

Lecture 8: Recovery (Part 2)

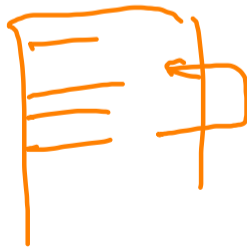
7.5%
12.5% each } 45%

£ sheet:
→ SiloR

Log Sequence Numbers

BM / LM / TM

- Log Sequence Numbers:
 - ▶ LSNs identify log records; linked into backwards chains per transaction via prevLSN.
 - ▶ pageLSN allows comparison of data page and log records.



ARIES

- Mains ideas of ARIES:
 - ▶ WAL with STEAL/NO-FORCE
 - ▶ Fuzzy Checkpoints (snapshot of dirty page ids)
 - ▶ Write CLR^s when undoing, to survive failures during restarts
 - ▶ ATT tells the DBMS which txns were active at time of crash.
 - ▶ DPT tells the DBMS which dirty pages might not have made it to disk.

Fuzzy Checkpointing

- The LSN of the <CHECKPOINT-BEGIN> record is written to the database's MasterRecord entry on disk when the checkpoint successfully completes.
- Any txn that starts after the checkpoint is excluded from the ATT in the <CHECKPOINT-END> record.

TXN-END Record: Abort

- First write an <ABORT> record to log for the txn.
- Then play back the txn's updates in reverse order. For each update record:
 - ▶ Write a CLR entry to the log.
 - ▶ Restore old value.
- When a txn aborts, we immediately tell the application that it is aborted.
- We don't need to wait to flush the CLR's
- At end, write a <TXN-END> log record.
- Notice: CLR's never need to be undone.

TXN-END Record: Commit

- Write <COMMIT> Record to Log
- All log records up to the transaction's LastLSN are flushed.
 - ▶ Log flushes are sequential, synchronous writes to disk
- Commit() returns
- Write <TXN-END> record to log
- Besides flushing, <TXN-END> record is related to releasing locks

Early Lock Release

Purpose of CLR

/ . 30 → 40

 / . 40 → 30

- Before restoring the old value of a page, write a Compensation Log Record (CLR).
- Logging continues during UNDO processing
- CLR contains REDO info
- CLR is never UNDONE
 - ▶ Undo need not be idempotent (>1 UNDO won't happen)
 - ▶ But they might be Redone when repeating history (=1 UNDO guaranteed)
- By appropriate changing of the CLR to log records written during forward processing, a bounded amount of logging is ensured during rollbacks, even in the face of repeated failures during restart.

Today's Agenda

- Phases of ARIES
- Analysis Phase
- Redo and Undo Phases
- Full Example
- Additional Crash Issues

Phases of ARIES

ARIES – Phases

A C I D

- Phase 1 – Analysis

- ▶ Read WAL from last checkpoint to identify dirty pages in the buffer pool and active txns at the time of the crash.

- Phase 2 – Redo

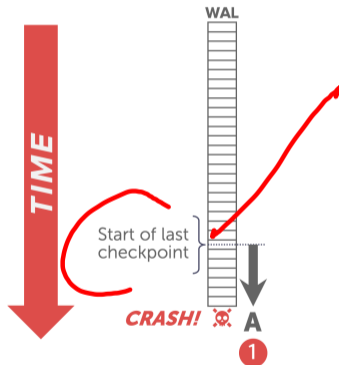
- ▶ Repeat all actions starting from an appropriate point in the log (even txns that will abort).

- Phase 3 – Undo

- ▶ Reverse the actions of txns that did not commit before the crash.

ARIES – Overview

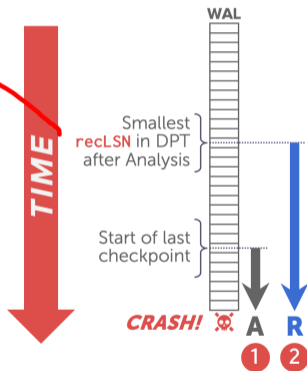
- Start from last <BEGIN-CHECKPOINT> found via MasterRecord.
- Analysis: Figure out which txns committed or failed since checkpoint.
- Redo: Repeat all actions.
- Undo: Reverse effects of failed txns.



