Bases de Datos

Protocolo de Validación

Dr. Diego R. Garcia

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Protocolo de Validación

- Las transacciones tienen 3 etapas en el siguiente orden:
  1. **Etapa de Lectura** (entre Start y Validation): Se leen datos de la B.D. y las escrituras se realizan en variables locales (se usa modificación diferida).
  2. **Etapa de Validación**: se determina si transacción puede copiar a la B.D. el contenido de sus variables locales o debe retroceder
  3. **Etapa de Escritura**: (entre Validation y Finish): Si pasa la validación (no retrocedió), se actualiza la B.D. con los valores de las variables locales.
- A cada transacción Ti se le asocian 3 estampillas de tiempo:
  - Start(Ti) (o abreviado **Si**): hora en que comenzó a ejecutarse
  - Validation(Ti) (o **Vi**): hora en que comenzó su etapa de validación
  - Finish(Ti) (o **Fi**): hora en que termino su etapa de escritura (i.e. cometió)

Protocolo de Validación

- Para analizar si una transacción valida (pasa la etapa de validación) se procede siguiendo el orden de la estampillas de validación (se analiza 1ero la que tiene el tiempo de validación mas viejo)
- Una transacción Tj valida si para toda transacción Ti (que no haya retrocedido) tal que Validation(Ti) < Validation(Tj) se cumple alguna de las siguientes condiciones:
  - Finish(Ti) < Start(Tj) (están en serie) **o**
  - Los datos que Ti escribe **no** tienen intersección con los datos que Tj lee (WS(Ti) ∩ RS(Tj)= Ø) **y** Start(Tj) < Finish(Ti) < Validation(Tj) (esto es, no validan concurrentemente, Ti terminó antes de que Tj comience a validar).

Si no se cumple ningúna de estas condiciones la transacción Tj no valida, retrocede y no será considerada para la validación de las transacciones que validen después.

Protocolo de Validación: ejemplo

T1	T2	T3	T4	T5
START				
R(B)				
		START		
	START			
	R(A)			
			START	
			R(C)	
	R(B)			
W(B)				
	W(C)			
		W(C)		
			W(C)	
VALID ✓				
OUTPUT(B)				
FINISH				
	VALID ✗			
				START
	OUTPUT(C)			R(B)
	FINISH			
			VALID ✓	
		VALID ✗		
		OUTPUT(C)		
			OUTPUT(C)	
			FINISH	
		FINISH		
				VALID ✓
				FINISH

Representación alternativa

S1,S3,S2,S4,V1✓,F1,V2✗,S5,F2,V4✓,V3✗,F4,F3,V5✓,F5

	Read-Set (RS)	Write-Set (WS)
T1	{B}	{B}
T2	{A,B}	{C}
T3	Ø	{C}
T4	{C}	{C}
T5	{B}	Ø

Analicemos que transacciones validan y cuales retroceden:

1. T1 valida porque es la primera en validar, es decir no existe Ti tal que Vi<V1
2. T2 se valida contra T1 (V1<V2) y no valida (retrocede) porque:
   S2 < F1 (no están en serie) y WS(T1) ∩ RS(T2)={B} ≠ Ø
3. T4 se valida sólo contra T1 (porque T2 retrocedió). T4 valida porque:
   S4 < F1 < V4 y WS(T1) ∩ RS(T4)= Ø
4. T3 se valida contra T1 y T4 que lograron validar antes.
   - Contra T1 valida porque: S3 < F1 < V3 y WS(T1) ∩ RS(T3)= Ø
   - Contra T4 no valida porque: S3 < F4 (no están en serie) y no se cumple S3 < F4 < V3 porque V3 < F4 (validan concurrentemente)
5. T5 se valida contra T1 y T4 que lograron validar antes.
   - Contra T1 valida porque F1 < S5 (están en serie)
   - Contra T4 valida porque S5 < F4 < V5 y WS(T4) ∩ RS(T5)= Ø