CA675 Cloud Technologies

Name	Salman Khaleel Sab
Student Number	18210266
Programme	MCM (Cloud)
Module Code	CA675
Assignment Title	Assignment One - Large Data Analysis
Submission date	08-March-2019
Module coordinator	Alessandra Mileo

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offenses in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the referencing guidelines found recommended in the assignment guidelines.

Name : Salman Khaleel Sab
Date : 08-March-2019

I. INTRODUCTION

Stack Exchange [SE] is a website where any users can post questions on various topics in any field. It is a network of questions and answers where anybody can answer to the queries posted on a particular topic. Many major websites are part of SE such as Stack Overflow, Server Fault, Ask Ubuntu, Super User, and many others. In this assignment, Analysis of Large dataset of Stack Overflow has been done using Big Data Technologies: Pig, Hive on Hadoop framework. MapReduce [1] is a programming technique that can be implemented to process and analyze Big datasets on a distributed, parallel computing cluster. Hadoop [2] is a framework comprised of open source software utilities built on the Java platform. The framework uses the MapReduce technique to perform parallel and distributed multiprocessing. Apache Pig [3] is an opensource technology that is built on Hadoop framework. Apache Hive [4] is a data warehouse infrastructure built on the Hadoop system which facilitates querying large datasets residing in the Hadoop storage. The rest of the report is structured in the following way: Section II presents the details of data acquisition, dataset explanation and preprocessing of data. Section III focuses on the roles of Pig and Hive technologies used in this work. Section IV presents a brief information of Google Dataproc in this work. Finally, Section V discusses over all conclusions.

II. DATASET

Stack Overflow's [5] data is extracted from SE by querying on the table POSTS. It has 22 columns and viewCount is one of the columns. We have analyzed 200000 records in this work. It is possible to fetch only 50000 records per query so, we obtained four separate csv files by running four queries to obtain the top 200000 records ordered by viewCount in descending order. The queries are provided along with this report in the file named "*DataAcquisition.txt*".

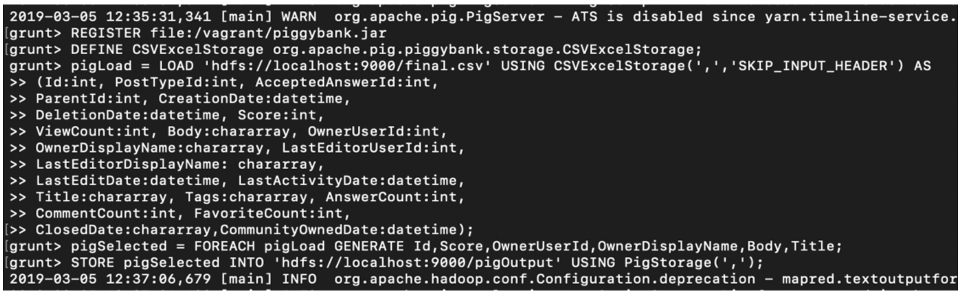
It is necessary to clean the data to load it into Pig. So, we made use of Python 3.7.1 [6] with pandas and re (regular expressions) libraries to clean the data. HTML tags, new-line characters, commas, and special characters are removed from Body and Title columns using Regular expressions. Four csv files are then merged to a single file named "*final.csv*". The source code for the same is submitted along with the report.

III. PIG AND HIVE USAGE

Apache Pig has a Java repository to support User Defined Functions(UDF) called Piggybank [7]. It is a jar file where useful functions for data loading has been written in Java, this Jar file should be registered by specifying the path of the directory where it resides. Once it is registered, we can use any functions defined in the Jar. We can register the jar files with the following syntax "REGISTER <path> ". After registering the UDF, it is a good practice to define the function alias that we want to use. We can specify the alias name with the "DEFINE" keyword. Before we start loading the data into Pig it is mandatory to make sure all necessary Hadoop Daemons are up and running. We can use command "*jps*" to check if the daemons have started. Basic Hadoop commands are provided in a file named "*HadoopCommands.txt*".

