

# CS660: Intro to Database Systems

## Class 26: More Recovery & NoSQL

Instructor: Manos Athanassoulis

<https://bu-disc.github.io/CS660/>

# Course Evaluation

12:30-12:45 course evaluation

<https://tinyurl.com/CS660-F23-CourseEval>

if the above does not work:

<https://go.blueja.io/inAWTDZkT0CDuuMCUTba5g>



Recovery

# **EXAMPLE 1**

# Example 1

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, prevLSN=NULL
15	<b>Update</b> , T1, P2, prevLSN=10
20	<b>Commit</b> , T1, prevLSN=15
	CRASH



## Active Transaction Table

[xact\_id, status, lastLSN]

## Dirty Page Table

[page\_id, recLSN]

**Master Record:**

*last checkpoint at LSN 00*

# Example 1: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, prevLSN=NULL
15	<b>Update</b> , T1, P2, prevLSN=10
20	<b>Commit</b> , T1, prevLSN=15



## Active Transaction Table

[xact\_id, status, lastLSN]

## Dirty Page Table

[page\_id, recLSN]

**Master Record:**

*last checkpoint at LSN 00*

# Example 1: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, prevLSN=NULL
15	<b>Update</b> , T1, P2, prevLSN=10
20	<b>Commit</b> , T1, prevLSN=15



**Active Transaction Table**

T1, running, 10

**Dirty Page Table**

P1, 10

**Master Record:**

*last checkpoint at LSN 00*

# Example 1: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, prevLSN=NULL
15	<b>Update</b> , T1, P2, prevLSN=10
20	<b>Commit</b> , T1, prevLSN=15



## Active Transaction Table

T1, running, ~~10~~ 15

## Dirty Page Table

P1, 10

P2, 15

**Master Record:**

*last checkpoint at LSN 00*

# Example 1: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, prevLSN=NULL
15	<b>Update</b> , T1, P2, prevLSN=10
20	<b>Commit</b> , T1, prevLSN=15

**Master Record:**  
*last checkpoint at LSN 00*



## Active Transaction Table

T1, ~~running, 15~~ committing, 20

## Dirty Page Table

P1, 10

P2, 15

Analysis phase done!

→ need to REDO from oldest recLSN (10)

→ T1 is committed (need to add End)

→ nothing to UNDO



# Example 1: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, prevLSN=NULL
15	<b>Update</b> , T1, P2, prevLSN=10
20	<b>Commit</b> , T1, prevLSN=15



## Active Transaction Table

T1, ~~running~~, 15 committing, 20

## Dirty Page Table

P1, 10

P2, 15

Start from LSN 10:

Read P1

if (pageLSN(P1) < 10):

redo LSN 10

set pageLSN(P1)=10

# Example 1: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, prevLSN=NULL
15	<i>Update</i> , T1, P2, prevLSN=10
20	<i>Commit</i> , T1, prevLSN=15



## Active Transaction Table

T1, ~~running, 15~~ committing, 20

## Dirty Page Table

P1, 10

P2, 15

Move to LSN 15:

Read P2

if (pageLSN(P2) < 15):

redo LSN 15

set pageLSN(P2)=15

# Example 1: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, prevLSN=NULL
15	<i>Update</i> , T1, P2, prevLSN=10
20	<i>Commit</i> , T1, prevLSN=15



## Active Transaction Table

T1, ~~running~~, 15 committing, 20

## Dirty Page Table

P1, 10

P2, 15

Move to LSN 20:

Prepare End Record

# Example 1: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, prevLSN=NULL
15	<i>Update</i> , T1, P2, prevLSN=10
20	<i>Commit</i> , T1, prevLSN=15
25	<i>End</i> , T1, prevLSN=20



Active Transaction Table  
~~T1, committing, 20~~

Dirty Page Table  
 P1, 10  
 P2, 15

Add End to LSN 25: Remove T1 from ATT

Redo phase done!  
 → nothing to UNDO  
 → Recovery is completed!

Recovery

## **EXAMPLE 2**

# Example 2

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20
	CRASH



## Active Transaction Table

[xact\_id, status, lastLSN]

## Dirty Page Table

[page\_id, recLSN]

**Master Record:**

*last checkpoint at LSN 00*

# Example 2: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

[xact\_id, status, lastLSN]

## Dirty Page Table

[page\_id, recLSN]

**Master Record:**

*last checkpoint at LSN 00*

# Example 2: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



**Active Transaction Table**

T1, running, 10

**Dirty Page Table**

P1, 10

**Master Record:**

*last checkpoint at LSN 00*



# Example 2: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

T1, running, 15

## Dirty Page Table

P1, 10

P2, 15

**Master Record:**

*last checkpoint at LSN 00*

# Example 2: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

T1, running, 15  
T2, running, 20

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

**Master Record:**

*last checkpoint at LSN 00*

# Example 2: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

T1, ~~running, 15~~ committing, 25  
T2, running, 20

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

**Master Record:**

*last checkpoint at LSN 00*

# Example 2: Analysis Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20

**Master Record:**  
*last checkpoint at LSN 00*



## Active Transaction Table

T1, committing, 25  
 T2, running, 20 30

## Dirty Page Table

P1, 10  
 P2, 15  
 P3, 20

Analysis phase done!  
 → need to REDO from oldest recLSN (10)  
 → T1 is committed (need to add End)  
 → need to undo T2: ToUndo={30}

# Example 2: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

T1, committing, 25  
T2, running, 30

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

Start from LSN 10:

Read P1

if (pageLSN(P1) < 10):

redo LSN 10 (set value to ZZZ)

set pageLSN(P1)=10

# Example 2: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<i>Update</i> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<i>Commit</i> , T1, prevLSN=15
30	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

T1, committing, 25  
T2, running, 30

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

Move to LSN 15:

Read P2

if (pageLSN(P2) < 15):

redo LSN 15 (set value to XXX)

set pageLSN(P2)=15

# Example 2: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<i>Update</i> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<i>Commit</i> , T1, prevLSN=15
30	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

T1, committing, 25  
T2, running, 30

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

Move to LSN 20:

Read P3

if (pageLSN(P3) < 20):

redo LSN 20 (set value to PPP)

set pageLSN(P3)=20

# Example 2: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<i>Update</i> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<i>Commit</i> , T1, prevLSN=15
30	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

T1, committing, 25

T2, running, 30

## Dirty Page Table

P1, 10

P2, 15

P3, 20

Move to LSN 25:      Prepare End record



# Example 2: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20



## Active Transaction Table

T1, committing, 25  
T2, running, 30

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

Move to LSN 30:

Read P1

if (pageLSN(P1) < 30):

redo LSN 30 (set value to TTT)

set pageLSN(P1)=30

# Example 2: Redo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20
35	<b>End</b> , T1, prevLSN=25



## Active Transaction Table

~~T1, committing, 25~~  
T2, running, 30

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

Redo phase done!

→ need to UNDO T2: ToUndo={30}

Add End to 35:

Remove T1 from ATT

# Example 2: Undo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<i>Update</i> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<i>Commit</i> , T1, prevLSN=15
30	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=20
35	<i>End</i> , T1, prevLSN=25



## Active Transaction Table

~~T1, committing, 25~~  
T2, running, 30

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

ToUndo={30}

# Example 2: Undo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<i>Update</i> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<i>Commit</i> , T1, prevLSN=15
30	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=20
35	<i>End</i> , T1, prevLSN=25
40	<i>CLR</i> , T2, Undo LSN 30, P1 (ZZZ), undoNextLSN=20



## Active Transaction Table

~~T1, committing, 25~~  
T2, running, 30

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

ToUndo={30}

Undo LSN 30:

write CLR for P1; set ZZZ to P1  
ToUndo ← 20

# Example 2: Undo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<i>Update</i> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<i>Commit</i> , T1, prevLSN=15
30	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=20
35	<i>End</i> , T1, prevLSN=25
40	<i>CLR</i> , T2 Undo LSN 30, P1 (ZZZ), undoNextLSN=20
45	<i>CLR</i> , T2 Undo LSN 20, P3 (LLL), undoNextLSN=NULL



## Active Transaction Table

~~T1, committing, 25~~  
T2, running, 30

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

ToUndo={20}

Undo LSN 20:

write CLR for P3; set LLL to P3  
ToUndo  $\leftarrow \emptyset$

# Example 2: Undo Phase

LSN	LOG
00	Begin Checkpoint
05	End Checkpoint (empty ATT, DPT)
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
20	<b>Update</b> , T2, P3, old=LLL, new=PPP, prevLSN=NULL
25	<b>Commit</b> , T1, prevLSN=15
30	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=20
35	<b>End</b> , T1, prevLSN=25
40	<b>CLR</b> , T2 Undo LSN 30, P1 (ZZZ), undoNextLSN=20
45	<b>CLR</b> , T2 Undo LSN 20, P3 (LLL), undoNextLSN=NULL
50	<b>End</b> , T2, prevLSN=45



## Active Transaction Table

~~T1, committing, 25~~  
~~T2, running, 30~~

## Dirty Page Table

P1, 10  
P2, 15  
P3, 20

ToUndo={∅}

Write End for T2 & Remove T2 from ATT

Recovery completed!

