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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, 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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, 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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, 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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, 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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, T1, prevLSN=15

Active Transaction Table

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

Dirty Page Table

P1, 10

P2, 15

Master Record:

last checkpoint at LSN 00

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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, 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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, 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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, 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	***Update***, T1, P1, prevLSN=NULL
15	***Update***, T1, P2, prevLSN=10
20	***Commit***, T1, prevLSN=15
25	***End***, T1, prevLSN=20

Active Transaction Table

~~T1, committing, 20~~

Dirty Page Table

P1, 10

P2, 15

Redo phase done!

→ nothing to UNDO

→ Recovery is completed!

Add End to LSN 25: Remove T1 from ATT

Recovery

EXAMPLE 2

Example 2

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

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}

Master Record:

last checkpoint at LSN 00

Example 2: Redo Phase

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

Example 2: Undo Phase

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

Active Transaction Table

~~T1, committing, 25~~

~~T2, running, 30~~

Dirty Page Table

P1, 10

P2, 15

P3, 20

ToUndo={Ø}

Write End for T2 & Remove T2 from ATT

Recovery completed!

Recovery

EXAMPLE 3

Example 3

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
	CRASH

Master Record:

last checkpoint at LSN 20

Active Transaction Table

[xact_id, status, lastLSN]

Dirty Page Table

[page_id, recLSN]

Example 3: Analysis Phase

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:

last checkpoint at LSN 20

Active Transaction Table

[xact_id, status, lastLSN]

Dirty Page Table

[page_id, recLSN]

Example 3: Analysis Phase

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:

last checkpoint at LSN 20

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

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:

last checkpoint at LSN 20

Active Transaction Table

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

Dirty Page Table

P1, 10
P3, 15
P2, 30

Example 3: Analysis Phase

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:

last checkpoint at LSN 20

Active Transaction Table

T1, ~~running, 30~~ committing, 35

T2, running, 15

Dirty Page Table

P1, 10

P3, 15

P2, 30

Example 3: Analysis Phase

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:

last checkpoint at LSN 20

Active Transaction Table

T1, committing, 35

T2, running, ~~15~~ 40

Dirty Page Table

P1, 10

P3, 15

P2, 30

Example 3: Analysis Phase

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15

Master Record:

last checkpoint at LSN 20

Active Transaction Table

T1, committing, 35

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

Dirty Page Table

P1, 10

P3, 15

P2, 30

Example 3: Analysis Phase

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, 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, ~~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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, 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 40:

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

redo LSN 40 (set value to TTT)

set pageLSN(P1)=40

Example 3: Redo Phase

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, 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 50:

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

Example 3: Redo Phase

LSN	LOG
10	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	***End***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	***End***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	***End***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	***End***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	***End***, T1, prevLSN=35
60	***CLR***, 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	***Update***, T1, P1, old=YYY, new=ZZZ, prevLSN=NULL
15	***Update***, 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	***Update***, T1, P2, old=WWW, new=XXX, prevLSN=10
35	***Commit***, T1, prevLSN=30
40	***Update***, T2, P1, old=ZZZ, new=TTT, prevLSN=15
45	***Abort***, T2, prevLSN=40
50	***CLR***, T2 Undo LSN 40, P1 (ZZZ), prevLSN=45, undoNextLSN=15
55	***End***, T1, prevLSN=35
60	***CLR***, T2 Undo LSN 15, P3 (UUU), prevLSN=50, undoNext=NULL
65	***End***, 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={Ø}

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 CLRs.

Redo “repeats history”: Simplifies the logic!

