Good morning!

CS 744: DATAFLOW

Shivaram Venkataraman

Fall 2021

ADMINISTRIVIA

- Assignment 2 grades are up! → Yien Yu, TA
- Midterm grading in progress → next week
- Course project proposal comments

- Mid-Semester feedback (next slide)

MID-SEMESTER FEEDBACK

"Reading papers before lecture on that paper. I struggle to understand and in the class I get some knowledge"

"Learning from papers/understanding them is hard for someone like me who hasn't read a lot of papers"

→ Piazza post / Review form section numbers that are more important

"The in-class discussion times are very short"

frame review qns to emphasize that

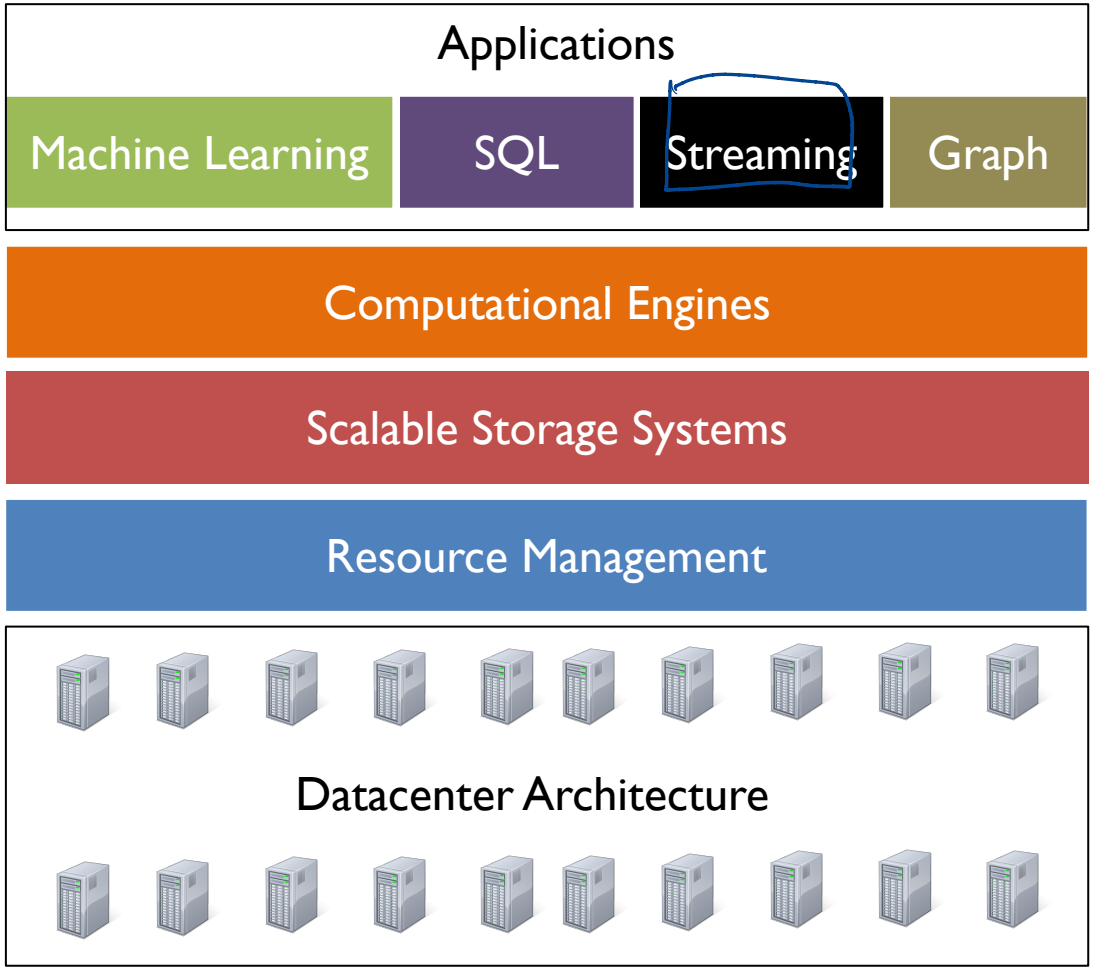
"Around 80-85 mins for the lecture and 20 mins for the discussion."

"More discussion on applying ideas from paper to different settings"

Exams

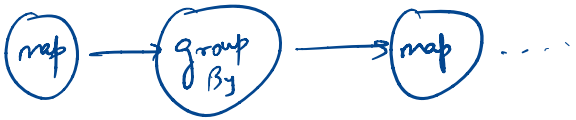
"not enough time to even think for each question, it felt very congested"

"Smaller (perhaps bi-weekly) quizzes, take home exams with a partner, … solo, etc."


