

CSE 444: Database Internals

Section 7:

Transactions – Recovery with ARIES

Review in this section

Recovery for ARIES

Example.

1. T_{1000} changes the value of **A** from “abc” to “def” on page P500
2. T_{2000} changes the value of **B** from “hij” to “klm” on page P600
3. T_{2000} changes the value of **D** from “mnp” to “qrs” on page P500
4. T_{1000} changes the value of **C** from “tuv” to “wxy” on page P505
5. T_{2000} commits and the end log record is written
6. T_{1000} changes the value of **E** from “pq” to “rs” on page P700
7. P600 is flushed to disk
8. **Crash!!**

Same as in Section 6

Example is adopted from Ramakrishnan-Gehrke book

ARIES Data Structures

Dirty page table

pageID	recLSN

LSN
101

Log

prevLSN	tID	pID	Log entry	Type	undoNextLSN

Transaction table

transID	lastLSN	status

Buffer Pool

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Log Record “Types”

- **Update:** easy
- **Commit:** log-tail forced-written to disk, up to & including commit (note that still no-force, the actual modified pages may not be written, and much smaller cost)
- **Abort:** abort type log record is written + undo is initiated for this transaction
- **End:** when a transaction is aborted or committed, some additional actions are performed, after that an end record is written
- **CLR:**
Undoing updates (during abort or recovery from crash), for every update record undone, write a CLR (Compensation Log Record)

First operation:

1. T_{1000} changes the value of **A** from “abc” to “def” on **page P500**?

Dirty page table

pageID	recLSN

LSN
101

Log

prevLSN	tID	pID	Log entry	Type	undoNextLSN

Transaction table

transID	lastLSN	status

Buffer Pool

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Changes

1. T_{1000} changes the value of **A** from “abc” to “def” on **page P500**

Dirty page table

pageID	recLSN
P500	101

LSN
101

Log

prevLSN	tID	pID	Log entry	Type	undoNextLSN
-	T1000	P500	Write A “abc” -> “def”	Update	-

Transaction table

transID	lastLSN	status
T_{1000}	101	Running

Buffer Pool

P500 PageLSN= 101 A = def D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Next:

2. T_{2000} changes the value of **B** from “hij” to “klm” on page **P600** ?

Dirty page table

pageID	recLSN
P500	101

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A “abc” -> “def”	Update	-

Transaction table

transID	lastLSN	status
T_{1000}	101	Running

Buffer Pool

P500 PageLSN= 101 A = def D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Changes:

2. T_{2000} changes the value of **B** from “hij” to “klm” on **page P600** ?

Dirty page table

pageID	recLSN
P500	101
P600	102

LSN
101
102

Log

prevLSN	tID	pID	Log entry	Type	undoNextLSN
-	T1000	P500	Write A “abc” -> “def”	Update	-
-	T_{2000}	P600	Write B “hij” -> “klm”		

Transaction table

transID	lastLSN	status
T_{1000}	101	Running
T_{2000}	102	Running

Buffer Pool

P500 PageLSN= 101 A = def D = mnp	P600 PageLSN= 102 B = klm
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Next:

3. T_{2000} changes the value of **D** from “mnp” to “qrs” on page **P500**?

Dirty page table

pageID	recLSN
P500	101
P600	102

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A “abc” -> “def”	Update	-
102	-	T_{2000}	P600	Write B “hij” -> “klm”		

Transaction table

transID	lastLSN	status
T_{1000}	101	Running
T_{2000}	102	Running

Buffer Pool

P500 PageLSN= 101 A = def D = mnp	P600 PageLSN= 102 B = klm
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Changes:

3. T_{2000} changes the value of **D** from “mnp” to “qrs” on page **P500**

Dirty page table

pageID	recLSN
P500	101
P600	102

Transaction table

transID	lastLSN	status
T_{1000}	101	Running
T_{2000}	103	Running

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T_{1000}	P500	Write A “abc” -> “def”	Update	-
102	-	T_{2000}	P600	Write B “hij” -> “klm”	Update	-
103	102	T_{2000}	P500	Write D “mnp” -> “qrs”	Update	-