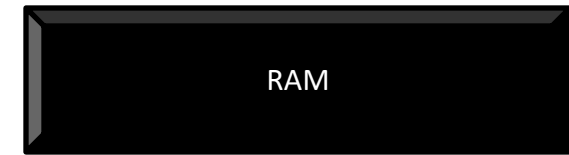
Recovery

## **EXAMPLE 3**

# Example 3

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
	<b>CRASH</b>

**Master Record:**  
*last checkpoint at LSN 20*



**Active Transaction Table**  
 [xact\_id, status, lastLSN]

**Dirty Page Table**  
 [page\_id, recLSN]



# Example 3: Analysis Phase

Master Record:  
last checkpoint at LSN 20

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15



Active Transaction Table  
[xact\_id, status, lastLSN]

Dirty Page Table  
[page\_id, recLSN]

# Example 3: Analysis Phase

Master Record:  
last checkpoint at LSN 20

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15



## Active Transaction Table

T1, running, 10  
T2, running, 15

## Dirty Page Table

P1, 10  
P3, 15

load ATT & DPT from checkpoint!

# Example 3: Analysis Phase

Master Record:  
last checkpoint at LSN 20

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15



## Active Transaction Table

T1, running, ~~10~~ 30  
T2, running, 15

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

# Example 3: Analysis Phase

Master Record:  
last checkpoint at LSN 20

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15



## Active Transaction Table

T1, ~~running, 30~~ committing, 35  
T2, running, 15

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

# Example 3: Analysis Phase

Master Record:  
last checkpoint at LSN 20

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15



## Active Transaction Table

T1, committing, 35  
T2, running, 40

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

# Example 3: Analysis Phase

Master Record:  
last checkpoint at LSN 20

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15



## Active Transaction Table

T1, committing, 35

T2, ~~running, 40~~ aborting, 45

## Dirty Page Table

P1, 10

P3, 15

P2, 30

# Example 3: Analysis Phase

Master Record:  
last checkpoint at LSN 20

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15



## Active Transaction Table

T1, committing, 35  
T2, aborting, 45 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

Analysis phase done!

- need to REDO from oldest recLSN (10)
- T1 is committed (need to add End)
- need to (continue) undo T2: ToUndo={50}

# Example 3: Redo Phase

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:  
last checkpoint at LSN 20



## Active Transaction Table

T1, committing, 35  
T2, aborting, 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

Start from LSN 10:

- Read P1
- if (pageLSN(P1) < 10):
  - redo LSN 10 (set value to ZZZ)
  - set pageLSN(P1)=10



# Example 3: Redo Phase

LSN	LOG
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<i>Commit</i> , T1, prevLSN=30
40	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<i>Abort</i> , T2, prevLSN=40
50	<i>CLR</i> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:  
last checkpoint at LSN 20



## Active Transaction Table

T1, committing, 35  
T2, aborting, 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

Move to LSN 15:

Read P3

if (pageLSN(P3) < 15):

redo LSN 15 (set value to VVV)

set pageLSN(P3)=15

# Example 3: Redo Phase

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:  
last checkpoint at LSN 20



## Active Transaction Table

T1, committing, 35  
T2, aborting, 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

Move to LSN 30:

Read P2

if (pageLSN(P2) < 30):

redo LSN 30 (set value to XXX)

set pageLSN(P2)=30

# Example 3: Redo Phase

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:  
last checkpoint at LSN 20



## Active Transaction Table

T1, committing, 35  
T2, aborting, 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

Move to LSN 35: Prepare End Record

# Example 3: Redo Phase

LSN	LOG
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<i>Commit</i> , T1, prevLSN=30
40	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<i>Abort</i> , T2, prevLSN=40
50	<i>CLR</i> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:  
last checkpoint at LSN 20



## Active Transaction Table

T1, committing, 35  
T2, aborting, 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

Read P1 (if it has been flushed at chkpt, check pageLSN in storage; it should be 10)

Move to LSN 40:

redo LSN 40 (set value to TTT)  
set pageLSN(P1)=40

# Example 3: Redo Phase

Master Record:  
last checkpoint at LSN 20

LSN	LOG
10	<b>Update</b> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<b>Update</b> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<b>Update</b> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<b>Commit</b> , T1, prevLSN=30
40	<b>Update</b> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<b>Abort</b> , T2, prevLSN=40
50	<b>CLR</b> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15



## Active Transaction Table

T1, committing, 35  
T2, aborting, 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

Read P1 (if it has been flushed, check pageLSN in storage; it should be 40)

redo LSN 50 (set value to ZZZ)  
set pageLSN(P1)=50

Move to LSN 50:

# Example 3: Redo Phase

LSN	LOG
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<i>Commit</i> , T1, prevLSN=30
40	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<i>Abort</i> , T2, prevLSN=40
50	<i>CLR</i> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	<i>End</i> , T1, prevLSN=35

Master Record:  
last checkpoint at LSN 20



## Active Transaction Table

~~T1, committing, 35~~  
T2, aborting, 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

Add End to 55:

Remove T1 from ATT

Redo phase done!

→ need to UNDO T2: ToUndo={50}

# Example 3: Undo Phase

LSN	LOG
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<i>Commit</i> , T1, prevLSN=30
40	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<i>Abort</i> , T2, prevLSN=40
50	<i>CLR</i> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	<i>End</i> , T1, prevLSN=35

Master Record:  
last checkpoint at LSN 20



Active Transaction Table

~~T1, committing, 35~~  
T2, aborting, 50

Dirty Page Table

P1, 10  
P3, 15  
P2, 30

ToUndo={50}

# Example 3: Undo Phase

LSN	LOG
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<i>Commit</i> , T1, prevLSN=30
40	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<i>Abort</i> , T2, prevLSN=40
50	<i>CLR</i> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	<i>End</i> , T1, prevLSN=35

Master Record:  
last checkpoint at LSN 20



Active Transaction Table

~~T1, committing, 35~~  
T2, aborting, 50

Dirty Page Table

P1, 10  
P3, 15  
P2, 30

ToUndo={50}

Undo LSN 50:

ToUndo ← 15



# Example 3: Undo Phase

LSN	LOG
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<i>Commit</i> , T1, prevLSN=30
40	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<i>Abort</i> , T2, prevLSN=40
50	<i>CLR</i> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	<i>End</i> , T1, prevLSN=35

Master Record:  
last checkpoint at LSN 20



Active Transaction Table

~~T1, committing, 35~~  
T2, aborting, 50

Dirty Page Table

P1, 10  
P3, 15  
P2, 30

ToUndo={50 15}

Undo LSN 50:

ToUndo ← 15

# Example 3: Undo Phase

LSN	LOG
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<i>Commit</i> , T1, prevLSN=30
40	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<i>Abort</i> , T2, prevLSN=40
50	<i>CLR</i> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	<i>End</i> , T1, prevLSN=35
60	<i>CLR</i> , T2 Undo LSN 15, P3 (UUU), prevLSN=50, undoNext=NULL

Master Record:  
last checkpoint at LSN 20



## Active Transaction Table

~~T1, committing, 35~~  
T2, aborting, 50

## Dirty Page Table

P1, 10  
P3, 15  
P2, 30

ToUndo={15}

Undo LSN 15:

write CLR for P3; set UUU to P3  
ToUndo ← ∅

# Example 3: Undo Phase

LSN	LOG
10	<i>Update</i> , T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	<i>Update</i> , T2, P3, old=UUU, new=VVV, prevLSN=NULL
20	Begin Checkpoint
25	End Checkpoint: ATT=[{T1,r,10}, {T2,r,15}] DPT=[{P1,10}, {P3,15}]
30	<i>Update</i> , T1, P2, old=WWW, new=XXX, prevLSN=10
35	<i>Commit</i> , T1, prevLSN=30
40	<i>Update</i> , T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	<i>Abort</i> , T2, prevLSN=40
50	<i>CLR</i> , T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	<i>End</i> , T1, prevLSN=35
60	<i>CLR</i> , T2 Undo LSN 15, P3 (UUU), prevLSN=50, undoNext=NULL
65	<i>End</i> , T2, prevLSN=60

Master Record:  
last checkpoint at LSN 20



Active Transaction Table

~~T1, committing, 35~~  
~~T2, aborting, 50~~

Dirty Page Table

P1, 10  
P3, 15  
P2, 30

ToUndo={∅}

Write End for T2 & Remove T2 from ATT

Recovery completed!