Applications
Machine Learning
SQL
Streaming
Graph
Computational Engines
Scalable Storage Systems
Resource Management
Datacenter Architecture

Spark

dataflow

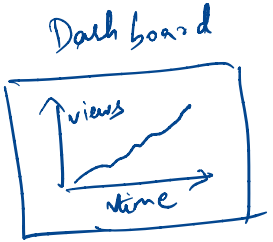


Pytorch

DATAFLOW MODEL (?)

What is a streaming workload and desirable properties?

MOTIVATION



Streaming Video Provider

- How much to bill each advertiser ?
- Need per-user, per-video viewing sessions
- Handle out of order data

- how long did each user watch
- which ads were shown to which user

Goals

- Easy to program → API design
- Balance correctness, latency and cost

↳ inputs that are used for a given output

Data is streaming
- Unbounded data

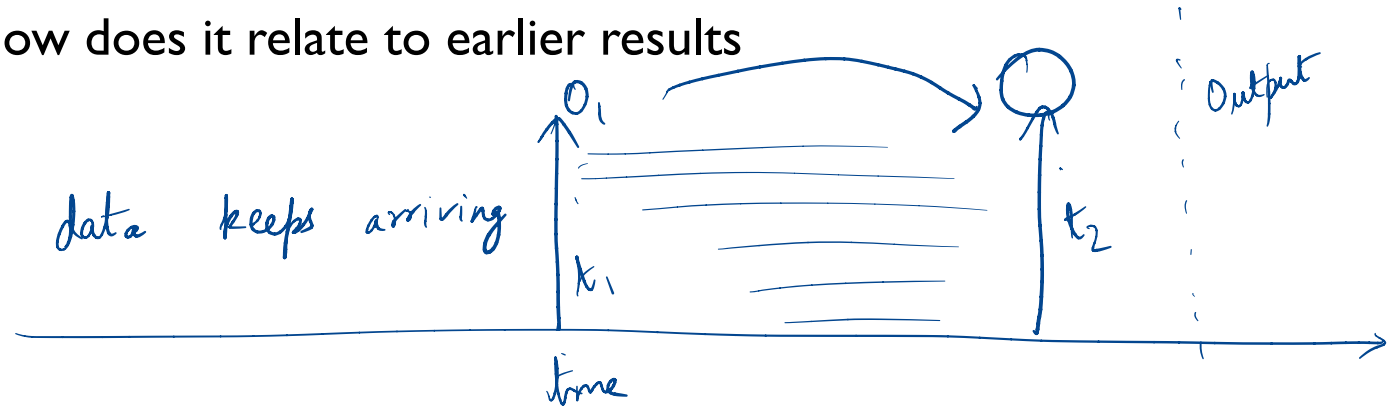
Out of order data arrival

APPROACH

Separate user API from execution

Decompose queries into

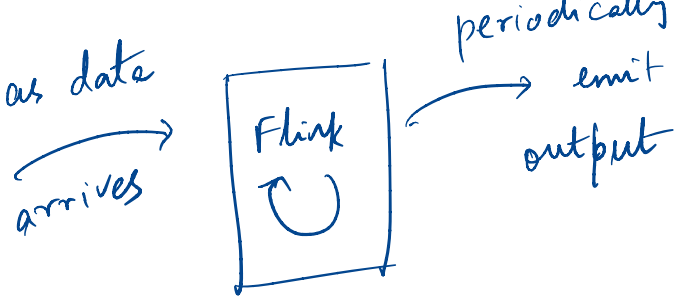
- What is being computed → SQL query or MapReduce / Spark job
- Where in time is it computed
- When is it materialized – Output
- How does it relate to earlier results



STREAMING VS. BATCH

Streaming

- Event-based system

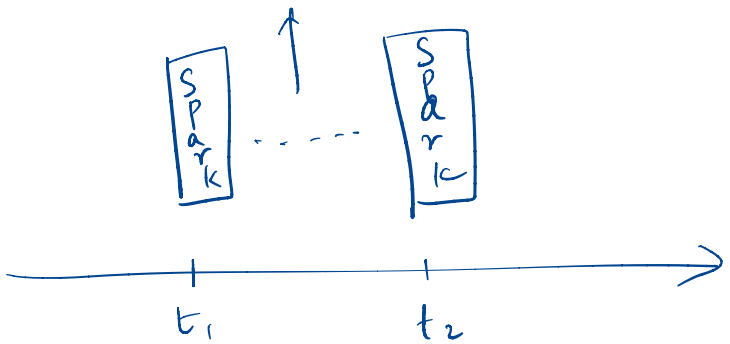


Naiad

Batch

Spark Streaming

- Wait for time t, accumulate data that arrives
- Trigger batch computation → Spark accumulate data



