CA675-ASSIGNMENT-1

Student Name: Rahul Rajendra Sidhapurker

Student Number: 20211401

The main objective of the assignments is:

1. Acquire the top 200,000 posts by viewcount.
2. Using Pig or MapReduce, extract, transform and load the data.
3. Using Hive and/or MapReduce, get:

I. The top 10 posts by score

II. The top 10 users by post score

III. The number of distinct users, who used the word “Hadoop” in one of their

Posts.

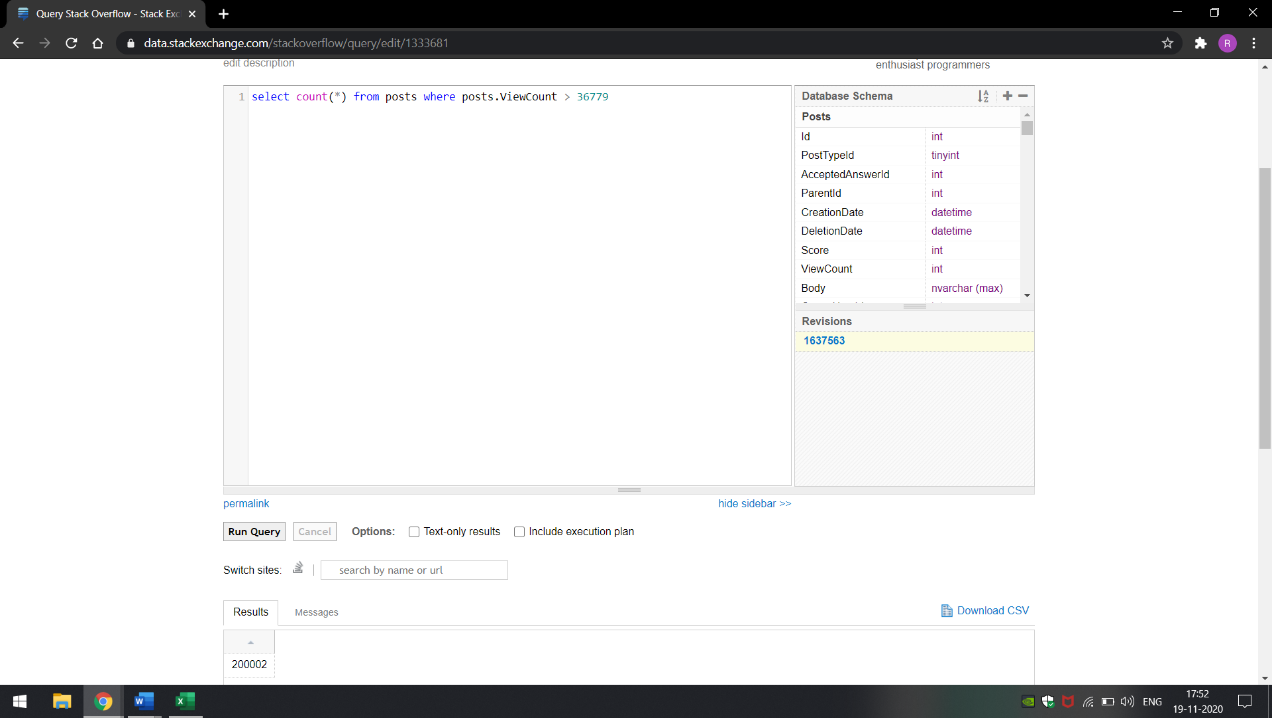
1. calculate the per-user TF-IDF for the top10 terms for each of the top 10 users from Query 3.II

To find the top 200,000 posts we needed to query it through SQL as follows:

In order to find the lower limit for the top 20000 posts we run the following query

**select count(\*) from posts where posts.ViewCount > 36779**

this gives us the output of 200002 which can be considered for top 200000 posts.



