

Hello!

# CS 744: FLINK

Shivaram Venkataraman

Spring 2025

# ADMINISTRIVIA

## Grading

- Assignment 2 grading → Done. Today?
- Course Project Proposal feedback → end of this week
- Midterm → after Spring break

## Resources for Course Projects

- Cloudlab (Reservations?)

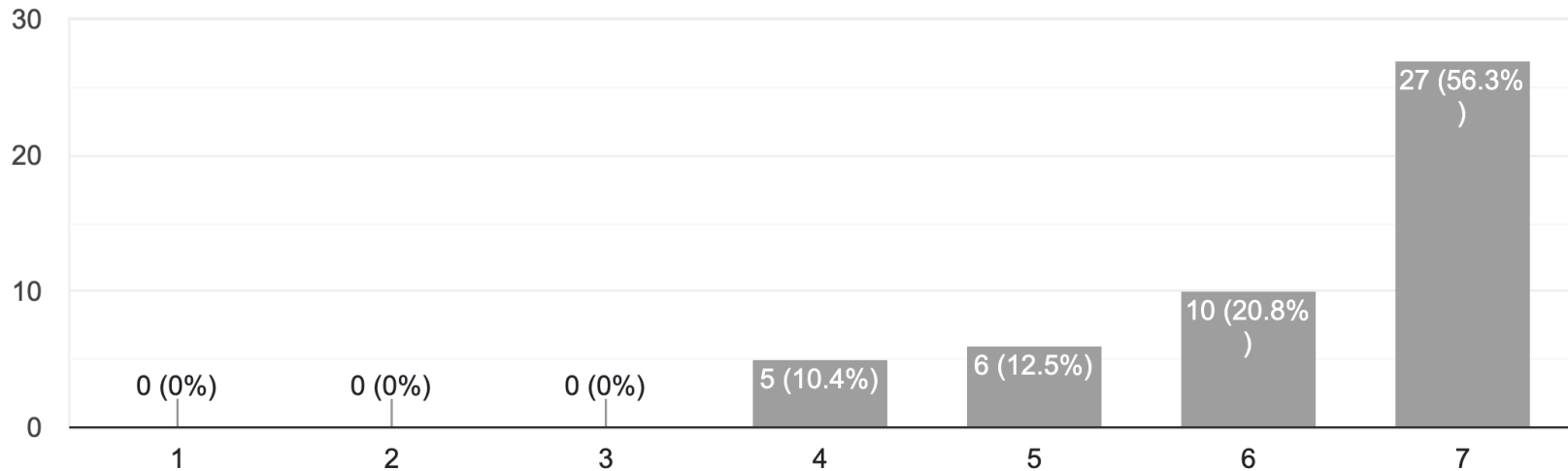
- Azure credits

↳ Documentation  
↳ not course but per student \$100

# MID-SEMESTER SURVEY

Overall how useful are you finding the course so far?

