Principles of Computer Systems Design

Assignment 2

Tudor Dragan
Gabriel Carp

December 2, 2014

Question 1: Serializability & Locking

Conflict-serializability is defined by equivalence to a serial schedule (no overlapping transactions) with the same transactions, such that both schedules have the same sets of respective chronologically ordered pairs of conflicting operations (same precedence relations of respective conflicting operations).

A schedule is conflict-serializable if and only if it's precedence graph has no cycles. This is a graph of nodes and vertices, where the nodes are the transaction names and the vertices are attribute collisions.

Schedule 1

Schedule 1 *is not* conflict serializable because the graph is *cyclic*:

- T_1 - T_2: read-write conflict on X.
- T_2 - T_3: write-read conflict on Z.
- T_3 - T_1: read-write conflict on Y.

Schedule 2

Schedule 2 *is* conflict serializable because the precedence graph is *acyclic*:

- T_1 - T_2: X is accessed by T_2 after T_1 has committed.
- T_2: Y is only accessed by T_2.
- T_3 - T_2: Z exclusively locked by T_3 is released prior to T_2 acquiring a shared lock.

Question 2: Optimistic Concurrency Control

Scenario 1

Because T_1 finishes before T_3 starts, the 1$^{\text{st}}$ condition holds. We have to check that the 2$^{\text{nd}}$ condition holds for T_2 and T_3, but because T_2 writes the object that T_3 reads from, this condition does not hold. Therefore T_3 hast to *rollback*.

Programming Task