

2021/2022 CA675 Cloud Technologies

Assignment 1 - Data Analysis using Cloud Technologies

Romil Sakariya

Student ID: 21264095

GitLab project for this assignment: [CA-675 Assignment 1](#)

Email: romil.sakariya2@mail.dcu.ie

Step 1: Getting data from Stack Exchange (Data Acquisition/Collection)

After carefully analysing the tasks and going through database schema of StackExchange, I could deduce that the following columns are required to complete the assignment tasks:

1. id:int
2. Score:int
3. ViewCount:int
4. Body:chararray
5. OwnerUserId:int
6. OwnerDisplayName:chararray
7. Title:chararray
8. Tags:chararray

I ran the following query on the Data Explorer feature of the StackExchange system:

```
SELECT id, Score, ViewCount, Body, OwnerUserId, OwnerDisplayName, Title, Tags
FROM posts where posts.ViewCount>(upperlimit) AND
posts.ViewCount<=(lowerlimit)
```

The following table describes the values of lowerlimit and upperlimit I used in the above query and the different batch files I attained from them:

File Name	upperlimit value	lowerlimit value	Number of rows returned
QueryResults1.csv	1000000	No lowerlimit for first round	1506
QueryResults2.csv	125000	1000000	49955
QueryResults3.csv	75000	125000	48183
QueryResults4.csv	55000	75000	45298
QueryResults5.csv	43000	55000	46933
QueryResults6.csv	42000	43000	5180

All commands used to accomplish this step: [link](#)

The screenshot shows the StackExchange Data Explorer interface. At the top, there are navigation links for Home, Queries, and Users, along with a Compose Query button. The main section is titled "Editing Query" and contains a text input for a query title and a description. The query editor shows a SQL query: `1 Select id, Score, ViewCount, Body, OwnerUserId, OwnerDisplayName, Title, Tags from p`. To the right of the query editor is a "Database Schema" panel showing the structure of the "Posts" table, including fields like Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, and Body. Below the schema is a "Revisions" panel showing two revisions for the selected post. At the bottom of the interface is a "Results" panel displaying a table with 1506 rows. The table has columns for Id, Score, ViewCount, Body, OwnerUserId, OwnerDisplayName, Title, and Tags. The first few rows of the table are visible, showing posts with high view counts.

Id	Score	ViewCount	Body	OwnerUserId	OwnerDisplayName	Title	Tags
25969	1630	2899654	<p>I am trying to <code>INSERT INTO</code>...	244	Shadow_x99	Insert into ... values (SELECT ... FROM ...)	sql database syntax database-agnostic an
26551	1265	2095679	<p>I need to pass an ID and a password to a ...	730	Keng	How can I pass arguments to a batch file?	batch-file arguments
2972600	306	1393143	<p>Sometimes I get the following error while ...	358237		No connection could be made because the ba...	cf .net asp.net-web-api2 socketexception
2988017	581	1055908	<p>I noticed a Python script I was writing wa...	262271		String comparison in Python: is vs. ==	python string comparison equality
3480771	1091	1802702	<p>I have a shopping cart that displays prod...	420022		How do I check if string contains substring?	javascript query string substring contains
3481826	1803	4300719	<p>I have a string, <code>"004-034556"</code>...	303459		How to split a string in Java	java string split
3500197	470	1131459	<p>I have a slider that can be pulled up and t...	401025		How to display Toast in Android?	android android-mapview android-asynctask
3501382	1030	1611336	<p>How do I check whether a variable is an i...	221149		Checking whether a variable is an integer or ...	python integer
3508805	2467	1333888	<p>I am trying to make a <code><u>></code>...	383759		How can I transition height: 0; to height: auto...	css css-transitions
3514784	1821	1751065	<p>Is there a way to detect whether or not a ...	54259		What is the best way to detect a mobile device?	javascript query mobile browser-detection
3518002	1724	3070250	<p>I thought that adding a <code>"value"</code>...	193251		How can I set the default value for an HTML ...	html form-select
109918...	509	1308113	<p>How do I <code>auto increment</code> t...	808208		Auto increment primary key in SQL Server M...	sql-server sql-server-2012 identity
3548453	1133	1008421	<p>I have been trying to work out the syntax ...	97767		Negative matching using grep (match lines th...	regex grep
194913...	702	1464503	<p>I have seen lots of jQuery examples wher...	1002758		How to get URL parameter using jQuery or pl...	jquery url parameters query-string querystr
3552461	2840	5018250	<p>In JavaScript, how can I format a date obj...	4653		How to format a JavaScript date	javascript date date-formatting

Fig 1: StackExchange Data Explorer with query to extract posts having a viewcount greater than 1000000

Step 2: Data Cleaning using Excel and Pig

Once I had acquired all the above batch files, I had to clean them and combine them into one CSV file comprising of approximately 200000 posts sorted by their ViewCount.

