**HU Extension Assignment 05 E63 Big Data Analytics**

Issued on: February 27, 2016 Due by 11:30PM EST, March 04, 2016

Please work in Hue’s Hive Editor.

**Problem 1.** Create your own tables KINGJAMES with columns for words and frequencies and insert into the table the result of Hadoop MapReduce GREP program which produce word counts on file all-bible. File is provided with this assignment. Tell us all words in table KINGJAMES which start with letter “w” and are 4 or more characters long and appear more than 250. There are not that many of those words so you can count them by hand. However, you want to be more automated so please change your query so that it gives you the number of such words as its output. When comparing a word with a string your use LIKE operator, like

word like ‘a%’ or word like ‘%th%’

Symbol ‘%’ means any number of characters. You measure the length of a string using function length() and you change the case of a word to all lower characters using function lower().

First I created a directory called all-bible in hdfs:

[cloudera@quickstart ~]$ hadoop fs -mkdir all-bible

[cloudera@quickstart ~]$

Then I copied all-bible file from local to hdfs directory:

[cloudera@quickstart ~]$ hadoop fs –copyFromLocal /mnt/hgfs/shared\_dir\_with\_cloudera\_vm/all-bible all-bible/

[cloudera@quickstart ~]$

Then I ran mapreduce grep program on all-bible:

[cloudera@quickstart ~]$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar grep all-bible/all-bible hadoop-grep-bible-output '\w+'

I made sure that the job ran successfully and then I confirmed that output is generated by the mapreduce program:

[cloudera@quickstart ~]$ hadoop fs -ls hadoop-grep-bible-output

Found 2 items

-rw-r--r-- 1 cloudera cloudera 0 2016-03-02 19:48 hadoop-grep-bible-output/\_SUCCESS

-rw-r--r-- 1 cloudera cloudera 147408 2016-03-02 19:48 hadoop-grep-bible-output/part-r-00000

[cloudera@quickstart ~]$

Then I created KINGJAMES table in hive (in hue’s hive editor in Firefox browser):

create table KINGJAMES (freq INT, word STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as textfile;

Then I loaded output of grep into KINGJAMES table by giving following command in Hue’s hive editor:

LOAD DATA INPATH "/user/cloudera/hadoop-grep-bible-output/part-r-00000" INTO TABLE KINGJAMES;

Then I gave the query to get all words starting with letter “w” and are 4 or more characters long and appear more than 250:

select \* from KINGJAMES where lower(word) like 'w%' and length(word)>=4 and freq>250 order by freq;

kingjames.freq,kingjames.word

253,well

261,Wherefore

283,written

284,whose

286,waters

288,world

301,works

304,wilderness

335,where

335,What

343,wicked

349,When

355,woman

396,water

407,wife

436,without

443,would

512,work

546,words

652,what

694,word

732,whom

1399,went

2487,when

2767,were

3819,will

4297,which

6057,with

These are the words in table KINGJAMES that start with letter ‘w’ and are 4 or more characters long and appear more than 250 times (left is the frequency, right is the word)

select count(1) from KINGJAMES where lower(word) like 'w%' and length(word)>=4 and freq>250;

28

So basically there are **28 words** that start with letter ‘w’ and are 4 or more characters long and appear more than 250 times in all-bible. (They are listed above)

Deliverables:

* assignment5\_problem1.sql (that lists all hive commands)
* assignment5\_problem1\_output.txt (which shows all words of KINGJAMES table that start with letter ‘w’ and are 4 or more characters long and appear more than 250 times.

**Problem 2**. Create your own table SHAKE similar to the one we used in class and populate it with results of MapReduce GREP program applied to the file all-shakespeare which is provided with this assignment. Create your own MERGED table similar to the one we used in class. The table will list all the word and the frequencies with which they appear in either table SHAKE or KINGJAMES. Your table will be “better” than the one we used in class. In class we only inserted into that table words that appear in both texts. Please use **outer joins** to populate the table with words that also appear in one but not the other text. Tell us how many words appear in table SHAKE but not in KINGJAMES and how many appear in KINGJAMES and not in SHAKE. Select 10 words from each group for us. To solve this problem you will have to consult Hive Tutorial at <https://cwiki.apache.org/confluence/display/Hive/Tutorial> or simply Google around the Web.

