15-319 / 15-619 Cloud Computing

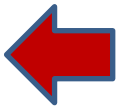
Recitation 15

Overview

- **Last week's reflection**
  - Team Project - Phase 3 - Live Test
- **This week's schedule**
  - Phase 3 report
    - Deadline **TUESDAY** Dec 8, 23:59:00 ET
  - Project 4.3
    - Deadline **FRIDAY** Dec 11, 23:59:59 ET
  - Project 4.3 Reflection Discussion
    - Deadline **SUNDAY** Dec 13, 23:59:59 ET
  - Course survey (2% bonus!)
    - To be announced on Piazza

Project 4

- Project 4.1
  - Iterative Programming Using Apache Spark
- Project 4.2
  - Machine Learning on the Cloud
- Project 4.3
  - Stream Processing using Kafka & Samza



Stream vs Batch Processing

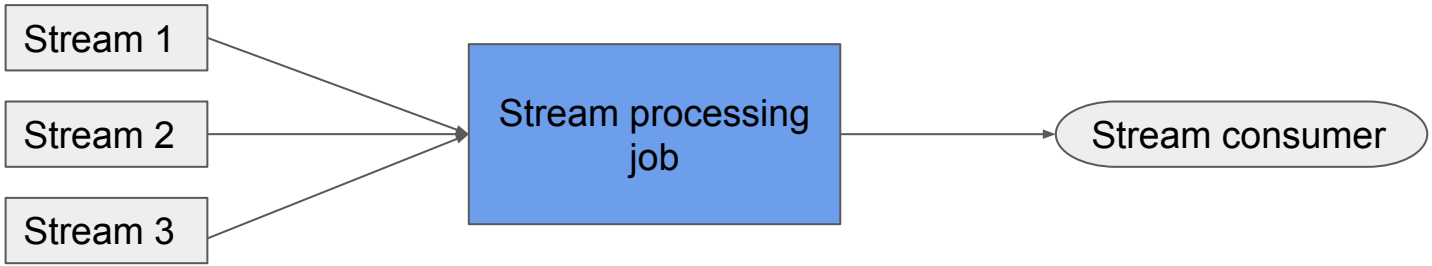
- Batch processing
  - Data parallel, graph parallel
  - Iterative, non-iterative
  - Runs once in few hours/days
  - Historical data analysis
  - Not well suited for real time events streams
- Stream processing
  - Process events as they come
  - Real time decision making
  - Sensor streams/web event data

Typical batch processing job

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

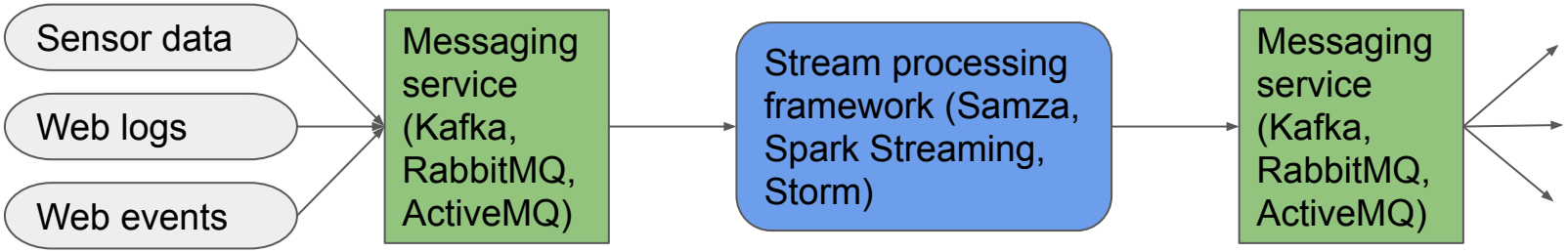
Typical stream processing job

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



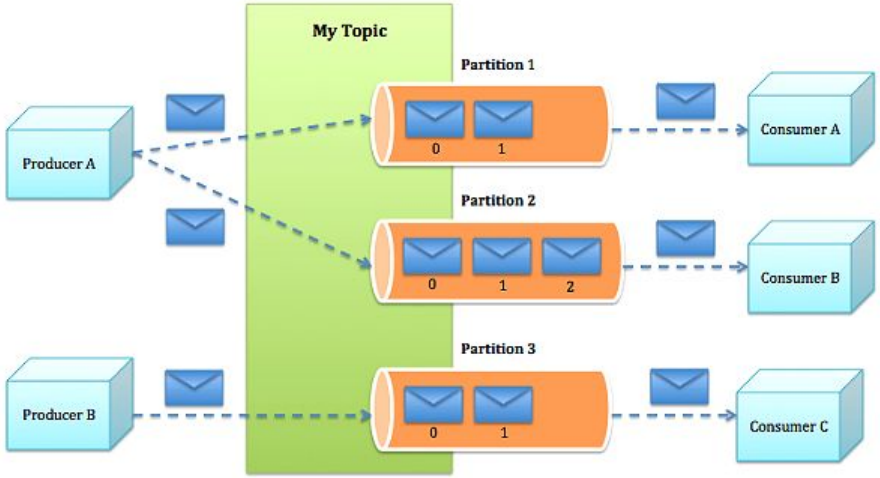
Typical stream processing components

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



Apache Kafka

- A distributed messaging system developed at LinkedIn.

