CS348-Homework 5 (Extra Credit)

Fall 2018

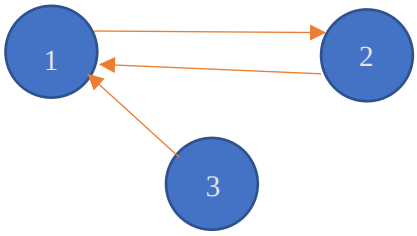
Extra Credit Homework. Will count for up to 5 points of your total grade in the course.
Due: Friday December 7, 2018, 11:59PM **(Firm Deadline - No Extensions or late days)**

1) (30 Points) Consider the transaction schedule S below where R(X) and W(X) stand for read and write:

S =R1(A); W1(A); W1(E); R2(A); R1(B); R3(C); W2(A); W3(C); W1(B); R2(D); R1(C); W2(D); W1(C); W2(E); R3(F); R1(D); W3(F); W1(D); W2(A)

For the above schedule-

a) (15 Points) Draw the precedence graph.



b) (5 Points) State whether or not the schedule is conflict serializable.
There is a cycle between 1 and 2 in the graph, therefore it is **not conflict serializable**.

c) (10 Points) If the schedule is conflict serializable, provide the equivalent serial schedule. If the schedule is not conflict serializable, briefly explain why.
The schedule is not conflict serializable because there is a cycle between point 1 and point 2.

2) (20 Points) Consider the lock requests in the table below:
Here, S(·) and X(·) stand for 'Shared Lock' and 'Exclusive Lock', respectively. T1, T2, and T3 represent three transactions. LM stands for 'lock manager' and transactions will never release a granted lock. t_i refers to time, where $1 \leq i \leq n$.

Time	t_1	t_2	t_3	t_4	t_5	t_6	t_7
T_1	X(P)						S(R)
T_2			S(Q)	S(R)		S(P)	
T_3		S(R)			X(Q)		
LM	BLCK	BLCK	BLCK	BLCK	BLCK	BLCK	BLCK

For the lock requests in above table, decide which lock will be granted or blocked by LM. Fill up the entries in LM row.

a) (14 Points) Write 'GRNT' in the entry when the lock can be granted at time t_+ and 'BLCK' when the lock is blocked (the lock acquisition fails).

b) (6 Points) Provide a brief explanation to justify your answer.

3) (30 Points) Given the following transaction schedules, state whether there is any of the following concurrency related problem:

Overwriting Uncommitted Data (WW conflict) / Dirty Read(WR conflict)/ Unrepeatable Reads (RW conflict)) in the each of the schedules, and explain why (with details). a) (15 Points)

TIME	T1	T2
1	Read(X);	
2	X = X - 50;	
3		Read(X);
4		X = X + 10;
5		Write(X);
6		Commit;
7	Read(X);	
8	Write(X);	
9	Commit;	

There is an overwriting uncommitted data in the transaction schedule. X is read and manipulated on times 1 and 2 in T1, but then it is read again on time 3 in T2 before it was committed in T1. Also, there is an unrepeatable read. Since, in T1 X is read on time 1 and manipulated on time 2, but then X is read again on time 7 before the manipulated value for X was written.

b) (15 Points)

TIME	T1	T2
1	Read(X);	
2	X= X – 200;	
3		Read (Y);
4		Y = Y - 200;
5	Write(X);	
6		Write(Y);
7	Read(Z);	
8	Z = Z+200;	
9		Read(Z);

10		Z = Z+200;
11		Write(Z);
12		Commit;
13	Write(Z);	
14	Commit;	

There is an overwriting uncommitted data in the transaction schedule. In T1at time 7 Z is read in and at time 8 Z is manipulated. Then in T2 and time 9 Z is read in again before it could be written in T1.

4) (20 Points) Explain each of the ACID properties (in details) with suitable examples.

Atomicity- This means that either the entire transaction takes place and completes or none take place at all. An example would be if a bank transfer of $50 is being made from account A to account B and the transfer fails after deducting $50 from account A and before adding $50 to account B. The bank transfer must be completed all the way through to ensure correctness and have atomicity.

Consistency- This means that the execution of a transaction in isolation preserves the consistency of the database. In other words, the database should be consistent before and after the transaction has finished executing. An example would be if bank account A has $100 and account B has $100 and $50 is transferred from account A to account B the sum of both accounts should still be the same. Thus, account A would be $50 and account B would have $150, but they still have a sum of $200 as they did before the transfer.

Isolation- This allows multiple transactions to happen at the same and does not allow the transactions to be aware of the other transactions happening. For example, if a balance transfer between account A and B begins and another balance transfer between account B and C begins after account A has been deducted for its transfer to account B, then at the end they will display the right balances because of isolation. If isolation was not a part of this example, then it is possible that account B would display the wrong balance depending on when it was printed.

Durability- This means that once a transaction has completed successfully it will be shown in the database even if there is a system failure. For instance, if a balance transfer is made between two accounts and the system fails right after the transfer completed successfully. When the system is booted up again and fixed the balance transfer will be left in the correct state it was after the transfer completion.