15-319 / 15-619 Cloud Computing

Recitation 14

April 19th 2016

Overview

- **Recent Tasks reflection**
  - Project 4.2
- **Budget issues**
  - Tagging
- **This week's schedule**
  - Unit 5 - Modules 21 & 22
  - Quiz 12
  - Project 4.3

Reminders

- Monitor AWS expenses regularly and tag all resources
  - Check your bill (Cost Explorer > filter by tags).
- Piazza Guidelines
  - Please tag your questions appropriately
  - Search for an existing answer first
- Provide clean, modular and well documented code
  - Large penalties for not doing so.
  - **Double check** that your code is submitted!! (verify by downloading it from TPZ from the submissions page)
- Utilize Office Hours
  - We are here to help (but not to give solutions)

Become a TA

- Why?
  - Because it's Awesome.
  - Learn more from other TAs

- How?
  - Step 1: Do well on the projects/quizzes/forum
  - Step 2: Fill the application form very thoughtfully
  - Step 3: Ace the interview

- When?
  - By Saturday 4/23/2016

* terms and conditions apply

Module to Read

- UNIT 5: Distributed Programming and Analytics Engines for the Cloud
  - Module 18: Introduction to Distributed Programming for the Cloud
  - Module 19: Distributed Analytics Engines for the Cloud: MapReduce
  - Module 20: Distributed Analytics Engines for the Cloud: Spark
  - Module 21: Distributed Analytics Engines for the Cloud: GraphLab
  - Module 22: Message Queues and Stream Processing: Kafka and Samza

Project 4.2 FAQ

- How to calculate the contributions in PageRank? How to deal with dangling nodes?
  - Refer to the formula in the writeup. A node receives contributions from its followers, not followee. Distribute the rank of dangling nodes equally to all nodes.
- My Spark job ran for hours and finished after the deadline
  - Sorry we don’t accept late submissions.
  - Processing big data can be very time-consuming, especially without optimization.
  - Start early!
- Memory related exceptions
  - RDD materialization, Spark configuration.

Project 4

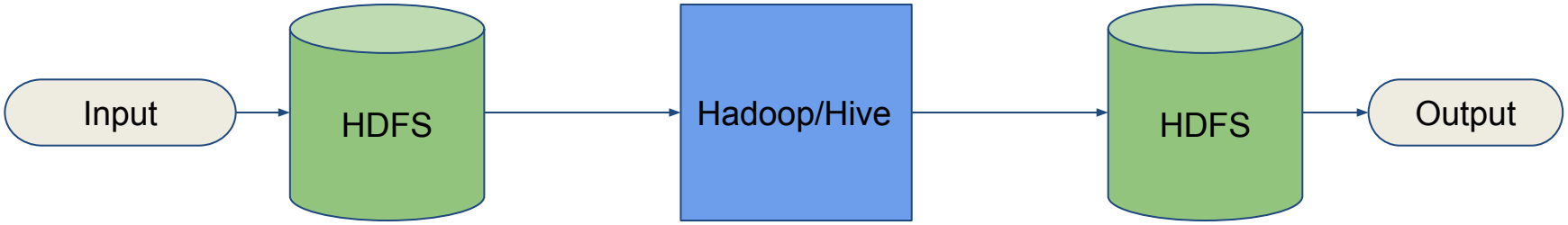
- Project 4.1
  - MapReduce Programming Using YARN
- Project 4.2
  - Iterative Programming Using Apache Spark
- Project 4.3
  - Stream Processing using Kafka & Samza

Stream vs Batch Processing

- Batch processing
  - Data parallel, graph parallel
  - Iterative, non-iterative
  - Runs once in few hours/days
  - Historical data analysis
  - Unsuited for real time events streams
- Stream processing
  - Streams are an infinite sequence of messages
  - Process events as they come
  - Real time decision making
  - Sensor streams/ web event data

