HU Extension School Assignment 04 E-185 Big Data Analytics

Handed out: 03/02/2013 Due by 5:30PM on Friday, 03/08/2013

**Files all-bible.tar and all-shakespeare.tar used during the lecture are attached. Files kv1.txt, kv2.txt, kv3.txt referenced in the lecture notes could be found on ami-5abc2f33 and most probably any other instance created for a Hive interactive job flow by the Elastic Map Reduce service. The files reside in the directory:**

**/home/hadoop/hive/examples/files.**

**Using AWS Elastic Map Reduce service open an interactive Hive job with a minimal cluster. Use installation of Hive on the master node of that cluster.**

**Problem 1) Add regular expression serializer/deserializer RegexSerDe by issuing the following command on the hive prompt:**

**hive> add jar hive/contrib/hive\_contrib-0.8.1.jar**

**The version of your hive-contrib.jar file might be different. Now, you are ready to create hive table apachelog using create table command presented on slide 61 of the lecture notes.**

**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;**

**Verify that table apachelog accepts apache access logs by loading both apache log files in the directory /home/hadoop/hive/examples/files. Please make sure that you load all rows present in those files. Select all rows and present the result.**

After creating the Hive Interactive job and connecting to the AWS instance, we started Hive up and added the JAR file as requested.

hadoop@domU-12-31-39-00-64-46:~$ hive

Logging initialized using configuration in file:/home/hadoop/.versions/hive-0.8.1/conf/hive-log4j.properties

Hive history file=/mnt/var/lib/hive\_081/tmp/history/hive\_job\_log\_hadoop\_201303072224\_1241679347.txt

hive> add jar hive/contrib/hive-contrib-0.8.1.jar;

Added hive/contrib/hive-contrib-0.8.1.jar to class path

Added resource: hive/contrib/hive-contrib-0.8.1.jar

After adding the JAR file successfully, we create the table according to the specs defined:

hive> 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;

OK

Time taken: 18.202 seconds

Adding the data from the files in the examples directories:

hive> LOAD DATA LOCAL INPATH '/home/hadoop/hive/examples/files/apache.access.log' INTO TABLE apachelog;

Copying data from file:/home/hadoop/hive/examples/files/apache.access.log

Loading data to table default.apachelog

OK

Time taken: 2.066 seconds

hive> LOAD DATA LOCAL INPATH '/home/hadoop/hive/examples/files/apache.access.2.log' INTO TABLE apachelog;

Copying data from file:/home/hadoop/hive/examples/files/apache.access.2.log

Loading data to table default.apachelog

OK

Time taken: 0.787 seconds

Doing a simple select on the tables, we can tell that it only has 2 rows on it.

hive> select \* from apachelog;

OK

127.0.0.1 - - [26/May/2009:00:00:00 +0000] "GET /someurl/?track=Blabla(Main) HTTP/1.1" 200 5864 - "Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US) AppleWebKit/525.19 (KHTML, like Gecko) Chrome/1.0.154.65 Safari/525.19"

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache\_pb.gif HTTP/1.0" 200 2326 NULL NULL

Time taken: 1.264 seconds

The count operator tells us that the table has 2 rows on it also.

hive> select count(\*) from apachelog;

Total MapReduce 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 mapred.reduce.tasks=<number>

Starting Job = job\_201303072219\_0001, Tracking URL = http://domU-12-31-39-00-64-46.compute-1.internal:9100/jobdetails.jsp?jobid=job\_201303072219\_0001

Kill Command = /home/hadoop/bin/hadoop job -Dmapred.job.tracker=10.254.107.180:9001 -kill job\_201303072219\_0001

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

(...)

