CSE 444 Intro to Databases

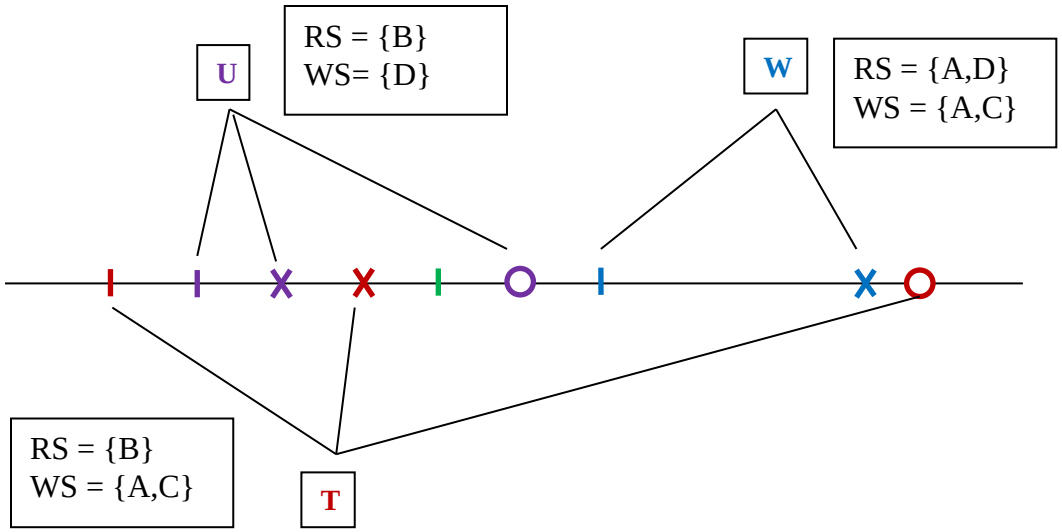
Validation and ARIES

Section 6

Validation

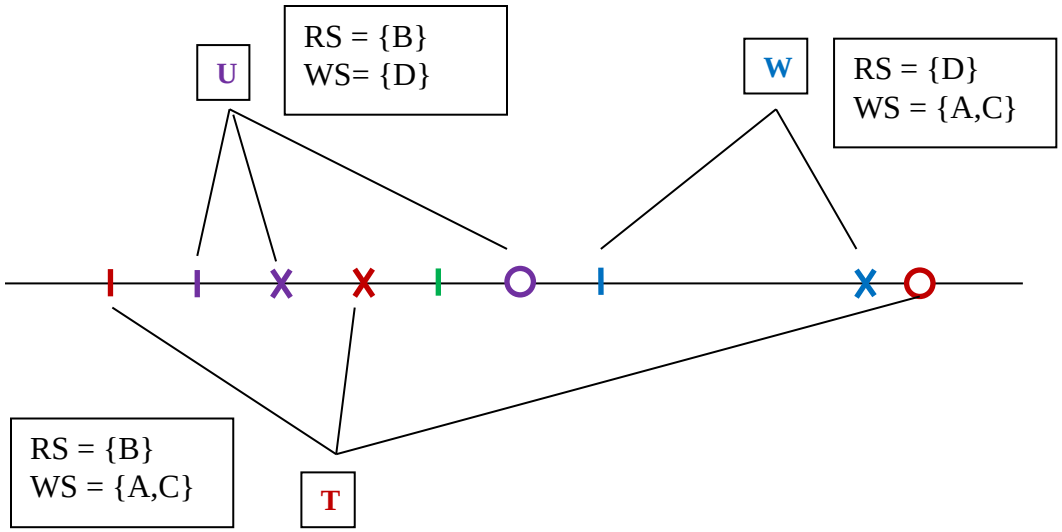
- Sets
  - START, VAL, FIN (maintained by scheduler)
  - RS and WS (told to the scheduler per Txn)
- Serial order?
- Rules
  - For any previously validated transaction U that did not finish before T started, check: RS(T) INTERSECT WS(U) = {} [for FIN(U) > START(T)]
  - For any previously validated transaction U that did not finish before T validated, check: WS(T) INTERSECT WS(U) = {} [for FIN(U) > VAL(T)]