First I created all-shakespeare directory in hdfs:

[cloudera@quickstart ~]$ hadoop fs -mkdir all-shakespeare

[cloudera@quickstart ~]$

Then I copied all-shakespeare file to that directory:

[cloudera@quickstart ~]$ hadoop fs -copyFromLocal /mnt/hgfs/shared\_dir\_with\_cloudera\_vm/all-shakespeare all-shakespeare

[cloudera@quickstart ~]$

Then I ran mapreduce grep program on all-shakespeare file:

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar grep all-shakespeare/all-shakespeare hadoop-grep-shakespeare-output '\w+'

I made sure that the job ran successfully and that output is generated by grep program:

[cloudera@quickstart ~]$ hadoop fs -ls hadoop-grep-shakespeare-output

Found 2 items

-rw-r--r-- 1 cloudera cloudera 0 2016-03-02 20:41 hadoop-grep-shakespeare-output/\_SUCCESS

-rw-r--r-- 1 cloudera cloudera 299379 2016-03-02 20:41 hadoop-grep-shakespeare-output/part-r-00000

[cloudera@quickstart ~]$

Then I created SHAKE table in Hue’s hive query editor in Firefox browser:

create table SHAKE (freq INT, word STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as textfile;

I confirmed that SHAKE table is indeed created:

[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse

Found 5 items

drwxrwxrwx - cloudera supergroup 0 2016-02-29 16:41 /user/hive/warehouse/customers

drwxrwxrwx - cloudera supergroup 0 2016-03-02 20:07 /user/hive/warehouse/kingjames

drwxrwxrwx - cloudera supergroup 0 2016-02-29 16:41 /user/hive/warehouse/sample\_07

drwxrwxrwx - cloudera supergroup 0 2016-02-29 16:41 /user/hive/warehouse/sample\_08

drwxrwxrwx - cloudera supergroup 0 2016-03-02 20:46 /user/hive/warehouse/shake

[cloudera@quickstart ~]$

Then I loaded output of grep of Shakespeare into SHAKE table:

LOAD DATA INPATH "/user/cloudera/hadoop-grep-shakespeare-output/part-r-00000"

INTO TABLE SHAKE;

Then I created MERGED table in Hue’s hive query editor in Firefox browser:

create table MERGED (freq\_in\_bible INT, freq\_in\_shake INT, word STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as textfile;

Then I inserted data into MERGED table:

INSERT OVERWRITE TABLE MERGED

select k.freq AS freq\_in\_bible, s.freq AS freq\_in\_shake,

COALESCE(s.word, k.word) AS word

from KINGJAMES k FULL OUTER JOIN SHAKE s ON (k.word = s.word);

Then I counted how many words appear in SHAKE but not in KINGJAMES:

select count(1) from MERGED where freq\_in\_shake IS NOT NULL AND freq\_in\_bible IS NULL;

21428

So, number of words that are in table SHAKE, but not in table KINGJAMES = **21428**

Then I counted how many words appear in KINGJAMES and not in SHAKE:

select count(1) from MERGED where freq\_in\_bible IS NOT NULL AND freq\_in\_shake IS NULL;

6575

So, number of words that are in table KINGJAMES but not in table SHAKE = **6575**

Then I gave the below to get 10 words that appear in SHAKE but not in KINGJAMES:

select word from MERGED where freq\_in\_shake IS NOT NULL AND freq\_in\_bible IS NULL LIMIT 10;

2d

2s

4d

5s

6d

8d

AARON

ABERGAVENNY

ABHORSON

ABOUT

Above are the words that are in table SHAKE, but not in table KINGJAMES

Then I gave the below query to get 10 words that appear in KINGJAMES but not in SHAKE:

select word from MERGED where freq\_in\_bible IS NOT NULL AND freq\_in\_shake IS NULL LIMIT 10;

