



Trabajo Practico Control de Concurrencia

Objetivo

- Asimilar los contenidos, comprender el problema de control de concurrencia y planificación en un sistema de gestión de bases de datos centralizado.
- Explicitar algunas definiciones extraídas de otras fuentes de información adicionales a la bibliografía básica para facilitar la comprensión del contenido

Requisitos

- Contenidos teóricos de Sistema Control de Concurrencia, Unidad IV

Introducción - Definiciones

Transacción=T=Una transacción T es una ejecución de un programa P que accede a la BD. Un mismo programa P puede ejecutarse varias veces. Cada ejecución de P es una transacción T_i . Una transacción es una sucesión de acciones u operaciones. Una acción es un paso atómico y pueden ser:

leer un ítem X de T_i : $ri[X]$

escribir un ítem X de T_i : $wi[X]$

abortar T_i : ai

commit T_i : ci

Planificación o Historia o Schedule P de n transacciones concurrentes $T_1, T_2 \dots T_n$ es una secuencia de las operaciones realizadas por dichas transacciones, sujeta a la restricción de que para cada transacción T_i que participa en P, sus operaciones aparecen en P en el mismo orden en el que ocurren en T_i .

Ejecución concurrente de n T's (ejecución entrelazada de T's)=problemas interferencia (ejemplo: actualización perdida (lost update)y lectura sucia (dirty read))

actualización perdida=se pierde la actualización hecha por una transacción T_1 por la acción de otra transacción T_2 sobre el mismo ítem.



UNIVERSIDAD NACIONAL DE LUJÁN
Departamento de Ciencias Básicas, División Sistemas
Licenciatura en Sistemas de Información (RES.HCS 009/12)
11078 Base de Datos II

Ejemplo:

Supongamos el programa $P = \text{Read}(A); A := A + 1; \text{Write}(A);$

y dos ejecuciones de P, T1 y T2 sobre el ítem A, con el siguiente entrelazamiento:

T1	T2	A (valor inicial en disco: 5)	
Read(A)		5	
	Read(A)	5	
$A := A + 1$		5	
	$A := A + 1$	5	
	Write(A)	6	
Write(A)		6	<-- valor incorrecto de A, se perdió la actualización hecha por T2

Nota:

Read(A) copia el valor del ítem A en disco a la variable local de la transacción.

Write(A) copia el valor de la variable local de la transacción al ítem A en disco.

$A := A + 1$ se hace sobre la variable local de la transacción.



UNIVERSIDAD NACIONAL DE LUJÁN
Departamento de Ciencias Básicas, División Sistemas
Licenciatura en Sistemas de Información (RES.HCS 009/12)
11078 Base de Datos II

Lectura sucia=ocurre cuando una transacción T2 lee un valor de un ítem dejado por otra transacción T1 que no hizo commit antes de que T2 leyera el ítem. Observemos que si T1 aborta, T2 se quedó con un valor sucio que será deshecho por el rollback de T1. Esto además podría producir un fenómeno no deseado como es el abort en cascada (si T2 leyó de T1 que abortó, deberíamos abortar T2, luego si T3 leyó de T2 deberíamos abortar a su vez T3 y así sucesivamente).

Ejemplo: Supongamos los programas : P1= Read(A); A:=A-1; Write(A); Read(B); B:=B/A; Write(B) y P2= Read(A); A:=A*2; Write(A); y dos ejecuciones: una T1 de P1 sobre los ítems A y B, y una T2 de P2 sobre el ítem A, con el siguiente entrelazamiento:

T1	T2	A (valor inicial en disco: 1)	B (valor inicial en disco: 2)	
Read(A)		1	2	
A:=A-1		1	2	
Write(A)		0	2	
	Read(A)	0	2	
	A:=A*2	0	2	
Read(B)		0	2	
	Write(A)	0	2	
B:=B/A		0	2	<-- T1 falla por la división por cero y aborta volviendo A al valor anterior, pero T2 ya leyó A

Fallos en la ejecución concurrente de n T's (ejecución entrelaza de T's)=problemas de recuperación



Clasificación de las planificaciones según su recuperabilidad:

No Recuperables (No RC)

Recuperables (RC)=una planificación es recuperable, si cada T_i hace su commit después de que hayan hecho commit todas las T_j de las cuales lee.

