Good morning!!

CS 744: FLINK

Shivaram Venkataraman

Spring 2024

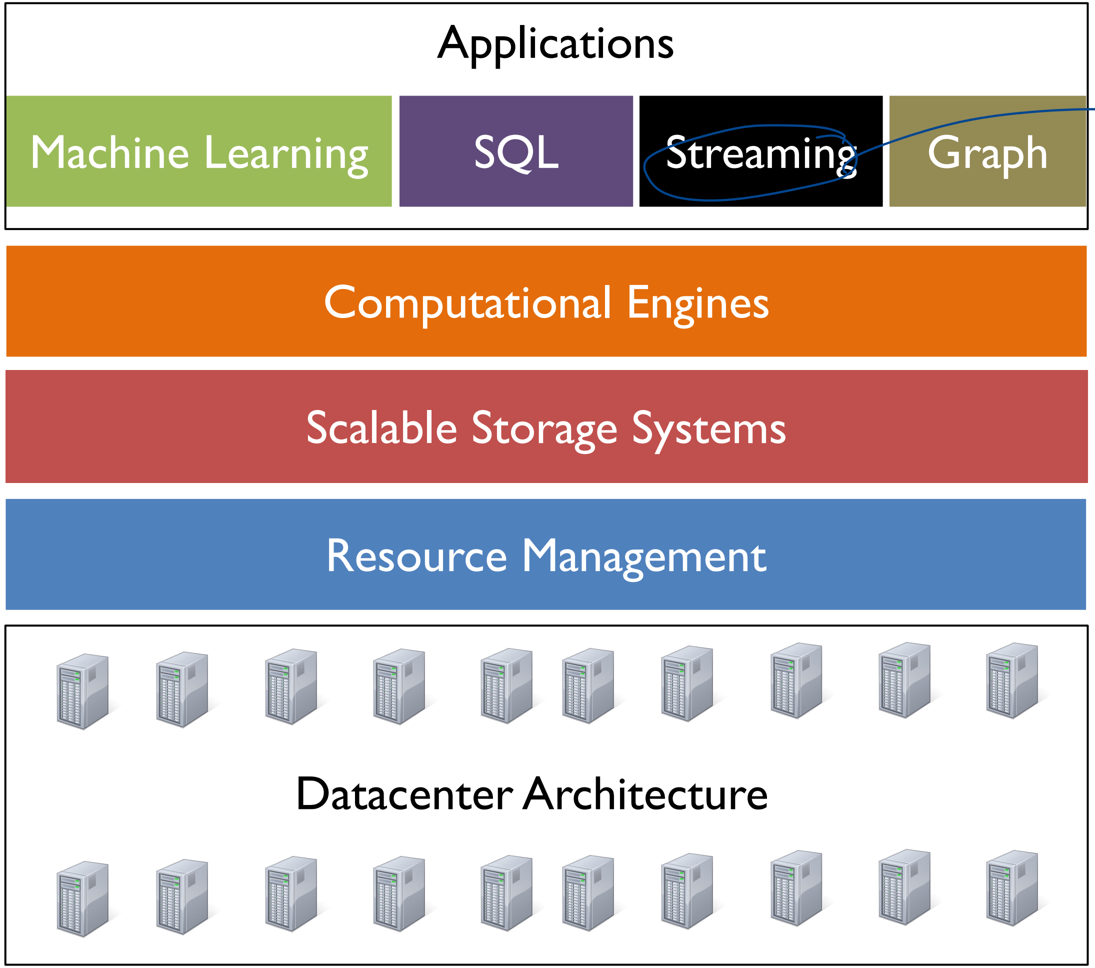
ADMINISTRIVIA

Grading

- Assignment 2 grading → released
- Course Project Proposal feedback → today / tomorrow
- Midterm → next week

Resources for Course Projects

- Cloudlab (Reservations?) → Survey ?
- GCP credits (Email Tzu-Tao and me) → link



