# Assignment 2

Jing Luo B00722667
Zewen Wang B00756586

We worked in a group to finish this assignment.
Zewen created the keysapce and the table and uploaded the data to Cassandra and performed four basic queries.
Jing created UI application and restful service, loaded balancer and tested on JMeter.
Zewen and Jing wrote the report together.

## a. Task Description

### 1.Task description and requirements

We divided the task into 5 parts as following.
(1) Create a Cassandra cluster on https://www.instaclustr.com/managed/cassandra/:
We connect to Cassandra via CQLSH and upload the dataset (The dataset can be downloaded from https://www.opendataphilly.org/dataset/crime-incidents)
Then, we created a keyspace called “test” and a table called “crime”. After that, four queries are performed.
(2) Develop UI application and restful services:
We applied HTML and javascript to implement the UI application and restful services.
(3) Create two instances on AWS
Install JRE and tomcat 8 on both instances.
Upload web files to tomcat on both instances.
(4) Create a load balancer:
Add the two created instances to the load balancer.
(5) Test using JMeter:
Download JMeter and run “sh jmeter.sh” to open JMeter. Then, test the load balancer by simulating to send multiple requests at the same time.

### 2. Application scenario

This application can be used by the police to study crime behavior and arrange more people for patrolling in the places where crime occurs frequently. The user interface of the web application consists of three parts, one for particular attributes with values, one for queried attributes and one for submit button. After specifying the first two parts, the police can click the submit button to get the corresponding result in the next page.