# Summary

**Checkpointing:** A quick way to limit the amount of log to scan on recovery.

Recovery works in 3 phases:

**Analysis:** Forward from checkpoint.

**Redo:** Forward from oldest recLSN.

**Undo:** Backward from end to first LSN of oldest Xact alive at crash.

Upon Undo, write CLR.

Redo “repeats history”: Simplifies the logic!

# CS660: Intro to Database Systems

## Class 26: NoSQL

Instructor: Manos Athanassoulis

<https://bu-disc.github.io/CS660/>

# What is NoSQL?

from "Geek and Poke"

## *HOW TO WRITE A CV*

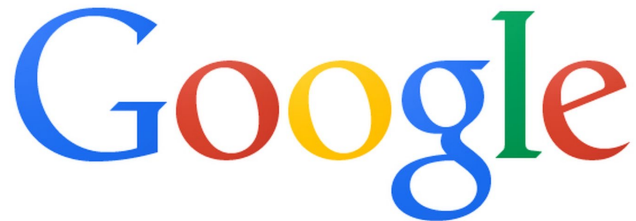


Leverage the NoSQL boom

# What is NoSQL?

An emerging “movement” around non-relational software for Big Data

Roots are in the Google, Amazon, Facebook homegrown software stacks



A NoSQL system provides a **mechanism for storage and retrieval of** data that uses looser consistency models than traditional relational databases in order to achieve horizontal scaling and higher availability.

NoSQL comes from “Not SQL” or “Not only SQL” to emphasize that some NoSQL systems allow SQL-like queries.

# NoSQL Stores

offer an easy to program storage model

**simplification of relational**

two attributes (a key and a value)

value has variable size



# NoSQL features

## Scalability is crucial!

- load increased rapidly for many applications

## Large servers are expensive

## Solution: use clusters of small commodity machines

- need to **shard** the data (maybe use replication)
- cheap (usually open source!)
- cloud-based storage

```
create table people (
  user varchar(24),
  first varchar(24),
  last varchar(24),
  shard key (first)
);
```

people

people_0			people_1			people_2			people_3		
user	first	last	user	first	last	user	first	last	user	first	last
areece	alex	reece	jdoe	john	doe	ndrew	nancy	drew	amonson	amanda	monson
anick	alex	nick	jsmith	john	smith						
amace	alex	mace									
tholmes	tom	holmes									

image from: [singlestore.com](https://singlestore.com)

# NoSQL features

Sometimes not a well-defined schema

Allow for semi-structured data

- still need to provide ways to query efficiently (use of index methods)
- need to express specific types of queries easily

# Scalability

Often cited as the main reason for moving from DB technology to NoSQL

DB Position: there is no reason a parallel DBMS cannot scale to 1000's of nodes

NoSQL Position: a) Prove it; b) it will cost too much anyway

# Flavors of NoSQL

Four main types:

*key-value* stores

*document* databases

*column-family* (aka big-table) stores

*graph* databases

Here we will talk more about “Document” databases (MongoDB)

# Key-Value Stores

There are many systems like that:



redis



Memcached



**amazon**  
DynamoDB

Simple data model: key/value pairs

the DBMS ***does not attempt to interpret*** the value

Queries are limited to ***query by key***

- get/put/update/delete a key/value pair
- iterate over key/value pairs

# Document Databases

Examples include:



Special type of key/value that ***value is a document***.

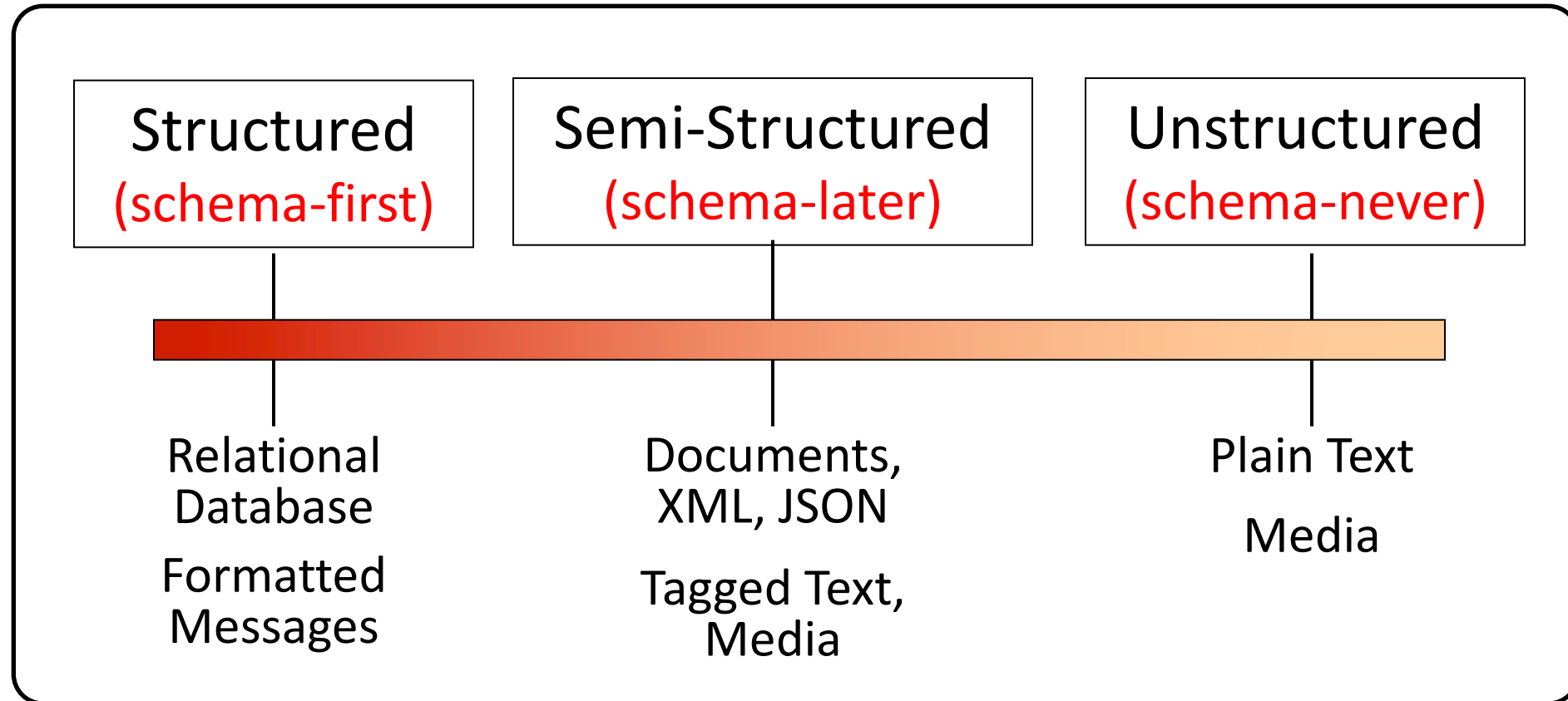
- use some sort of ***semi-structured data model***: XML/JSON
- the ***value can be examined*** and used by the system (unlike in key/data stores)

Queries based on key (as in key/value stores), but ***also on the document*** (value).

Here again, there is support for ***sharding*** and ***replication***.

- the sharding can be based on values within the document

# The Structure Spectrum



# Recap: NoSQL

**simplification of relational: {key, value}**

unique key

variable-size value

Key/Value stores:

- get/put/update/delete a key/value pair
- iterate over key/value pairs

Document stores (value is a ***semi-structured*** document):

- use some sort of semi-structured data model: XML/JSON
- the value can be examined and used by the system (unlike in key/data stores)
- queries based on key (as in key/value stores), but also on the document (value).



# MongoDB (An example of a Document Database)

Data are organized in ***collections***. A collection stores a ***set of documents***.