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Next:

4. T_{1000} changes the value of **C** from “tuv” to “wxy” on page P505?

Dirty page table

pageID	recLSN
P500	101
P600	102

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T ₁₀₀₀	P500	Write A “abc” -> “def”	Update	-
102	-	T ₂₀₀₀	P600	Write B “hij” -> “klm”	Update	-
103	102	T ₂₀₀₀	P500	Write D “mnp” -> “qrs”	Update	-

Transaction table

transID	lastLSN	status
T ₁₀₀₀	101	Running
T ₂₀₀₀	103	Running

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Changes:

4. T_{1000} changes the value of **C** from “tuv” to “wxy” on page P505?

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T_{1000}	104	Running
T_{2000}	103	Running

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T_{1000}	P500	Write A “abc” -> “def”	Update	-
102	-	T_{2000}	P600	Write B “hij” -> “klm”	Update	-
103	102	T_{2000}	P500	Write D “mnp” -> “qrs”	Update	-
104	101	T_{1000}	P505	Write C “tuv” -> “wxy”	Update	-

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = wxy	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Next:

5. T_{2000} commits and the end log record is written

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T₁₀₀₀	104	Running
T ₂₀₀₀	103	Running

Log

LSN
101
102
103
104

prevLSN	tID	pID	Log entry	Type	undoNextLSN
-	T ₁₀₀₀	P500	Write A "abc" -> "def"	Update	-
-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
101	T₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Changes:

5. T_{2000} commits and the end log record is written --- step 1

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T_{1000}	104	Running
T_{2000}	103	Committed

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T_{1000}	P500	Write A "abc" -> "def"	Update	-
102	-	T_{2000}	P600	Write B "hij" -> "klm"	Update	-
103	102	T_{2000}	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T_{1000}	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T_{2000}			Commit	
106	105	T_{2000}			End	

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Changes:

5. T_{2000} commits and the end log record is written --- step 2

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

T2000 removed from transaction table

Transaction table

transID	lastLSN	status
T_{1000}	104	Running
T_{2000}	103	Committed

LSN

prevLSN	tID	pID	Log entry	Type	undoNextLSN
-	T_{1000}	P500	Write A "abc" -> "def"	Update	-
101	T_{2000}	P600	Write B "hij" -> "klm"	Update	-
102	T_{2000}	P500	Write D "mnp" -> "qrs"	Update	-
104	T_{1000}	P505	Write C "tuv" -> "wxy"	Update	-
105	T_{2000}			Commit	
106	T_{2000}			End	

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Changes:

5. T_{2000} commits and the end log record is written --- step 2

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T_{1000}	104	Running
T_{2000}	103	Committed

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T_{1000}	P500	Write A "abc" -> "def"	Update	-
102	-	T_{2000}	P600	Write B "hij" -> "klm"	Update	-
103	102	T_{2000}	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T_{1000}	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T_{2000}			Commit	
106	105	T_{2000}			End	

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= - E = pq

Disk

A = abc D = mnp	B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Log written to disk
 Note: no force = not the dirty pages changed by T_{2000} !

Whenever a transaction commits,
log is flushed to the disk == the log-tail is written to disk

NOTE:

1. The “Commit” record is required to be flushed (i.e. all logs up to and including that commit record)
2. The “End” record is not required to be flushed, in this case we are only assuming that it has been flushed as well (so that we have a good example while doing recovery 😊)

Next:

6. T_{1000} changes the value of **E** from “pq” to “rs” on page **P700**
 Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T_{1000}	104	Running

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T_{1000}	P500	Write A “abc” -> “def”	Update	-
102	-	T_{2000}	P600	Write B “hij” -> “klm”	Update	-
103	102	T_{2000}	P500	Write D “mnp” -> “qrs”	Update	-
104	101	T_{1000}	P505	Write C “tuv” -> “wxy”	Update	-
105	103	T_{2000}			Commit	
106	105	T_{2000}			End	

