Database Recovery

By,

Harshil T. Kanakia

Database Recovery

The techniques used to recover the lost data due to system crash, transaction errors, viruses, incorrect commands execution etc. are database recovery techniques.

Recovery Techniques

Recovery techniques are heavily dependent upon the existence of a special file known as a system log. It contains information about the start and end of each transaction and any updates which occur in the transaction. The log keeps track of all transaction operations that affect the values of database items. This information is needed to recover from transaction failure.

Information needed to recover from transaction failure

- **The log is kept on disk start_transaction(T)**: This log entry records that transaction T starts the execution.
- **read_item(T, X)**: This log entry records that transaction T reads the value of database item X.
- **write_item(T, X, old_value, new_value)**: This log entry records that transaction T changes the value of the database item X from old_value to new_value.

Information needed to recover from transaction failure (Contd..)

- **commit(T)**: This log entry records that transaction T has completed all accesses to the database successfully and its effect can be committed (recorded permanently) to the database.
- **abort(T)**: This records that transaction T has been aborted.: This records that transaction T has been aborted.
- **checkpoint**: Checkpoint declares a point before which the DBMS was in consistent state, and all the transactions were committed.

Recovery Technique based on **Undoing**

If a transaction crashes, then the recovery manager may undo transactions i.e. reverse the operations of a transaction. This involves examining a transaction for the log entry write_item(T, x, old_value, new_value) and setting the value of item x in the database to old-value.

Recovery Technique based on **Deffered Update**

- This technique does not physically update the database on disk until a transaction has reached its commit point.
- Before reaching commit, all transaction updates are recorded in the local transaction workspace.
- If a transaction fails before reaching its commit point, it will not have changed the database in any way so UNDO is not needed.
- It may be necessary to REDO the effect of the operations that are recorded in the local transaction workspace, because their effect may not yet have been written in the database.
- Hence, a deferred update is also known as the **No-undo/redo algorithm**

Example of **Deffered Update**

Step	Details of Log
1	<T0 Start>
2	<T0, A, 950>
3	<T0, B, 2050>
4	<T1 Start>
5	<T1, C, 200>
6	<T1 Commit>

- If the crash occurs just after step 6 and the recovery of the system is successfully completed.
- Then Redo T1 and No action for T0 as partial commit of T0 is not done.

Recovery Technique based on **Immediate Update**

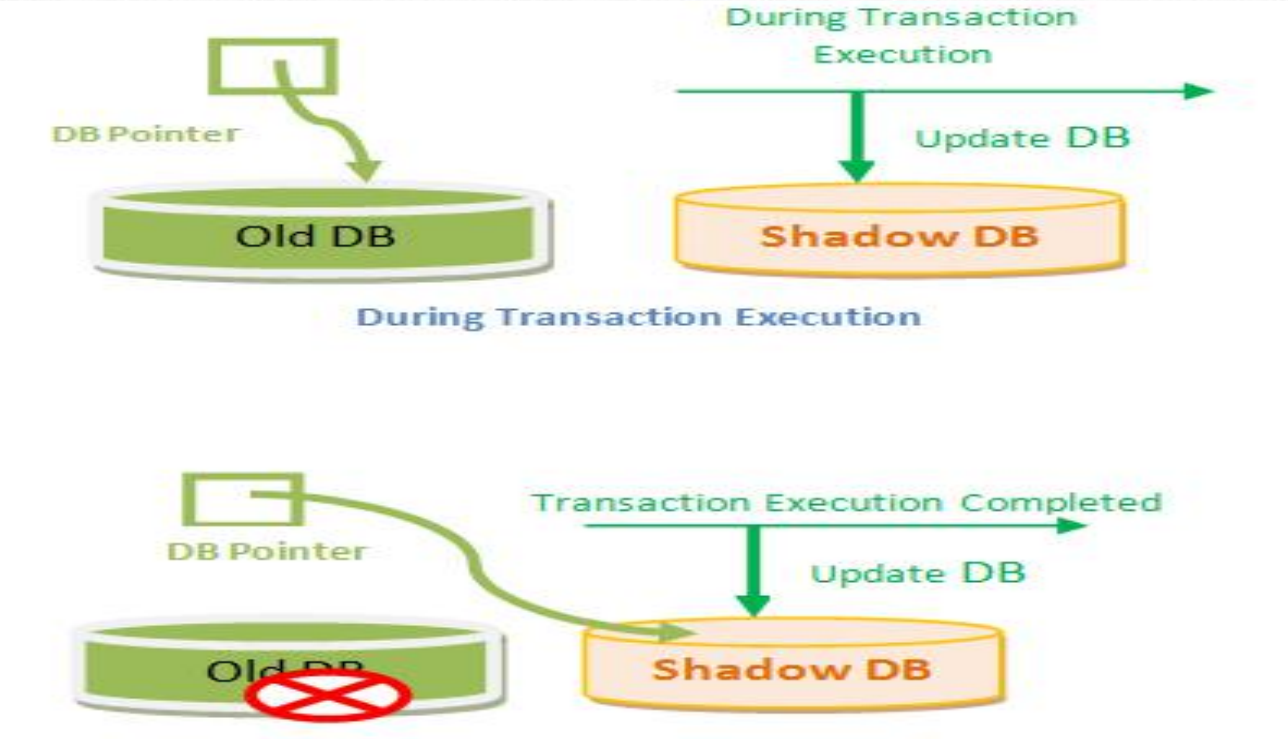
- In the immediate update, the database may be updated by some operations of a transaction before the transaction reaches its commit point.
- However, these operations are recorded in a log on disk before they are applied to the database, making recovery still possible.
- If a transaction fails to reach its commit point, the effect of its operation must be undone i.e. the transaction must be rolled back hence we require both undo and redo.
- This technique is known as **undo/redo algorithm**.

Example of **Immediate Update**

Step	Details of Log
1	<T0 Start>
2	<T0, A, 100, 200>
3	<T0, A, 200, 300>
4	<T1 Start>
5	<T0 Commit>
6	<T1, B, 500, 400>
7	<T1 Commit>
8	<T2 Start>
9	<T2, A, 300, 1500>

- If a crash occurs just after step 9
- undo(T2) then redo(T0) then redo(T1)

Shadow Paging



Shadow Paging

- Shadow paging is a technique for providing atomicity and durability.
- A database pointer will be always pointing to the consistent copy of the database, and copy of the database is used by transactions to update. Once all the transactions are complete, the DB pointer is modified to point to new copy of DB, and old copy is deleted.
- If there is any failure during the transaction, the pointer will be still pointing to old copy of database, and shadow database will be deleted. If the transactions are complete then the pointer is changed to point to shadow DB, and old DB is deleted.

ARIES Recovery Algorithm

- Algorithm for Recovery and Isolation Exploiting Semantics (ARIES).
- Algorithm for Recovery and Isolation Exploiting Semantics (ARIES) is based on the Write Ahead Log (WAL) protocol.

ARIES Recovery Algorithm (Contd..)

- Every update operation writes a log record which is one of the following :

1. **Undo-only log record**: Only the before image is logged. Thus, an undo operation can be done to retrieve the old data.

2. **Redo-only log record**: Only the after image is logged. Thus, a redo operation can be attempted.

3. **Undo-redo log record**: Both before images and after images are logged.

End