15-415 Homework 10 Solutions

December 10, 2009

Question 1: Serializability

Q1.1 See Figures 1 and 2.

Q1.2 S1 is not conflict-serializable since the dependency graph has a cycle. S2 is conflict-serializable as the dependency graph is acylic. The order T2-T3-T1 is the only equivalent serial order.

Q1.3 S1 is not view serializable. S2 is trivially view-serializable as it is conflict serializable. The only serial order allowed is T2-T3-T1.

Question 2: Two-Phase Locking

Q2.1 S1 will be permitted by 2PL, but not by Strict 2PL. This is because when T1 tries to request an exclusive lock to write Y, T2 has not committed yet, and therefore has not released its exclusive lock on Y. It will be permitted by 2PL because then T2 can release the lock on Y immediately after it acquires a shared lock for X (prior to reading X). Then T1 will be able to acquire its exclusive lock on Y.

Q2.2 S2 will be permitted by 2Pl, but not by strict 2PL. It will not be permitted by strict 2PL because when T2 tries acquire an x-lock prior to writing Y, it will not be able to get this lock since T3 has a shared lock on Y. Since T2 makes the request before T3 commits, T2 will not be able to get the lock. However, in 2PL, after reading Z, T3 can immediately acquire an x-lock for x and release its lock for Y. So the given schedule will follow 2PL.

Question 3: Deadlock Management

Q3.1 (g= granted, b= blocked)
S1: g, b, g, b, b
S2: g, g, g, b, b

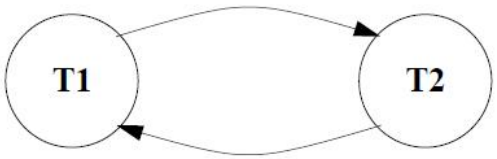

Figure 1: Precedence Graph for S1, Q1

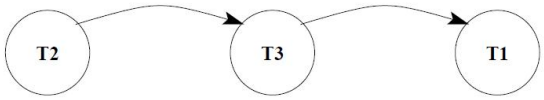

Figure 2: Precedence Graph for S2, Q1

Q3.2 S1: There is a deadlock.

$$T1 \longleftarrow T2$$
$$\searrow \quad \nearrow$$
$$T3$$

S2: No deadlock.

$$T2 \longleftarrow T3$$
$$\searrow \quad \swarrow$$
$$T1$$

Q3.3 T1:X(B) will cause a deadlock in S2.

Question 4: Hierarchical Locking

Q4.1 IS[D], IS[P], IS[P600], S[P600:4], U[P600:4], U[P600], U[P], U[D]

Q4.2 IS[D], IS[P], S[P10]...S[P80], U[P80]...U[P10], U[P], U[D]

Q4.3 SIX[D], SIX[A], X[TENPGS], U[TENPGS], U[A], U[D]

Q4.4 IX[D], X[P], U[P], X[A], U[A], U[D]

Question 5: B+ Tree Locking

Q5.1 S(A), S(B), U(A), S(C), U(B), S(D), U(C), U(D)

Q5.2 X(A), X(B), U(A), X(C), X(E), U(B), U(C), U(E)

Q5.3 X(A), X(B), U(A), X(C), X(D), U(B), U(C), U(D)

Question 6: Crash Recovery

Q6.1 The columns are shown as below:

LSN	log entry	prevLSN	undonextLSN
00	begin checkpoint	-	-
10	end checkpoint	-	-
20	T1 update Page P11	null	-
30	T1 update Page P12	20	-
40	T2 update Page P25	null	-
50	T3 update Page P34	null	-
60	T3 commit	50	-
70	T2 update Page P25	40	-
80	T2 update Page P23	70	-
90	T2 abort	80	-

Q6.2 Play back updates of T2 in reverse order. For each update, write a CLR record, restore the old value. At the end, write end log record.

Q6.3 The log will look like (only records after LSN 90 are shown):

LSN	log entry	prevLSN	undonextLSN
100	T2 CLR: UNDO LSN 80	90	70
110	T2 CLR: UNDO LSN 70	100	40
120	T2 CLR: UNDO LSN 40	110	NULL
130	T2 end	120	-

Question 7: Crash Recovery Again

All transactions T1, T2 and T3 will be redone. Only transactions T1 and T2 will be undone.