ARIES – Overview

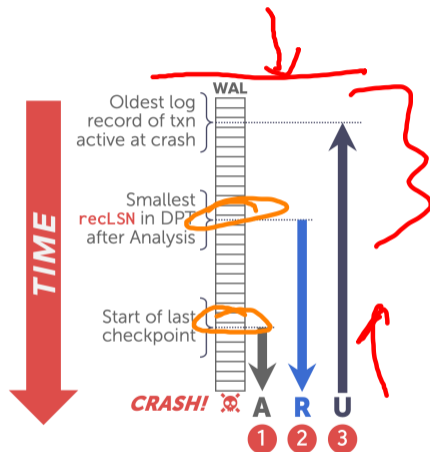
page log, rec log

- Start from last <BEGIN-CHECKPOINT> found via MasterRecord.
- Analysis: Figure out which txns committed or failed since checkpoint.
- Redo: Repeat all actions.
- Undo: Reverse effects of failed txns.



ARIES – Overview

- Start from last <BEGIN-CHECKPOINT> found via MasterRecord.
- Analysis: Figure out which txns committed or failed since checkpoint.
- Redo: Repeat all actions.
- Undo: Reverse effects of failed txns.



Analysis Phase

Analysis Phase

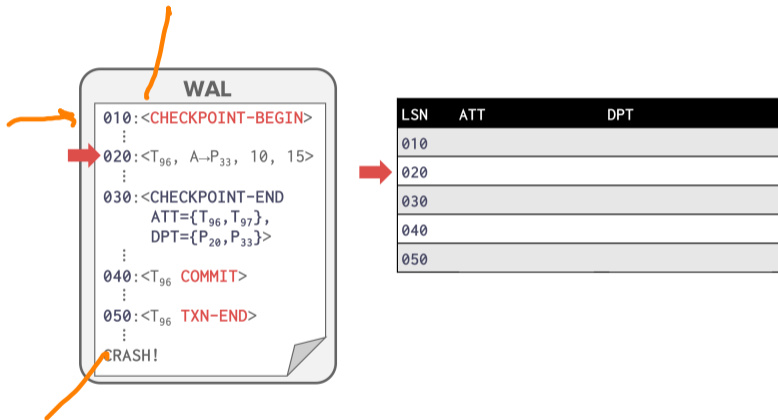
- Scan log forward from last successful checkpoint.
- If you find a TXN-END record, remove its corresponding txn from ATT.
- All other records:
 - ▶ Add txn to ATT with status UNDO.
 - ▶ On commit, change txn status to COMMIT.
- For UPDATE records:
 - ▶ If page P not in DPT, add P to DPT, set its recLSN = LSN.
 - ▶ recLSN: LSN of the log record which first caused the page to be dirty

LSN: 15 → P 10
 20 ←

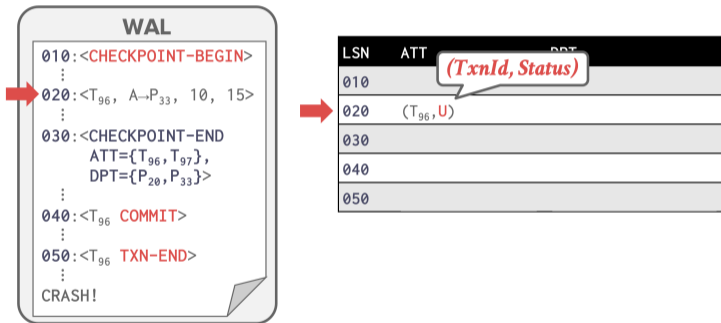
Analysis Phase

- At end of the Analysis Phase:
 - ▶ ATT tells the DBMS which txns were active at time of crash.
 - ▶ DPT tells the DBMS which dirty pages might not have made it to disk.