Now in order to get the top 50000 posts we run the query

**select top 50000 \* from posts where posts.ViewCount>36779 order by posts.ViewCount desc**

For the next 50000 posts we use the last row as the limit for the next 50k

**select top 50000 \* from posts where posts.ViewCount < 112523 order by posts.ViewCount desc**

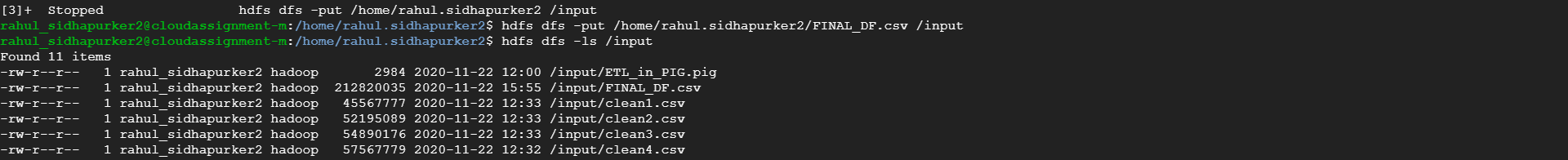
Similarly, same steps are repeated to get 4 csv files, each containing 50000 entries.

**select top 50000 \* from posts where posts.ViewCount < 66243 order by posts.ViewCount desc**

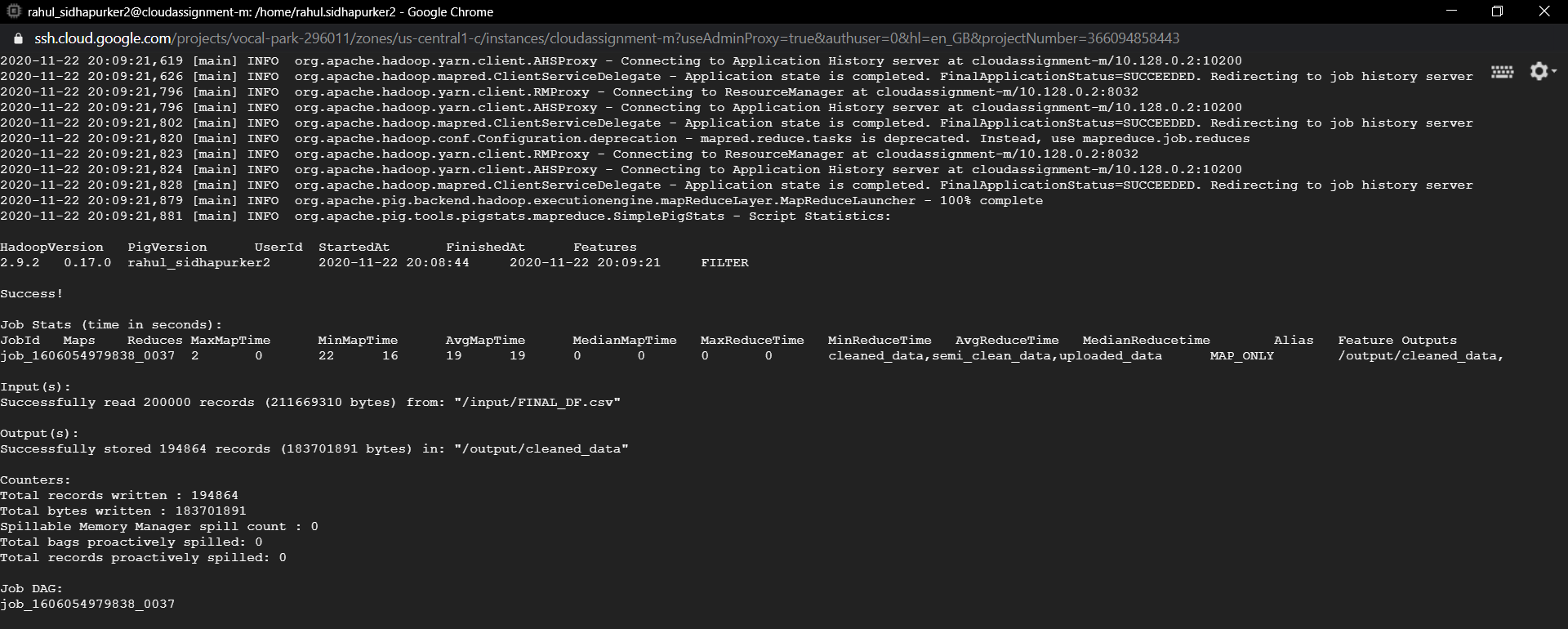
**select top 50000 \* from posts where posts.ViewCount < 47290 order by posts.ViewCount desc**