CS660: Intro to Database Systems

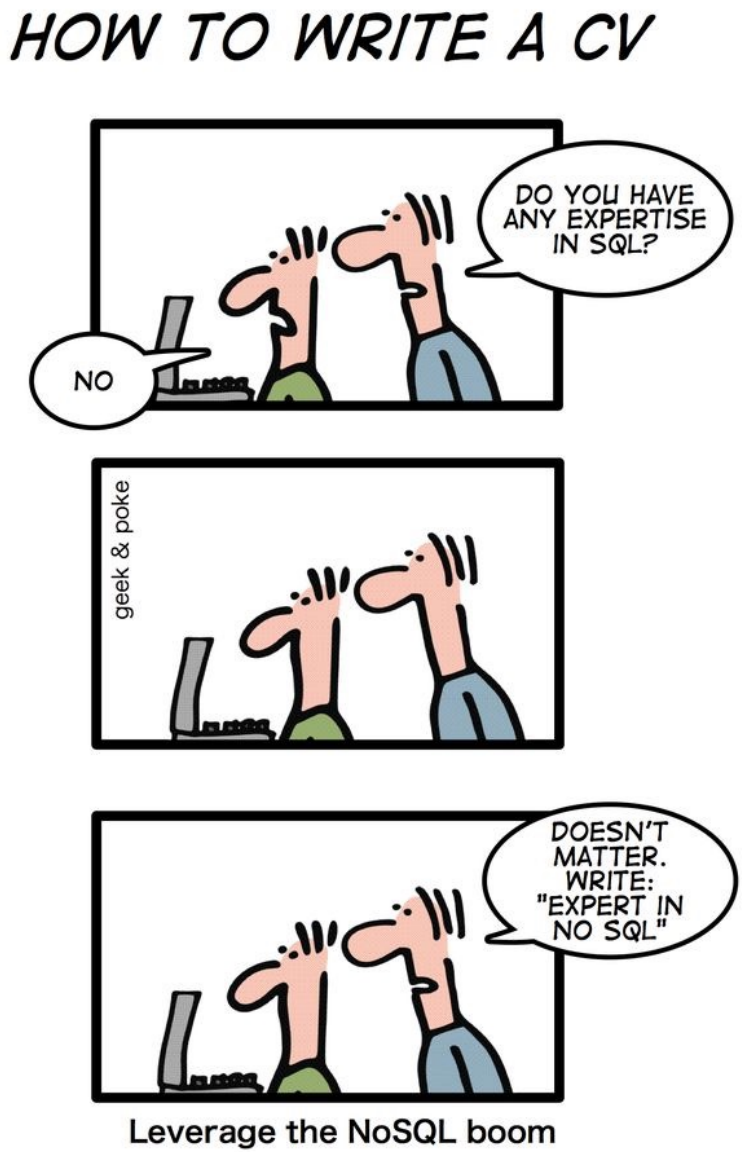
Class 26: NoSQL

Instructor: Manos Athanassoulis

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

What is NoSQL?

from “Geek and Poke”