48 responses

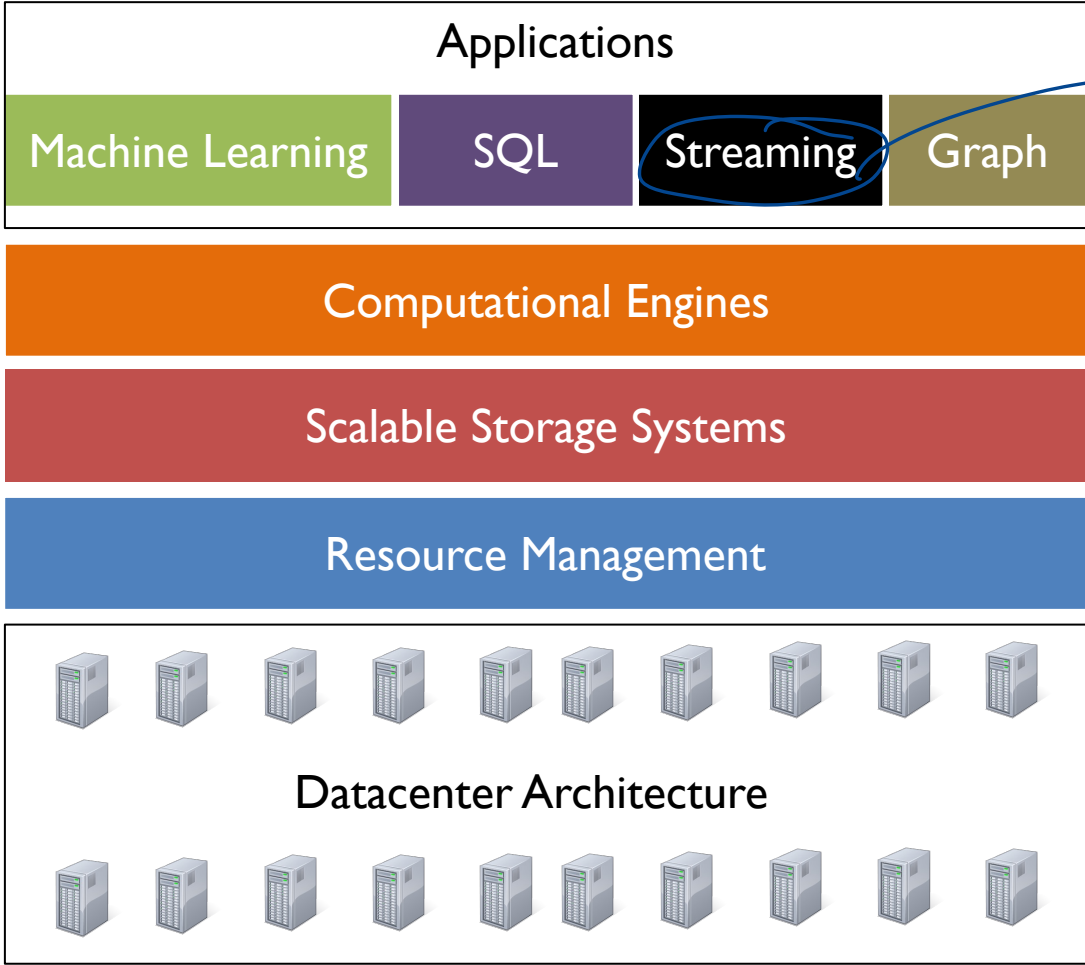


## *Hindrances*

- ...not very familiar with the sys topics..*
- ... lack of background knowledge on some topics...*
- ... more time to read papers for the Thursday lecture...*

## *Suggestions*

- ...expand the regions of our discussion...*
- ...more what-if and thought provoking questions for discussions...*
- ...discuss our findings from the reading reviews ...instead of answering new questions*
  
- ...I'd like a smaller paper reading group...*
- ...Paper reading group sounds great...*
  
- ... bring the papers to the next exam...*



→ Dataflow  
API  
Semantics for  
stream  
processing

# DASHBOARDS

*updates few seconds*

## Sales Dashboard

Total Sales

**\$3,256.8M**

Number of Deals

17,164

Avg Deal Size

**\$189,545**

Rev. per Salesperson

**\$20.5M**

Week of Date Closed

December 6, 200 December 25, 20



Region

(All)

Country

(All)

