CS-410: Database Engineering
Low-stakes Assignment: Transaction Processing

Nick Alexander

April 26, 2015

1 Purpose

The purpose of this assignment is to understand database transaction processing concepts. You need to read chapter 21 in the textbook and refer to the associated lecture notes.

2 Submission instructions and grading rubric

Use the LaTeX template to generate your solution in PDF format.

3 Questions

1. Distinguish between parallel execution and interleaved execution.

   In systems that contain multiple hardware Central Processing Units (CPUs), the system is capable of executing instructions simultaneously on the separate CPUs. As an analogy, parallel execution is similar to having 2 people writing sections of a paper at the same time; two jobs can be processed simultaneously. If there is only one CPU, though, the system is capable of interleaved processing. When a system interleaves jobs, it works on job 1 for a small amount of time and then switches execution over to job 2 for another amount of time, then switches back to job 1 and so forth until all jobs are complete. The purpose of interleaving is to provide a means of concurrency without needing multiple physical CPUs.

2. What is a transaction?

   A transaction is the a single unit of database operations. A transaction may contain a number of contained operations, such as inserts, updates, and deletes, but altogether the set of contained operations forms one concrete, meaningful result.

3. Why do you need concurrency control?

   Just like any human task, when there are too many people working on the same job at the same time, the job becomes harder. The same can be said for computer and database systems. When multiple people are attempting to access the same information at the exact same time, it is possible that one user will get a different result than another which reduces the integrity of the database as a whole. With concurrency control, the DBMS is able to ensure that the data is always in a valid state.

4. Describe the lost update problem.

   When a system is interleaved, it is possible that two transactions can cause an update to be lost. Assume we have two transactions t1 and t2. First, t1 reads a value X and performs some operation on it, such as an add or subtract, forming result X'. The system then switches execution to t2 which reads value X as well and updates it to X". Execution is then switched to t1 which writes the value of X' and performs some other trivial operations. Finally, execution is given back to t2 which then writes the value of X". When t2 writes the value of X", the value of X' was lost because it was immediately overwritten.

5. Describe the temporary update (or dirty read) problem.

   The dirty update problem occurs when a transaction fails but another transaction has read data written by the failed transaction. Assume there are two transactions t1 and t2. When t1 starts execution, it reads a value from the database X and performs some operations to form result X'. The transaction then writes the value of X' to the database. Operation is then switched to t2 which reads the value of X' and performs some operation to form result X" and continues to perform some trivial operations until execution is pass back to t1. When t1 retains execution, it fails for some reason or another and must rollback. The problem, though, is that t2 has already read the value of X to be X' meaning that t2 executed in a bad state.

6. Describe the incorrect summary problem.

   When a summary operation is performed by a transaction, another transaction may gain execution and change the values of multiple items in the database that have already been summed. When this happens, the summary operation is going to return a bad summary. This problem is known as the incorrect summary problem.

7. Describe the unrepeatable read problem.

   When a transaction reads a value X from the database, another transaction changes the value of X, and then the first transaction reads the value of X again, it will get a different result for X each time. This problem is known as the unrepeatable read problem.

8. List and describe six different reasons for a transaction failure.

   There are 6 major reasons that a transaction may fail:

   - Computer crash - Hardware or software on a computer fails.
   - Transaction or System error - A parameter or programming error causes a transaction to fail such as buffer overflow.
   - Exception Condition - Data cannot be found or conditions of the transaction are not met, such as having sufficient account funds.

- Concurrency Control - A transaction may be terminated by the concurrency controller if the transaction is causing deadlock or some other concurrency-related problem.
- Disk failure - A hard disk fails meaning data cannot be read or written properly.
- Physical Problem - Natural disasters, network disconnection, power failure, etc..

9. Define the following terms: active state, partially committed state, committed state, failed state, and terminated state.

   A transaction is in the active state if it has been given execution time and can perform read and write operations. When the transaction is finished, it must store the result in a temporary area, like the log, to protect against failure. When the transaction does this, it is said to be in the partially commit ed state. If the transaction successfully passes the partially committed state, the transaction is in the committed state. Otherwise, the transaction is in the failed state. The transaction can also enter the failed state if something goes wrong during execution in the active state.

10. List and describe the types of entries that go into system log.

    Typically, there are around 5 types of log entries found in the database log:

    - start_transaction, T - Specifies the start of a transaction T.
    - write_item, T, X, old, new - Specifies that the value of X has been updated in the database by transaction T from old to new.
    - read_item, T, X - Specifies that transaction T has read the value of X from the database.
    - commit, T - Specifies that the transaction T has been committed
    - abort, T - Specifies that the transaction T has been aborted

11. What a commit point of a transaction?

    A transaction reaches the commit point when all operations of the transaction have been completed and all the effects of the operations have been written to the system log.

