CS 186 Discussion 8

Recovery

Logistics

- Midterm 2 this Thursday (3/31) in class:
  - Review Session today: 5:30 - 7pm, 155 Dwinelle
  - One-page cheatsheet
  - Check room assignments

Recovery

- Logging enables atomicity and durability
- Works tightly with buffer pool and lock manager
- Strict 2PL

Steal / No-Force

- Steal - okay to evict dirty pages before commit
  - Watch out for:
  - Log Info:
- No-Force - can write updates to disk after commit
  - Watch out for:
  - Log Info:

Steal / No-Force

- Steal - okay to evict dirty pages before commit
  - Watch out for: Atomicity (xact aborts)
  - Log Info: UNDO
- No-Force - can write updates to disk after commit
  - Watch out for:
  - Log Info:

Steal / No-Force

- Steal - okay to evict dirty pages before commit
  - Watch out for: Atomicity (xact aborts)
  - Log Info: UNDO
- No-Force - can write updates to disk after commit
  - Watch out for: Durability (commit lost after crash)
  - Log Info: REDO

Steal / No-Force

- Steal - okay to evict dirty pages before commit
- No-Force - can write updates to disk after commit
- So when are updates to data pages (in memory):
    - Persisted to disk?
    - Logged in memory?
    - Logged on disk?

Steal / No-Force

- Steal - okay to evict dirty pages before commit
- No-Force - can write updates to disk after commit
- So when are updates to data pages (in memory):
  - Persisted to disk? On eviction
  - Logged in memory? Immediately (on update)
  - Logged on disk? On eviction OR commit (whichever comes first)

↑ (This isn’t comprehensive…)

Write-Ahead Logging

- The log record is flushed to disk whenever:
  - An update is flushed to disk
    - Why?
  - A transaction commits
    - Why?

ARIES Logging

- Log Sequence Numbers (increasing "timestamps")
- pageLSN - each page, most recent update LSN
- Record Types:
  - Update, Commit, Abort
  - Checkpoint
  - CLR (an undo)
  - End (done committing/aborting)

ARIES Logging

- Transaction Table
  - Tracks <u>active</u> xacts
  - lastLSN - most recent LSN written by xact
- Dirty Page Table
  - Tracks dirty pages in buffer
  - recLSN - first record that dirtied page
- flushedLSN - max LSN flushed so far