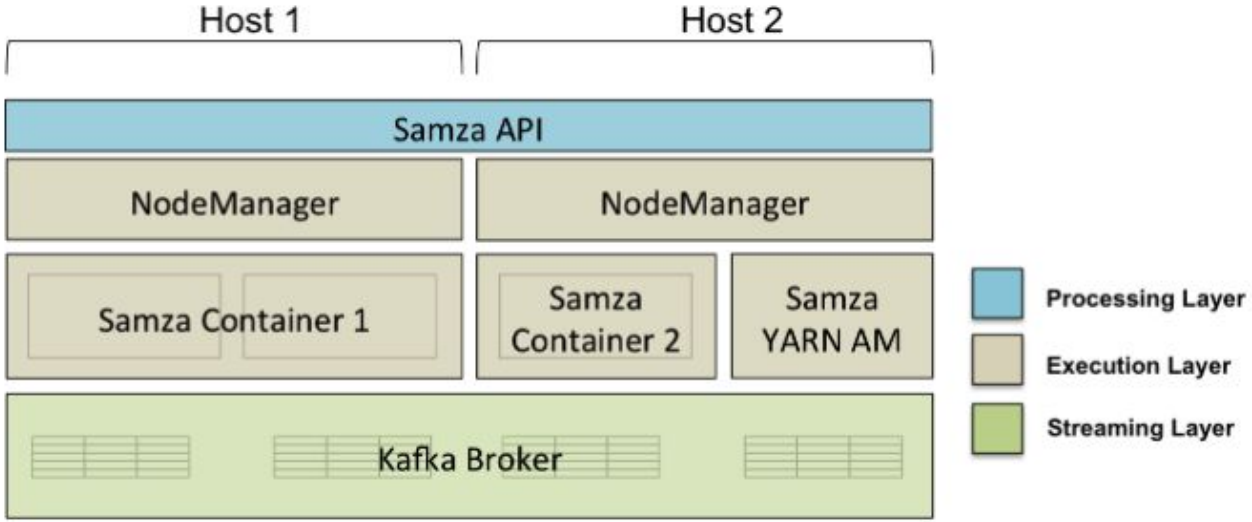


Semantic partitioning in Kafka

- Each topic (stream) is partitioned for scalability across all nodes in the Kafka cluster
- Default partitioning attempts to load balance the messages
- Streams can also be partitioned semantically by user - key of the message
- All messages with the same key arrive to the same partition
- Fault-tolerance: Replication
  - One leader and zero/more followers
  - Replication factor
  - ISR (in-sync replicas)

