1 ARIES Recovery Algorithm

Briefly answer the following questions:

1. How does the recovery manager ensure atomicity of transactions? How does it ensure durability?
2. What is the difference between stable storage and disk?
3. Explain the WAL protocol.
4. Describe the steal and no-force policies.

Solution

1. The Recovery Manager ensures atomicity of transactions by undoing the actions of transactions that do not commit. It ensures durability by making sure that all actions of committed transactions survive system crashes and media failures.
2. Stable storage is guaranteed (with very high probability) to survive crashes and media failures. A disk might get corrupted or fail but the stable storage is still expected to retain whatever is stored in it. One of the ways of achieving stable storage is to store the information in a set of disks rather than in a single disk with some information duplicated so that the information is available even if one or two of the disks fail.
3. WAL Protocol: Whenever a change is made to a database object, the change is first recorded in the log and the log is written to stable storage before the change is written to disk.
4. If a steal policy is in effect, the changes made to an object in the buffer pool by a transaction can be written to disk before the transaction commits. This might be because some other transaction might "steal" the buffer page presently occupied by an uncommitted transaction. A no-force policy is in effect if, when a transaction commits, we need not ensure that all the changes it has made to objects in the buffer pool are immediately forced to disk.

2 ARIES Analysis, Redo, and Undo Phases

Briefly answer the following questions:

1. What are the roles of the Analysis, Redo, and Undo phases in ARIES?
2. Consider the execution shown in Figure 1.
   (a) What is done during Analysis? (Be precise about the points at which Analysis begins and ends and describe the contents of any tables constructed in this phase.)
   (b) What is done during Redo? (Be precise about the points at which Redo begins and ends.)
   (c) What is done during Undo? (Be precise about the points at which Undo begins and ends.)

Solution

1. The Analysis phase starts with the most recent begin checkpoint record and proceeds forward in the log until the last log record. It determines
   (a) The point in the log at which to start the Redo pass.

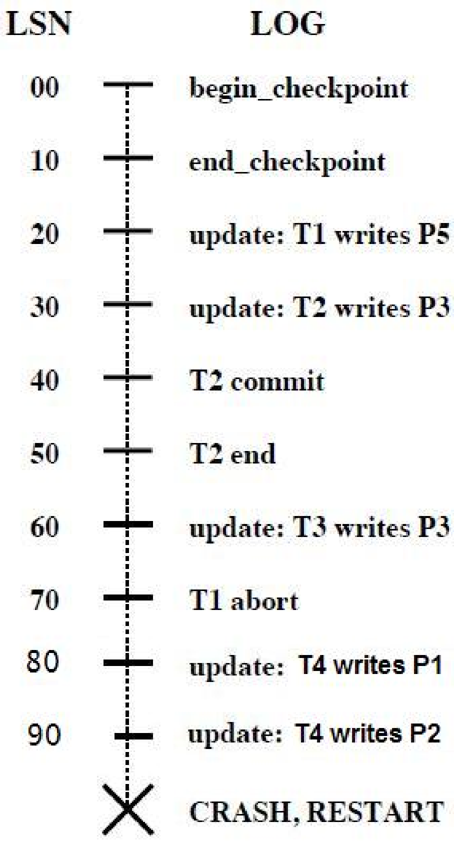

Figure 1: Execution with a Crash.

(b) The dirty pages in the buffer pool at the time of the crash.

(c) Transactions that were active at the time of the crash which need to be undone.

The Redo phase follows Analysis and redoes all changes to any page that might have been dirty at the time of the crash. The Undo phase follows Redo and undoes the changes of all transactions that were active at the time of the crash.

2. **(a)** For this example, we will assume that the Dirty Page Table and Transaction Table were empty before the start of the log. Analysis determines that the last begin checkpoint was at LSN 00 and starts at the corresponding end checkpoint (LSN 10). We will denote Transaction Table records as (transID, lastLSN) and Dirty Page Table records as (pageID, recLSN) sets. Then Analysis phase runs until LSN 90, and does the following:

LSN 20	Adds (T1, 20) to TT and (P5, 20) to DPT
LSN 30	Adds (T2, 30) to TT and (P3, 30) to DPT
LSN 40	Changes status of T2 to "C" from "U"
LSN 50	Deletes entry for T2 from Transaction Table
LSN 60	Adds (T3, 60) to TT. Does not change P3 entry in DPT
LSN 70	Changes (T1, 20) to (T1, 70)
LSN 80	Adds (T4, 80) to TT and (P1, 80) to DPT
LSN 90	Changes (T4, 80) to (T4, 90) and adds (P2, 90) to DPT

The final Transaction Table has three entries: (T1, 70), (T3, 60), and (T4, 90). The final Dirty Page Table has four entries: (P5, 20), (P3, 30), (P1, 80), and (P2, 90).

(b) Redo Phase: Redo starts at LSN 20 (smallest recLSN in DPT).

LSN 20	Changes to P5 are redone
LSN 30	P3 is retrieved and its pageLSN is checked. If the page had been written to disk before the crash (i.e. if pageLSN >= 30), nothing is re-done otherwise the changes are redone
LSN 40,50	No action
LSN 60	Changes to P3 are redone
LSN 70	No action
LSN 80	Changes to P1 are redone
LSN 90	Changes to P2 are redone

(c) Undo Phase: The ToUndo set consists of LSNs 70, 60 and 90. Undo starts at LSN 90 (highest lastLSN in TT).

