Data Analysis Using Map/Reduce (Bonus)

Submitted by

Team 2

Yash Avlani (1001670008)

Harshit Modi (1001662262)

Under Supervision of: Prof. Sharma Chakravarthy and GTA, Mr. Abhishek Santra

Department of Computer Science,

University of Texas at Arlington

Acknowledgements

My sincere thanks to **Prof. Sharma Chakravarthy,** Department of Computer Science & Engineering, University of Texas at Arlington, who gave us great lecture on Cloud Computing and Map/Reduce and gave us opportunity to implement project on Hadoop Cluster. Apart from this we thank Prof. Sharma for make us understand the other DBMS topic in depth.

We are also grateful to the **GTA, Mr. Abhishek Santra** for solving all our queries as soon as possible on Blackboard and in person. For his kind assistance and cooperation during the development of the project, we are delighted to have such a supportive GTA.

Overview:

This project implements bonus data analysis using Hadoop Cluster. The given 2 .txt input files with huge data will make us follow the method to produce a productive analysis in the real word. Here we have to analyse the IMDB dataset (IMDB_Actors and IMDB_Directors) and generate <key, value> pair indicating the the actor with their i. By concating both the dataset will give the matching acotrId and directorId will unable us to get the desired result. So, this is how it merges together these values to produce possibly smaller set of values.

Overall Status & File Description:

This bonus part of project is successfully completed. A mapper and reducer are implemented to derive the expected output. We implemented it with the 2 jobs chaining 2 separate mappers/reducers.

The major components of our implementations and their details are as follows:

- Making two different mappers for directors and actors file that outputs same keys
- The reducer for first job outputs only those keys for which only 2 entries exist
- The mapper of second job takes the input from the first job and map every entry with one
- The reducer of second job aggregate all the values given as the input and output those for which the sum is greater than or equal to 100.

Configurations:

- Hadoop 2.9.1
- OS : Ubuntu 16.04
- JAVA 10.0.2

Setup:

- After going through the project details and pdf given I got the basic idea about where to begin with and how to setup installation of hadoop single node cluster. We used the link given in project material to follow the installation process.
- After successfully completing the installation we made sure that we were able to solve basic WordCount problem using the java file provided.
- Starting and Stopping Services.
    - start-dfs.sh start-yarn.sh
    - stop-dfs.sh stop-yarn.sh
    - Namenode format : hadoop namenode -format

HDFS File System:

- HDFS file system is what Hadoop uses for the mechanism it provides for map/reduce.
- Input can be loaded to HDFS file system by following command.
    - Hadoop -dfs copyFromLocal IMDBTitles.txt

Generate jar and Run Map/Reduce:

- Jar file will allow the directory to access file
    - *bin/hadoop com.sun.tools.javac.Main QueryIMDB.java*

- *jar cf project3.jar QueryIMDB*.class*
- *hadoop jar project3.jar QueryIMDB inputhd outputhd*

Get output from HDFS to local:

- The output would be added to the HDFS directory
  - *dfs -copyToLocal /user/hadoop/output titlesOutput*

Where we encountered difficulty

The major difficulty for us was at installation. Even after completing the **installation** we were not able to connect to the **namenode** after restarting the system. It took us more than 2 hours to connect to the namenode. Again the same query came up with **datanode** but we solved it easily. The simple word count problem took us many attempts to be done. As while reaching to the **hadoop directory** and local host we faced many issues. Apart from this .jar file made some trouble for us. Analysing the output was a big task indeed.

Division of Labor

We have completed the project in a group of 2 members. As the skeleton code provided was with hadoop single node cluster and there was depth documentation available, though it took us time to get familiar with some components of the projects. We also spent significant time in making documentation on some methods and flows of project for our future reference, which helped us in further implementation. After completing the big task of install and setup hadoop, it didn't take much time for us to run the WordCount problem and the basic project implementation as well. The analysis task was really challenging as we had never dealt with analysing such data in past. Overall it took **5-6 sessions** to completing whole project.

Total time spent: 15 hrs approx

Logical errors

1. Taking **multiple inputfiles** to mappers

   As there were two databases given, we need to process two files through two different mappers. To do that we made research and found in hadoop documentation the way to implement it. We solved it by using **MultipleInputs** class that was given in Hadoop directory.

2. **Chaining** multiple mappers and reducers

   As the problem definition was complex, it was hard to solve it with only one mapper/reducer code. To achieve our goal, we made two different jobs with separate mappers and reducers. In first job we calculated the actors who appeared in both files IMDB_ACTORS.txt and IMDB_DIRECTOR.txt. This gives us the names of actors who appeared in same movie that they have directed. In second job, we calculated the count of entries and gave output of the entries for which the count is greater than or equal to 100. This gave us the required result.

3. **Analysis** phase of the project.

   This time the task was what should be the analysis. Because we were confused about writing results which are really useful in real word. Then we managed to get some graphs and came with some important results which can be really useful for producers whether to invest on the actor-director concept or not. We can derive some important data about the directors who have worked with this concept most number of times. Apart from this, we can fetch some category in which Actor-Director fall and by that we can derive the popularity of this concept by comparison.