Recovery

R&G Chapter 18

(slides adapted from content by J.Gehrke, J.Shanmugasundaram, and/or C.Koch)

Announcements

- Project 3 to be posted by Tuesday. Due mid-April
- Homework 6 to be posted Tonight. Due Monday.

- Project 2 Extra Credit: Schedule a 20 min meeting
  - http://doodle.com/okennedy

Recap A.C.I.D.

- **Atomicity**: All actions in a transaction happen, or none happen.
- **Consistency**: If the transaction maintains consistency, and the DB starts consistent, then the database ends consistent.
- **Isolation**: The execution of one transaction is isolated from all other transactions.
- **Durability**: If a transaction commits, its effects persist.

Recap: Write-Ahead Logging

- The write-ahead logging protocol
  - Force the log record for an update <u>before</u> the corresponding data page is <u>written</u>.
    - Guarantees Atomicity (REDO).
  - Force all log records for a transaction <u>before</u> the transaction <u>commits</u>.
    - Guarantees Durability (UNDO).

Recap: The Big Picture

The Log	The DB (Disk)	The DB (Ram)

Log Records
`Prev LSN`
`XID`
`Record Type`
`Page ID`
`Length`
`Offset`
`Before-Image`
`After-Image`
	Data Pages
	(each with a PageLSN)
	
	Master Record
		Transaction Table
		`XID`
		`LastLSN`
		`Status`
		
		Dirty Page Table
		`RecLSN`
		
		Flushed LSN

Checkpointing

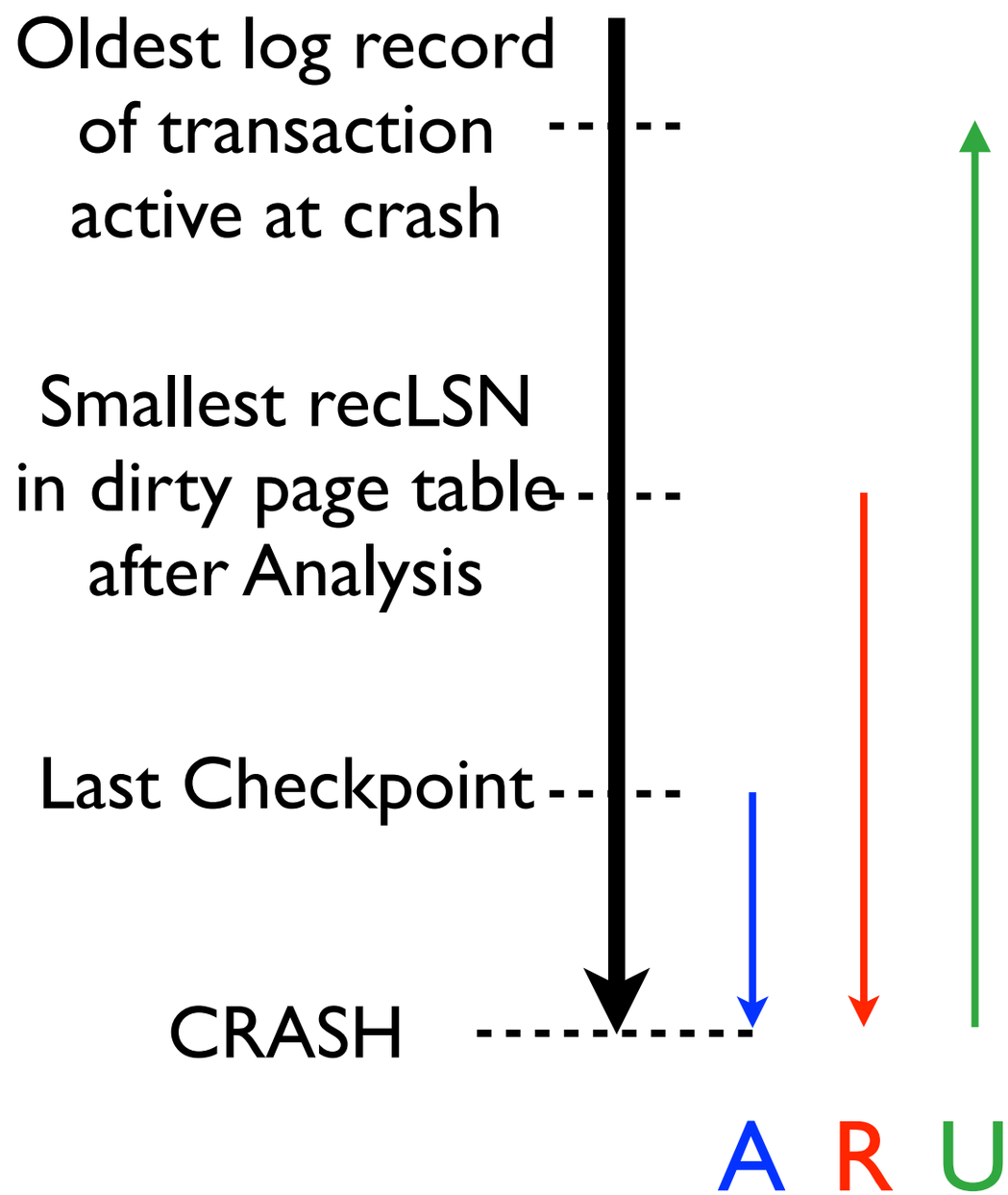
- Log grows indefinitely; Need a way to speed up the recovery process.
- Periodically, the DBMS creates a checkpoint.
  - **begin_checkpoint** record indicates when the checkpoint began.
  - **end_checkpoint** record contains the current transaction table and the dirty page table.