ARIES - Commit

1. Write commit record
2. Flush log to disk
3. Write end record

ARIES - Abort

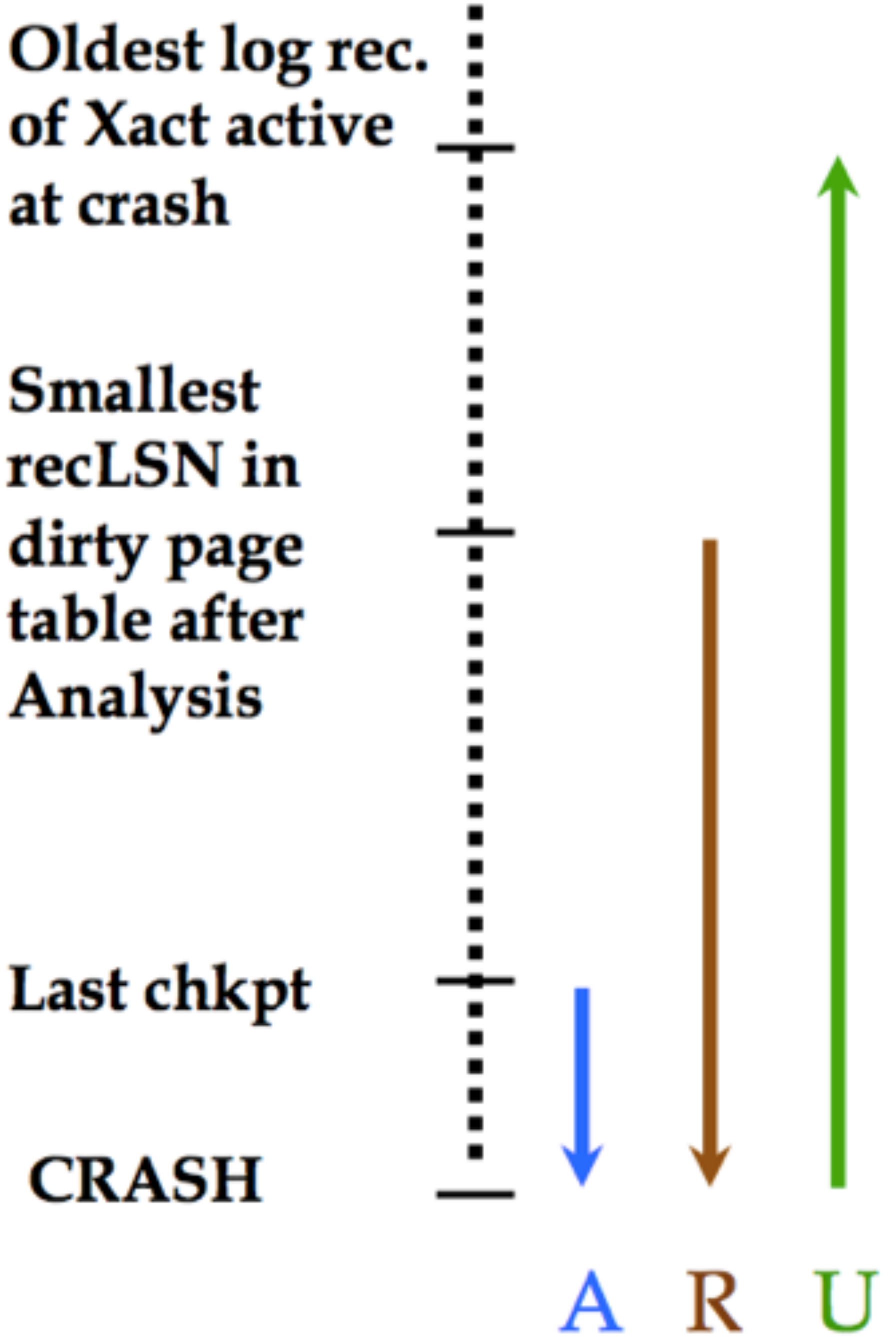
1. Write abort record
2. Rollback - fix buffer pool page with before image
   - Write CLR records per undo, update pageLSN
   - Redo CLRs if you crash
3. Write end record

ARIES - Checkpoint

- Faster recovery! Store most recent checkpoint
- Checkpoint record stores current xact table + DPT
- No need to force dirty pages to disk (recLSN)