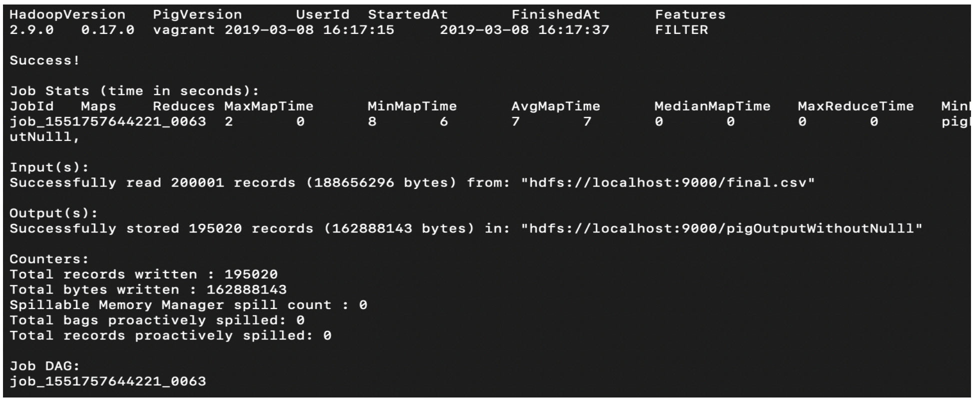
The LOAD operator can be used to load the input data which is stored in the local or Hadoop directory. We have used CSVExcelStorage defined in Piggybank jar.

Null values in the Body and Title fields of the table are removed using the Pig FILTER operator. Once Null values are removed, we can select the necessary columns or fields required into a new relation and then it can be stored into local or Hadoop directory using STORE Keyword. The Pig script used for this work is submitted in the file named "*PigScripts.txt*". Below are the screenshots of Pig Scripts used in this work.



```
2019-03-05 12:35:31,341 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.
grunt> REGISTER file:/vagrant/piggybank.jar
grunt> DEFINE CSVExcelStorage org.apache.pig.piggybank.storage.CSVExcelStorage;
grunt> pigLoad = LOAD 'hdfs://localhost:9000/final.csv' USING CSVExcelStorage(',','SKIP_INPUT_HEADER') AS
>> (Id:int, PostTypeId:int, AcceptedAnswerId:int,
>> ParentId:int, CreationDate:datetime,
>> DeletionDate:datetime, Score:int,
>> ViewCount:int, Body:chararray, OwnerUserId:int,
>> OwnerDisplayName:chararray, LastEditorUserId:int,
>> LastEditorDisplayName: chararray,
>> LastEditDate:datetime, LastActivityDate:datetime,
>> Title:chararray, Tags:chararray, AnswerCount:int,
>> CommentCount:int, FavoriteCount:int,
>> ClosedDate:chararray,CommunityOwnedDate:datetime);
grunt> pigSelected = FOREACH pigLoad GENERATE Id,Score,OwnerUserId,OwnerDisplayName,Body,Title;
grunt> STORE pigSelected INTO 'hdfs://localhost:9000/pigOutput' USING PigStorage(',');
2019-03-05 12:37:06,679 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.textoutputfor
```

