**CA675: Cloud Technologies Assignment 1: Data Analysis**

Email: qinyanchen96@gmail.com

# **1. Acquire the top 200,000 posts by viewcount**

The original data are included in four CSV files, I put them into RAW\_DATA folder.

1. Get top 1 to 50000

Firstly, I try to find top 50000, I use this statement to see how many posts' ViewCount larger than 120000:

select count(\*) from posts where posts.ViewCount > 110000

Then I get 51382, it is close to 50000.

Secondly, I modify multiply times then I get exactly 50000 posts. Because I used 'ViewCount > 112209', so the result won't miss any records.

select count(\*) from posts where posts.ViewCount > 112209

Graphical user interface, text, application, email

Description automatically generated

Thirdly, change my query to get records. To make things easier later, I use a sort statement so that the records are sorted in reverse order by ViewCount:

select \* from posts where posts.ViewCount > 112209 order by ViewCount DESC

1. Get top 50001 to 100000

Use the same way to find 50001 to 100000:

### select count(\*) from posts where posts.ViewCount > 66056 and posts.ViewCount <= 112209

I got exactly 50000 records.

Get records:

### select \* from posts where posts.ViewCount > 66056 and posts.ViewCount <= 112209 order by ViewCount DESC

1. Get top 100001 to 150000

Repeat the above steps, first try to find the 150000th position, and then get the record.

select count(\*) from posts where posts.ViewCount > 47161 and posts.ViewCount <= 66056

select \* from posts where posts.ViewCount > 47161 and posts.ViewCount <= 66056 order by posts.ViewCount DESC

1. Get top 150001 to 200000

Because I just need 50000 records which meet 'ViewCount <=47161' and I can only download maximum 50000 records at a time, so I only need to find a number which let the records approximately close to 50000 but larger 500000, so I run this query:

select count(\*) from posts where posts.ViewCount > 36000 and posts.ViewCount <= 47161

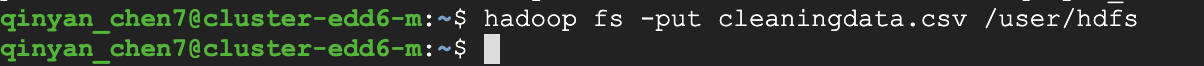
It returns 54323, which meets my need.

Then run this query and download 50000 records. Because I use DESC, so I won't miss any records or get wrong data in this step.

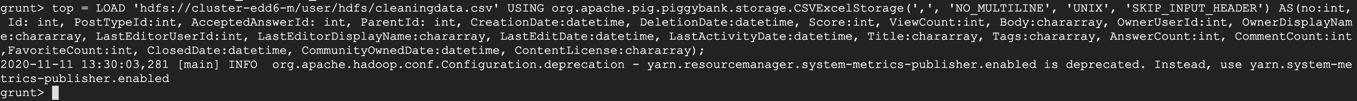
select \* from posts where posts.ViewCount > 36000 and posts.ViewCount <= 47161 order by ViewCount DESC

# **2. Using Pig or MapReduce, extract, transform and load the data as applicable**

1. Cleaning data in Python. The code is in raw\_data\_clean.py.
2. Extract, transform and load the data with Pig. Pig commands are included in ETL\_Pig.txt



# Load data



# Generate, transform data

Text

Description automatically generated

# Store data

Graphical user interface, text

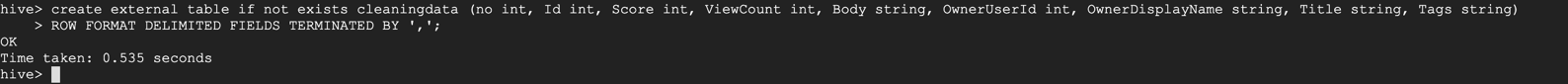
Description automatically generated

# **3. Using Hive and/or MapReduce, get:**

Hive commands are included in Hive\_cmd.txt

Create table and load data before creating queries.

# create table



# load data



Text

Description automatically generated

1. The top 10 posts by score

# Just included Title for clearly displaying.

Text

Description automatically generated

# Include all the information of 10 posts.

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

1. The top 10 users by post score

Text

Description automatically generated

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

Text

Description automatically generated

# **4. Using Mapreduce/Pig/Hive calculate the per-user TF-IDF (just submit the top 10 terms for each of the top 10 users from Query 3.II)**

I used Hive to complete this task. The code of this task is included in TFIDF\_Hive.txt.

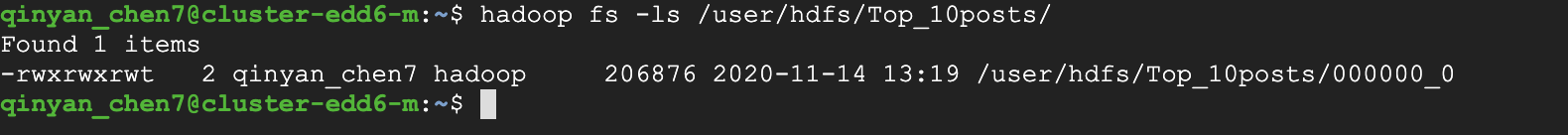
1. Use hive to create table ‘top\_10users’.

Text

Description automatically generated

1. Create table ‘top\_10posts’ to store all top ten users’ posts.
2. Store the result to HDFS.

# Then download top\_10posts, use python to clean data.



Text

Description automatically generated

1. Use python to clean Top\_10posts. The code is in CleanUserPosts.py. The input is top\_10\_posts.csv which is downloaded from hdfs in last step. The output file is cleaned\_posts.txt, upload it to hdfs.
2. Use Hive to calculate the per-user TF-IDF of the top 10 users.

qinyan\_chen7@cluster-edd6-m:~$ hadoop fs -put cleaned\_posts.txt /user/hdfs

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generatedText

Description automatically generated