MapReduce Total cumulative CPU time: 3 seconds 0 msec

Ended Job = job\_201303072219\_0001

Counters:

MapReduce Jobs Launched:

Job 0: Map: 1 Reduce: 1 Accumulative CPU: 3.0 sec HDFS Read: 632 HDFS Write: 2 SUCCESS

Total MapReduce CPU Time Spent: 3 seconds 0 msec

OK

2

Time taken: 58.691 seconds

Which is consistent with the fact that each file only has one line each:

hadoop@domU-12-31-39-00-64-46:~$ cat /home/hadoop/hive/examples/files/apache.access.log | wc -l

1

hadoop@domU-12-31-39-00-64-46:~$ cat /home/hadoop/hive/examples/files/apache.access.2.log | wc -l

1

**Problem 2) S3 bucket s3n://elasticmapreduce/samples/pig-apache/input contains 6 apache\_access\_logs. Copy those logs to an HDFS directory on your cluster. Load all 6 access\_logs into table apachelog. Query the number of rows you loaded.**

The 6 access log files are as follows:

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -ls s3n://elasticmapreduce/samples/pig-apache/input

Found 6 items

-rwxrwxrwx 1 8754118 2009-08-04 20:33 /samples/pig-apache/input/access\_log\_1

-rwxrwxrwx 1 8902171 2009-08-04 20:33 /samples/pig-apache/input/access\_log\_2

-rwxrwxrwx 1 8896201 2009-08-04 20:33 /samples/pig-apache/input/access\_log\_3

-rwxrwxrwx 1 8886636 2009-08-04 20:33 /samples/pig-apache/input/access\_log\_4

-rwxrwxrwx 1 8902365 2009-08-04 20:34 /samples/pig-apache/input/access\_log\_5

-rwxrwxrwx 1 8892828 2009-08-04 20:34 /samples/pig-apache/input/access\_log\_6

We proceed to copy them to the local HDFS cluster:

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -mkdir problem2

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -ls /user/hadoop/problem2

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cp s3n://elasticmapreduce/samples/pig-apache/input/\* /user/hadoop/problem2/

13/03/07 22:59:27 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_1' for reading

13/03/07 22:59:30 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_2' for reading

13/03/07 22:59:31 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_3' for reading

13/03/07 22:59:32 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_4' for reading

13/03/07 22:59:34 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_5' for reading

13/03/07 22:59:35 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_6' for reading

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -ls /user/hadoop/problem2

Found 6 items

-rw-r--r-- 1 hadoop supergroup 8754118 2013-03-07 22:59 /user/hadoop/problem2/access\_log\_1

-rw-r--r-- 1 hadoop supergroup 8902171 2013-03-07 22:59 /user/hadoop/problem2/access\_log\_2

-rw-r--r-- 1 hadoop supergroup 8896201 2013-03-07 22:59 /user/hadoop/problem2/access\_log\_3

-rw-r--r-- 1 hadoop supergroup 8886636 2013-03-07 22:59 /user/hadoop/problem2/access\_log\_4

-rw-r--r-- 1 hadoop supergroup 8902365 2013-03-07 22:59 /user/hadoop/problem2/access\_log\_5

-rw-r--r-- 1 hadoop supergroup 8892828 2013-03-07 22:59 /user/hadoop/problem2/access\_log\_6

And load them in the table apachelog we created on the previous problem:

hive> LOAD DATA INPATH '/user/hadoop/problem2/access\_log\_1' INTO TABLE apachelog;

Loading data to table default.apachelog

OK

Time taken: 0.8 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem2/access\_log\_2' INTO TABLE apachelog;

Loading data to table default.apachelog

OK

Time taken: 0.417 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem2/access\_log\_3' INTO TABLE apachelog;

Loading data to table default.apachelog

OK

Time taken: 0.732 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem2/access\_log\_4' INTO TABLE apachelog;

Loading data to table default.apachelog

OK

Time taken: 0.63 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem2/access\_log\_5' INTO TABLE apachelog;

Loading data to table default.apachelog

OK

Time taken: 0.529 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem2/access\_log\_6' INTO TABLE apachelog;

Loading data to table default.apachelog

OK

Time taken: 0.497 seconds

We query the size of the table and verify that it has 239346 rows:

hive> select count(\*) from apachelog;

Total MapReduce 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 mapred.reduce.tasks=<number>

Starting Job = job\_201303072219\_0003, Tracking URL = http://domU-12-31-39-00-64-46.compute-1.internal:9100/jobdetails.jsp?jobid=job\_201303072219\_0003

Kill Command = /home/hadoop/bin/hadoop job -Dmapred.job.tracker=10.254.107.180:9001 -kill job\_201303072219\_0003

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1

(...)