Collection (*like table*) and document (*like record*)

- **BUT** each document can have *different attributes* even in the same collection
- Semi-structured schema!

Only requirement: every document should have an “\_id” field

- humongous => Mongo

# Example MongoDB

```
{ "_id": ObjectId("4efa8d2b7d284dad101e4bc9"),  
  "Last Name": " Cousteau",  
  "First Name": " Jacques-Yves",  
  "Date of Birth": "06-1-1910" },  
  
{ "_id": ObjectId("4efa8d2b7d284dad101e4bc7"),  
  "Last Name": "PELLERIN",  
  "First Name": "Franck",  
  "Date of Birth": "09-19-1983",  
  "Address": "1 chemin des Loges",  
  "City": "VERSAILLES" }
```

# Example Document Database: MongoDB

Key features include:

JSON-style documents

– actually, uses BSON (JSON's binary format)

***replication*** for high availability

***auto-sharding*** for scalability

***key & document-based*** queries

can create an index on any attribute for faster reads

under the hood, a simple key-value store called WiredTiger!  
design based on LSM-trees/B-Trees

# MongoDB Terminology

relational term  $\Leftrightarrow$  MongoDB equivalent

---

database  $\Leftrightarrow$  database

table  $\Leftrightarrow$  collection

row  $\Leftrightarrow$  document

attributes  $\Leftrightarrow$  fields (field-name:value pairs)

primary key  $\Leftrightarrow$  the `_id` field, which is the key associated with the document

# JSON

JSON is an alternative data model for semi-structured data

- JavaScript Object Notation

Built on two key structures:

- an **object**, which is a sequence of name/value pairs  
`{ "_id": "1000", "name": "Sanders Theatre", "capacity": 1000 }`
- an **array of values** [ "123", "222", "333" ]

A **value** can be:

- an atomic value: string, number, true, false, null
- an object
- an array

# The `_id` Field

Every MongoDB document must have an `_id` field.

its value must be unique within the collection

acts as the primary key of the collection

it is the key in the key/value pair

If you create a document without an `_id` field:

MongoDB adds the field for you

assigns it a unique BSON (binary JSON) ObjectID

example from the MongoDB shell:

```
> db.test.save({ rating: "PG-13" })  
> db.test.find() { "_id" : ObjectId("528bf38ce6d3df97b49a0569"), "rating" : "PG-13" }
```

Note: quoting field names is optional (see `rating` above)

# Capturing Relationships in MongoDB

Two options:

1. store references to other documents using their `_id` values
2. embed documents within other documents

# Example relationships

```
{
  "_id":ObjectId("52ffc33cd85242f436000001"),
  "name": "Tom Benzamin ",
  "contact": "987654321",
  "dob": "01-01-1991"
}
{
  "_id":ObjectId("52ffc4a5d85242602e000000"),
  "building": "22 A, Indiana Apt",
  "pincode": 123456,
  "city": "Los Angeles",
  "state": "California"
}
```

## Here an example of reference-based relationship

```
{
  "_id":ObjectId("52ffc33cd85242f436000001"),
  "contact": "987654321",
  "dob": "01-01-1991",
  "name": "Tom Benzamin",
  "address_ids": [
    ObjectId("52ffc4a5d85242602e000000")
  ]
}
```

## And, here is an example of embedded relationship:

```
{
  "_id":ObjectId("52ffc33cd85242f436000001"),
  "contact": "987654321",
  "dob": "01-01-1991",
  "name": "Tom Benzamin",
  "address": [
    {
      "building": "22 A, Indiana Apt",
      "pincode": 123456,
      "city": "Los Angeles",
      "state": "California"
    },
    {
      "building": "170 A, Acropolis Apt",
      "pincode": 456789,
      "city": "Chicago",
      "state": "Illinois"
    }
  ]
}
```



# CRUD

- **Create**

```
> db.collection.insert( <document> )  
> db.collection.save( <document> )  
> db.collection.update( <query>, <update>, { upsert: true } )
```

- **Read**

```
> db.collection.find( <query>, <projection> )  
> db.collection.findOne( <query>, <projection> )
```

- **Update**

```
> db.collection.update( <query>, <update>, <options> )
```

- **Delete**

```
> db.collection.remove( <query>, <justOne> )
```

# Queries in MongoDB

Each query can only access a single collection of documents.

Use a method called

```
> db.collection.find(<selection>, <projection>)
```

**Example:** find the titles of all R-rated movies:

```
> db.movies.find({ rating: 'R' }, { title: 1 })
```

# Projection

Specify the name of the fields that you want in the output with 1 ( 0 hides the value)

Example:

```
> db.movies.find({}, {"title":1, _id:0})
```

(will report the title but not the id)

# Selection

You can specify the condition on the corresponding attributes using the find:

```
> db.movies.find({ rating: "R", year: 2000 }, { title: 1, runtime: 1 })
```

Operators for other types of comparisons:

MongoDB	SQL equivalent
\$gt, \$gte	>, >=
\$lt, \$lte	<, <=
\$ne	!=

**Example:** find the names of movies with an earnings <= 200000

```
> db.movies.find({ earnings: { $lte: 200000 } })
```

For logical operators \$and, \$or, \$nor

use an array of conditions and apply the logical operator among the array conditions:

```
> db.movies.find({ $or: [ { rating: "R" }, { rating: "PG-13" } ] })
```

# Demo

<https://mongoplayground.net/>

# Other Structure Issues

## NoSQL

- a) Tables are unnatural
- b) “joins” are evil
- c) need to be able to “grep” my data

## DB

- a) Tables are a natural/neutral structure
- b) data independence lets you precompute joins under the covers
- c) this is a price of all the DBMS goodness you get

This is an Old Debate – Object-oriented databases, XML DBs, Hierarchical, ...

# Fault Tolerance

DBs: coarse-grained FT – if trouble, restart transaction

- Fewer, Better nodes, so failures are rare
- Transactions allow you to kill a job and easily restart it

NoSQL: Massive amounts of cheap HW, **failures are the norm** and massive data means **long running jobs**

- So must be able to do mini-recoveries
- This causes some overhead (file writes)

CS660: Intro to Database Systems

# Database Systems and Beyond

Instructor: Manos Athanassoulis

<https://bu-disc.github.io/CS660/>



# Database Systems

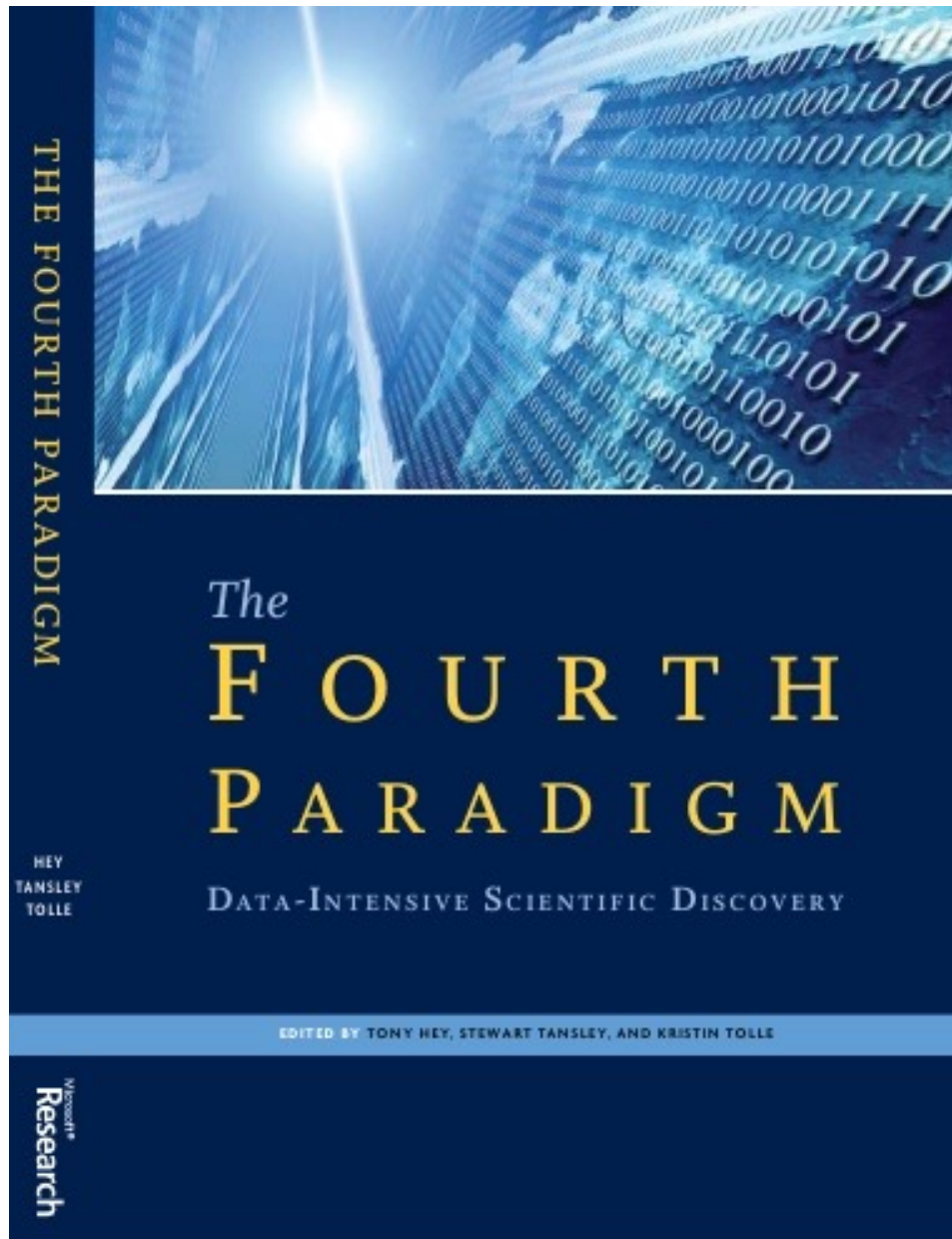
we spent a whole semester on Database Systems  
what is next?