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)
);
```

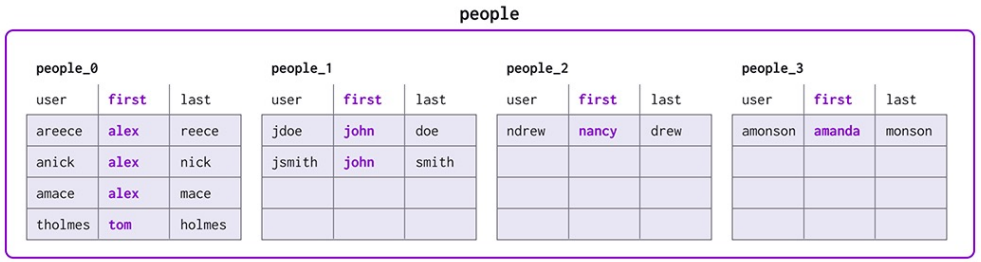


image from: singlestore.com

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:

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***.

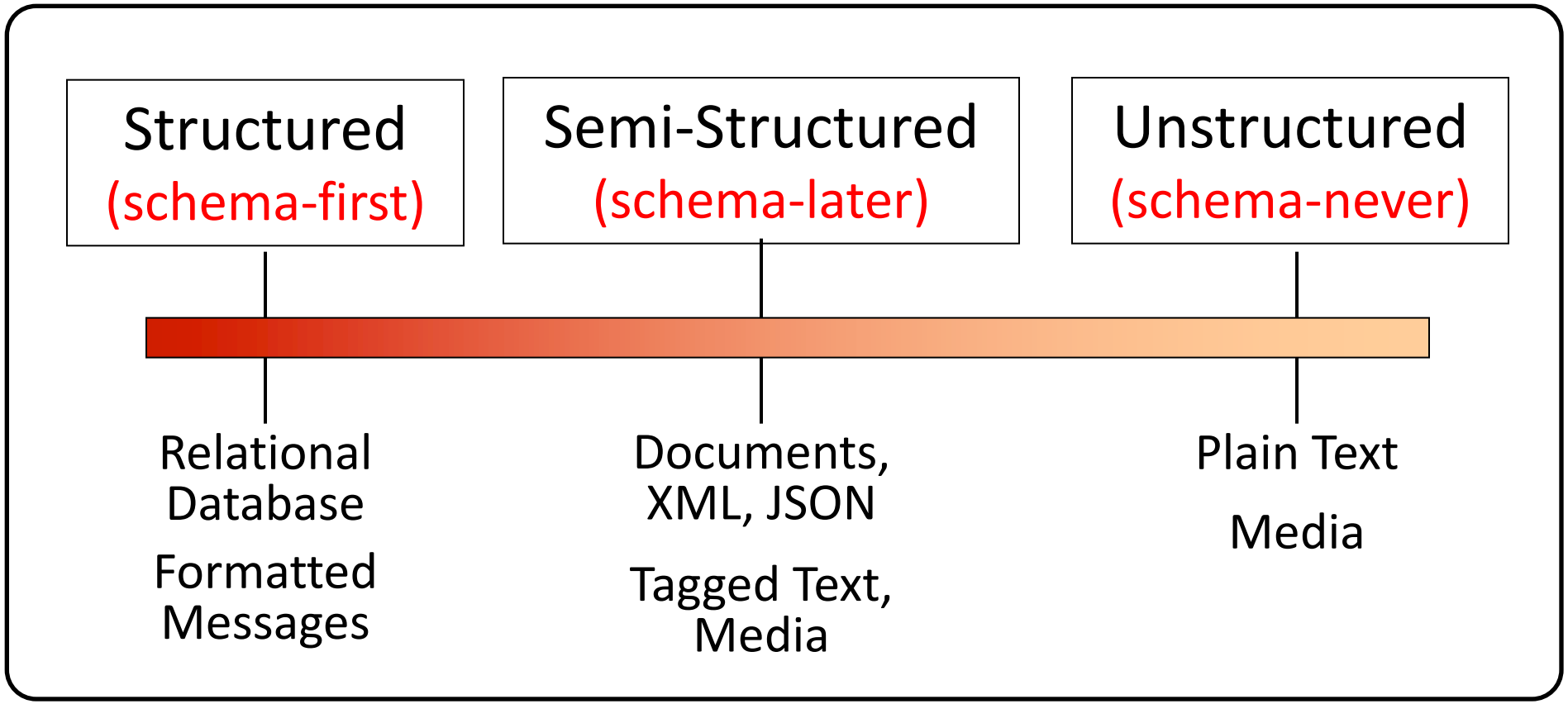
- use some sort of ***semi-structured data model***: XML/JSON
- the ***value can be examined*** and used by the system (unlike in key/data stores)

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

- hu**mongo**us => 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 <==> MongoDB equivalent

---

database <==> database

table <==> collection

row <==> document

attributes <==> fields (field-name:value pairs)