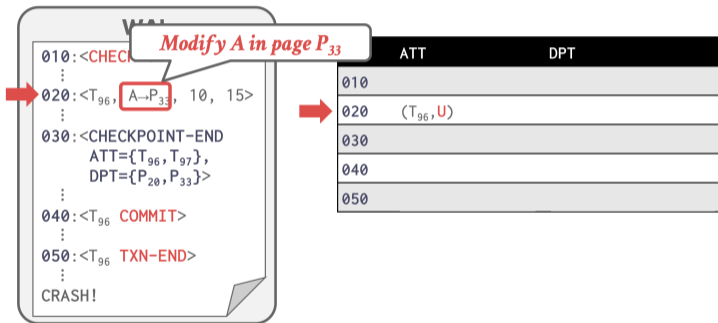
Analysis Phase: Example



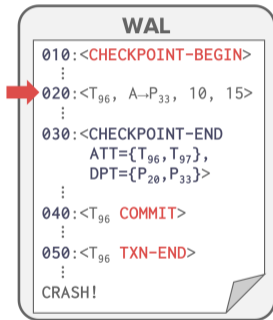
Analysis Phase: Example



Analysis Phase: Example



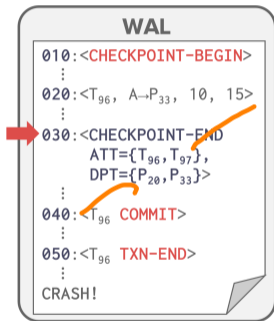
Analysis Phase: Example



| LSN | ATT | DPT |
|-----|-----------------------|-------------------------|
| 010 | | |
| 020 | (T ₉₆ , U) | (P ₃₃ , 020) |
| 030 | | |
| 040 | | |
| 050 | | |

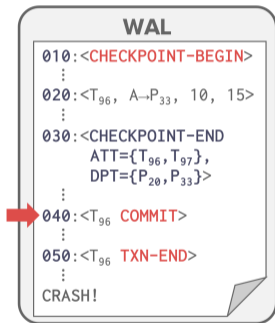
reclsn

Analysis Phase: Example



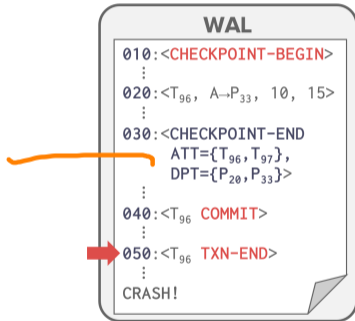
| LSN | ATT | DPT |
|-----|--|--|
| 010 | | |
| 020 | (T ₉₆ , U) | (P ₃₃ , 020) |
| 030 | (T ₉₆ , U), (T ₉₇ , U) | (P ₃₃ , 020), (P ₂₀ , 022) |
| 040 | | |
| 050 | | |

Analysis Phase: Example



| LSN | ATT | DPT |
|-----|--|--|
| 010 | | |
| 020 | (T ₉₆ , U) | (P ₃₃ , 020) |
| 030 | (T ₉₆ , U), (T ₉₇ , U) | (P ₃₃ , 020), (P ₂₀ , 022) |
| 040 | (T ₉₆ , C), (T ₉₇ , U) | (P ₃₃ , 020), (P ₂₀ , 022) |
| 050 | | |

Analysis Phase: Example



| LSN | ATT | DPT |
|-----|--|--|
| 010 | | |
| 020 | (T ₉₆ , U) | (P ₃₃ , 020) |
| 030 | (T ₉₆ , U), (T ₉₇ , U) | (P ₃₃ , 020), (P ₂₀ , 022) |
| 040 | (T ₉₆ , C), (T ₉₇ , U) | (P ₃₃ , 020), (P ₂₀ , 022) |
| 050 | (T ₉₇ , U) | (P ₃₃ , 020), (P ₂₀ , 022) |

A red arrow points from the TXN-END record (050) in the WAL to the first row of the table.



Redo and Undo Phases

Redo Phase

→ 1M

Defensive
Updates

- The goal is to repeat history to reconstruct state at the moment of the crash:
 - ▶ Reapply all updates (even aborted txns!) and redo CLRs.
- There techniques that allow the DBMS to avoid unnecessary reads/writes, but we will ignore that in this lecture...

physiological

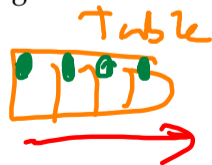
→ 1K
100

→ 10K
100K

Redo Phase

self-contained pages

- Scan forward from the log record containing smallest/oldest recLSN in DPT.
- For each update log record or CLR with a given LSN, redo the action unless:
 - Affected page is not in DPT, or
 - Affected page is in DPT but that record's LSN is older than page's recLSN.
- Apply changes for pages in DPT and pageLSN (in DB) < LSN
- Everything before the oldest recLSN in DPT is guaranteed to have been flushed.
- If a page's recLSN is newer than LSN, then no need to read page in from disk to check pageLSN



Redo Phase

pageLSN / recLSN

- To redo an action:
 - ▶ Reapply logged action.
 - ▶ Set **pageLSN** to log record's LSN.
 - ▶ No additional logging, no forced flushes!
- At the end of Redo Phase, write <TXN-END> log records for all txns with status C and remove them from the ATT.

