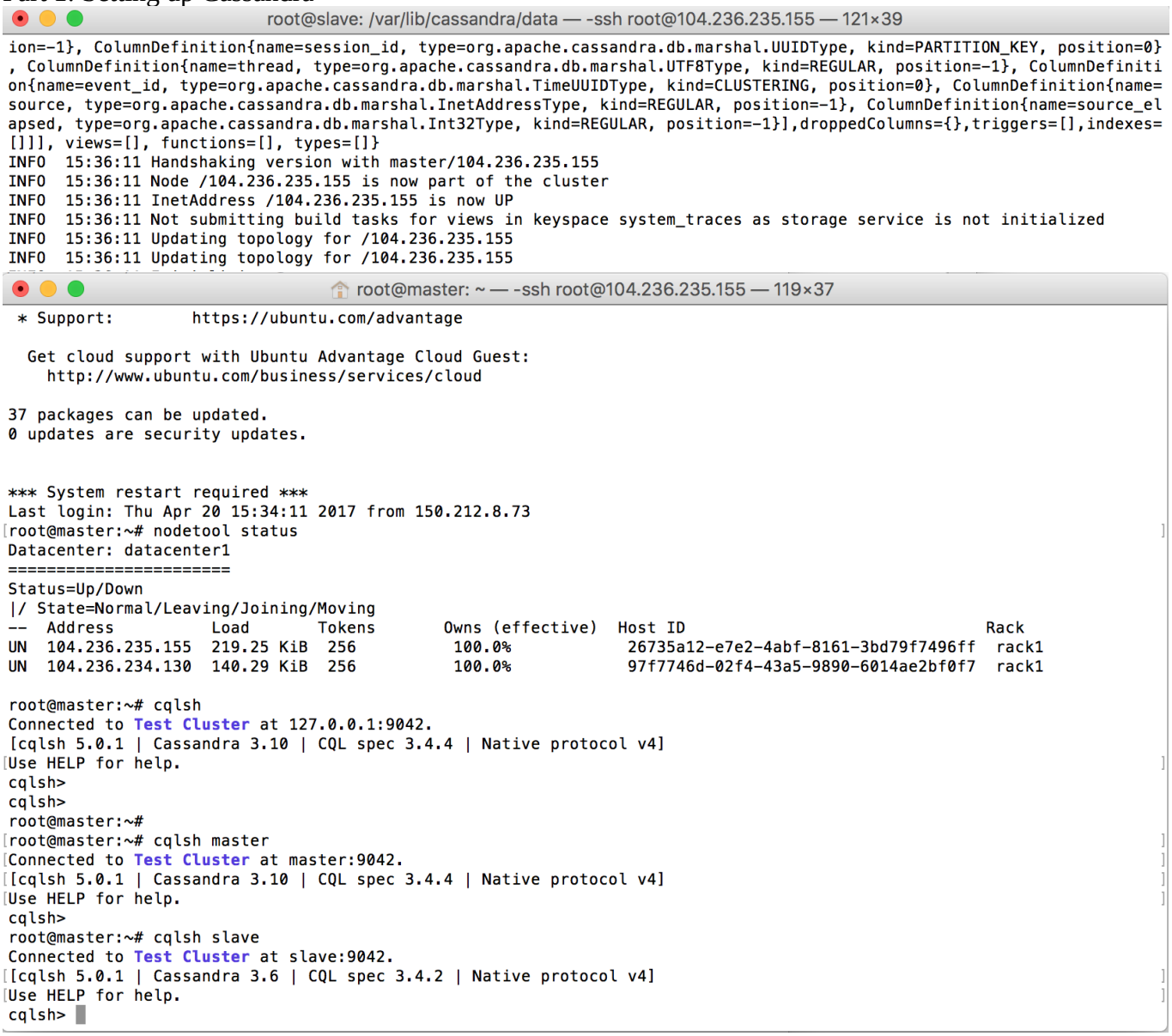
Cloud Computing

Part 1: Setting up Cassandra



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

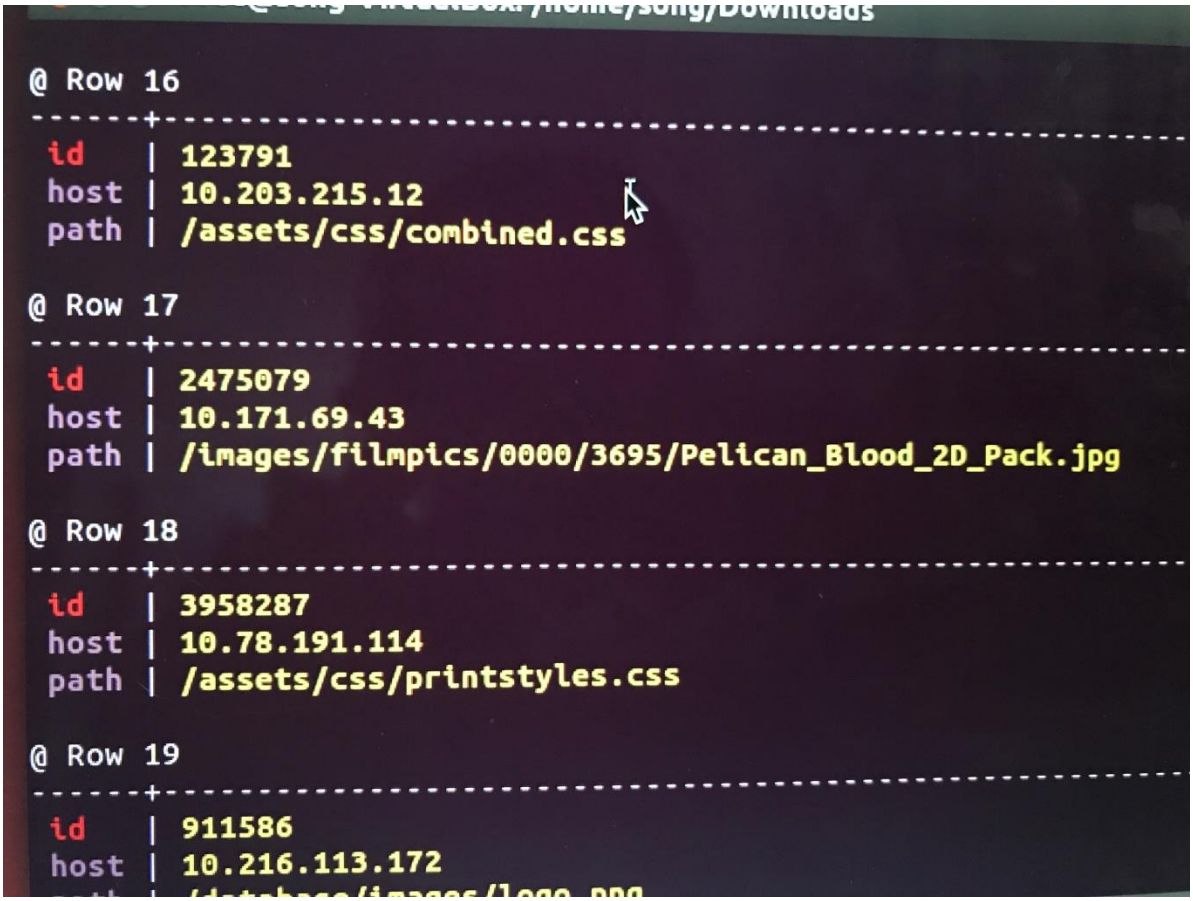
```
COPY accesslog (id,host,request,path) FROM ‘access1.csv’ WITH header = FALSE;
```

```
cqlsh:test> create table accesslog4(
        ... id text primary key,
        ... host text,
        ... path text);
cqlsh:test> copy accesslog4(id,host,path) from 'newtable.csv' with header = fal
e;
Using 1 child processes

Starting copy of test.accesslog4 with columns [id, host, path].
Failed to import 4 rows: ParseError - Invalid row length 11 should be 3,  given
up without retries
Failed to import 3 rows: ParseError - Invalid row length 11 should be 3,  given
up without retries
Failed to process 7 rows; failed rows written to import_test_accesslog4.err
Processed: 4477843 rows; Rate:    5167 rows/s; Avg. rate:    9879 rows/s
4477843 rows imported from 1 files in 7 minutes and 33.248 seconds (0 skipped).
cqlsh:test>
```

the figure above is the screenshot of the sample imported data in accesslog table.

```
SELECT * FROM accesslog4
```

