CA675 Cloud Technologies

Name	Paritosh Gupta
Student Number	18210686
Programme	MCM – DA
Module Code	CA675
Assignment Title	Cloud Technology Assignment 1
Submission date	09th March 2019
Module coordinator	Alessandra Mileo

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the referencing guidelines found recommended in the assignment guidelines.

Name: Paritosh Gupta Date: 09th March 2019

Task 1 - Data Extraction:

The top 200,000 posts as per view count has been extracted from stackexchange portal. As only 50,000 records can be downloaded in single query run, the execution was performed in 4 batches. Below are the screenshots for the query execution on each batch.

Batch1-

```
select p.Id,
p.Score,
p.ViewCount,
p.Body,
p.Title,
p.OwnerUserId,
u.DisplayName 'UserName'
from Posts p
left outer join Users u on p.OwnerUserId=u.Id
where p.ViewCount > 50000
order by ViewCount desc
```

Batch2-

```
select p.Id,
p.Score,
p.ViewCount,
p.Body,
p.Title,
p.OwnerUserId,
u.DisplayName 'UserName'
from Posts p
left outer join Users u on p.OwnerUserId=u.Id
where p.ViewCount <= 86658 and p.Id not in (37745051) and  p.ViewCount > 20000
order by ViewCount desc
```

Batch3-

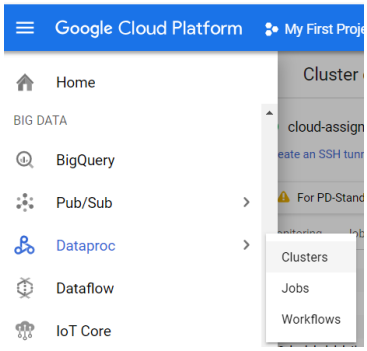
```
select p.Id,
p.Score,
p.ViewCount,
p.Body,
p.Title,
p.OwnerUserId,
u.DisplayName 'UserName'
from Posts p
left outer join Users u on p.OwnerUserId=u.Id
where p.ViewCount <= 51008 and p.Id not in (26881838,
3551966,
8881713,
34454081) and  p.ViewCount > 10000
order by ViewCount desc
```

Batch4-

```
select p.Id,
p.Score,
p.ViewCount,
p.Body,
p.Title,
p.OwnerUserId,
u.DisplayName 'UserName'
from Posts p
left outer join Users u on p.OwnerUserId=u.Id
where p.ViewCount <= 36583 and p.Id not in (4987476,
253834,
2050174,
8339529,
5973811,
19739507) and  p.ViewCount > 5000
order by ViewCount desc
```

Cluster creation in Google Dataproc:

- Create account in Google Cloud Platform and once created, go to console section.
- On left Tab pane, go to BIG DATA → Dataproc → Clusters . Then Enable API and click on 'Create Cluster'



- Below configuration details need to be filled to create single node cluster.

Monitoring | Jobs | VM Instances | Configuration

Name	cloud-assignment
Region	global
Zone	europe-west3-a
Scheduled deletion	Off
Master node	Single Node (1 master, 0 workers)
Machine type	n1-standard-4 (4 vCPU, 15.0 GB memory)
Primary disk type	pd-standard
Primary disk size	500 GB
Primary disk size	GB
Cloud Storage staging bucket	dataproc-1d859455-857a-4329-95d6-19923c3c6abf-europe-west3
Subnetwork	default
Network tags	None
Internal IP only	No
Image version	1.3.27-deb9
Created	Mar 6, 2019, 7:23:32 PM
Properties	Show properties

- Create the Firewall rule from **VPC Network** > **Firewall rules.**

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network
default-allow-dataproc-access	Ingress	Apply to all	IP ranges: 109.255.24.95/32	tcp:8088,9870,8080	Allow	1000	default

