CA675 - Assignment 1

Name : Bhargav Anant Athavale

Student ID : 20210278

Email : bhargav.athavale2@mail.dcu.ie

Git Repository Link - https://github.com/bhargavdcu/CA675Assignment1

Task 1 – Data Acquisition

Fetch a total of 200,000 records. The website restricts us to obtain a maximum of 50,000 records in each query. Therefore, run a total of 4 queries and combine the CSV files to obtain a single CSV file. Order the records by viewcount Obtain the dataset from the below link.

https://data.stackexchange.com/stackoverflow/query/new

Query 1

```
select top 50000 pos.*,usr.DisplayName from posts AS pos join users AS usr on
pos.OwnerUserId=usr.Id ORDER BY pos.ViewCount DESC
```

Query 2

```
select top 50000 pos.*,usr.DisplayName from posts AS pos join users AS usr on
 pos.OwnerUserId=usr.Id
 and pos.ViewCount<124974 ORDER BY pos.ViewCount DESC
```

Query 3

```
select top 50000 pos.*,usr.DisplayName from posts AS pos join users AS usr on
pos.OwnerUserId=usr.Id
 and pos.ViewCount<73139 ORDER BY pos.ViewCount DESC
```

Query 4

```
select top 50000 pos.*,usr.DisplayName from posts AS pos join users AS usr on
pos.OwnerUserId=usr.Id
 and pos.ViewCount<52110 ORDER BY pos.ViewCount DESC
```

The files were combined using the below command

```
cat FQ1.csv FQ2.csv FQ3.csv FQ4.csv > FQ.csv
```

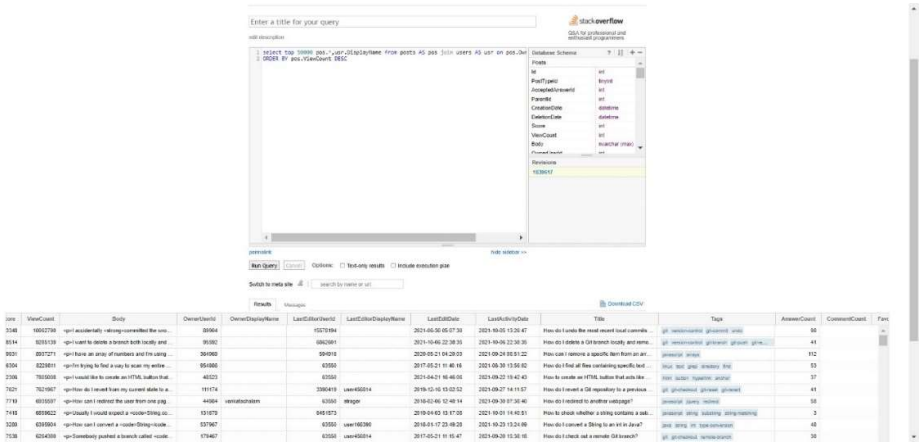

Figure 1:Stack Exchange Data

Task 2 – Extract, Transform and Load

The body, title columns of the data need to be cleaned before loading it into the hive database.**Python,Pig** was used for this purpose. **Pig was not able to clean the 'title' column efficiently**. Hence we used python for the purpose. After cleaning, we export the CSV. The python code for the same has been attached in the repository. Find below snippet

```
# In[13]:

#To Clean Body Column

stack_posts['Body'] = stack_posts['Body'].str.replace(r'\n','',regex=True)

stack_posts['Body'] = stack_posts['Body'].str.replace('[^A-Za-z0-9 ]+',' ',regex=True)

#To Clean Title Column

stack_posts['Title'] = stack_posts['Title'].str.replace(r'\n','',regex=True)

stack_posts['Title'] = stack_posts['Title'].str.replace('[^A-Za-z0-9 ]+',' ',regex=True)

Exporting to CSV

stack_posts.to_csv('FQ_Stack.csv', index=False)
```

Load File Into HDFS

hadoop fs -put /home/sunny/675_Submission/FQ_Stack.csv /

Pig was used to further clean the dataset and the final file was loaded into Hive database. CSVExcelStorage was used for this purpose.For cleaning, use 'Replace' function in Pig and export it into HDFS, from which new CSV File is obtained which can be loaded into Hive.
In Hive create a table with the necessary columns and load the data. Hive Query Language is used because it allows us to write map-reduce programs in similar to SQL queries.