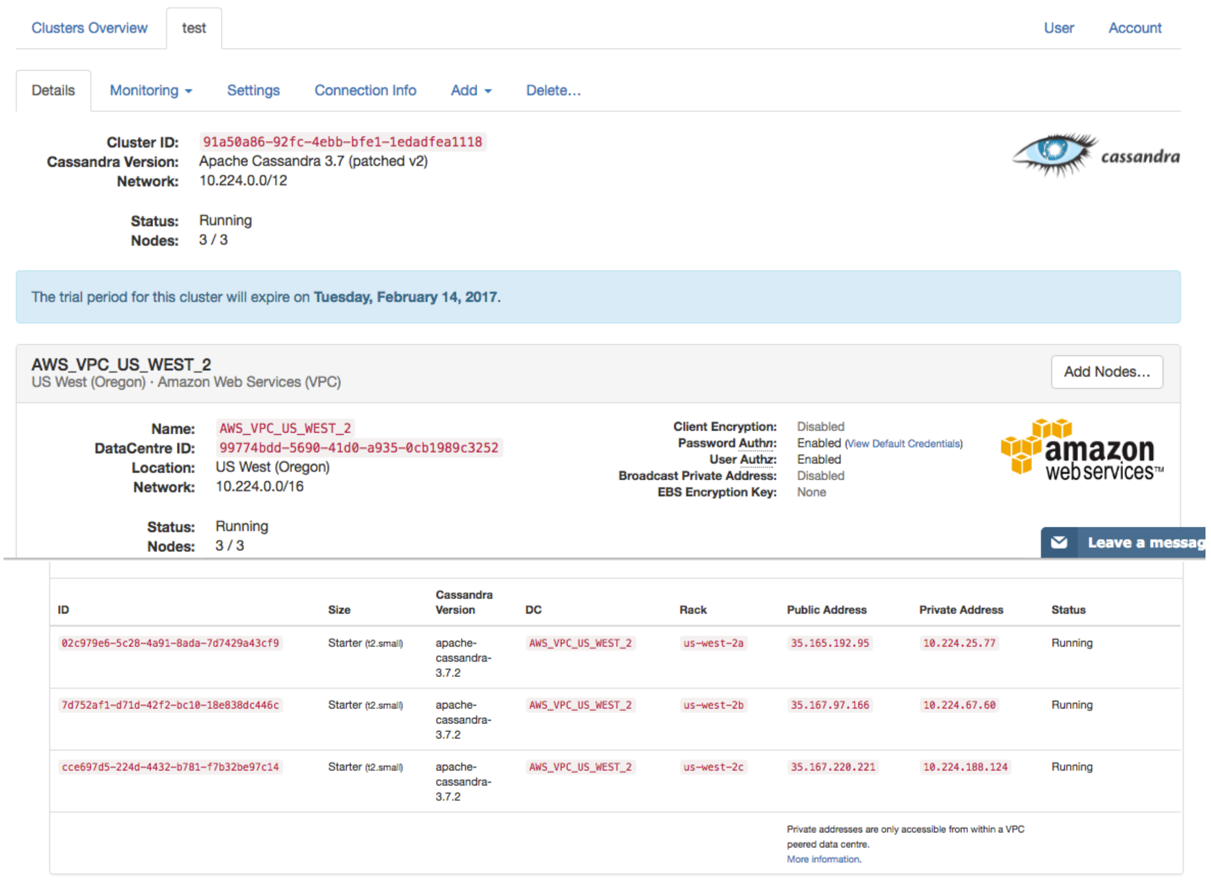
### 3.Dataset description

There are fourteen attributes and more than two million records in the original dataset. The primary key is the attribute, Dc_Key. We preprocessed the dataset by deleting “Dispatch_Date_Time” and “Month” columns because these information is provided by other attributes.
An instance of database:
12,P,2010-11-04,10:58:00,10,199812085407,1400 BLOCK S 58TH ST,1700,Other Sex Offenses (Not Commercialized),POINT (-75.235061 39.941442),9

## b. Cassandra Database Design

### 1. Configuration steps to setup Cassandra database on a cloud

(0)Go to [https://console.instaclustr.com/user/login](https://console.instaclustr.com/user/login) to create a cluster.



Cluster ID: 91a50a86-92fc-4ebb-bfe1-1edadfea1118
Cassandra Version: Apache Cassandra 3.7 (patched v2)
Network: 10.224.0.0/12

Status: Running
Nodes: 3 / 3

The trial period for this cluster will expire on **Tuesday, February 14, 2017.**

AWS_VPC_US_WEST_2
US West (Oregon) · Amazon Web Services (VPC)

Name: AWS_VPC_US_WEST_2
DataCentre ID: 99774bdd-5690-41d0-a935-0cb1989c3252
Location: US West (Oregon)
Network: 10.224.0.0/16

Status: Running
Nodes: 3 / 3

Client Encryption: Disabled
Password Authn: Enabled (View Default Credentials)
User Authz: Enabled
Broadcast Private Address: Disabled
EBS Encryption Key: None

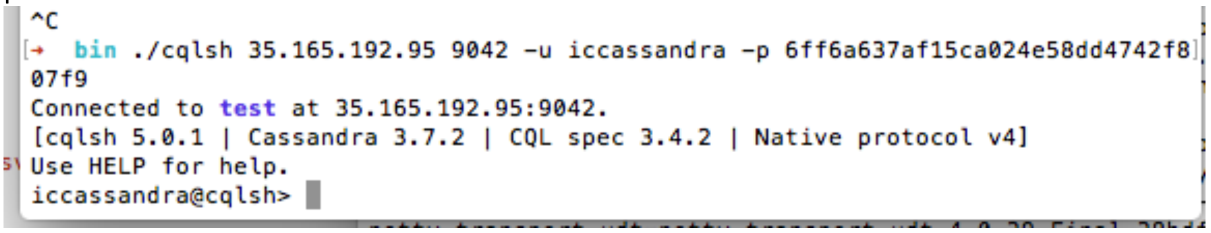
| ID | Size | Cassandra Version | DC | Rack | Public Address | Private Address | Status |
|---|---|---|---|---|---|---|---|
| 02c979e6-5c28-4a91-8ada-7d7429a43cf9 | Starter (t2.small) | apache-cassandra-3.7.2 | AWS_VPC_US_WEST_2 | us-west-2a | 35.165.192.95 | 10.224.25.77 | Running |
| 7d752af1-d71d-42f2-bc10-18e838dc446c | Starter (t2.small) | apache-cassandra-3.7.2 | AWS_VPC_US_WEST_2 | us-west-2b | 35.167.97.166 | 10.224.67.60 | Running |
| cce697d5-224d-4432-b781-f7b32be97c14 | Starter (t2.small) | apache-cassandra-3.7.2 | AWS_VPC_US_WEST_2 | us-west-2c | 35.167.220.221 | 10.224.188.124 | Running |

Private addresses are only accessible from within a VPC peered data centre.
More information.