TIMESTAMPS

Event time:

Time at which an event took place

e.g. time at which video was watched

Processing time:

Time at which an event was processed by the system

e.g: time at which video watch event is processed by system

Proc time ≠ Event time

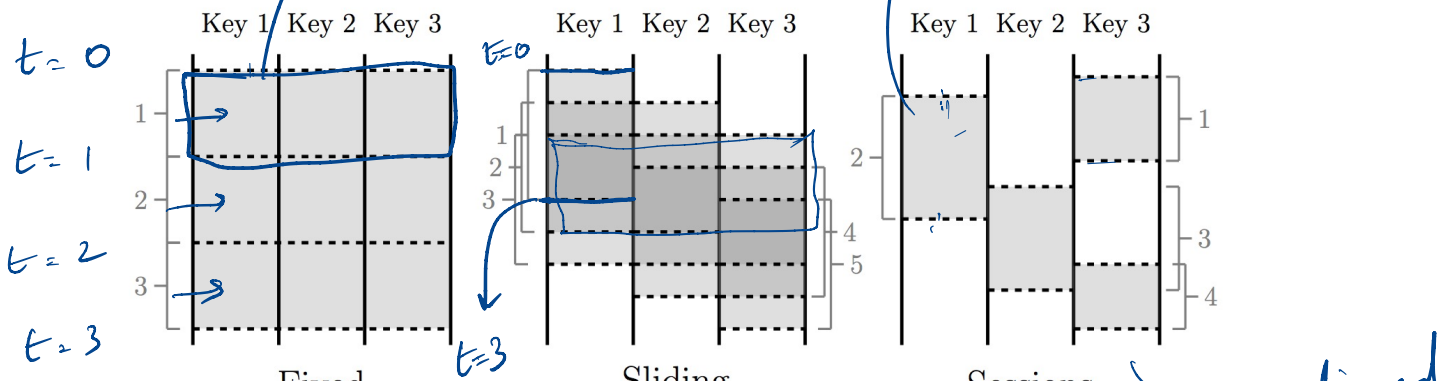
WINDOWING

Key → video_id or user_id

aligned windows

all events for a user within 5 min range = session

window/bucket every 1s



unaligned

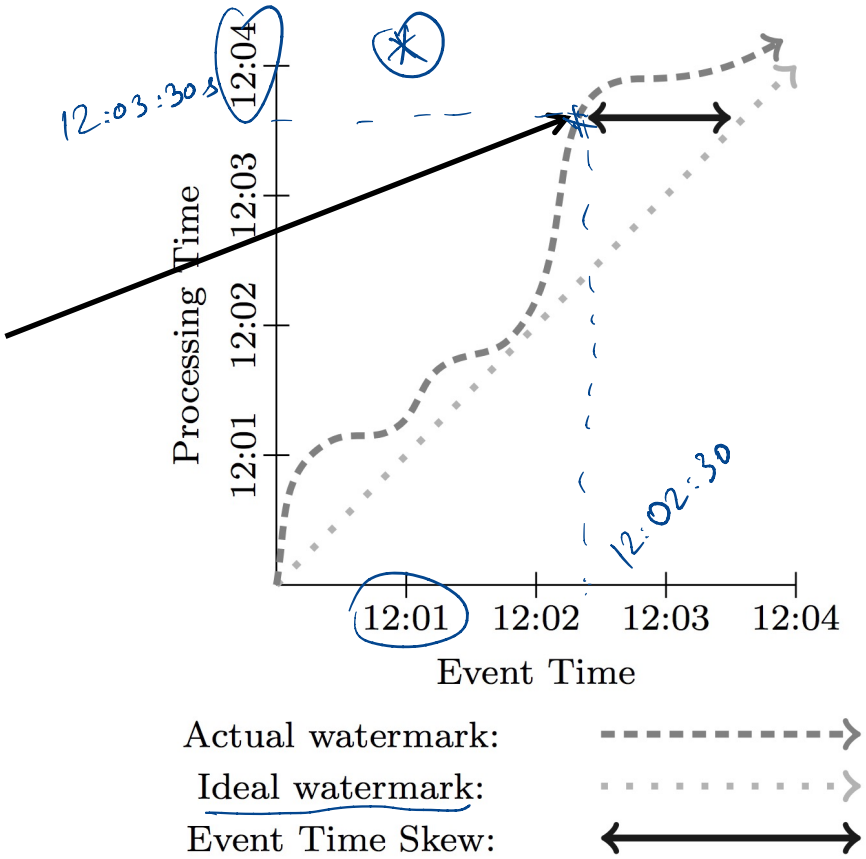
t = 0 to 3
t = 1 to 4
t = 2 to 5 ....