Applications

Machine Learning

SQL

Streaming

Graph

Computational Engines

Scalable Storage Systems

Resource Management

Datacenter Architecture

API

features

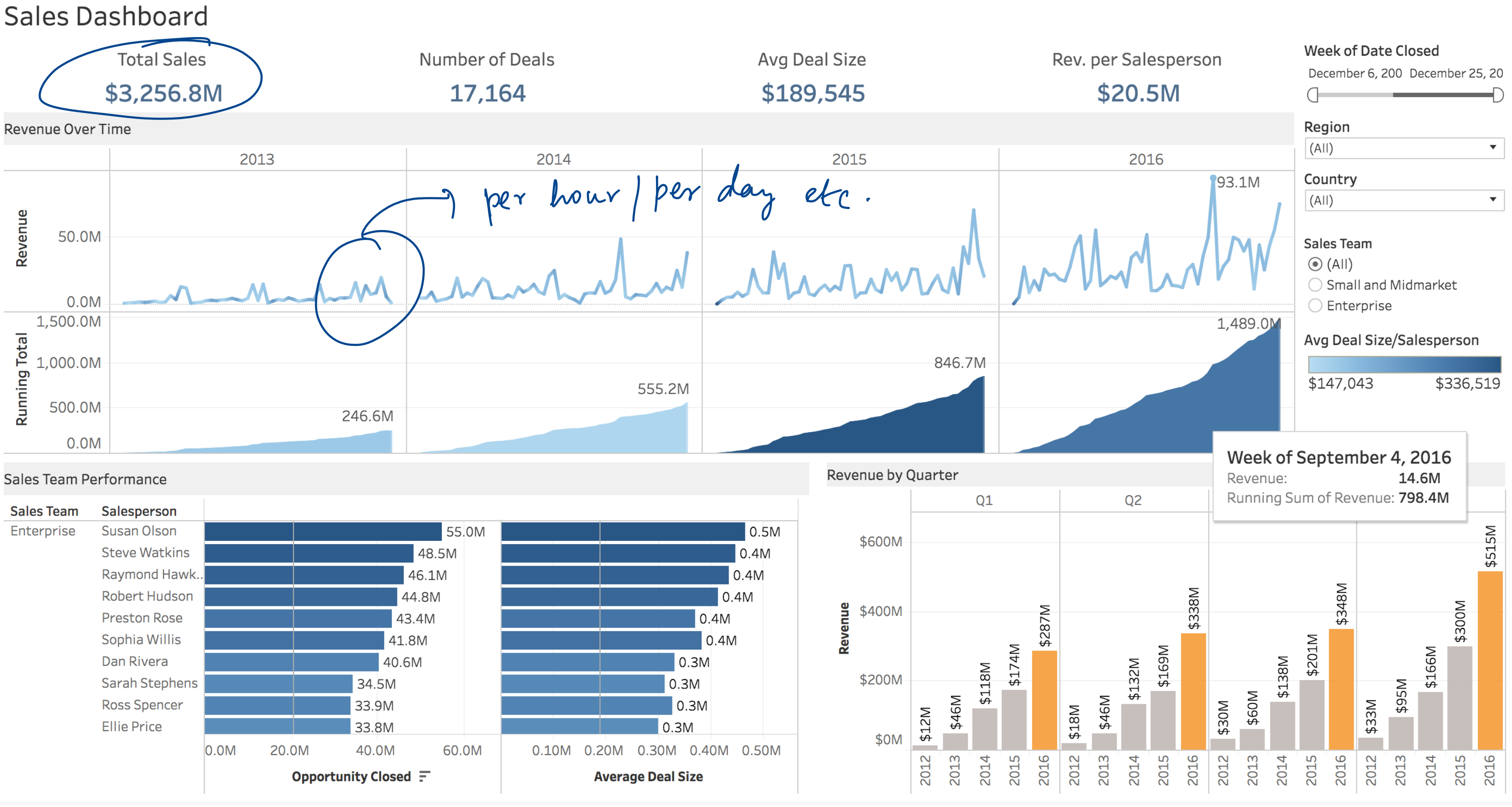
Windowing

Triggers

- Watermark

DASHBOARDS

→ update regularly



Sales Dashboard

Total Sales	Number of Deals	Avg Deal Size	Rev. per Salesperson
$3,256.8M	17,164	$189,545	$20.5M

per hour | per day etc.