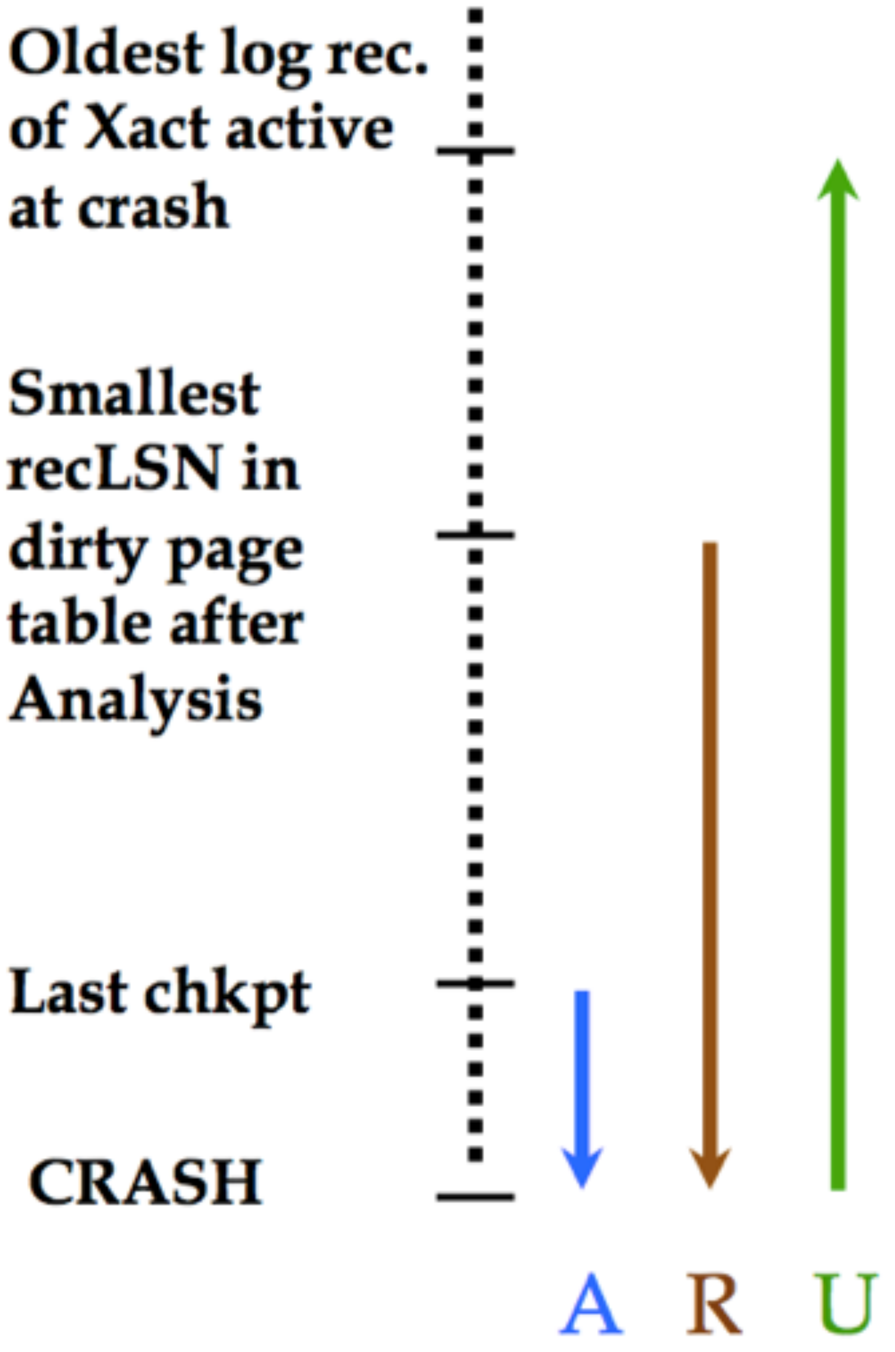
ARIES - Recovery

- If we crash:
    - What do we need to restore?
    - What do we need to undo?


Oldest log rec. of Xact active at crash
Smallest recLSN in dirty page table after Analysis
Last chkpt
CRASH
A
R
U