WATERMARK OR SKEW

Watermark

↓

System has processed all events up to 12:02:30



- Watermark varies over time
- Hard to know!

processing time = event time

API

ParDo: → Map in MapReduce / Spark

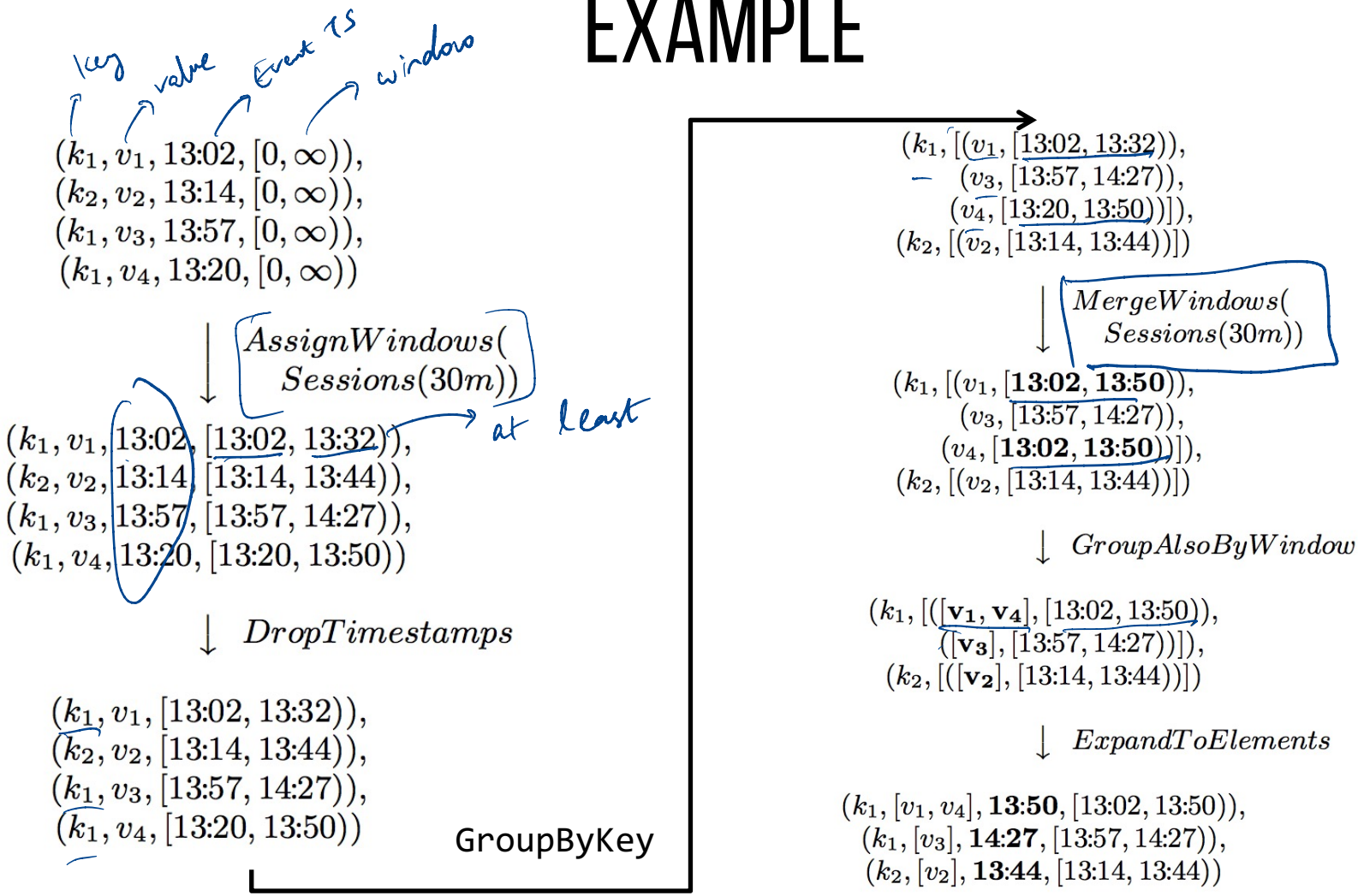
GroupByKey: → Very similar in MR / Spark / SCOPE

Windowing

AssignWindow → Takes a key and puts it in a window

MergeWindow → merge adjacent windows (Session semantics)

