Cloud Technologies - Assignement 1

Large Dataset analysis

Akash Bhargava- SN : 18210613

akash.bhargava2@mail.dcu.ie

(Dated: 07 March 2018)

The objective of this assignment is to work on a large amount of data, with the technologies see in class (MapReduce, Hive and Pig)

Tasks :

* 1. Acquire the top 200,000 posts by viewcount.
  2. Using pig or mapreduce, extract, transform and load the data as applicable
  3. Using hive and/or mapreduce, get:
  + The top 10 posts by score
  + The top 10 users by post score
  + The number of distinct users, who used the word hadoop in one of their posts
  1. Using mapreduce calculate the per-user TF-IDF (just submit the top 10 terms for each user)
  2. Bonus use elastic mapreduce to execute one or more of these tasks (if so, provide logs / screen-shots)

**INTRODUCTION**

The first step is to collect the data from Stack Exchange database (a network of question-and-answer websites on topics in varied fields) using SQL queries.

The database is available here:

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

We will work with Posts, all Posts have this structure:

Id - int, PostTypeId – tinyint, AcceptedAnswerId – int, ParentId – int, CreationDate – datetime, DeletionDate – datetime, Score – int, ViewCount – int, Body - nvarchar (max), OwnerUserId – int, OwnerDisplayName - nvarchar (40), LastEditorUserId – int, LastEditorDisplayName -nvarchar (40), LastEditDate – datetime, LastActivityDate – datetime, Title - nvarchar (250), Tags - nvarchar (250), AnswerCount – int, CommentCount – int, FavoriteCount – int, ClosedDate – datetime, CommunityOwnedDate – datetime

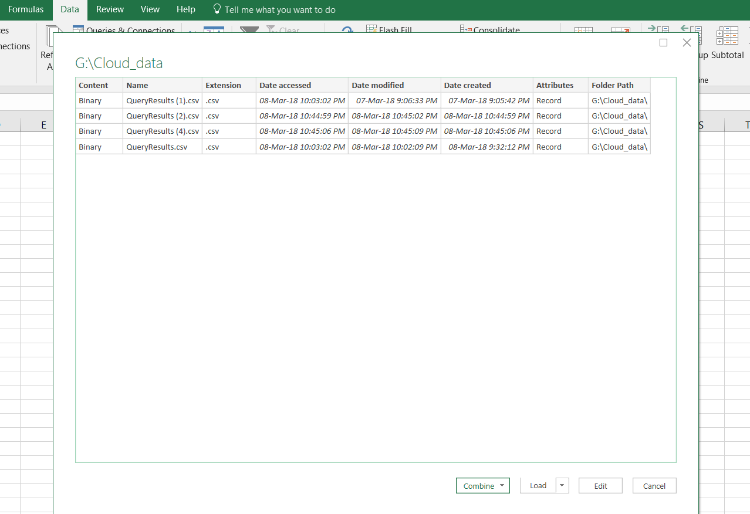
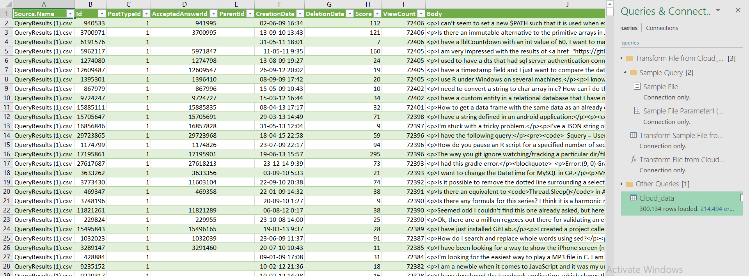
Step 1: **Create the file with the data**

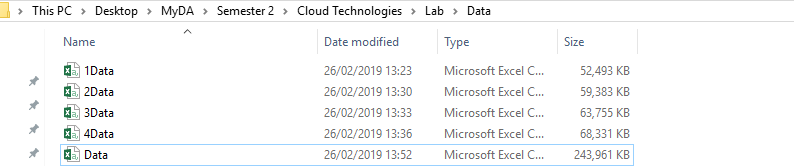
On StackExchange we can only dump 50K posts, so the idea is to dump the top 50K most viewed posts with this command below:

select count(\*) from posts where posts.ViewCount > 58000

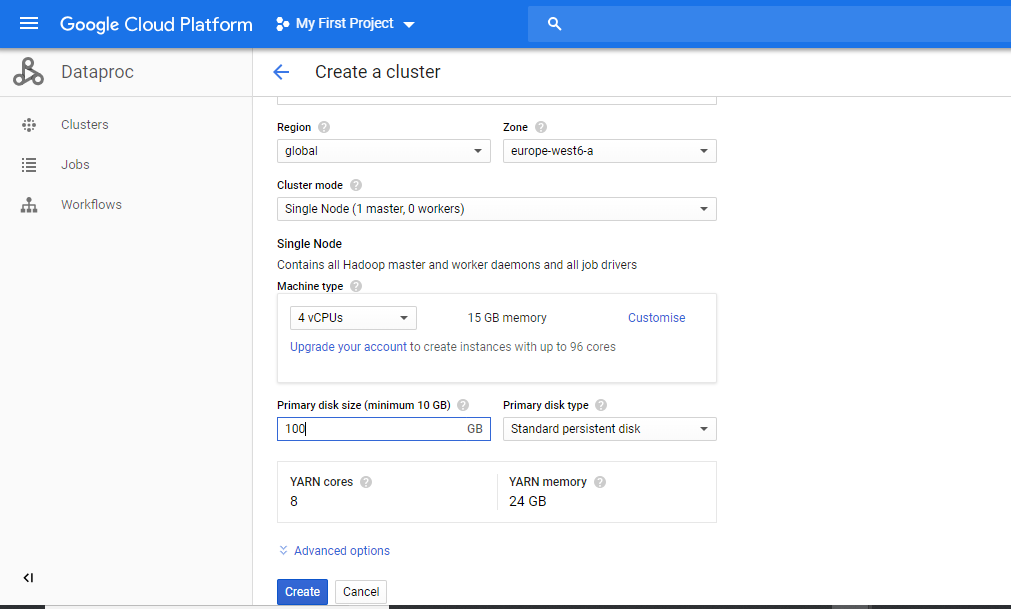
We change the View Count and run the query four times to get four files with 50K rows each.

The four individual datasets are merged into a single csv file through command line. Final File is called Data.csv

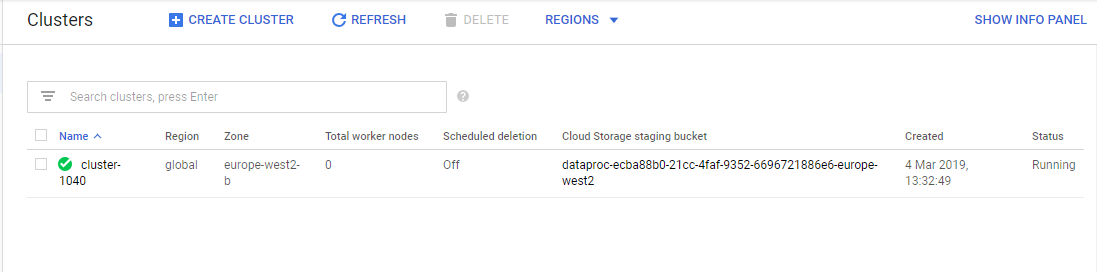
 

Step 2: **Understanding Google Cloud Platform and creating a cluster in Data Proc**

1. First, we create an account using Google Cloud Platform. After that the next task go to data proc and create a cluster. I created a cluster with 1 master and 0 nodes because of the requirement. I also decreased the size of Primary disk to 100 gb. After that we click on Create Cluster. This might take a while.