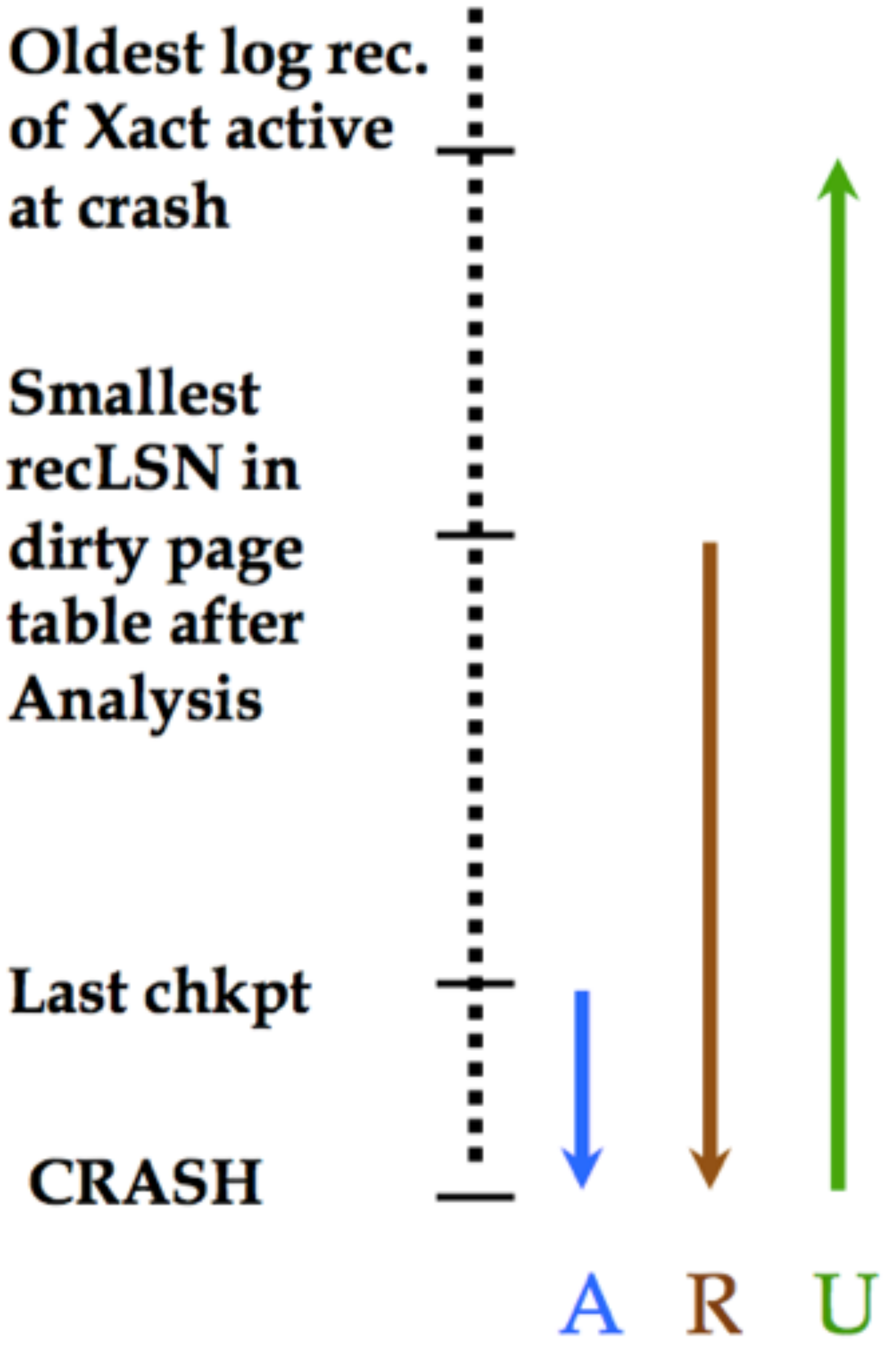
ARIES - Recovery

- If we crash:
  - What do we need to restore?
    - Anything that committed
  - What do we need to undo?
    - Anything that aborted(?)
    - or… didn't finish


Oldest log rec. of Xact active at crash
Smallest recLSN in dirty page table after Analysis
Last chkpt
CRASH
A
R
U

ARIES - Recovery

- Analysis / REDO / UNDO
  - Need to know which xacts started/committed/aborted after checkpoint
  - xact table = loser xacts
    - Remove xact if END
    - Add xact if anything else
  - Add new pages to DPT for updates (+ recLSN)


Oldest log rec. of Xact active at crash
Smallest recLSN in dirty page table after Analysis
Last chkpt
CRASH
A
R
U

Worksheet - Analysis

LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	70	Running
T2	60	Running
T3	30	Running
T4	50	Running

PageID	recLSN
P5	50
P1	40

Worksheet - Analysis

LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	90	Running
T2	60	Running
T3	30	Running
T4	50	Running

PageID	recLSN
P5	50
P1	40
P3	90

Worksheet - Analysis

LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	90	Running
T2	110	Running
T3	30	Running
T4	50	Running

PageID	recLSN
P5	50
P1	40
P3	90

Worksheet - Analysis

LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	90	Running
T2	110	Committed
T3	30	Running
T4	50	Running

PageID	recLSN
P5	50
P1	40
P3	90

Worksheet - Analysis

LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	90	Running
T2	110	Running
T3	30	Running
T4	130	Running

PageID	recLSN
P5	50
P1	40
P3	90

Worksheet - Analysis

LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	90	Running
T3	30	Running
T4	130	Running

PageID	recLSN
P5	50
P1	40
P3	90

Worksheet - Analysis

LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	90	Running
T3	30	Running
T4	130	Aborting

PageID	recLSN
P5	50
P1	40
P3	90

Worksheet - Analysis

LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	90	Running
T5	160	Running
T3	30	Running
T4	130	Aborting