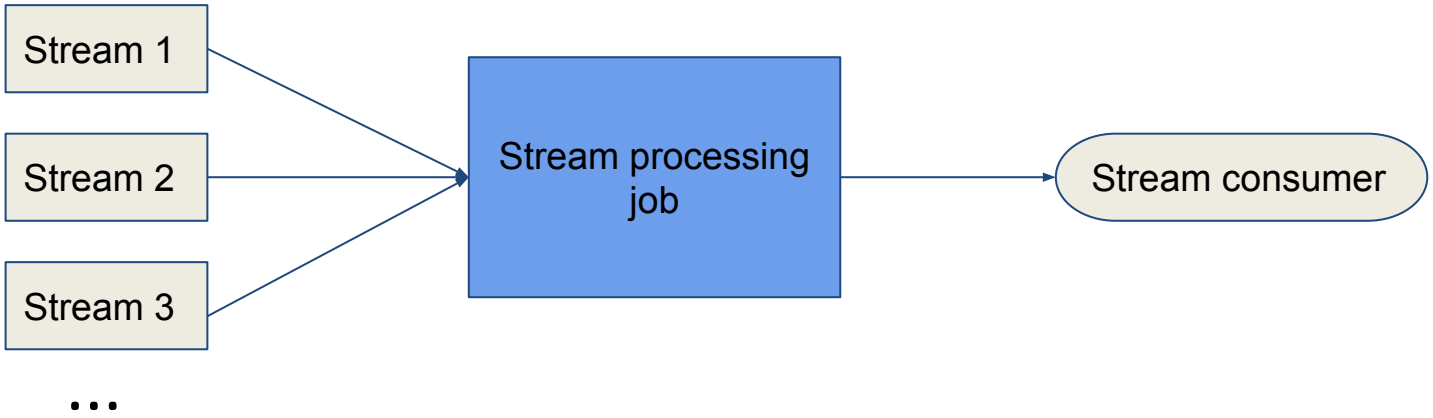
Example of a batch processing job

- Input is collected into batches and processing is performed on the input data
- Output is consumed later at any point of time - the data does not lose much of its “value” with time



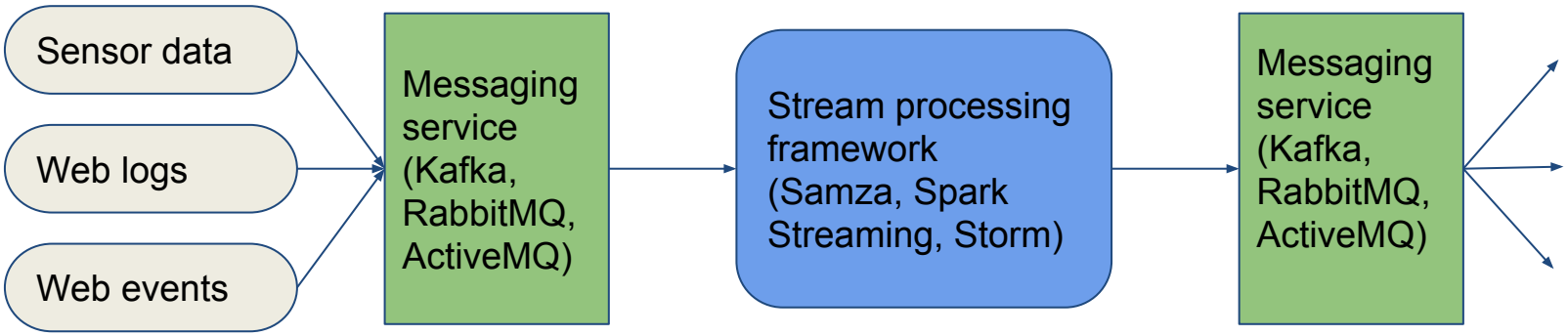
Typical stream processing job

- Data is processed immediately (few seconds)
- The processed data is used by downstream consumers for real time decision/analytics immediately