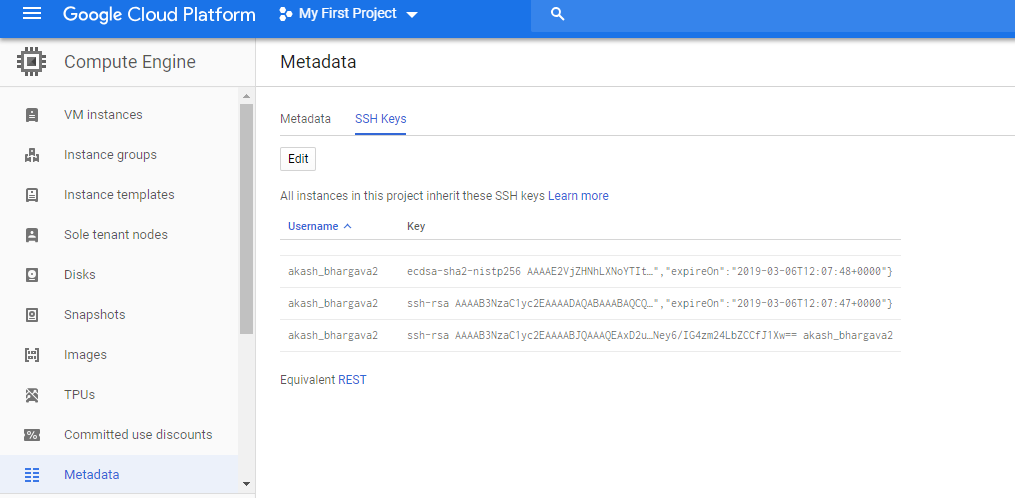
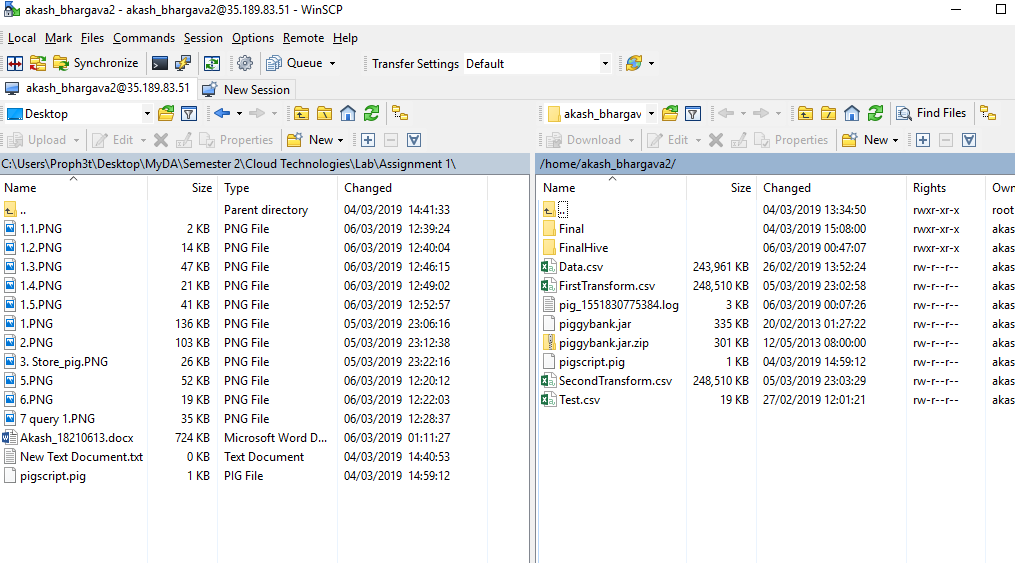


1. Once the cluster gets created it looks like this.

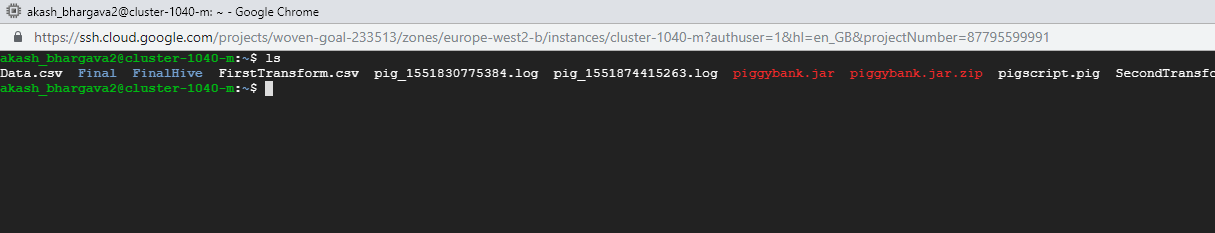


1. Next, I downloaded WinSCP to create a connection to the cluster. I created a .ppk file and added the contents in the metadata to create a connection.

For that click the cluster, then go to VM instances and then click on metadata. In that we go to SSH keys and add the contents of .ppk file. Now we can transfer file from WinSCP to the cluster. I added Data.csv file to the cluster.

1. Now we go back to the cluster and click on SSH. It opens a console and we are in the cluster. We already have the files. Now we can continue our preprocessing in Hadoop.



Step 3: DATA FORMATTING WITH PIG

To begin with data preprocessing, we use two commands which will help in cleaning the data from the csv file.

