

Transaction schedules \rightarrow only reqd when multiple transactions executing parallelly

T_1 : read(A);
A := A - 50;
write(A);
read(B);
B := B + 50;
write(B);

T_2 : read(A);
temp := A * 0.1;
A := A - temp;
write(A);
read(B);
B := B + temp;
write(B);

A = 1000
B = 2000

A = 950
B = 2050

		Serial schedule			
T_1	T_2	T_1	T_2	T_1	T_2
read(A)			read(A)		read(A)
A := A - 50			temp := A * 0.1		temp := A * 0.1
write(A)			A := A - temp		A := A - temp
read(B)			write(A)		write(A)
B := B + 50			read(B)		read(B)
write(B)			B := B + temp		B := B + temp
	read(A)		write(B)		write(B)
	temp := A * 0.1				
	A := A - temp		read(A)		
	write(A)		A := A - 50		
	read(B)		write(A)		
	B := B + temp		read(B)		
	write(B)		B := B + 50		
			write(B)		

Schedule 1: A + B = 3000, consistent

Schedule 2: Different from schedule 1
A = 850
B = 2150

τ_1	τ_2
read(A)	
$A := A - 50$	
write(A)	
	read(A)
	temp := A * 0.1
	$A := A - \text{temp}$
	write(A)
read(B)	
$B := B + 50$	
write(B)	
	read(B)
	$B := B + \text{temp}$
	write(B)

→ concurrent schedule

Schedule 3

Schedule 1 & schedule 3 are equivalent-

τ_1	τ_2
read(A)	
$A := A - 50$	
950	
	read(A)
	temp := A * 0.1
	$A := A - \text{temp}$ 900
	write(A) → $A = 900$
	read(B)
	2000
$A = 950$ write(A)	
2000 read(B)	
$B := B + 50$	
2050 write(B)	
$B := B + \text{temp}$ → 2100	
	write(B) → $A = 2100$

both will have different local buffers

Schedule 4

Another form of schedule 3

class

Date

Page

T_1	T_2
read(A)	
write(A)	
	read(A)
	write(A)
read(B)	
write(B)	
	read(B)
	write(B)

read & write opsⁿ of diff. data items can be swapped to different order.

$\begin{matrix} \text{read(A)} \nearrow \\ \text{read(B)} \searrow \end{matrix}$
 $\begin{matrix} \text{read(A)} \nearrow \\ \text{write(B)} \searrow \end{matrix}$

Non-conflicting instructions

Conflict serializability

Schedule S for transactions T_i & T_j having consecutive instructions I_i and I_j .

1. $I_i = \text{read}(Q), I_j = \text{read}(Q) \rightarrow$ No conflict
2. $I_i = \text{read}(Q), I_j = \text{read} \rightarrow \text{write}(Q)$
3. $I_i = \text{write}(Q), I_j = \text{read}(Q)$
4. $I_i = \text{write}(Q), I_j = \text{write}(Q)$