Week of Date Closed
December 6, 200 December 25, 20

Region
(All)

Country
(All)

Sales Team
● (All)
○ Small and Midmarket
○ Enterprise

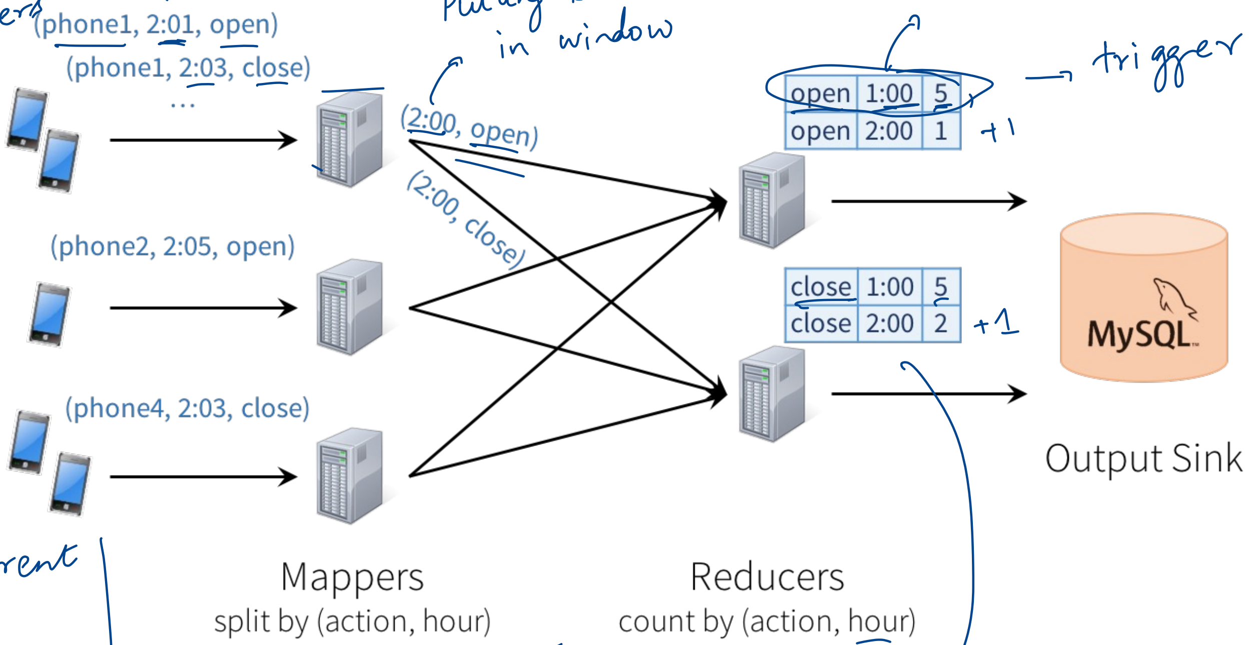
Avg Deal Size/Salesperson
$147,043 $336,519

Week of September 4, 2016
Revenue: 14.6M
Running Sum of Revenue: 798.4M

STREAMING COMPUTATION

① MR → wait for all mappers before reducers

→ Intermediate data disk. Streaming them here!



open	1:00	5
open	2:00	1

+1

close	1:00	5
close	2:00	2

+1

event TS

Putting it in window

windows on event ts

→ trigger

Here: we are doing them concurrently

Database / dashboard etc.

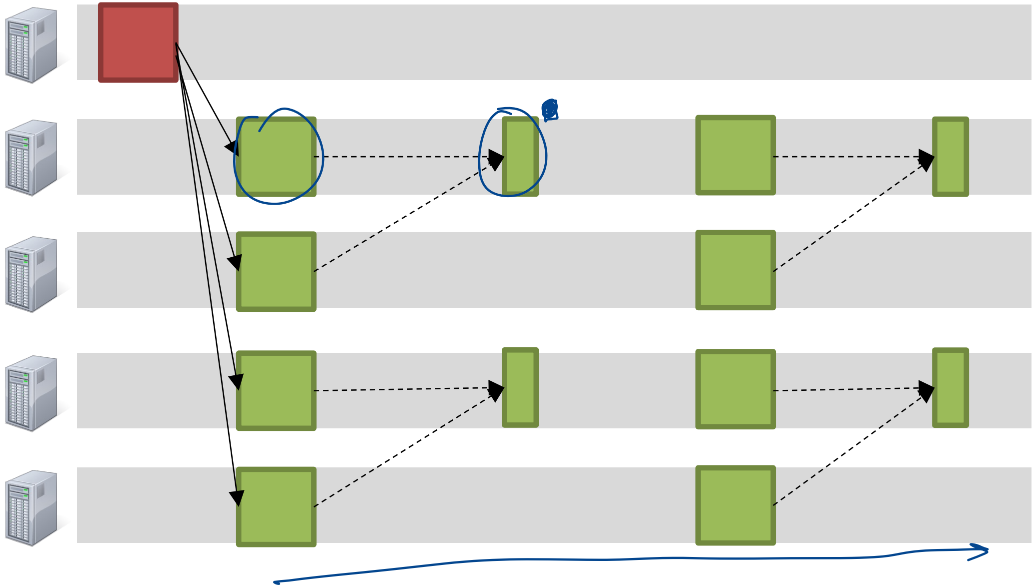
mutable state that is present in operators