primary key <==> 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

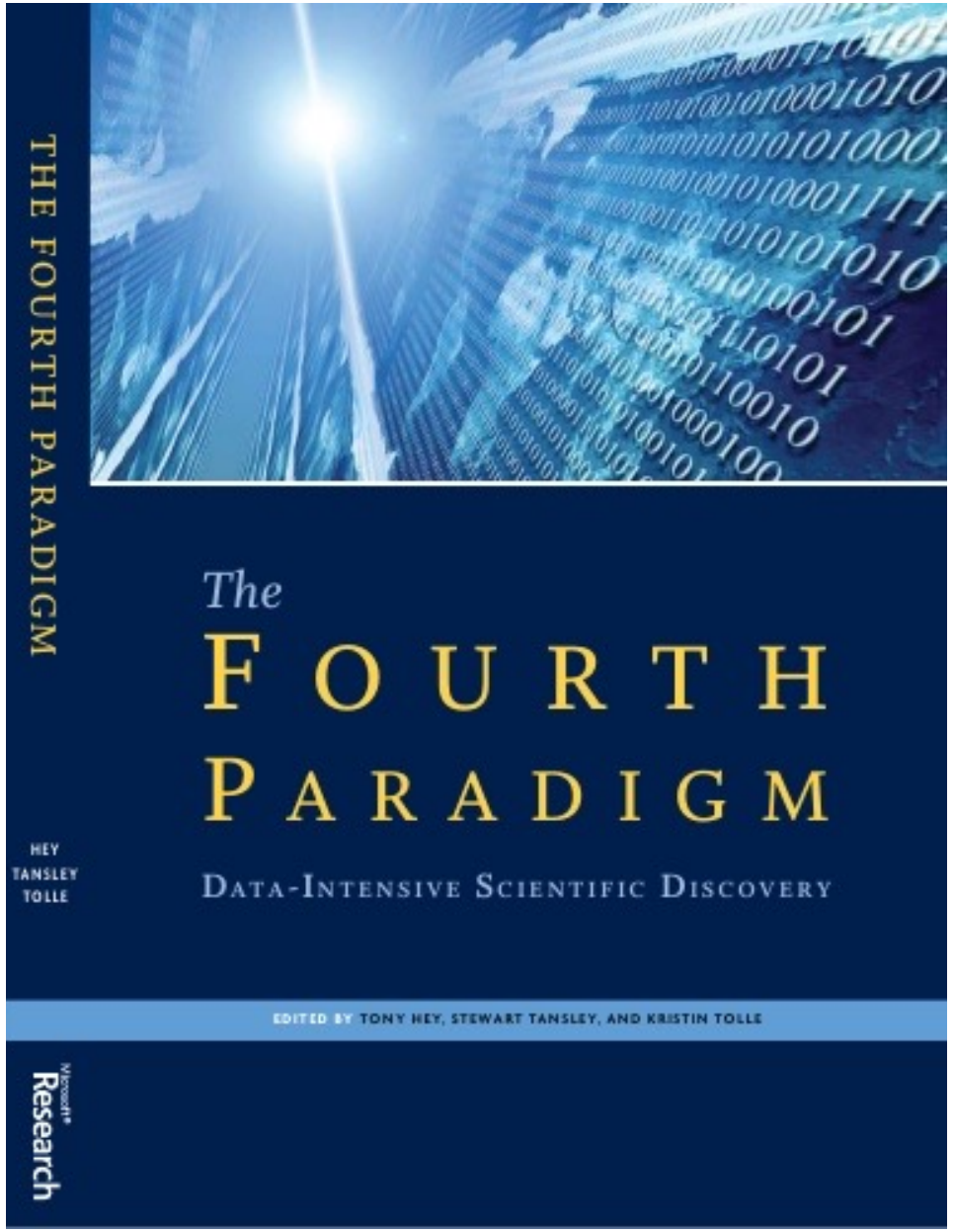
we spent a whole semester on Database Systems
what is next?

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.****"*

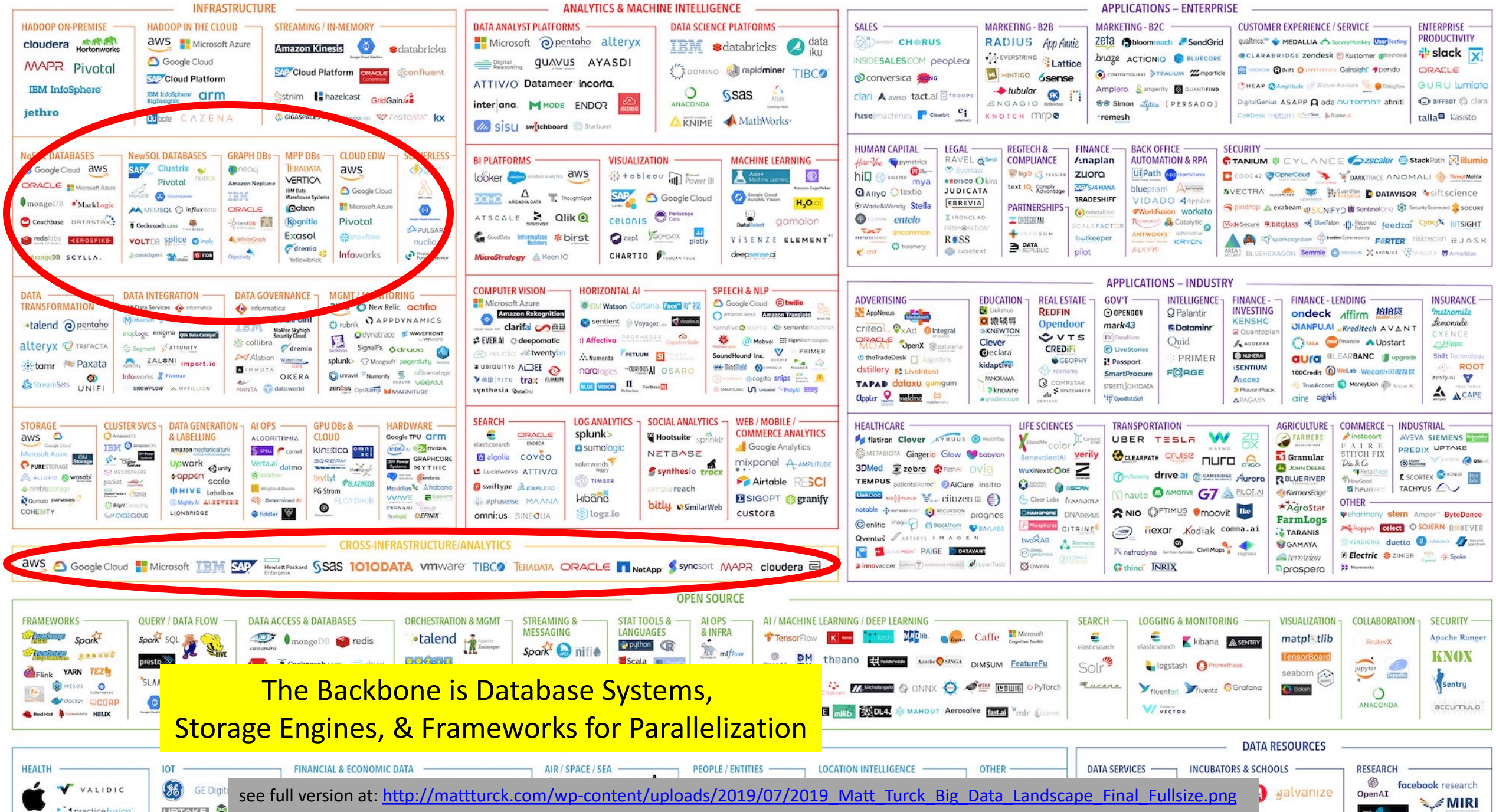
Faster Innovation through Data-Intensive Approaches

Need for Innovation in Data Management!

DATA & AI LANDSCAPE 2019

see full version at: http://mattturck.com/wp-content/uploads/2019/07/2019_Matt_Turck_Big_Data_Landscape_Final_Fullsize.png

DATA & AI LANDSCAPE 2019



see full version at: 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”

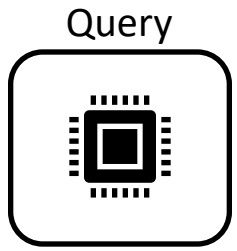