MapReduce Total cumulative CPU time: 15 seconds 400 msec

Ended Job = job\_201303072219\_0003

Counters:

MapReduce Jobs Launched:

Job 0: Map: 2 Reduce: 1 Accumulative CPU: 15.4 sec HDFS Read: 53235622 HDFS Write: 7 SUCCESS

Total MapReduce CPU Time Spent: 15 seconds 400 msec

OK

239346

Time taken: 62.402 seconds

Which is consistent with the sum of the number of lines on the log files added to the table plus the previous 2 we had before:

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cat s3n://elasticmapreduce/samples/pig-apache/input/access\_log\_1 | wc -l

13/03/07 23:35:12 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_1' for reading

39344

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cat s3n://elasticmapreduce/samples/pig-apache/input/access\_log\_2 | wc -l

13/03/07 23:36:20 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_2' for reading

40000

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cat s3n://elasticmapreduce/samples/pig-apache/input/access\_log\_3 | wc -l

13/03/07 23:57:58 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_3' for reading

40000

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cat s3n://elasticmapreduce/samples/pig-apache/input/access\_log\_4 | wc -l

13/03/07 23:58:38 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_4' for reading

40000

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cat s3n://elasticmapreduce/samples/pig-apache/input/access\_log\_5 | wc -l

13/03/07 23:58:57 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_5' for reading

40000

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cat s3n://elasticmapreduce/samples/pig-apache/input/access\_log\_6 | wc -l

13/03/07 23:59:24 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_6' for reading

40000

**Problem 3) You recall that we said that hive stores all values in its tables in a HDFS directory. Locate that directory and present a small portion of the file containing table apachelog.**

Once we locate the configuration file, we can query it to find the directory where the table values are being stored.

hadoop@domU-12-31-39-00-64-46:~$ cat /home/hadoop/.versions/hive-0.8.1/conf/hive-default.xml | grep warehouse

<name>hive.metastore.warehouse.dir</name>

<value>/mnt/hive\_081/warehouse</value>

<description>location of default database for the warehouse</description>

Once at this directory, it is easy to locate the correct table location and query the directory listing, where we find the original files we added.

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -ls /mnt/hive\_081/warehouse

Found 1 items

drwxr-xr-x - hadoop supergroup 0 2013-03-07 23:19 /mnt/hive\_081/warehouse/apachelog

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -ls /mnt/hive\_081/warehouse/apachelog

Found 8 items

-rw-r--r-- 1 hadoop supergroup 8754118 2013-03-07 22:59 /mnt/hive\_081/warehouse/apachelog/access\_log\_1

-rw-r--r-- 1 hadoop supergroup 8902171 2013-03-07 22:59 /mnt/hive\_081/warehouse/apachelog/access\_log\_2

-rw-r--r-- 1 hadoop supergroup 8896201 2013-03-07 22:59 /mnt/hive\_081/warehouse/apachelog/access\_log\_3

-rw-r--r-- 1 hadoop supergroup 8886636 2013-03-07 22:59 /mnt/hive\_081/warehouse/apachelog/access\_log\_4

-rw-r--r-- 1 hadoop supergroup 8902365 2013-03-07 22:59 /mnt/hive\_081/warehouse/apachelog/access\_log\_5

-rw-r--r-- 1 hadoop supergroup 8892828 2013-03-07 22:59 /mnt/hive\_081/warehouse/apachelog/access\_log\_6

-rw-r--r-- 1 hadoop supergroup 219 2013-03-07 22:41 /mnt/hive\_081/warehouse/apachelog/apache.access.2.log

-rw-r--r-- 1 hadoop supergroup 86 2013-03-07 22:41 /mnt/hive\_081/warehouse/apachelog/apache.access.log

Getting the first 10 lines from the first file on the directory, we verify that the files are stored on their original formatting in the table directory.