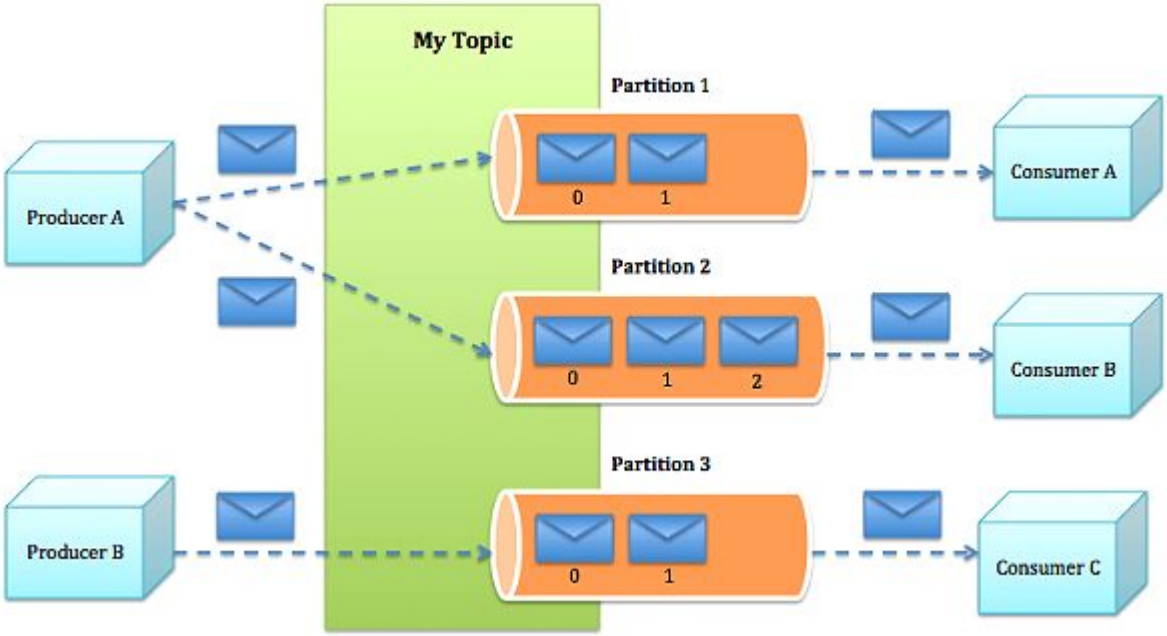
Typical stream processing components

- An event producer - Sensors, web logs, web events
- A messaging service - Kafka, RabbitMQ, ActiveMQ
- A stream processing framework - Samza, Spark Streaming, Storm



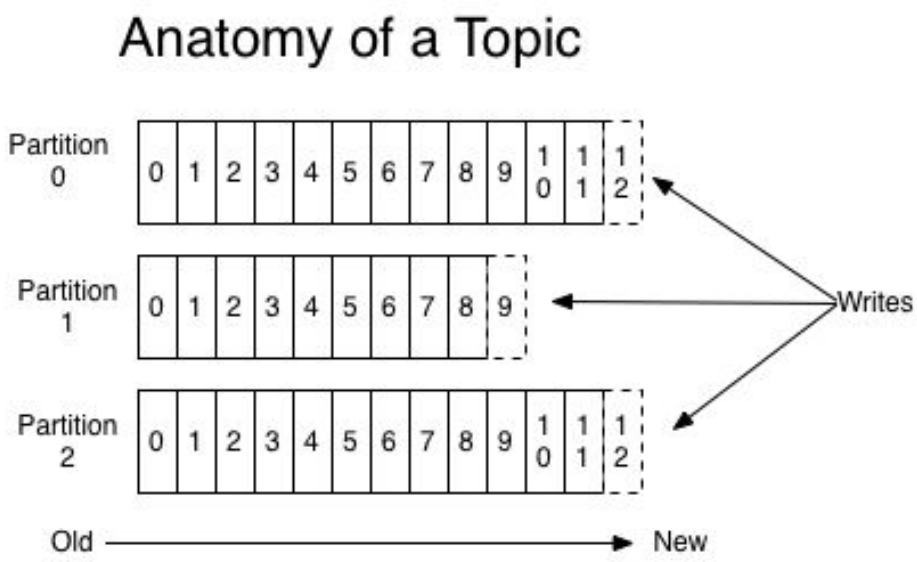
Apache Kafka

- Developed at LinkedIn as a distributed messaging system.



Apache Kafka

- Used to integrate data from multiple sources
- Streams (or topics) in Kafka modelled as a “log”
- Different consumers read independently at different offsets in the log

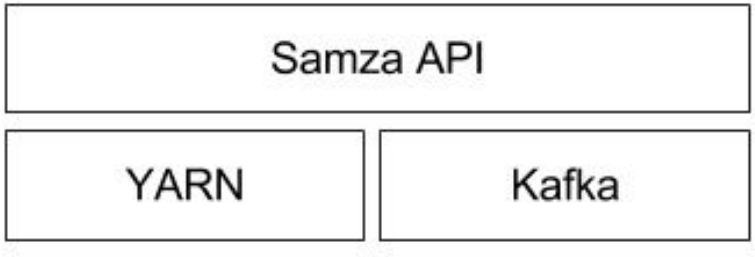


Semantic partitioning in Kafka

- Each topic (stream) is partitioned for scalability across all nodes in the Kafka cluster
- Default partitioning attempts to load balance
- Streams can also be partitioned semantically by user - key of the message
- All messages with the same key are sent to the same partition