0

000

001

002

003

004

005

006

007

008

Above are the words that are in table KINGJAMES but not in table SHAKE

Then I gave the below query to get 10 common words that appear in both KINGJAMES and SHAKE:

select word from MERGED where freq\_in\_bible IS NOT NULL AND freq\_in\_shake IS NOT NULL LIMIT 10;

1

10

2

3

4

5

6

7

8

9

Above are the common words present in both KINGJAMES and SHAKE

Deliverables:

* assignment5\_problem2.sql (the file with hive commands)
* assignment5\_problem2\_output.txt (shows output of each command)

**Problem 3**. When you have your three queries for counting common words, words that are present in Bible but not in Shakespeare and the words present in Shakespeare but not in Bible refined and working, collect the execution times of those queries. This is not straightforward, since Hive does not give you a simple tool to time your queries. You can look in query logs (a tab next to the Results tab) and sum execution times of map and reduce jobs. That is close enough. Then change your Hue Query Editor and switch to Impala Editor. Run your queries in that editor. This time you have no way of read the time. You just make a subjective estimate. Compare the execution time of queries with Impala and Hive. Impala is usually much faster. One thing to notice here is that you can use Impala on some of Hive tables. Unfortunately not all. Hive is more versatile than Impala.

Hue’s Hive query editor:

To count common words in table KINGJAMES and SHAKE:

select count(1) from MERGED where freq\_in\_bible IS NOT NULL AND freq\_in\_shake IS NOT NULL;

Time taken: 2 seconds 600 msec (taken from log statement that provides mapreduce time)

Time taken: 16 seconds (taken from mapreduce tracking URL: Elapsed time field)

Hue’s Hive query editor:

To count words which appear in Bible but not in Shakespeare:

select count(1) from MERGED where freq\_in\_bible IS NOT NULL AND freq\_in\_shake IS NULL;

Time taken: 2 seconds 840 msec (taken from log statement that provides mapreduce time)

Time taken: 17 seconds (taken from mapreduce tracking URL: Elapsed time field)

Hue’s Hive query editor:

To count words which appear in Shakespeare but not in Bible:

select count(1) from MERGED where freq\_in\_bible IS NULL AND freq\_in\_shake IS NOT NULL;

Time taken: 2 seconds 810 msec (taken from log statement that provides mapreduce time)

Time taken: 17 seconds (taken from mapreduce tracking URL: Elapsed time field)

Hue’s Impala editor:

To count common words in table KINGJAMES and SHAKE:

select count(1) from MERGED where freq\_in\_bible IS NOT NULL AND freq\_in\_shake IS NOT NULL;

Time taken: 2 sec (subjective estimate made with the aid of a stopwatch)

Hue’s Impala editor:

To count words which appear in Bible but not in Shakespeare:

select count(1) from MERGED where freq\_in\_bible IS NOT NULL AND freq\_in\_shake IS NULL;

Time taken: 2 sec (subjective estimate made with the aid of a stopwatch)

Hue’s Impala editor:

To count words which appear in Shakespeare but not in Bible:

select count(1) from MERGED where freq\_in\_bible IS NULL AND freq\_in\_shake IS NOT NULL;

Time taken: 2 sec (subjective estimate made with the aid of a stopwatch)

Queries in Impala have ran (in this case) faster than same queries in Hive.

**Problem 4.** Please create Hive table APACHELOG for extraction of the content of Apache server logs:

CREATE TABLE apachelog (

host STRING,

identity STRING,

user STRING,

time STRING,

request STRING,

status STRING,

size STRING,

referer STRING,

agent STRING)

ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe' WITH SERDEPROPERTIES ( "input.regex" = "([^ ]\*) ([^ ]\*) ([^ ]\*) (-|\\[[^\\]]\*\\]) ([^ \"]\*|\"[^\"]\*\") (-|[0-9]\*) (-|[0-9]\*)(?: ([^ \"]\*|\"[^\"]\*\") ([^ \"]\*|\"[^\"]\*\"))?", "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s" )

STORED AS TEXTFILE;

Please expand the above regular expression to single line before copying the entire statement to Hue Hive editor.

