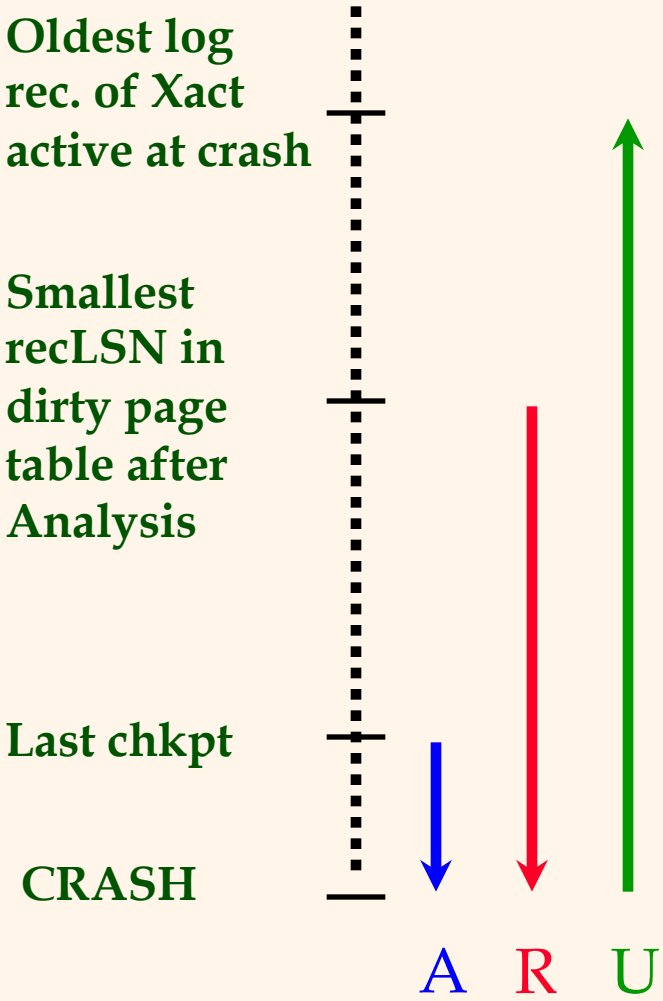
Crash Recovery

Chapter 18

Crash Recovery: Big Picture



- Start from a checkpoint (found via master record).
- Three phases. Need to:
  - ANALYSIS. Figure out which Xacts committed/failed since checkpoint.
  - REDO actions of dirty pages.
  - UNDO effects of failed Xacts.

Recovery: The Analysis Phase

- The *analysis* phase performs three tasks:

1. It determines the point in the log at which to start the redo pass

2. It determines pages in the buffer pool that were dirty at the time of crash

3. It identifies transactions that were active at the time of crash and must be undone

Recovery: The Analysis Phase

- *Analysis* begins by examining the most recent begin_checkpoint log record
  - Initialize the dirty page table and transaction table to the copies of those structures in the next end_checkpoint record
  - If additional log records are between begin_checkpoint and end_checkpoint records, the tables must be adjusted accordingly.

Recovery: The Analysis Phase

- Scan log forward from checkpoint.
  - End record for T: Remove T from the transaction table where it is not active anymore.
  - Other records for T: Add T to the transaction table if it is not there. The entry for T is modified to be:
    - The LastLSN field is set to current LSN
    - The status is set to C if this log record is commit, o.w., the status is set to U
  - Update record for Page P: Add P to the dirty table if it is not already there with RecLSN = LSN

Recovery: The REDO Phase

- During the REDO phase, we reapply the updates of all transactions to reconstruct state at crash (*repeat History)*
  - Reapply *all* updates (even of aborted Xacts!), redo CLRs.

- The REDO phase starts with the log record that has the smallest recLSN of all pages in the dirty page table.
  - This log record identifies the oldest update that may not have been written to disk

Recovery: The REDO Phase

- Scan forward from smallest recLSN in dirty page table until the end of the log.
- For each rodoable log record (Update or CLR), check weather the logged action must be redone.
- To REDO an action:
  - Reapply logged action.
  - Set pageLSN to LSN. No additional logging!

Recovery: The REDO Phase

- All redoable actions must be redone unless one of the following conditions holds:
  1. Affected page is not in the Dirty Page Table. This means that this page has alreay made it to disk
  2. Affected page is in the Dirty Page Table, but has recLSN > LSN. This means that this change is not the one that is responsible for making this page dirty, i.e., this change has already made it to disk
  3. Affected page is in the Dirty Page Table, but has pageLSN that is greater than or equal LSN. This means that either this update or a later update to the page was written to disk.

Recovery: The UNDO Phase

- Unlike the other two phases, the UNDO phase scans backward from the end of the log
- The goal of this phase is to undo the actions of all transactions active at the time of the crash.
- Undo identifies a set of *loser transactions* by scanning the transaction table constructed by the analysis phase and selecting those transactions with status U

Recovery: The UNDO Phase

ToUndo={ *l* | *l* a lastLSN of a "loser" Xact}

Repeat:

- Choose largest LSN among ToUndo.
- If this LSN is an update. Undo the update, write a CLR, add prevLSN to ToUndo.
- If this LSN is a CLR and undonextLSN==NULL
  - Write an End record for this Xact.
- If this LSN is a CLR, and undonextLSN != NULL
  - Add undonextLSN to ToUndo

Until ToUndo is empty.

Recovery Example

Xact Table
lastLSN
status
Dirty Page Table
recLSN
flushedLSN

ToUndo

LSN	LOG
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update T2 writes P3
30	T1 abort
40	CLR: Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5
	CRASH, RESTART
70	CLR: Undo T2 LSN 60
80	CLR: Undo T3 LSN 50
85	T3 end
	CRASH, RESTART
90	CLR: Undo T2 LSN 20
95	T2 end

Summary of Logging/Recovery

- Recovery Manager guarantees Atomicity & Durability.
- Use WAL to allow STEAL/NO-FORCE w/o sacrificing correctness.
- LSNs identify log records; linked into backwards chains per transaction (via prevLSN).
- pageLSN allows comparison of data page and log records.

Summary, Cont.

- Checkpointing: A quick way to limit the amount of log to scan on recovery.
- Recovery works in 3 phases:
  - Analysis: Forward from checkpoint.
  - Redo: Forward from oldest recLSN.
  - Undo: Backward from end to first LSN of oldest Xact alive at crash.
- Upon Undo, write CLRs.
- Redo "repeats history": Simplifies the logic!