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= - E = pq

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Changes:

6. T_{1000} changes the value of **E** from “pq” to “rs” on **page P700**

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104
P700	107

Transaction table

transID	lastLSN	status
T_{1000}	107	Running

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T_{1000}	P500	Write A “abc” -> “def”	Update	-
102	-	T_{2000}	P600	Write B “hij” -> “klm”	Update	-
103	102	T_{2000}	P500	Write D “mnp” -> “qrs”	Update	-
104	101	T_{1000}	P505	Write C “tuv” -> “wxy”	Update	-
105	103	T_{2000}			Commit	
106	105	T_{2000}			End	
107	104	T_{1000}	P700	Write E “pq” -> “rs”	Update	-

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= 107 E = rs

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Next:

7. Page P600 is flushed to disk

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104
P700	107

Transaction table

transID	lastLSN	status
T₁₀₀₀	107	Running

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T ₁₀₀₀	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
107	104	T ₁₀₀₀	P700	Write E "pq" -> "rs"	Update	-

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= 107 E = rs

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= - B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Next:

7. Page P600 is flushed to disk – Step 1

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104
P700	107

Transaction table

transID	lastLSN	status
T ₁₀₀₀	107	Running

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T ₁₀₀₀	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
107	104	T ₁₀₀₀	P700	Write E "pq" -> "rs"	Update	-

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= 107 E = rs

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= 102 B = klm
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq



Next:

7. Page P600 is flushed to disk – Step 2

Dirty page table

pageID	recLSN
P500	101
P505	104
P700	107

Transaction table

transID	lastLSN	status
T ₁₀₀₀	107	Running

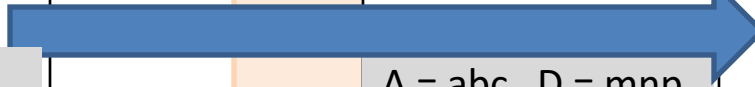
LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T ₁₀₀₀	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
107	104	T ₁₀₀₀	P700	Write E "pq" -> "rs"	Update	-

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN= 102 B = klm
P505 PageLSN= 104 C = tuv	P700 PageLSN= 107 E = rs

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= 102 B = klm
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq





8. CRASH!!

8. Crash!! ---- These are gone from memory

Dirty page

P700

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
107						

Transaction table

transID	lastLSN	status
		ing

Buffer Pool

P500 PageLSN= 103 A = def	
P505 PageLSN= 104 C = tuv	N= 107 E = rs

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= 102 B = klm
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Three Recovery Phases of ARIES

- Analysis
 - Reconstructs **Dirty page table** (for Redo) + **Transaction Table** (for Undo, active transactions at crash)
- Redo
 - Restores the database state at the time of crash by repeating Updates + CLR
- Undo
 - Undoes the actions of uncommitted transactions
 - Only updates can be undone, No CLR is ever undone!