The username and password of the cluster:
username: iccassandra
password: 6ff6a637af15ca024e58dd4742f807f9
(1) Download apache-cassandra from this website http://cassandra.apache.org/

(2) Open terminal and go to "bin" of Cassandra
run ./Cassandra -f

(3) Use the command "./cqlsh 35.165.192.95 9042 -u iccassandra -p 6ff6a637af15ca024e58dd4742f807f9" to connect to Cassandra

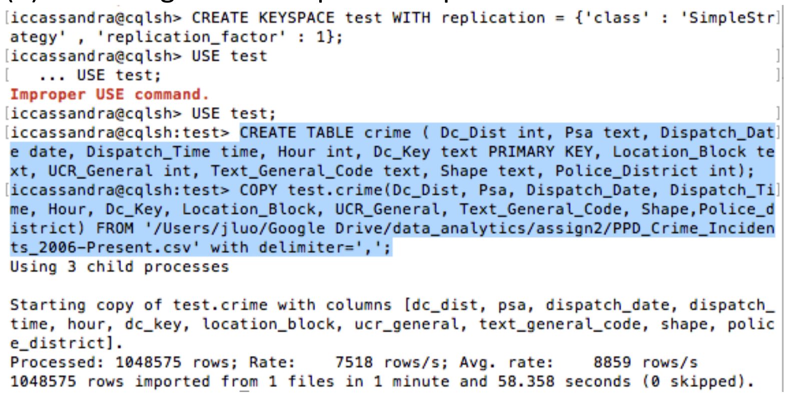


```
^C
→  bin ./cqlsh 35.165.192.95 9042 -u iccassandra -p 6ff6a637af15ca024e58dd4742f8
07f9
Connected to test at 35.165.192.95:9042.
[cqlsh 5.0.1 | Cassandra 3.7.2 | CQL spec 3.4.2 | Native protocol v4]
Use HELP for help.
iccassandra@cqlsh>
```

(4) Create keyspace and table using crime.csv.
CREATE KEYSPACE test WITH replication = {'class' : 'SimpleStrategy' , 'replication_factor' : 1};
CREATE TABLE crime ( Dc_Dist int, Psa text, Dispatch_Date date, Dispatch_Time time, Hour int, Dc_Key text PRIMARY KEY, Location_Block text, UCR_General int, Text_General_Code text, Shape text, Police_District int);

(5)Use established keyspace and table and upload the data on the cloud.
iccassandra@cqlsh:test> COPY test.crime(Dc_Dist, Psa, Dispatch_Date, Dispatch_Time, Hour, Dc_Key, Location_Block, UCR_General, Text_General_Code, Shape,Police_district) FROM '/Users/jluo/Google Drive/data_analytics/assign2/PPD_Crime_Incidents_2006-Present.csv' with delimiter=',';

(6)The configuration steps and implementation results can be seen from the picture below.



```
iccassandra@cqlsh> CREATE KEYSPACE test WITH replication = {'class' : 'SimpleStr
ategy' , 'replication_factor' : 1};
iccassandra@cqlsh> USE test
   ... USE test;
Improper USE command.
iccassandra@cqlsh> USE test;
iccassandra@cqlsh:test> CREATE TABLE crime ( Dc_Dist int, Psa text, Dispatch_Dat
e date, Dispatch_Time time, Hour int, Dc_Key text PRIMARY KEY, Location_Block te
xt, UCR_General int, Text_General_Code text, Shape text, Police_District int);
iccassandra@cqlsh:test> COPY test.crime(Dc_Dist, Psa, Dispatch_Date, Dispatch_Ti
me, Hour, Dc_Key, Location_Block, UCR_General, Text_General_Code, Shape,Police_d
istrict) FROM '/Users/jluo/Google Drive/data_analytics/assign2/PPD_Crime_Inciden
ts_2006-Present.csv' with delimiter=',';
Using 3 child processes

Starting copy of test.crime with columns [dc_dist, psa, dispatch_date, dispatch_
time, hour, dc_key, location_block, ucr_general, text_general_code, shape, polic
e_district].
Processed: 1048575 rows; Rate:    7518 rows/s; Avg. rate:    8859 rows/s
1048575 rows imported from 1 files in 1 minute and 58.358 seconds (0 skipped).
```