Evitan Aborts en Cascada (ACA) (Avoids Cascade Aborts)=una planificación evita aborts en cascada, si cada T_i puede leer solamente aquellos valores que fueron escritos por transacciones que ya hicieron commit (o por si misma).

Estrictas (ST)= una planificación es estricta, si ningún item X puede ser leído o sobrescrito hasta que la transacción que previamente escribió X haya finalizado haciendo commit o abort.

Operacion conflictiva=dos operaciones son conflictivas si pertenecen a distintas T y aplican al mismo item de dato y al menos una de ellas es una operación de escribir.

Planificación Equivalente=dos planificaciones $S1$ y $S2$ son equivalentes si:

- 1) Si están definidas sobre el mismo conjunto de transacciones.
- 2) Las operaciones conflictivas tienen el mismo orden.

Planificación serial= $S1$ es serial si para todo par de transacciones T_i y T_j en $S1$, todas las operaciones de T_i preceden a las de T_j o viceversa.

Las planificaciones seriales (y las equivalentes a estas) son las que consideraremos como planificaciones correctas.

Una planificación S es serializable (SR) si es equivalente a una planificación serial.

Grafo de precedencia o seriabilidad=se construye poniendo un nodo por cada transacción, y un arco de una transacción T_i a T_j si alguna operación de T_i precede y conflictúa con otra de T_j . Es decir, dadas dos transacciones que operan sobre un mismo item, si alguna de las dos hace una operación de escritura (write), se hace un arco de la primera que accede el item hacia la segunda.

Si el Grafo de precedencia de una planificación es acíclico \rightarrow planificación serializable.

Objetivo=implementar un sistema de control de concurrencia (SCC) que tenga un planificador paralelo (permite ejecuciones de n T 's en paralelo) que utilice un protocolo que asegure seriabilidad.



Ejercicios

1) Realice los grafos de precedencia de las siguientes planificaciones:

1.a) $S1=R1(X), R2(Y), W2(X), W1(Y), C1, C2$

T1	T2
R1(X)	
	R2(Y)
	W2(X)
W1(Y)	

1.b) $S2=R1(X), W1(Y), R2(Y), W2(X), C2, C1$

T1	T2
R1(X)	
W1(Y)	
	R2(Y)
	W2(X)

1.c) $S3=R1(X), R2(Y), W1(Y), W2(X), C2, C1$

T1	T2
R1(X)	
	R2(Y)
W1(Y)	
	W2(X)

1.d) $S4=R0(X), R1(X), R1(Y), R2(Z), W2(Z), R1(Z), W1(Z), W0(X), W0(Z)$

T0	T1	T2
R0(X)		
	R1(X)	
	R1(Y)	
		R2(Z)
		W2(Z)
	R1(Z)	
	W1(Z)	
W0(X)		
W0(Z)		



UNIVERSIDAD NACIONAL DE LUJÁN
Departamento de Ciencias Básicas, División Sistemas
Licenciatura en Sistemas de Información (RES.HCS 009/12)
11078 Base de Datos II

2) Sean las transacciones:

$T1=R1(X), W1(Y), C1$

$T2=R2(Y), W2(X), C2$

Se pide para las siguientes planificaciones:

2.a) $S1=R1(X), R2(Y), W2(X), W1(Y), C1, C2$

2.b) $S2=R1(X), W1(Y), R2(Y), W2(X), C2, C1$

2.c) $S3=R1(X), R2(Y), W1(Y), W2(X), C2, C1$

decir si son:

- 1) Serializables
- 2) Recuperables
- 3) Evitan Abort en Cascada
- 4) Estrictas.

Atte. Guillermo Cherencio / Juan C. Romero

11078 Base de Datos II

División Sistemas

Departamento de Ciencias Básicas

UNLu