Checkpointing

- This is a ‘fuzzy’ checkpoint
- Other transactions continue to run, so the tables are only accurate as of the **begin_checkpoint** record.
- Checkpoints don’t force data to disk
  - Recovery process needs to go to oldest unwritten change as of the checkpoint.
- Store the LSN of the most recent checkpoint record in a safe place on disk (the master record).

Crash Recovery

- Start from checkpoint stored in master record.
- Analysis: Figure out which transactions committed since checkpoint.
- Redo all actions
- Undo effects of failed transactions.



Recovery: Analysis

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

Recovery: REDO

- Repeat History to reconstruct state at crash
- Reapply **all** updates, even those of aborted transactions. Redo CLRs as they appear.
- REDO actions by:
  - Reapplying logged action.
  - Set pageLSN to LSN.
  - Don't need to log anything (already logged)

Recovery: REDO

- Scan forward from the log record containing the smallest recLSN in the DPT.
- For each CLR or update log record LSN, REDO the action unless:
  - Affected page not in the DPT, or
  - Affected page is in DPT but has recLSN > LSN, or
  - pageLSN (in DB) ≥ LSN.

Recovery: UNDO

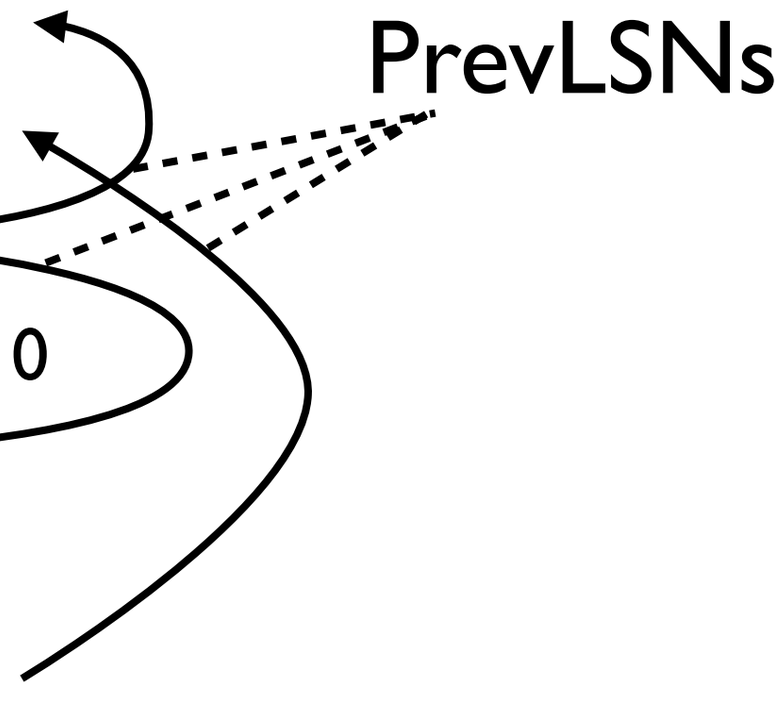
- Define a priority queue:
  - ToUndo := {all lastLSNs of undone xacts}
- Choose largest LSN in ToUndo
  - If this LSN is the last CLR for an xact
    - Write an END record
  - If this LSN is a non-terminal CLR
    - Add the CLRs' undonextLSN to ToUNDO

Recovery: UNDO

- Choose largest LSN in ToUndo
  - …
  - Otherwise, this LSN is an update.
    - Undo the update, write a CLR
    - Add the update's prevLSN to ToUndo if it exists.
- Repeat until ToUndo is empty

Recovery Example

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5



CRASH! Restart!

