Project - Sprint 3

Summary

In this sprint, you need to use your expertise in Spark and Kafka to process “Trip” data with “Station Information”.

Skillset

- Scala
- Hive
- HDFS
- Spark (Core, SQL, Streaming)
- Kafka
- Confluent Schema Registry
- Parquet
- Avro

Input

Trips

The main data for this pipeline is historical data of **trip**s. We will create a stream and process this stream in our pipeline.

To prepare data, go to the BiXi website (link provided in the documentation section) and under “Trip History” section, you will find links to the data from previous year all the way back to 2014. Download the data of the previous year. There is one file per month. Take the last month. The files are quite big, so we just take a sample. Extract the **first 100 lines** of the data after skipping the first line which is the CSV header. In Linux or Mac, you can run:

```
head -n 101 <filename> | tail -n 100 > 100_trips.csv
```

Prepare a command line to produce the content of *100_trips.csv* file to Kafka. You can use Kafka console producer command or implement a tool of your choice. It is better to be able to produce data in small batches e.g. 10 lines. It gives a better chance to monitor your pipeline.

Enriched Station Information

This is the artifact of your sprint 2. Note the location on HDFS.

Prerequisites

In order to start the project, you need to prepare the environment which is

- Create required Kafka topics
- Prepare the input
- Register Avro schema

Kafka

Create required Kafka topics:

- The input topic called `trip` that stores trip information in CSV format.
- The topic to hold enriched trip information in Avro format that is called `enriched_trip`

For all the topics, set the number of partitions to **1** and the replication factor to **1** as well.

Prepare input

Set a mechanism to read trip information and produced to input topic called trip. You are free to choose any method that you are comfortable the most with, e.g. Java, Python, Scala or just a shell script.

Notes:

- Kafka topic names should start with **<group-name>_<student-name>_** for example **bdsf1901_iraj_trip** etc.
- You must drop the first line that is the header of CSV file.
- Produce a sample set of 100 records.

Avro schema

Create the Avro schema for the enriched trip should be registered under **<group-name>_<student-name>_enriched_trip-value** subject.

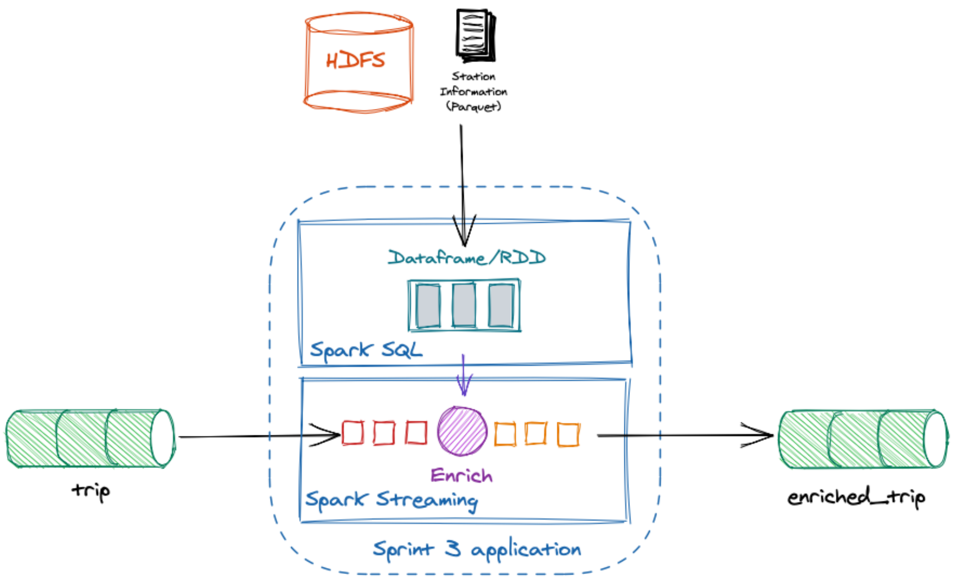
Curator

Implement a Spark application to curate trips with the enriched station information.

- **Spark SQL**: use Spark SQL to read the enriched station information from HDFS in the form of CSV.
- **Spark Streaming**: stream Kafka topic of `trip` into Spark `DStream` using Spark Streaming component. For each RDD in `DStream`, join it with the enriched station information RDD created in the previous step. The result would be a curated trip in Avro format.
- **Output:** the output is the Kafka topic of `enriched_trip`

Hint: it is easier if you have either two DataFrames or two key-value RDDs.

- **Two RDDs**: when you read the files using Spark session, it will give you a DataFrame that can be converted to RDD of string. In the result, each row is a CSV record. It is possible to convert it a Ket/Value RDD. The same method could be applied to the RDDs of `DStream`. Then, in order to curate trip with station information, you would be able to use join API of key/value RDD that results a new key/value RDD. For more information, check the resources section.
- **Two DataFrames**: you can convert RDDs of `DStream` to DataFrame using the method we discovered in the class. Then, there are two options: 1) use DataFrame API to join 2) create temp view and run SQL (check course materials).



To learn more about conversion, refer to the course materials.

Schema

Trip

Field name	*Type*
start_date	String
start_station_code	Integer
end_date	String
end_station_code	Integer
duration_sec	Integer
is_member	Integer

Enriched Station Information

Field name	*Type*
system_id	String
timezone	String

station_id	Integer
name	String
short_name	String
lat	Double
lon	Double
capacity	Integer

Enriched Trip

Field name	*Type*
start_date	String
start_station_code	Integer
end_date	String
end_station_code	Integer
duration_sec	Integer
is_member	Integer
system_id	String
timezone	String
station_id	Integer
name	String
short_name	String
lat	Double
lon	Double
capacity	Integer

Bonus

Visualize data in a visualization tool such as Tableau. Suggested visuals are:

- A card to show the average trip duration
- A histogram on the number of trips at each hour of day (if data is big, you can use average)
- A card to show percentage of trips taken by non-members
- Visualize stations on a map
- [Advanced] enable filter on trips to draw the start/end straight line on a map
- [Advanced] a heatmap to reflect busy stations

Provided documentation and artifacts

- Bixi is using General Bikeshare Feed Specification (GBFS) that you can find the documentation on GitHub (https://github.com/NABSA/gbfs/blob/master/gbfs.md)
- To get online feed of Bixi visit BiXi Open Data
- To get to know how to use Google web service to get address information using latitude and longitude information, check out (https://developers.google.com/maps/documentation/geocoding/intro#ReverseGeocoding)