Homework 5

CS143 Fall 2009

- Problem I: Consider the following schedule:involving transactions T1,T2, T3, T4 performing the following sequence of actions (w3(A) means that T3 performs write(A), c2 means that T2 commits, etc.)

  w3(A) r1(A) c3 w1(B) c1 r2(B) w2(C) r4(B) c4 c2

(a) Is it a serial schedule?
(b) Is the schedule conflict serializable? If so, what are all the equivalent serial schedules?
(c) Is the schedule recoverable?
(d) Is the schedule cascadeless? If not, can we make it cascadeless by moving a single commit operation to a different position?

Recoverability

- An unrecoverable schedule:

$$G = \begin{bmatrix} T1 & T2 \\ R(A) & \\ W(A) & \\ & R(A) \\ & W(A) \\ & Com. \\ Abort & \end{bmatrix}$$

- To be recoverable, transactions must wait and commit only after all transactions whose changes they read commit:

$$F = \begin{bmatrix} T1 & T2 \\ R(A) & \\ W(A) & \\ & R(A) \\ & W(A) \\ Com. & \\ & Com. \end{bmatrix} \quad F2 = \begin{bmatrix} T1 & T2 \\ R(A) & \\ W(A) & \\ & R(A) \\ & W(A) \\ Abort & \\ & Abort \end{bmatrix}$$

Avoid cascading aborts (rollbacks)

The following schedule for F avoids the situation F2

$$F3 = \begin{bmatrix} T1 & T2 \\ & R(A) \\ R(A) & \\ W(A) & \\ & W(A) \\ Abort & \\ & Commit \end{bmatrix}$$

A schedule is strict if for any two transactions T1, T2, if a write operation of T1 precedes a conflicting operation of T2 (either read or write), then the commit event of T1 also precedes that conflicting operation of T2.

Any strict schedule is cascadeless, and expedites recovery from failures

Realations Between Schedules

Serial ⊂ conflict-serializable ⊂ view-serializable ⊂ all schedules
Serial ⊂ strict ⊂ cascadless ⊂ recoverable ⊂ all schedules

More Problems

16.28: Is starvation still possible using deadlock-avoidance scheme?