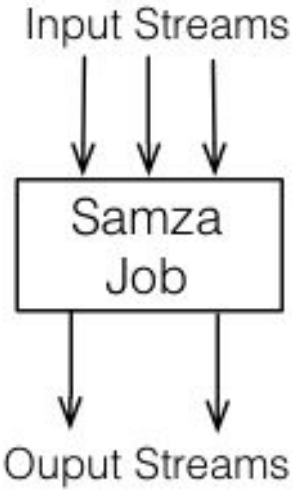
Apache Samza

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



Apache Samza

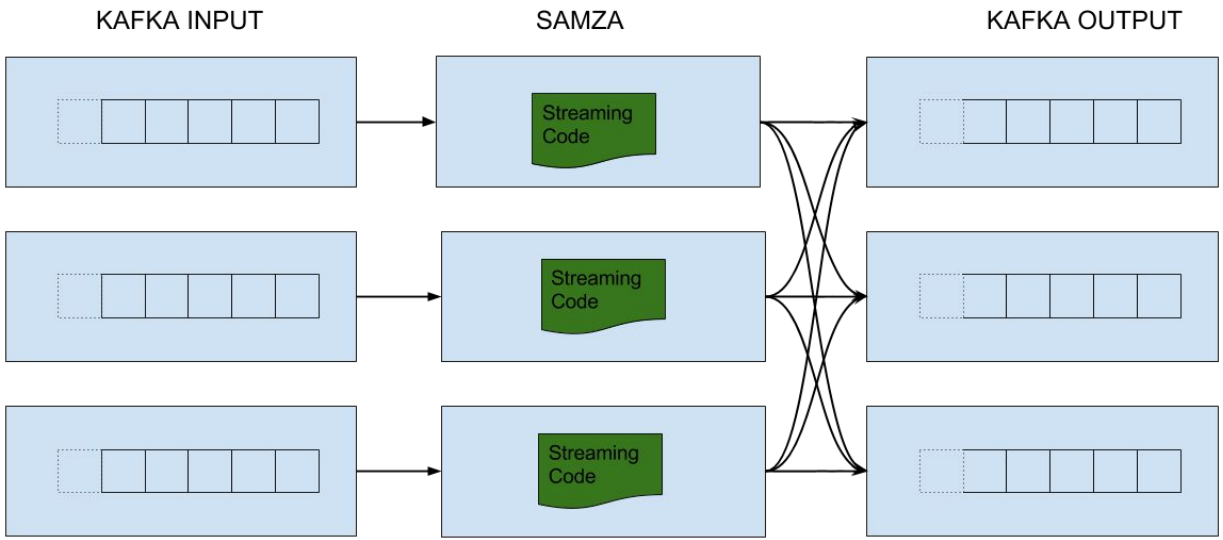
- Programmer uses the Samza API to perform stream processing
- Each partition in Kafka is assigned to a single Samza task instance



Stateful stream processing in Apache Samza

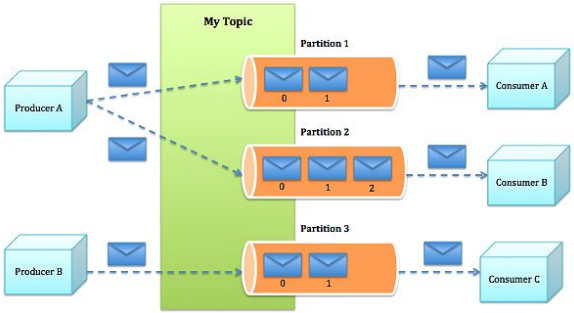
- Calculate sum, avg, count, etc.
- State in remote data store? - slow
- State in local memory? - machine might crash
- Solution - persistent KV store provided by Samza
  - Changes to KV store persisted to a different stream (usually Kafka) - replay on failure
  - RocksDB currently supported as a persistent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

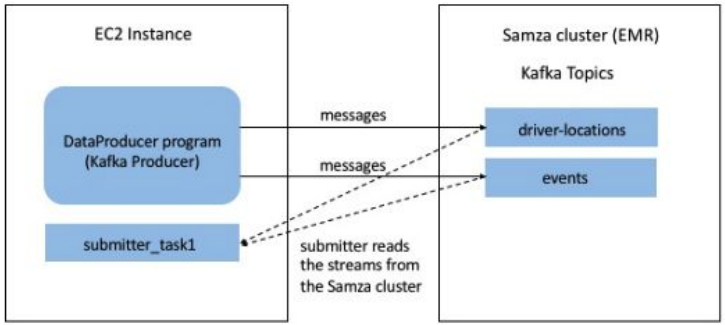
Putting Kafka and Samza Together



P4.3: Uber-like Application for NYC

- Stream Processing with Kafka/Samza
  - Stream 1: Car GPS coordinates
  - Stream 2: Customers
- Task:
  - Match customers with cars to minimize travel time & other constraints



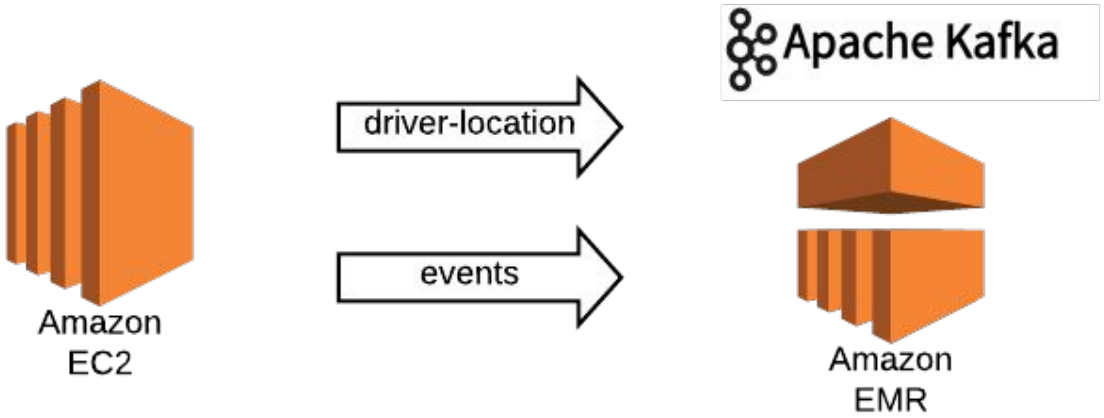


Project 4.3 - Task 1

- Simulate the scenario that the **drivers** update their locations on a regular basis as they move in the city and the **clients** request rides at some time.
  - Data
    - Tracefile -> Two streams
    - Type:
      - DRIVER_LOCATION
        -> driver_locations stream
      - LEAVING_BLOCK, ENTERING_BLOCK, RIDE_REQUEST, RIDE_COMPLETE
        -> events stream

