Stream Data Analytics over Data Lake Presentation

Submitted By

Rajat Dutt Sharma & Prasoon Dwivedi

Project Guide

Prof. R. Chandrashekhar

Mr. Vivek Yadav

Presentation Overview

Overview

Project objective

Challenges

Work Overview

Architecture

Tools

UI

Technologies Used

Overview

- Enable ingestion of Stream Data on the Data Lake.
- Perform Stream Data Analytics.

Project Goals

- Perform Stream Data Analytics using :
    - Apache Flink
    - Apache Spark

  on the BMTC gprs trace data to calculate average speed of BMTC buses at any point of time.

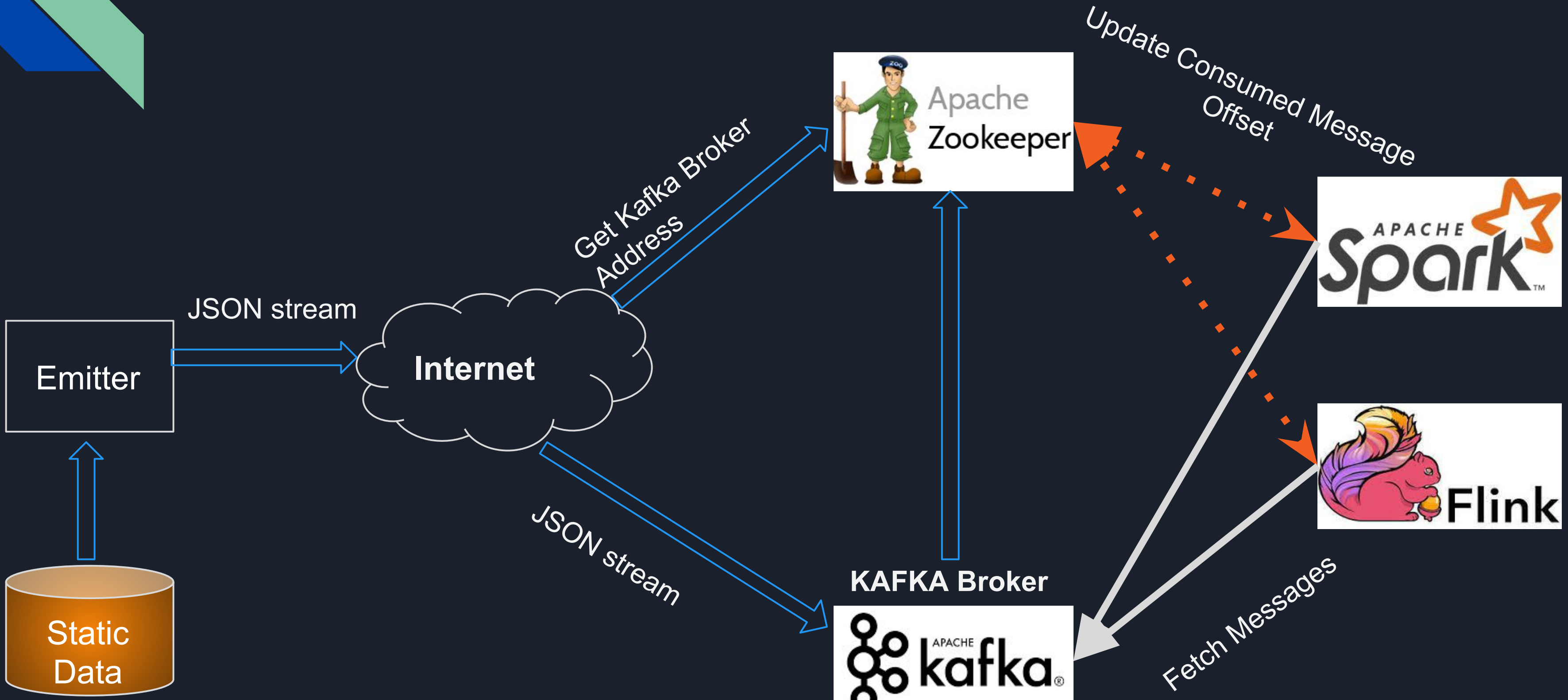
Challenges

- Develop data emitter to generate Stream Data from the static data.
- Select Tools for :
  - Stream Data Ingestion
  - Stream Data Analytics
- Develop Consumers for Stream Data :
  - Spark Consumer
  - Flink Consumer

Work Overview

- Emitter :
  - Streams data at the same rate as the original data stream by mimicking the time difference between consecutive stream records from the timestamp of the records.
- The Data Analytics Flow in the consumers :
  - The stream data is first converted to object stream .
  - The object stream is then mapped to Sliding Keyed Window of size 10 for vehicle specific calculation of Average Speed.