*what can we do with data?*

data-driven science

data-driven discovery

data-driven governance



*“Experimental, theoretical, and computational science are all being affected by the data deluge, and a fourth, ‘**data-intensive**’ science paradigm is emerging.*”

*The goal is to have a world in which all of the science literature is online, all of the science data is online, and they interoperate with each other. **Lots of new tools are needed to make this happen.**”*

Faster Innovation through Data-Intensive Approaches

Need for Innovation in Data Management!

# DATA & AI LANDSCAPE 2019

## INFRASTRUCTURE

**HADOOP ON-PREMISE**  
 cloudera Hortonworks  
 MAPR Pivotal  
 IBM InfoSphere  
 jethro

**HADOOP IN THE CLOUD**  
 aws Microsoft Azure  
 Google Cloud  
 SAP Cloud Platform  
 IBM InfoSphere BigInsights  
 arm  
 Dubble CAZENA

**STREAMING / IN-MEMORY**  
 Amazon Kinesis  
 databricks  
 SAP Cloud Platform  
 ORACLE  
 confluent  
 strimzi  
 hazelcast  
 GridGain  
 GIGASPACEs  
 Wallaroo AI  
 FASTDATA  
 KX

**NoSQL DATABASES**  
 Google Cloud AWS  
 ORACLE  
 mongoDB MarkLogic  
 Couchbase DISTRAX  
 redislabs  
 KEROPIKE  
 ArangoDB SCYLLA

**NewSQL DATABASES**  
 SAP Clustrix Pivotal  
 MEMSQL infuxdata  
 Cockroach LABS  
 VOLTDB splice  
 paradigms  
 IBM Amazon Neptune  
 ORACLE  
 OrientDB  
 InfiniteGraph  
 Objectivity

**GRAPH DBs**  
 Neo4j  
 Amazon Neptune  
 IBM  
 ORACLE  
 OrientDB  
 InfiniteGraph  
 Objectivity

**MPP DBs**  
 TERADATA  
 IBM Data Warehouse Systems  
 Kognitio  
 Exasol  
 dremio  
 Yellowbrick

**CLOUD EDW**  
 aws  
 Google Cloud  
 Microsoft Azure  
 Pivotal  
 snowflake  
 Infoworks

**SERVERLESS**  
 AWS Lambda  
 Azure Functions  
 Google Cloud Functions  
 PULSAR  
 nuclio  
 PULSAR  
 nuclio  
 PULSAR  
 nuclio

**DATA TRANSFORMATION**  
 talend pentaho  
 alteryx TRIFACTA  
 tamr Paxata  
 StreamSets UNIFI

**DATA INTEGRATION**  
 SAP Data Services Informatica  
 MuleSoft TEALUM  
 snaplogic enigma  
 Segment ATTUNITY  
 xplenty ZALONI import.io  
 Infosys Fivetran  
 SNOWFLOW MATILLION

**DATA GOVERNANCE**  
 Informatica  
 SailPoint  
 IBM  
 Alation  
 HMDTA  
 OKERA  
 MANTA  
 dataworld

**MGMT / MONITORING**  
 aws New Relic octio  
 rubrik  
 dynamtrace  
 SignalFX  
 splunk  
 Moogsoft pagerduty  
 Unavai Numerify  
 zentao OpsRamp  
 APPDYNAMICS  
 WAVEFRONT  
 druid  
 pageduty  
 VEBAM  
 MAGNITUDE

**STORAGE**  
 aws  
 Microsoft Azure  
 PURE STORAGE  
 ALLUXIO wasabi  
 nimbustorage  
 Qumulo panache  
 COHERITY

**CLUSTER SVCS**  
 IBM  
 MESOS  
 docker  
 CDAP  
 HELIX

**DATA GENERATION & LABELLING**  
 amazon mechanicalturk  
 upwork  
 open scale  
 HIPVEN  
 Labelbox  
 Mighty AI  
 ALBERTE  
 LIONBRIDGE

**AI OPS**  
 ALGORITHMIA  
 SPILL comet  
 Verta.ai datmo  
 distatton  
 Whiggle AI  
 Determined AI  
 fiddler

**GPU DBs & CLOUD**  
 kinetica  
 SQUEAM  
 bryllyt  
 BLAZINGDB  
 PG-Stream  
 LOYDHUB

**HARDWARE**  
 Google TPU arm  
 intel AI NVIDIA  
 GRAPHCORE  
 MYTHIC  
 Movidius habana  
 WAVE  
 CERNAMI  
 FALLO  
 DEFNIX

## ANALYTICS & MACHINE INTELLIGENCE

**DATA ANALYST PLATFORMS**  
 Microsoft pentaho alteryx  
 Digital Reasoning  
 GUAVUS AYASDI  
 ATTIVO Datameer incorta  
 interana MODE ENDOR  
 sisu switchboard Starburst

**DATA SCIENCE PLATFORMS**  
 IBM databricks dataiku  
 DOMINO rapidminer TIBCO  
 ANACONDA SAS  
 KNIME MathWorks

**BI PLATFORMS**  
 looker  
 DOME  
 ATSCALE  
 Qlik  
 MicroStrategy Keen IO

**VISUALIZATION**  
 tableau Power BI  
 SAP  
 celonis  
 zepl  
 CHARTIO  
 TUCAN TOGO

**MACHINE LEARNING**  
 Azure Machine Learning  
 Google Cloud AI Platform Vision  
 H2O  
 DataRobot gamalon  
 VISENZE ELEMENT  
 deepsense.ai

**COMPUTER VISION**  
 Microsoft Azure  
 Amazon Rekognition  
 clarifai  
 EVERAI  
 neuro  
 UBUIQUITY  
 ABBE  
 trax  
 synthesisia  
 IBM Watson Cortana Face++  
 sentient Voyager  
 Affective  
 Numenta  
 narologics  
 BLUE VISION  
 II  
 Fortress

**HORIZONTAL AI**  
 IBM Watson Cortana Face++  
 sentient Voyager  
 Affective  
 Numenta  
 narologics  
 BLUE VISION  
 II  
 Fortress

**SPEECH & NLP**  
 Google Cloud  
 Amazon Transcribe  
 narrative science  
 Movable  
 SoundHound Inc  
 Mindfield  
 cogito snips  
 SMARTLING Unbabel PolyAI

**SEARCH**  
 elasticsearch ORACLE ENIGA  
 algolia covéo  
 Lucidworks ATTIVO  
 swifttype EXALEAD  
 alphasense MAANA  
 omnius SINEQUA

**LOG ANALYTICS**  
 splunk  
 sumologic  
 solarwinds  
 TIMBER  
 kibana  
 logz.io

**SOCIAL ANALYTICS**  
 Hootsuite sprinklr  
 NETBASE  
 synthosio trackx  
 simple reach  
 bitly SimilarWeb

**WEB / MOBILE / COMMERCE ANALYTICS**  
 Google Analytics  
 mixpanel AMPITUDE  
 Airtable RESCI  
 SIGOPT granify  
 custora