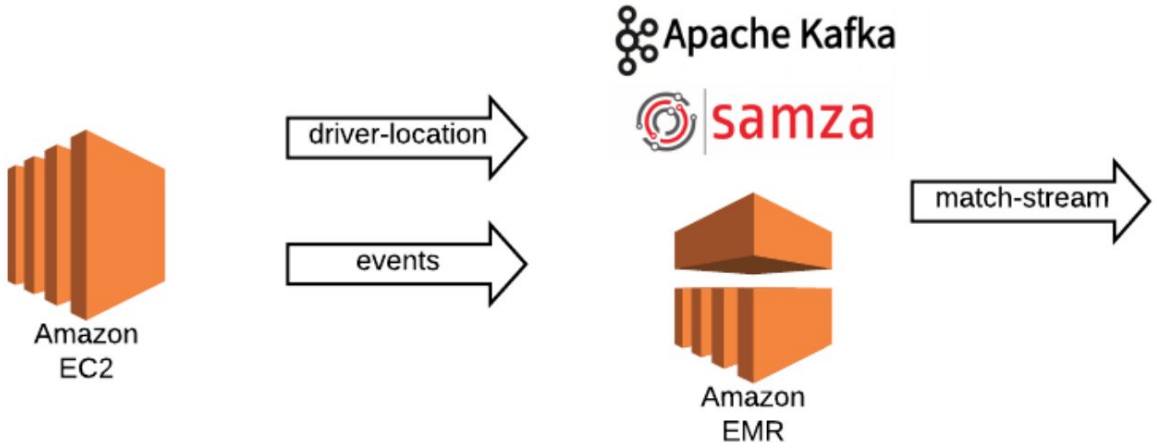
Project 4.3 - Task 1

- You will run your producer program on your student AMI instance.
- The producer program will publish the data into Kafka brokers.
- The submitter for Task 1 is located on the student AMI instance.



Project 4.3 - Task 2

- Use the same producer program used in Task 1.
- You need to find the best match of a ride request with a driver located in the same block as the rider based on published data.

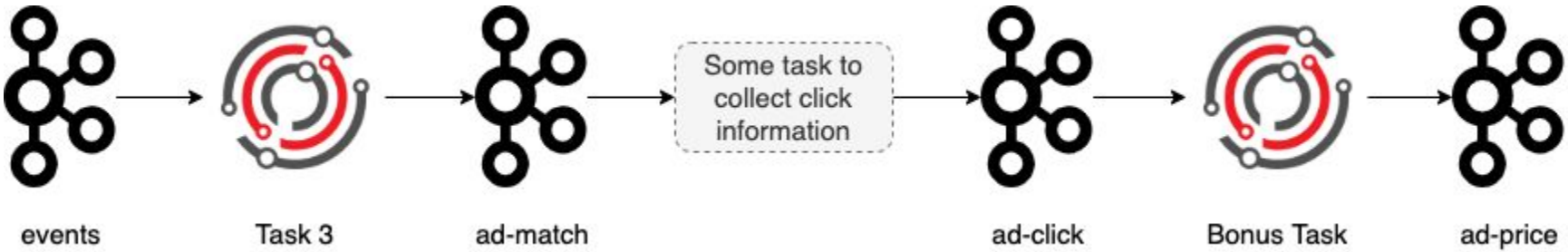


Project 4.3 - Task 3

- You need to find the best advertisement to place for a specific user.
- You need to utilize static data(user profile, health status and interests) and stream data to make this decision.

Project 4.3 - Bonus Task

- Find the advertisement price that a restaurant is paying to NYCabs and the advertisement company.
- Write **at least 2** unit test test cases.



Project 4.3 - Debugging

- Debugging (IMPORTANT!)
  - Use the YARN UI
  - Output a kafka stream for debugging
  - Yarn application commands
    - yarn application -list
  - YARN container logs
    - on the machine where the YARN container is running
- Read the debugging section in the primer carefully!
- Include the error message when you post on Piazza!

TEAM PROJECT
Twitter Data Analytics

Team Project, Overall Winners

- Attend the **Thursday** (12/10) virtual cupcake party recitation
  - To see the winners of the Team Project
  - To listen to the top teams and their implementations
  - Have fun!