To clean the data, I initially used Microsoft Excel. Since the batch files were reasonably small, I could open them in Excel comfortably and view the schema, analyse the type of data and perform minor cleansing operations such as getting rid of any empty or duplicate entries and special characters from the content in the body column.

Now in order to combine all these batch files and cleanse the data even further, I used Apache Pig.

I used Google Cloud Platform's Dataproc service to create a Hadoop cluster of VMs with one master node and two datanodes. I uploaded the batch files to the master VM and then transferred them to the HDFS system of the GCP cluster.

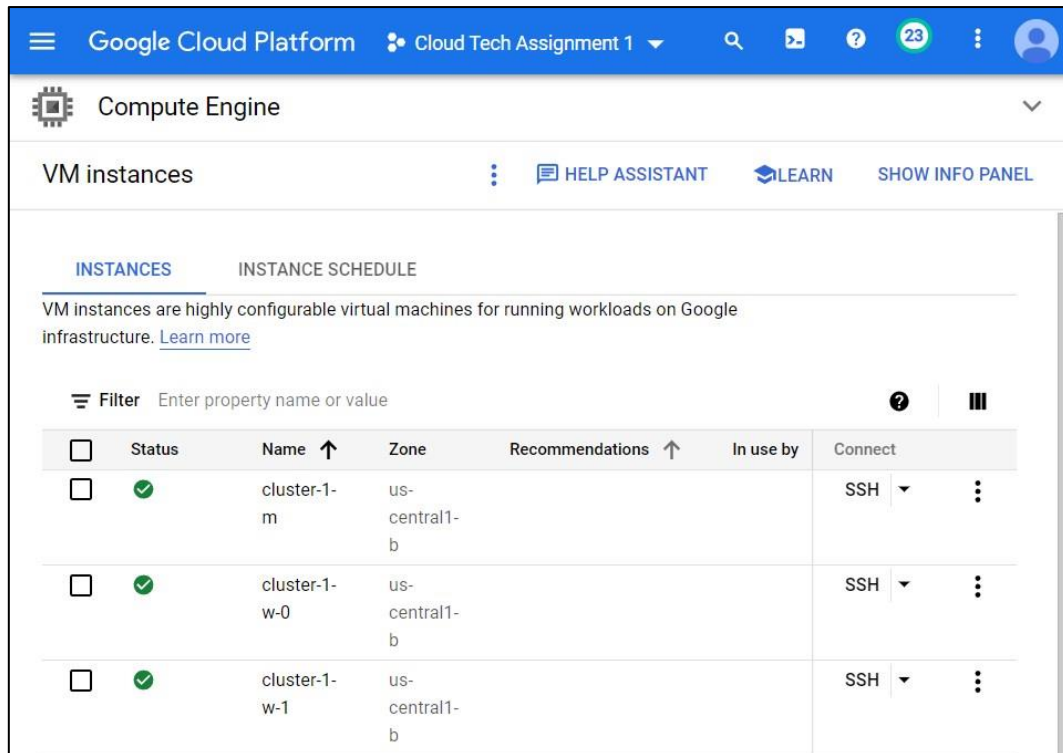


Fig 2: GCP Dataproc Cluster with all the nodes listed

```
romil_sakariya2@cluster-1-m:~$ ls
test.csv
romil_sakariya2@cluster-1-m:~$ ls
QueryResults1.csv QueryResults3.csv QueryResults5.csv test.csv
QueryResults2.csv QueryResults4.csv QueryResults6.csv
romil_sakariya2@cluster-1-m:~$ hdfs dfs -ls /
Found 3 items
drwxr-xr-x - romil_sakariya2 hadoop 0 2021-10-25 18:01 /assignment
drwxrwxrwt - hdfs hadoop 0 2021-10-25 15:21 /tmp
drwxrwxrwt - hdfs hadoop 0 2021-10-25 15:20 /user
romil_sakariya2@cluster-1-m:~$ hdfs dfs -ls /assignment
Found 1 items
-rw-r--r-- 2 romil_sakariya2 hadoop 43 2021-10-25 18:01 /assignment/test.csv
romil_sakariya2@cluster-1-m:~$ hdfs dfs -put QueryResults1.csv /assignment
romil_sakariya2@cluster-1-m:~$ hdfs dfs -put QueryResults2.csv /assignment
romil_sakariya2@cluster-1-m:~$ hdfs dfs -put QueryResults3.csv /assignment
romil_sakariya2@cluster-1-m:~$ hdfs dfs -put QueryResults4.csv /assignment