② FT is different

③ Bottlenecks

④ Input source

FLINK: COMPUTATION MODEL



🟥 Manager/Driver

🟩 Task/Computation

⟶ Control Message

⇢ Network Transfer

Started at beginning

Long-lived operators

Mutable State

Google MillWheel

Flink

Streaming DBs: Borealis, Flux etc

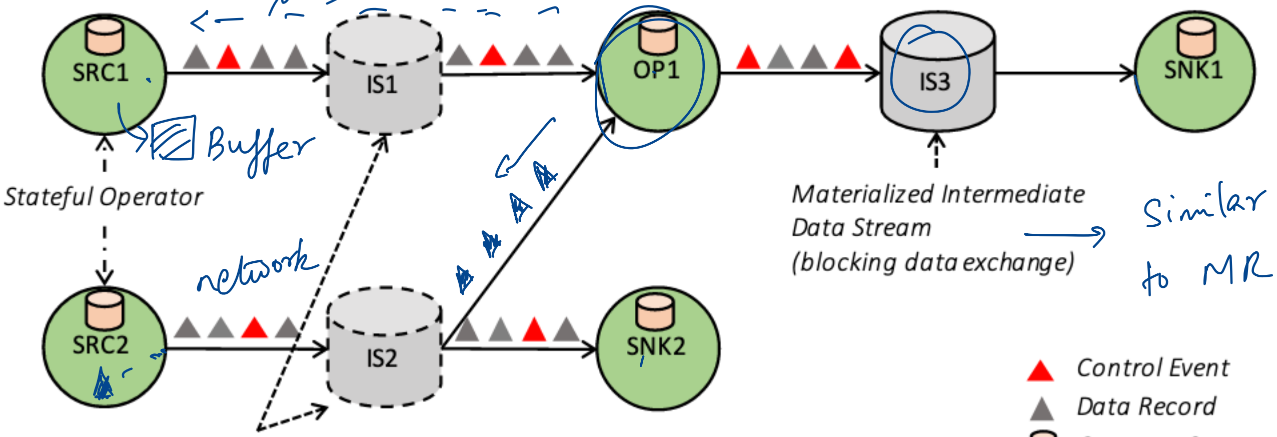
Naiad

INTERMEDIATE DATA STREAMS

10 tuples /sec

~~8~~ 5

5 tuples /sec



Stateful Operator

Materialized Intermediate Data Stream (blocking data exchange) → Similar to MR

Transient Intermediate Data Stream (pipelined data exchange)

🟥 Control Event

▲ Data Record

🛢 Operator State

Data Stream

Transient

↳ Pipeline

Comm. with computation

↳ 1-1 or 1-many or many-1 dependency

Trigger

- Buffer full
- Timeout [20 ms]

Back pressure

- signal that goes from dest to src to say slow down

STATEFUL OPERATORS

Examples?

→ Windowing operations

→ Aggregation

Challenge

How to ensure fault tolerance?