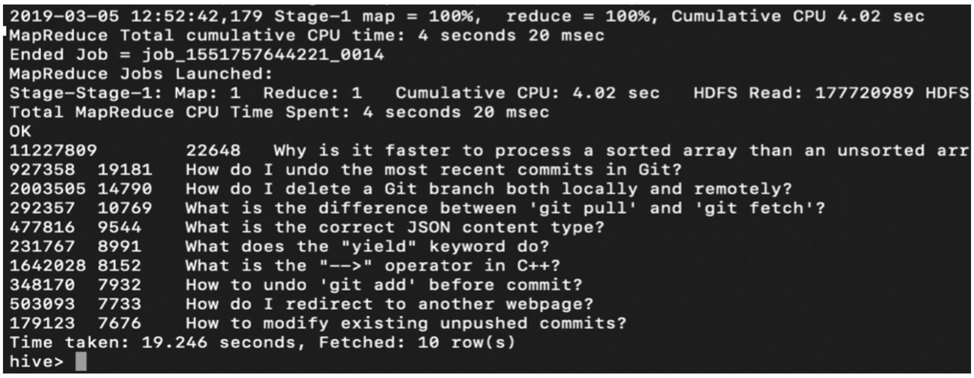
Image 1. Pig Loading Screenshot



```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.9.0   0.17.0  vagrant 2019-03-08 16:17:15     2019-03-08 16:17:37     FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   Min
job_1551757644221_0063  2       0       8       6       7       7       0       0       0       0       pig
utNulll,

Input(s):
Successfully read 200001 records (188656296 bytes) from: "hdfs://localhost:9000/final.csv"

Output(s):
Successfully stored 195020 records (162888143 bytes) in: "hdfs://localhost:9000/pigOutputWithoutNulll"

Counters:
Total records written : 195020
Total bytes written : 162888143
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1551757644221_0063
```

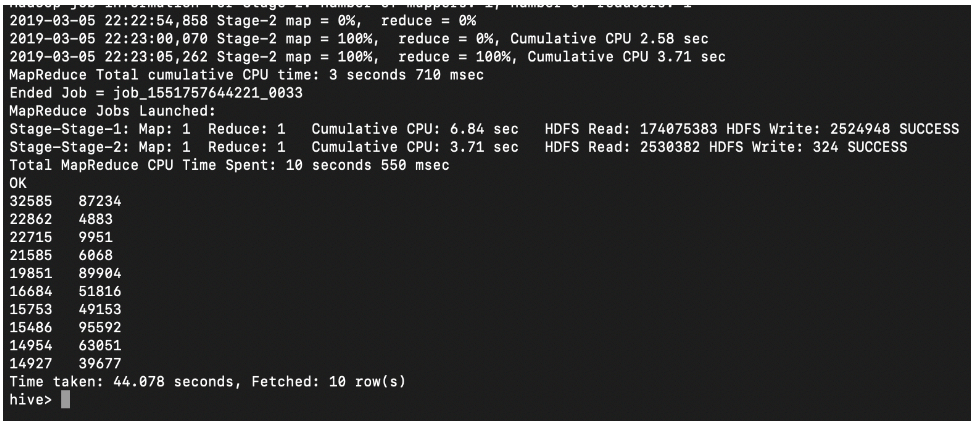
Image 2. Pig Storage Successful Screenshot

Apache Hive is SQL like querying language which facilitates querying of large datasets. In this work, the data stored in the Hadoop directory which is the output of Pig explained earlier is loaded directly into the Hive table while creating. Once the data is loaded into hive tables, we can start querying on the tasks mentioned as a part of this work. Below are the screenshots of Task 1, Task 2 and Task 3 output.