PageID	recLSN
P5	50
P1	40
P3	90
P2	160

Worksheet - Analysis

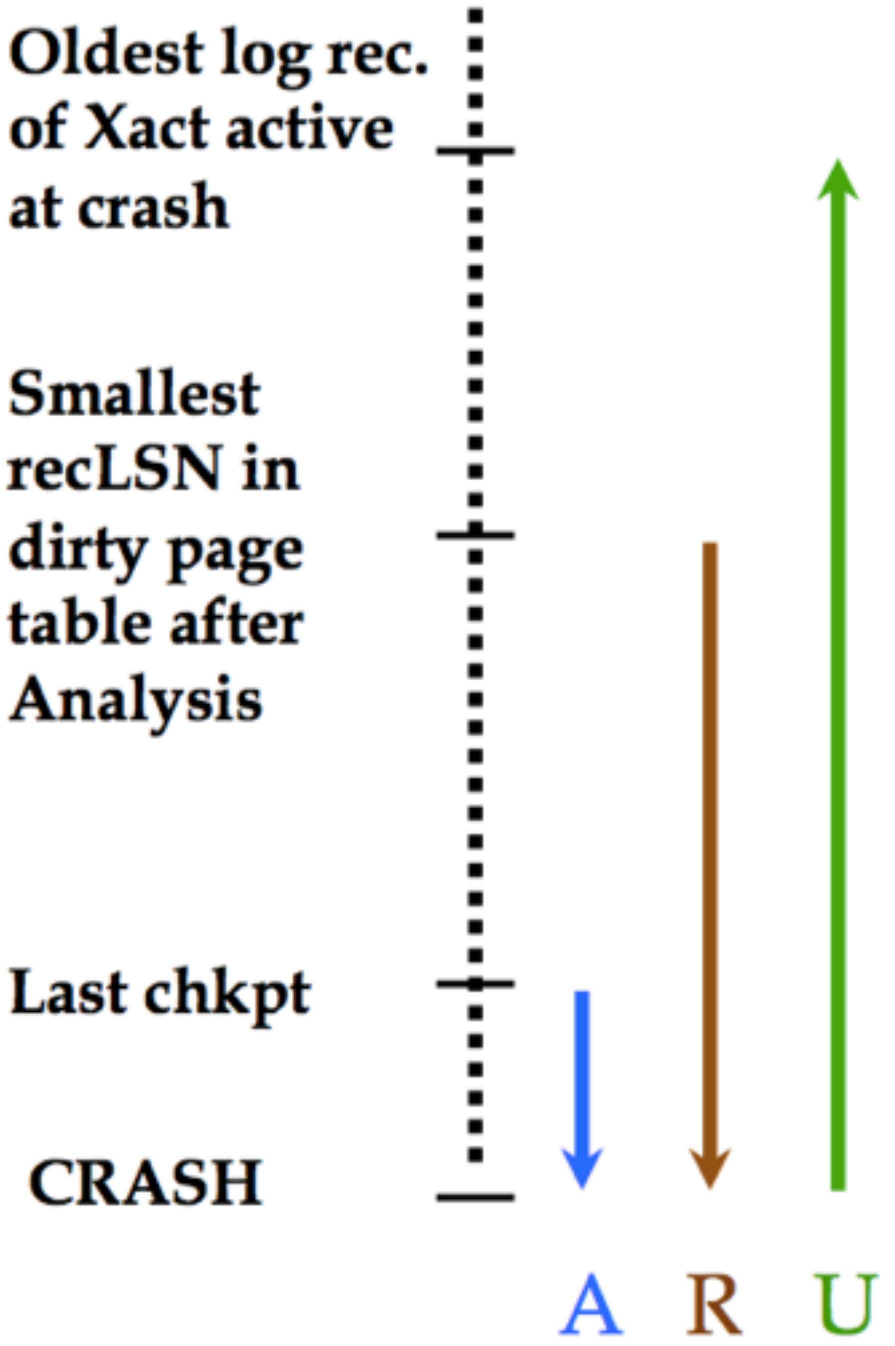
LSN	Record
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Transaction	lastLSN	Status
T1	90	Running
T5	160	Running
T3	30	Running
T4	180	Aborting

