Naya College

Cloud Data Engineering

Apache Spark

Lab6: Real-time Review Data Processing with Spark Streaming

Contents

Lab6-Task1: Producing Data From Kafka To Kafka

Objective:
Build a real-time Spark Streaming application that integrates user review data from Kafka with static application details from S3, computes sentiment aggregates, and then publishes the enriched data back to Kafka and also saves it in Parquet format.

Guided Solution:

Setup Spark Streaming:

- Initialize a Spark session.
- Define the Kafka bootstrap servers and source topic (gps-user-review-source).

Consume Data from Kafka:

- Read data from the Kafka topic using Spark structured streaming. You already populated the topic,named **"gps-user-review-source**" with data in the previous exercise **Lab5-Task3**.
- Parse the incoming data (assuming it's in JSON format).

Load and Cache Static Data:

- Load static data from the path provided (/data/source/google_apps).
- Cache this data for better performance as it will be used in joins.

Data Enrichment and Aggregation:

- Group the streaming data by the application name.
- Calculate the following aggregates:
  - Count of positive sentiments.
  - Count of neutral sentiments.
  - Count of negative sentiments.
  - Average sentiment polarity.
  - Average sentiment subjectivity.
- Join the aggregated data with the static data on the application name.

Push Processed Data to Kafka:

- Convert the processed data to JSON format.
- Write the JSON data to the target Kafka topic (gps-with-reviews).
- Make sure to set the appropriate checkpointing directory for streaming resilience.

Store Processed Data in Parquet:

- Define the destination path for Parquet storage (/data/target/google_reviews_calc).
- Set a trigger to run the job every minute.
- Store the data in Parquet format.

- Use a checkpointing mechanism to track processed messages and ensure resilience.

Remember, each time data is stored, it will append to the Parquet file, offering a cumulative view of the processed data.
After completing the above steps, you will have a streaming application that integrates real-time review data with static data, calculates necessary aggregates, and persistently stores the results.

Lab6-Task2: Producing Data From Kafka To S3

Description:
The purpose of the **Data Update** script is to ingest, parse, and persist streaming data that contains both sentiment analysis results and app metadata. The streaming data originates from a Kafka topic that accumulates reviews and sentiments, and the script saves this enriched data into an S3 storage in Parquet format.

Instructions for Creating a Folder and Script in PyCharm:

1. **PyCharm setting**:
    - Name your directory **exercise_six**.
    - Right-click on the **exercise_six** directory you just created, Select **New** > **Python File**. Name your script **store_results.py**.

2. **Script Overview**:
    - The script **store_results.py** will be designed to handle specific data-related tasks. For example, it will pull data from a Kafka stream topic named **gps-with-reviews**, process it, and save the processed data in Parquet format in the path: **s3a://spark/data/target/google_reviews_calc**
    - The script will run every minute and continuously update and save new data.