Explicitly register local variables

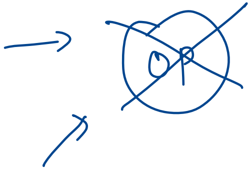
StateBackends that are automatically saved/recovered

→ annotate some variables in operator as state variables → Periodically checkpoint state variables

Stateless operators

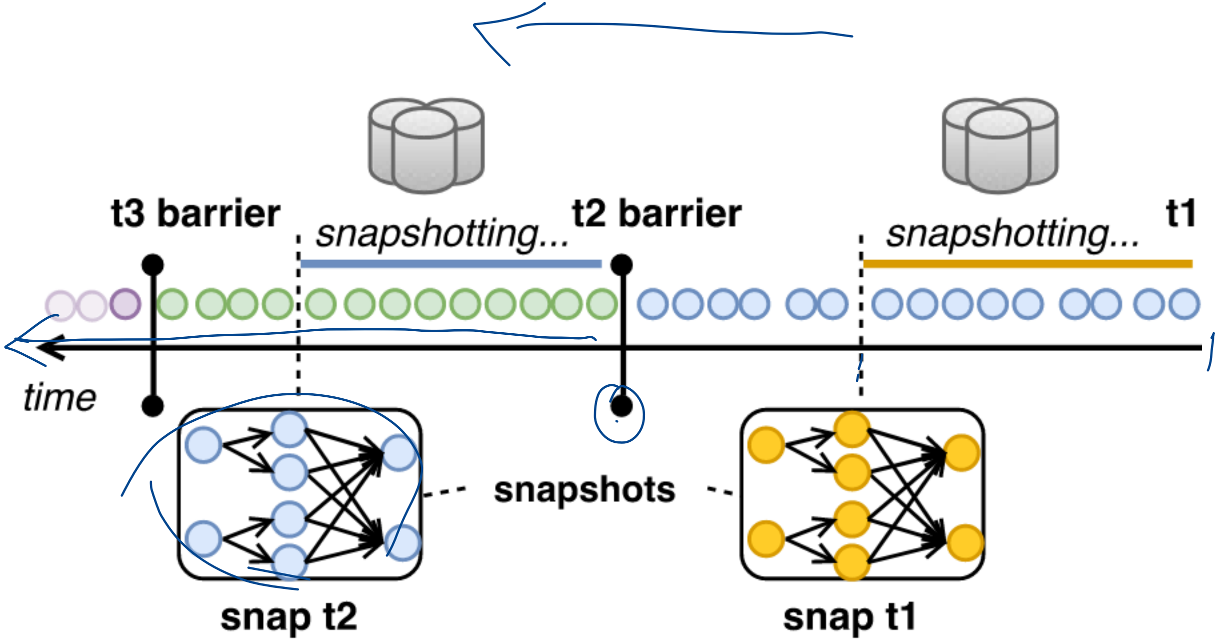
→ map() → input tuple ↓ output tuple

→ filter()



→ what do we do?

FAULT TOLERANCE: CHECKPOINTING



Guarantee that we want:

→ Exactly once semantics

↳ useful

each input's effect is seen exactly once in the output!

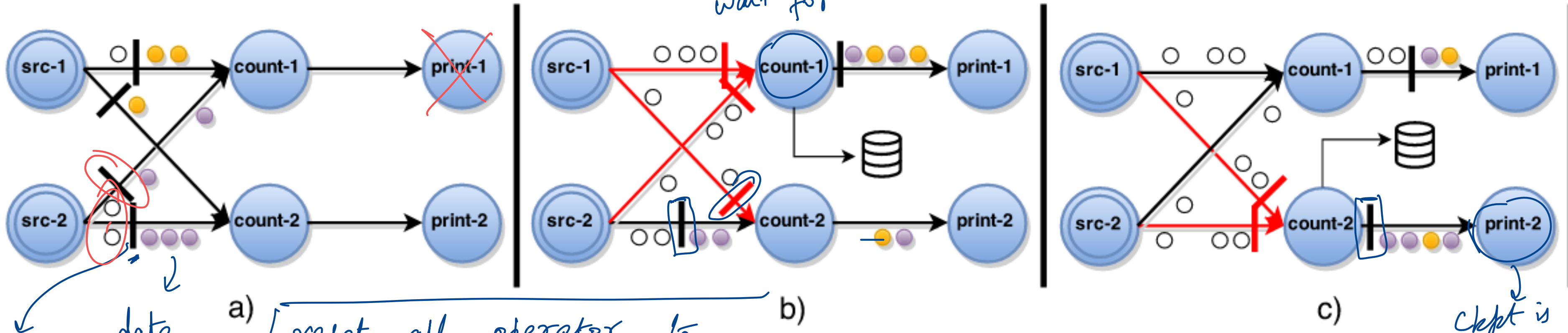
Periodically take checkpoints → Restart the operator and replay tuples that came after checkpoint.

