CSE 444 Practice Problems

Transactions: Recovery

1. **Transactions: recovery**

   (a) What are STEAL and NO-STEAL policies?
   Solution:

   - **STEAL:** Non-committed updates can overwrite committed values on disk. Requires undo on recovery.
   - **NO-STEAL:** Non-committed updates cannot overwrite committed values on disk.

   (b) What are FORCE and NO-FORCE policies?
   Solution:

   - **FORCE:** All updates made by a transaction must be written to disk before the transaction is allowed to commit.
   - **NO-FORCE:** Updates made by a transaction are not forced to disk before the transaction commits. The buffer manager may write updates to disk after the transaction committed. Requires redo on recovery.

2. **UNDO Logging**

Consider the content of the following **undo log**:

LSN1	<START T1>
LSN2	<T1 X 5>
LSN3	<START T2>
LSN4	<T1 Y 7>
LSN5	<T2 X 9>
LSN6	<START T3>
LSN7	<T3 Z 11>
LSN8	<COMMIT T1>
LSN9	<START CKPT(T2,T3)>
LSN10	<T2 X 13>
LSN11	<T3 Y 15>
	*C*R*A*S*H*

- Show how far back in the recovery manager needs to read the log. Write below the earliest LSN that the recovery manager reads.

  Solution:
  LSN3

- Show below the actions of the recovery manager during recovery:

  Solution:

$$\begin{aligned} Y &= 15 \\ X &= 13 \\ Z &= 11 \\ X &= 9 \end{aligned}$$

- What is the value of X at the end of the recovery ?

  Solution:
  9

3. **REDO Logging**

After a system crash, the **redo-log** using non-quiescent checkpointing contains the following data:

< START T1 >
< T1, A, 10 >
< START T2 >
< T2, B, 5 >
< T1, C, 7 >
< START T3 >
< T3, D, 12 >
< COMMIT T1 >
< START CKPT ???? >
< START T4 >
< T2, E, 5 >
< COMMIT T2 >
< T3, F, 1 >
< T4, G, 15 >
< END CKPT >
< COMMIT T3 >
< START T5 >
< T5, H, 3 >
< START CKPT ???? >
< COMMIT T5 >

(a) What are the correct values of the two <START CKPT ????> records? You have to provide two correct values for the two ????s.

Solution:

First START CKPT: < START CKPT (T2, T3) >

Second START CKPT: < START CKPT (T4, T5) >

(b) Indicate and **explain** what fragment of the log the recovery manager needs to read.

Solution:

Since the second START CKTP does not have an associated END CKPT, we cannot be sure that committed transactions prior to the start of this checkpoint had their changes written to disk. Thus, we must search for the previous checkpoint. In the previous START CKPT, T2 and T3 were the two active transactions. Both transactions committed and must thus be redone. T2 was the first one to start. The recovery manager must thus read the log record starting from < START T2 > and must read until the end of the log file.

(c) Assuming that the two < START CKPT ??? > records are correctly stored in the log, according to your answer above, show which elements are recovered by the redo recovery manager and compute their values after recovery.

Solution:

We must redo the changes made by all *committed* transactions there were either active during the first START CKPT or that started after that point. T2 and T3 were active during the first START CKPT and comitted. T4 and T5 started after the checkpoint but only T5 committed. We must thus redo the changes by T2, T3, and T5. Elements B, D, E, F, and H are thus recovered. Their values after recovery are as follows:

- B=5
- D=12
- E=5
- F=1
- H=3

4. **Aries**

(a) In the ARIES method, assuming NO checkpoints have been used, explain what happens during the ANALYSIS pass of recovery. Your answer should indicate at least (1) what part of the log the system reads and in what direction and (2) what data structures the system rebuilds.

Solution:
In the absence of checkpoints, the analysis pass reads the log from the beginning to the end. It rebuilds the Dirty Pages Table and the Transactions Table to determine the state of the system as of the time of the crash. It rebuilds these data structures by updating them according to the log records that it encounters during the forward scan.

(b) After the analysis pass, the protocol performs a REDO pass. Explain (1) where does REDO start reading the log and in what direction it reads the log, (2) what happens during the REDO pass.

Solution:
The REDO pass begins at the log record whose LSN corresponds to the earliest recoveryLSN of all the entries in the Dirty Page Table. From that point, the REDO pass redoes the updates of all transactions (committed or otherwise). At the end of this pass, the database is in the same state as it was right before the crash.

(c) The last pass during recovery is the UNDO pass. Explain (1) where does the UNDO start reading the log and in what direction it reads the log, (2) what happens during the UNDO pass.

Solution:
The UNDO pass scans backward from the end of the log. The pass undoes the updates by all transactions that had not committed by the time of the crash. These transactions can be found in the Transactions Table rebuilt during the analysis pass. All updates are undone unconditionally (since the REDO pass ensured that all logged updates have been applied to affected pages). For each update that is undone, the undo operation is logged with a Compensation Log Record (CLR).