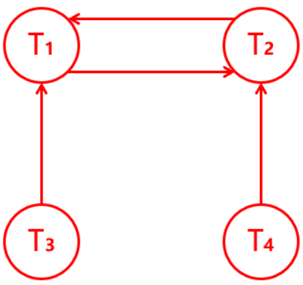
CS150A Homework 3 – Writing

School of Information Science and Technology
January 6, 2025

1 Concurrency Control(20 pts)

T1	R(A)		R(B)				W(A)	
T2		R(A)		R(B)				W(B)
T3					R(A)			
T4						R(B)		

1. (1) Draw the conflict dependency graph for the schedule.

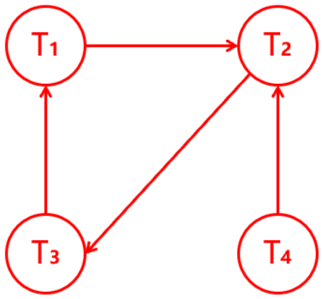


(2) Is this schedule conflict serializability? If so, write down all the conflict equivalent serial schedules? If not,explain the reason.

No. There is a cycle between T_1 and T_2.

T1	S(A)	S(D)		S(B)					
T2			X(B)				X(C)		
T3					S(D)	S(C)			X(A)
T4								X(B)	

2. (1) Draw the "waits-for" graph and explain whether or not there is a deadlock.



There is a deadlock because there is a cycle between T_1, T_2 and T_3.

(2) If we try to avoid deadlock by using the wait-die deadlock avoidance policy, would any transactions be aborted? Assume T1 < T2 < T3 < T4 for the priority.

At timestep = 4, T_1 wants to get an S lock from T_2, since T_1 has lower priority, then T_1 will be aborted.
At timestep = 7, T_2 wants to get an X lock from T_3, since T_2 has lower priority, then T_2 will be aborted.

2 Recovery(15 pts)

The database has just crashed owing to the operator error, and this time you follow the recovery process with the **STEAL/NO FORCE** policy. You boot the database server back up, and find logging information on disk at the following tables:

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	Begin Checkpoint	-
90	update: T1 writes P3	70
100	End Checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

Table 2.1: log records

Transaction ID	LastLSN	Status
T1	70	Running
T2	60	Running
T3	30	Running
T4	50	Running

Table 2.2: Transaction Table

Page ID	RecLSN
P5	50
P1	40

Table 2.3: Dirty Page Table

1. Was the update to page 5 at LSN 60 successfully written to disk? Also, was the update to page 2 at LSN 70 successfully written to disk? Please explain both cases briefly.

   The update at LSN 60 may have been written to disk. The log entry was flushed before the write itself. It was not yet flushed at the time of the checkpoint, but may have been flushed later. The update at LSN 70 was flushed to disk. We know this because it's not in the dirty page table at the time of the checkpoint.

2. At the end of the Analysis phase, what transactions will be in the transaction table, and with what lastLSN and Status values? What pages will be in the dirty page table, and with what recLSN values?

Transaction Table		
Transaction	**lastLSN**	**Status**
T1	190	Aborting
T3	200	Aborting
T4	180	Aborting
T5	210	Aborting

Dirty Page Table	
PageID	**recLSN**
P1	40
P2	160
P3	90
P5	50

We also accept the following answer for the transaction table.

Transaction ID	lastLSN	Status
T1	90	Running
T3	30	Running
T4	180	Aborting
T5	160	Running

3. At which LSN in the log should redo begin? Which log records will be redone (list their LSNs)? All other log records will be skipped.

   Redo should begin at LSN 40, the smallest of the recLSNs in the dirty page table. The following log records should be redone: 40, 50, 60, 90, 110, 130, 160, 180. 30 is skipped because it precedes LSN 40. 70 is skipped because P2.recLSN = 160 > 70. Entries that are not updates are skipped. The CLR record is not skipped, nor is the LSN that it undoes.

3 Parallel Query Processing(15 pts)

We've discussed 4 kinds of parallel join in class:

- **Parallel Hash Joins**: Use hash partitioning on both relations with the same hash function, then perform a normal hash join on each machine independently.
- **Parallel Sort Merge Join**: Use range partitioning with the same ranges on both relations, then perform sort merge join on each machine independently.
- **One-sided shuffle Join**: When one relation's data is already partitioned the way we want (hash partitioned or range partitioned on a key), just partition the other relation, then run local join (using any algorithm) at every node and union results.
- **Broadcast Join**: If one relation is small, send it to all nodes that have a partition of the other relation. Do a local join at each node (using any algorithm) and union results.

In shared nothing, the machines communicate with each other solely through the network by sending data to each other. Here, network cost is referred to the amount of data sent between machines.

Now we have a Relation R that has 10,000 pages, round-robin partitioned across 4 machines (M1, M2, M3, M4). Relation S has 20 pages, all of which are only stored on M1. We want to join R and S on the condition R.col = C.col. Assume the size of each page is 1 KB.

1. Which type of join would have the lowest network cost in this scenario?
   a) Parallel Hash Joins
   b) Parallel Sort Merge Join
   c) One-sided shuffle Join

d) Broadcast Join

D

2. How many KB of data must be sent over the network to join R and S using this join method?

   3*20=60KB

3. Would the amount of data sent over the network change if R was hash or range partitioned among the 4 machines rather than round-robin partitioned using this join method?

   a) The network cost will not change under both of the partitioning methods.

   b) The network cost will change under both of the partitioning methods.

   c) The network cost will only change under range partitioning.

   d) The network cost will only change under hash partitioning.

   A

4 No SQL(25 pts)

1. For each workloads, choose whether it's better characterized as Online Transaction Processing (OLTP) or Online Analytical Processing (OLAP). Just write the answer and don't need to give a reason.

   (a) A social media site with millions of users needs to track all the "likes" and "dislikes" that each post receives.

   OLTP

   (b) An online book store needs to aggregate and analyze its users book purchases by genre over the last eight months.

   OLAP

   (c) An multiplayer online game has added updated areas to its map and now wants to assess how users behave in those areas, and how user playtime has changed as a result.

   OLAP

2. Answer the questions about scaling, you need to give a brief analysis.

   (a) A small startup realizes that its current database can't sustain their growing workloads. Given that these workloads involve lot of write but few reads, should it invest in more partitioning or more replication?

   Partitioning.
   Partitioning distributes write workloads across multiple machines, increasing throughput for writes, while replication would make writes more expensive, as it requires synchronizing updates across all replicas.

(b) A mechanical failure causes some of the startup's database machines to permanently crash, losing data in the process. If the startup wants to prevent similar losses in the future, should it invest more in partitioning or more replication?

Replication.
Replication creates multiple copies of the data across machines. If one machine fails, the data remains available from the replicas, while partitioning does not provide redundancy and cannot prevent data loss.