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cat /mnt/hive\_081/warehouse/apachelog/access\_log\_1 | head -10

66.249.67.3 - - [20/Jul/2009:20:12:22 -0700] "GET /gallery/main.php?g2\_controller=exif.SwitchDetailMode&g2\_mode=detailed&g2\_return=%2Fgallery%2Fmain.php%3Fg2\_itemId%3D15741&g2\_returnName=photo HTTP/1.1" 302 5 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.67.3 - - [20/Jul/2009:20:12:25 -0700] "GET /gallery/main.php?g2\_itemId=15741&g2\_fromNavId=x8fa12efc HTTP/1.1" 200 8068 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

64.233.172.17 - - [20/Jul/2009:20:12:26 -0700] "GET /gwidgets/alexa.xml HTTP/1.1" 200 2969 "-" "Mozilla/5.0 (compatible) Feedfetcher-Google; (+http://www.google.com/feedfetcher.html)"

74.125.74.193 - - [20/Jul/2009:20:13:01 -0700] "GET /gwidgets/alexa.xml HTTP/1.1" 200 2969 "-" "Mozilla/5.0 (compatible) Feedfetcher-Google; (+http://www.google.com/feedfetcher.html)"

192.168.1.198 - - [20/Jul/2009:20:13:18 -0700] "GET / HTTP/1.1" 200 17935 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10\_5\_7; en-us) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17"

192.168.1.198 - - [20/Jul/2009:20:13:18 -0700] "GET /style.css HTTP/1.1" 200 1504 "http://example.org/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10\_5\_7; en-us) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17"

192.168.1.198 - - [20/Jul/2009:20:13:19 -0700] "GET /favicon.ico HTTP/1.1" 404 146 "http://example.org/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10\_5\_7; en-us) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17"

66.249.67.3 - - [20/Jul/2009:20:13:21 -0700] "GET /gallery/main.php?g2\_controller=exif.SwitchDetailMode&g2\_mode=detailed&g2\_return=%2Fgallery%2Fmain.php%3Fg2\_itemId%3D30893&g2\_returnName=photo HTTP/1.1" 302 5 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.67.3 - - [20/Jul/2009:20:13:24 -0700] "GET /gallery/main.php?g2\_itemId=30893&g2\_fromNavId=xfc647d65 HTTP/1.1" 200 8196 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.67.3 - - [20/Jul/2009:20:13:29 -0700] "GET /gallery/main.php?g2\_view=search.SearchScan&g2\_form%5BuseDefaultSettings%5D=1&g2\_return=%2Fgallery%2Fmain.php%3Fg2\_itemId%3D15789&g2\_returnName=photo HTTP/1.1" 200 6360 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

**Problem 4) Change the name of your table apachelog to apachelogold and create new table apachelog which will support partitioning. Load each of apache\_access\_log\_x files into a separate partition. Find out how many rows you have in each partition.**

First, we alter the original table to the new name and verify that the old table name is gone.

hive> ALTER TABLE apachelog RENAME TO apachelogold;

OK

Time taken: 16.639 seconds

hive> show tables;

OK

apachelogold

Time taken: 0.888 seconds

And then we create the new apachelog table with the partitioning option on a new variable called idx.

hive> CREATE TABLE apachelog ( host STRING,

> identity STRING,

> user STRING,

> time STRING,

> request STRING,

> status STRING,

> size STRING,

> referer STRING,

> agent STRING)

> partitioned by (idx 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;

OK

Time taken: 0.399 seconds

Adding the log files to different partitions.

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -mkdir problem4

hadoop@domU-12-31-39-00-64-46:~$ hadoop fs -cp s3n://elasticmapreduce/samples/pig-apache/input/\* /user/hadoop/problem4/

13/03/08 01:18:54 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_1' for reading

13/03/08 01:19:03 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_2' for reading

13/03/08 01:19:10 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_3' for reading

13/03/08 01:19:18 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_4' for reading

13/03/08 01:19:27 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_5' for reading

13/03/08 01:19:35 INFO s3native.NativeS3FileSystem: Opening 's3n://elasticmapreduce/samples/pig-apache/input/access\_log\_6' for reading

----

hive> LOAD DATA INPATH '/user/hadoop/problem4/access\_log\_1' INTO TABLE apachelog PARTITION (idx='1');

Loading data to table default.apachelog partition (idx=1)

OK

Time taken: 0.863 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem4/access\_log\_2' INTO TABLE apachelog PARTITION (idx='2');

Loading data to table default.apachelog partition (idx=2)

OK

Time taken: 0.901 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem4/access\_log\_3' INTO TABLE apachelog PARTITION (idx='3');

Loading data to table default.apachelog partition (idx=3)