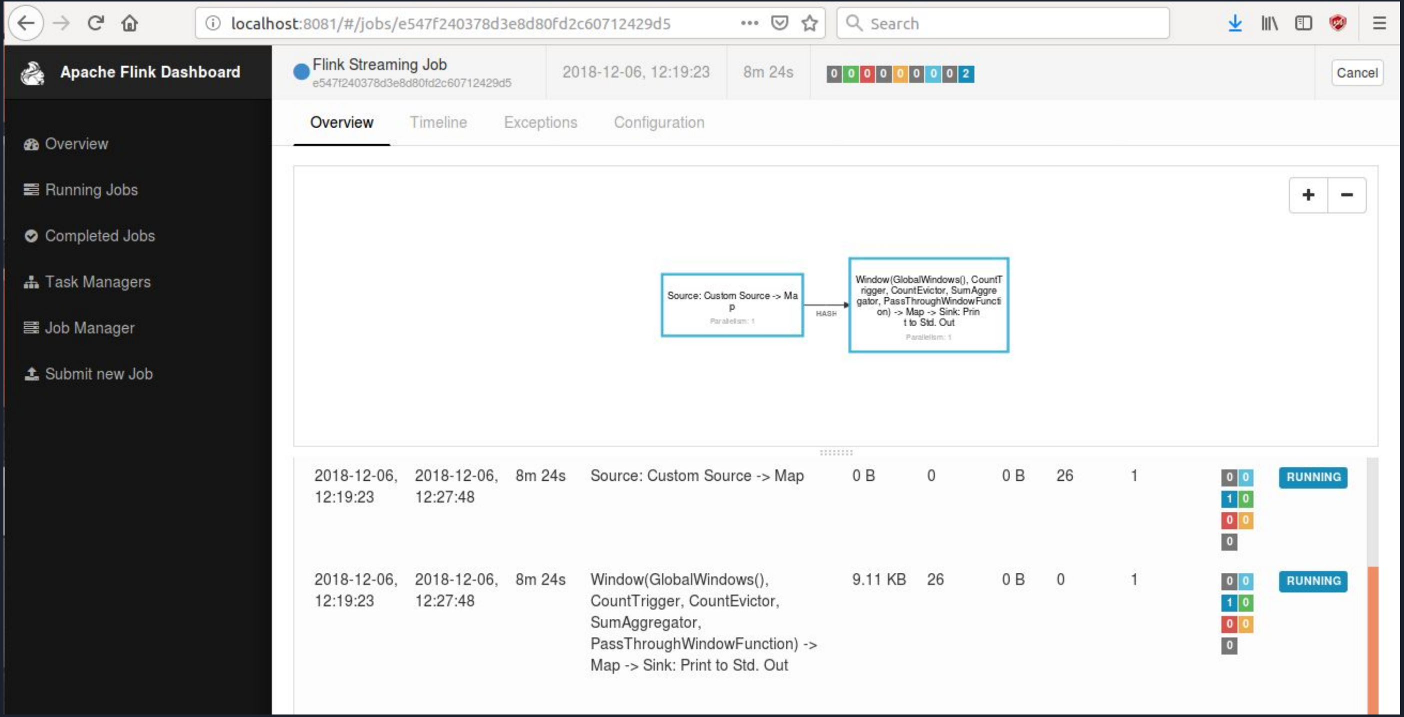
Architecture



Tools

- Stream Data Ingestion
  - Apache Flume Vs Apache Kafka
    - Choice Made : Apache Kafka
      - API’s to work with Spark And Flink.
      - Stream of records can be categorised into topics.
- Stream Data Analytics
  - Apache Spark Vs Apache Flink
  - Both have been tried