Recall: ARIES Method Illustration

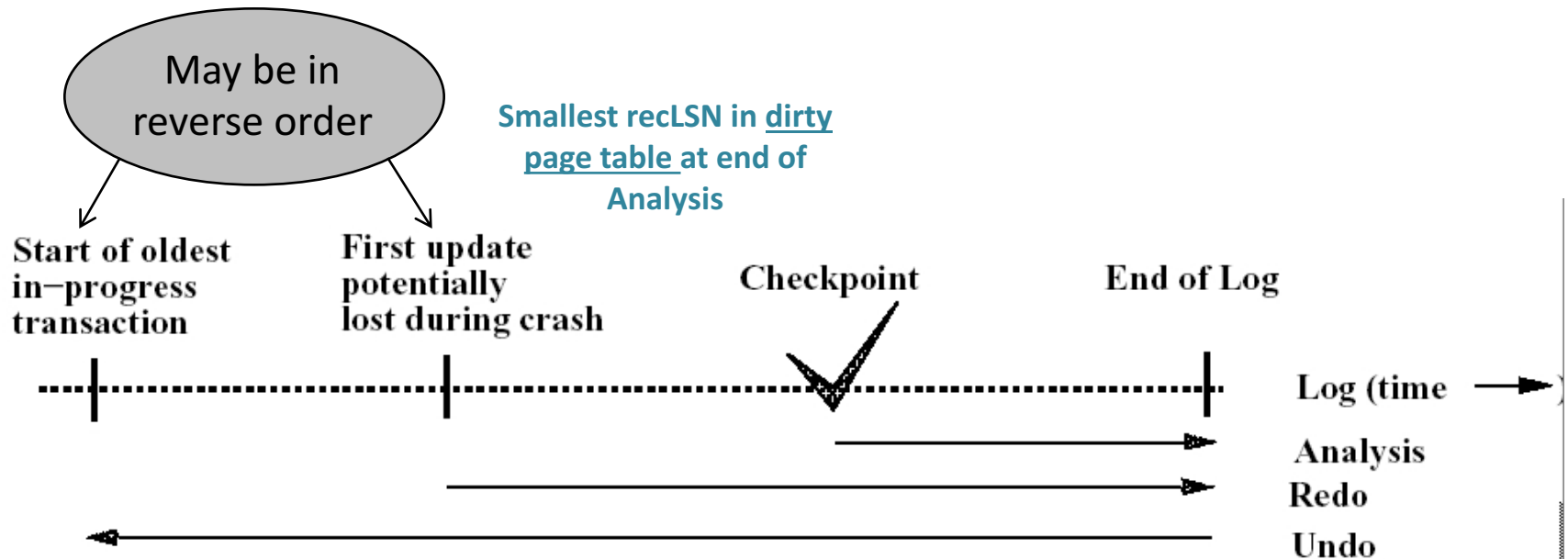


Figure 3: The Three Passes of ARIES Restart

Oldest log record of active transaction at the time of crash

(Trace back using transaction Table)

[Franklin97]

About Checkpointing

- This example has no checkpointing
- Analysis phase in the recovery starts with empty Dirty Page table and empty Transaction Table
 - Otherwise the latest copies of these tables have to be read from disk from the last checkpoint

Analysis Phase

- Reconstruct Dirty Page Table and Transaction Table
- Read the log from the last checkpoint
- Actual pages are not used

Analysis Phase

Log

Dirty page table

pageID	recLSN

Transaction table

transID	lastLSN	status

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
						Disk

Buffer Pool

Will not show the buffer pool from the next slide

P500 PageLSN= - A = abc D = mnp	P600 PageLSN= 102 B = klm
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Analysis Phase

Log

Dirty page table

pageID	recLSN
P500	101

Transaction table

transID	lastLSN	status
T1000	101	U=Unknown

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

Analysis Phase

Log

Dirty page table

pageID	recLSN
P500	101
P600	102

Transaction table

transID	lastLSN	status
T1000	101	U
T2000	102	U

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

Analysis Phase

Log

Dirty page table

pageID	recLSN
P500	101
P600	102

Transaction table

transID	lastLSN	status
T1000	103	U
T2000	102	U

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

Analysis Phase

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U
T2000	102	U

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

Analysis Phase

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U
T2000	105	C

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

Write A or Abort if you see an Abort log instead

Analysis Phase

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U
T2000	105	C

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

Remove entry from Transaction Table if you see an End record (both for Aborted and Committed transactions)

