T4 | dato | val | Rts | Wts |
|----|----------|----------|----------|----------|---------------------------------------|-----|----------------------|----------------------|
| 1 | | | | C = 31 | C | 31 | | |
| 2 | | | | write(C) | C | 31 | 0 | ts (T ₄) |
| 3 | | | read(B) | | B | 20 | ts (T ₃) | 0 |
| 4 | | | B=B+2 | | B | 22 | | |
| 5 | read(A) | | | | A | 10 | ts (T ₁) | 0 |
| 6 | | | C=32 | | C | 32 | | |
| 7 | | | write(C) | | Se ignora (ts(T3) < Wts(C)) | | | |
| 8 | A=A+3 | | | | A | 13 | | |
| 9 | write(A) | | | | A | 13 | ts (T ₁) | ts (T ₁) |
| 10 | | A=15 | | | A | 15 | | |
| 11 | | B=A | | | B | 15 | | |
| 12 | | write(A) | | | A | 15 | ts (T ₁) | ts (T ₂) |
| 13 | | | | read(A) | A | 15 | ts (T ₄) | ts (T ₂) |
| 14 | | | write(B) | | B | 22 | ts (T ₃) | ts (T ₃) |
| 15 | | write(B) | | | T2 retrocede (ts(T2) < Rts(B)) y T4 | | | |
| 16 | | | | | | | | |

Suponga que los **valores iniciales** de los datos son: **A=10, B=20, C=30**, que la **modificación** es **inmediata** y que las estampillas de tiempo son: **Ts(T1) < Ts(T2) < Ts(T3) < Ts(T4)**. Todas las transacciones realizan su commit luego del instante 22 si no han sido retrocedidas.

Complete la siguiente tabla: Deberá mostrarse como se actualizan las estampillas de tiempo y los valores de los datos.

En caso de producirse un retroceso por la violación del protocolo indique:

- en que punto se produce,
- que regla del protocolo a fallado
- que transacciones están en conflicto,
- cuales transacciones retroceden.

Indique además los valores finales de los datos.

| t | T1 | T2 | T3 | T4 | dato | val | Rts | Wts |
|----|----------|----------|----------|---------|--|-----|-----------|-----------|
| 7 | | | write(C) | | C | 22 | $ts(T_3)$ | $Wts(C)$ |
| 8 | A=A+3 | | | | A | 13 | | |
| 9 | write(A) | | | | A | 13 | $ts(T_1)$ | $ts(T_1)$ |
| 10 | | A=15 | | | | | | |
| 11 | | B=A | | | | | | |
| 12 | | write(A) | | | | | | |
| 13 | | | | read(A) | | | | |
| 14 | | | write(B) | | B | 22 | $ts(T_3)$ | $ts(T_3)$ |
| 15 | | write(B) | | | T2 retrocede ($ts(T_2) < Rts(B)$) y T4 | | | |
| 16 | | | | | | | | |
| 17 | C=A+10 | | | | C | 23 | | |
| 18 | write(C) | | | | Se ignora ($ts(T_1) < Wts(C)$) | | | |
| 19 | read(B) | | | | T1 retrocede ($ts(T_1) < Rts(B)$) A=10 | | | |
| 20 | | | A=40 | | A | 40 | | |
| 21 | | | write(A) | | A | 40 | 0 | $ts(T_3)$ |
| 22 | | | | | | | | |

Ullman pag 974

4. Suppose the scheduler receives a request to abort T or decides to Rollback T as in 1b or 2c. Then any transaction that was waiting on an element X that T wrote **must repeat its attempt to read or write** and see whether the action is now legal after the aborted transaction's writes are cancelled.

Indique además los valores finales de los datos.

A = 40

B = 22

C = 32

Multiversión

- Con cada **ítem de dato Q** se asocia una secuencia de **versiones $\langle Q_1, Q_2, \dots, Q_n \rangle$** .
- Para cada versión Q_k se tienen tres ítems de dato:
 - **Content**: es el valor de la versión de Q.
 - **$R-ts(Q_k)$** : la mayor estampilla de tiempo de una transacción que leyó exitosamente la versión Q_k .
 - **$W-ts(Q_k)$** : la estampilla de tiempo de la transacción que creó la versión de Q_k .
- Un **Write(Q) exitoso eventualmente crea una nueva versión de Q**. Los valores de $R-ts$ y $W-ts$ de la nueva versión se inicializan con $ts(T_i)$.
- El valor de $R-ts$ se actualiza **si una transacción T_j lee el contenido de Q_k y $R-ts(Q_k) < ts(T_j)$** .