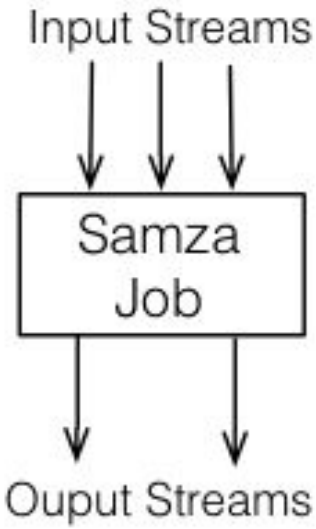
Apache Samza

- Stream processing framework developed at LinkedIn
- The framework consists of 3 layers: streaming, execution and processing (Samza) layer
- Most common with Samza: Kafka for streaming, YARN for execution



Apache Samza

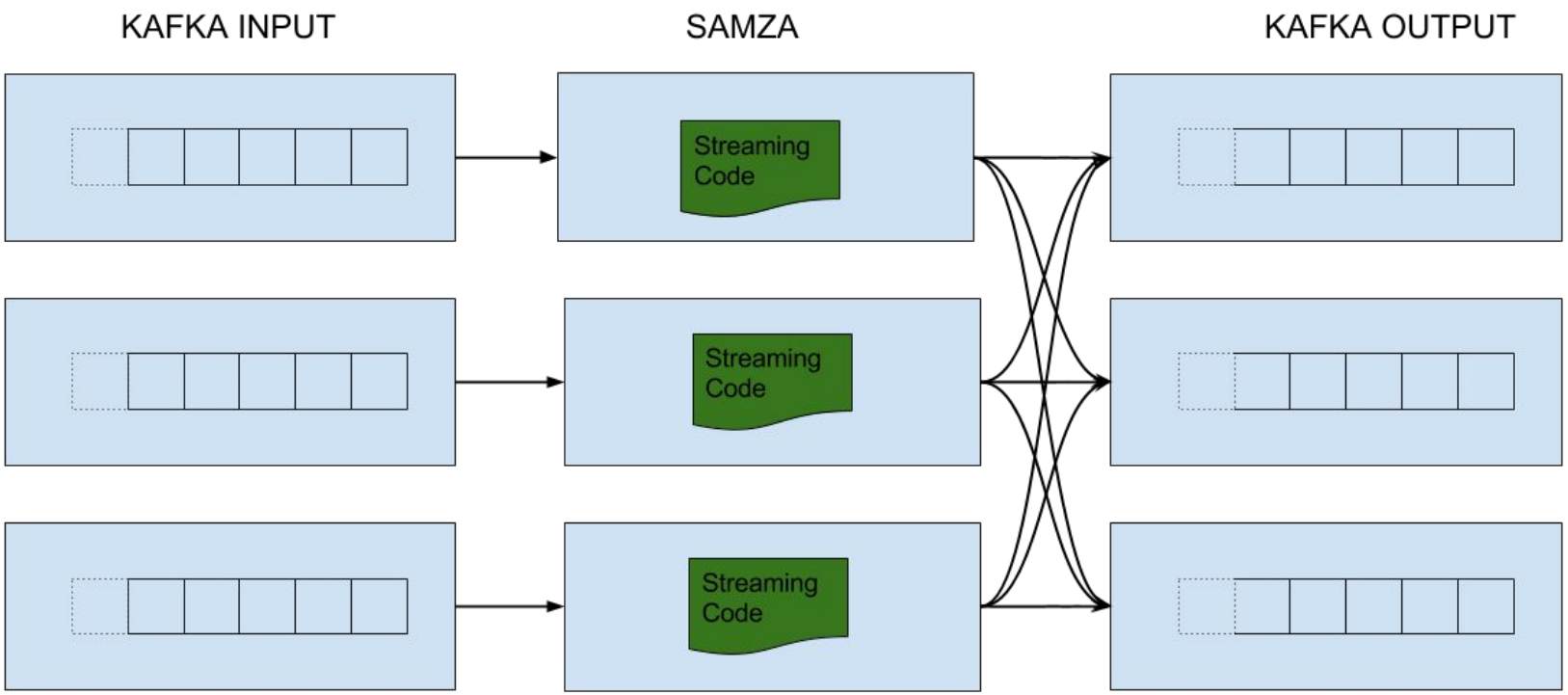
- Programmer uses the Samza API to perform stream processing
- Semantic partitioning in Kafka ⇒ streaming MapReduce
- Each partition in Kafka is assigned to a <u>single</u> Samza task instance