**2. Describe the attributes:**
After preprocessing the dataset, there are 12 attributes remained.

| Attritutes | Meaning |
|---|---|
| Dc_Key | Document Key of the crime, Primary key. The first six numbers are the combination of the year and dc_dist of the crime |
| Dc_Dist | |
| Psa | One police district is divided into several regions. |
| Dispatch_Date | The particular date when the crime occurred |
| Dispatch_Time | The accurate time when the crime occurred |
| Hour | What hour when the crime occurred |
| Location_Block | The address of the crime |
| UCR_General | The code number for the corresponding crime type |
| Text_General_Code | Crime Type |
| Police_Districts | Which police district corresponded the location of the crime |
| Lon | The longitude when the crime occurred |
| Lat | The latitude when the crime occurred |

## c. Application Queries:

**Query1:**
(1)Explanation of the query:
Find out the crime type for the crime case whose document key is 2006041826.
(2)Query and query result for CQL

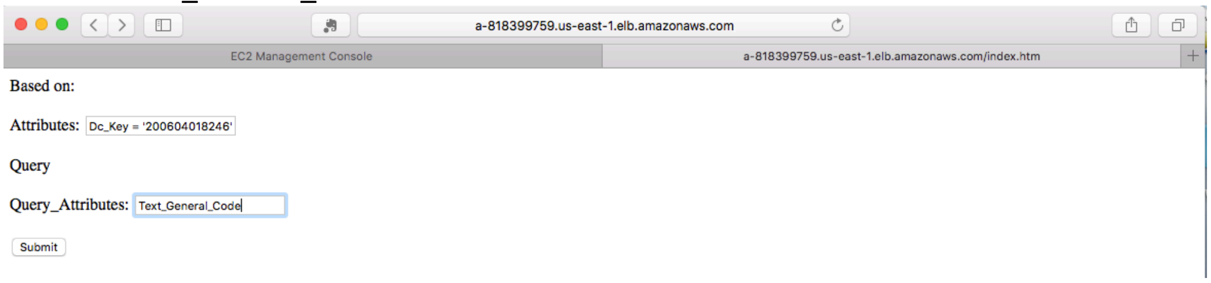
- CQL syntax of the queries :
  SELECT Text_General_Code FROM test.crime WHERE Dc_Key = '200604018246';
- Results returned from the query via cqlsh:

```
iccassandra@cqlsh:test> SELECT Text_General_Code FROM test.crime WHERE Dc_Key = '200604018246';

 text_general_code
---------------------------
 Aggravated Assault Firearm

(1 rows)
```

(3) Query and query result for web application
This query is based on the condition “Dc_Key = '200604018246'” and the query attribute for result is “Text_General_Code”.



a-818399759.us-east-1.elb.amazonaws.com
EC2 Management Console | a-818399759.us-east-1.elb.amazonaws.com/index.htm

Based on:

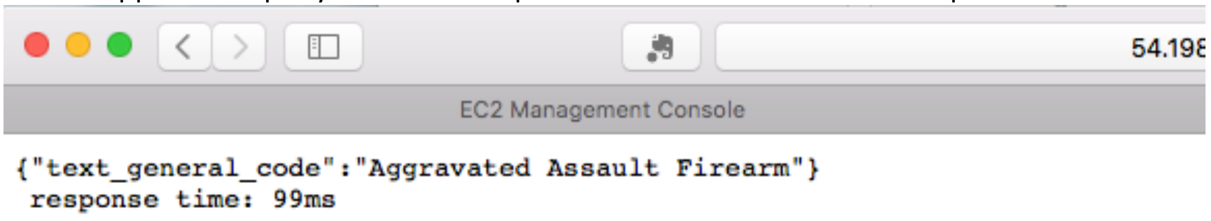
Attributes: Dc_Key = '200604018246'

Query

Query_Attributes: Text_General_Code

Submit

- Application query result and response time can be seen from the picture below



54.198
EC2 Management Console

```
{"text_general_code":"Aggravated Assault Firearm"}
 response time: 99ms
```

(4) Service endpoint URL of web services
http://54.198.203.19:8081/process_get?Attribute=Dc_Key+%3D+%27200604018246%27&Query_Attribute=Text_General_Code+