romil_sakariya2@cluster-1-m:~$ hdfs dfs -put QueryResults5.csv /assignment
romil_sakariya2@cluster-1-m:~$ hdfs dfs -put QueryResults6.csv /assignment
romil_sakariya2@cluster-1-m:~$ hdfs dfs -ls /assignment
Found 7 items
-rw-r--r-- 2 romil_sakariya2 hadoop 1084982 2021-10-25 23:37 /assignment/QueryResults1.csv
-rw-r--r-- 2 romil_sakariya2 hadoop 48788602 2021-10-25 23:37 /assignment/QueryResults2.csv
-rw-r--r-- 2 romil_sakariya2 hadoop 53840533 2021-10-25 23:37 /assignment/QueryResults3.csv
-rw-r--r-- 2 romil_sakariya2 hadoop 53177616 2021-10-25 23:37 /assignment/QueryResults4.csv
-rw-r--r-- 2 romil_sakariya2 hadoop 58429121 2021-10-25 23:37 /assignment/QueryResults5.csv
-rw-r--r-- 2 romil_sakariya2 hadoop 6476415 2021-10-25 23:37 /assignment/QueryResults6.csv
-rw-r--r-- 2 romil_sakariya2 hadoop 43 2021-10-25 18:01 /assignment/test.csv
romil_sakariya2@cluster-1-m:~$
```

Fig 3: Putting batch files in HDFS from local cloud storage

Once all batch files were in the HDFS system of my cluster, I used the following pig commands to load, transform and extract data:

```
A = LOAD 'QueryResults1.csv' USING PigStorage(',') AS (Id:int, Score:int,
ViewCount:int, Body:chararray, OwnerUserId:int, OwnerDisplayName:chararray,
Title:chararray, Tags:chararray);

a = FILTER A BY Id>1;
```

```

*****

F = LOAD 'QueryResults6.csv' USING PigStorage(',') AS (Id:int, Score:int,
ViewCount:int, Body:chararray, OwnerUserId:int, OwnerDisplayName:chararray,
Title:chararray, Tags:chararray);

f = FILTER F BY Id>1;

G = UNION a,b,c,d,e,f;

H = DISTINCT G;

STORE H INTO '/assignment/FinalData' USING PigStorage(',');

hdfs dfs -get /assignment/FinalData/part-r-00000 /home/romil_sakariya2