PageID	recLSN
P5	50
P1	40
P3	90
P2	160

ARIES - Recovery

- Analysis / REDO / UNDO
  - Redo everything from min recLSN, unless:
    - page not in DPT
    - recLSN of page > LSN
    - pageLSN ≥ LSN
  - Redo with after-image
  - Update pageLSNs


Oldest log rec. of Xact active at crash
Smallest recLSN in dirty page table after Analysis
Last chkpt
CRASH
A
R
U

Worksheet - REDO

LSN	Record
40	update: T4 writes P1
50	update: T4 writes P5
60	update: T2 writes P5
70	update: T1 writes P2
80	begin checkpoint
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Skip redo if:

- page not in DPT
- recLSN of page > LSN
- pageLSN ≥ LSN

PageID	recLSN
P5	50
P1	40
P3	90
P2	160

Worksheet - REDO

LSN	Record
40	update: T4 writes P1
50	update: T4 writes P5
60	update: T2 writes P5
70	update: T1 writes P2
80	begin checkpoint
90	update: T1 writes P3
100	end checkpoint
110	update: T2 writes P3
120	T2 commit
130	update: T4 writes P1
140	T2 end
150	T4 abort
160	update: T5 writes P2
180	CLR: undo T4 LSN 130

Skip redo if:

