Assignment 6: Transaction Management

Dirk Van Gucht & Mo Zhou

December 8, 2008

1. Concurrency Control:

(a) What is the significance of imposing the following constraints on histories: i. serializability, ii. avoiding cascading abort, iii. strictness. You can use examples to illustrate your arguments.

(b) Consider a database with objects X and Y and assume that there are two transactions T1 and T2. Transaction T1 reads objects X and Y and then writes object X. Transaction T2 reads objects X and Y and then writes objects X and Y. Give three example histories over transactions T1 and T2 that result in a write-read conflict, a read-write conflict and a write-write conflict respectively.

Show that each serial history involving transactions T1 and T2 preserves the consistency requirement of the database.

(c) i. Give an example of two conflict-equivalent, but different, non-serializable schedules.

ii. Give an example of a serializable schedule which is conflict-equivalent with two different serial schedules.

(d) Consider the following transactions:

T1: read(A);
    read(B);
    if A = 0 then B := B+1;
    write(B).

T2: read(B);
    read(A);
    if B = 0 then A := A+1;
Let the consistency requirement be \( A = 0 \lor B = 0 \), and let \( A = B = 0 \) be the initial values.

i. Construct a history on T1 and T2 that produces a non-serializable history.

ii. Is there a non-serial history on T1 and T2 that produces a serializable history. If so, give an example.

iii. A. Add lock and unlock instructions to T1 and T2, so that they observe the two-phase locking protocol, but in such a way that interleaving between operations in T1 and T2 is still possible.

B. Can the execution of these transactions result in a deadlock? If so, give an example.

(e) Prove that if H and H’ are complete, recoverable, and view equivalent histories then they are conflict equivalent. Also prove that the requirements complete and recoverable are necessary for the truth of this statement.

(f) Define a lock point of a transaction to be any moment at which it owns all of its locks; that is, it is a moment after it has performed its last lock operation and before it has released any lock. Using serializability theory, prove that for every history H produced by a 2PL scheduler there is an equivalent serial history in which transactions are in the order of their locked points in H.

(g) Exercise 17.2 in Cow Book.

2. Recovery

(a) Exercise 18.5 in Cow Book.

(b) In logical logging, suppose each update record describes an operation that is applied to at most one data item. Suppose we implement undo and redo procedures for all operations so that for each log record LR on data item X, undo(LR) has no effect if X does not include LR’s update, and redo(LR) has no effect if X already includes LR’s update. Does Restart in the partial data item logging algorithm work correctly on a log with this structure? That is, assuming fuzzy checkpointing, is it correct to undo
all uncommitted updates during a backward scan of the log, and then redo all committed updates during a forward scan? If so, argue the correctness of the algorithm. If not, explain why.

(c) Specify a recovery strategy which would never require undoing or redoing side-effects (both undoing and redoing are not required). How would such a recovery strategy be implemented?

(d) Specify a recovery strategy which would never require re-doing side-effects of write operations caused by committed transactions (only redoing is not required while undoing is required). How would such a recovery strategy be implemented?

(e) Specify a recovery strategy which would never require undoing (only undoing is not required while redoing is required). How would such a recovery strategy be implemented?

Hints: Reading chapter 6 of the online book is helpful.

(f) Suppose that there is a database system that never fails. Is a concurrency control scheduler required for this system? Is a recovery manager required for this system? Don’t just answer “yes” or “no”. Justify your answers. (Hint: consider this question in the context of the ACID requirements associated with transaction management.)