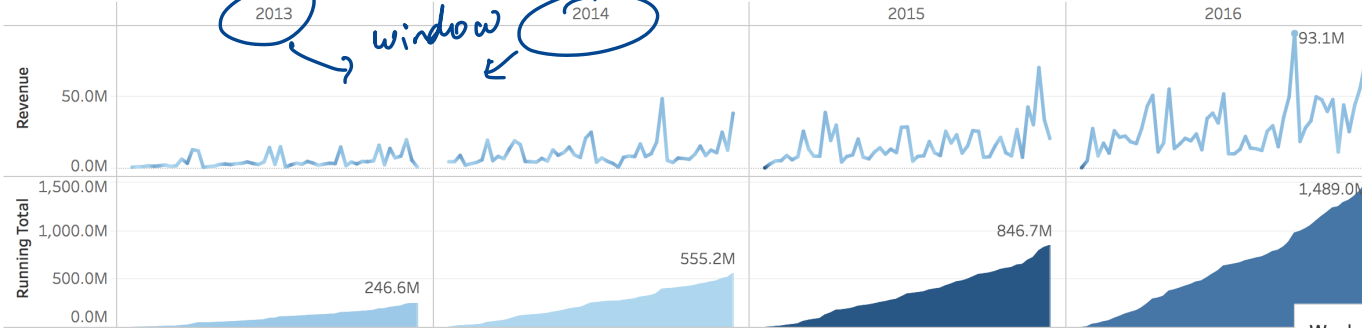
Sales Team

- (All)
- Small and Midmarket
- Enterprise

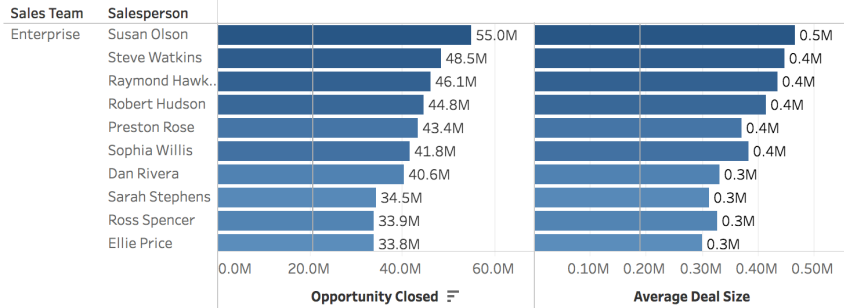
Avg Deal Size/Salesperson



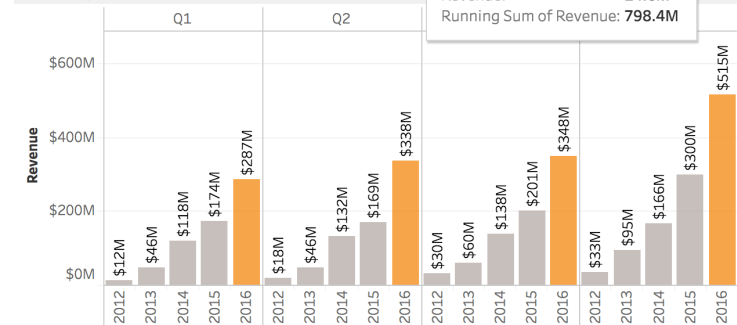