OK

Time taken: 1.005 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem4/access\_log\_4' INTO TABLE apachelog PARTITION (idx='4');

Loading data to table default.apachelog partition (idx=4)

OK

Time taken: 0.72 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem4/access\_log\_5' INTO TABLE apachelog PARTITION (idx='5');

Loading data to table default.apachelog partition (idx=5)

OK

Time taken: 0.619 seconds

hive> LOAD DATA INPATH '/user/hadoop/problem4/access\_log\_6' INTO TABLE apachelog PARTITION (idx='6');

Loading data to table default.apachelog partition (idx=6)

OK

Time taken: 0.589 seconds

And finally we query each partition to make sure that we have the expected number of rows in each (as per listed by each file on Problem 2).

hive> select count(\*) from apachelog where idx='1';

Total MapReduce jobs = 1

Launching Job 1 out of 1

(...)

Total MapReduce CPU Time Spent: 5 seconds 490 msec

OK

39344

Time taken: 55.359 seconds

-----

hive> select count(\*) from apachelog where idx='2';

Total MapReduce jobs = 1

Launching Job 1 out of 1

(...)

Total MapReduce CPU Time Spent: 5 seconds 740 msec

OK

40000

Time taken: 53.064 seconds

-----

hive> select count(\*) from apachelog where idx='3';

Total MapReduce jobs = 1

Launching Job 1 out of 1

(...)

Total MapReduce CPU Time Spent: 5 seconds 550 msec

OK

40000

Time taken: 53.287 seconds

-----

hive> select count(\*) from apachelog where idx='4';

Total MapReduce jobs = 1

Launching Job 1 out of 1

(...)

Total MapReduce CPU Time Spent: 5 seconds 200 msec

OK

40000

Time taken: 46.256 seconds

-----

hive> select count(\*) from apachelog where idx='5';

Total MapReduce jobs = 1

Launching Job 1 out of 1

(...)

Total MapReduce CPU Time Spent: 5 seconds 570 msec

OK

40000

Time taken: 51.848 seconds

-----

hive> select count(\*) from apachelog where idx='6';

Total MapReduce jobs = 1

Launching Job 1 out of 1

(...)

Total MapReduce CPU Time Spent: 5 seconds 660 msec

OK

40000

Time taken: 52.067 seconds

**Problem 5) Export data from one of above partitions to an OS file. Examine and display the first 10 rows of the new file.**

Exporting one of the partitions to a local directory called problem5:

hive> INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/problem5' SELECT \* FROM apachelog WHERE idx='1';

Total MapReduce jobs = 1

Launching Job 1 out of 1

Number of reduce tasks is set to 0 since there's no reduce operator

Starting Job = job\_201303072219\_0010, Tracking URL = http://domU-12-31-39-00-64-46.compute-1.internal:9100/jobdetails.jsp?jobid=job\_201303072219\_0010

Kill Command = /home/hadoop/bin/hadoop job -Dmapred.job.tracker=10.254.107.180:9001 -kill job\_201303072219\_0010

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0

2013-03-08 01:43:46,321 Stage-1 map = 0%, reduce = 0%

2013-03-08 01:43:56,396 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.4 sec

2013-03-08 01:43:57,404 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.4 sec

2013-03-08 01:43:58,414 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.4 sec

2013-03-08 01:43:59,423 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.4 sec

2013-03-08 01:44:00,433 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.4 sec

2013-03-08 01:44:01,442 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.4 sec

2013-03-08 01:44:02,451 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.4 sec

MapReduce Total cumulative CPU time: 4 seconds 400 msec

Ended Job = job\_201303072219\_0010

Counters:

Copying data to local directory /home/hadoop/problem5

Copying data to local directory /home/hadoop/problem5

39344 Rows loaded to /home/hadoop/problem5

MapReduce Jobs Launched:

Job 0: Map: 1 Accumulative CPU: 4.4 sec HDFS Read: 8754350 HDFS Write: 8832806 SUCCESS