We now load these csv files via python to remove unnecessary HTML tags, new lines, tab spaces from “Body” and “Title” Columns. These csv files are now merged into single csv file using the append function.

This single csv file is now uploaded into the GCP and transferred into the hdfs file location

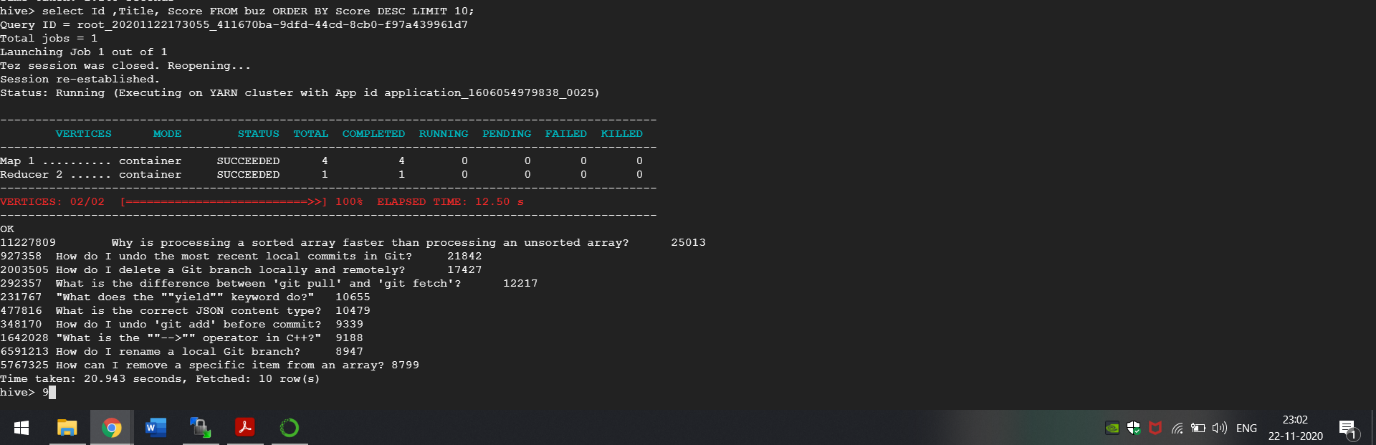


Now our Pig Script is run which loads the csv file from hdfs location, removes unnecessary columns and commas from Body, Title, Tags Columns. Futher columns containing NULL values are removed.

