Recovery Systems

Log based Recovery

A log file is a listing of all operations which are performed on a database. This is the most widely used system for recording data modifications to the database. Typically a log file has a sequence of log records. Generally a log record consists of the following fields:

1. Transaction Identifier:	Unique identifier for the transaction.
2. Data item Identifier:	Unique identifier for the data item written to.
3. Old Value:	Value of Data Prior to writing.
4. New Value:	Value of data after writing.

Log entries take the following shape:

- < t_i Start> transaction starts.
- <t_i, x_j, V_1, V_2> transaction t_i wrote to data item x_j which had value V_1 before and which was changed to V_2 after the transaction.
- < t_i Commit> OR
- < t_i Abort>

For a log to be able to provide the base for a successful recovery, it needs to be protected from system and disk failures. It should reside on a stable storage.

Deferred Database Modification

This technique ensures transaction atomicity by recording all data modifications to the log but the writing to the database is deferred till the transaction becomes partially committed(All the operations are performed and commit is pending). In this method if the transaction aborts before it finishes then its entries in the log are ignored. If the transaction has successfully entered partial committed stage then the log is written to a stable storage and then updates to the database will be made.

e.g. Let's take two transactions and see how they are reflected in the log.

t_0	t_1
read(A);	read(A);
A:=A+1;	A:=A+10;
write(A);	write(A);
read(B);	
B:=B+2;	
write(B);	

The Log:

< t_0 Start>
< t_0 , A, 100,101>
< t_0 , B,200,202>
< t_0 , Commit>
< t_1 , Start>
< t_1 , A,101,110>
< t_1 , Commit>

Using the log, the system can handle any failures that results in the loss of information on volatile storage. Once the log is decided to be replayed, the redo operation is idempotent(executing several times produce the same result) in nature.

Immediate Database Modification

The immediate -modification technique allows modification to be made to a database while the transaction is still in the active stage. Data modifications done by active transactions are called uncommitted modifications. In the case of a transaction failure the system must rely on logs for recovery.

- Before any transaction (t_i) does any operation it is first entered into the log.
- Since information in Log is very essential for construction of the database in the case of a failure, the log is written on a stable storage even before anything is recorded into the database.
- Using the log, any undo or redo operation is performed on the database idempotently.

Checkpoints

When a system fails, we must consult the log to determine those transactions that need to be redone and those that need to be undone. This involves making investigations into the log (making a search over the entire log) to determine the point from which we have to undo/redo operations. There are two problems here:

1. The search process is quite time consuming
2. Depending on the algorithms used to determine a logical point to start the process, the recovery may take too long.

To reduce such an overhead, checkpoints are used. In this method, during execution the log will be maintained as usual and additionally the system would periodically perform checkpoints. Performing checkpoints require the following operations:

1. All log records currently in the main memory is transferred to stable storage.
2. All modified blocks is stored in the disk permanently.
3. The log record itself is put in the stable storage.
4. No transaction is allowed to perform any update action (writing to a buffer block or writing a log record) while a checkpoint is in progress.

The presence of a checkpoint in the log allows the system to streamline the recovery process.

- When a failure occurs, rollback will be from the point of failure till the last checkpoint.
- Further, for any transaction (t_i) which has a commit but not stored in database, all operations will be replayed.

Recovery with checkpoints if transactions are running during a checkpoint

The situation is a bit more complex if transactions are running when the checkpoint is made. In such a case the checkpoint log will have entry like

<checkpoint L>

Here L is the list of transactions active during the checkpoint. Here also no update to the database is allowed during checkpoint. If updates has to be permitted during checkpoint, a fuzzy checkpoint system is used.

Fuzzy Checkpoint:

The checkpoint allows updates to the database after the checkpoint record has been written to disk and before the actual database is modified by writing buffer blocks to it.