Stateful stream processing in Apache Samza

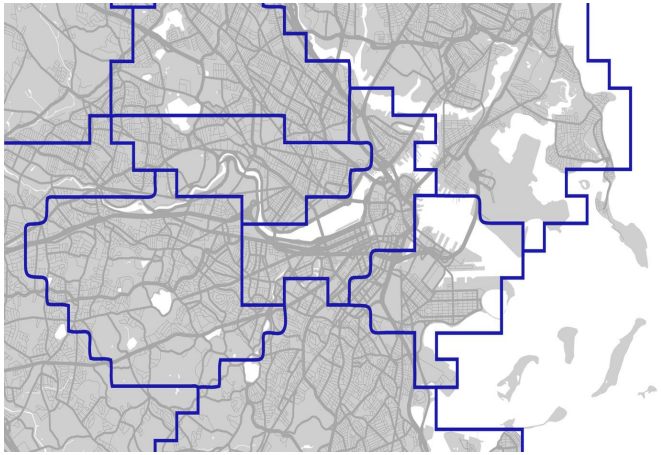
- Calculate sum, avg, count etc.
- State in remote data store? - slow
- State in memory locally? - machine crashes
- Solution - persistent KV store provided by Samza
- Changes to KV store persisted to a different stream (usually Kafka) - replay on failure
- RocksDB currently supported as a persistent KV store
  - You MUST use a persistent KV store for P4.3!

Putting Kafka and Samza together



Project 4.3

- Use Kafka and Samza to develop components of a ride hailing app
- Two Streams
  - Stream of driver locations
  - Stream of ride requests
- Using some algorithm to find a driver to a client



Project 4.3 - Overview

- We provide a load generator to provide Kafka streams
- Use the Samza API to maintain state and find the a driver with the highest match score to a client request
  - <u>driver-locations stream</u> - stream of driver locations as they move through city
  - <u>events stream</u> - stream of events (client requests, ride complete, etc.)
- Algorithm to find a driver
  - Distance
  - Gender preference
  - Driver’s rating
  - Driver’s salary

Project 4.3

- Bonus task - implement dynamic (aka surge) pricing
- Same streams but different state and different calculations required
- Careful when you move drivers around blocks! - bonus grader is more sensitive to sloppy state management
  - For example: ensure that the count of drivers is not off by one
  - ...

Grading

- Skeleton code also provides the submitters

- We will look for the usage of KV stores and reasonably efficient code
  - no iterating through ALL drivers to find closest!

Upcoming Deadlines

- Project 4.3 : Stream Processing with Kafka/Samza
  - Due: 04/24/2016 11:59 PM Pittsburgh
- Apply for F16 of S17 TA job, there is still time
  - link
- Complete the course survey (announced on Piazza)
  - 2% bonus for the overall course grade (Don't miss it!!!)
- Cupcake Party (Pittsburgh and SV)
  - Thursday 04/28/2016 4:30 PM Pittsburgh, 1:30 PM SV

TWITTER DATA ANALYTICS: 15619 PROJECT

15619Project Wrap-Up

- Come to the cupcake party to meet the winners of the 15619Project.
- Thursday 04/28/2016 4:30 PM Pittsburgh, 1:30 PM SV

Don’t Forget!

- Project 4.3 : Stream Processing with Kafka/Samza
  - Due: 04/24/2016 11:59 PM Pittsburgh
- Apply for F16 of S17 TA job, there is still time
  - link
- Complete the course survey (announced on Piazza)
  - 2% bonus for the overall course grade (Don’t miss it!!!)
- Cupcake Party (Pittsburgh and SV)
  - Thursday 04/28/2016 4:30 PM Pittsburgh, 1:30 PM SV