ACS Theory Assignment 2

This assignment is due via Absalon on December 17, 23:59. While individual hand-ins will be accepted, we strongly recommend that this assignment be solved in groups of maximum two students. NOTE: The KU IDs of ALL group members MUST be stated on a separate `group.txt` file to ensure all group members get feedback and get the assignment accounted for in Absalon.

A well-formed solution to this assignment should include a PDF file with answers to all theory questions. Note that all homework assignments have to be submitted via Absalon in electronic format. It is your responsibility to make sure in time that the upload of your files succeeds. While it is allowed to submit scanned PDF files of your homework assignments, we strongly suggest composing the assignment using a text editor or LaTeX and creating a PDF file for submission. Email or paper submissions will not be accepted.

Learning Goals

This assignment targets the following learning goals:

- Analyze the relationship of serializability definitions and concurrency control protocols to the schedules they possibly generate.
- Discuss basic concepts in recovery for durability of a two-level memory under a restricted set of fail-stop failures.
- Explain in detail the functioning of the ARIES algorithm in recovery scenarios.

Question 1: Concurrency Control Concepts

For each of the following items, you are asked to exhibit a schedule fulfilling the requirements stated, as well as to justify why the schedule fulfills the requirements:

1. Show a schedule that is view-serializable, but not conflict-serializable. Explain.
2. Show a schedule that is conflict-serializable, but could not have been generated by a two-phase locking (2PL) scheduler. Explain.
3. Show a schedule that could be generated by a 2PL scheduler, but not by a strict two-phase locking (S2PL) scheduler. Explain.
4. Show a schedule that could be generated by a S2PL scheduler, but not by a conservative strict two-phase locking (CS2PL) scheduler. Explain.

Question 2: Recovery Concepts

Regarding recovery techniques, answer the following questions:

1. In a system implementing force and no-steal, is it necessary to implement a scheme for redo? What about a scheme for undo? Explain why.

2. What is the difference between nonvolatile and stable storage? What types of failures are survived by each type of storage?

3. In a system that implements Write-Ahead Logging, which are the two situations in which the log tail must be forced to stable storage? Explain why log forces are necessary in these situations and argue why they are sufficient for durability.

Question 3: More Concurrency Control

Consider the following two transaction schedules with time increasing from left to right. C indicates commit. Transactions are denoted Ta, Tb, and Tc so as to avoid implying any pre-determined relative order.

```
Schedule 1
Ta: R(Z)                              W(Y)  C
Tb:      R(Y)  W(Y)        C
Tc:                R(Z)      W(Y)  C

Schedule 2
Ta: R(X)                                        W(Y)  C
Tb:      R(Z) R(Y) W(Z) W(Y)  C
Tc:                                R(Z) W(Z)  C
```

Now answer the following questions:

1. Could Schedule 1 have been generated by a strict two-phase locking (S2PL) scheduler? Explain why or why not.

2. Could Schedule 1 have been generated by a two-phase locking (2PL) scheduler? Explain why or why not.

3. Could Schedule 2 have been genereted by a conservative-strict two-phase locking (CS2PL) scheduler? Explain why or why not.

4. Could Schedule 2 have been generated by a Kung-Robinson optimistic concurrency control (OCC) scheduler? Explain why or why not. NOTE: For this item, you may assume that the read and write operations shown represent operations against the shared database, not the transaction's private copies.

Question 4: ARIES

Consider the recovery scenario described in the following, in which we use the ARIES recovery algorithm. At the beginning of time, there are no transactions active in the system and no dirty pages. A checkpoint is taken. After that, three transactions, T1, T2, and T3, enter the system and perform various operations. The detailed log follows:

```
LOG

LSN     PREV_LSN     XACT_ID     TYPE         PAGE_ID
---     --------     -------     -----        --------
1       -            -           begin CKPT   -
2       -            -           end CKPT     -
3       NULL         T1          update       P2
4       3            T1          update       P1
5       NULL         T2          update       P5
6       NULL         T3          update       P3
7       6            T3          commit       -
8       5            T2          update       P5
9       8            T2          update       P3
10      6            T3          end          -
-------- XXXXXXX ------  CRASH!  ------ XXXXXXX --------
```

After LSN 10, the system crashes. Apply the ARIES recovery algorithm to the scenario above. Show:

1. the state of the transaction and dirty page tables after the analysis phase of recovery;
2. the sets of winner and loser transactions;
3. the values for the LSNs where the redo phase starts and where the undo phase ends;
4. the set of log records that may cause pages to be rewritten during the redo phase;
5. the set of log records undone during the undo phase;
6. the contents of the log after the recovery procedure completes.

For each of the items above, justify your answer.

Question 5: More ARIES

In the recovery scenario described in the following, we ask you to explain the functioning and justification behind the ARIES recovery algorithm. At the beginning of time, there are no transactions active in the system and no dirty pages. A checkpoint is taken. After that, four transactions, T1, T2, T3, and T4, enter the system and perform various operations. The detailed log follows:

```
LOG

LSN     PREV_LSN     XACT_ID     TYPE         PAGE_ID UNDONEXTLSN
---     --------     -------     ----         ------- -----------
1       -            -           begin CKPT   -       -
2       -            -           end CKPT     -       -
3       NULL         T1          update       P3      -
4       3           T1           update       P1      -
5       NULL         T2          update       P1      -
6       NULL         T3          update       P3      -
7       5            T2          update       P2      -
8       NULL         T4          update       P2      -
9       4            T1          update       P4      -
10      8            T4          commit       -       -
11      9            T1          abort        -       -
12      7            T2          abort        -       -
13      12           T2          CLR          P2      A
14      10           T4          end          -       -
15      11           T1          CLR          P4      B
16      15           T1          CLR          P1      C
17      13           T2          CLR          P1      D
18      16           T1          CLR          P3      E
19      18           T1          end          -       -
-------- XXXXXXX ------  CRASH!  ------ XXXXXXX --------
```

A crash occurs at the end of the log as shown. Considering again that the system employs the ARIES recovery algorithm, answer the following questions and justify each of your answers.

1. In the log, the undonextLSN values for several CLRs are represented by the letters A-E. For each letter, state the value of the corresponding undonextLSN and why it has the given value.
2. What are the states of the transaction table and of the dirty page table after the analysis phase of recovery? Explain why.

3. Show the additional contents of the log after the recovery procedure completes. Provide a brief explanation for why any new log records shown need to be added.

4. In the system shown, let us assume that the database is memory-resident, i.e., upon start-up all contents of nonvolatile storage are loaded into volatile storage (which is large enough to cache everything). Thus, no page replacements are ever necessary during normal operation. Furthermore, assume plenty of log buffer space is available and no background page writes are performed. Under these conditions, answer the following:

   (a) Which log records could trigger writes to stable storage during normal operation in the scenario above? Why?

   (b) Following the ARIES log record structure introduced in the course for update records, is there any information that would not need to be written to stable storage in such a scenario? If so, explain which information and why. Otherwise, justify why all information in these records needs to reach stable storage.