EXAMPLE


key
value
Event TS
window
$(k_1, v_1, 13{:}02, [0, \infty))$,
$(k_2, v_2, 13{:}14, [0, \infty))$,
$(k_1, v_3, 13{:}57, [0, \infty))$,
$(k_1, v_4, 13{:}20, [0, \infty))$
$AssignWindows(Sessions(30m))$
at least
$(k_1, v_1, 13{:}02, [13{:}02, 13{:}32))$,
$(k_2, v_2, 13{:}14, [13{:}14, 13{:}44))$,
$(k_1, v_3, 13{:}57, [13{:}57, 14{:}27))$,
$(k_1, v_4, 13{:}20, [13{:}20, 13{:}50))$
$DropTimestamps$
$(k_1, v_1, [13{:}02, 13{:}32))$,
$(k_2, v_2, [13{:}14, 13{:}44))$,
$(k_1, v_3, [13{:}57, 14{:}27))$,
$(k_1, v_4, [13{:}20, 13{:}50))$
GroupByKey
$(k_1, [(v_1, [13{:}02, 13{:}32)),$
$(v_3, [13{:}57, 14{:}27)),$
$(v_4, [13{:}20, 13{:}50))]),$
$(k_2, [(v_2, [13{:}14, 13{:}44))])$
$MergeWindows(Sessions(30m))$
$(k_1, [(v_1, [\mathbf{13{:}02}, \mathbf{13{:}50})),$
$(v_3, [13{:}57, 14{:}27)),$
$(v_4, [\mathbf{13{:}02}, \mathbf{13{:}50}))]),$
$(k_2, [(v_2, [13{:}14, 13{:}44))])$
$GroupAlsoByWindow$
$(k_1, [([\mathbf{v_1}, \mathbf{v_4}], [13{:}02, 13{:}50)),$
$([\mathbf{v_3}], [13{:}57, 14{:}27))]),$
$(k_2, [([\mathbf{v_2}], [13{:}14, 13{:}44))])$
$ExpandToElements$
$(k_1, [v_1, v_4], \mathbf{13{:}50}, [13{:}02, 13{:}50))$,
$(k_1, [v_3], \mathbf{14{:}27}, [13{:}57, 14{:}27))$,
$(k_2, [v_2], \mathbf{13{:}44}, [13{:}14, 13{:}44))$

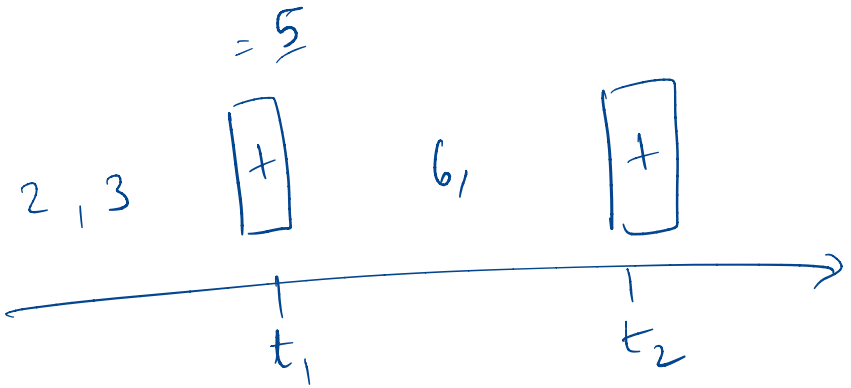
TRIGGERS AND INCREMENTAL PROCESSING

Windowing: where in event time are data grouped

Triggering: when in processing time are groups emitted

Strategies

- Discarding = 6 (discard 5)
- Accumulating = 11 (5+6)
- Accumulating & Retracting