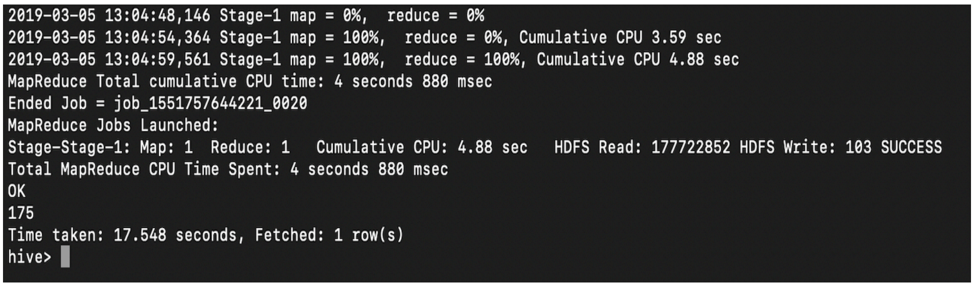
```
2019-03-05 12:52:42,179 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.02 sec
MapReduce Total cumulative CPU time: 4 seconds 20 msec
Ended Job = job_1551757644221_0014
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.02 sec   HDFS Read: 177720989 HDFS
Total MapReduce CPU Time Spent: 4 seconds 20 msec
OK
11227809        22648   Why is it faster to process a sorted array than an unsorted arr
927358  19181   How do I undo the most recent commits in Git?
2003505 14790   How do I delete a Git branch both locally and remotely?
292357  10769   What is the difference between 'git pull' and 'git fetch'?
477816  9544    What is the correct JSON content type?
231767  8991    What does the "yield" keyword do?
1642028 8152    What is the "-->" operator in C++?
348170  7932    How to undo 'git add' before commit?
503093  7733    How do I redirect to another webpage?
179123  7676    How to modify existing unpushed commits?
Time taken: 19.246 seconds, Fetched: 10 row(s)
hive> █
```

Image 3. Task 1 Output



```
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2019-03-05 22:22:54,858 Stage-2 map = 0%,  reduce = 0%
2019-03-05 22:23:00,070 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.58 sec
2019-03-05 22:23:05,262 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.71 sec
MapReduce Total cumulative CPU time: 3 seconds 710 msec
Ended Job = job_1551757644221_0033
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.84 sec   HDFS Read: 174075383 HDFS Write: 2524948 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.71 sec   HDFS Read: 2530382 HDFS Write: 324 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 550 msec
OK
32585   87234
22862   4883
22715   9951
21585   6068
19851   89904
16684   51816
15753   49153
15486   95592
14954   63051
14927   39677
Time taken: 44.078 seconds, Fetched: 10 row(s)
hive> █
```

Image 4. Task 2 Output

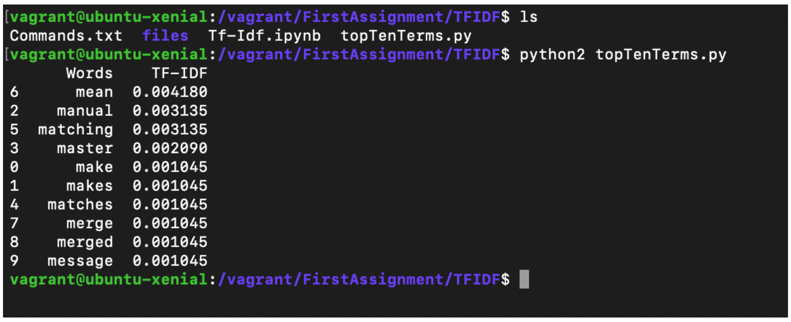


```
2019-03-05 13:04:48,146 Stage-1 map = 0%,  reduce = 0%
2019-03-05 13:04:54,364 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.59 sec
2019-03-05 13:04:59,561 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.88 sec
MapReduce Total cumulative CPU time: 4 seconds 880 msec
Ended Job = job_1551757644221_0020
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.88 sec   HDFS Read: 177722852 HDFS Write: 103 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 880 msec
OK
175
Time taken: 17.548 seconds, Fetched: 1 row(s)
hive> █
```

Image 5. Task 3 Output

As the result of Task two we will get the top ten users by post score and for each of these users we need to find out the top 10 terms used by each one of them by calculating tf-idf [8], This is the Task four. A new table called TFIDF is created to store the results of Task two which will give the top 10 users. By selecting each user from the TFIDF table, Body and Title content of each post of the top ten users is extracted and stored into text file <userId>.txt format where userId represents each user. So, we have now ten text files of the top ten users containing Body and Title data.