→ Input is replayable!

ASYNCHRONOUS BARRIER SNAPSHOTTING

all operators have processed upto some tuple → consistent snapshot

wait for



check point

data records

reset all operator to check point state

ckpt is complete

● Preshot records src-1

● Preshot records src-2

○ Postshot records

Operator snapshot

| Snapshot barrier

→ Blocked channel

① Control message to ckpt state is saved when message is recvd.

② checkpointing takes time / slowdown during ckpt

WATERMARKS, WINDOWS

Implements similar model as Dataflow

outside of Flink

" Watermarks originate at the sources of a topology"

Propagate through the other operators of dataflow

same as in Dataflow paper.

Windows based on event-time, processing time, ingest time(?)

```
stream
.window(SlidingTimeWindows.of(
        Time.of(6, SECONDS), Time.of(2, SECONDS))
.trigger(EventTimeTrigger.create())
```

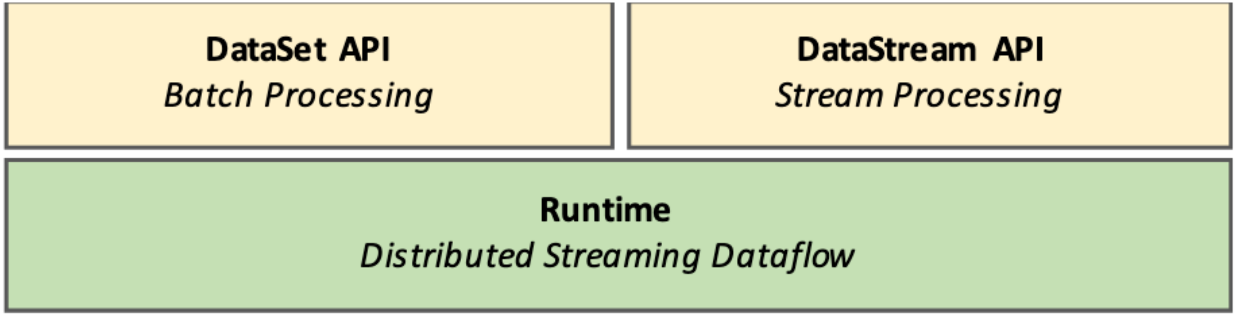
COMBINING BATCH, STREAMING

Blocked DataStreams → intermediate data to disk

Turn off periodic snapshots

Blocking operators (e.g., sort)