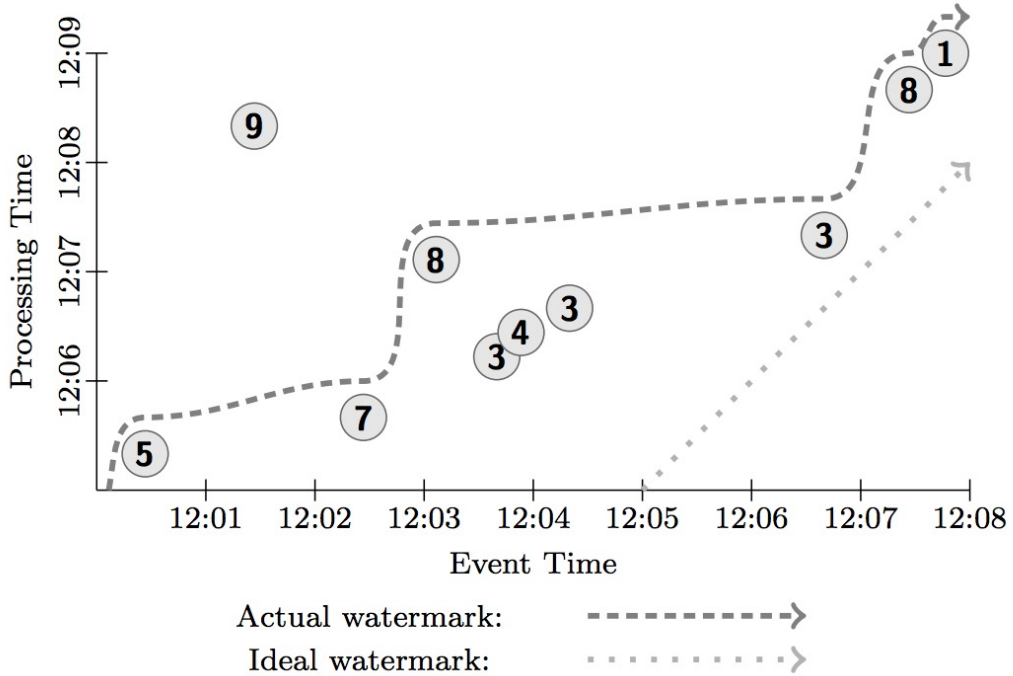
−5 , 11

↓ retract ↓ accumulate

RUNNING EXAMPLE

```
PCollection<KV<String, Integer>> input = IO.read(...);
PCollection<KV<String, Integer>> output =
        input.apply(Sum.integersPerKey());
```

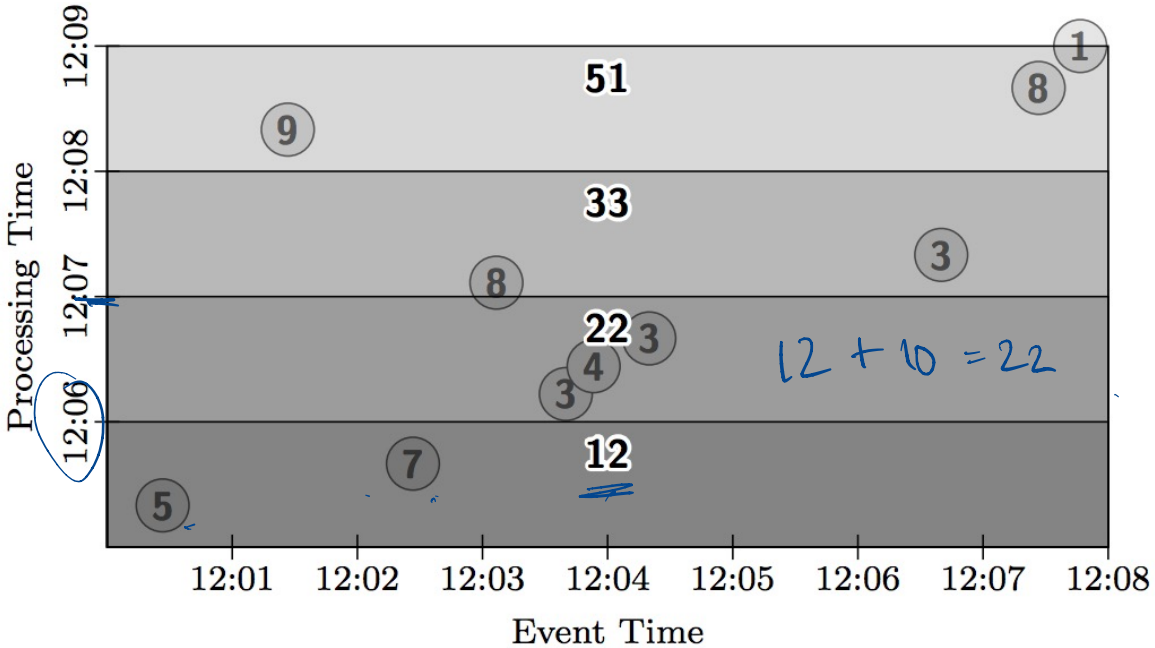
sum of values for one key



GLOBAL WINDOWS, ACCUMULATE

```
PCollection<KV<String, Integer>> output = input
    .apply(Window.trigger(Repeat(AtPeriod(1, MINUTE)))
                  .accumulating())
    .apply(Sum.integersPerKey());
```