- Copy the query extracted files in Google cloud storage by drag and drop.
- Files fetched from google bucket to VM as below. Creating directory in Hadoop file system and further copying these files from VM to hdfs.

```
paritoshg2010@cloud-assignment1-m:~/assignment1$ gsutil -m cp gs://dataproc-1d859455-857a-4329-95d6-19923c3c6abf-eu
rope-west3/query-data/* .
Copying gs://dataproc-1d859455-857a-4329-95d6-19923c3c6abf-europe-west3/query-data/QueryResults (1).csv...
Copying gs://dataproc-1d859455-857a-4329-95d6-19923c3c6abf-europe-west3/query-data/QueryResults (2).csv...
Copying gs://dataproc-1d859455-857a-4329-95d6-19923c3c6abf-europe-west3/query-data/QueryResults (3).csv...
Copying gs://dataproc-1d859455-857a-4329-95d6-19923c3c6abf-europe-west3/query-data/QueryResults.csv...
```

```
paritoshg2010@cloud-assignment1-m:~/assignment1$ hadoop fs -mkdir /cloud
paritoshg2010@cloud-assignment1-m:~/assignment1$ hadoop fs -mkdir /cloud/assignment1
paritoshg2010@cloud-assignment-m:~/cloud$ hdfs dfs -put QueryResults*.csv /cloud/assignment1
```

Task 2 - PIG ETL:

Using pig or mapreduce, extract, transform & load the data as applicable

- To load the data in pig variable, UDF from piggybank has been used to fetch the csv data with multiline for a row, skipping the header.
- Sequence of pig commands has been executed to clean the data, such as removing html tags, new line and all type of special character from title & body (post) column.
- Once the data is cleaned, data has been stored with comma delimiter in output file.

```
grunt> A = LOAD '/cloud/assignment1/*' using org.apache.pig.piggybank.storage.CSVExcelStorage(',','YES_MULTILINE','
NOCHANGE','SKIP_INPUT_HEADER') as (id, score, viewcount, body, title, owneruserid,  displayname);
2019-03-08 00:45:57,134 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system
-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
grunt> B = FOREACH A GENERATE id, score, viewcount,REPLACE (body, '\\n|\\r|\\t|<br>',' ') as body_mod,  REPLACE(tit
le,'\\n|\\r|\\t|<br>',' ') as title_mod, owneruserid, displayname;
2019-03-08 00:45:57,452 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO
_CHARARRAY 2 time(s).
2019-03-08 00:45:57,453 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning USING_OVERLOADED
_FUNCTION 2 time(s).
grunt> C = FOREACH B GENERATE id, score, viewcount,REPLACE (body_mod, '<[^>]*>','') as body_mod,  REPLACE(title_mod
,'<[^>]*>','') as title_mod, owneruserid, displayname;
2019-03-08 00:45:57,537 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO
_CHARARRAY 2 time(s).
2019-03-08 00:45:57,537 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning USING_OVERLOADED
_FUNCTION 2 time(s).
grunt> D = FOREACH C GENERATE id, score, viewcount,REPLACE (body_mod, '([^a-zA-Z\\s]+)',' ') as body,  REPLACE(titl
e_mod,'([^a-zA-Z\\s]+)',' ') as title, owneruserid, displayname;
2019-03-08 00:45:57,619 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO
_CHARARRAY 2 time(s).
2019-03-08 00:45:57,619 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning USING_OVERLOADED
_FUNCTION 2 time(s).
grunt>
grunt> Store D into '/cloud/assignment1/resultset' USING PigStorage(',');
```

Task 3 - HIVE Queries:

- Create table in Hive.

CREATE TABLE userposts (id BIGINT ,score BIGINT, viewCount BIGINT, body STRING, title STRING, owneruserid BIGINT, ownerdisplayname STRING) row format delimited fields terminated by ',';

- Load the extracted data from pig queries into hive table.