It adds a \n tab at the spaces. Below are the commands.

sed ':a;N;$!ba;s/\n/\\n/g' Data.csv > FirstTransform.csv

sed s/\\n//g <FirstTransform.csv>SecondTransform.csv

Now we have a file looking like below:

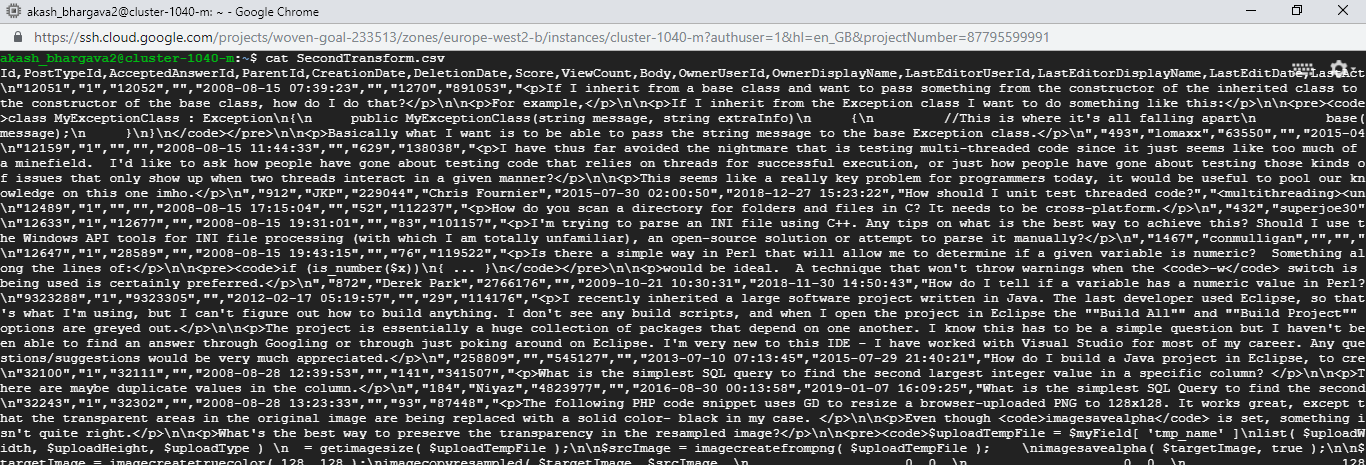


FIG. File preprocessed

After initial formatting of the file, we then put this file in the location of hdfs which could be achieved using the command.

>> hdfs dfs -put SecondTransform.csv /F5

In order to load the data with the separator presented, we can use CSVLoader instead of pigStorage. It’s a function available in the piggybank. See:  
  
https://cwiki.apache.org/confluence/display/PIG/PiggyBank

Next, we can write a simple script to start with:

REGISTER piggybank.jar;

DEFINE CSVLoader org.apache.pig.piggybank.storage.CSVLoader();

A = LOAD '/F5' USING CSVLoader(',') 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);

B = FOREACH A GENERATE Id, PostTypeId, Score, ViewCount, OwnerUserId, OwnerDisplayName, Title;

DUMP B;

We run this script and we get this:

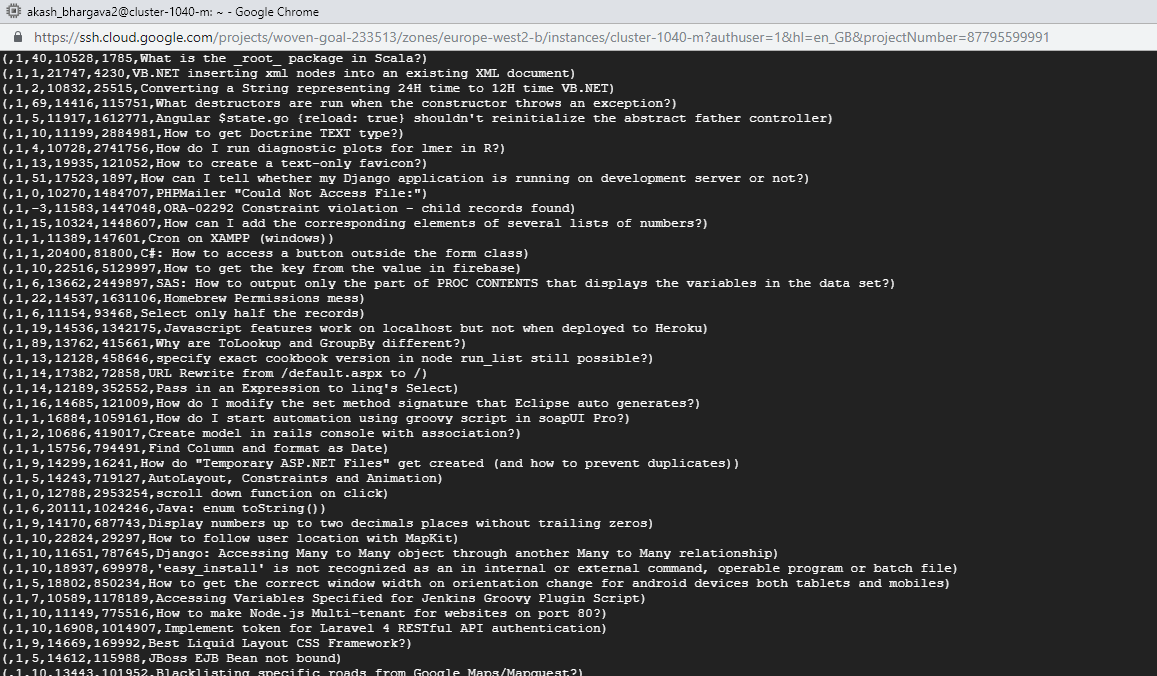
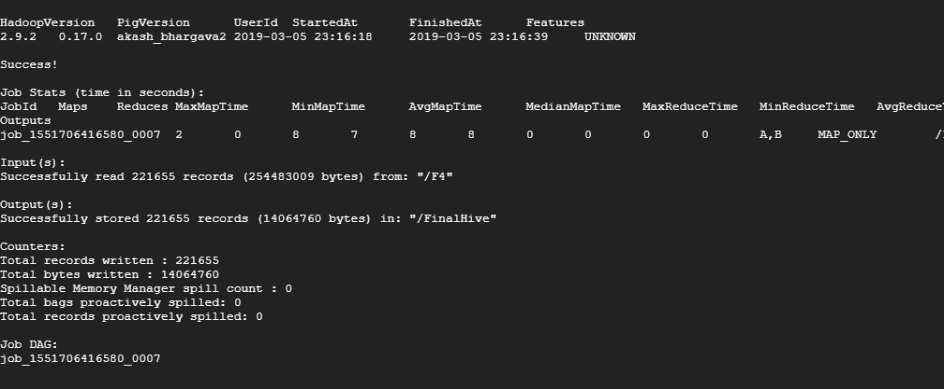


FIG. Result after pig process

­­As we can see, for each of the post we have specific record of ID, PostTypeId, Score, ViewCount, OwnerUserId and Title

The second step now is to export the dumped data in a readable format for Hive and for this we just have to replace the DUMP B by

STORE B INTO '/Final' USING PigStorage('\*');



After this update, we can see that Pig has divided the result in 2 files in the hdfs and we have a log file who block the load function of hive so we have to delete the \_SUCCESS le before continuing. ­­­

So just run this command to delete the log file:

>>hdfs dfs -rm /Final/\_SUCCESS

Step 4: DATA QUERYING WITH HIVE

For this first we must create a Database Schema and then tables. Then with the help of HCatalog I was able to store pig results in Hive tables.

In Hive Console:

hive> Create database pig;

hive> Create table stackposts (Id int, PostTypeId int, Score int, ViewCount int, OwnerUserId int, OwnerDisplayName string, Title string);

In pig -useHCatalog Console:

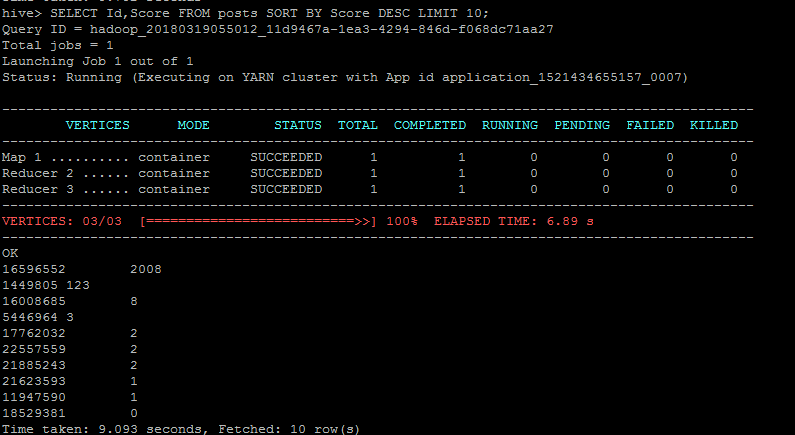
grunt> A = LOAD '/Final' using PigStorage('\*') AS (id:int,posttypeId:int, score:int, viewcount:int, owneruserId:int, ownerdisplayname:chararray, title:chararray);

grunt> STORE A INTO 'pig.stackposts' USING org.apache.hive.hcatalog.pig.HCatStorer;