## CROSS-INFRASTRUCTURE/ANALYTICS

aws Google Cloud Microsoft IBM SAP Hewlett Packard Enterprise SAS IOIOWATA vmware TIBCO TERADATA ORACLE NetApp syncsort MAPR cloudera

## APPLICATIONS - ENTERPRISE

**SALES**  
 CHORUS  
 INSIDESALES.COM peopleai  
 conversica  
 clari aviso tact.ai  
 fuse machines Clearbit

**MARKETING - B2B**  
 RADIUS App Annie  
 EVERSTRING  
 MINTIGO  
 sense  
 tubular  
 NGAGIO  
 KNOTCH mrpe

**MARKETING - B2C**  
 Zeta bloomreach SendGrid  
 braze ACTIONIQ BLUECORE  
 CONTENT SQUARE TEALUM  
 Ampero amperity QUANTIFIND  
 Simon Lytica PERSADO  
 remesh

**CUSTOMER EXPERIENCE / SERVICE**  
 qualtrics MEDALLIA SurveyMonkey UserTesting  
 CLARABRIDGE zendesk Customer Freshdesk  
 INTERCOM Drift LIVEPERSON Gainsight pendo  
 HEAP Amplitude Watson Assistant  
 DigitalGenius A.S.A.P.P ada NUTOMAT ahniti  
 CaSDesk NFORMI TestSet frame ai

**ENTERPRISE PRODUCTIVITY**  
 slack  
 ORACLE  
 GURU lumia  
 DIFFBOT clara  
 talla Kasisto

**HUMAN CAPITAL**  
 HireVue pyometrics  
 hiQ GOSTER mya  
 Allyo textio  
 Wade & Wendy Stella  
 entelo  
 uncommon  
 eat beemey

**LEGAL**  
 RAVEL  
 Everlaw  
 DISCO kiro  
 JUDICATA  
 BREVIATA  
 IRONCLAD  
 PREPOSITION  
 ROSS  
 Casetext

**REGTECH & COMPLIANCE**  
 BigID TESSIAN  
 text IQ Comply Advantage  
 PARTNERSHIPS  
 CROSSCHECK  
 DATA REPUBLIC

**FINANCE**  
 Anaplan  
 ZUORA  
 SAP S/4HANA  
 TRADESHIFT  
 SCALE FACTOR  
 baakeeper  
 pilot

**BACK OFFICE AUTOMATION & RPA**  
 UiPath  
 blueprism  
 VIDADO AppZen  
 WorkFusion workato  
 REXCESS Catalytic  
 ANTWORKS  
 ALKYRI

**SECURITY**  
 TANIUM CYLANCE zscaler StackPath illumio  
 CODE42 CyberCloud  
 VECTRA  
 pindrop exabeam  
 Vade Secure bitglass  
 sparkcognition  
 ARCA SECURITY BLUEHEXAGON Semble

## APPLICATIONS - INDUSTRY

**ADVERTISING**  
 AppNexus  
 critico  
 ORACLE MOAT  
 theTradeDesk  
 dstillery  
 LiveIntent  
 TAPAD dataxum gumgum  
 Cppier

**EDUCATION**  
 Lishuluo  
 猿辅导  
 KNEWTON  
 Clever  
 Cleara  
 kidaptive  
 PANORAMA  
 knowre  
 gradescope

**REAL ESTATE**  
 REDFIN  
 Opendoor  
 VTS  
 CREDIFI  
 GEOPHY  
 reonomy  
 COMPSTAK  
 SKYLINE

**GOV'T**  
 OPENGOV  
 mark43  
 LiveStories  
 Passport  
 SmartProcure  
 STREETLIGHTDATA  
 OpenDataSoft

**INTELLIGENCE**  
 Palantir  
 Dataminr  
 Quid  
 PRIMER  
 FORGE

**FINANCE - INVESTING**  
 KENSHC  
 Quantopian  
 ADAPPER  
 NUMERA  
 SENTIUM  
 ALGORIZ  
 FlavourPack  
 PAGAYA

**FINANCE - LENDING**  
 ondeck affirm 拍拍贷  
 JIANPU.AI Kreditech AVANT  
 TALA  
 CLEARBANC  
 100Credit WeLab Wecash  
 TrueAccord MoneyLion  
 aire cignifi

**INSURANCE**  
 Metromile  
 Lemonade  
 CYENCE  
 Hippo  
 Shift Technology  
 ROOT  
 zesty.ai  
 CAPE

**HEALTHCARE**  
 flatiron Clover KYRUS  
 METABIOTA Gingerio Glow  
 3DMed zebra PathAI Ovia  
 TEMPUS patientslikemr AiCure insitro  
 LIAODOO  
 notable  
 Qventus ARTERYS IMA GEN  
 innovacore

**LIFE SCIENCES**  
 color  
 BenevolentAI verily  
 WuXiNextCODE  
 Clear Labs  
 Phosphorix  
 twoAR  
 QWIKIN

**TRANSPORTATION**  
 UBER TESLA  
 CLEARPATH CRUISE  
 nuro  
 nauto AMOTIVE G7 PILOT.AI  
 NIO OPTIMUS moovit  
 nexar Kodiak comma.ai  
 netradynae  
 thinc INRIX

**AGRICULTURE**  
 FARMERS  
 Granular  
 JOHN DEERE  
 BLUE RIVER  
 FarmersEdge  
 AgroStar  
 FARMLOGS  
 TARANIS  
 GAMAYA  
 Terraviva  
 prospera

**COMMERCE**  
 instacart  
 STITCH FIX  
 Stitch Fix  
 lowGood  
 heurixact

**INDUSTRIAL**  
 AVEVA SIEMENS  
 PREDIX UPTAKE  
 SCORTEX  
 TACHYUS  
 OTHER  
 eHarmony stem Amper ByteDance  
 huggies elect SOJERN BBOXEVER  
 VERDIGRIS duetto  
 Electric ZINER  
 Moveworks

## OPEN SOURCE

**FRAMEWORKS**  
 Spark  
 Flink YARN TEZ  
 MESOS  
 docker CDAP  
 Red Hat HELIX

**QUERY / DATA FLOW**  
 Spark SQL  
 presto  
 SLAMDATA  
 APACHE DRILL  
 GraphQL  
 Flink

**DATA ACCESS & DATABASES**  
 cassandra mongoDB redis  
 CouchDB  
 Cockroach LABS  
 druid  
 riak  
 ORACLE  
 SciDB  
 ACCUTILLO

**ORCHESTRATION & MGMT**  
 talend  
 Apache Zookeeper  
 Apache Ambari  
 Apache Airflow  
 MESOS  
 etcd Kong

**STREAMING & MESSAGING**  
 Spark  
 nifi  
 Flink  
 beam  
 kafka STORM  
 Apache RocketMQ

**STAT TOOLS & LANGUAGES**  
 python  
 Scala  
 Studio  
 SciPy julia

**AI OPS & INFRA**  
 miflow  
 Kubeflow  
 mlops  
 SELDON  
 PyTorch

**AI / MACHINE LEARNING / DEEP LEARNING**  
 TensorFlow Keras  
 OpenAI DM TK  
 theano  
 mxnet VELES  
 Chainer  
 PyTorch  
 Microsoft Cognitive Toolkit  
 Caffe  
 DIMSUM FeatureFu  
 ONNX  
 PyTorch

**SEARCH**  
 elasticsearch  
 Solr

**LOGGING & MONITORING**  
 elasticsearch kibana SENTRY  
 logstash Prometheus  
 fluentbit fluentd Grafana  
 VECTOR

**VISUALIZATION**  
 matplotlib  
 TensorBoard  
 seaborn  
 Bokeh

**COLLABORATION**  
 BeakerX  
 jupyter  
 ANACONDA

**SECURITY**  
 Apache Ranger  
 KNOX  
 Sentry  
 ACCUTILLO

## DATA SOURCES & APIs

HEALTH IOT FINANCIAL & ECONOMIC DATA AIR / SPACE / SEA PEOPLE / ENTITIES LOCATION INTELLIGENCE OTHER DATA SERVICES INCUBATORS & SCHOOLS RESEARCH

see full version at: [http://mattturck.com/wp-content/uploads/2019/07/2019\\_Matt\\_Turck\\_Big\\_Data\\_Landscape\\_Final\\_Fullsize.png](http://mattturck.com/wp-content/uploads/2019/07/2019_Matt_Turck_Big_Data_Landscape_Final_Fullsize.png)





increase throughput by parallelization

“scale-up”

use more powerful machines (>#CPUs, >RAM)

“scale-out”

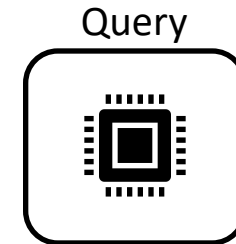
use more machines

# Scale Up Execution

how to use more cores (threads)?

