CS 186
Fall 2024

Introduction to Database Systems
Alvin Cheung

DIS 9

# 1 Undo Logging

The following table gives the operations for Transaction T in the first column. Disk A = 7 and Disk B = 3 at the beginning.

| Operation | Mem A | Mem B | Disk A | Disk B | UNDO Log |
|---|---|---|---|---|---|
| | | | | | |
| READ(A) | | | | | |
| READ(B) | | | | | |
| WRITE(A, A+B) | | | | | |
| WRITE(B, A-B) | | | | | |
| FLUSH(A) | | | | | |
| FLUSH(B) | | | | | |
| COMMIT | | | | | |

(a) Fill in the columns Mem A, Mem B, Disk A, Disk B.

**Solution:**

| Operation | Mem A | Mem B | Disk A | Disk B | UNDO Log |
|---|---|---|---|---|---|
| | | | | | |
| READ(A) | 7 | | 7 | 3 | |
| READ(B) | 7 | 3 | 7 | 3 | |
| WRITE(A, A+B) | 10 | 3 | 7 | 3 | |
| WRITE(B, A-B) | 10 | 7 | 7 | 3 | |
| FLUSH(A) | 10 | 7 | 10 | 3 | |
| FLUSH(B) | 10 | 7 | 10 | 7 | |
| COMMIT | 10 | 7 | 10 | 7 | |

(b) If the system crashes right before COMMIT, how will we recover? Fill in the UNDO column.

**Solution:** Undo by setting A = 7 and B = 3

| Operation | Mem A | Mem B | Disk A | Disk B | UNDO Log |
|---|---|---|---|---|---|
| | | | | | <START T> |
| READ(A) | 7 | | 7 | 3 | |
| READ(B) | 7 | 3 | 7 | 3 | |
| WRITE(A, A+B) | 10 | 3 | 7 | 3 | <T, A, 7> |
| WRITE(B, A-B) | 10 | 7 | 7 | 3 | <T, B, 3> |
| FLUSH(A) | 10 | 7 | 10 | 3 | |
| FLUSH(B) | 10 | 7 | 10 | 7 | |
| COMMIT | 10 | 7 | 10 | 7 | <COMMIT T> |

CRASH!

(c) What happens if the system crashes again while we're undoing? How do we recover?

**Solution:** If the system crashes, then we've done some of the undos but maybe not all of them. We need to do all the undos again.

# 2 Redo Logging

The following table gives the operations for Transaction T in the first column. Disk A = 5 and Disk B = 4 at the beginning.

| Operation | Mem A | Mem B | Disk A | Disk B | REDO Log |
|---|---|---|---|---|---|
| | | | | | |
| READ(A) | | | | | |
| READ(B) | | | | | |
| WRITE(A, A+B) | | | | | |
| WRITE(B, A-B) | | | | | |
| COMMIT | | | | | |
| FLUSH(A) | | | | | |
| FLUSH(B) | | | | | |

(a) Fill in the columns Mem A, Mem B, Disk A, Disk B.

**Solution:**

| Operation | Mem A | Mem B | Disk A | Disk B | REDO Log |
|---|---|---|---|---|---|
| | | | | | |
| READ(A) | 5 | | 5 | 4 | |
| READ(B) | 5 | 4 | 5 | 4 | |
| WRITE(A, A+B) | 9 | 4 | 5 | 4 | |
| WRITE(B, A-B) | 9 | 5 | 5 | 4 | |
| COMMIT | | | | | |
| FLUSH(A) | 9 | 5 | 9 | 4 | |
| FLUSH(B) | 9 | 5 | 9 | 5 | |

(b) If the system crashes right after COMMIT, how will we recover? Fill in the REDO column.

**Solution:** Recover by REDO-ing and setting A = 9, B = 5.

| Operation | Mem A | Mem B | Disk A | Disk B | REDO Log |
|---|---|---|---|---|---|
| | | | | | <START T> |
| READ(A) | 5 | | 5 | 4 | |
| READ(B) | 5 | 4 | 5 | 4 | |
| WRITE(A, A+B) | 9 | 4 | 5 | 4 | <T, A, 9> |
| WRITE(B, A-B) | 9 | 5 | 5 | 4 | <T, B, 5> |
| COMMIT | | | | | <COMMIT> |
| FLUSH(A) | 9 | 5 | 9 | 4 | |
| FLUSH(B) | 9 | 5 | 9 | 5 | |

CRASH!

# 3 Recovery Q1

Consider the execution of the ARIES recovery algorithm given the following log (assume a checkpoint is completed before LSN 0 and the Dirty Page Table and Transaction Table for that checkpoint are empty):

| LSN | Log Record |
|---|---|
| 10 | update: T1 writes P1 |
| 20 | update: T2 writes P3 |
| 30 | T1 commit |
| 40 | update: T3 writes P4 |
| 50 | update: T2 writes P1 |
| 60 | T1 end |
| 70 | update: T3 writes P2 |
| 80 | T2 abort |

(a) During Analysis, what log records are read? What are the contents of the transaction table and the dirty page table at the end of the analysis stage?

**Solution:** All records (since the last checkpoint) are read. We read through the log forwards and add entries to the transaction table and the dirty page table. Note that lastLSN is the last LSN written by a transaction, while recLSN is the LSN which first caused a page to be dirty.

| Transaction Table | | | Dirty Page Table | |
|---|---|---|---|---|
| Transaction | lastLSN | Status | PageID | recLSN |
| T3 | 90 | Aborting | P1 | 10 |
| T2 | 80 | Aborting | P2 | 70 |
| | | | P3 | 20 |
| | | | P4 | 40 |

(b) During Redo, what log records are read? What data pages are read? What operations are redone (assuming no updates made it out to disk before the crash)?