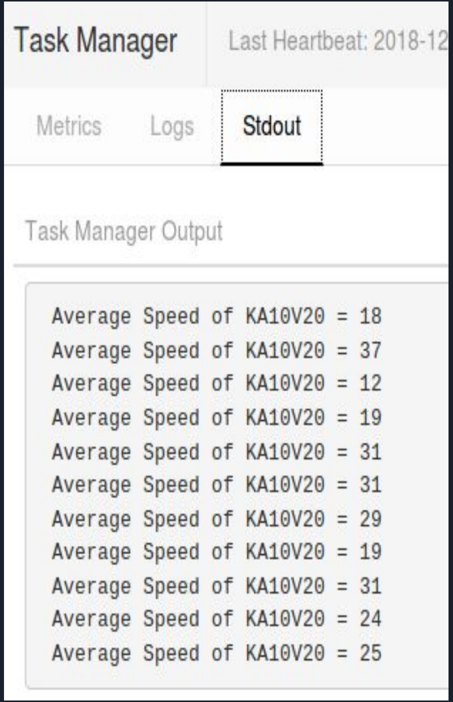
UI
(Apache Flink)


localhost:8081/#/jobs/e547f240378d3e8d80fd2c60712429d5
Search
Apache Flink Dashboard
Overview
Running Jobs
Completed Jobs
Task Managers
Job Manager
Submit new Job
Cancel
Overview
Timeline
Exceptions
Configuration
Source: Custom Source -> Map
Parallelism: 1
HASH
Window(GlobalWindows(), CountTrigger, CountEvictor, SumAggregator, PassThroughWindowFunction) -> Map -> Sink: Print to Std. Out
Parallelism: 1

Flink Streaming Job
e547f240378d3e8d80fd2c60712429d5

2018-12-06, 12:19:23 | 8m 24s | 0 0 0 0 0 0 0 0 2

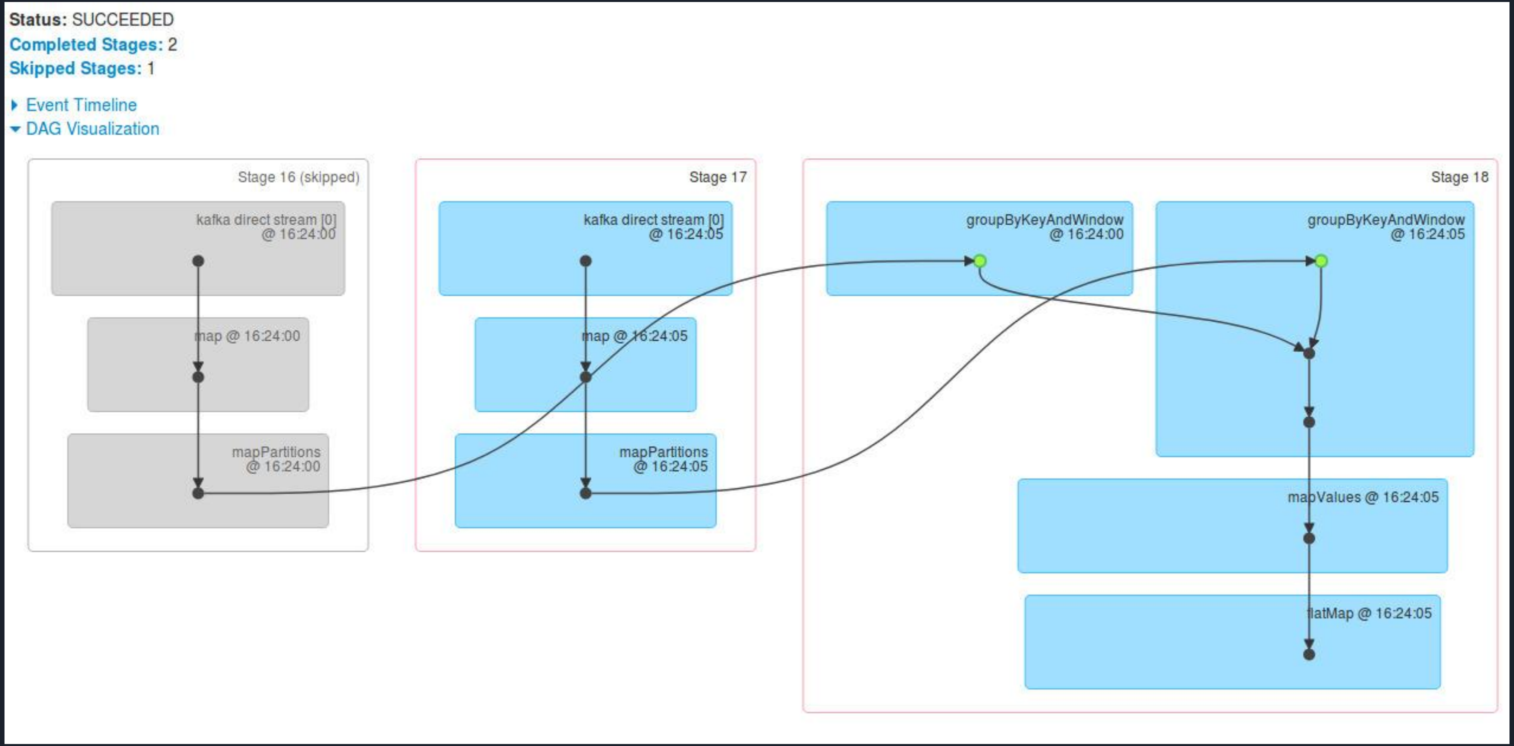
2018-12-06, 12:19:23	2018-12-06, 12:27:48	8m 24s	Source: Custom Source -> Map	0 B	0	0 B	26	1	0 0 1 0 0 0 0	RUNNING
2018-12-06, 12:19:23	2018-12-06, 12:27:48	8m 24s	Window(GlobalWindows(), CountTrigger, CountEvictor, SumAggregator, PassThroughWindowFunction) -> Map -> Sink: Print to Std. Out	9.11 KB	26	0 B	0	1	0 0 1 0 0 0 0	RUNNING