Test success of creation of that table using two single line samples of Apache logs contained in files apache.access.2.log and apache.access.log (note files do not have .txt suffix) contained in the attached file examples\_older.zip. Once you are convinced that you can safely insert those two samples into your table apachelog, insert a bigger log contained in file apache\_log\_1.txt. Tell us how many lines of apache logs you have in table apachelog.

We are also attaching two groups of example data files for Hive: examples\_older.zip and examples.zip. You might find those files useful if you want to keep on learning about the technology. You could get those files by downloading Hive distributions, as described in notes.

I executed the following command in Hue’s Hive editor and created table apachelog:

CREATE TABLE apachelog ( host STRING, identity STRING, user STRING, time STRING, request STRING, status STRING, size STRING, referer STRING, agent STRING) ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe' WITH SERDEPROPERTIES ( "input.regex" = "([^ ]\*) ([^ ]\*) ([^ ]\*) (-|\\[[^\\]]\*\\]) ([^ \"]\*|\"[^\"]\*\") (-|[0-9]\*) (-|[0-9]\*)(?: ([^ \"]\*|\"[^\"]\*\") ([^ \"]\*|\"[^\"]\*\"))?", "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s" ) STORED AS TEXTFILE;

Then I confirmed that the table is created:

select \* from apachelog;

*The command is executed with no output results.*

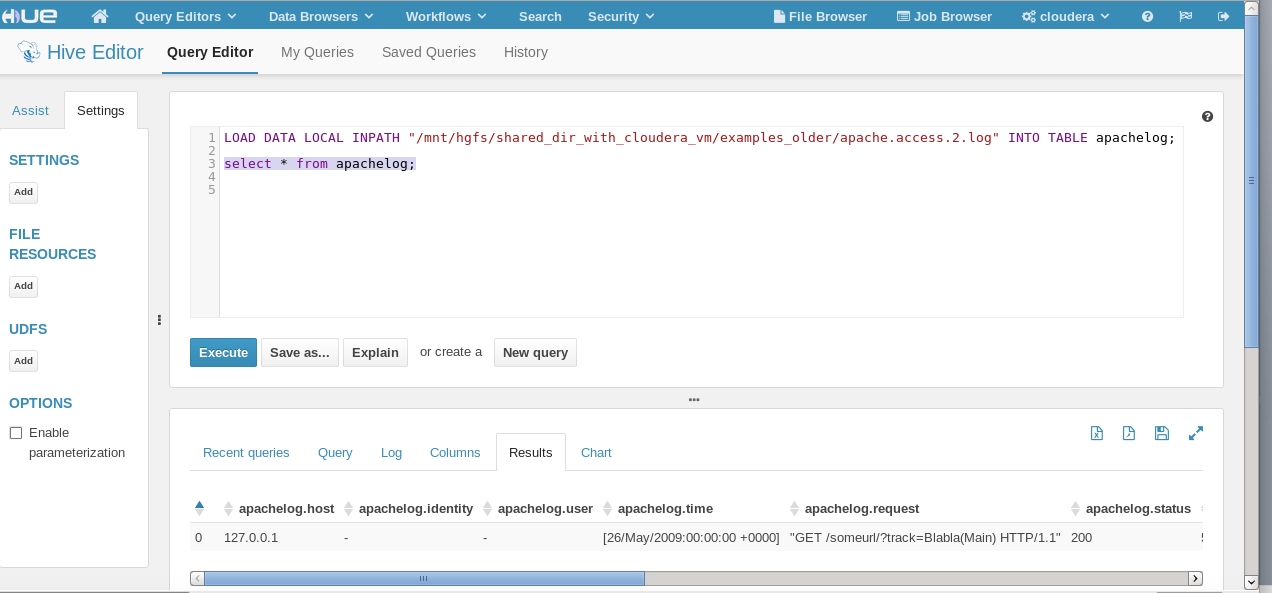
Then I executed following command to load data from apache.access.2.log into apachelog:

LOAD DATA LOCAL INPATH "/mnt/hgfs/shared\_dir\_with\_cloudera\_vm/examples\_older/apache.access.2.log" INTO TABLE apachelog;

Then I checked the apachelog table again:

select \* from apachelog;

I saw one result as shown in this image:



Then I executed this command to load apache.access.log into apachelog table:

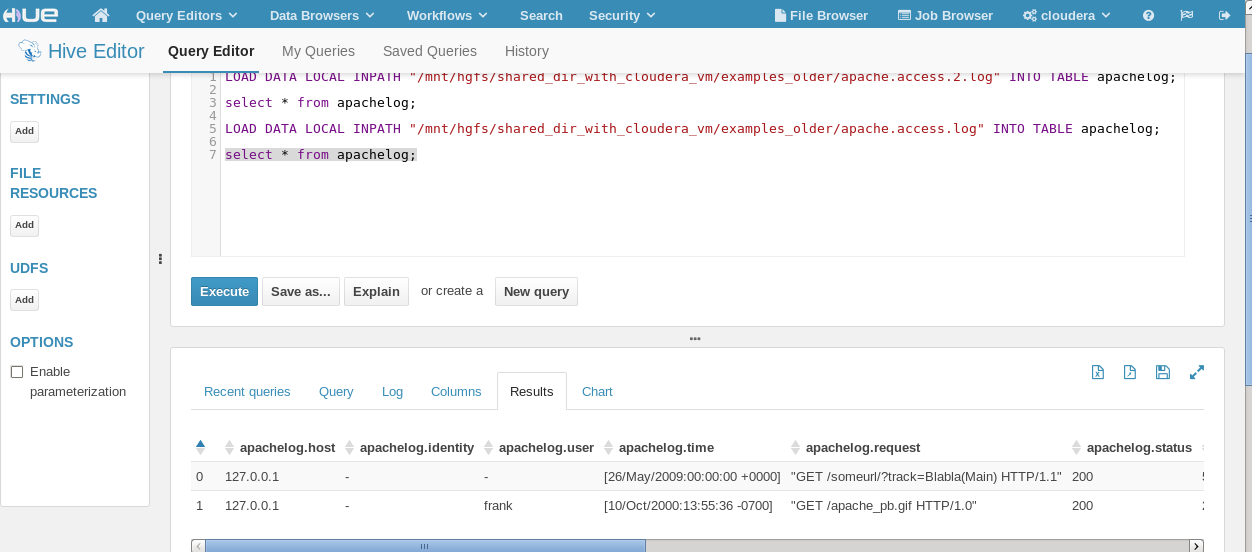
LOAD DATA LOCAL INPATH "/mnt/hgfs/shared\_dir\_with\_cloudera\_vm/examples\_older/apache.access.log"

INTO TABLE apachelog;

Then I again checked apachelog table:

select \* from apachelog;

I saw 2 records in the table as shown in this image:



Next, the problem statement mentions apache-log-1.txt (I assume it meant to say access\_log\_1.txt).

I executed this command to load data from access\_log\_1.txt into table apachelog:

LOAD DATA LOCAL INPATH "/mnt/hgfs/shared\_dir\_with\_cloudera\_vm/access\_log\_1.txt"

INTO TABLE apachelog;

Then I counted the number of records in apachelog table:

select count(1) from apachelog;

39346

So, there are **39346 lines of apache logs** in table apachelog:

39344 are from access\_log\_1.txt ,

1 is from apache.access.2.log ,

1 is from apache.access.log

Deliverables:

* assignment5\_problem4.sql (that lists all hive commands)
* assignment5\_problem4\_output.txt (that shows 10 lines of apachelog table)

Please, describe every step of your work and present all intermediate and final results in a Word document. Please, copy past text version of your command. We cannot retype text that is in JPG images. Please, always submit a copy of original, working scripts and class files you used as separate files. Sometimes we need to run your code and retyping is too costly. Please, submit to the class drop box. For issues and comments visit the class Discussion Board .