LOAD DATA INPATH '/cloud/assignment1/resultset/part*' INTO TABLE userposts;

```
hive> CREATE TABLE userposts (id BIGINT ,score BIGINT, viewCount BIGINT, body STRING, title STRING, owneruserid BIGINT, ownerdisplayname STRING) row for
OK
Time taken: 0.803 seconds
hive> LOAD DATA INPATH '/cloud/assignment1/resultset/part*' INTO TABLE userposts;
Loading data to table default.userposts
OK
Time taken: 0.992 seconds
```

3.1- The top 10 post by score.

select id, score, title from userposts order by score desc limit 10;

```
hive> select id, score, title from userposts order by score desc limit 10;
Query ID = paritoshg2010_20190307124115_e6fac561-5915-4bdc-bf81-f0e4b9549096
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1551900259274_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.52 s
--------------------------------------------------------------------------------
OK
id      score   title
11227809        22623   Why is it faster to process a sorted array than an unsorted array
927358  19132   How do I undo the most recent commits in Git
2003505 14752   How do I delete a Git branch both locally and remotely
292357  10745   What is the difference between  git pull  and  git fetch
477816  9533    What is the correct JSON content type
231767  8967    What does the  yield  keyword do
1642028 8140    What is the   operator in C
348170  7912    How to undo  git add  before commit
503093  7733    How do I redirect to another webpage
179123  7676    How to modify existing  unpushed commits
Time taken: 12.139 seconds, Fetched: 10 row(s)
```

3.2- The top 10 users by post score- The top 10 users of the post with highest score has been selected.

select owneruserid, ownerdisplayname, score from userposts order by score desc limit 10;

```
hive> select id, score, ownerdisplayname from userposts order by score desc limit 10;
Query ID = paritoshg2010_20190307124530_ee63566e-1b80-4fca-88f9-b7e9d7622971
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1551900259274_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.51 s
--------------------------------------------------------------------------------
OK
id      score   ownerdisplayname
11227809        22623   GManNickG
927358  19132   Hamza Yerlikaya
2003505 14752   Matthew Rankin
292357  10745   pupeno
477816  9533    Oli
231767  8967    Alex. S.
1642028 8140    GManNickG
348170  7912    paxos1977
503093  7733    venkatachalam
179123  7676    Laurie Young
Time taken: 12.076 seconds, Fetched: 10 row(s)
```

3.3- The number of distinct users, who used the word ‘hadoop’ in one of their posts.

`select count(distinct owneruserid) from userposts where body like '%hadoop%'`

```
hive> select count(distinct owneruserid) from userposts where body like '%hadoop%';
Query ID = paritoshg2010_20190307125009_b02b21bb-00d9-4691-ad71-8cdfdb4abdd3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1551900259274_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 12.12 s
--------------------------------------------------------------------------------
OK
_c0
170
Time taken: 13.248 seconds, Fetched: 1 row(s)
```

Task 4 – TFIDF Calculation:

Map reduce paradigm was used to find TFIDF for the posts of top 10 users by score. Below are the sequence of commands were executed to create the temporary macro and views for calculation of term frequency and inverse document frequency for identified users. Below is the screenshot for calculated TFIDF for some of the users.

```
hive> select * from tfidf;
Query ID = paritoshg2010_20190308221556_b59403b6-3fd2-4b9f-9440-6fc72de9d54c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1552000308997_0013)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 4 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 5 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 8.26 s
--------------------------------------------------------------------------------
OK
95592   branch  0.2702702581882477
7473    commit  0.26874331466496004
95592   bugfix  0.2432432472705841
7473    message 0.23529411852359772
44984   jquery  0.2222222238779068
44984   pure    0.2222222238779068
44984   redirect        0.2222222238779068
44984   user    0.2222222238779068
44984   page    0.2222222238779068
95592   origin  0.21621622145175934
44984   javascript      0.18877444633270352
44984   another 0.18877444633270352
44984   using   0.18877444633270352
7473    files   0.17916220599452976
```