Shradha Shivani (21261104)
Date: 28th October 2021

In this document the top 200000 post by viewcount data set is extracted from StackExchange Posts table and further cleaned and processed to obtain query results.

Data Extraction:

Link: https://data.stackexchange.com/meta.codereview/query/new

Queries ran on the online terminal of StackExchange:

select top 50000 * from Posts where ViewCount > 100000 order by ViewCount desc;

select top 50000 * from Posts where ViewCount >= 65000 and ViewCount <= 100000 order by ViewCount desc;

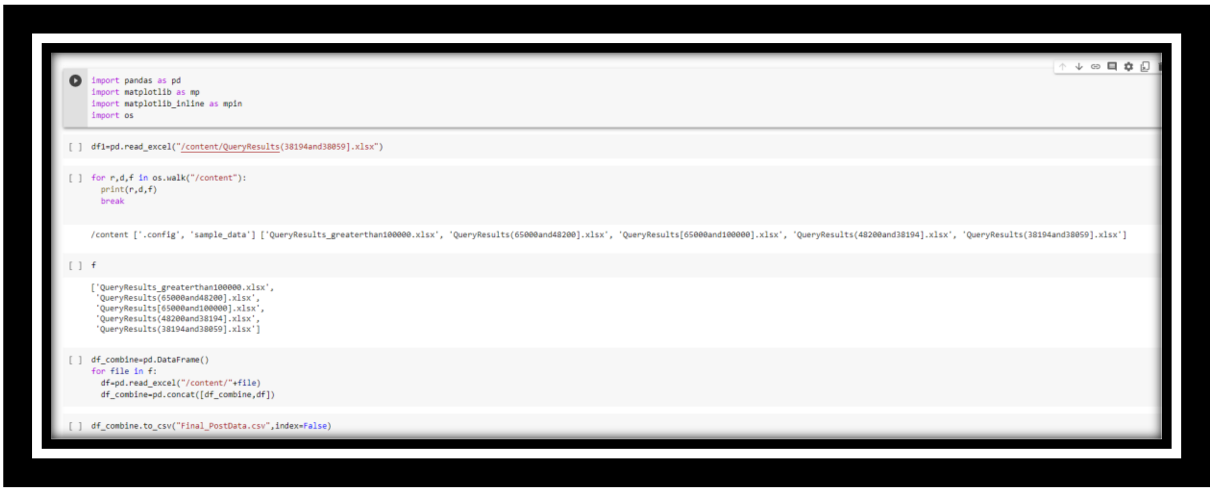
select top 50000 * from Posts where ViewCount >= 48200 and ViewCount < 65000 order by ViewCount desc;

select top 50000 * from Posts where ViewCount >= 38194 and ViewCount < 48200 order by ViewCount desc;

select top 50000 * from Posts where ViewCount >= 38059 and ViewCount < 38194 order by ViewCount desc;

Few extra records were captured; hence those were deleted manually from the last dataset collected.

The files were merged into a single dataset Final_PostData.csv using Google Colab JupyterNotebook:

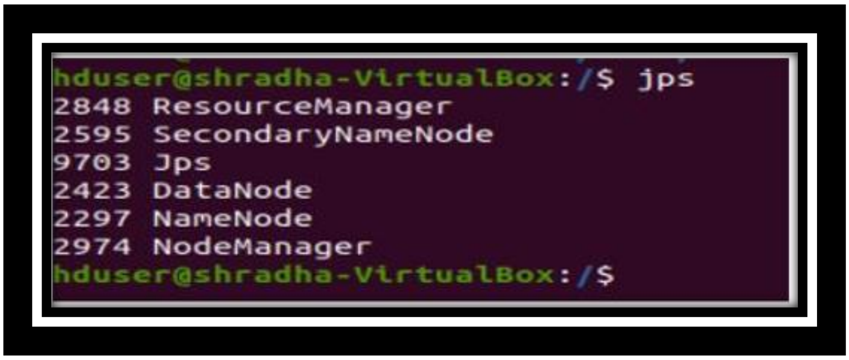


```
import pandas as pd
import matplotlib as mp
import matplotlib_inline as mpin
import os

[ ] df1=pd.read_excel("/content/QueryResults(38194and38059].xlsx")

[ ] for r,d,f in os.walk("/content"):
      print(r,d,f)
      break

    /content ['.config', 'sample_data'] ['QueryResults_greaterthan100000.xlsx', 'QueryResults(65000and48200].xlsx', 'QueryResults[65000and100000].xlsx', 'QueryResults(48200and38194].xlsx', 'QueryResults(38194and38059].xlsx']

[ ] f

    ['QueryResults_greaterthan100000.xlsx',
     'QueryResults(65000and48200].xlsx',
     'QueryResults[65000and100000].xlsx',
     'QueryResults(48200and38194].xlsx',
     'QueryResults(38194and38059].xlsx']

[ ] df_combine=pd.DataFrame()
    for file in f:
      df=pd.read_excel("/content/"+file)
      df_combine=pd.concat([df_combine,df])

[ ] df_combine.to_csv("Final_PostData.csv",index=False)
```

The merged file is then downloaded.