12. Describe ACID properties of a transaction.

    - Atomicity - A transaction is an all-or-nothing atomic unit of processing.
    - Consistency - A transaction should take the database from a valid state into a new, valid state.
    - Isolation - A transaction should execute without interference from other concurrent transactions.
    - Durability - Changes by a transaction should persist even through failure.

13. Define a schedule S of n transactions.

    A schedule S is an ordering of operations of n transactions. While operations may be interleaved, a transaction T must appear in S in the order that T occurs.

14. A schedule S of n transactions is a complete schedule under what conditions?

    A schedule S is complete if the operations in S are exactly the same as the operations in the transactions T1 to TN; for any pair of operations, their order in S is the same as their order in T; and for any two conflicts, one of the two operations must occur before the other in S.

15. Define the concept of committed projection $C(S)$ of schedule S.

    The committed projection of S is the set of operations that belong to committed transactions in T.

16. Distinguish between recoverable and nonrecoverable schedules.

    A nonrecoverable schedule is a schedule that satisfies the requirement that any transaction T that is committed should never need to be rolled back. A recoverable schedule is defined as a schedule wherein no transaction T commits until all the transactions T' that write to a value X that T reads have been committed.

17. What is cascading rollback? Illustrate with an example.

    A cascading rollback is when a failed transaction T1 is rolled back but the operations in T1 affected an operation in transaction T2 so therefore T2 must also be rolled back.

    For example, assume there are 2 ATM machines accessing the same account. Transaction T1 reads the account balance and withdraws 200 dollars. At that moment, transaction T2 reads the account balance as balance minus 200 and performs operations. Later, t1 fails due to power failure and must be rolled back. Because T2 had read from the changed account balance, T2 must also roll back.

18. Describe strict schedule with an example.

    In a strict schedule, transactions cannot read or write an item X from the database until the last transaction that wrote to X has been committed.

    Furthering the ATM example, transaction T2 would not have been allowed to start until T1 had completed and been committed. In the same scenario as above, transaction T2 would not have started because T1 failed.

19. Compare and contrast serial, non-serial, and conflict-serializable schedules.

    A schedule S is serial for every operation in all transactions in the set T are executed consecutively in S. If the operations are not consecutive, then S is non-serial.

20. When are two schedules conflict equivalent?

    Two schedules are conflict equivalent when the order of conflicting operations is the same in both schedules.

21. Describe how you test for conflict serializability of a schedule S.

    To test for conflict serializability, create a precedence graph. If there is a cycle in the graph, then S is not conflict serializable.

22. Define view equivalence of two schedules S_1 and S_2.

    Two schedules are considered to be view equivalent if the same set of transactions, that contain the same operations, are in both S1 and S2; for any operation that reads the value of X after X has been written in S1, the value of X must be the same in S2; finally, if an operation is the last operation to write an item Y in S, the same transaction must be the last to write Y in S2.

23. Define view serializability.

    A schedule is said to be view serializable if it is view equivalent to a serial schedule.

4 Rubric

Use the following rubric to evaluate your response to this assignment.

Perf Level Trait	*Outstanding*	*Good*	*Fair*	*Poor*
Depth of Analysis	Skillfully paraphrases text from the book and other sources using precise and technical terminology. Provides insightful observations and comments.	Skillfully paraphrases text from the book and other sources using precise and technical terminology. Provides no insightful observations and comments.	Paraphrases text from the book using non-technical technical terminology.	Reproduces text from the book verbatim.
Completeness	*Thorough and concise answers* to all questions in the assignment are provided.	*Thorough and concise answers* to less than 75% of the questions in the assignment are provided	Answers to less than 50% of the questions in the assignment are provided. Some answers are verbose.	Answers to less than 25% of the questions in the assignment are provided. Some answers are verbose, ambiguous, and factually incorrect.
Diction	Sentences are short and written in active voice. Writing is precise and concise.	Sentences are short but written in passive voice. Writing is precise and concise.	Sentences are long and hard to read. Sentences are prone to multiple interpretations due to ambiguity.	Easily noticeable grammatical errors and stylistic issues. Difficult to understand writing.

5 Self-assessment

Use the following table and the rubric of section 4 to score your solution. Circle the appropriate number in each row. For example, to circle 20, use the LaTeX markup code `\circled{20}`, which produces ⑳.

Perf Level	*Outstanding*	*Good*	*Fair*	*Poor*
Trait				
Depth of Analysis	[✓ 50]	40	30	20
Completeness	[✓ 30]	25	20	15
Diction	[✓ 20]	15	10	5