INFSCI 2750: Cloud Computing
Mini Project 3
Deadline: April 27, 2017, 11:59 pm

Part 1: Setting up Cassandra

root@slave: /var/lib/cassandra/data — -ssh root@104.236.235.155 — 121×39

```
ion=-1}, ColumnDefinition{name=session_id, type=org.apache.cassandra.db.marshal.UUIDType, kind=PARTITION_KEY, position=0}
, ColumnDefinition{name=thread, type=org.apache.cassandra.db.marshal.UTF8Type, kind=REGULAR, position=-1}, ColumnDefiniti
on{name=event_id, type=org.apache.cassandra.db.marshal.TimeUUIDType, kind=CLUSTERING, position=0}, ColumnDefinition{name=
source, type=org.apache.cassandra.db.marshal.InetAddressType, kind=REGULAR, position=-1}, ColumnDefinition{name=source_el
apsed, type=org.apache.cassandra.db.marshal.Int32Type, kind=REGULAR, position=-1}],droppedColumns={},triggers=[],indexes=
[]]], views=[], functions=[], types=[]}
INFO  15:36:11 Handshaking version with master/104.236.235.155
INFO  15:36:11 Node /104.236.235.155 is now part of the cluster
INFO  15:36:11 InetAddress /104.236.235.155 is now UP
INFO  15:36:11 Not submitting build tasks for views in keyspace system_traces as storage service is not initialized
INFO  15:36:11 Updating topology for /104.236.235.155
INFO  15:36:11 Updating topology for /104.236.235.155
```

root@master: ~ — -ssh root@104.236.235.155 — 119×37

```
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

37 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Thu Apr 20 15:34:11 2017 from 150.212.8.73
root@master:~# nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load       Tokens       Owns (effective)  Host ID                               Rack
UN  104.236.235.155  219.25 KiB  256          100.0%            26735a12-e7e2-4abf-8161-3bd79f7496ff  rack1
UN  104.236.234.130  140.29 KiB  256          100.0%            97f7746d-02f4-43a5-9890-6014ae2bf0f7  rack1

root@master:~# cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh>
cqlsh>
root@master:~#
root@master:~# cqlsh master
Connected to Test Cluster at master:9042.
[cqlsh 5.0.1 | Cassandra 3.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh>
root@master:~# cqlsh slave
Connected to Test Cluster at slave:9042.
[cqlsh 5.0.1 | Cassandra 3.6 | CQL spec 3.4.2 | Native protocol v4]
Use HELP for help.
cqlsh>
```

for now the Cassandra is in the cluster mode.

Part 2: Import data into Cassandra

#first we should convert the access log file into modified csv file

newtable.csv

```
"1","10.223.157.186","/"
"2","10.223.157.186","/favicon.ico"
"3","10.223.157.186","/"
"4","10.223.157.186","/assets/js/lowpro.js"
"5","10.223.157.186","/assets/css/reset.css"
"6","10.223.157.186","/assets/css/960.css"
"7","10.223.157.186","/assets/css/the-associates.css"
"8","10.223.157.186","/assets/js/the-associates.js"
"9","10.223.157.186","/assets/js/lightbox.js"
"10","10.223.157.186","/assets/img/search-button.gif"
"11","10.223.157.186","/assets/img/dummy/secondary-news-3.jpg"
"12","10.223.157.186","/assets/img/dummy/primary-news-1.jpg"
"13","10.223.157.186","/assets/img/dummy/primary-news-2.jpg"
"14","10.223.157.186","/assets/img/closelabel.gif"
"15","10.223.157.186","/assets/img/home-logo.png"
"16","10.223.157.186","/assets/img/dummy/secondary-news-2.jpg"
"17","10.223.157.186","/assets/img/loading.gif"
"18","10.223.157.186","/assets/img/dummy/secondary-news-4.jpg"
"19","10.223.157.186","/assets/img/home-media-block-placeholder.jpg"
"20","10.223.157.186","/assets/img/dummy/secondary-news-1.jpg"
"21","10.223.157.186","/assets/swf/home-media-block.swf"
```

Before we start to import data, we should open the Cassandra service.

1. Open a terminal and type ‘cassandra -Rf’
2. Then open another new terminal and type ‘cqlsh –request-time 600 localhost’

#then to import the data in the Cassandra, we should create at least one keyspace and one table