A R X A
↑

Undo Phase

→ 10 (log)
!:
20 (log)

- Undo all txns that were active at the time of crash and therefore will never commit.
 - ▶ These are all the txns with U status in the ATT after the Analysis Phase.
- Process them in reverse LSN order using the lastLSN to speed up traversal.
- Write a CLR for every modification.

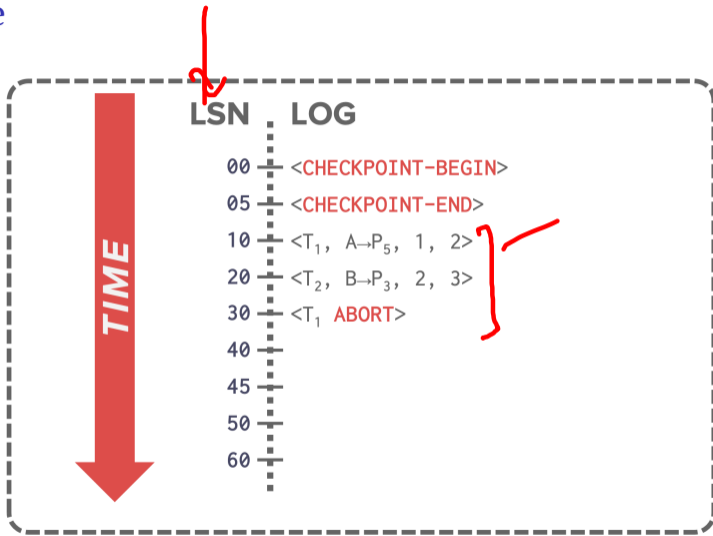
prev LSN, last LSN

Undo Phase

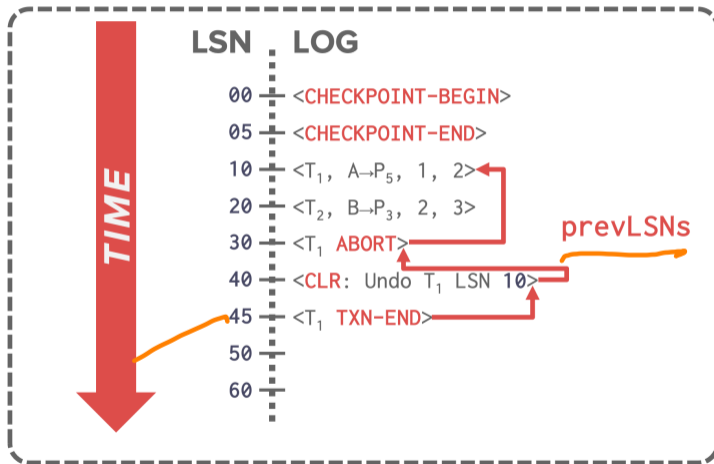
- ToUndo = **lastLSN** of "loser" txns
- Repeat until ToUndo is empty:
 - ▶ Pop largest LSN from ToUndo.
 - ▶ If this LSN is a CLR and **undoNext** = nil, then write an **TXN-END** record for this txn.
 - ▶ If this LSN is a CLR, and **undoNext** != nil, then add **undoNext** to ToUndo
 - ▶ Else this LSN is an update. Undo the update, write a CLR, add prevLSN to ToUndo.