- page not in DPT
- recLSN of page > LSN
- pageLSN ≥ LSN

PageID	recLSN
P5	50
P1	40
P3	90
P2	160

Worksheet - 1ab

Transaction	lastLSN	Status

PageID	recLSN

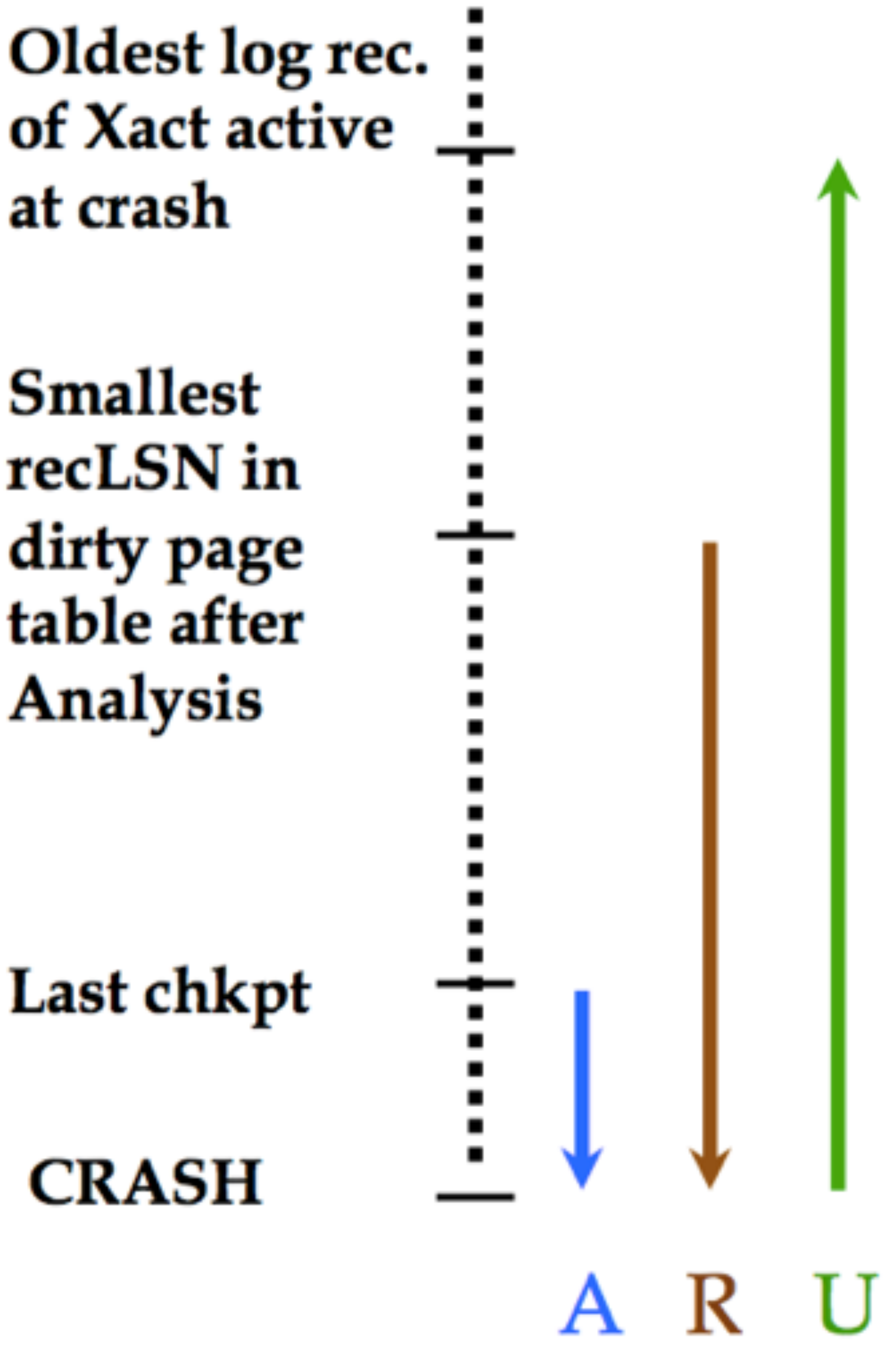
Worksheet - 1ab

Transaction	lastLSN	Status
T3	70	Running
T2	80	Aborting

PageID	recLSN
P1	10
P2	70
P3	20
P4	40

ARIES - Recovery

- Analysis / REDO / UNDO
  - Undo everything in xact table -> abort these xacts
  - Performance optimization:
    - Repeat with decreasing LSN, until table empty
    - Undo updates and continue writing CLRs


Oldest log rec. of Xact active at crash
Smallest recLSN in dirty page table after Analysis
Last chkpt
CRASH
A
R
U

Worksheet - UNDO

Xact	lastLSN	Status
T3	70	Running
T2	80	Aborting

PageID	recLSN
P1	10
P2	70
P3	20
P4	40

- Start from max lastLSN, work backwards
- ToUndo = {70, 80}
  - prevLSN of 80?

Worksheet - UNDO

Xact	lastLSN	Status
T3	70	Running
T2	80	Aborting

PageID	recLSN
P1	10
P2	70
P3	20
P4	40

- Start from max lastLSN, work backwards
- ToUndo = {50, 70}

Worksheet - UNDO

Xact	lastLSN	Status
T3	70	Running
T2	80	Aborting

PageID	recLSN
P1	10
P2	70
P3	20
P4	40

- Start from max lastLSN, work backwards
- ToUndo = {40, 50}

LSN	Record
100	CLR: T3 LSN = 70; undoNextLSN = 40

Worksheet - UNDO

Xact	lastLSN	Status
T3	70	Running
T2	80	Aborting

PageID	recLSN
P1	10
P2	70
P3	20
P4	40

- Start from max lastLSN, work backwards
- ToUndo = {20, 40}

LSN	Record
100	CLR: T3 LSN = 70; undoNextLSN = 40
110	CLR: T2 LSN = 50; undoNextLSN = 20

Worksheet - UNDO

Xact	lastLSN	Status
T3	70	Running
T2	80	Aborting

- Start from max lastLSN, work backwards
- ToUndo = {20}

PageID	recLSN
P1	10
P2	70
P3	20
P4	40

LSN	Record
100	CLR: T3 LSN = 70; undoNextLSN = 40
110	CLR: T2 LSN = 50; undoNextLSN = 20
120	CLR: T3 LSN = 40; undoNextLSN = null
130	T3 end

Worksheet - UNDO

Xact	lastLSN	Status
T3	70	Running
T2	80	Aborting

PageID	recLSN
P1	10
P2	70
P3	20
P4	40

- Start from max lastLSN, work backwards
- ToUndo = {}

LSN	Record
100	CLR: T3 LSN = 70; undoNextLSN = 40
110	CLR: T2 LSN = 50; undoNextLSN = 20
120	CLR: T3 LSN = 40; undoNextLSN = null
130	T3 end
140	CLR: T2 LSN = 20; undoNextLSN = null
150	T2 end