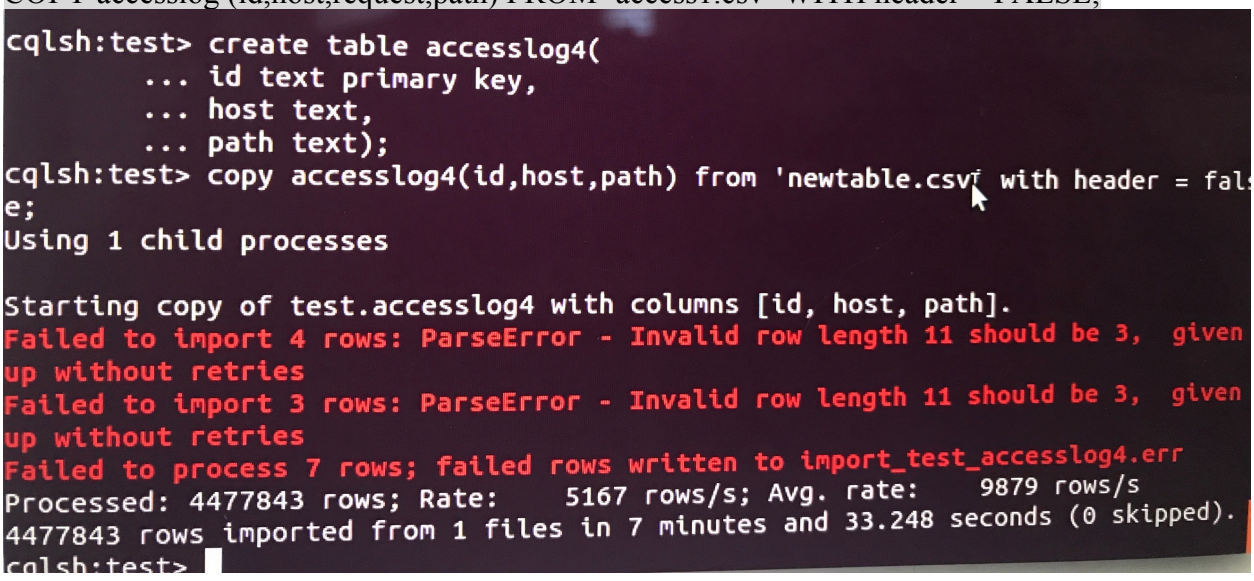
```
CREATE KEYSPACE test WITH REPLICATION = {
    ‘class’ “ ‘SImpleStrategy’,
    ‘replication’ = 3  };

use test;
```

#then create a table called accesslog in keyspace test.

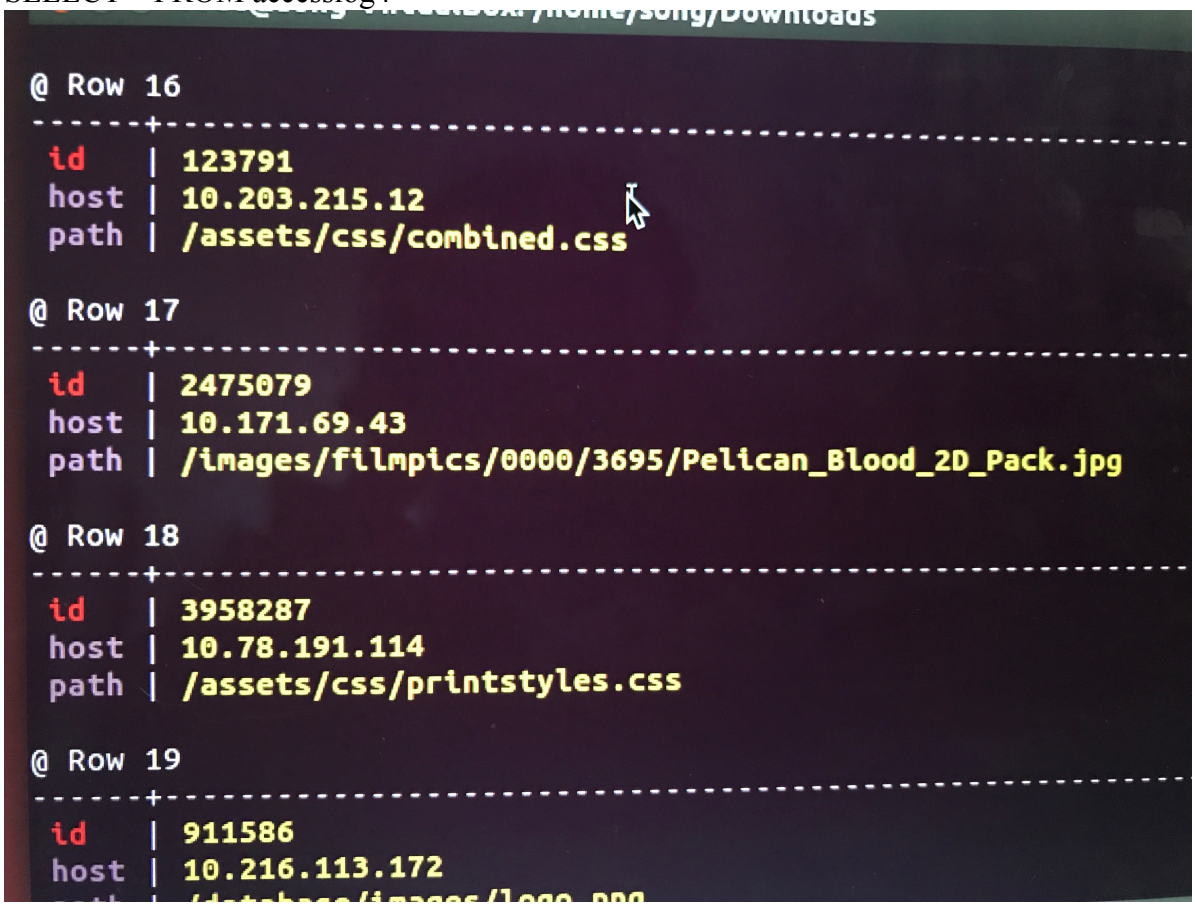
```
CREATE TABLE accesslog(
    Id text PRIMARY KEY,
    host text,
   path text );
```