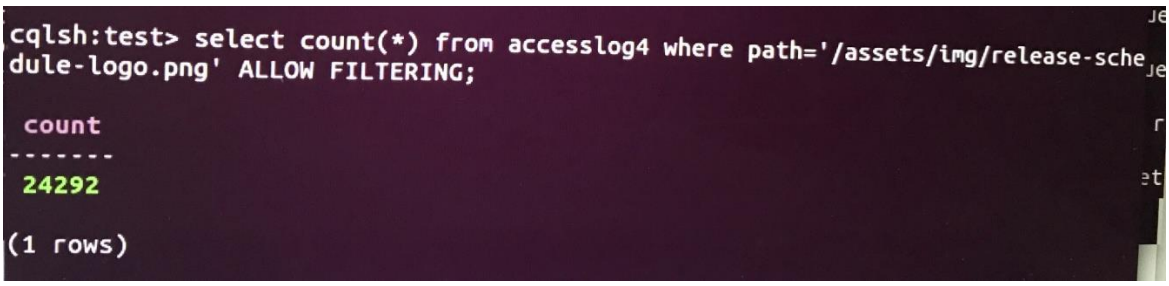


Part3: Operate data in Cassandra

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’

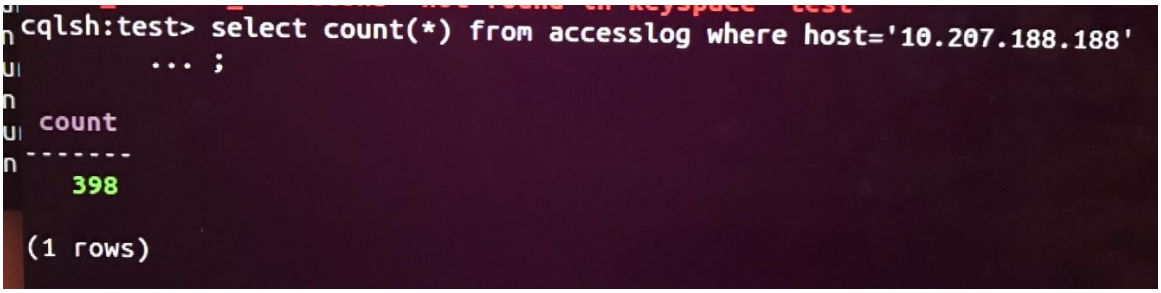
The result: there are 24292 hits



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



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 {};
```

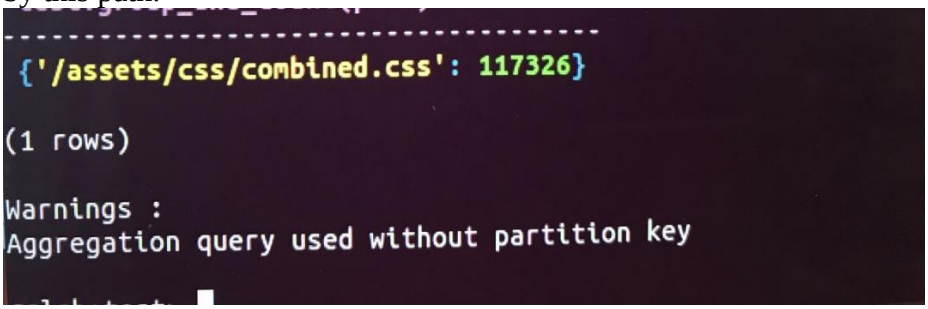
   Then we export the query result to the csv file as result1.csv
   And then we create a new table to import the result1
   Then we use maxAgg(count) to find the maximum count

```
#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
   SELECTmaxAgg(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