# Tutorial 10: Concurrency Control

## CS3402 Database Systems

# Question 1

- Consider the following arrival order of operations to the scheduler. If the scheduler adopts a serial execution method for concurrency control, define the serial schedule if the arrival order of operations remains the same as those shown in the table. If the scheduler uses strict two phase locking to schedule the operations, modify the above table to show the new schedule.

| $T_a$ | $T_b$ | $T_c$ |
|---|---|---|
| | Write(x) | |
| Read(y) | | |
| | Read(z) | |
| | | Read(x) |
| | Write(y) | |
| Write(x) | | |
| | Read(x) | |
| | Commit | |
| | | Write(z) |
| Commit | | |
| | | Commit |

# Question 1 (Answer)

- Serial schedule: $T_b$, $T_a$, $T_c$

| $T_a$ | $T_b$ | $T_c$ |
|---|---|---|
| | WL(x); Write(x) | |
| RL(y); Read(y) | | |
| | RL(z); Read(z) | |
| | | RL(x); Read(x); blocked |
| | WL(y); Write(y); blocked | |
| WL(x); Write(x); blocked | | |

- There is a deadlock; (a cycle in the wait-for-graph: $T_b \rightarrow T_a \rightarrow T_b$)

# Question 2

- Consider the following schedule at a single server system.

| $T_1$ | $T_2$ |
|---|---|
| Read(a) | |
| | Read(a) |
| Write(a) | |
| | Write(a) |

a) Add lock and unlock operations to the schedule if Conservative 2PL is adopted.

b) Add lock and unlock operations to the schedule if Strict 2PL is adopted.

c) Which one (S2PL or C2PL) will you choose for scheduling the two transactions?

# Question 2a (Answer)

| $T_1$ | $T_2$ |
|---|---|
| WriteLock(a) | |
| Read(a) | |
| Write(a) | |
| Unlock(a) | |
| | WriteLock(a) |
| | Read(a) |
| | Write(a) |
| | Unlock(a) |

# Question 2b (Answer)

| $T_1$ | $T_2$ |
|---|---|
| ReadLock(a) | |
| Read(a) | |
| | ReadLock(a) |
| | Read(a) |
| WriteLock(a); blocked | |
| | WriteLock(a); blocked |

# Question 2c (Answer)

- C2PL since it does not have the deadlock problem and the transactions are short.

# Question 3

- The following table shows the schedule for transactions $T_1$ and $T_2$ with $T_1$ having an “older” time-stamp than $T_2$.

| $T_1$ | $T_2$ |
|---|---|
| Read(a) | |
| | Read(b) |
| Write(b) | |
| | Write(a) |

a) Data items *a* and *b* are maintained by servers *X* and *Y*, respectively, and Strict Two Phase Locking is used for concurrency control. Define the wait-for-graph at each server.

b) Show the new schedule if the wait-die method is used.

c) Show the new schedule if the wound-wait method is used.

# Question 3 (Answer)

- Part a
  - a: T2 → T1
  - b: T1 → T2
- Part b
  - Wait-die: If $TS(T_i) < TS(T_j)$, $T_i$ waits else $T_i$ dies
  - Thus, Write(a) from $T_2$ will make it to abort and release the lock of data item b. Thus, the final schedule will be $T_1$ and then $T_2$.
- Part c
  - Wound-wait: If $TS(T_i) < TS(T_j)$, $T_j$ wounds else $T_i$ waits
  - When the Write(b) from $T_1$ arrives, $T_2$ is aborted. Thus, the final schedule will also be $T_1$ and then $T_2$.