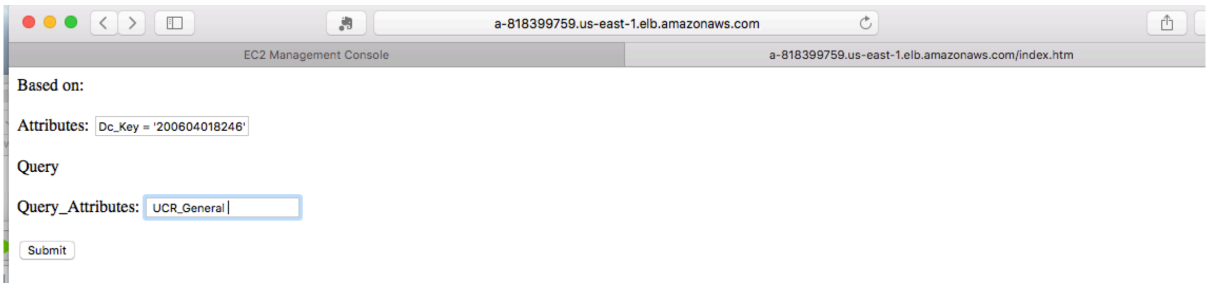
**Query 2:**
(1)explanation of the query:
Fine out the code number for the crime case whose document key is 2006041826.
(2) Query and query result for CQL
- CQL syntax of the queries :
  SELECT UCR_General FROM test.crime WHERE Dc_Key = '200604018246';
- Results returned from the query via cqlsh:

```
iccassandra@cqlsh:test> SELECT UCR_General FROM test.crime WHERE Dc_Key = '200604018246';

 ucr_general
-------------
         400

(1 rows)
```

(3)Query and query result for web application
This query is based on the condition “Dc_Key = '200604018246'” and the query attribute for result is “UCR_General”.


a-818399759.us-east-1.elb.amazonaws.com
EC2 Management Console
a-818399759.us-east-1.elb.amazonaws.com/index.htm
Submit

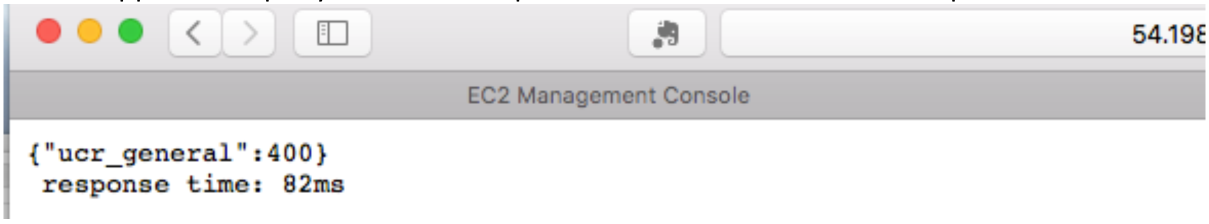
Based on:

Attributes: Dc_Key = '200604018246'