Revenue Over Time



Sales Team Performance

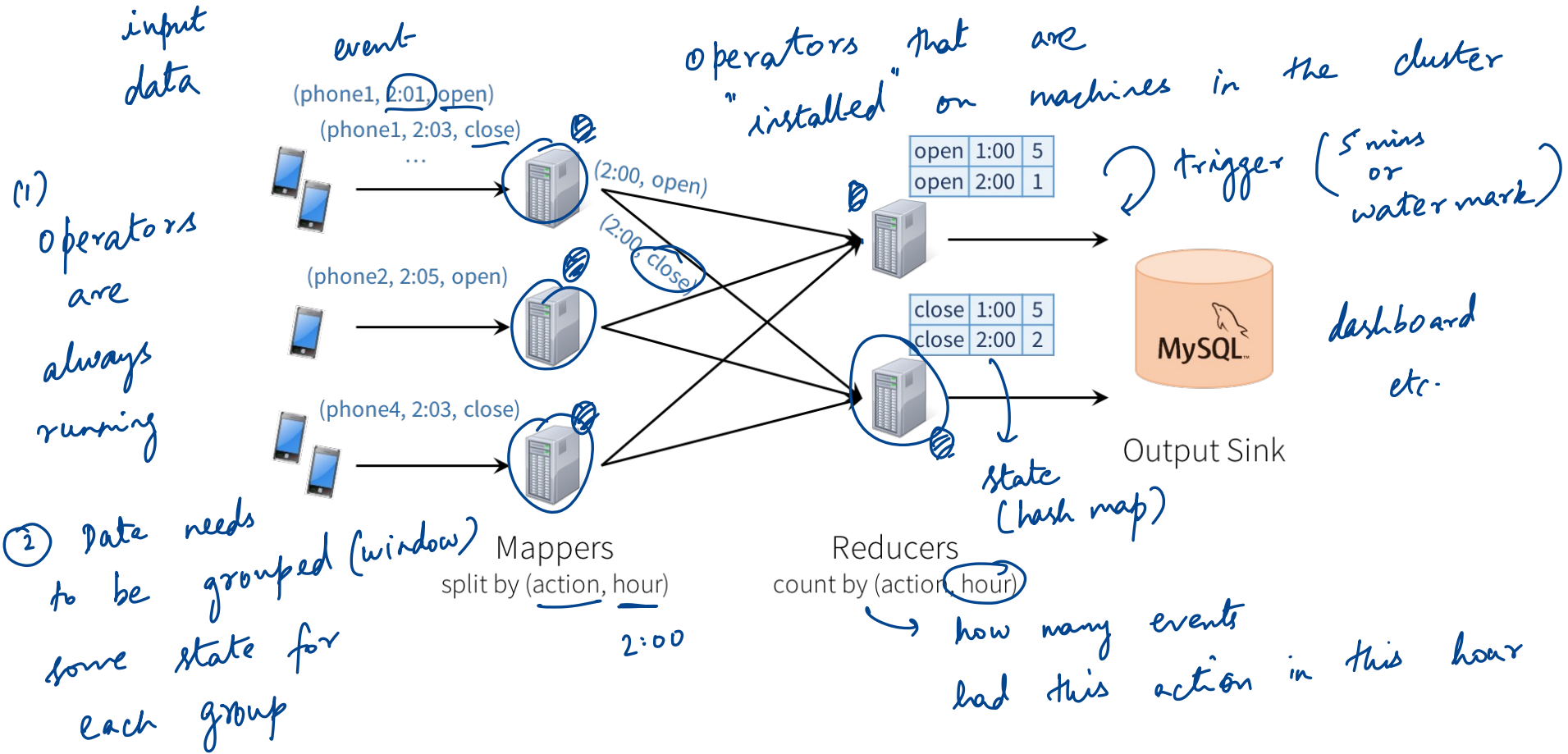


Revenue by Quarter

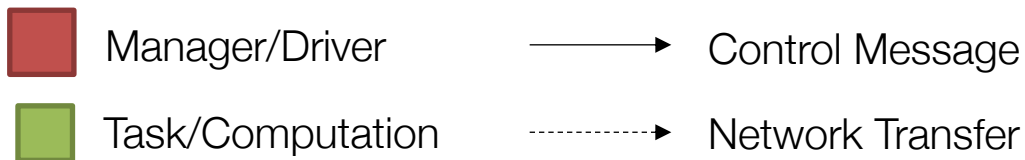
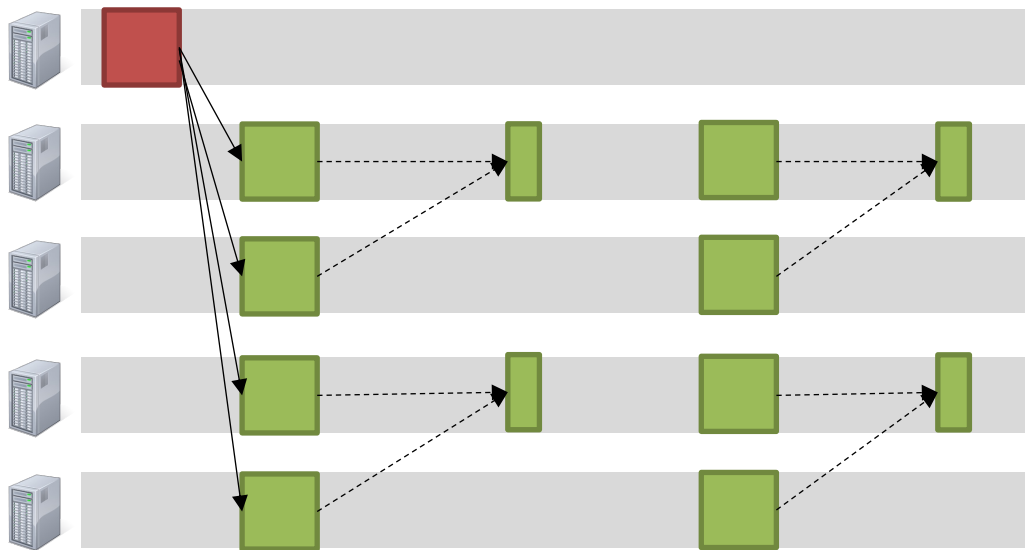