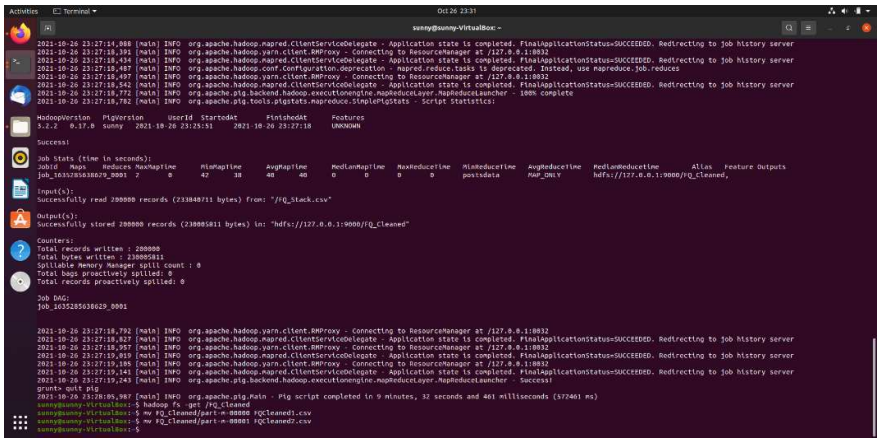

Figure 2:Clean Data in Pig and Export

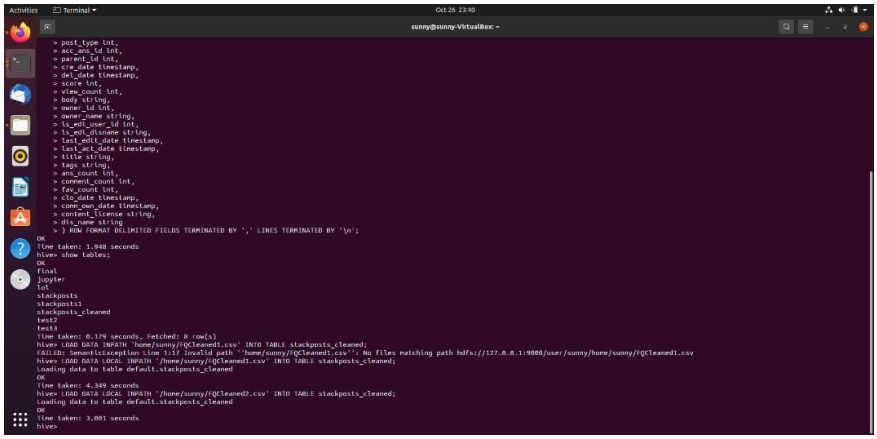

Figure 3:Load Data Into Hive Table

Task 3 – Queries

2.2.1. **The top 10 posts by score**

SELECT post_id,title,score FROM stackposts_cleaned SORT BY score DESC LIMIT 10

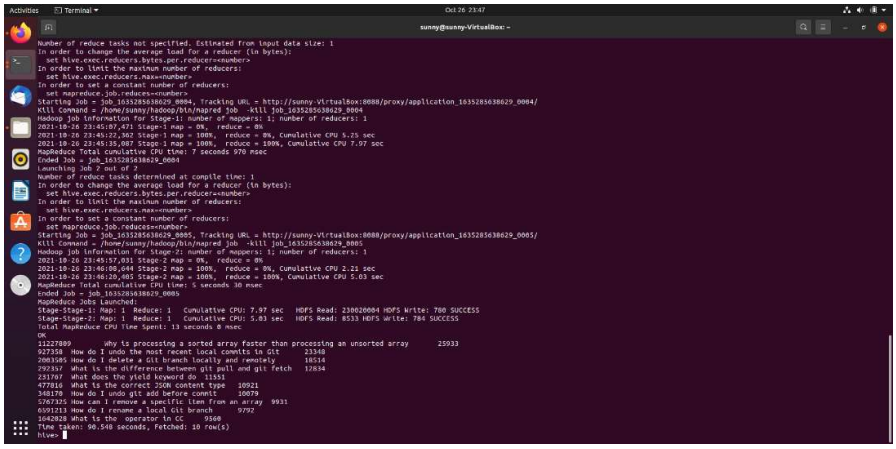

Figure 4:Top 10 posts by score

2.2.2. **The top 10 users by post score**

SELECT owner_id,sum(score) AS sum_score FROM stackposts_cleaned WHERE owner_id IS NOT NULL GROUP BY owner_id ORDER BY sum_score DESC LIMIT 10;

