Project 1

Final Report

Big Data and Cloud Computing

João Agulha up201607930

Faculdade de Ciências da Universidade do Porto
Departamento de Ciência de Computadores

April 7, 2020

1 Summary

In the context of Big Data and Cloud Computing (BDCC) course, a study of techniques used for handling large amounts of data was developed. In this project, different tools were used, such as Python, Jupyter Notebook, Apache Spark and Google Cloud Platform to process MovieLens datasets. The final objective being the understanding of which approaches best work for machine learning applications, which approaches scale well enough as the amount of data increases and how all of this can be deployed through cloud computing utilities.

Specifically, Term Frequency-Inverse Document Frequency (tf-idf) and Jaccard Index metric with information processed by the LCF and SCF for similarity calculations were developed. Additionally, some queries and utility functions were added to the original provided notebooks to extend and provide new functionalities.

2 Technologies & Workflow

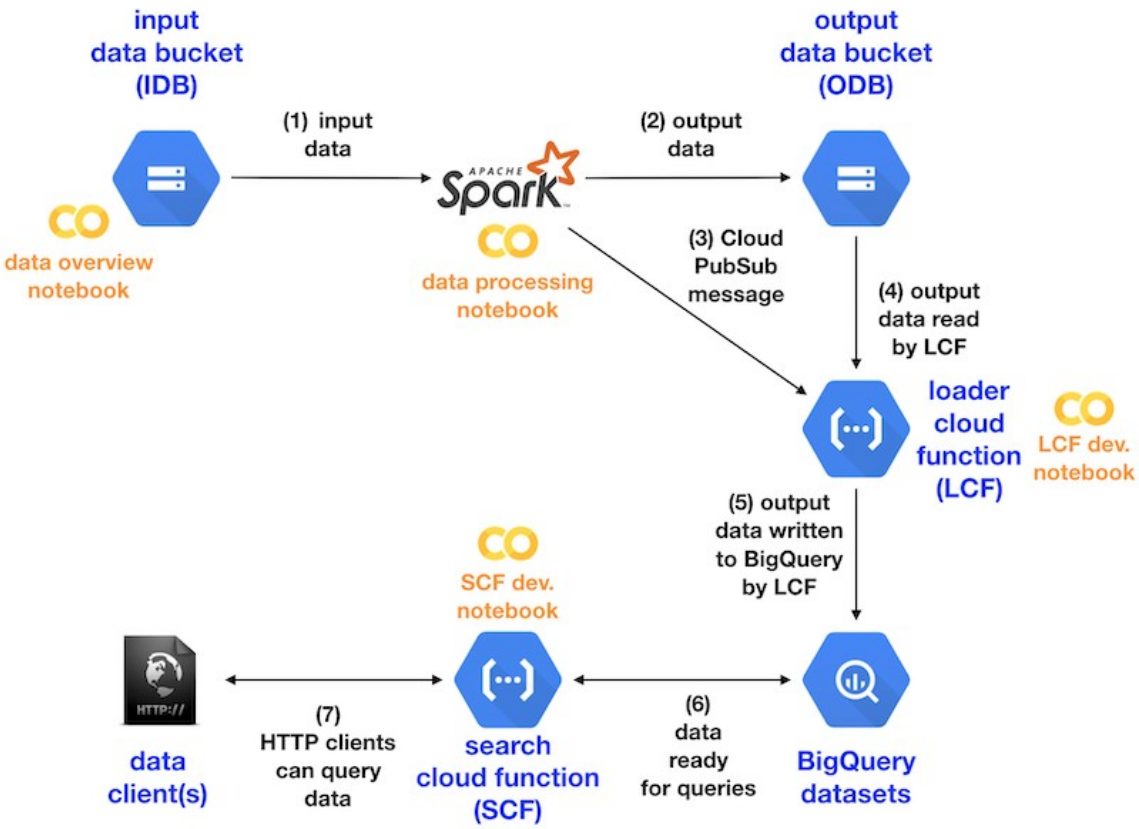

Figure 1: Technologies & Workflow Diagram

3 Development

The development of the project relies on two major concepts – Spark and GCP. Notebooks provide the Spark integration and functions while the GCP provides the underlying infrastructure. Use of GCP Buckets, Functions and BigQuery was made.

With this in mind, a generic approach was followed that would easily allow the reuse of these frequent notions. Notebooks were developed in a top-down structure of dependencies, as is recommended.

Google cloud project under the name **bdcc20-p1**
BigQuery query datasets **tiny1** and **medium1**

- **Dataset**

From the MovieLens Dataset, the following tables were used:

- **Movies**, with columns **movieId, title, year, imdbId**
- **Actors**, with columns **movieId, name**
- **Genres**, with columns **movieId, genre**
- **Tags**, with columns **userId, movieId, tag**
- **Ratings**, with columns **userId, movieId, rating**

- **TF-IDF**

TF-IDF metric is a numerical statistic metric which measures the importance of a word in the corpus of text documents. To implement it:

- Calculate the number of users in tag group term has been used in document.
- Calculate the maximum absolute frequency of any term used in document.
- Calculate the term-frequency value of term in document.
- Make join operation with number of documents with term appearing at least once.
- Calculate the inverse document frequency of term considering all documents with term at least once.
- Calculate the term frequency – inverse document frequency of term for document.

- **Jaccard Index**

Calculates the Jaccard similarity between tags based on the movies. It is applied to recommend tags. Given movie m, returns n tags based on their similarity to the tags associated to m.

- Remove ratings below threshold value (considered for a rating to be a like).
- Group liked ratings by movieId.
- Aggregate a bunch userIds corresponding to users who rated the movie.
- Later used to filter movies that have less than the minimum_number_ratings.
- Columns are renamed to user1 and movie1.
- Duplicate previous dataframe and rename columns to user2 and movie2.
- Cross product between both dataframes so that movie1 < movie2 thus

avoiding duplicates.

- Intersect both user sets.
- Union both user sets.
- Previous dataframe sizes used to generate the Jaccard Index.
- Unwanted columns are removed.

- **Other relevant notes**

- Good use of Spark overall concept and primitives.
- Detailed care with complexity, not only spatial, but also, temporal performance issues.
- Good use of User-Defined Functions.
- Code commented for better understanding algorithm steps.
- Use of debug messages to help to explain algorithm steps

4. Conclusions

All data transactions and querying expected to develop were successfully met. Development of this project provided extensive learning. I was able to acquire a greater knowledge in the area of big data and cloud computing, in particular, in the area of data manipulation, more specifically using Spark techniques. All requirements were met.