```

I used Pig to clean my data owing to its ability to handle unstructured data and quick computation time. At this stage using Hive would have been counterproductive since the data was still in a very raw unstructured format.

```

her.enabled
grunt> A = LOAD 'QueryResults1.csv' USING PigStorage(',') AS (Id:int, Score:int, ViewCount:int, B
ody:chararray, OwnerUserId:int, OwnerDisplayName:chararray, Title:chararray, Tags:chararray);
2021-10-26 23:41:04,367 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resou
rcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publis
her.enabled
grunt> a = FILTER A BY Id>1;
grunt> B = LOAD 'QueryResults2.csv' USING PigStorage(',') AS (Id:int, Score:int, ViewCount:int, B
ody:chararray, OwnerUserId:int, OwnerDisplayName:chararray, Title:chararray, Tags:chararray);
2021-10-26 23:41:21,153 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resou
rcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publis
her.enabled
grunt> b = FILTER B BY ID>1;
2021-10-26 23:41:21,212 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 1025: <line 4, colum
n 16> Invalid field projection. Projected field [ID] does not exist in schema: Id:int,Score:int,Vi
ewCount:int,Body:chararray,OwnerUserId:int,OwnerDisplayName:chararray,Title:chararray,Tags:chararr
ay.
Details at logfile: /home/romil_sakariya2/pig_1635291644451.log
grunt> b = FILTER B BY Id>1;
grunt> C = LOAD 'QueryResults3.csv' USING PigStorage(',') AS (Id:int, Score:int, ViewCount:int, B
ody:chararray, OwnerUserId:int, OwnerDisplayName:chararray, Title:chararray, Tags:chararray);
2021-10-26 23:42:06,917 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resou
rcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publis
her.enabled
grunt> c = FILTER C BY Id>1;
grunt> D = LOAD 'QueryResults4.csv' USING PigStorage(',') AS (Id:int, Score:int, ViewCount:int, B
ody:chararray, OwnerUserId:int, OwnerDisplayName:chararray, Title:chararray, Tags:chararray);
2021-10-26 23:42:28,044 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resou
rcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publis
her.enabled
grunt> d = FILTER D BY Id>1;
grunt> E = LOAD 'QueryResults5.csv' USING PigStorage(',') AS (Id:int, Score:int, ViewCount:int, B
ody:chararray, OwnerUserId:int, OwnerDisplayName:chararray, Title:chararray, Tags:chararray);
2021-10-26 23:42:45,298 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resou
rcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publis
her.enabled
grunt> e = FILTER E BY Id>1;
grunt> e = FILTER E BY Id>1;
grunt> F = LOAD 'QueryResults6.csv' USING PigStorage(',') AS (Id:int, Score:int, ViewCount:int, B
ody:chararray, OwnerUserId:int, OwnerDisplayName:chararray, Title:chararray, Tags:chararray);
2021-10-26 23:43:05,556 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resou
rcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publis
her.enabled
grunt> f = FILTER F BY Id>1;
grunt> f = FILTER F BY Id>1;
grunt> G = UNION a,b,c,d,e,f;
grunt> H = DISTINCT G;

```

Fig 4: Loading and cleaning data in Pig

All commands used to accomplish this task: [link](#)

Step 3: Fetching the top 10 posts by score using Pig

After obtaining the complete data set from Pig, I used the same relation (H) to fetch the top 10 posts by score using the following Pig commands:

```
sorted_by_score = ORDER H BY Score DESC;  
task1 = LIMIT sorted_by_score 10;
```

I used Pig to complete this task because all the required cleaned data was loaded in relation H. Using just a sort command I was able to sort the entire data set in descending order of score. I could have also done the same using Hive but that would require explicitly creating data tables and loading data again.

All commands used to accomplish this task: [link](#)

Output File: [Task2.2.1](#)

Step 4: Fetching top 10 users by total post score

Now that I have the complete **structured** data set, I loaded it into a Hive table:

```
CREATE EXTERNAL TABLE finaldata (id INT, score INT, viewcount INT, body  
STRING, owneruserid INT, ownerdisplayname STRING, title STRING, tags STRING)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';  
  
LOAD DATA LOCAL INPATH '/home/romil_sakariya2/finaloutput.csv' INTO TABLE  
finaldata;
```

Now using Hive table functions like SUM() and partitions I could easily calculate cumulative sum of post scores of a particular user in the following manner:

```
CREATE TABLE users1 AS SELECT id, score, owneruserid, ownerdisplayname FROM  
finaldata WHERE owneruserid IS NOT NULL;  
  
CREATE TABLE userspostsum AS SELECT id, score, owneruserid, ownerdisplayname,  
SUM(score) OVER (PARTITION BY owneruserid ORDER BY owneruserid ROWS BETWEEN  
UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) FROM users1;
```

I ranked the rows further in a partition of ownerID and added the first row of each partition into a new table (called tempunique1). Now simply sorting and limiting this table gave me the required result.

```
CREATE TABLE temp AS SELECT id, score, owneruserid, ownerdisplayname,  
sum_window_0, row_number() OVER (PARTITION BY owneruserid) AS rownumber FROM  
userspostsum;  
  
CREATE TABLE tempunique1 AS SELECT id, score, owneruserid, ownerdisplayname,  
sum_window_0 asscoresum FROM temp WHERE rownumber=1;  
  
CREATE TABLE task_2_2 AS SELECT id, score, owneruserid, ownerdisplayname,  
asscoresum totalscoresum FROM temp ORDER BY totalscoresum DESC LIMIT 10;
```