↳ batch specific operators

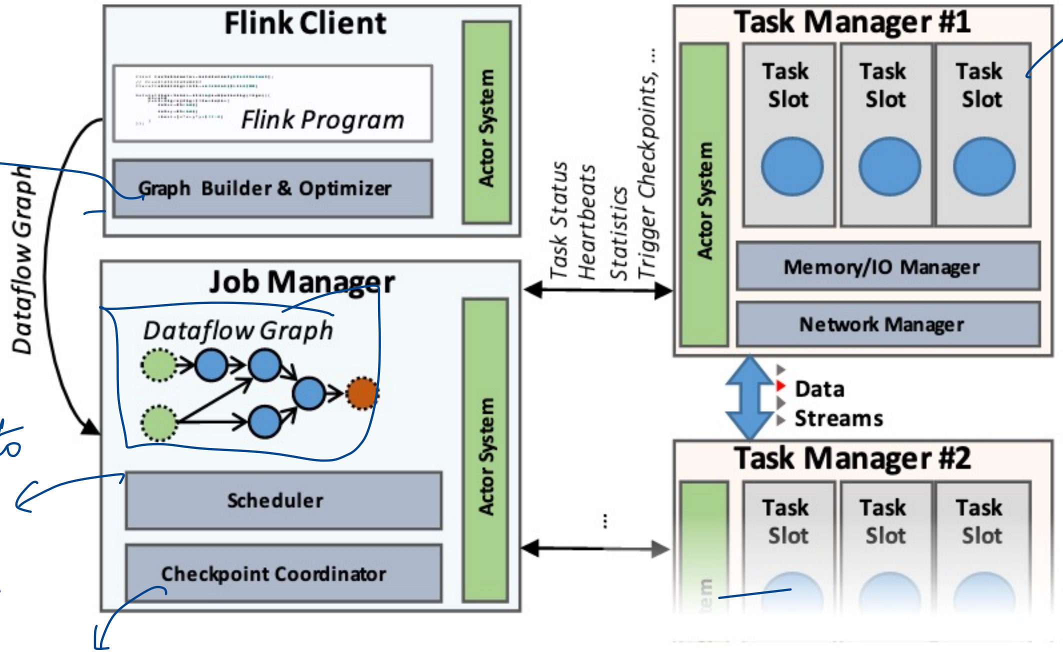


DataSet API
Batch Processing

DataStream API
Stream Processing

Runtime
Distributed Streaming Dataflow

OVERALL ARCHITECTURE



Scope compile / optimize

similar to shadow master ?

latest checkpoint ?

SUMMARY

Stream processing → Increasingly important workload trend

Flink: Distribtuted streaming dataflow to run streaming, batch, iterative

Distribtuted streaming dataflow

- Stateful operators
- Checkpointing based FT

DISCUSSION

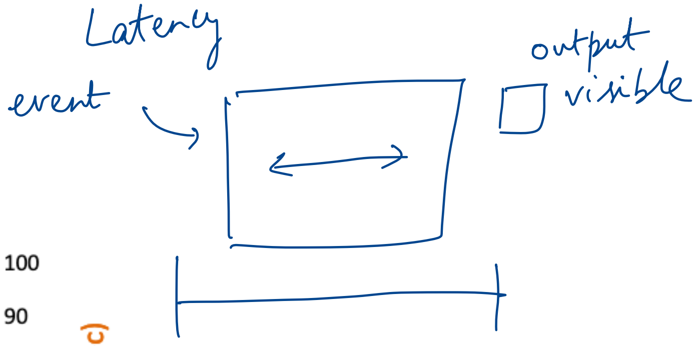
https://forms.gle/j9Z7rm4qQpogbz5W8

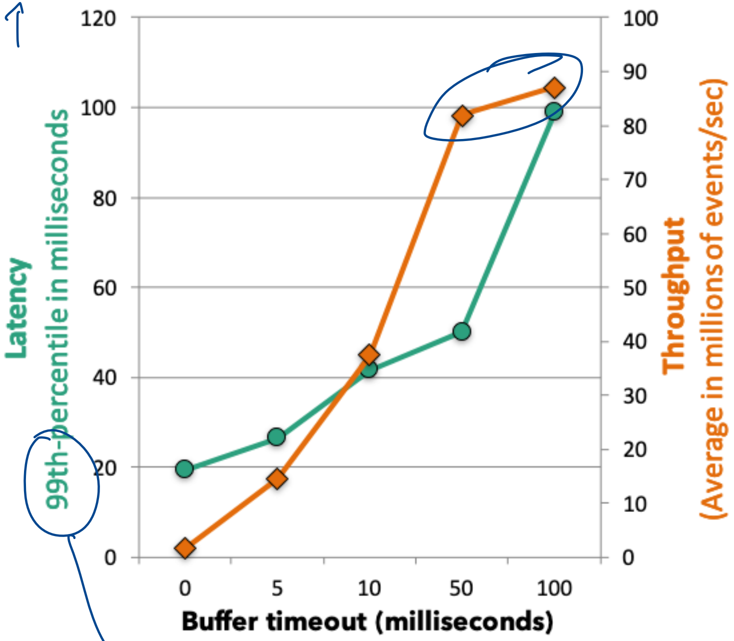
① Timeout ↑

- higher latency ↑
- higher tput ↑

② Tput wins flatten out

- You might buffer size limit before ?





worst case behavior

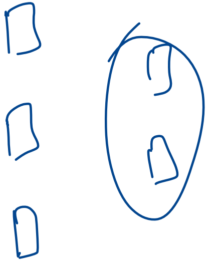
Consider you are implementing a micro-batch streaming API on top of Apache Spark. What are some of the bottlenecks/challenges you might have in building such a system?

→ Events to be processed fast

→ Tasks launched when prev. stage finishes ??

→ Windows ?

→



→ output for window → data arrives out of order ??

→ Data shuffling? → each batch is new data

→ State across batches? → memory ?

SUMMARY

Next week: Spring break!!

Next class: Spark Streaming