Analysis Phase

Log

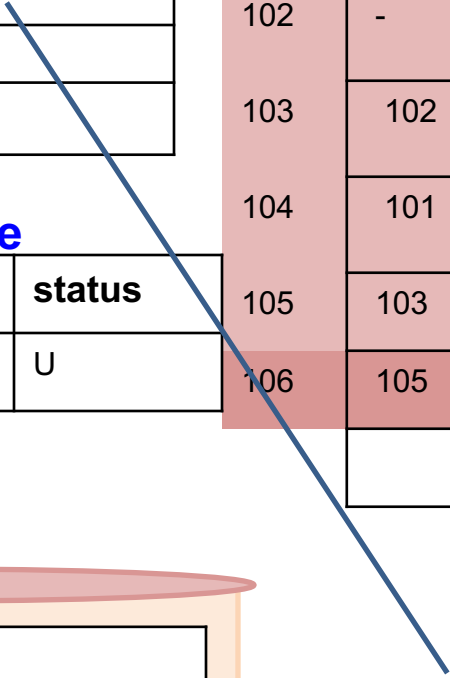
Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	



Already written to disk,
but reappears

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

REDO Phase

- “Repeating history” (including actions by transactions that would be aborted in the undo phase)
- Work with the Dirty Page Table
- Find the smallest recLSN in the dirty page table = FirstLSN
- Redo the “Update/CLR” action, unless (in this order)
 - Affected page is not in the dirty page table
 - recLSN > this LSN being checked
 - pageLSN >= this LSN being checked
- End/Commit/Abort LSNs are “skipped”
- **In HW, write “Redone” or “Skipped” for each LSN**

Analysis Phase

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

REDO Phase: find firstLSN

Log

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104



Transaction table

transID	lastLSN	status
T1000	104	U

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Disk

REDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= "-" to 101	
A = def D = mnp	

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

- Affected page is not in the dirty page table: **N**
- Else, recLSN > LSN being checked: **N**
- Else, pageLSN >= LSN being checked: **N**
- **REDO**

Disk

P500 PageLSN= -	P600 PageLSN=102
A = abc D = mnp	B = hij
P505 PageLSN= -	P700 PageLSN= -
C = tuv	E = pq

REDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= 101 A = def D = mnp	P600 PageLSN="-" to 102 B = klm

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

- Affected page is not in the dirty page table: **N**
- Else, recLSN > LSN being checked: **N**
- Else, pageLSN >= LSN being checked: **Y**
- **NO REDO = SKIPPED**

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

REDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= 101 to 103 A = def D = qrs	P600 PageLSN=102 B = klm

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

- Affected page is not in the dirty page table: **N**
- Else, recLSN > LSN being checked: **N**
- Else, pageLSN >= LSN being checked: **N**
- **REDO**

Disk

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

REDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN=102 B = klm
P505 PageLSN="" to 104 C = wxy	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

- Affected page is not in the dirty page table: **N**
- Else, recLSN > LSN being checked: **N**
- Else, pageLSN >= LSN being checked: **N**
- **REDO**

Disk

UNDO Phase

- Work with the Transaction table in the analysis phase
 - “Loser transactions” must be undone
 - Changes during undo phases are written (CLR) so that it is not repeated at the time of repeated restarts
- Scan backward
- Maintain a set ToUndo
 - *Initialize* to lastLSNs of all “U” transactions at Transaction Table
 - *Undo* the “largest LSN” in ToUndo at each step (the latest one in bottom-up order)

UNDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN=102 B = klm
P505 PageLSN= 104 C = wxy	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	

ToUNDO = {104}

Disk

UNDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN=102 B = hij
P505 PageLSN= 107 C = tuv	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
107		T ₁₀₀₀		UndoT ₁₀₀₀ LSN104	CLR	101

- A CLR is written
- PageLSN = LSN (CLR)
- Value of C is undone

ToUNDO = {101}

Disk

UNDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= 103 A = def D = qrs	P600 PageLSN=102 B = hij
P505 PageLSN= 107 C =tuv	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
107		T ₁₀₀₀		UndoT ₁₀₀₀ LSN104	CLR	101

Disk

ToUNDO = {101}

UNDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= 108 A = abc D = qrs	P600 PageLSN=102 B = hij
P505 PageLSN= 107 C = tuv	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
107		T ₁₀₀₀		UndoT ₁₀₀₀ LSN104	CLR	101
108		T ₁₀₀₀		UndoT ₁₀₀₀ LSN101	CLR	-