Task Manager Output

```
Average Speed of KA10V20 = 18
Average Speed of KA10V20 = 37
Average Speed of KA10V20 = 12
Average Speed of KA10V20 = 19
Average Speed of KA10V20 = 31
Average Speed of KA10V20 = 31
Average Speed of KA10V20 = 29
Average Speed of KA10V20 = 19
Average Speed of KA10V20 = 31
Average Speed of KA10V20 = 24
Average Speed of KA10V20 = 25
```

UI
(Apache Spark)



Status: SUCCEEDED

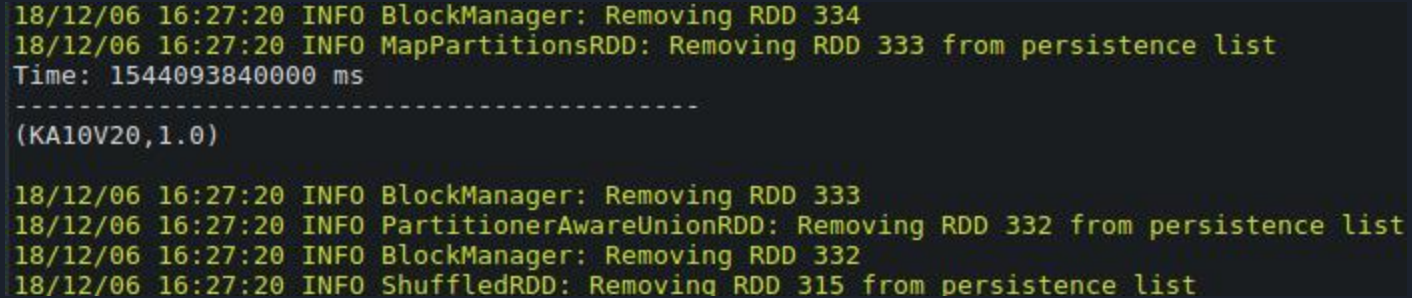
Completed Stages: 2

Skipped Stages: 1

▸ Event Timeline

▾ DAG Visualization

UI
(Apache Spark)



```
18/12/06 16:27:20 INFO BlockManager: Removing RDD 334
18/12/06 16:27:20 INFO MapPartitionsRDD: Removing RDD 333 from persistence list
Time: 1544093840000 ms
-------------------------------------------
(KA10V20,1.0)

18/12/06 16:27:20 INFO BlockManager: Removing RDD 333
18/12/06 16:27:20 INFO PartitionerAwareUnionRDD: Removing RDD 332 from persistence list
18/12/06 16:27:20 INFO BlockManager: Removing RDD 332
18/12/06 16:27:20 INFO ShuffledRDD: Removing RDD 315 from persistence list
```

Technologies Used

- Core Processing Modules : Java for Spark and Flink consumer programs.
- Tools Used :
  - Apache Kafka
  - Apache Flink
  - Apache Spark
  - Apache Zookeeper
- Kafka APIs for Spark and Flink

thank you!