**Solution:** Redo starts at LSN 10 (smallest recLSN in the dirty page table). All pages in the dirty page table are read from disk (i.e. P1, P2, P3, P4). Assuming no updates made it to disk, the pageLSN of each page on disk is always less than the current LSN, so all updates and CLR's are redone. The LSN's of these operations are: 10, 20, 40, 50, 70.

(c) During Undo, what log records are read? What operations are undone? Show any new log records that are written for CLR's. Start at LSN 100. Be sure to show the undoNextLSN.

**Solution:** The lastLSN in the transaction table is 90. Starting from here, we will read: 90, 80, 70, 50, 40, 20. Of these, the update operations are: 70, 50, 40, 20. Therefore, the new log records are:

| LSN | Record | prevLSN | undoNextLSN |
|---|---|---|---|
| 100 | CLR: undo T3 LSN 70 | 90 | 40 |
| 110 | CLR: undo T2 LSN 50 | 80 | 20 |
| 120 | CLR: undo T3 LSN 40 | 100 | null |
| 130 | T3 end | 120 | - |
| 140 | CLR: undo T2 LSN 20 | 110 | null |
| 150 | T2 end | 140 | - |

# 4 Recovery Q2

Your database server has just crashed due to a power outage. You boot it up, find the following log and checkpoint information on disk, and begin the recovery process. Assume we use a STEAL/NO FORCE recovery policy.

| LSN | Record | prevLSN |
|---|---|---|
| 30 | update: T3 writes P5 | null |
| 40 | update: T4 writes P1 | null |
| 50 | update: T4 writes P5 | 40 |
| 60 | update: T2 writes P5 | null |
| 70 | update: T1 writes P2 | null |
| 80 | begin_checkpoint | - |
| 90 | update: T1 writes P3 | 70 |
| 100 | end_checkpoint | - |
| 110 | update: T2 writes P3 | 60 |
| 120 | T2 commit | 110 |
| 130 | update: T4 writes P1 | 50 |
| 140 | T2 end | 120 |
| 150 | T4 abort | 130 |
| 160 | update: T5 writes P2 | null |
| 180 | CLR: undo T4 LSN 130 | 150 |

| Transaction Table | | | Dirty Page Table | |
|---|---|---|---|---|
| Transaction | lastLSN | Status | PageID | recLSN |
| T1 | 70 | Running | P5 | 50 |
| T2 | 60 | Running | P1 | 40 |
| T3 | 30 | Running | | |
| T4 | 50 | Running | | |

| On-Disk Page LSN Table | |
|---|---|
| PageID | pageLSN |
| P1 | 0 |
| P2 | 70 |
| P3 | 110 |
| P4 | 0 |
| P5 | 30 |

(a) The log record at LSN 60 says that transaction 2 updated page 5. Was this update to page 5 successfully written to disk? The log record at LSN 70 says that transaction 1 updated page 2. Was this update to page 2 successfully written to disk? Assume for this question we do not have access to the On-Disk Page LSN Table.

**Solution:** The update at LSN 60 may have been written to disk. The log entry was flushed before the write itself. It was not yet flushed at the time of the checkpoint, but may have been flushed later. The update at LSN 70 was flushed to disk. We know this because it's not in the dirty page table at the time of the checkpoint.

(b) At the end of the analysis phase, what transactions will be in the transaction table, and what pages will be in the dirty page table?

**Solution:** Note that P1 and P5 were already in the dirty page table during the checkpoint, and their recLSN's do not change during analysis

| **Transaction Table** | | | **Dirty Page Table** | |
|---|---|---|---|---|
| **Transaction** | **lastLSN** | **Status** | **PageID** | **recLSN** |
| T1 | 190 | Aborting | P1 | 40 |
| T3 | 200 | Aborting | P2 | 160 |
| T4 | 180 | Aborting | P3 | 90 |
| T5 | 210 | Aborting | P5 | 50 |

(c) At which LSN in the log should the redo phase begin? Which log records will be redone (list their LSNs)? All other log records will be skipped.

**Solution:** Redo should begin at LSN 40, the smallest of the recLSNs in the dirty page table. The following log records should be redone: 40, 50, 60, 130, 160, 180. 30 is skipped because it precedes LSN 40. 70 is skipped because P2.recLSN = 160 > 70. 90 and 110 are skipped because the LSNs are <= relevant pageLSN on disk. Entries that are not updates are skipped. The CLR record is not skipped, nor is the LSN that it undoes.

(d) Write down all of the log records written during the undo phase.

**Solution:** Undo should begin at LSN 210 because of additional abort records written during the Analysis phase. The update operations that are undone are: 160, 90, 70, 50, 40, 30. After all operations of an aborting transaction are undone, we also write an end record. Therefore, log records written during the Undo phase are:

| LSN | Log Record | prevLSN | undoNextLSN |
|---|---|---|---|
| 220 | CLR: undo T5 LSN 160 | 210 | null |
| 230 | T5 end | 220 | - |
| 240 | CLR: undo T1 LSN 90 | 190 | 70 |
| 250 | CLR: undo T1 LSN 70 | 240 | null |
| 260 | T1 end | 250 | - |
| 270 | CLR: undo T4 LSN 50 | 180 | 40 |
| 280 | CLR: undo T4 LSN 40 | 270 | null |
| 290 | T4 end | 280 | - |
| 300 | CLR: undo T3 LSN 30 | 200 | null |
| 310 | T3 end | 300 | - |