Full Example

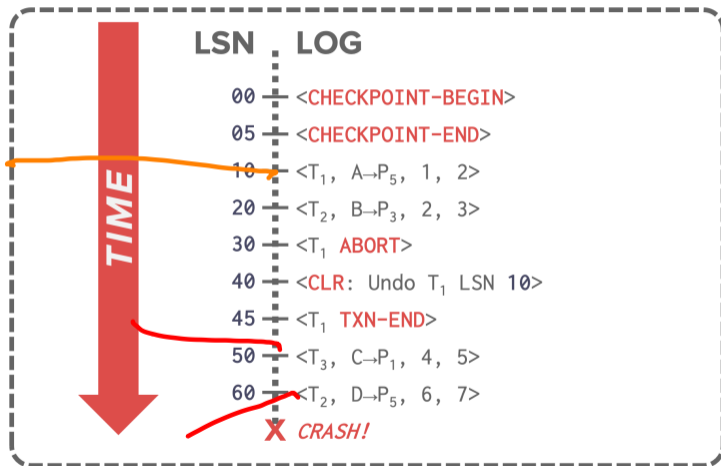
Full Example



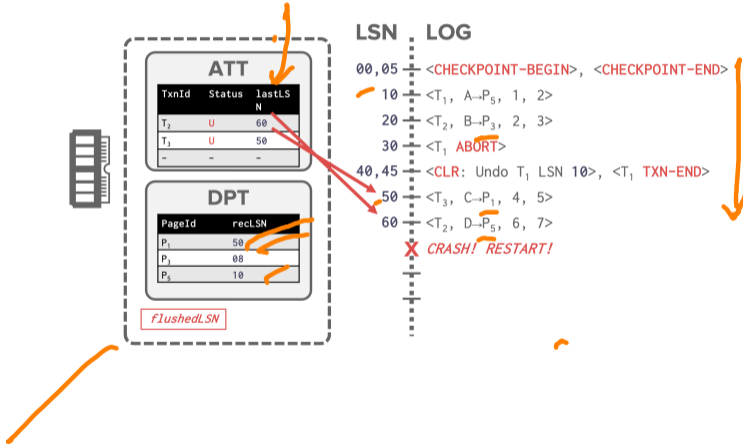
Full Example



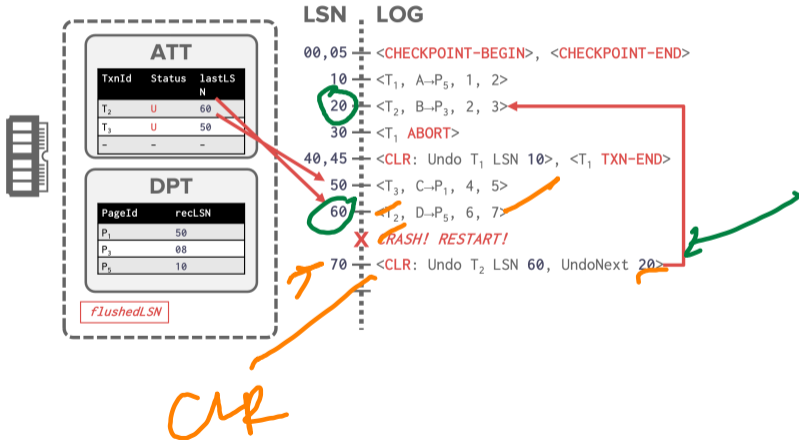
Full Example



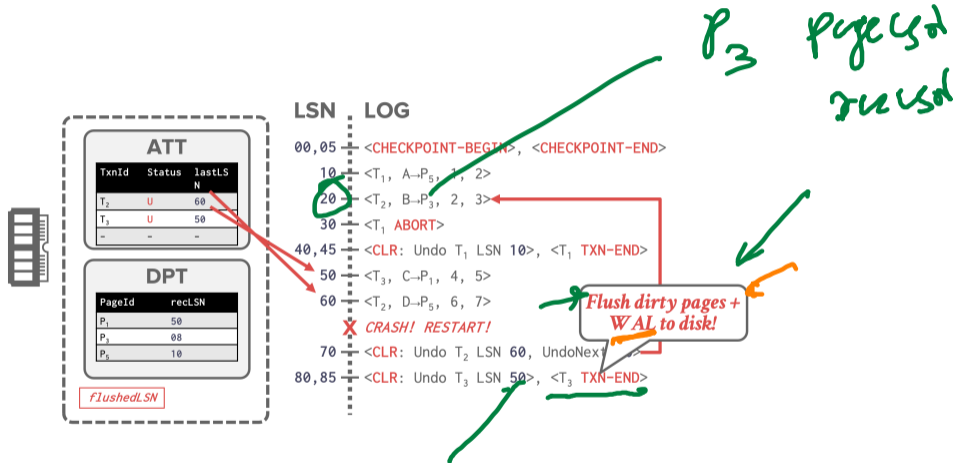
Full Example



Full Example



Full Example



Full Example

Disk

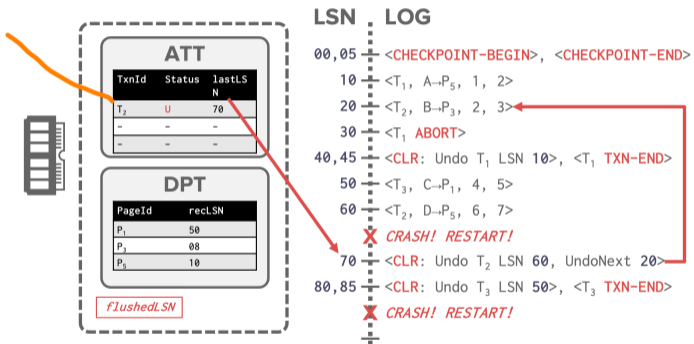
WAL



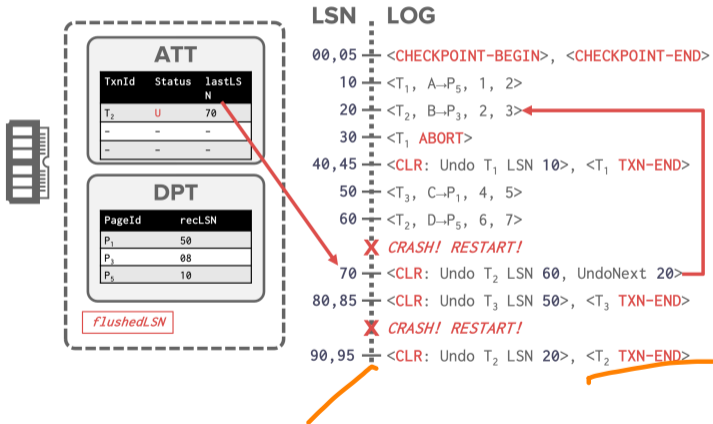
| LSN | LOG |
|-------|---|
| 00,05 | <CHECKPOINT-BEGIN>, <CHECKPOINT-END> |
| 10 | <T ₁ , A→P ₅ , 1, 2> |
| 20 | <T ₂ , B→P ₃ , 2, 3> |
| 30 | <T ₁ ABORT> |
| 40,45 | <CLR: Undo T ₁ LSN 10>, <T ₁ TXN-END> |
| 50 | <T ₃ , C→P ₁ , 4, 5> |
| 60 | <T ₂ , D→P ₅ , 6, 7> |
| | X CRASH! RESTART! |
| 70 | <CLR: Undo T ₂ LSN 60, UndoNext> |
| 80,85 | <CLR: Undo T ₃ LSN 50>, <T ₃ TXN-END> |
| | X CRASH! RESTART! |

Flush dirty pages +
WAL to disk!

Full Example




Full Example



Additional Crash Issues

Additional Crash Issues (1)

- What does the DBMS do if it crashes during recovery in the Analysis Phase?
 - What does the DBMS do if it crashes during recovery in the Redo Phase?
- 

Additional Crash Issues (1)

- What does the DBMS do if it crashes during recovery in the Analysis Phase?
 - ▶ Nothing. Just run recovery again.
- What does the DBMS do if it crashes during recovery in the Redo Phase?
 - ▶ Again nothing. Redo everything again.

+ Log Updates
+ CLRs

Additional Crash Issues (2)

- How can the DBMS improve performance during recovery in the Redo Phase?
- How can the DBMS improve performance during recovery in the Undo Phase?

!

Additional Crash Issues (2)

~~NO~~
Background Writer

- How can the DBMS improve performance during recovery in the Redo Phase?
 - ▶ Assume that it is not going to crash again and flush all changes to disk asynchronously in the background.
- How can the DBMS improve performance during recovery in the Undo Phase?
 - ▶ Lazily rollback changes before new txns access pages.
 - ▶ Rewrite the application to avoid long-running txns.

REDO + NO-UNDO

Concurrency Control

Conclusion

Parting Thoughts

- Mains ideas of ARIES:
 - ▶ WAL with STEAL/NO-FORCE
 - ▶ Fuzzy Checkpoints (snapshot of dirty page ids)
 - ▶ Redo everything since the earliest dirty page
 - ▶ Undo txns that never commit
 - ▶ Write CLRs when undoing, to survive failures during restarts

Atomicity

Next Class

- Deconstruct ARIES