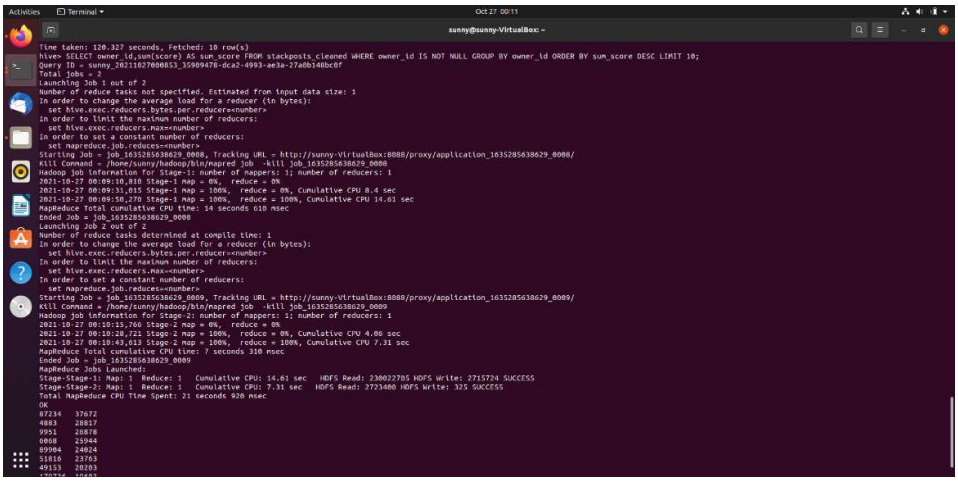

Figure 5:Top 10 Users By Post Score

2.2.3. The number of distinct users, who used the word “cloud” in one of their posts

SELECT COUNT(DISTINCT(owner_id)) FROM stackposts_cleaned WHERE lower(body) like '%cloud%' OR lower(title) like '%cloud%' OR lower(tags) like '%cloud%';

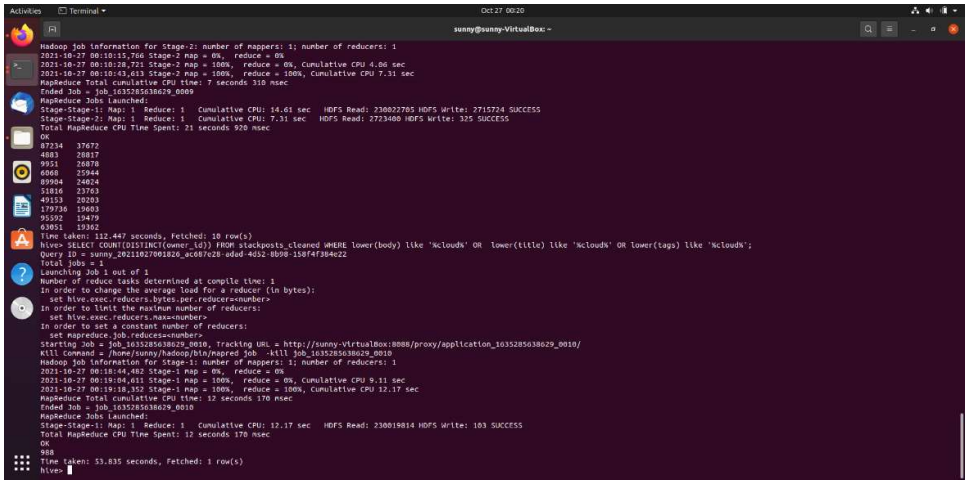

Figure 6:Number of distinct users, who used the word “cloud”

Task 4 - Calculate the per-user TF-IDF of the top 10 terms for each of the top 10 users

TF – IDF is Term Frequency/Inverse Document Frequency. To calculate TF-IDF , first get the whole data for top ten users.Using 2nd query, get 10 users and store into table. Using mappers and reducer files (reference given below), get the output and store into table. Iterate the table to fetch TF/IDF of top 10 terms for each of the top 10 users. We have chnaged Hadoop commands from source according to our file path and jar version. Mapper and Reducer programs remain unchanged due to logic being consistent The mapreduce programs are from

