

Tugas Individu 2 SISP

Marcus wahid Sabri Hansen 11102133

Soal 1

langkah	T ₁	T ₂	langkah	T ₁	T ₂
1			9		read (R)
2		read (P)	10		write (Q)
3		read (Q)	11		read (Q)
4			12		write (Q)
5		write (Q)	13		read (P)
6			14		read (P)
7		read (R)	15		write (Q)
8		write (P)	16		read (Q)

① Apakah schedule Tx conflict serializable, view setia atau tidaknya, atau tidak keduanya?

-> Schedule diketahui conflict serializable, jika urutannya dapat direpresentasikan sebagai precedence graph yang tidak punya siklus

-> Schedule diketahui view serializable, jika urutannya dapat dilibatkan menjadi serial schedule yg ekivalen secara view
-> Setelah dilakukan analisis, schedule ini tidak conflict serializable karena menghasilkan siklus dalam precedence graph

-> Namun, schedule ini view serializable karena memenuhi syarat view serializable (yaitu reads, update order, dan final writes)

② Buktikan bahwa schedule Tx conflict serializable, view serializable, keduanya atau tidak keduanya dan buktikan serial schedule yg ekivalen dengan schedule ^{ya!}
TANAKI

→ Untuk membuat serial schedule yg ekivalen secara view dengan schedule yang diberikan, kita perlu memastikan bahwa urutan operasi mengikuti syarat view serializability:

1. Initial reads: harus fetus dilakukan oleh transaksi yang sama pada item yang sama.

2. update order (write): pada item yang sama harus sesuai dengan urutan dalam schedule asli.

3. final writes : pada setiap item data dilakukan oleh transaksi yg sama seperti dalam schedule asli

Tabelnya:

Langkah	T_1	T_2	Langkah	T_1	T_2
1	Read (P)		9		Write (S)
2	Read (Q)		10		Read (P)
3	Write (Q)		11		Read (Q)
4	Write (P)		12		Write (Q)
5	Read (P)		13		Read (P)
6	Write (Q)		14		Write (Q)
7		Read (P)			
8		Read (S)			

Pengakalan konflik (dari schedule awal)

-> Item P:

→ Langkah 8 (T_1) write (P) dan langkah 14 (T_2) read (P) → WR Conflict

-> Item Q:

→ Langkah 3 (T_1) read (Q) dan langkah 10 (T_1) write (Q) → RW Conflict

→ Langkah 5 (T_1) write (Q) dan langkah 12 (T_2) write (Q) → WW Conflict

→ Langkah 10 (T_1) write (Q) dan langkah 11 (T_2) read (Q) → RW Conflict

- Alasan schedule TX conflict serializable
 - Berdasarkan grafik dependensi, tidak ada siklus
 - Pada transaksi yg terlibat, keduanya sudah memidentifikasi konflik antara T_1 , T_2 , dan T_3
- Alasan schedule TX view serializable
 - Hasil akhir yg konsisten, dapat menemukan uniten serial dimana setiap transaksi mendapatkan data yg sama seperti yg dituliskan dalam schedule TX
 - Akses data konsisten, jika T_1 menuju data ke item 1 dan T_3 membaca item 2, maka uniten serial harus pasti T_1 diksekusi dahulu sebelum T_3 .
 - Icesesision hasil, jika dapat produkli hasil yg sama schedule tersebut juga bisa dikatakan view serializable

Soal 2

Langkah	T ₁	T ₂	T ₃	
1	{		read(x)	
2			read(z)	
3	write(z)			
4		read(y)		
5		write(y)		
6	read(g)			
7	write(g)			
8			write(x)	
9	read(z)			
10	write(x)			

① Apakah schedule Tx conflict serializable, view serializable, redundant, atau tidak redundannya?

→ • conflict serializable : schedule dikatakan conflict serializable jika dapat disusun menjadi serial schedule tanpa adanya konflik yang tidak bisa diatasi, khususnya mencari konflik antar operasi read & write

• view serializable : schedule dikatakan view serializable jika hasil dari semua operasi yang dilakukan oleh schedule sama dengan hasil dari beberapa serial schedule.

(langkah-langkah analisis):

→ Identifikasi konflik:

- item y:

langkah 5 (T_2) write (y) dan langkah 6 (T_1) read (y) \rightarrow WR conflict

langkah 6 (T_1) read (y) dan langkah 7 (T_1) write (y) \rightarrow RW conflict

- item z:

langkah 3 (T_1) write (z) dan langkah 2 (T_3) read (z) \rightarrow WR conflict

langkah 9 (T_1) read (z) dan langkah 3 (T_1) write (z) \rightarrow RW conflict

- item x:

langkah 1 (T_2) read (x) dan langkah 8 (T_3) write (x) \rightarrow RW conflict

Maka, schedule T_x adalah conflict serializable

② Buktikan bahwa schedule T_x conflict serializable, view serializable, (edarnya atau tidak) jadilah dan buktikan serial schedule yang ekivalen dengan schedule T_x .

→ Schedule T_x adalah conflict serializable dan view serializable, buktinya adalah sebagai berikut:

1. Grafik Dependensi:

- Dari T_2 ke T_1 (karena T_2 write (y) mempengaruhi T_1 read (y))
- Dari T_1 ke T_3 (karena T_1 write (z) mempengaruhi T_3 read (z))
- Dari T_2 ke T_3 (karena T_2 read (x) mempengaruhi T_3 write (x))

dengan demikian grafik dependensi:

• $T_2 \rightarrow T_1$

• $T_1 \rightarrow T_2$

• $T_2 \rightarrow T_3$

Kesimpulan: Confict serializable dikatakan demikian apabila dapat disusun ulang (e dalam bentuk serial (urutan transaksi yg dijalankan sebenarnya (satu persatu) tanpa mengulah hasil akhir yg dihasilkan oleh operasi dalam transaksi tersebut

TANAKI