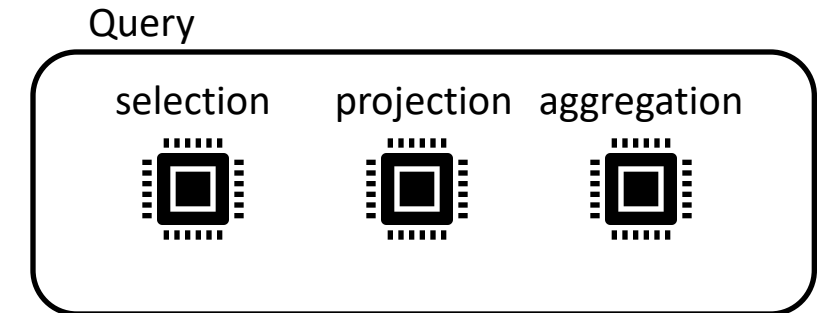
## **inter-query parallelism**

each query runs on one processor



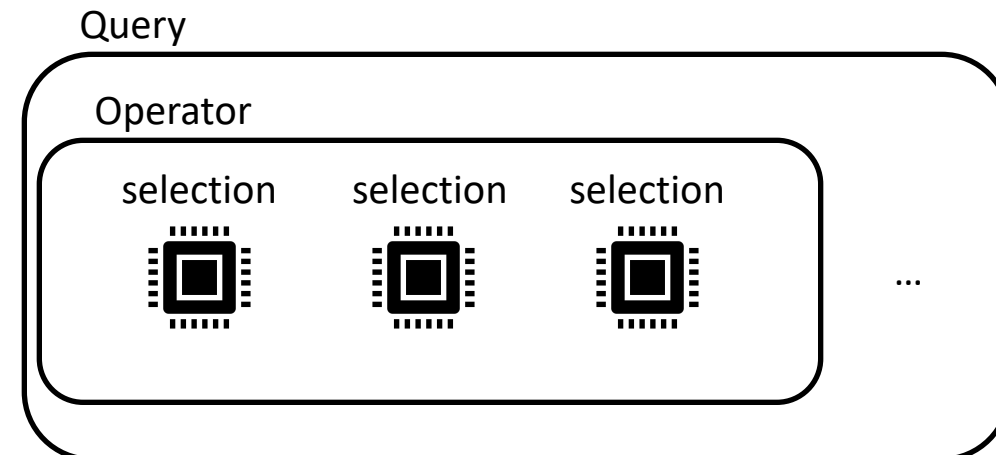
## **inter-operator parallelism**

each query runs on multiple processors  
an operator runs on one processor



## **intra-operator parallelism**

an operator runs on multiple processors



# Scale Up Storage

needs more disks!

how to distribute data?

**block partition**

**hash partition**

**range partition**

how to distribute data accesses?

# Scale Out

similar questions across machines

new bottlenecks?

move data across machines: network!



# DATA & AI LANDSCAPE 2019

## INFRASTRUCTURE

**HADOOP ON-PREMISE**  
cloudera Hortonworks  
MAPR Pivotal  
IBM InfoSphere  
jethro

**HADOOP IN THE CLOUD**  
aws Microsoft Azure  
Google Cloud  
SAP Cloud Platform  
IBM InfoSphere BigInsights  
arm  
Dubble CAZENA

**STREAMING / IN-MEMORY**  
Amazon Kinesis  
databricks  
SAP Cloud Platform  
ORACLE  
confluent  
strimio hazelcast  
GridGain  
GIGASPACEs Wallaroo  
FASTDATA kx

**SQL DATABASES**  
Google Cloud  
ORACLE  
mongoDB MarkLogic  
Couchbase  
redislabs  
ArangoDB SCYLLA

**NewSQL DATABASES**  
SAP Clustrix  
Pivotal  
MongoDB  
MEMSQL  
Cockroach LABS  
VOLTO DB splice  
paradigm

**GRAPH DBs**  
Amazon Neptune  
IBM  
ORACLE  
InfinitiGoth

**MPP DBs**  
TERADATA  
IBM Data Warehouse Systems  
Cognitio  
Kognitio  
Exasol  
dremio  
Yellowbrick

**CLOUD EDW**  
aws  
Microsoft Azure  
Pivotal  
snowflake  
Infoworks

**SERVERLESS**  
Amazon SageMaker  
Google Cloud  
Microsoft Azure  
Pulsar  
nuclio  
Pulumi  
Phase Function Service

**DATA TRANSFORMATION**  
talend pentaho  
alteryx TRIFACTA  
tamr Paxata  
StreamSets UNIFI

**DATA INTEGRATION**  
SAP Data Services  
MuleSoft TEALUM  
snaplogic enigma  
Segment ATTUNITY  
ZALONI import.io  
InfoWorks Fivetran  
SNOWFLOW MATILLION

**DATA GOVERNANCE**  
Informatica  
SailPoint  
Maize Skyhigh Security Cloud  
colibra  
Alation  
HMDTA  
OKERA  
MANTA dataworld

**MGMT / MONITORING**  
aws New Relic  
AppDynamics  
rubrik  
dynatrace  
SignalFX  
splunk  
Moogsoft pagerduty  
unavai  
Numerify  
zenoss OpsRamp

**STORAGE**  
aws  
Microsoft Azure  
PURE STORAGE  
ALLUXIO wasabi  
Qumulo Panache  
COHERITY

**CLUSTER SVCS**  
Amazon EKS  
IBM Storage  
Google Cloud  
Microsoft Azure  
MESOS  
docker CDAP  
HELDX

**DATA GENERATION & LABELLING**  
amazon mechanicalturk  
Upwork  
unity

**AI OPS**  
ALGORITHMIA  
SPILL comet  
Verta.ai datmo

**GPU DBs & CLOUD**  
kinetica  
SQUEAM  
IBM Power  
GRAPHCORE  
MYTHIC

**HARDWARE**  
Google TPU arm  
intel AI NVIDIA  
GRAPHCORE  
MYTHIC

**CROSS-INFRASTRUCTURE/ANALYTICS**  
aws Google Cloud Microsoft IBM SAP Hewlett Packard Enterprise SAS IOI DATA vmware TIBCO TERADATA ORACLE NetApp syncsort MAPR cloudera

## ANALYTICS & MACHINE INTELLIGENCE

**DATA ANALYST PLATFORMS**  
Microsoft pentaho alteryx  
Digital Reasoning  
GUAVUS AYASDI  
ATTIVO Datameer incorta  
interana MODE ENDOR  
sisu switchboard Starburst

**DATA SCIENCE PLATFORMS**  
IBM databricks dataiku  
DOMINO rapidminer TIBCO  
ANACONDA SAS  
KNIME MathWorks

**BI PLATFORMS**  
looker  
amazon analytics  
DOME  
ARCADIA DATA ThoughtSpot  
ATSCALE Qlik  
GoodData Information Builders birst  
MicroStrategy Keon IO

**VISUALIZATION**  
tableau  
SAP  
celonis  
zepl  
CHARTIO

**MACHINE LEARNING**  
Amazon SageMaker  
Google Cloud  
DataRobot gamalon  
VISENZE ELEMENT  
deepense.ai

**COMPUTER VISION**  
Microsoft Azure  
Amazon Rekognition  
clarifai

**HORIZONTAL AI**  
IBM Watson Cortana  
sentient  
Affectiva  
Numenta  
narologics  
synthesia

**SPEECH & NLP**  
Google Cloud  
amazon alexa  
narrative science  
Mobval  
SoundHound Inc  
PRIMER  
cogito snips  
SMARTLING Unbabel PolyAI