Total MapReduce CPU Time Spent: 4 seconds 400 msec

OK

Time taken: 39.249 seconds

Checking the first 10 rows of the file that was written to the local environment:

hadoop@domU-12-31-39-00-64-46:~$ head -10 /home/hadoop/problem5/000000\_0

66.249.67.3--[20/Jul/2009:20:12:22 -0700]"GET /gallery/main.php?g2\_controller=exif.SwitchDetailMode&g2\_mode=detailed&g2\_return=%2Fgallery%2Fmain.php%3Fg2\_itemId%3D15741&g2\_returnName=photo HTTP/1.1"3025"-""Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"1

66.249.67.3--[20/Jul/2009:20:12:25 -0700]"GET /gallery/main.php?g2\_itemId=15741&g2\_fromNavId=x8fa12efc HTTP/1.1"2008068"-""Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"1

64.233.172.17--[20/Jul/2009:20:12:26 -0700]"GET /gwidgets/alexa.xml HTTP/1.1"2002969"-""Mozilla/5.0 (compatible) Feedfetcher-Google; (+http://www.google.com/feedfetcher.html)"1

74.125.74.193--[20/Jul/2009:20:13:01 -0700]"GET /gwidgets/alexa.xml HTTP/1.1"2002969"-""Mozilla/5.0 (compatible) Feedfetcher-Google; (+http://www.google.com/feedfetcher.html)"1

192.168.1.198--[20/Jul/2009:20:13:18 -0700]"GET / HTTP/1.1"20017935"-""Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10\_5\_7; en-us) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17"1

192.168.1.198--[20/Jul/2009:20:13:18 -0700]"GET /style.css HTTP/1.1"2001504"http://example.org/""Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10\_5\_7; en-us) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17"1

192.168.1.198--[20/Jul/2009:20:13:19 -0700]"GET /favicon.ico HTTP/1.1"404146"http://example.org/""Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10\_5\_7; en-us) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17"1

66.249.67.3--[20/Jul/2009:20:13:21 -0700]"GET /gallery/main.php?g2\_controller=exif.SwitchDetailMode&g2\_mode=detailed&g2\_return=%2Fgallery%2Fmain.php%3Fg2\_itemId%3D30893&g2\_returnName=photo HTTP/1.1"3025"-""Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"1

66.249.67.3--[20/Jul/2009:20:13:24 -0700]"GET /gallery/main.php?g2\_itemId=30893&g2\_fromNavId=xfc647d65 HTTP/1.1"2008196"-""Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"1

66.249.67.3--[20/Jul/2009:20:13:29 -0700]"GET /gallery/main.php?g2\_view=search.SearchScan&g2\_form%5BuseDefaultSettings%5D=1&g2\_return=%2Fgallery%2Fmain.php%3Fg2\_itemId%3D15789&g2\_returnName=photo HTTP/1.1"2006360"-""Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"1

**Problem 6) You noticed how Hive provides you with the kill command for every job that you run. Our jobs are not very long, still demonstrate that you can kill one of them. You need to open another terminal window to do that.**

Initially, we run a relatively long job, which calculate the row count on the full apachelog table.

hive> select count(\*) from apachelog;

Total MapReduce jobs = 1

Launching Job 1 out of 1

(...)

Starting Job = job\_201303072219\_0011, Tracking URL = http://domU-12-31-39-00-64-46.compute-1.internal:9100/jobdetails.jsp?jobid=job\_201303072219\_0011

Kill Command = **/home/hadoop/bin/hadoop job -Dmapred.job.tracker=10.254.107.180:9001 -kill job\_201303072219\_0011**

Hadoop job information for Stage-1: number of mappers: 6; number of reducers: 1

(...)

Then we run the kill command on a different window.

hadoop@domU-12-31-39-00-64-46:~$ /home/hadoop/bin/hadoop job -Dmapred.job.tracker=10.254.107.180:9001 -kill job\_201303072219\_0011

Killed job job\_201303072219\_0011

And we can see the running job on Hive being cancelled after a few seconds.

Ended Job = job\_201303072219\_0011 with errors

Error during job, obtaining debugging information...

Examining task ID: task\_201303072219\_0011\_m\_000007 (and more) from job job\_201303072219\_0011

Counters:

FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.MapRedTask

MapReduce Jobs Launched:

Job 0: Map: 6 Reduce: 1 Accumulative CPU: 20.14 sec HDFS Read: 53235711 HDFS Write: 0 FAIL

Total MapReduce CPU Time Spent: 20 seconds 140 msec