#then we copy the data from access1.csv to the table accesslog

```
COPY accesslog (id,host,request,path) FROM ‘access1.csv’ WITH header = FALSE;
```



the figure above is the screenshot of the sample imported data in accesslog table.
SELECT * FROM accesslog4

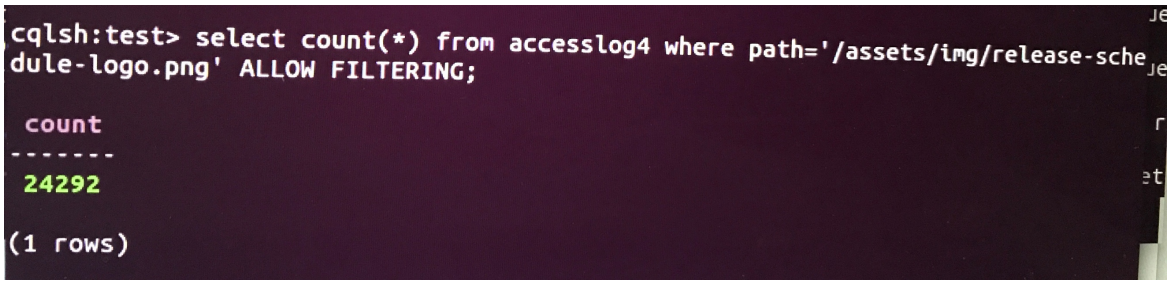


Part3: Operate data in Cassandra

1. how many hit were made to the website item ‘/assests/img/release-schedulelogo.png’

SELECT COUNT(*) from accesslog4 WHERE path=’ ‘/assests/img/release-schedulelogo.png’

The result: there are 24292 hits



2.how many hits were made from the host ’10.207.188.188’

SELECT COUNT(*) FROM accesslog WHERE host=’10.207.188.188’

The result: there are 398 hits

```
cqlsh:test> select count(*) from accesslog where host='10.207.188.188'
        ... ;

 count
-------
   398

(1 rows)
```

3. which path in the website has been hit the most and how many hits were made to the path?
   Fot the groupby function, there are no groupby in the Cassandra, so in this part, we should create User Defined Function to implement the groupby and return the max count.

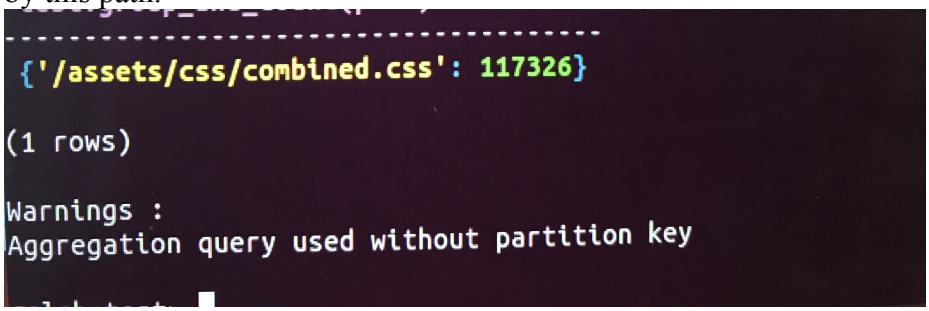
```
CREATE OR REPLACE FUNCTION group_count( state map<text, int>, type text)
CALLED ON NULL INPUT
RETURNS map<text, int>
LANGUAGE java AS    '
Integer count = (Integer) state.get(type);
If(count==null) count=1;
else
count++;
state.put(type, count);
return state;    '    ;

CREATE OR REPLACE AGGREGATE group_and_count_max(text)
SFUNC group_count,
STYPE map<text, int>
INITICOND {};
```

Then we export the query result to the csv file as result1.csv
And then we create a new table to import the result1
Then we use maxAgg(count) to find the maximum count

```
#find the maximum count
CREATE FUNCTION maxI(current int,candidate int)
CALLED  ON NULL INPUT
RETURN int LANGUAGE java AS
' if (current==null) return candidate; else return Math.max(current, candidate);';

CREATE AGGREGATE maxAgg(int)
SFUNC maxI
STYPE int
INITCOND null;
```

Then we execute this aggregate function in the cqlsh with

SELECT maxAgg(count) FROM result1;
The result is showed below:
The path '/assets/css/combind.css' has been hit the most there are 117348 hits were made by this path.



4. Which IP access the website most and how many access were made by it?
   Same with the question 3 use the same aggregate function to find the IP.
   We still need to export a query result to the new csv file result2
   SELECT maxAgg(count) FROM result2;
   the result is showed below:
   the '10.216.113.172' access the website most and there are 158614 were made by this IP address