Course 5 – Project

Design a batch ETL job using Spark

Objective

The objectives of this project are to get experience of coding with

- Spark
- Spark SQL
- Spark Streaming
- Scala and functional programming

Data set

The data set is the one that you analyzed in Course 1 and it is STM GTFS data.

Problem statement

We get the information of STM every day and need to run an ETL pipeline to enrich data for reporting and analysis purpose in real-time. Data is split in two

1. A set of tables that build dimension (batch style)
2. Stop times that needed to be enriched for analysis and reporting (streaming)

Project requirements

1. Data pipeline installation
Create a directory on HDFS for staging area called **/user/[GROUP]/[YOUR NAME]/project5/** where **[GROUP]** is the program. Ask the instructor if you don't have it and **[YOUR NAME]** is a nickname of your choice with only lowercase letters.
Create a directory for each source table called **/user/[GROUP]/[YOUR NAME]/project5/[TABLE NAME]** where **[TABLE NAME]** is from the following list • trips • calendar_dates • routes
Create a database called **[GROUP]_[YOUR NAME]** in Hive. If you already have one, just use and don't try to create multiple databases.
Create Kafka topic called **stop_times**

Create a directory for the result: **/user/[GROUP]/[YOUR NAME]/project5/enriched_stop_time**

Create an external table in Hive that points to this folder so we can verify the results. (Schema follows)

2. Extract data from STM to staging area

Download the data set of STM GTFS from http://stm.info/sites/default/files/gtfs/gtfs_stm.zip

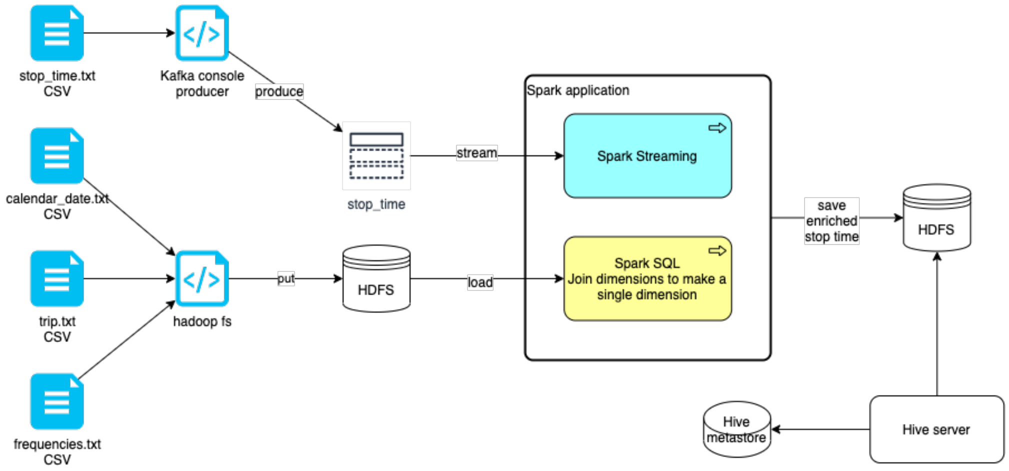
Put extracted version into **/user/[GROUP]/[YOUR NAME]/project5/[TABLE NAME]** path on HDFS where **[TABLE NAME]** here is the name of file without extension.

We just need the following tables

- trips
- calendar_dates
- routes

3. Data pipeline

1. BATCH: Enrich *trips* with *calendar dates* and *routes*
   a. Read trips, calendar dates and frequencies into DataFrame
   b. Enrich them to create an **enrichedTrip** DataFrame. You can use either of SQL query or “join” API
2. STREAM: stream *stop times* through Kafka and enrich them with *enriched trip* information.
3. Use a command line tool (kafka-console-producer) to produce *stop times* to **stop_time** topic (The stop times is a huge dataset. In order to avoid breaking the cluster, produce only 100 records.)
4. Stream **stop_time** into the Spark Streaming application
5. For each micro-batch, enrich the RDD of stop times with enriched trips dimension
6. Save enriched stop times on HDFS under result directory



SCHEMA

TRIP

Field Name	Data Type
trip_id	Integer
service_id	String
route_id	String
trip_headsign	String
wheelchair_accessible	Boolean

CALENDAR DATE

Field Name	Data Type
service_id	String
date	String
exception_type	Integer

ROUTE

Field Name	Data Type
route_id	Integer
route_long_name	String
route_color	String

ENRICHED TRIP

Field Name	Data Type
trip_id	Integer
service_id	String
route_id	String
trip_headsign	String
wheelchair_accessible	Boolean
date	String
exception_type	Integer
route_long_name	String
route_color	String

ENRICHED STOP TIME

Field Name	Data Type
trip_id	Integer
service_id	String
route_id	String
trip_headsign	String
wheelchair_accessible	Boolean
date	String
exception_type	Integer
route_long_name	String
route_color	String
arrival_time	String
departure_time	String
stop_id	String
stop_sequence	Integer