Week of September 4, 2016  
 Revenue: 14.6M  
 Running Sum of Revenue: 798.4M

# STREAMING COMPUTATION



# FLINK: COMPUTATION MODEL



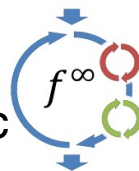
Long-lived operators

Mutable State

Google  
MillWheel



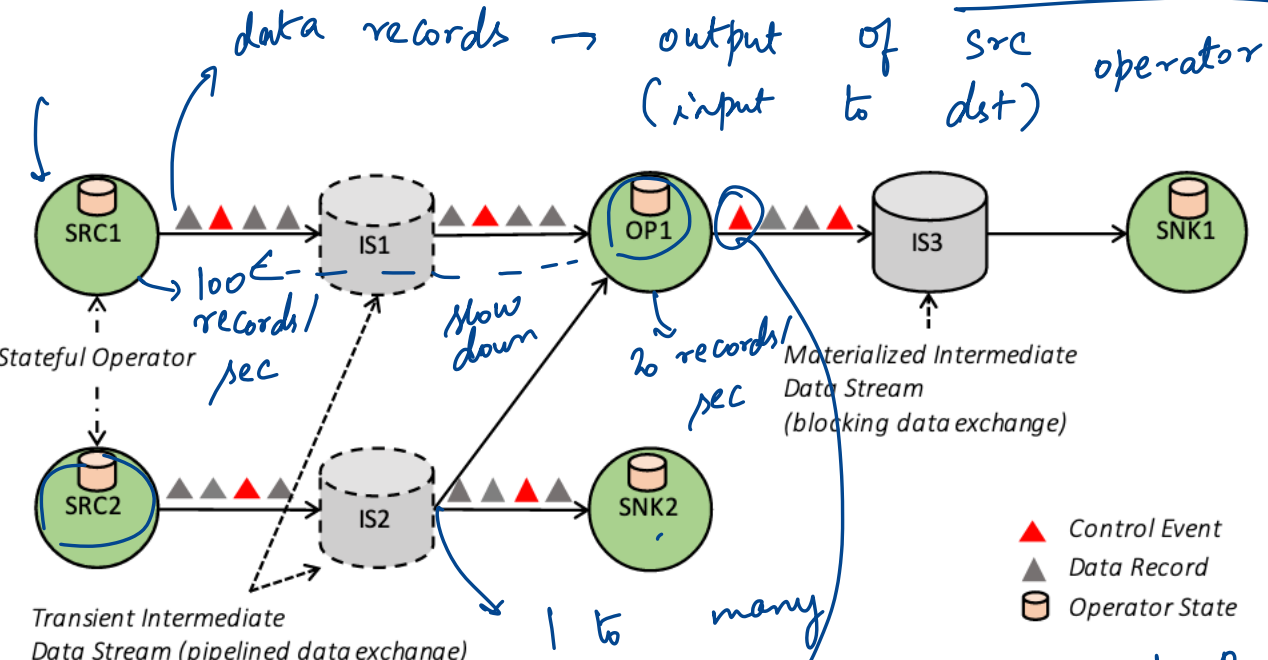
Streaming DBs:  
Borealis, Flux etc



Naiad

# INTERMEDIATE DATA STREAMS

FIFO → TCP



Query (user submits)

↓  
DAG of operators

→ 1-1 or  
1-many or  
many-1

- ▲ Control Event
- ▲ Data Record
- Operator State

Take a snapshot

watermark

back pressure

send an event  
dest to src

Buffered Streams

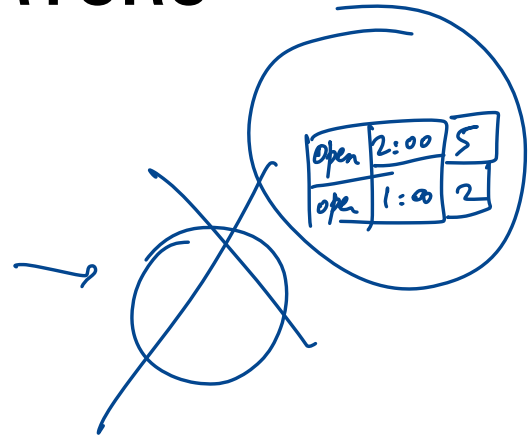
↳ Buffer size [8192 bytes]

↳ Timeout [50 ms]

# STATEFUL OPERATORS

Examples?