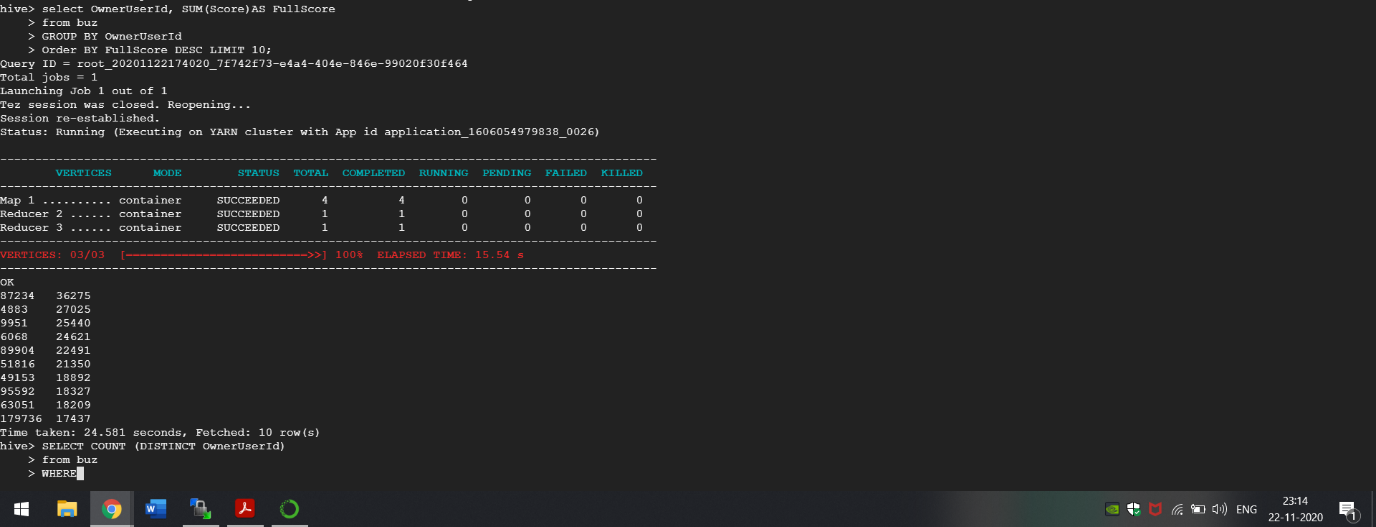
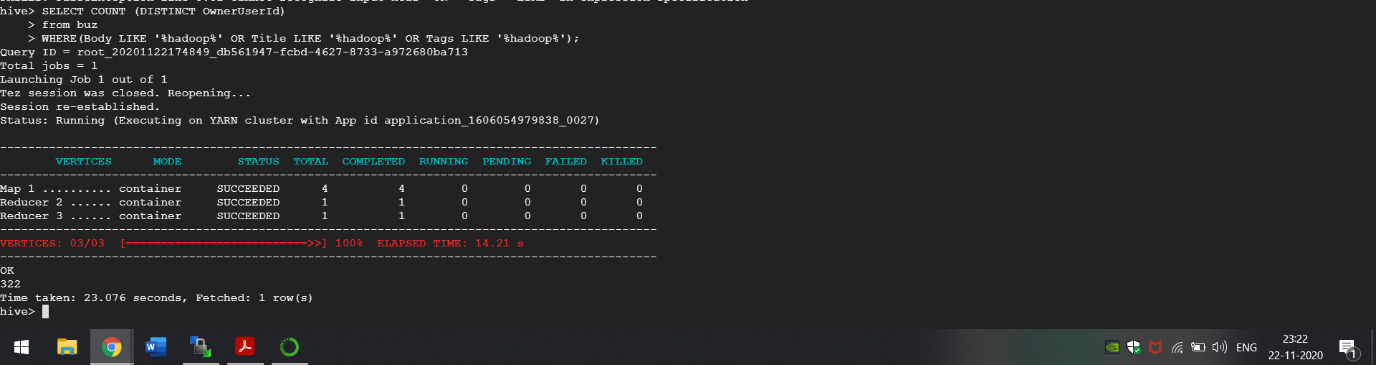


After the cleaned data is obtained, we now use hive to create a table containing only required columns for the analysis like Id, Score, Title, Tags, OwnerUserId, Body.

Now the first question is answer by querying the hive table and displaying 10 posts by score.