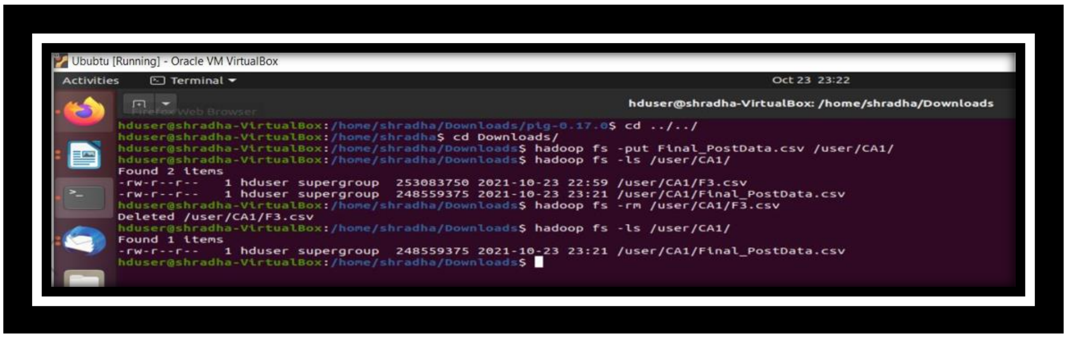
Local Ubuntu Server: Once Hadoop Cluster (Namenode, Datanode)(v3.3.1), HIVE(v3.1.2) and PIG(v0.17.0) are installed, further steps can be followed.
Cloud GCP: Since DataProc utility is utilized, all components are already installed.

1. The status of Hadoop Cluster is checked.
   Command: jps



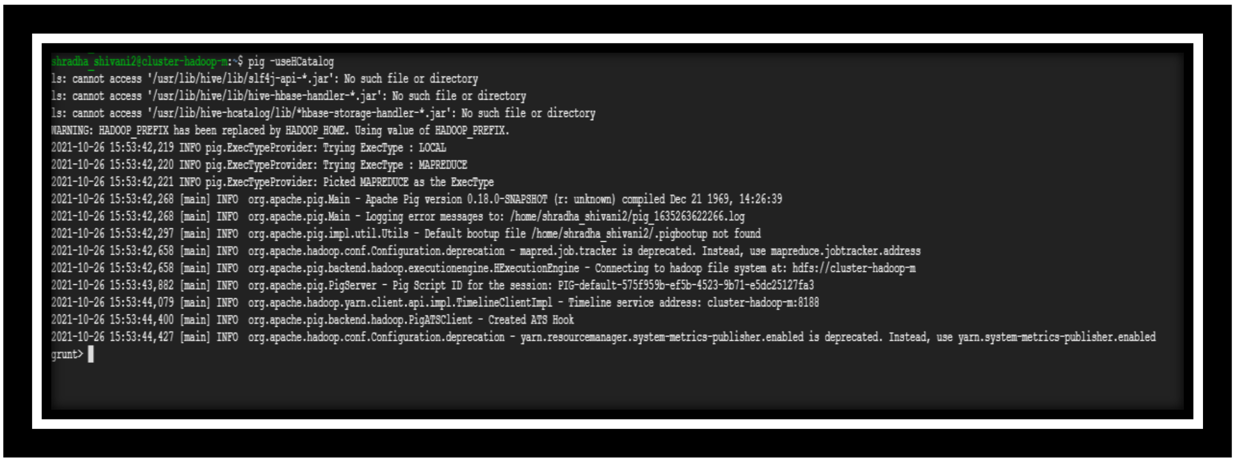
```
hduser@shradha-VirtualBox:/$ jps
2848 ResourceManager
2595 SecondaryNameNode
9703 Jps
2423 DataNode
2297 NameNode
2974 NodeManager
hduser@shradha-VirtualBox:/$
```

2. Dataset Final_PostData.csv is placed in HDFS location '/user/CA1'
   Command: hadoop fs -put Final_PostData.csv /user/CA1


Ububtu [Running] - Oracle VM VirtualBox
Activities
Terminal
Oct 23 23:22
hduser@shradha-VirtualBox: /home/shradha/Downloads

```
hduser@shradha-VirtualBox:/home/shradha/Downloads/pig-0.17.0$ cd ../../
hduser@shradha-VirtualBox:/home/shradha$ cd Downloads/
hduser@shradha-VirtualBox:/home/shradha/Downloads$ hadoop fs -put Final_PostData.csv /user/CA1/
hduser@shradha-VirtualBox:/home/shradha/Downloads$ hadoop fs -ls /user/CA1/
Found 2 items
-rw-r--r--   1 hduser supergroup  253083750 2021-10-23 22:59 /user/CA1/F3.csv
-rw-r--r--   1 hduser supergroup  248559375 2021-10-23 23:21 /user/CA1/Final_PostData.csv
hduser@shradha-VirtualBox:/home/shradha/Downloads$ hadoop fs -rm /user/CA1/F3.csv
Deleted /user/CA1/F3.csv
hduser@shradha-VirtualBox:/home/shradha/Downloads$ hadoop fs -ls /user/CA1/
Found 1 items
-rw-r--r--   1 hduser supergroup  248559375 2021-10-23 23:21 /user/CA1/Final_PostData.csv
hduser@shradha-VirtualBox:/home/shradha/Downloads$ 
```