- ↳ Windowing operators
- ↳ Aggregations etc.



Challenge

How to ensure fault tolerance?

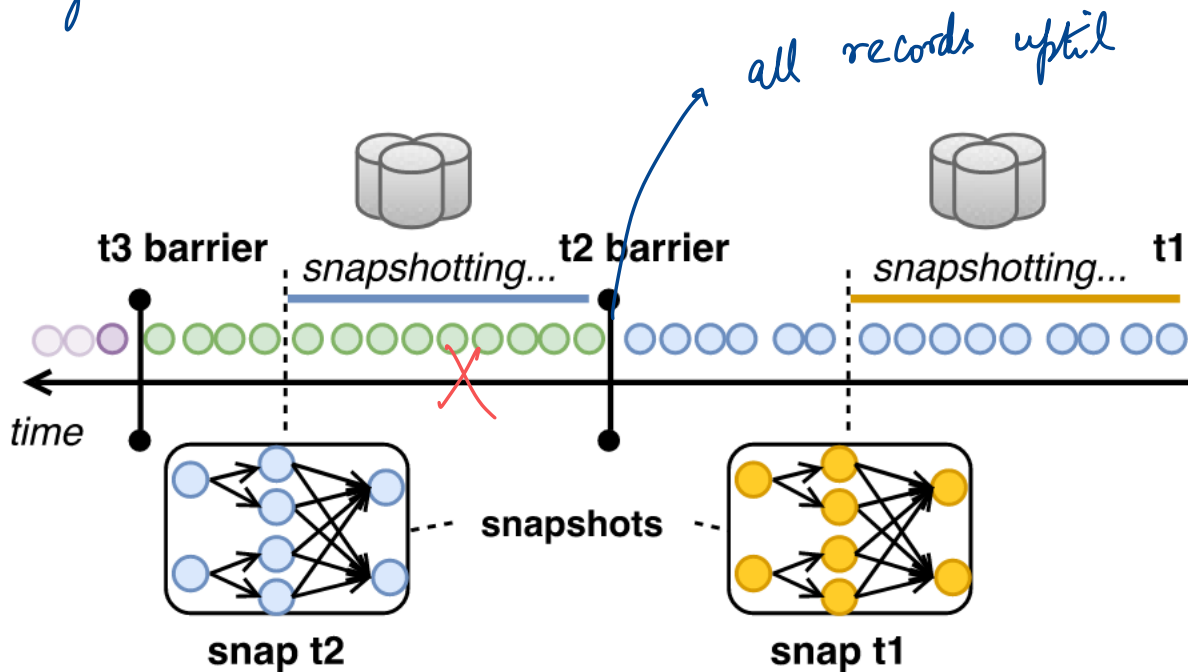
Explicitly register local variables

StateBackends that are automatically saved/recovered

annotate → this is state that should be checkpointed / restored

# FAULT TOLERANCE: CHECKPOINTING

log



all records uptil here are part of the snapshot

Failure

→ reset the input source to this barrier

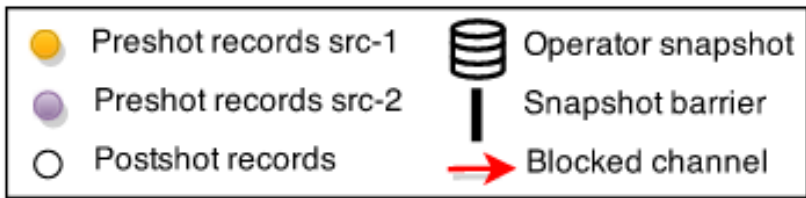
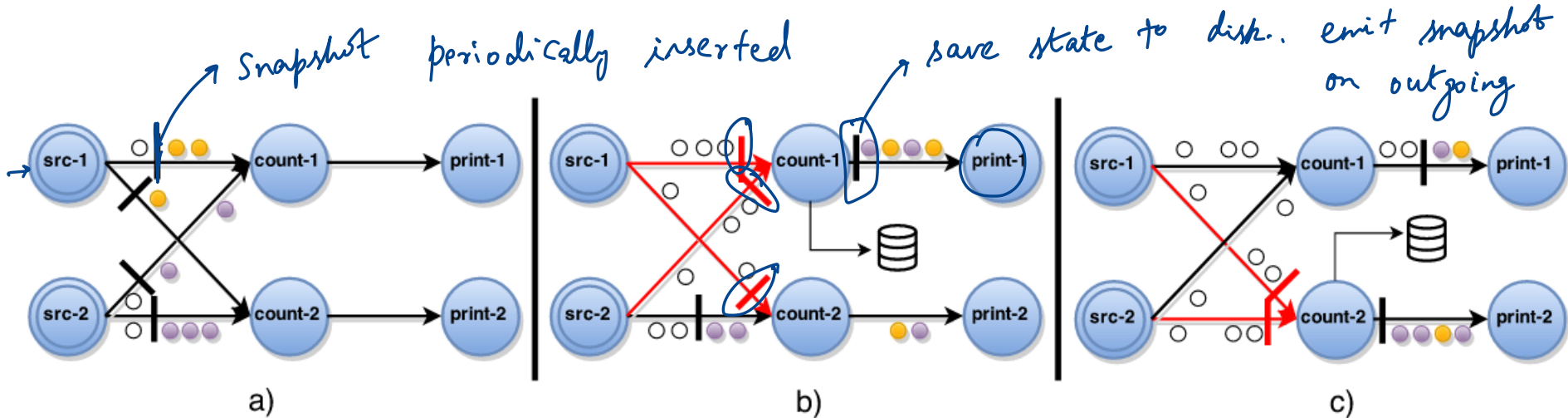
→ reset state of all ops at the barrier

→ Exactly once semantics → strong  