**SEARCH**  
elasticsearch  
algolia covéo  
Lucidworks ATTIVO

**LOG ANALYTICS**  
splunk  
sumologic  
solarwinds

**SOCIAL ANALYTICS**  
Hootsuite  
NETBASE

**WEB / MOBILE / COMMERCE ANALYTICS**  
Google Analytics  
mixpanel AMPITUDE

## APPLICATIONS - ENTERPRISE

**SALES**  
CHORUS  
INSIDESALES.COM people.ai  
conversica  
clari aviso tact.ai  
fuse machines Clearbit

**MARKETING - B2B**  
RADIUS App Annie  
EVERSTRING  
MINTIGO  
sense  
tubular  
KNOTCH mrpe

**MARKETING - B2C**  
Zeta  
bloomreach SendGrid  
braze ACTIONIQ BLUECORE  
CONTENT SQUARE TEALIUM  
Amplero amperity QUANTIFIND  
Simon Lytice PERSADO  
remesh

**CUSTOMER EXPERIENCE / SERVICE**  
qualtrics MEDALLIA SurveyMonkey UserTesting  
CLARABRIDGE zendesk Customer freshdesk  
INTERCOM Drift LIVEPERSON Gainsight pendo  
HEAP Amplitude Watson Assistant  
DigitalGenius A.S.A.P.P ada NUTOMAT ahniti  
CaDesk

**ENTERPRISE PRODUCTIVITY**  
slack  
ORACLE  
GURU lumina  
DIFFBOT clara  
talla Kasisto

**HUMAN CAPITAL**  
Hiive  
hiIQ  
Allyo textio  
Wade & Wendy Stella  
entelo  
uncommon  
eat beemey

**LEGAL**  
RAVEL  
Everlaw  
DISCO kiro  
JUDICATA  
BREVIA  
IRONCLAD  
PREMPTION  
ROSS  
Casetext

**REGTECH & COMPLIANCE**  
BigID  
text IQ  
PARTNERSHIPS  
CROSSSTREAM  
DATA REPUBLIC

**FINANCE**  
fnaplan  
ZUORA  
SAP S/4HANA  
TRADESHIFT  
SCALE FACTOR  
pilot

**BACK OFFICE AUTOMATION & RPA**  
UiPath  
blueprints  
VIDADO  
WorkFusion workato  
SCALE FACTOR  
ANTWORKS  
ALUKYI

**SECURITY**  
TANIUM  
CYLANCE  
zscaler  
StackPath  
illumio  
CODE42  
CIPHERCLoud  
DARKTRACE ANOMALI  
Vectra  
VECTRA  
Guardian  
DATAVISOR  
sift science  
pindrop  
exabeam  
SICNIFYD  
SentinelOne  
SecurityScorecard  
SECURE  
CodeSecure  
bitglass  
BlueTalon  
Recorded Future  
ANTWORKS  
KRYON  
ARKAT SECURITY  
BLUEHEXAGON  
Semble  
OSISSIM  
XANONIS  
SHIELD51  
ArmitBlax

## APPLICATIONS - INDUSTRY

**ADVERTISING**  
AppNexus  
critico  
ORACLE  
MOAT  
theTradeDesk  
dstillery  
LiveIntent  
Cappor

**EDUCATION**  
Lullishoo  
Knewton  
Clever  
Clarora  
kidaptive  
PANORAMA  
knowre  
gradescope

**REAL ESTATE**  
REDFIN  
Opendoor  
VTS  
CREDIFI  
GEOPHY  
geonomy  
COMPSTAK  
SPACEMAKER  
SKYLINE

**GOV'T**  
OPENGOV  
mark43  
LiveStories  
Passport  
SmartProcure  
STREETLIGHTDATA  
OpenDataSoft

**INTELLIGENCE**  
Palantir  
Dataminr  
Quid  
PRIMER  
FORGE

**FINANCE - INVESTING**  
KENSHC  
Quantopian  
ADAPTEP  
NUMERA  
SENTIUM  
ALGORIZ  
FlavourPack  
PAGAYA

**FINANCE - LENDING**  
ondeck  
affirm  
拍拍贷  
JIANPU.AI  
Kreditech  
AVANT  
TALA  
aurea  
100Credit  
WeLab  
Weecash  
TrueAccord  
MoneyLion  
Active AI  
aire  
cignif

**INSURANCE**  
Metromile  
Lemonade  
CYENCE  
Hippo  
Shift Technology  
ROOT  
zesty.ai  
TRAVELAI  
CAPE

**HEALTHCARE**  
flatiron Clover  
METABIOTA  
Gingerio  
Glow  
babylon  
3D Med  
zebra  
PathAI  
ovia

**LIFE SCIENCES**  
ZelusMe  
color  
Benevolenti AI  
verily  
WuXiNextCODE

**TRANSPORTATION**  
UBER TESLA  
CLEARPATH  
CRUISE  
NUFO  
drive ai  
CAMBRIDGE  
Aurora  
nauto  
AMOTIVE  
G7  
PILOT.AI  
NIO  
OPTIMUS  
moovit  
nexar  
Kodiak  
comma.ai  
netradynce  
Civil Maps  
cogitate  
thinc INRIX

**AGRICULTURE**  
FARMERS  
Granular  
JOHN DEERE  
BLUE RIVER  
FarmersEdge  
AgroStar  
FarmLogs  
TARANIS  
GAMAYA  
TerraVision  
prospera

**COMMERCE**  
instacart  
FAIRFIRE  
STITCH FIX  
Doo & Co  
loweGood  
heuristic  
eHarmony  
stem  
Amper  
ByteDance  
happyn  
elect  
SOJERN  
BBOXEVER  
VERDIGIS  
duetto  
Electric  
ZINER  
Mooovers

**INDUSTRIAL**  
AVEVA SIEMENS  
PREDIX UPTAKE  
SCORTEX  
KONICA  
TACHYUS  
Apac

**FRAMEWORKS**  
Spark  
Flink  
YARN  
TEZ  
MESOS  
docker  
CDAP  
HELDX

**QUERY / DATA FLOW**  
Spark SQL  
presto  
SLAM DATA  
APACHE DRILL  
GraphQL  
Flink

**DATA ACCESS & DATABASES**  
cassandra  
mongoDB  
redis  
Cockroach LABS  
druidd  
SLAM DATA  
APACHE DRILL  
GraphQL  
Flink

**ORCHESTRATION & MGMT**  
talend  
Apache Ambari  
Apache Airflow  
MESOS  
etcd  
Kong

**STREAMING & MESSAGING**  
Spark  
nifi  
Flink  
beam  
kafka  
STORM  
Apache RocketMQ

**STAT TOOLS & LANGUAGES**  
python  
Scala  
Studio  
SciPy julia

**AI OPS & INFRA**  
miflow  
Kubeflow  
misp  
DVC  
SELECON  
PyTorch

**AI / MACHINE LEARNING / DEEP LEARNING**  
TensorFlow  
Keras  
theano  
DM TK  
mxnet  
VELES  
Chainer  
Microsoft Cognitive Toolkit  
OpenAI  
Caffe  
Apache SINGA  
DIMSUM  
FeatureFu  
neon DSSTNE ml4j DL4J MAHOUT Aerosolve FastAI mir

**SEARCH**  
elasticsearch  
Solr

**LOGGING & MONITORING**  
elasticsearch  
kibana  
sentry  
logstash  
Prometheus  
fluentbit  
fluentd  
Grafana  
VECTOR

**VISUALIZATION**  
matplotlib  
TensorBoard  
seaborn  
Bakari

**COLLABORATION**  
BeakerX  
jupyter  
ANACONDA

**SECURITY**  
Apache Ranger  
KNOX  
Sentry  
accruML

**DATA SOURCES & APIs**  
HEALTH: Apple, VALIDIC, practicefusion  
IOT: GE Digital, UPTAKE  
FINANCIAL & ECONOMIC DATA: GALVANIZE  
AIR / SPACE / SEA: GALVANIZE  
PEOPLE / ENTITIES: facebook research, MIRI  
LOCATION INTELLIGENCE: GALVANIZE  
OTHER: GALVANIZE

**DATA RESOURCES**  
DATA SERVICES: GALVANIZE  
INCUBATORS & SCHOOLS: GALVANIZE  
RESEARCH: facebook research, MIRI

Versatile and popular infrastructure: NoSQL stores

diving into the internals of modern data systems

*cutting-edge* designs / *research* projects / *engineering* projects

# CS 561: Data Systems Architectures

Spring 2024

# A path in data science & data engineering

(1) strong data systems skills

(i) coding skills

(ii) system architecture insights

performance tradeoffs

(2) application domain knowledge

(3) statistics, machine learning, math tools

# Open Discussion

Questions?

for NoSQL

for DBMS

for next semester (CS561!)

for life after college (Academia vs. Industry vs. ?)

Next: Review and questions for final