3. Logged in to pig terminal using HCatalog. The Command used is pig -useHCatalog.

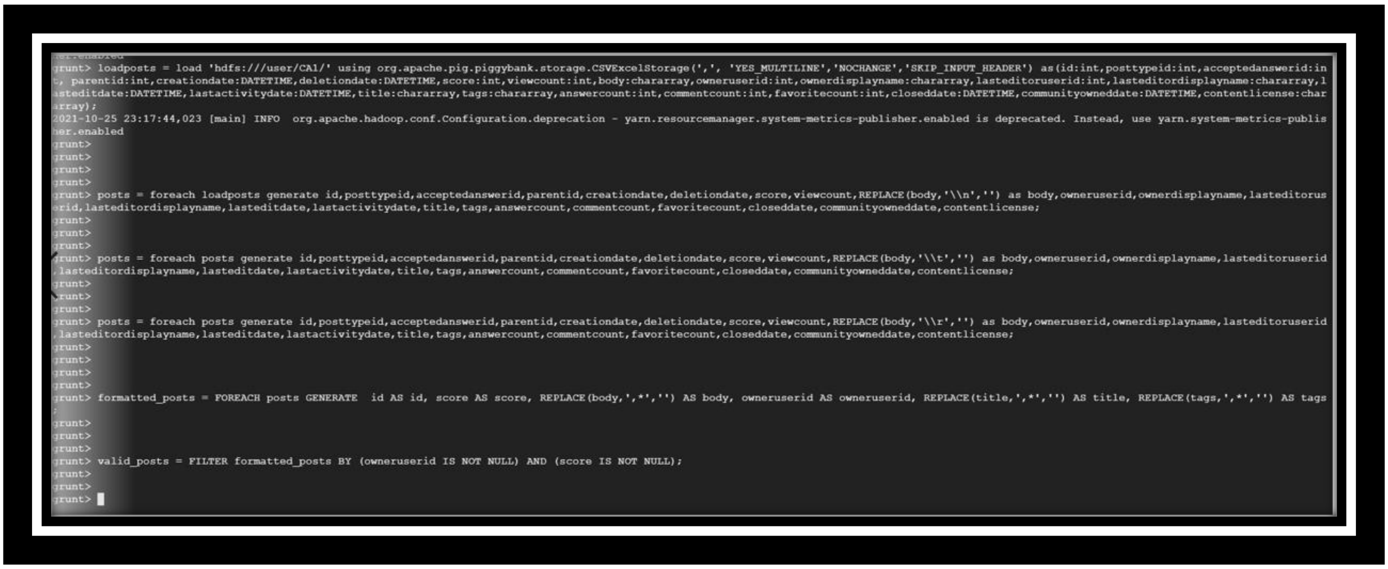


```
shradha_shivani2@cluster-hadoop-m:~$ pig -useHCatalog
ls: cannot access '/usr/lib/hive/lib/slf4j-api-*.jar': No such file or directory
ls: cannot access '/usr/lib/hive/lib/hive-hbase-handler-*.jar': No such file or directory
ls: cannot access '/usr/lib/hive-hcatalog/lib/*hbase-storage-handler-*.jar': No such file or directory
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
2021-10-26 15:53:42,219 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2021-10-26 15:53:42,220 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2021-10-26 15:53:42,221 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2021-10-26 15:53:42,268 [main] INFO  org.apache.pig.Main - Apache Pig version 0.18.0-SNAPSHOT (r: unknown) compiled Dec 21 1969, 14:26:39
2021-10-26 15:53:42,268 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/shradha_shivani2/pig_1635263622266.log
2021-10-26 15:53:42,297 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/shradha_shivani2/.pigbootup not found
2021-10-26 15:53:42,658 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2021-10-26 15:53:42,658 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://cluster-hadoop-m
2021-10-26 15:53:43,882 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-575f959b-ef5b-4523-9b71-e5dc25127fa3
2021-10-26 15:53:44,079 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: cluster-hadoop-m:8188
2021-10-26 15:53:44,400 [main] INFO  org.apache.pig.backend.hadoop.PigATSClient - Created ATS Hook
2021-10-26 15:53:44,427 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
grunt> 
```

4. Using Pig, the data present in Final_Posts.csv is further cleaned. New lines, tab or carriage return characters, single characters, special symbols etc are replaced with space in 'Body' column. Additionally, the dataset is filtered for NOT NULL OWNERUSERID tuples.

   Command:

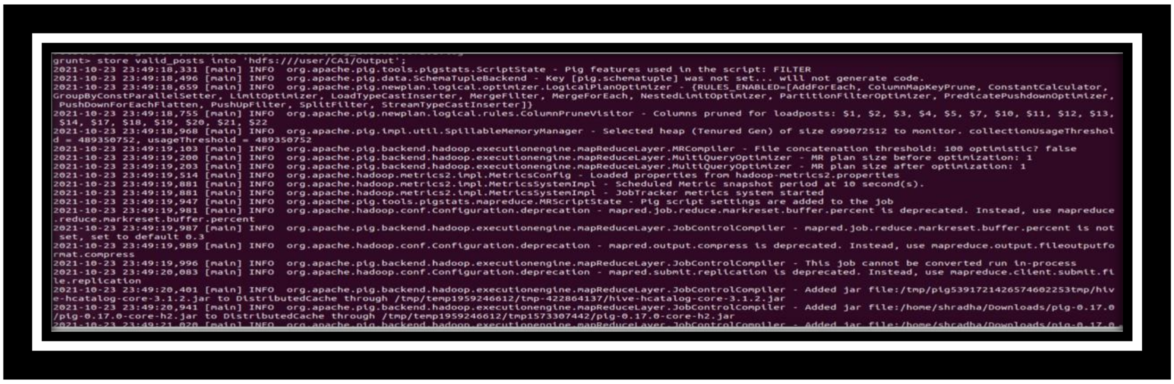
```
grunt>loadposts = load 'hdfs:///user/CA1/Final_PostData.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES_MULTILINE','NOCHANGE','SKIP_INPUT_HEADER') as(id:int,posttypeid:int,acceptedanswerid:int, parentid:int,creationdate:DATETIME,deletiondate:DATETIME,score:int,viewcount:int,body:chararray,owneruserid:int,ownerdisplayname:chararray,lasteditoruserid:int,lasteditordisplayname:chararray,lasteditdate:DATETIME,lastactivitydate:DATETIME,title:chararray,tags:chararray,answercount:int,commentcount:int,favoritecount:int,closeddate:DATETIME,communityowneddate:DATETIME,contentlicense:chararray);
grunt>posts = foreach loadposts generate id,posttypeid,acceptedanswerid,parentid,creationdate,deletiondate,score,viewcount,REPLACE(body,'\\n','') as body,owneruserid,ownerdisplayname,lasteditoruserid,lasteditordisplayname,lasteditdate,lastactivitydate,title,tags,answercount,commentcount,favoritecount,closeddate,communityowneddate,contentlicense;
grunt> posts = foreach posts generate id,posttypeid,acceptedanswerid,parentid,creationdate,deletiondate,score,viewcount,REPLACE(body,'\\t','') as body,owneruserid,ownerdisplayname,lasteditoruserid,lasteditordisplayname,lasteditdate,lastactivitydate,title,tags,answercount,commentcount,favoritecount,closeddate,communityowneddate,contentlicense;
grunt> posts = foreach posts generate id,posttypeid,acceptedanswerid,parentid,creationdate,deletiondate,score,viewcount,REPLACE(body,'\\r','') as body,owneruserid,ownerdisplayname,lasteditoruserid,lasteditordisplayname,lasteditdate,lastactivitydate,title,tags,answercount,commentcount,favoritecount,closeddate,communityowneddate,contentlicense;
grunt> formatted_posts = FOREACH posts GENERATE  id AS id, score AS score, REPLACE(body,',*','') AS body, owneruserid AS owneruserid, REPLACE(title,',*','') AS title, REPLACE(tags,',*','') AS tags;
grunt>
grunt> valid_posts = FILTER formatted_posts BY (owneruserid IS NOT NULL) AND (score IS NOT NULL);
grunt>
```



```
grunt> loadposts = load 'hdfs:///user/CA1/' using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES_MULTILINE','NOCHANGE','SKIP_INPUT_HEADER') as(id:int,posttypeid:int,acceptedanswerid:in
t, parentid:int,creationdate:DATETIME,deletiondate:DATETIME,score:int,viewcount:int,body:chararray,owneruserid:int,ownerdisplayname:chararray,lasteditoruserid:int,lasteditordisplayname:chararray,l
asteditdate:DATETIME,lastactivitydate:DATETIME,title:chararray,tags:chararray,answercount:int,commentcount:int,favoritecount:int,closeddate:DATETIME,communityowneddate:DATETIME,contentlicense:char
array);
2021-10-25 23:17:44,023 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publis
her.enabled
grunt>
grunt>
grunt>
grunt>
grunt> posts = foreach loadposts generate id,posttypeid,acceptedanswerid,parentid,creationdate,deletiondate,score,viewcount,REPLACE(body,'\\n','') as body,owneruserid,ownerdisplayname,lasteditorus
erid,lasteditordisplayname,lasteditdate,lastactivitydate,title,tags,answercount,commentcount,favoritecount,closeddate,communityowneddate,contentlicense;
grunt>
grunt>
grunt>
grunt> posts = foreach posts generate id,posttypeid,acceptedanswerid,parentid,creationdate,deletiondate,score,viewcount,REPLACE(body,'\\t','') as body,owneruserid,ownerdisplayname,lasteditoruserid
,lasteditordisplayname,lasteditdate,lastactivitydate,title,tags,answercount,commentcount,favoritecount,closeddate,communityowneddate,contentlicense;
grunt>
grunt>
grunt>
grunt> posts = foreach posts generate id,posttypeid,acceptedanswerid,parentid,creationdate,deletiondate,score,viewcount,REPLACE(body,'\\r','') as body,owneruserid,ownerdisplayname,lasteditoruserid
,lasteditordisplayname,lasteditdate,lastactivitydate,title,tags,answercount,commentcount,favoritecount,closeddate,communityowneddate,contentlicense;
grunt>
grunt>
grunt>
grunt>
grunt> formatted_posts = FOREACH posts GENERATE  id AS id, score AS score, REPLACE(body,',*','') AS body, owneruserid AS owneruserid, REPLACE(title,',*','') AS title, REPLACE(tags,',*','') AS tags
;
grunt>
grunt>
grunt>
grunt> valid_posts = FILTER formatted_posts BY (owneruserid IS NOT NULL) AND (score IS NOT NULL);
grunt>
grunt>
grunt> 
```