↳ at least semantics

→ one failure. All operators are restarted

→ replay events

# ASYNCHRONOUS BARRIER SNAPSHOTTING



many srcs at an operator  
 → block until you get snapshot from all sources

# WATERMARKS, WINDOWS

Implements similar model as Dataflow

“Watermarks originate at the sources of a topology”

Propagate through the other operators of dataflow

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

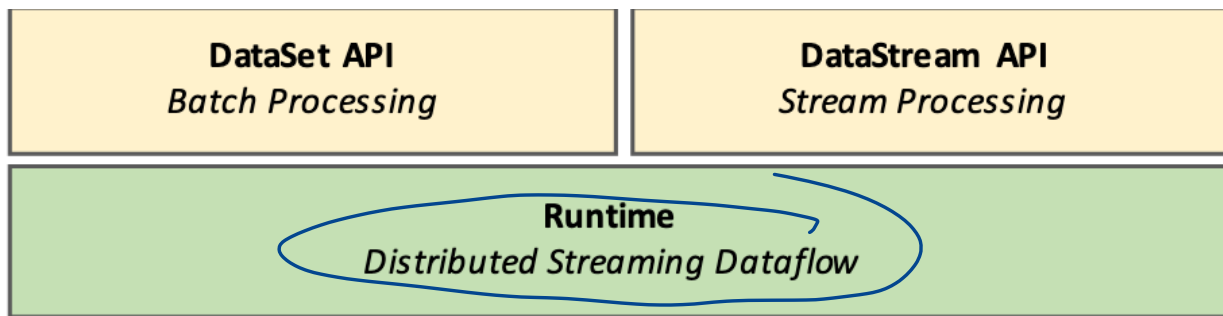
```
stream
  .window(SlidingTimeWindows.of(
    Time.of(6, SECONDS), Time.of(2, SECONDS))
  .trigger(EventTimeTrigger.create()))
```