LSN 90	Undoes the change on P2 Adds a CLR indicating this Undo Adds LSN 80 to ToUndo

ToUndo consists of (80, 70, 60)

LSN 80	Undoes the change on P1 Adds a CLR indicating this Undo

ToUndo consists of (70, 60)

LSN 70	Adds LSN 20 to ToUndo

ToUndo consists of (60, 20)

LSN 60	Undoes the change on P3 Adds a CLR indicating this Undo

ToUndo consists of (20)

LSN 20	Undoes the change on P5 Adds a CLR indicating this Undo

ToUndo is empty

3 ARIES Crash During Recovery

Consider the execution shown in Figure 2. In addition, the system crashes during recovery after writing two log records to stable storage and again after writing another two log records.

1. What is done during Analysis?
2. What is done during Redo?
3. What is done during Undo?
4. Show the log when recovery is complete, including all non-null prevLSN and undonextLSN values in log records.

Solution

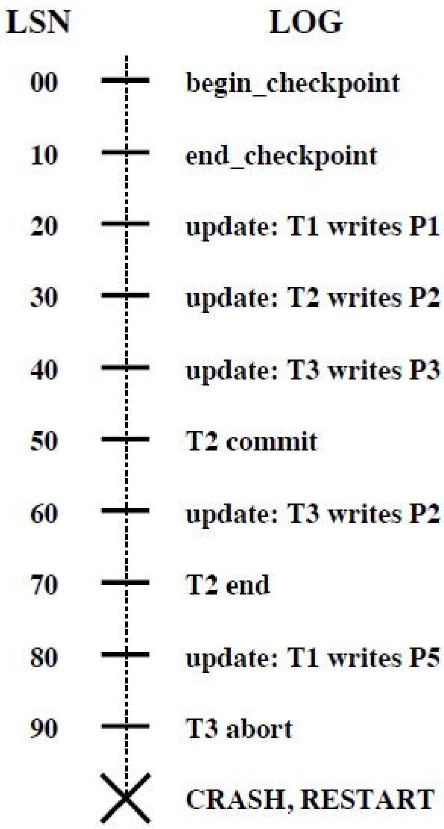

Figure 2: Execution with Multiple Crashes.

1. During analysis the following happens:

LSN 20	Adds (T1, 20) to TT and (P1, 20) to DPT
LSN 30	Adds (T2, 30) to TT and (P2, 30) to DPT
LSN 40	Adds (T3, 40) to TT and (P3, 40) to DPT
LSN 50	Changes status of T2 to "C"
LSN 60	Changes (T3, 40) to (T3, 60)
LSN 70	Removes T2 from TT
LSN 80	Changes (T1, 20) to (T1, 80) and adds (P5, 80) to DPT
LSN 90	No action

   At the end of analysis, the transaction table contains the following entries: (T1, 80), and (T3, 60). The Dirty Page Table has the following entries: (P1, 20), (P2, 30), (P3, 40), and (P5, 80).

2. Redo starts from LSN20 (minimum recLSN in DPT).

LSN 20	Check whether P1 has pageLSN more than 10 or not. Since it is a committed transaction, we probably need not redo this update
LSN 30	Redo the change in P2
LSN 40	Redo the change in P3
LSN 50	No action
LSN 60	Redo the changes on P2
LSN 70	No action
LSN 80	Redo the changes on P5
LSN 90	No action

3. ToUndo consists of (80, 60)

LSN 80	Undoes the change on P5 Appends a CLR: Undo T1 LSN 80, set undonextLSN = 20 Adds 20 to ToUndo

ToUndo consists of (60, 20)

LSN 60	Undoes the change on P2 Appends a CLR: Undo T3 LSN 60, set undonextLSN = 40 Adds 40 to ToUndo

ToUndo consists of (40, 20)

LSN 40	Undoes the change on P3 Appends a CLR: Undo T3 LSN 40, T3 end

ToUndo consists of (20)

LSN 20	Undoes the change on P1 Appends a CLR: Undo T1 LSN 20, T1 end

ToUndo is empty

4. The log looks like the following after recovery:

LSN 00	begin checkpoint	
LSN 10	end checkpoint	
LSN 20	update: T1 writes P1	
LSN 30	update: T2 writes P2	
LSN 40	update: T3 writes P3	
LSN 50	T2 commit	prevLSN = 30
LSN 60	update: T3 writes P2	prevLSN = 40
LSN 70	T2 end	prevLSN = 50
LSN 80	update: T1 writes P5	prevLSN = 20
LSN 90	T3 abort	prevLSN = 60
LSN 100	CLR: Undo T1 LSN 80	undonextLSN= 20
LSN 110	CLR: Undo T3 LSN 60	undonextLSN= 40
LSN 120,125	CLR: Undo T3 LSN 40	T3 end.
LSN 130,135	CLR: Undo T1 LSN 20	T1 end.

4 Project

Implement a first version of your project implementing the contest specification. You should use the code used for the first milestone of this lecture (main-memory index). Make sure your project passes all tests. Deadline: January 27^{th}, midnight. More info at: http://infosys.cs.uni-saarland.de/teaching/dbs0809/exercises/project.html

Solution

Programming assignment.