All commands used to accomplish this task: [link](#)

Output File: [Task2.2.2](#)

```

hive> SET hive.cli.print.header=true;
hive> select * from task_2_2;
Query ID = romil_sakariya2_20211027124843_fcf82658-8aaa-42fa-a6d1-9212f095ba4d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635327592353_0013)
-----
VERTICES      MODE           STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container      SUCCEEDED      1          1          0          0          0          0
-----
VERTICES: 01/01 [=====>>>] 100% ELAPSED TIME: 3.98 s
-----
OK
task_2_2.id      task_2_2.score  task_2_2.owneruserid  task_2_2.ownerdisplayname  task_2_2.t
otalsscoresum
3279543 2179      87234      37672
54867 1062      4883      28817
6650215 221      9951      26799
1116465 246      6068      25919
933329 190      89904      24024
807880 111      51816      23698
733219 56      49153      20184
9746303 117      179736      19530
4324558 247      95592      19479
1357945 66      63051      19345
Time taken: 4.761 seconds, Fetched: 10 row(s)
hive>

```

Fig 5: Result of top 10 users by total post score with their total post scores

Step 5: Fetching the total number of distinct users who used the word “cloud” in one of their posts

To make sure all the instances of the word “cloud” are covered, I first converted all the text in the body, title and tags column to lower case using the following commands:

```

CREATE TABLE lowercased_finaldata (id INT, score INT, viewcount INT, body
STRING, owneruserid INT, ownerdisplayname STRING, title STRING, tags STRING);

INSERT INTO lowercased_finaldata (id, score, viewcount, body, owneruserid,
ownerdisplayname, title, tags) SELECT id, score, viewcount, LOWER(body),
owneruserid, ownerdisplayname, LOWER (title), LOWER(tags) from finaldata;

```

Once the all the posts’ title, body and tags were converted into lower case, I began looking for all possible occurrences of the word “CLOUD” in them. I loaded all these occurrences in a separate table. Counting the distinct number of owneruserids in this table will give the total number of distinct users who used the word “cloud” in one of their posts.

```

CREATE TABLE cloud AS SELECT * FROM lowercased_finaldata WHERE
body||title||tags RLIKE 'cloud';

INSERT INTO cloud SELECT * FROM lowercased_finaldata WHERE body||title||tags
RLIKE '-cloud-';

.....

INSERT INTO cloud SELECT * FROM lowercased_finaldata WHERE body||title||tags
RLIKE 'cloud-';

SELECT COUNT(DISTINCT owneruserid) FROM cloud;

```

```

hive> INSERT INTO task2_2_3 SELECT DISTINCT * FROM cloud;
Query ID = romil_sakariya2_20211028205241_a030ff43-6613-4310-b69a-911cd3e13d54
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1635452261735_0006)

```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	container	SUCCEEDED	1	1	0	0	0	0	0
Reducer 2	container	SUCCEEDED	1	1	0	0	0	0	0
Reducer 3	container	SUCCEEDED	1	1	0	0	0	0	0

```

VERTICES: 03/03 [=====>>] 100% ELAPSED TIME: 6.62 s

```

```

Loading data to table default.task2_2_3
OK
Time taken: 14.316 seconds
hive> SELECT COUNT(DISTINCT owneruserid) FROM task2_2_3;
Query ID = romil_sakariya2_20211028205318_e9252f33-bc86-428b-8d83-dbcf6bffa8262
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635452261735_0006)

```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	container	SUCCEEDED	1	1	0	0	0	0	0
Reducer 2	container	SUCCEEDED	1	1	0	0	0	0	0
Reducer 3	container	SUCCEEDED	1	1	0	0	0	0	0

```

VERTICES: 03/03 [=====>>] 100% ELAPSED TIME: 5.03 s

```

```

OK
862
Time taken: 5.971 seconds, Fetched: 1 row(s)
hive> █

```

Fig 6: Output for task 2.2.3

The total number of distinct users who used the word “cloud” in one of their posts is 862.

All commands used to accomplish this task: [link](#)

Output File: [Task2.2.3](#)