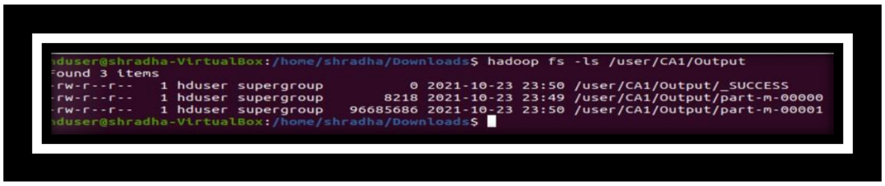
5. The cleaned file is then stored in HDFS at location /user/CA1/Output. (Note: while implementing in Google Cloud Platform, the cleaned file was stored in HDFS at location /user/CA1/combined')
   Command:

   grunt> store valid_posts into 'hdfs:///user/CA1/Output';



```
grunt> store valid_posts into 'hdfs:///user/CA1/Output';
2021-10-23 23:49:18,331 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: FILTER
2021-10-23 23:49:18,496 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2021-10-23 23:49:18,659 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, ConstantCalculator,
GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NestedLimitOptimizer, PartitionFilterOptimizer, PredicatePushdownOptimizer,
 PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter]}
2021-10-23 23:49:18,755 [main] INFO  org.apache.pig.newplan.logical.rules.ColumnPruneVisitor - Columns pruned for loadposts: $1, $2, $3, $4, $5, $7, $10, $11, $12, $13,
 $14, $17, $18, $19, $20, $21, $22
2021-10-23 23:49:18,968 [main] INFO  org.apache.pig.impl.util.SpillableMemoryManager - Selected heap (Tenured Gen) of size 699072512 to monitor. collectionUsageThreshol
d = 489350752, usageThreshold = 489350752
2021-10-23 23:49:19,103 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
2021-10-23 23:49:19,200 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size before optimization: 1
2021-10-23 23:49:19,203 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size after optimization: 1
2021-10-23 23:49:19,514 [main] INFO  org.apache.hadoop.metrics2.impl.MetricsConfig - Loaded properties from hadoop-metrics2.properties
2021-10-23 23:49:19,881 [main] INFO  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - Scheduled Metric snapshot period at 10 second(s).
2021-10-23 23:49:19,881 [main] INFO  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system started
2021-10-23 23:49:19,947 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.MRScriptState - Pig script settings are added to the job
2021-10-23 23:49:19,981 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.reduce.markreset.buffer.percent is deprecated. Instead, use mapreduce
.reduce.markreset.buffer.percent
2021-10-23 23:49:19,987 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - mapred.job.reduce.markreset.buffer.percent is not
 set, set to default 0.3
2021-10-23 23:49:19,989 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.output.compress is deprecated. Instead, use mapreduce.output.fileoutputfo
rmat.compress
2021-10-23 23:49:19,996 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - This job cannot be converted run in-process
2021-10-23 23:49:20,083 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.submit.replication is deprecated. Instead, use mapreduce.client.submit.fi
le.replication
2021-10-23 23:49:20,401 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Added jar file:/tmp/pig5391721426574602253tmp/hiv
e-hcatalog-core-3.1.2.jar to DistributedCache through /tmp/temp1959246612/tmp-422864137/hive-hcatalog-core-3.1.2.jar
2021-10-23 23:49:20,941 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Added jar file:/home/shradha/Downloads/pig-0.17.0
/pig-0.17.0-core-h2.jar to DistributedCache through /tmp/temp1959246612/tmp1573307442/pig-0.17.0-core-h2.jar
2021-10-23 23:49:21,020 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Added jar file:/home/shradha/Downloads/pig-0.17.0
```

6. After successful log generation when the store command was run, the cleaned files can been seen at '/user/CA1/Output'
   Command: hadoop fs -ls /user/CA1/Output

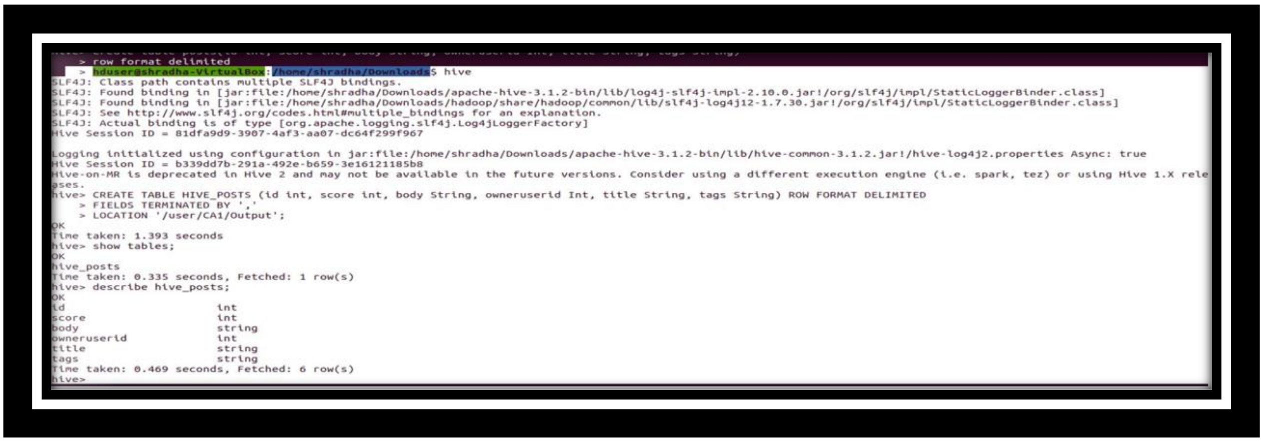


```
hduser@shradha-VirtualBox:/home/shradha/Downloads$ hadoop fs -ls /user/CA1/Output
Found 3 items
-rw-r--r--   1 hduser supergroup          0 2021-10-23 23:50 /user/CA1/Output/_SUCCESS
-rw-r--r--   1 hduser supergroup       8218 2021-10-23 23:49 /user/CA1/Output/part-m-00000
-rw-r--r--   1 hduser supergroup   96685686 2021-10-23 23:50 /user/CA1/Output/part-m-00001
hduser@shradha-VirtualBox:/home/shradha/Downloads$ 
```

