- Assignment 2 grades are up!
- Midterm grading in progress
- 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”

“The in-class discussion times are very short”

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

“More discussion on applying ideas from paper to different settings”

“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.”
DATAFLOW MODEL (?)
Streaming Video Provider
- How much to bill each advertiser?
- Need per-user, per-video viewing sessions
- Handle out of order data

Goals
- Easy to program
- Balance correctness, latency and cost
APPROACH

Separate user API from execution

Decompose queries into
- What is being computed
- Where in time is it computed
- When is it materialized
- How does it relate to earlier results
STREAMING VS. BATCH

Streaming

Batch
TIMESTAMPS

Event time:

Processing time:
WINDOWING

Fixed

Sliding

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

ParDo:

GroupByKey:

Windowing
  AssignWindow

MergeWindow
EXAMPLE

\[
\begin{align*}
(k_1, v_1, 13:02, [0, \infty)) , \\
(k_2, v_2, 13:14, [0, \infty)) , \\
(k_1, v_3, 13:57, [0, \infty)) , \\
(k_1, v_4, 13:20, [0, \infty)) ,
\end{align*}
\]

\[
\downarrow \text{AssignWindows(} \quad \text{Sessions(30m)} \quad \uparrow
\]

\[
\begin{align*}
(k_1, v_1, 13:02, [13:02, 13:32)) , \\
(k_2, v_2, 13:14, [13:14, 13:44)) , \\
(k_1, v_3, 13:57, [13:57, 14:27)) , \\
(k_1, v_4, 13:20, [13:20, 13:50)) ,
\end{align*}
\]

\[
\downarrow \text{DropTimestamps}
\]

\[
\begin{align*}
(k_1, v_1, [13:02, 13:32)) , \\
(k_2, v_2, [13:14, 13:44)) , \\
(k_1, v_3, [13:57, 14:27)) , \\
(k_1, v_4, [13:20, 13:50))
\end{align*}
\]

\[
\downarrow \text{GroupByKey}
\]

\[
\begin{align*}
(k_1, [(v_1, [13:02, 13:32)), \\
(v_3, [13:57, 14:27)), \\
(v_4, [13:20, 13:50])) , \\
(k_2, [(v_2, [13:14, 13:44]))
\end{align*}
\]

\[
\downarrow \text{MergeWindows(} \quad \text{Sessions(30m)} \quad \uparrow
\]

\[
\begin{align*}
(k_1, [(v_1, \textbf{13:02, 13:50})], \\
(v_3, [13:57, 14:27]), \\
(v_4, \textbf{13:02, 13:50})), \\
(k_2, [(v_2, [13:14, 13:44]))
\end{align*}
\]

\[
\downarrow \text{GroupAlsoByWindow}
\]

\[
\begin{align*}
(k_1, [[v_1, v_4], [13:02, 13:50)), \\
([v_3], [13:57, 14:27])], \\
(k_2, [[v_2], [13:14, 13:44]])
\end{align*}
\]

\[
\downarrow \text{ExpandToElements}
\]

\[
\begin{align*}
(k_1, [v_1, v_4], \textbf{13:50}, [13:02, 13:50)), \\
(k_1, [v_3], \textbf{14:27}, [13:57, 14:27]), \\
(k_2, [v_2], \textbf{13:44}, [13:14, 13:44])
\end{align*}
\]
TRIGGERS AND INCREMENTAL PROCESSING

Windowing: where in event time are data grouped
Triggering: when in processing time are groups emitted

Strategies
- Discarding
- Accumulating
- Accumulating & Retracting
RUNNING EXAMPLE

PCollection<KV<String, Integer>> input = IO.read(...);
PCollection<KV<String, Integer>> output =
    input.apply(Sum.integersPerKey());
PCollection<KV<String, Integer>> output = input
    .apply(Window.trigger(Repeat(AtPeriod(1, MINUTE)))
          .accumulating())
    .apply(Sum.integersPerKey());
PCollection<KV<String, Integer>> output = input
    .apply(Window.trigger(Repeat(AtCount(2)))
        .discarding())
    .apply(Sum.integersPerKey());
PCollection<KV<String, Integer>> output = input
    .apply(Window.into(FixedWindows.of(2, MINUTES)))
    .trigger(Repeat(AtWatermark()))
    .accumulating()
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
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