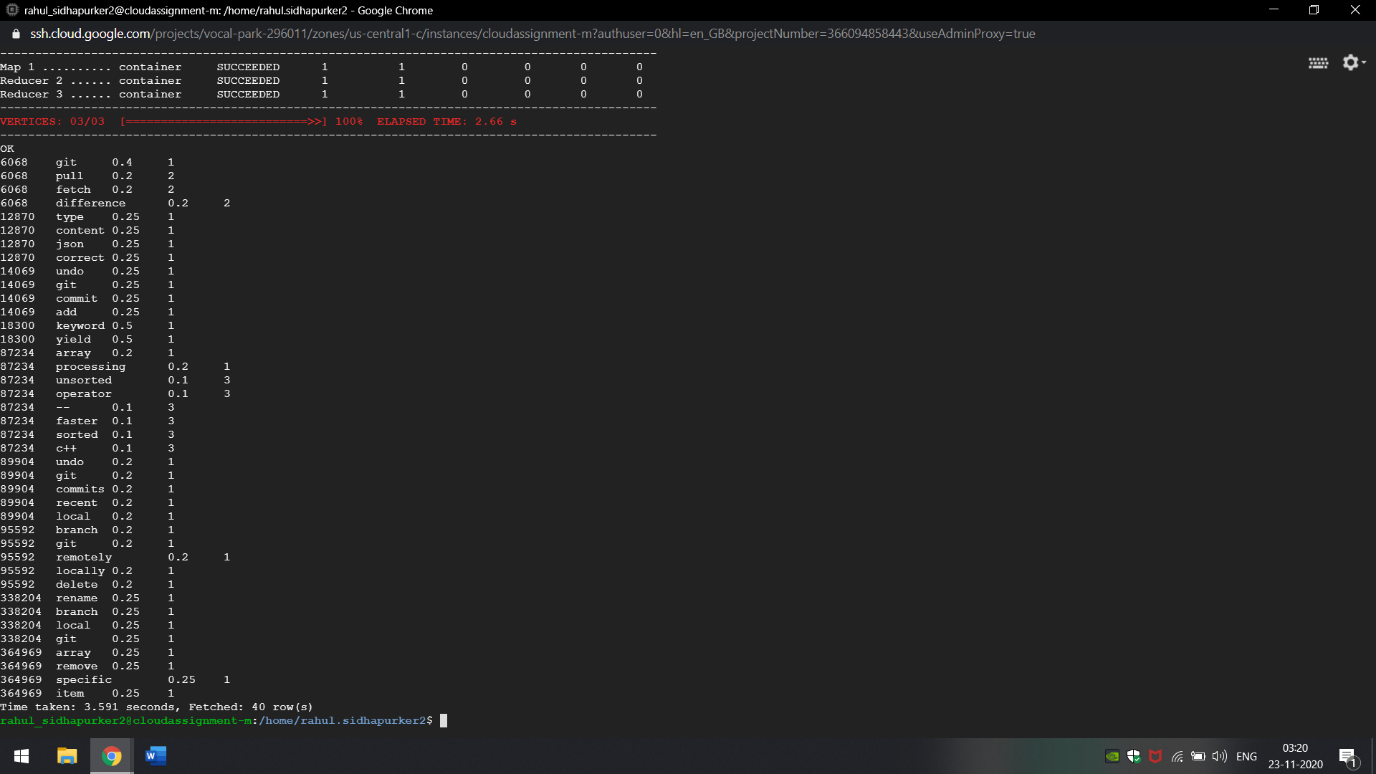


Similarly second and third questions are also answered by querying the hive table.



The Last question to calculate the per-user TF-IDF for the top10 terms for each of the top 10 users from Query 3.II is done as follows:

TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. Hivemall was used to calculate the per user Tf-IDF. The Documentation was the main source for our reference. We have run the script containing the code to get the per user TF-IDF.



Github Link: www.github.com/rahul9830/CA675-Assignment\_1/

References:

1. https://www.tutorialspoint.com/apache\_pig/apache\_pig\_reading\_data.htm
2. <http://xquaredl.blogspot.com/2016/08/load-and-store-data-using-pig.html>
3. https://hivemall.incubator.apache.org/userguide/ft\_engineering/tfidf.html
4. https://en.wikipedia.org/wiki/Tf%E2%80%93idf