7. The _SUCCESS log was removed.
   Command:
   hduser@shradha-VirtualBox:/home/shradha/Downloads$ hadoop fs -rm /user/CA1/Output/_SUCCESS
   Deleted /user/CA1/Output/_SUCCESS
   hduser@shradha-VirtualBox:/home/shradha/Downloads$

8. HIVE_POSTS table was created in HIVE.
Command:

```
  > hduser@shradha-VirtualBox:/home/shradha/Downloads$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/shradha/Downloads/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-
2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/shradha/Downloads/hadoop/share/hadoop/common/lib/slf4j-log4j12-
1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 81dfa9d9-3907-4af3-aa07-dc64f299f967

Logging initialized using configuration in jar:file:/home/shradha/Downloads/apache-hive-3.1.2-bin/lib/hive-common-
3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = b339dd7b-291a-492e-b659-3e16121185b8
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution
engine (i.e. spark, tez) or using Hive 1.X releases.
hive> CREATE TABLE HIVE_POSTS (id int, score int, body String, owneruserid Int, title String, tags String) ROW
FORMAT DELIMITED
    > FIELDS TERMINATED BY ','

OK
Time taken: 1.393 seconds
hive> show tables;
OK
hive_posts
Time taken: 0.335 seconds, Fetched: 1 row(s)
hive> describe hive_posts;
OK
id                 int
score              int
body               string
owneruserid        int
title              string
tags               string
Time taken: 0.469 seconds, Fetched: 6 row(s)
hive>
```



```
    > row format delimited
    > hduser@shradha-VirtualBox:/home/shradha/Downloads$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/shradha/Downloads/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/shradha/Downloads/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 81dfa9d9-3907-4af3-aa07-dc64f299f967

Logging initialized using configuration in jar:file:/home/shradha/Downloads/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = b339dd7b-291a-492e-b659-3e16121185b8
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X rele
ases.
hive> CREATE TABLE HIVE_POSTS (id int, score int, body String, owneruserid Int, title String, tags String) ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > LOCATION '/user/CA1/Output';
OK
Time taken: 1.393 seconds
hive> show tables;
OK
hive_posts
Time taken: 0.335 seconds, Fetched: 1 row(s)
hive> describe hive_posts;
OK
id                      int
score                   int
body                    string
owneruserid             int
title                   string
tags                    string
Time taken: 0.469 seconds, Fetched: 6 row(s)
hive>
```

9. The cleaned file part-m-00001 at HDFS location '/user/CA1/Output/' was loaded to the created HIVE_POSTS table using HIVE LOAD command. The cleaned file is selected depending on the size and content of the file. For this implementation part-m-00001 was used for further processing.
Command:

```
hive> LOAD DATA INPATH 'hdfs:///user/CA1/Output/part-m-00001' INTO TABLE hive_posts;
```

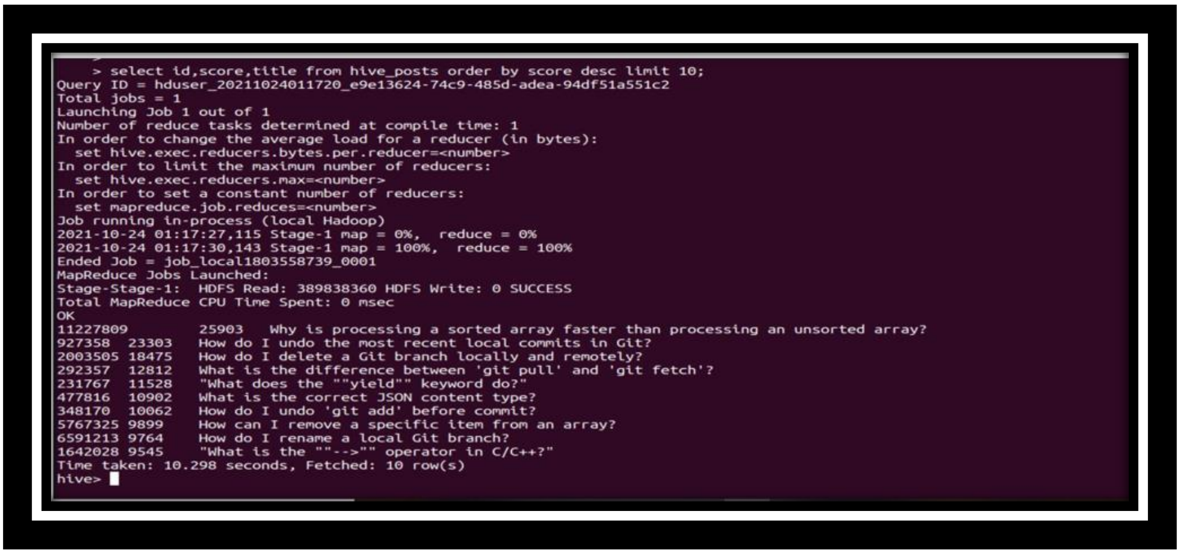
Queries Ran:

Query1:

Use Pig/Hive/MapReduce - Extract, Transform and Load the data as applicable to get the top 10 posts by score

Command:

hive> select id, score, title from hive_posts order by score desc limit 10;



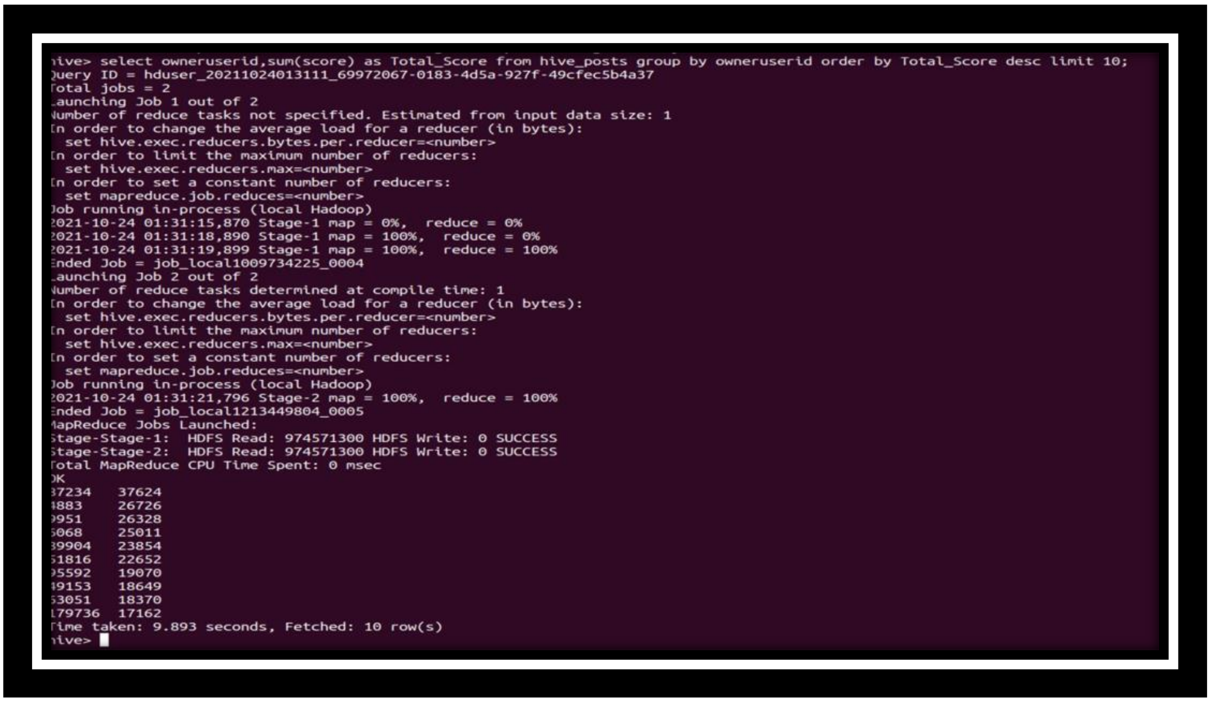
```
    > select id,score,title from hive_posts order by score desc limit 10;
Query ID = hduser_20211024011720_e9e13624-74c9-485d-adea-94df51a551c2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2021-10-24 01:17:27,115 Stage-1 map = 0%,  reduce = 0%
2021-10-24 01:17:30,143 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1803558739_0001
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 389838360 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
11227809        25903   Why is processing a sorted array faster than processing an unsorted array?
927358  23303   How do I undo the most recent local commits in Git?
2003505 18475   How do I delete a Git branch locally and remotely?
292357  12812   What is the difference between 'git pull' and 'git fetch'?
231767  11528   "What does the ""yield"" keyword do?"
477816  10902   What is the correct JSON content type?
348170  10062   How do I undo 'git add' before commit?
5767325 9899    How can I remove a specific item from an array?
6591213 9764    How do I rename a local Git branch?
1642028 9545    "What is the ""-->"" operator in C/C++?"
Time taken: 10.298 seconds, Fetched: 10 row(s)
hive> █
```

Query2:

Use Pig/Hive/MapReduce - Extract, Transform and Load the data as applicable to get the top 10 users by post score

Command:

hive> select owneruserid,sum(score) as Total_Score from hive_posts group by owneruserid order by Total_Score desc limit 10;



```
ive> select owneruserid,sum(score) as Total_Score from hive_posts group by owneruserid order by Total_Score desc limit 10;
uery ID = hduser_20211024013111_69972067-0183-4d5a-927f-49cfec5b4a37
otal jobs = 2
aunching Job 1 out of 2
umber of reduce tasks not specified. Estimated from input data size: 1
n order to change the average load for a reducer (in bytes):
 set hive.exec.reducers.bytes.per.reducer=<number>
n order to limit the maximum number of reducers:
 set hive.exec.reducers.max=<number>
n order to set a constant number of reducers:
 set mapreduce.job.reduces=<number>
ob running in-process (local Hadoop)
021-10-24 01:31:15,870 Stage-1 map = 0%,  reduce = 0%
021-10-24 01:31:18,890 Stage-1 map = 100%,  reduce = 0%
021-10-24 01:31:19,899 Stage-1 map = 100%,  reduce = 100%
nded Job = job_local1009734225_0004
aunching Job 2 out of 2
umber of reduce tasks determined at compile time: 1
n order to change the average load for a reducer (in bytes):
 set hive.exec.reducers.bytes.per.reducer=<number>
n order to limit the maximum number of reducers:
 set hive.exec.reducers.max=<number>
n order to set a constant number of reducers:
 set mapreduce.job.reduces=<number>
ob running in-process (local Hadoop)
021-10-24 01:31:21,796 Stage-2 map = 100%,  reduce = 100%
nded Job = job_local1213449804_0005
apReduce Jobs Launched:
tage-Stage-1:  HDFS Read: 974571300 HDFS Write: 0 SUCCESS
tage-Stage-2:  HDFS Read: 974571300 HDFS Write: 0 SUCCESS
otal MapReduce CPU Time Spent: 0 msec
K
7234   37624
883    26726
951    26328
068    25011
9904   23854
1816   22652
5592   19070
9153   18649
3051   18370
79736  17162
ime taken: 9.893 seconds, Fetched: 10 row(s)
ive> █
```

Query3:

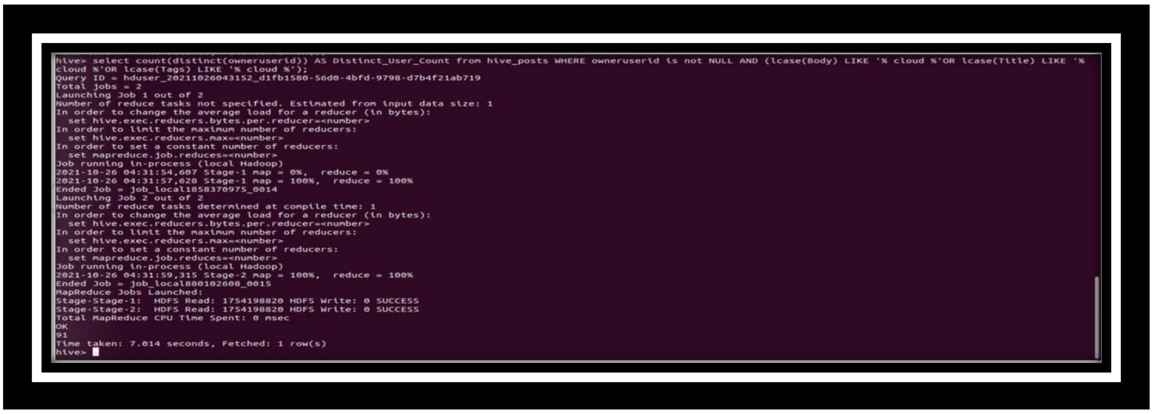
Use Pig/Hive/MapReduce - Extract, Transform and Load the data as applicable to get The number of distinct users, who used the word “cloud” in one of their posts.

Command: Note(Anyone of these commands can be used for this query)

hive> select count(distinct(owneruserid)) AS Distinct_User_Count from hive_posts WHERE locate (" cloud ",concat(Body,Title,Tags))>0;

OR

hive> select count(distinct(owneruserid)) AS Distinct_User_Count from hive_posts where (Body like '% cloud %' OR title like '% cloud %' OR tags like '% cloud %');



```
hive> select count(distinct(owneruserid)) AS Distinct_User_Count from hive_posts WHERE owneruserid is not NULL AND (lcase(Body) LIKE '% cloud %'OR lcase(Title) LIKE '%
cloud %'OR lcase(Tags) LIKE '% cloud %');
Query ID = hduser_20211026043152_d1fb1580-56d0-4bfd-9798-d7b4f21ab719
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2021-10-26 04:31:54,607 Stage-1 map = 0%,  reduce = 0%
2021-10-26 04:31:57,628 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1858370975_0014
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2021-10-26 04:31:59,315 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local880102608_0015
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 1754198820 HDFS Write: 0 SUCCESS
Stage-Stage-2:  HDFS Read: 1754198820 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
91
Time taken: 7.014 seconds, Fetched: 1 row(s)
hive> 
```

We can used hive -e directly form the terminal to run the above 3 Hive queries and capture the output in a text file.

Commands:

Query1:

hduser@shradha-VirtualBox:/home/shradha/Downloads/hadoop/sbin$ hive -e "SELECT ID, SCORE, TITLE FROM HIVE_POSTS ORDER BY SCORE DESC LIMIT 10;" > /home/shradha/Downloads/Query1_Output.txt

Query2:

hduser@shradha-VirtualBox:/home/shradha/Downloads/hadoop/sbin$ hive -e "SELECT OWNERUSERID, SUM(SCORE) as TOTAL_SCORE FROM HIVE_POSTS GROUP BY OWNERUSERID ORDER BY TOTAL_SCORE DESC LIMIT 10;" > /home/shradha/Downloads/Query2_Output.txt

Query3:

hduser@shradha-VirtualBox:/home/shradha/Downloads$ hive -e "SELECT COUNT(DISTINCT(OWNERUSERID)) AS DISTINCT_OWNERS_COUNT FROM HIVE_POSTS WHERE LOCATE(' cloud ',concat(BODY,TITLE,TAGS))>0;" > /home/shradha/Downloads/Query3_Output.txt