Disk

ToUNDO = {}

UNDO Phase

Dirty page table

pageID	recLSN
P500	101
P600	102
P505	104

Transaction table

transID	lastLSN	status
T1000	104	U

Buffer Pool

P500 PageLSN= 108 A = abc D = qrs	P600 PageLSN=102 B = hij
P505 PageLSN= 107 C = tuv	

P500 PageLSN= - A = abc D = mnp	P600 PageLSN=102 B = hij
P505 PageLSN= - C = tuv	P700 PageLSN= - E = pq

Log

LSN	prevLSN	tID	pID	Log entry	Type	undoNextLSN
101	-	T1000	P500	Write A "abc" -> "def"	Update	-
102	-	T ₂₀₀₀	P600	Write B "hij" -> "klm"	Update	-
103	102	T ₂₀₀₀	P500	Write D "mnp" -> "qrs"	Update	-
104	101	T ₁₀₀₀	P505	Write C "tuv" -> "wxy"	Update	-
105	103	T ₂₀₀₀			Commit	
106	105	T ₂₀₀₀			End	
107		T ₁₀₀₀		UndoT ₁₀₀₀ LSN104	CLR	101
108		T ₁₀₀₀		UndoT ₁₀₀₀ LSN101	CLR	-
109		T ₁₀₀₀			End	-

Disk

Write an END record
(explicitly mentioned in hw 4 for
the aborted transaction)

What happens if T aborts?

1. Write an “abort” log record for T
2. Follow “prevLSN” to undo all updates by T
 - Like the “UNDO” phase
 - Write the CLR records
 - Log entry like “Undo T3 LSN5”
 - Until undoNextLSN is null
3. Write an “End” log record for T

Handling Crashes during Undo

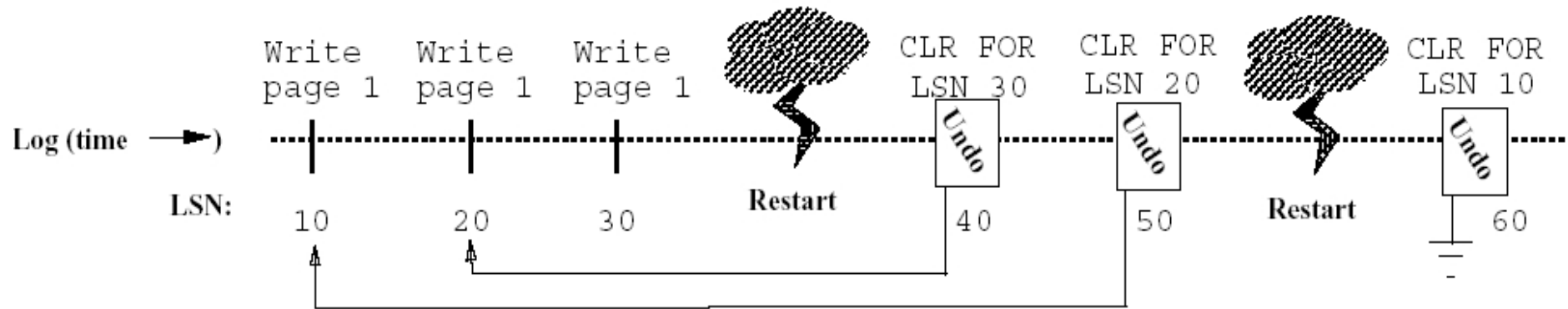


Figure 4: The Use of CLR for UNDO

[Figure 4 from Franklin97]

In general, for every single crash (even for crash during Analysis/Redo/Undo phases), start again with Analysis

If some CLR records are written to disk during an UNDO phase, then a crash happens (e.g. here LSN 40, 50 are written to disk before the second crash), then the next UNDO phase will skip undoing those CLR records.