University of California, Berkeley
Master of Information and Data Science (MIDS)
W205 – Fundamentals of Data Engineering

Week 13 – Enterprise Message Queues, Data Lakes, and Serverless SQL

Agenda for Today’s Class

- Attendance and Participation
- Announcements
- Schedule and Due Dates
- Work / Life / School Balance
- Asynch High Level Review in a Nutshell
- Breakouts
- Summary

Attendance and Participation

Please record your attendance and participation for today's class:

GitHub => ucb_mids_w205_repo => README.md => Attendance and Participation

Announcements

- Upcoming holidays and/or breaks
- Makeup classes for holidays
- Upcoming events
- Student evaluations
- Etc.

Schedule and Due Dates

Take a quick look at the next couple of weeks’ due dates:

GitHub => ucb_mids_w205_repo => README.md => Schedule and Due Dates

Work / Life / School Balance
Open Discussion

Student feedback

- About 5 minutes
- How are things going related to work / life / school balance?
- How is w205 going? Difficulty? Time?
- Impact of any natural and/or man-made disasters
- Etc.

Asynch High Level Review in a Nutshell

Each week we will spend about 15 minutes reviewing the most important high level concepts from the asynch

Getting Data from Production Silos

- Silo / Stovepipe
  - Production system with little or no outside connectivity
  - Analogy to grain silos or wood or coal burning stovepipe
- Goal
  - Get fresh data from production silo
  - Any system that needs data can get it
  - Scales up

Enterprise Message Queues

- Allow production silos to send data outside with minimal impact
- 3 Models
  - Publisher-subscriber model
  - Producer-consumer model
  - Streaming model
- Topic
  - Create a topic and send messages to the topic
  - Typically use JSON
  - Big message may use JSON with pointer to Object Store

Publisher-Subscriber Model

- Publish once to a topic (or topics)
- Enterprise message queue system buffers, stores, and distributes the message for us
- Subscribers subscribe to a topic and read the messages
  - Unlimited number of subscribers
  - Each subscriber keeps up with their last message ID read
- Messages deleted when retention period passes
- Good side effect is we can often pull data from topics that we cannot get elsewhere

Producer-Consumer Model

- Supports queueing systems
- Producers
  - One or more processes that write messages to a topic
- Consumers
  - One or more processes that read and delete messages from the topic
  - Ensures that no message is read more than once
- Scale up
  - Add more producers and/or consumers
- Queuing systems existed long before computers (folders and boxes)

Streaming Data

- Typically comes at a lightning-fast pace
- Do not read a message more than once
- Do not lose any messages
- Multiple processes to read the streaming data

Streaming Model

- Turns out enterprise message queues are a reasonable fit for the streaming model
  - Use the publisher-subscriber model
    - Good for a single subscriber
    - Would not scale out as we would read messages more than once
  - Use the producer-consumer model
    - Allows multiple consumers
    - Would scale out
    - Not an exact fit
  - Software designed specifically for streaming now available

Breakouts

GitHub => ucb_mids_w205_repo => breakouts

(time permitting, we may not get to all of them)

Summary

Instructor will give a brief (about 2 minute) summary of today’s class.