Now we can write queries in Hive:

A. We select the top 10 posts sorted by score (I only selected the Score and Title in order to have an output readable but you can select \* if you want all the post)

hive> SELECT id,score,title FROM posts SORT BY score desc limit 10;

 DESC LIMIT 10;

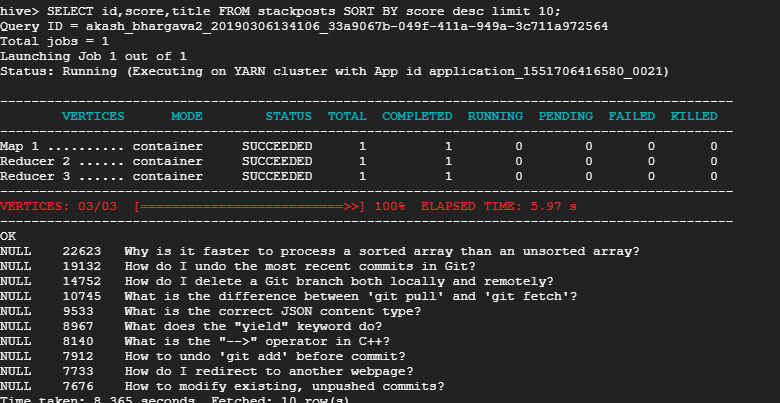
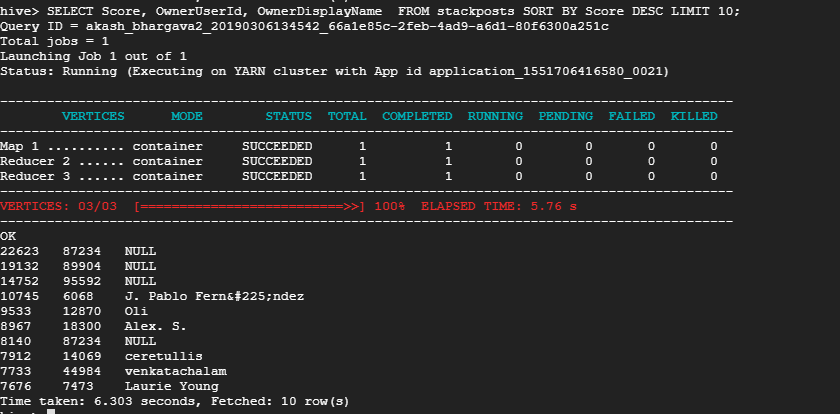


FIG. The top 10 posts by score

B. We select the top 10 UserName sort by score

hive> SELECT Score, OwnerUserId, OwnerDisplayName FROM stackposts SORT BY Score DESC LIMIT 10;



C. We select all the distincts user who use hadoop in the body or title of their post.

hive> SELECT Count(DISTINCT(OwnerUserId)) FROM stackposts WHERE (Body REGEXP 'hadoop') OR (Title REGEXP 'hadoop');

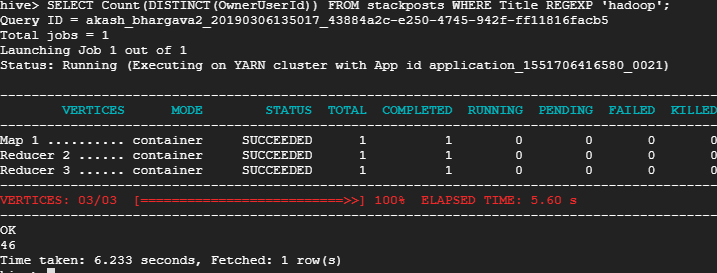


FIG. Distinct users, who used the word hadoop in a post

**Calculating the per-user TF-IDF (just submit the top 10 terms for each user):**

I have created a python script for calculating the TF-IDF and ran it locally on my system using Jupyter Notebooks application which gave me the below results. The source code of the same is attached in the submission.