Multiversión

Supongamos que una transacción T_i realiza una operación Read o Write. Sea Q_k la versión de Q cuya **estampilla de tiempo de escritura** es la más grande estampilla menor o igual a $ts(T_i)$.

- 1) Si T_i realiza un **Read(Q)**, entonces el valor retornado es el contenido de la versión Q_k .
- 2) Si T_i realiza un **Write(Q)**, entonces tenemos tres casos:
 - Si $ts(T_i) < R-ts(Q_k)$ entonces la transacción T_i **retrocede**.
 - Si $ts(T_i) = W-ts(Q_k)$ entonces se **sobreescribe** el contenido de Q_k .
 - En otro caso, se crea una **nueva versión** de Q .

| t | T1 | T2 | T3 | T4 | dato | val | Rts | Wts |
|----|----------|----------|----------|----------|---|-----|----------------------|----------------------|
| 1 | | | | C = 31 | C | 30 | | |
| 2 | | | | write(C) | C ₁ | 31 | ts (T ₄) | ts (T ₄) |
| 3 | | | read(B) | | B ₀ | 20 | ts (T ₃) | 0 |
| 4 | | | B=B+2 | | B | 22 | | |
| 5 | read(A) | | | | A ₀ | 10 | ts (T ₁) | 0 |
| 6 | | | C=32 | | C | 32 | | |
| 7 | | | write(C) | | C ₂ | 32 | ts (T ₃) | ts (T ₃) |
| 8 | A=A+3 | | | | A | 13 | | |
| 9 | write(A) | | | | A ₁ | 13 | ts (T ₁) | ts (T ₁) |
| 10 | | A=15 | | | A | 15 | | |
| 11 | | B=A | | | B | 15 | | |
| 12 | | write(A) | | | A ₂ | 15 | ts (T ₂) | ts (T ₂) |
| 13 | | | | read(A) | A ₂ | 15 | ts (T ₄) | ts (T ₂) |
| 14 | | | write(B) | | B ₁ | 22 | ts (T ₃) | ts (T ₃) |
| 15 | | write(B) | | | T2 retrocede (ts(T2) < Rts(B)) y T4 | | | |
| 16 | | | | | | | | |

Suponga que los **valores iniciales** de los datos son: **A=10, B=20, C=30**, que la **modificación** es **inmediata** y que las estampillas de tiempo son: **Ts(T1) < Ts(T2) < Ts(T3) < Ts(T4)**. Todas las transacciones realizan su commit luego del instante 22 si no han sido retrocedidas.

Complete la siguiente tabla: Deberá mostrarse como se actualizan las estampillas de tiempo y los valores de los datos.

En caso de producirse un retroceso por la violación del protocolo indique:

- en que punto se produce,
- que regla del protocolo a fallado
- que transacciones están en conflicto,
- cuales transacciones retroceden.

Indique además los valores finales de los datos.

| t | T1 | T2 | T3 | T4 | dato | val | Rts | Wts | | |
|----|----------|----------|----------------|----|---------------------------------------|---------------------|---------------------|---------------------|--|--|
| 7 | | | write(C) | | C ₂ | 32 | ts(T ₃) | ts(T ₃) | | |
| 8 | A=A+3 | | | | A | 13 | | | | |
| 9 | write(A) | | | | A ₁ | 13 | ts(T ₁) | ts(T ₁) | | |
| 10 | | A=15 | | | | | | | | |
| 11 | | B=A | | | | | | | | |
| 12 | | write(A) | | | | | | | | |
| 13 | | | read(A) | | | | | | | |
| 14 | | | write(B) | | B ₁ | 22 | ts(T ₃) | ts(T ₃) | | |
| 15 | | write(B) | | | T2 retrocede (ts(T2) < Rts(B)) y T4 | | | | | |
| 16 | | | | | | | | | | |
| 17 | C=A+10 | | | | | | | | | |
| 18 | write(C) | | C ₃ | | 23 | ts(T ₁) | ts(T ₁) | | | |
| 19 | read(B) | | B ₀ | | 20 | ts(T ₃) | 0 | | | |
| 20 | | | A=40 | | A | 40 | | | | |
| 21 | | | write(A) | | A ₃ | 40 | ts(T ₃) | ts(T ₃) | | |
| 22 | | | | | | | | | | |

Indique además los valores finales de los datos.

A = 40

B = 22

C = 23