Analysis

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

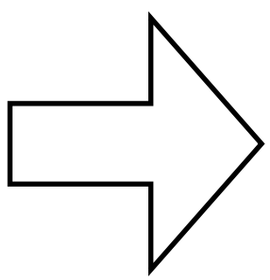
T1; <0

T2; <0

DP Table

Analysis

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5



Xact Table

T1; <0

T2; <0

DP Table

Analysis

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T1; 10

T2; <0

DP Table

P5; 10

Analysis

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T1; 10

T2; 20

DP Table

P5; 10

P3; 20

Analysis

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T1; 10

T2; 20

DP Table

P5; 10

P3; 20

Analysis

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 20

DP Table

P5; 10

P3; 20

Analysis

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 20

T3; 50

DP Table

P5; 10

P3; 20

P1; 50

Analysis

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 60

T3; 50

DP Table

P5; 10

P3; 20

P1; 50

Redo

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 60

T3; 50

DP Table

P5; 10

P3; 20

P1; 50

Redo

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 60

T3; 50

DP Table

P5; 10

P3; 20

P1; 50

Redo

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 60

T3; 50

DP Table

P5; 10

P3; 20

P1; 50

Redo

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 60

T3; 50

DP Table

P5; 10

P3; 20

P1; 50

Redo

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 60

T3; 50

DP Table

P5; 10

P3; 20

P1; 50

Redo

LSN	Log
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40	CLR Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5

Xact Table

T2; 60

T3; 50

DP Table

P5; 10

P3; 20

P1; 50

Undo

LSN	Log
00,05	begin_checkpoint, end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40, 45	CLR Undo T1 LSN 10; T1 End
50	update: T3 writes P1
60	update: T2 writes P5
70	CRASH

Xact Table

T2; 60

T3; 50

ToUndo

60

50

Undo

LSN	Log
00,05	begin_checkpoint, end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40, 45	CLR Undo T1 LSN 10; T1 End
50	update: T3 writes P1
60	update: T2 writes P5
70	CRASH
80	CLR: Undo T2, LSN 60

Xact Table

T2; 60

T3; 50

ToUndo

50

20

Undo

LSN	Log
00,05	begin_checkpoint, end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40, 45	CLR Undo T1 LSN 10; T1 End
50	update: T3 writes P1
60	update: T2 writes P5
70	CRASH
80	CLR: Undo T2, LSN 60
90,95	CLR: Undo T3, LSN 50; T3 End

Xact Table

T2; 60

T3; 50

ToUndo

20

Undo

LSN	Log
00,05	begin_checkpoint, end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40, 45	CLR Undo T1 LSN 10; T1 End
50	update: T3 writes P1
60	update: T2 writes P5
70	CRASH
80	CLR: Undo T2, LSN 60
90,95	CLR: Undo T3, LSN 50; T3 End

CRASH!

Xact Table

T2; 60

T3; 50

ToUndo

20

Undo

LSN	Log
00,05	begin_checkpoint, end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40, 45	CLR Undo T1 LSN 10; T1 End
50	update: T3 writes P1
60	update: T2 writes P5
70	CRASH
80	CLR: Undo T2, LSN 60
90,95	CLR: Undo T3, LSN 50; T3 End
100	CRASH

Xact Table

T2; 60

T3; 50

ToUndo

20

Undo

LSN	Log
00,05	begin_checkpoint, end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 Abort
40, 45	CLR Undo T1 LSN 10; T1 End
50	update: T3 writes P1
60	update: T2 writes P5
70	CRASH
80	CLR: Undo T2, LSN 60
90,95	CLR: Undo T3, LSN 50; T3 End
100	CRASH
110	CLR: Undo T2, LSN 20; T2 End

Xact Table

T2; 60

T3; 50

ToUndo

Other Crash Issues

- What happens if the system crashes during analysis?
- What happens if the system crashes during redo?

- How do you limit the amount of work in REDO?
- How do you limit the amount of work in UNDO?