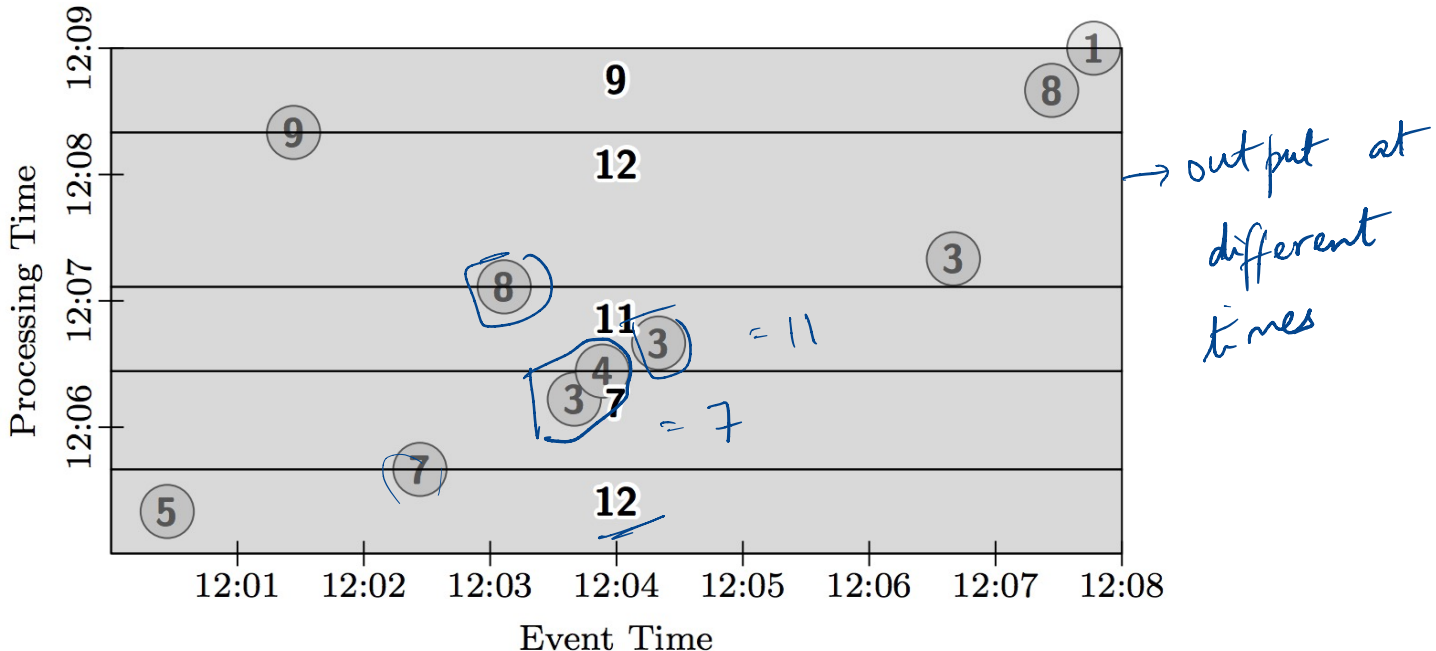
one window or one running sum



GLOBAL WINDOWS, COUNT, DISCARDING

```
PCollection<KV<String, Integer>> output = input
    .apply(Window.trigger(Repeat(AtCount(2)))
                  .discarding())
    .apply(Sum.integersPerKey());
```

