log Based Recovery

A computer system, like any other device is subject to failure from variety of causes. An integral part of database system is a recovery scheme that can restore the database to the consistent state that existed before the failure.

The most widely used structure for recording database modification is the log. The log is the sequence of log records, recording all the update activities in the database.

There are several types of log records. An update log record describe a single database write.

An update log record represented as <Ti, Xj, V1, V2>

Transaction identifies. (Ti) → Unique identifies of transaction that perform write operation.

Data Item (Xj) → which is unique identifies of the data item written. It is typically a disk location.

Old Value (V1) → which is the value of the data item prior to write.

New value (V2) → which is the value that the data item will have after the write.

Along with this, other special log exists related to transaction as,

<Ti start> Transaction Ti has started
<Ti commit> Transaction Ti has committed.
<Ti abort> Transaction Ti has aborted.

Transaction creates a log record prior to modifying the database. ~~From~~ From the logs, the system has available both old values prior to modification and new values that is written after modification.

The recovery scheme use two operations :-

1. Undo :- Using the log record set the data item specified in the log record to old value.
2. Redo :- using the log record sets the data item specified in the log record to the new value

For log modifications two techniques are used.

1) Deferred Modification technique :- If the transact^n does not modify the database until it has committed, it is called as ~~[illegible]~~ deferred modification.

2) Immediate Modification :- If the database modification occure while the transaction is still active, this technique is called immediate modification.

Using the log to redo and undo transactions ⇒ To understand which transactions need to be undone and redone consider the following log record.

{A=1000 , B=2000}.

<T_0 start>	<T_0 start>	< T_0 start>
<T_0, A, 1000, 950>	<T_0, A, 1000, 950>	<T_0, A, 1000, 950>
<T_0, B, 2000, 2050>	<T_0, B, 2000, 2050>	<T_0, B, 2000, 2050>
	<T_0 commit>	<T_0 commit>
(a)	<T_1 start>	<T_1 start>
	<T_1, C, 700, 600>	<T_1, C, 700, 600>
		<T_1 commit>
	(b)	
		(c)

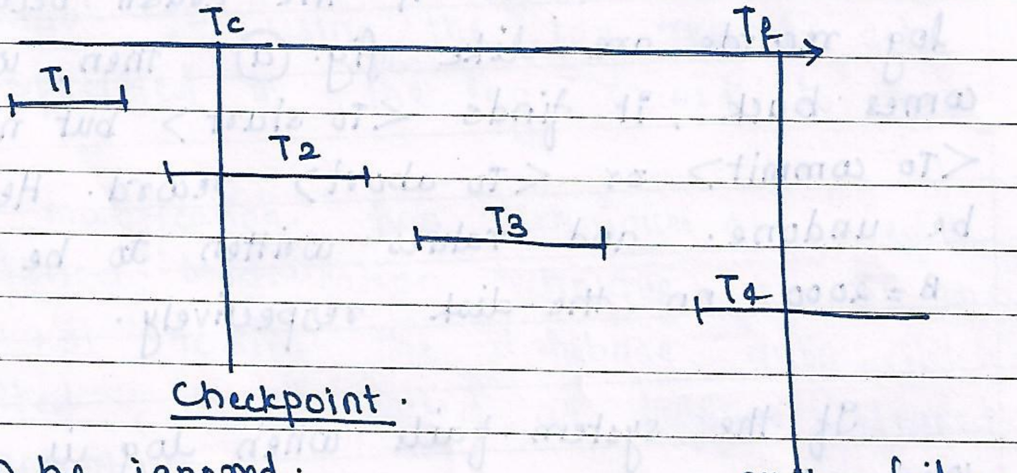
Now let us assume if the crash occures and log records are like fig. (a) then, when system comes back, it finds <T_0 start> but no corresponding <T_0 commit> or <T_0 abort> record. Hence T_0 must be undone. and values written to be A=1000 and B = 2000. on the disk respectively.

If the system fails when log is like figure (b) when the system comes back, the redo of T_0 must perform as <T_0 commit> and <T_0 start> both are present. But for <T_1 start> no corresponding <T_1 commit> or <T_1 abort> hence T_1 must be undone. and values of A = 950, B = 2050 and C = 700. will write on disk.

Finally if system fails when log records are like figure (c) then, T0 and T1 must be redone since records <T0 start> <T0 commit> and <T1 start> and <T1 commit> are present in the log.

Log based recovery scheme uses the checkpoints. When the system fails, the entire log must be ~~scanned~~ searched to determine information. But searching entire log is time consuming process. And also most of the transaction, that needs to be redone have already written their updates into the database hence it is additional overhead. Hence to avoid this checkpoints are introduced.

Example of checkpoints is,



- T1 can be ignored.
- T2 and T3 redone.
- T4 undone.

Recovery Algorithm :-

Recovery algorithm uses log records for recovery from transaction failure and a combination of the most recent checkpoint and log records to recover from a system crash.

Transaction Rollback :-

If $\langle T_i, X_j, V_1, V_2 \rangle$ is found in log & failure occurs then,

$\langle T_i, X_j, V_1 \rangle$ is written to log and

$\langle T_i \text{ abort} \rangle$ is written.

And recovery is done with the redo and undo phase depends upon the values of the log.