https://github.com/SatishUC15/TFIDF-HadoopMapReduce

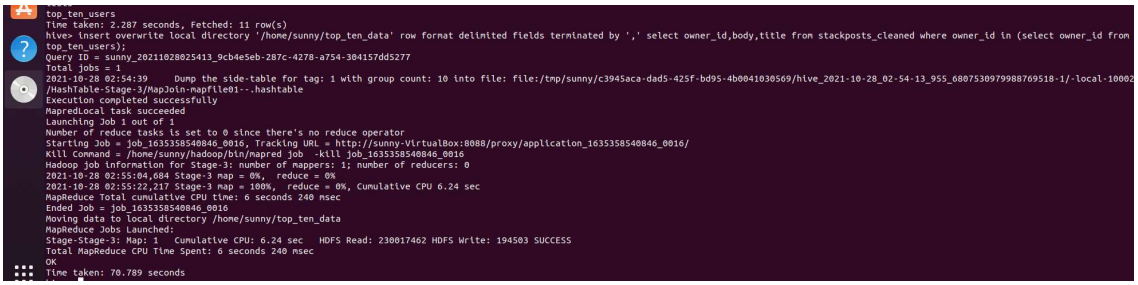

Figure 7:Dump whole data of top 10 users

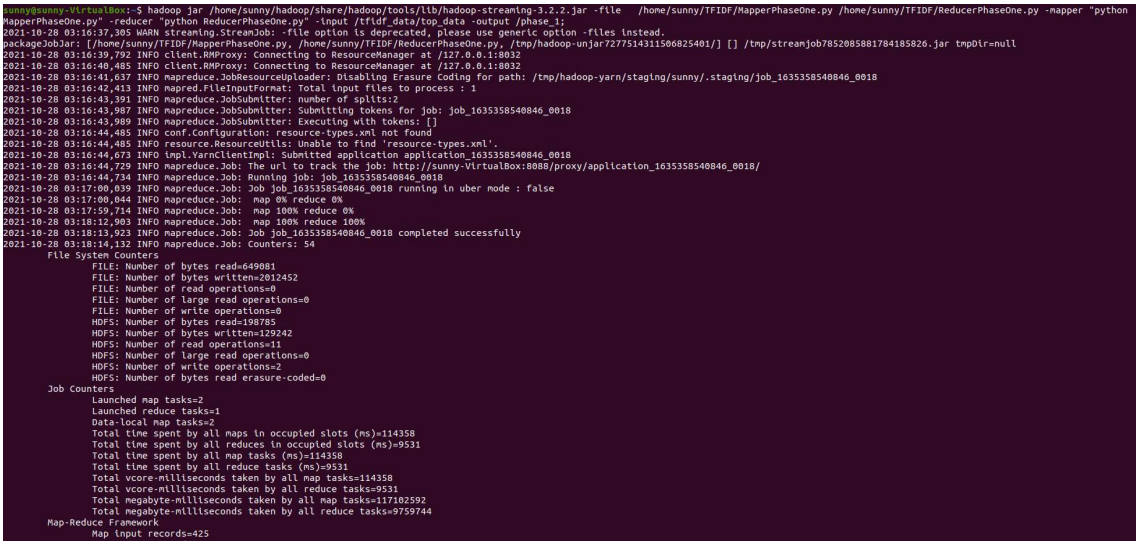

Figure 8:Mapper and Reducer

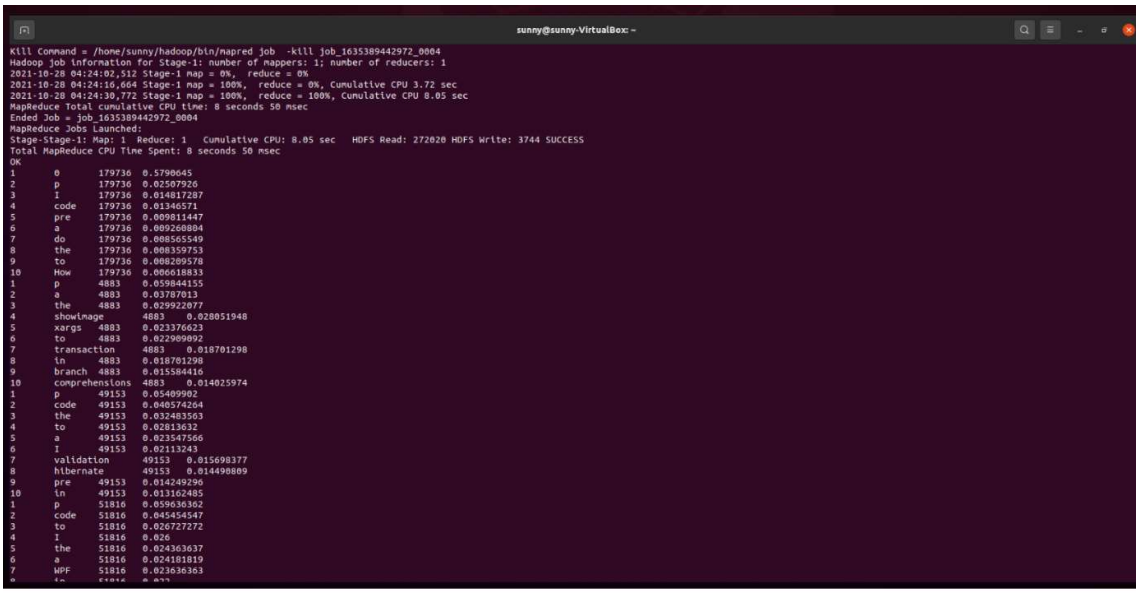

Figure 9:TFIDF

Appendix

Completion of Tasks

Tasks Completed

1. Data Acquisition
2. Data ETL
    i. Top 10 posts by score
    ii. Top 10 users by post score
    iii. The number of distinct users, who used the word “cloud” in one of their posts
3. Calculate the per-user TF-IDF of the top 10 terms for each of the top 10 users