use more machines

Scale Up Execution

how to use more cores (threads)?

inter-query parallelism

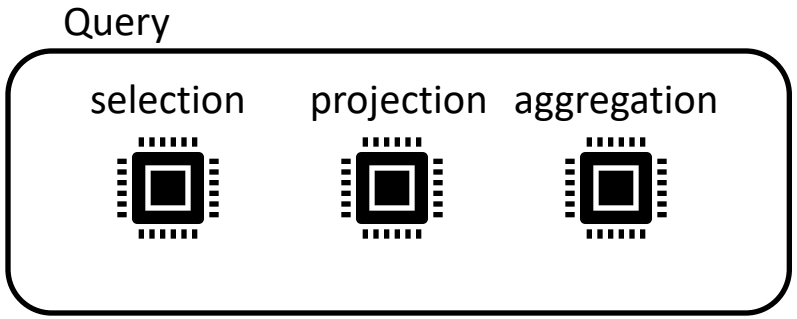
each query runs on one processor



inter-operator parallelism

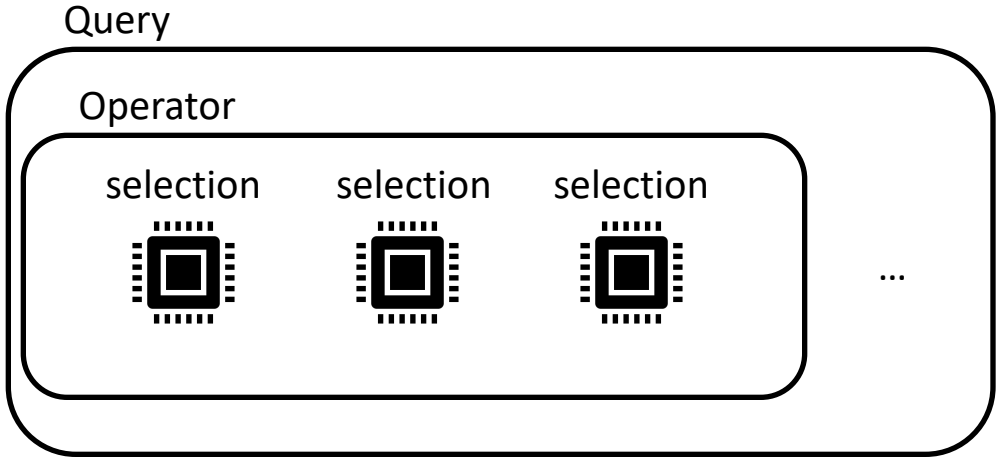
each query runs on multiple processors

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

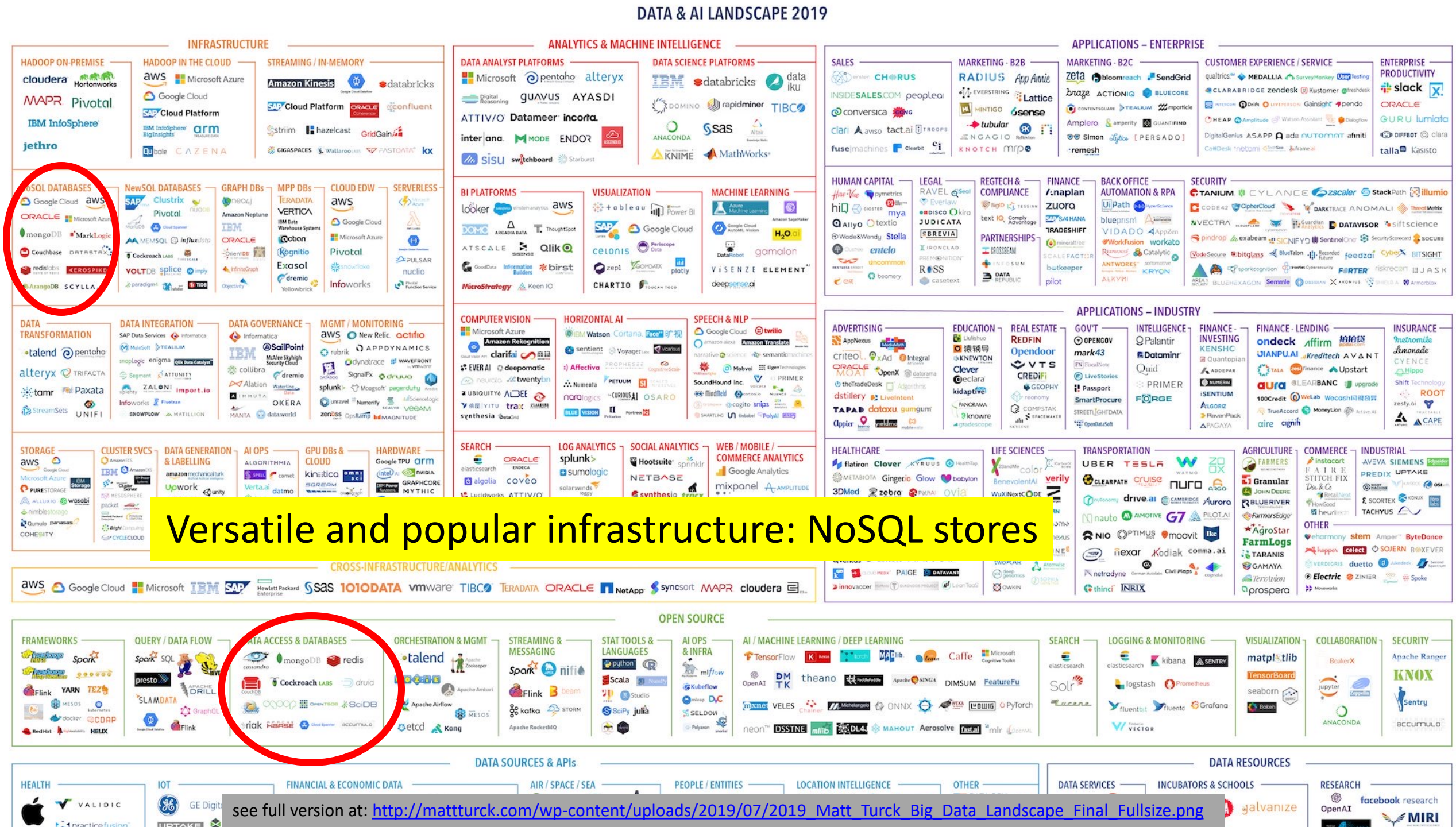
how to distribute data accesses?

Scale Out

similar questions across machines

new bottlenecks?

move data across machines: network!


DATA & AI LANDSCAPE 2019
INFRASTRUCTURE
HADOOP ON-PREMISE
HADOOP IN THE CLOUD
STREAMING / IN-MEMORY
NoSQL DATABASES
NewSQL DATABASES
GRAPH DBs
MPP DBs
CLOUD EDW
SERVERLESS