Example 1



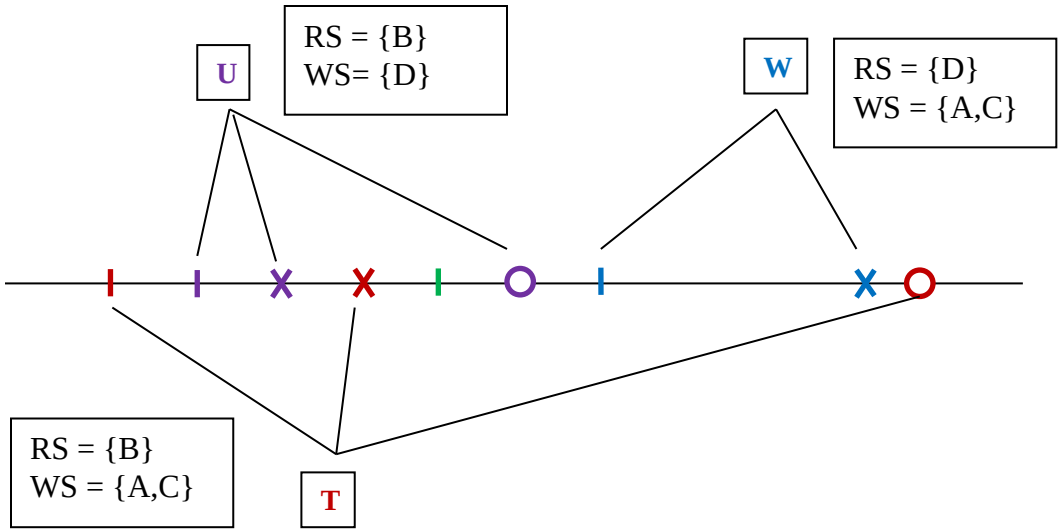
I - Start
X - Validate
O - Finish

Example 1



Validation of U: When U validates there are no other validated transactions, so there nothing to check. U validates successfully and can write a value for database element D.

Example 1



Validation of T:

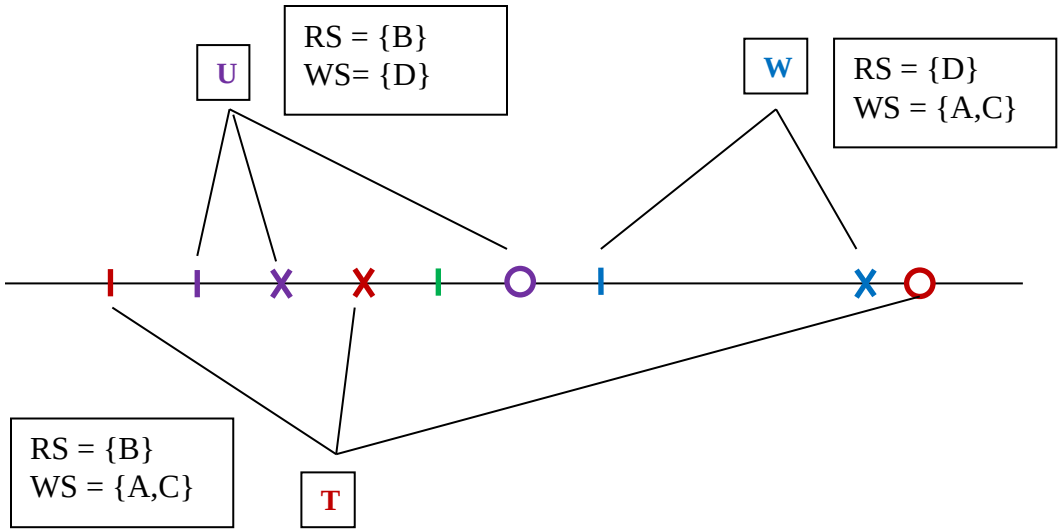
FIN(U) > START (T) check:

$RS(T) \cap WS(U) = \{B\} \cap \{D\} = \varnothing$

FIN(U) > VAL(T)

$WS(T) \cap WS(U) = \{A,C\} \cap \{D\} = \varnothing$

Example 1



I - Start
X - Validate
O - Finish

Validation of W:

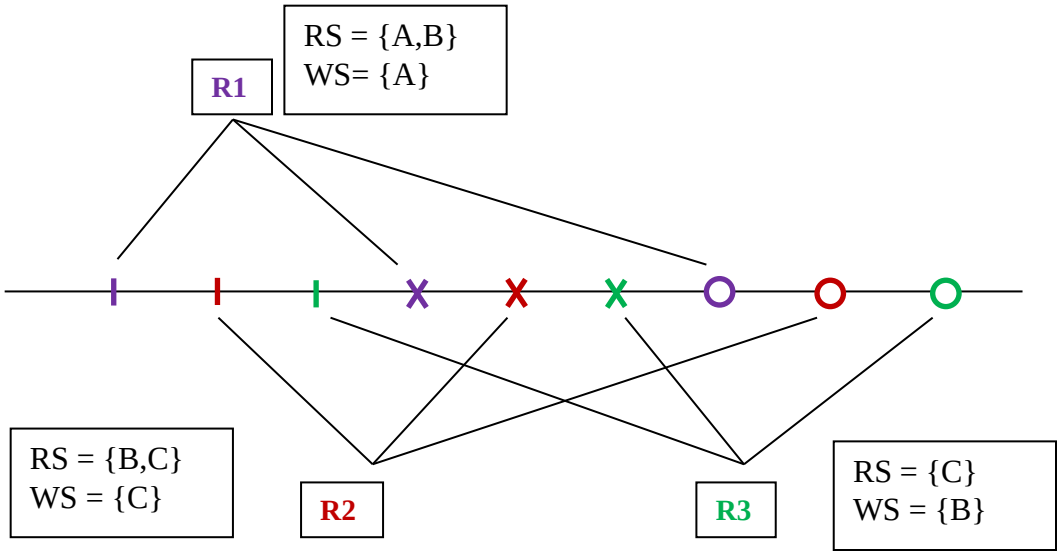
FIN(T) > START (W) check:

$RS(W) \cap WS(T) = \{D\} \cap \{A,C\} = \varnothing$

FIN(T) > VAL(W) check:

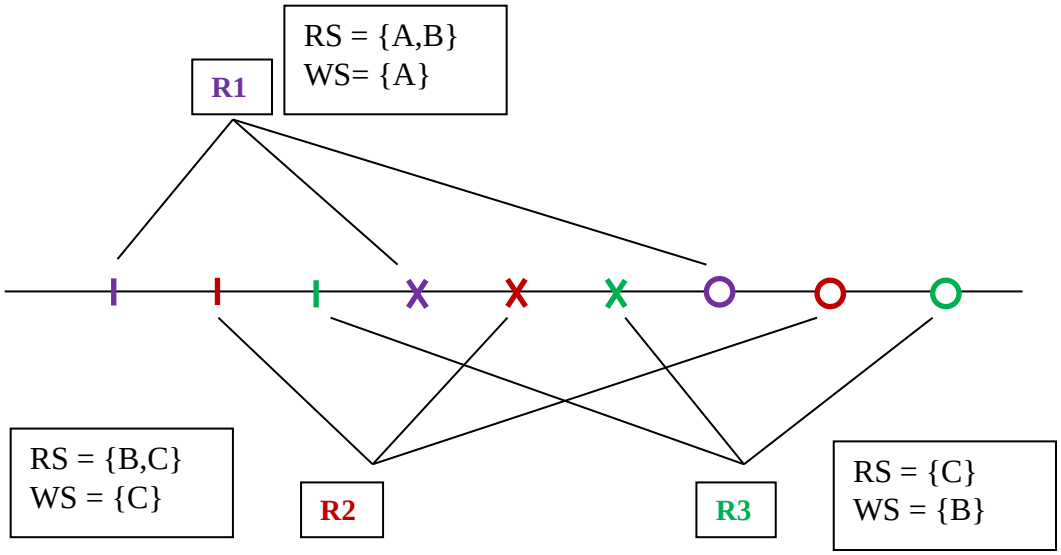
$WS(W) \cap WS(T) = \{A,C\} \cap \{A,C\} = \{A,C\}$

Example 2



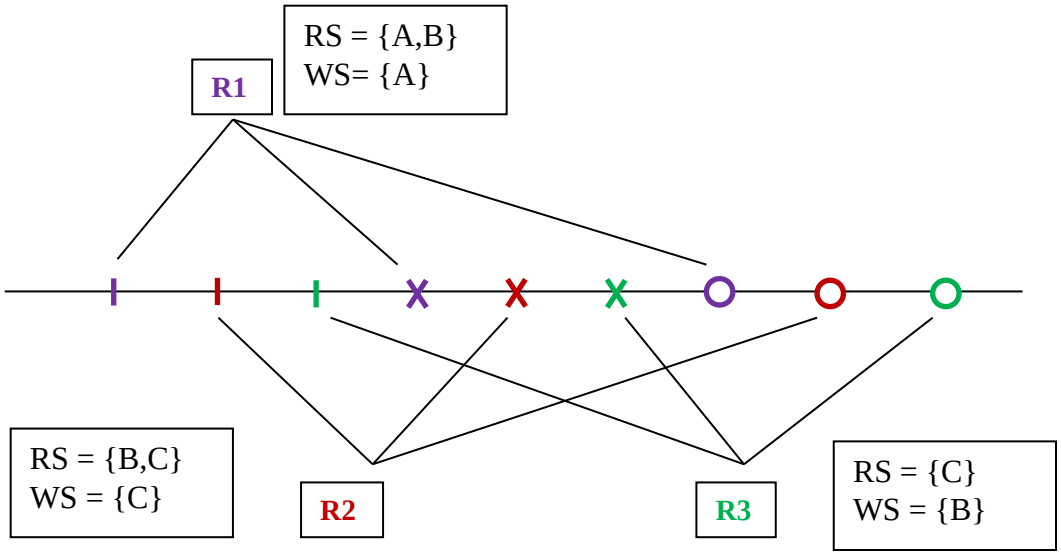
I - Start
X - Validate
O - Finish

Example 2



Validation of R1: When R1 validates there are no other validated transactions, so there nothing to check. R1 validates successfully and can write values Its elements in WS.