These ten text files are provided as inputs for tf-idf calculation using MapReduce. Python 2.7.12 is used to calculate tf-idf. Four mapper functions, three reducer functions are used for this task, the Source code for this task is submitted along with this report and it can also be found on GitHub [9]. Below is the screenshot of the top ten terms used by the user 4883.



```
[vagrant@ubuntu-xenial:/vagrant/FirstAssignment/TFIDF$ ls
Commands.txt  files  Tf-Idf.ipynb  topTenTerms.py
[vagrant@ubuntu-xenial:/vagrant/FirstAssignment/TFIDF$ python2 topTenTerms.py
      Words    TF-IDF
6      mean  0.004180
2    manual  0.003135
5  matching  0.003135
3    master  0.002090
0      make  0.001045
1     makes  0.001045
4   matches  0.001045
7     merge  0.001045
8    merged  0.001045
9   message  0.001045
vagrant@ubuntu-xenial:/vagrant/FirstAssignment/TFIDF$ 
```

Image 6. Task 4. Top Ten terms used by the user 4883(ownerUserId)

The Hive Queries used to perform all of these tasks are submitted in the file named "*HiveQueries.txt*". Since we have page limit for this report only one screenshot of tf-idf and cropped screenshots of other tasks are attached. Detailed and rest of the screenshots are submitted in a folder named "*Screenshots*".

IV. Google Dataproc

Google Cloud platform offers a public service for the analysis of Big Data called Cloud Dataproc [10].
This platform comprises of a wide variety of public offerings for running Apache Hadoop clusters and Apache Spark. It is simple, cost-effective and quick solutions for Big Data analysis where anybody can create and launch clusters within a few minutes. It provides many open source technologies such as Hive, Pig, Spark, and others for Extract, Transforms and Load processes. In this work, I created a single node cluster consisting of 4 vCPUs and a storage capacity of 20GB. Task one, two and three are executed in the Dataproc. Due to the limitation of space, screenshots are not attached in this report. Screenshots are provided along with report separately.

V. CONCLUSION

The task of Big Data analysis was challenging in earlier days before the cloud solutions. Public Cloud services offered today for Big Data analysis makes the whole process simple, quick and cheap. In this work, I have analyzed the data on both locally(vagrant) and on the cloud platform, working locally on a virtual machine with Hadoop framework, Hive and Pig technologies made me understand the Hadoop ecosystem but, it also has its disadvantages such as time-consuming, manual setup of each technology separately. On the other side, if we use cloud platforms, it has its advantages where the user need not worry about the setup and configuration also it is quick. Overall, whether it is a local platform or a cloud we see the major benefits of using the MapReduce programming technique for the Big Data analysis makes the whole process fast.

REFERENCES

[1] "MapReduce - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/MapReduce. [Accessed 07 March 2019].

[2] "Apache Hadoop," [Online]. Available: https://hadoop.apache.org/. [Accessed 07 March 2019].

[3] "Welcome to Apache Pig!," [Online]. Available: https://pig.apache.org/. [Accessed 07 March 2019].

[4] "Apache Hive TM," [Online]. Available: https://hive.apache.org/. [Accessed 07 March 2019].

[5] "Stack Overflow," [Online]. Available: https://stackoverflow.com/. [Accessed 08 March 2019].

[6] "Welcome to Python.org," [Online]. Available: https://www.python.org/. [Accessed 08 March 2019].

[7] "Apache Pig Tutorial," [Online]. Available: https://www.tutorialspoint.com/apache_pig/index.htm. [Accessed 08 March 2019].

[8] "Tf-idf :: A single page tutorial," [Online]. Available: http://www.tfidf.com/. [Accessed 08 March 2019].

[9] D. PATEL, "devangpatel01/TF-IDF-implementation-using-map-reduce-Hadoop-python-," [Online]. Available: https://github.com/devangpatel01/TF-IDF-implementation-using-map-reduce-Hadoop-python-. [Accessed 08 March 2019].

[10] "Cloud Dataproc," [Online]. Available: https://cloud.google.com/dataproc/. [Accessed 08 March 2019].