DATA TRANSFORMATION
DATA INTEGRATION
DATA GOVERNANCE
MGMT / MONITORING
STORAGE
CLUSTER SVCS
DATA GENERATION & LABELLING
AI OPS
GPU DBs & CLOUD
HARDWARE
ANALYTICS & MACHINE INTELLIGENCE
DATA ANALYST PLATFORMS
DATA SCIENCE PLATFORMS
BI PLATFORMS
VISUALIZATION
MACHINE LEARNING
COMPUTER VISION
HORIZONTAL AI
SPEECH & NLP
SEARCH
LOG ANALYTICS
SOCIAL ANALYTICS
WEB / MOBILE / COMMERCE ANALYTICS
APPLICATIONS – ENTERPRISE
SALES
MARKETING - B2B
MARKETING - B2C
CUSTOMER EXPERIENCE / SERVICE
ENTERPRISE PRODUCTIVITY
HUMAN CAPITAL
LEGAL
REGTECH & COMPLIANCE
FINANCE
BACK OFFICE AUTOMATION & RPA
SECURITY
APPLICATIONS – INDUSTRY
ADVERTISING
EDUCATION
REAL ESTATE
GOV'T
INTELLIGENCE
FINANCE - INVESTING
FINANCE - LENDING
INSURANCE
HEALTHCARE
LIFE SCIENCES
TRANSPORTATION
AGRICULTURE
COMMERCE
INDUSTRIAL
OTHER
Versatile and popular infrastructure: NoSQL stores
CROSS-INFRASTRUCTURE/ANALYTICS
OPEN SOURCE
FRAMEWORKS
QUERY / DATA FLOW
DATA ACCESS & DATABASES
ORCHESTRATION & MGMT
STREAMING & MESSAGING
STAT TOOLS & LANGUAGES
AI OPS & INFRA
AI / MACHINE LEARNING / DEEP LEARNING
SEARCH
LOGGING & MONITORING
VISUALIZATION
COLLABORATION
SECURITY
DATA SOURCES & APIs
HEALTH
IOT
FINANCIAL & ECONOMIC DATA
AIR / SPACE / SEA
PEOPLE / ENTITIES
LOCATION INTELLIGENCE
OTHER
DATA RESOURCES
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

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