Example 2



Validation of R2:

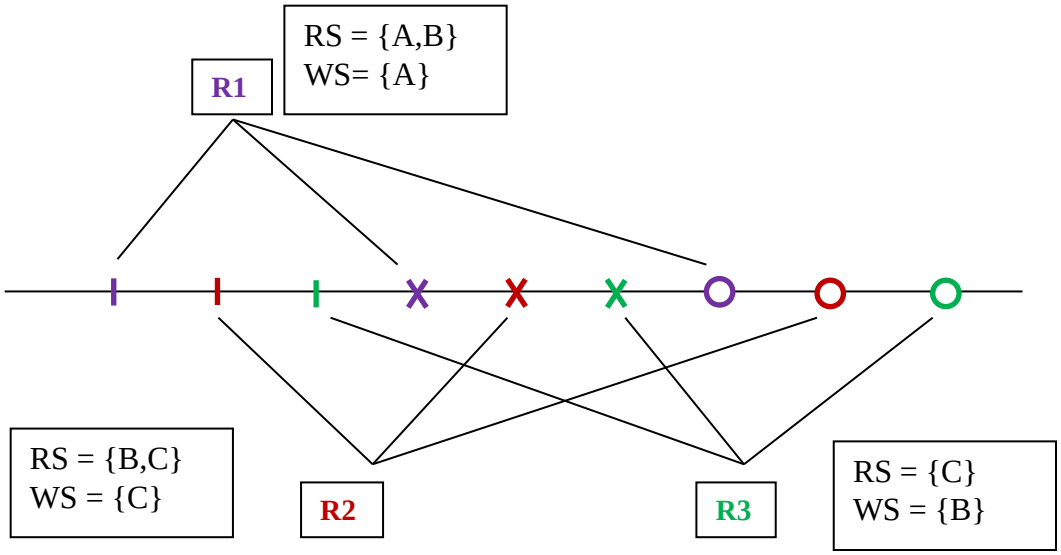
FIN(R1) > START (R2) check:

$RS(R2) \cap WS(R1) = \{B,C\} \cap \{A\} = \varnothing$

FIN(R1) > VAL(R2)

$WS(R2) \cap WS(R1) = \{C\} \cap \{A\} = \varnothing$

Example 2



I - Start
X - Validate
O - Finish

Validation of R3:

FIN(R1) > START (R3) check:

$RS(R3) \cap WS(R1) = \{C\} \cap \{A\} = \varnothing$

FIN(R1) > VAL(R3) check:

$WS(R3) \cap WS(R1) = \{B\} \cap \{A\} = \varnothing$

FIN(R2) > START (R3) check:

$RS(R3) \cap WS(R2) = \{C\} \cap \{C\} = \{C\}$

FIN(R2) > VAL(R3) check:

$WS(R3) \cap WS(R2) = \{B\} \cap \{C\} = \varnothing$

Logging

- Logical Logging
  - Logs high level information
- Physical Logging
  - Logs all information needed to recover a page
- Physiological Logging
  - Log records constrained to one page, may reflect logical operations on that page

Write-Ahead Logging (WAL)

- The Write-Ahead Logging Protocol:
  ① Must force the log record for an update *before* the corresponding data page gets to disk.
  ② Must write all log records for a Txn *before commit*.
- #1 guarantees Atomicity.
- #2 guarantees Durability.

Aries: The Big Picture: What's Stored Where

LogRecords

prevLSN
XID
type
pageID
length
offset
before-image
after-image

Data pages
each
with a
pageLSN

master record

Txn Table
lastLSN
status

Dirty Page Table
recLSN

flushedLSN

Log Records

LogRecord fields:

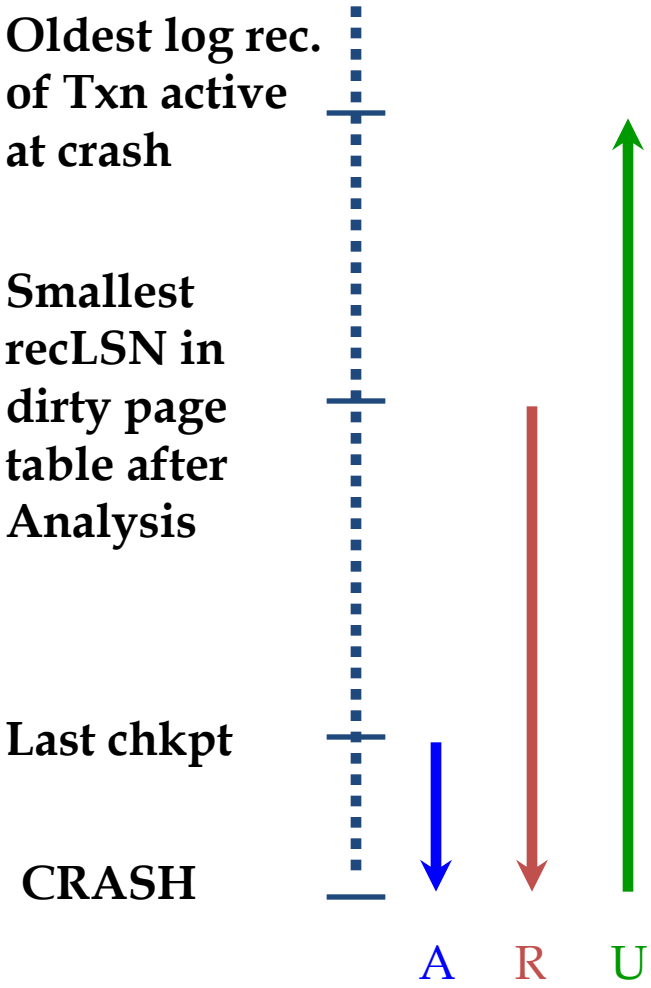
- prevLSN
- XID
- type
- pageID
- length
- offset
- before-image
- after-image

update records only

Possible log record types:

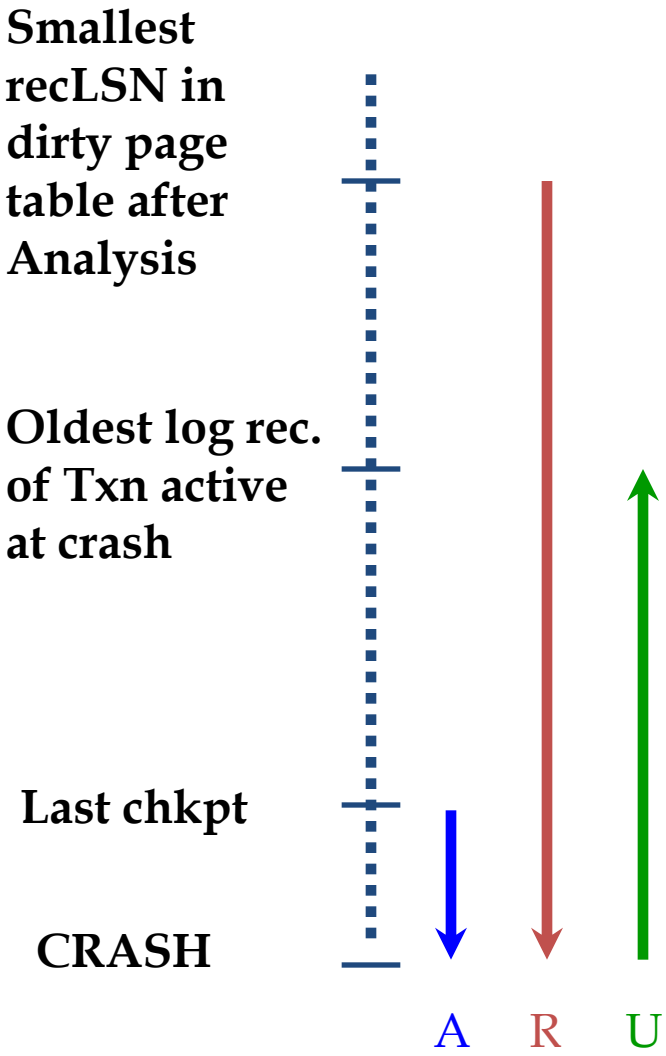
- **Update**
- **Commit**
- **Abort**
- **End** (signifies end of commit or abort)
- Compensation Log Records (CLRs)
  - for UNDO actions
  - Has undoNextLSN

Crash Recovery: Big Picture



- Start from a checkpoint (found via master record).
- Three phases. Need to:
  - Figure out which Txns committed since checkpoint, which failed (Analysis).
  - REDO ***all*** actions.
    - (repeat history)
  - UNDO effects of failed Txns.

Crash Recovery: Big Picture



- Start from a checkpoint (found via master record).
- Three phases. Need to:
  - Figure out which Txns committed since checkpoint, which failed (Analysis).
  - REDO ***all*** actions.
    - (repeat history)
  - UNDO effects of failed Txns.