trigger based on data items



output at different times

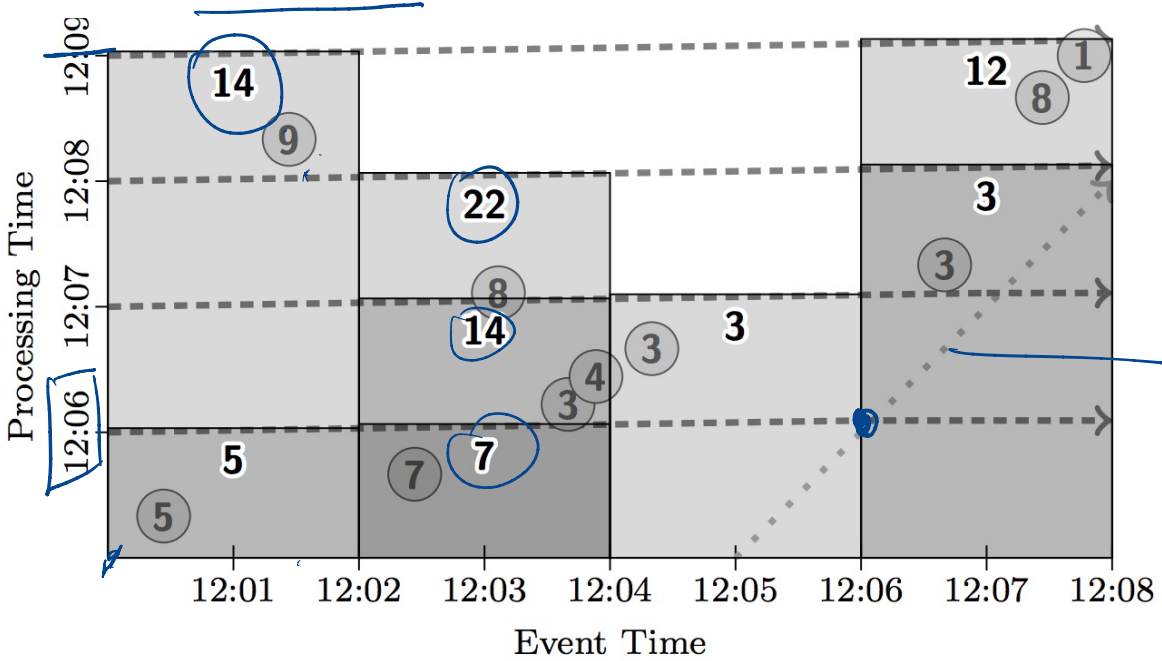
FIXED WINDOWS, MICRO BATCH

```
PCollection<KV<String, Integer>> output = input
    .apply(Window.into(FixedWindows.of(2, MINUTES))
                .trigger(Repeat(AtWatermark()))
                .accumulating())
```

event time

trigger at watermark

Batch System runs every 1 minute



ideal watermark

SUMMARY/LESSONS

Design for unbounded data: Don't rely on completeness

Be flexible, diverse use cases

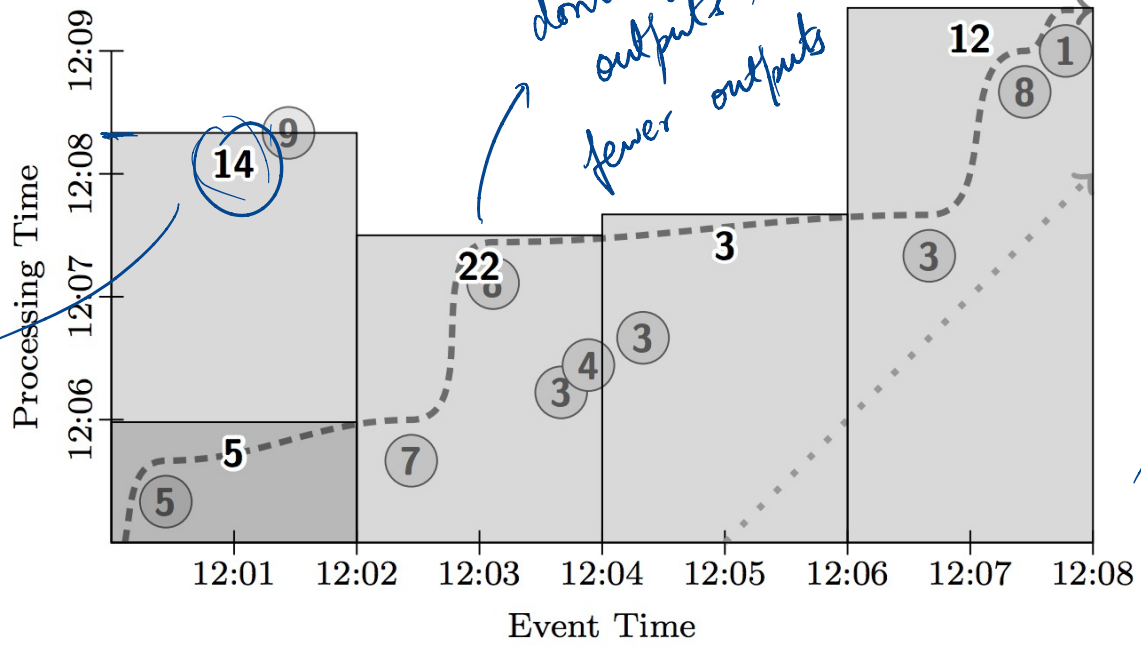
- Billing
- Recommendation
- Anomaly detection

Windowing, Trigger API to simplify programming on unbounded data

DISCUSSION

https://forms.gle/Yuvk4SfFoHyy4Et36

Outputs for windows are mostly produced once except for (9)



don't get partial outputs, but fewer outputs

microbatch - latency could be higher

latency of results

Streaming: (9) is handled as soon as it arrives

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?

NEXT STEPS

Next class: Naiad

Course project proposal feedback