Code Challenge Project Documentation

Bowen Liu

Environment

I did this project in Python and Spark 1.6.0 on Hadoop 2.6. I used PYSPARK to run the code.

How to run

First open the PYSPARK shell by typing the command below:

```
PYSPARK_DRIVER_PYTHON=ipython pyspark –packages com.databricks:spark-csv_2.10:1.3.0
```

The additional package I included is Spark-CSV package which is convenient for writing csv files. And then run Solution.py in the shell.

Basic ideas of my solution

The basic module I used is pyspark.sql module, first read the input csv files and then split each row by commas to form columns. And then I specified schemas by setting up the column name and data types. The schemas were applied to RDDs and the data frames were registered as tables. After registering tables for input files, I found all duplicated events from events table and used a list to store these duplicated events. After all duplicated events were found, the original events table will delete these rows by using "filter" function and registered a new events table. Finally, I wrote two SQL queries to select the desired results from this new events table and impressions table. And the desired results are stored in the output csv files.

Test cases

I made some test cases to test my program, the test cases of events are stored in test_events.csv, and the test cases of impressions are stored in test_impressions.csv. They have the same schemas with events.csv and impressions.csv. They only have a few rows for the simplicity of testing. But they cover everything: they have duplicated events, the events happened before any impressions, the events happened after some impression, two advertisers, two users, two event types. And the results for these test cases are stored in test_count_of_events.csv and test_count_of_users.csv. The results are perfectly correct.

Results

The results are stored in count_of_events.csv and count_of_users.csv files, they both contain three columns for each row: advertiser_id, event_type, count.