Recovery: The Analysis Phase

- Reconstruct state at checkpoint.
  - via end_checkpoint record.
- Scan log forward from checkpoint.
  - End record: Remove Txn from Txn table.
  - Other records: Add Txn to Txn table, set lastLSN=LSN, change Txn status on commit.
  - Update record: If P not in Dirty Page Table (DPT),
    - Add P to DPT, set its recLSN=LSN.

Recovery: The REDO Phase

- We *repeat History* to reconstruct state at crash:
  - Reapply *all* updates (even of aborted Txns!), redo CLRs.
- Scan forward from log rec containing smallest recLSN in DPT For each CLR or update log rec LSN, REDO the action unless:
  - Affected page is not in the DPT, or
  - Affected page is in DPT, but has recLSN > LSN, or
  - pageLSN (in DB) ≥ LSN.
- To REDO an action:
  - Reapply logged action.
  - Set pageLSN to LSN. No additional logging!

Recovery: The UNDO Phase

ToUndo={ / | / a lastLSN of a "loser" Txn}

Repeat:

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

Until ToUndo is empty.

LSN	Comment	Type	prevLSN/ nextUndoLSN*	Data…
00	Begin_checkpoint			
05	End_checkpoint			
10	Update: T1 writes P5	U		
20	Update: T2 writes P3	U		
30	T1 abort	A		
40	CLR: Undo T1 LSN 10	CLR		
45	T1 end	End		
50	Update: T3 writes P1	U		
60	Update: T2 writes P5	U		
SYSTEM CRASHES				
70	CLR: Undo T2 LSN 60	CLR		
80	CLR: Undo T3 LSN 50	CLR		
85	T3 end	End		
SYSTEM CRASHES				
90	CLR: Undo T2: LSN 20			
95	T2 end			

Example

Notes:

- **End => we are done with that transaction**
- **Do abort of Txn as a special case of Undo**

LSN	Comment	Type	prevLSN/ nextUndoLSN*	Data…
00	Begin_checkpoint		NULL	
05	End_checkpoint		NULL	
10	Update: T1 writes P5	U	NULL	
20	Update: T2 writes P3	U	NULL	
30	T1 abort	A	10	
40	CLR: Undo T1 LSN 10	CLR	NULL	
45	T1 end	End	40	
50	Update: T3 writes P1	U	NULL	
60	Update: T2 writes P5	U	20	
SYSTEM CRASHES				
70	CLR: Undo T2 LSN 60	CLR	20	
80	CLR: Undo T3 LSN 50	CLR	NULL	
85	T3 end	End	80	
SYSTEM CRASHES				
90	CLR: Undo T2: LSN 20		NULL	
95	T2 end		90	

Example LSN values

nextUndoLSN is NULL
Why?

LSN	Comment	Type	prevLSN/ nextUndoLSN*	Data…
00	Begin_checkpoint		NULL	
05	End_checkpoint		NULL	
10	Update: T1 writes P5	U	NULL	
20	Update: T2 writes P3	U	NULL	
30	T1 abort	A	10	
40	CLR: Undo T1 LSN 10	CLR	NULL	
45	T1 end	End	40	
50	Update: T3 writes P1	U	NULL	
60	Update: T2 writes P5	U	20	
SYSTEM CRASHES				

Example Analysis

TT

Txn	lastLSN
T2	~~20~~ 60
T3	50

DPT

Page#	recLSN
P5	10
P3	20
P1	50

Disclaimer: The sample solution to this example has not been proven to be fully accurate.

LSN	Comment	Type	prevLSN/ nextUndoLSN*	Data…
00	Begin_checkpoint		NULL	
05	End_checkpoint		NULL	
10	Update: T1 writes P5	U	NULL	
20	Update: T2 writes P3	U	NULL	
30	T1 abort	A	10	
40	CLR: Undo T1 LSN 10	CLR	NULL	
45	T1 end	End	40	
50	Update: T3 writes P1	U	NULL	
60	Update: T2 writes P5	U	20	
SYSTEM CRASHES				