# COMBINING BATCH, STREAMING

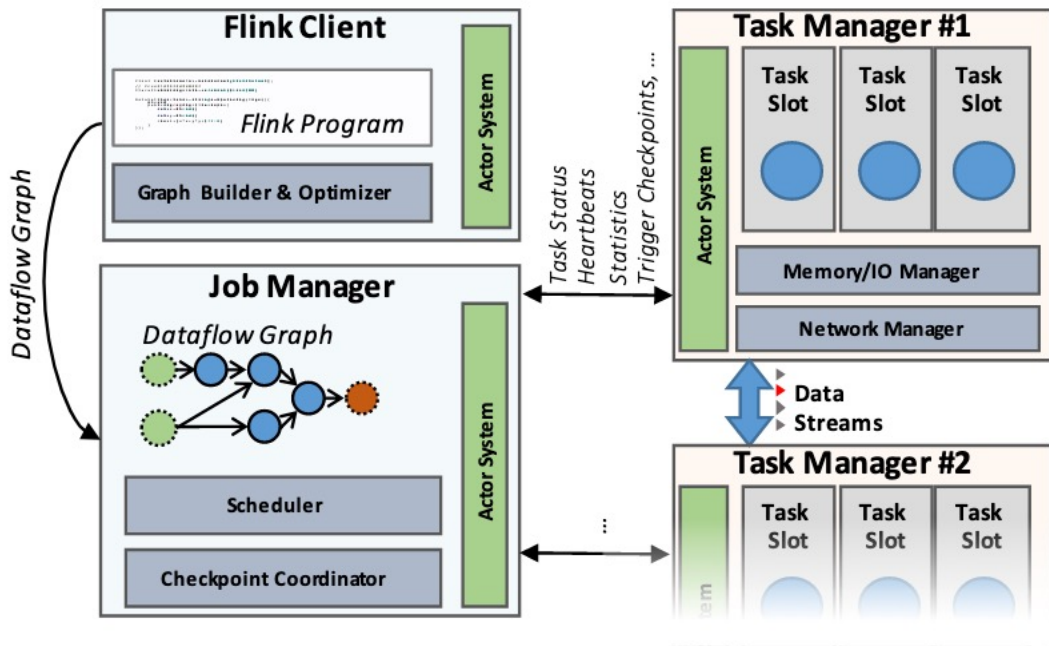
Blocked DataStreams → data is written to disk

Turn off periodic snapshots →

Blocking operators (e.g., sort)



# OVERALL ARCHITECTURE



# SUMMARY

Stream processing → Increasingly important workload trend

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

Distributed streaming dataflow

- Stateful operators
- Checkpointing based FT



# DISCUSSION

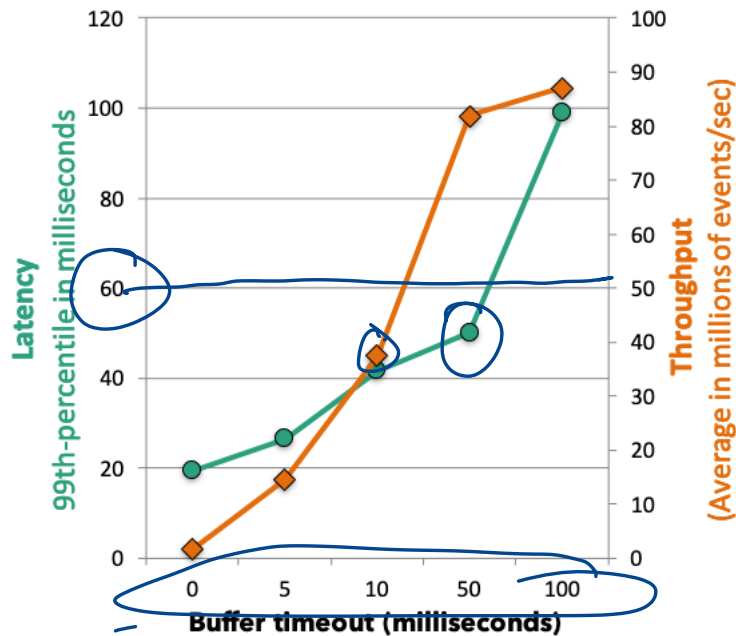
<https://forms.gle/EbTYYYd63zf3BnuE8>

make a legend!

sweet spot



latency



Buffer  $\uparrow$   
 $\rightarrow$  inc latency  
 $\rightarrow$  higher tput

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?

- keeping state ??
    - ↳ if state is RDD?
      - ↳ checkpoint the RDD?
        - ↳ or recompute it?
  - lazy computation → what trigger? → Timer
    - ↳ micro batch
  - Big Data
    - ↳ worthwhile?
    - ↳ overhead for running with small data?
- but RDDs are immutable

# SUMMARY

Next week: Spring break!!

Next class: Spark Streaming