Example Redo steps

Start redo, not updated
Since pageLSN > LSN

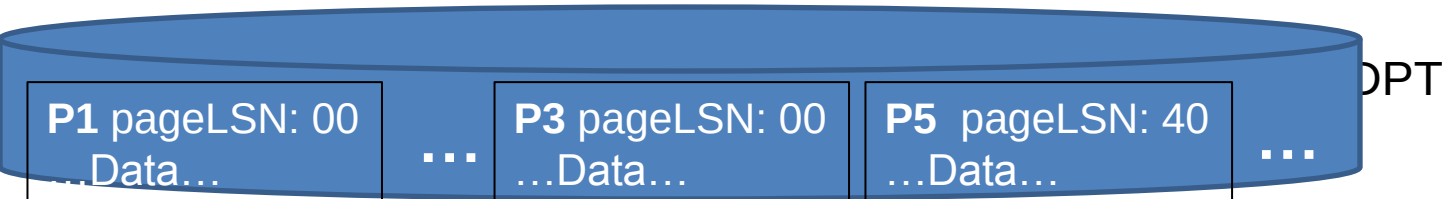
Redo P3
Update pageLSN of P3

Start redo, not updated
Since pageLSN > LSN

Redo P1
Update pageLSN of P1

Redo P5
Update pageLSN of P5

Page#	recLSN
P5 (min)	**10**
P3	20
P1	50



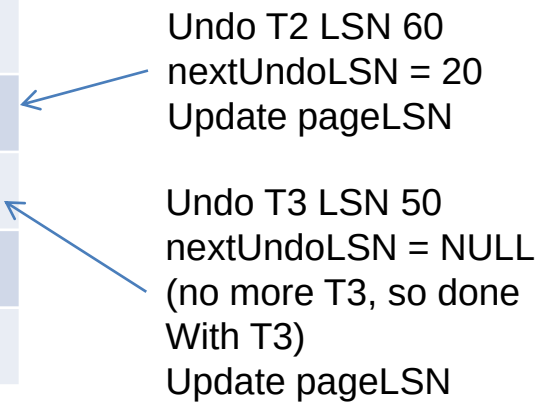
DPT

LSN	Comment	Type	prevLSN/ nextUndoLSN*	Data…
00	Begin_checkpoint		NULL	
05	End_checkpoint		NULL	
10	Update: T1 writes P5	U	NULL	
20	Update: T2 writes P3	U	NULL	
30	T1 abort	A	10	
40	CLR: Undo T1 LSN 10	CLR	NULL	
45	T1 end	End	40	
50	Update: T3 writes P1	U	NULL	
60	Update: T2 writes P5	U	20	
SYSTEM CRASHES				
70	CLR: Undo T2 LSN 60	CLR	20	
80	CLR: Undo T3 LSN 50	CLR	NULL	
85	T3 end	End	80	
SYSTEM CRASHES				

Example Undo steps

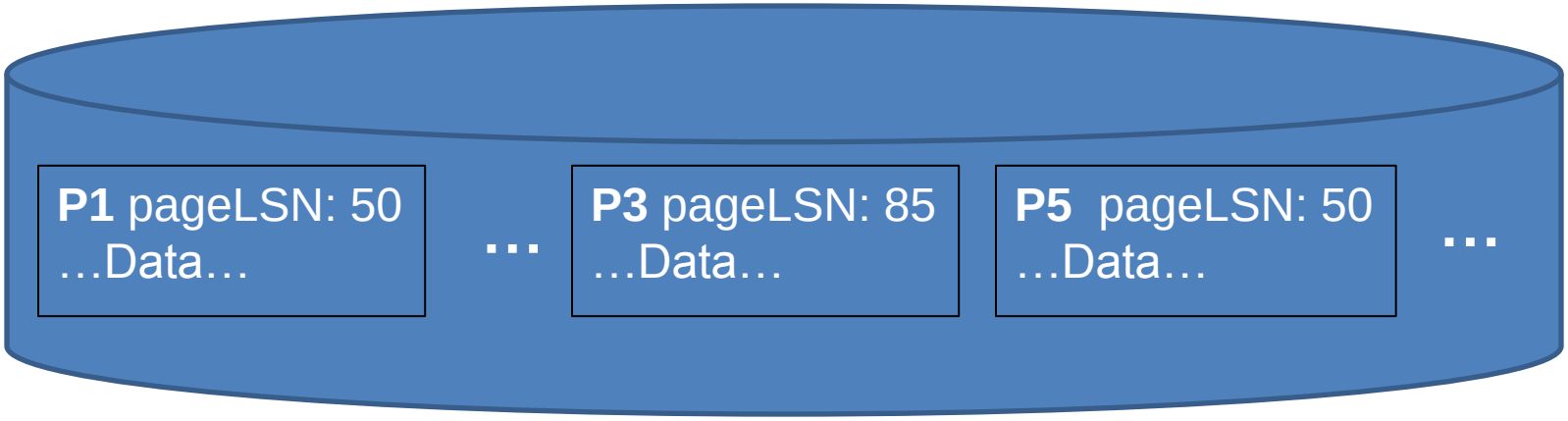
TT: aka Loser Txn

Txn	lastLSN
T2 (largest)	**~~20~~ 60**
T3	50



Disk

So lets say the disk looks like this:



That is to say:

- P1 was flushed to disk at or immediately after LSN: 50
- P3 was flushed to disk at or immediately LSN: 85
- P5 was flushed to disk at or immediately LSN: 50

LSN	Comment	Type	prevLSN/ nextUndoLSN*	Data…
00	Begin_checkpoint		NULL	
05	End_checkpoint		NULL	
10	Update: T1 writes P5	U	NULL	
20	Update: T2 writes P3	U	NULL	
30	T1 abort	A	10	
40	CLR: Undo T1 LSN 10	CLR	NULL	
45	T1 end	End	40	
50	Update: T3 writes P1	U	NULL	
60	Update: T2 writes P5	U	20	
SYSTEM CRASHES				
70	CLR: Undo T2 LSN 60	CLR	20	
80	CLR: Undo T3 LSN 50	CLR	NULL	
85	T3 end	End	80	
SYSTEM CRASHES				

Example Analysis

TT

Txn	lastLSN
T2	70

DPT

Page#	recLSN
P5	10
P3	20
P1	50

LSN	Comment	Type	prevLSN/ nextUndoLSN*	Data…
00	Begin_checkpoint		NULL	
05	End_checkpoint		NULL	
10	Update: T1 writes P5	U	NULL	
20	Update: T2 writes P3	U	NULL	
30	T1 abort	A	10	
40	CLR: Undo T1 LSN 10	CLR	NULL	
45	T1 end	End	40	
50	Update: T3 writes P1	U	NULL	
60	Update: T2 writes P5	U	20	
SYSTEM CRASHES				
70	CLR: Undo T2 LSN 60	CLR	20	
80	CLR: Undo T3 LSN 50	CLR	NULL	
85	T3 end	End	80	
SYSTEM CRASHES				

Example Redo steps

Start redo, not done
Since pageLSN > LSN

not done
Since pageLSN > LSN

not done
Since pageLSN > LSN

not done
Since pageLSN > LSN

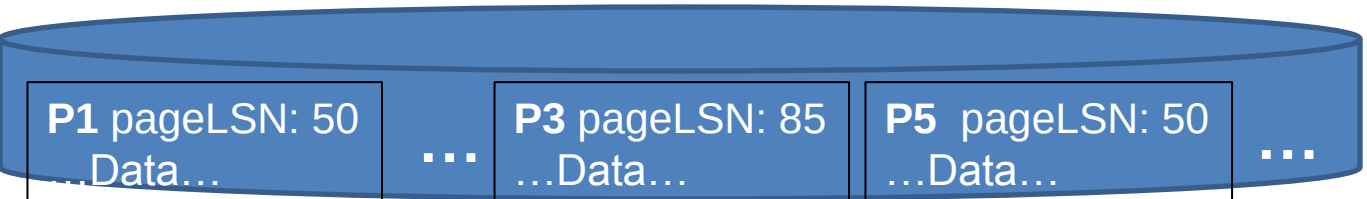
Redone
Update P5 pageLSN

Redone
Update P5 pageLSN

Redone and update LSN

DPT

Page#	recLSN
P5	10
P3	20
P1	50



LSN	Comment	Type	prevLSN/ nextUndoLSN*	Data…
00	Begin_checkpoint		NULL	
05	End_checkpoint		NULL	
10	Update: T1 writes P5	U	NULL	
20	Update: T2 writes P3	U	NULL	
30	T1 abort	A	10	
40	CLR: Undo T1 LSN 10	CLR	NULL	
45	T1 end	End	40	
50	Update: T3 writes P1	U	NULL	
60	Update: T2 writes P5	U	20	
SYSTEM CRASHES				
70	CLR: Undo T2 LSN 60	CLR	20	
80	CLR: Undo T3 LSN 50	CLR	NULL	
85	T3 end	End	80	
SYSTEM CRASHES				
90	CLR: Undo T2: LSN 20		NULL	
95	T2 end		90	

Example Undo steps

TT: aka Loser Txn

Txn	lastLSN
T2	70

Start at 70,
Go to LSN 20 and undo it

Undo 20,
Update pageLSN
no more prev T2s, hence done!

Crash During

- What does recovery manager do if we have crash at:
  - Analysis phase?
  - Redo phase?
  - Undo phase?
    - Just did an example

References

- Example for Validation taken modified from the example in chapter 18, Database Systems, 2E by H. Garcia-Molina, J. Ullman, J. Widom
- Slides taken from samples slides for chapter 18, 3E, Database Management Systems by R. Ramakrishnan and J. Gehrke
- Example for ARIES modified from undo example in chapter 18, 3E by Database Management Systems, R. Ramakrishnan and J. Gehrke