Query

Query_Attributes: UCR_General

- Application query result and response time can be seen from the picture below

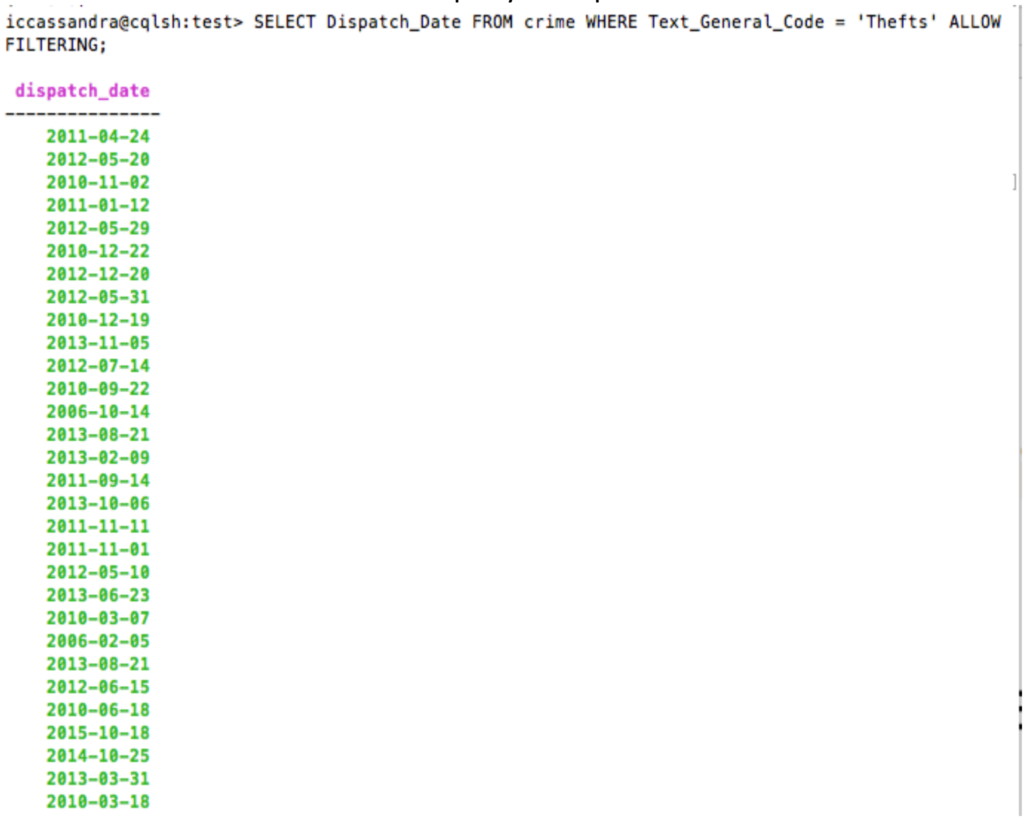


```
{"ucr_general":400}
 response time: 82ms
```

(4) Service endpoint URL of web services
http://54.198.203.19:8081/process_get?Attribute=Dc_Key+%3D+%27200604018246%27&Query_Attribute=UCR_General+

**Query3:**
(1) Explanation of the query:
List all the dates when thefts occurred.
(2) Query and query result for CQL

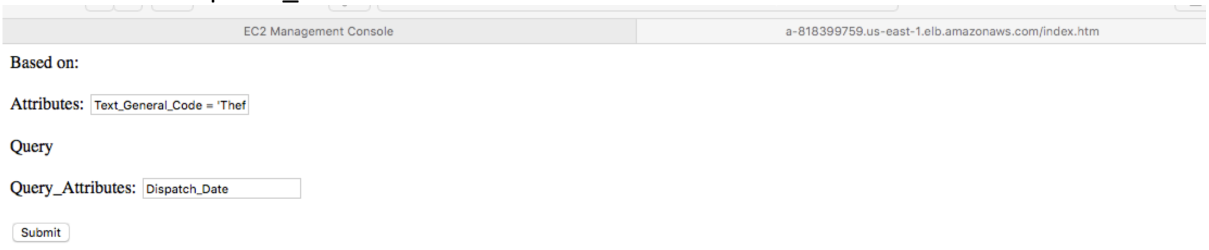
- CQL syntax of the queries :
  SELECT Dispatch_Date FROM crime WHERE Text_General_Code = 'Thefts' ALLOW FILTERING;
- Results returned from the query via cqlsh:



```
iccassandra@cqlsh:test> SELECT Dispatch_Date FROM crime WHERE Text_General_Code = 'Thefts' ALLOW
FILTERING;

 dispatch_date
---------------
    2011-04-24
    2012-05-20
    2010-11-02
    2011-01-12
    2012-05-29
    2010-12-22
    2012-12-20
    2012-05-31
    2010-12-19
    2013-11-05
    2012-07-14
    2010-09-22
    2006-10-14
    2013-08-21
    2013-02-09
    2011-09-14
    2013-10-06
    2011-11-11
    2011-11-01
    2012-05-10
    2013-06-23
    2010-03-07
    2006-02-05
    2013-08-21
    2012-06-15
    2010-06-18
    2015-10-18
    2014-10-25
    2013-03-31
    2010-03-18
```

(3)Query and query result for web application
This query is based on the condition “Text_General_Code = 'Thefts'” and the query attribute for result is “Dispatch_Date”.



EC2 Management Console | a-818399759.us-east-1.elb.amazonaws.com/index.htm

Based on:

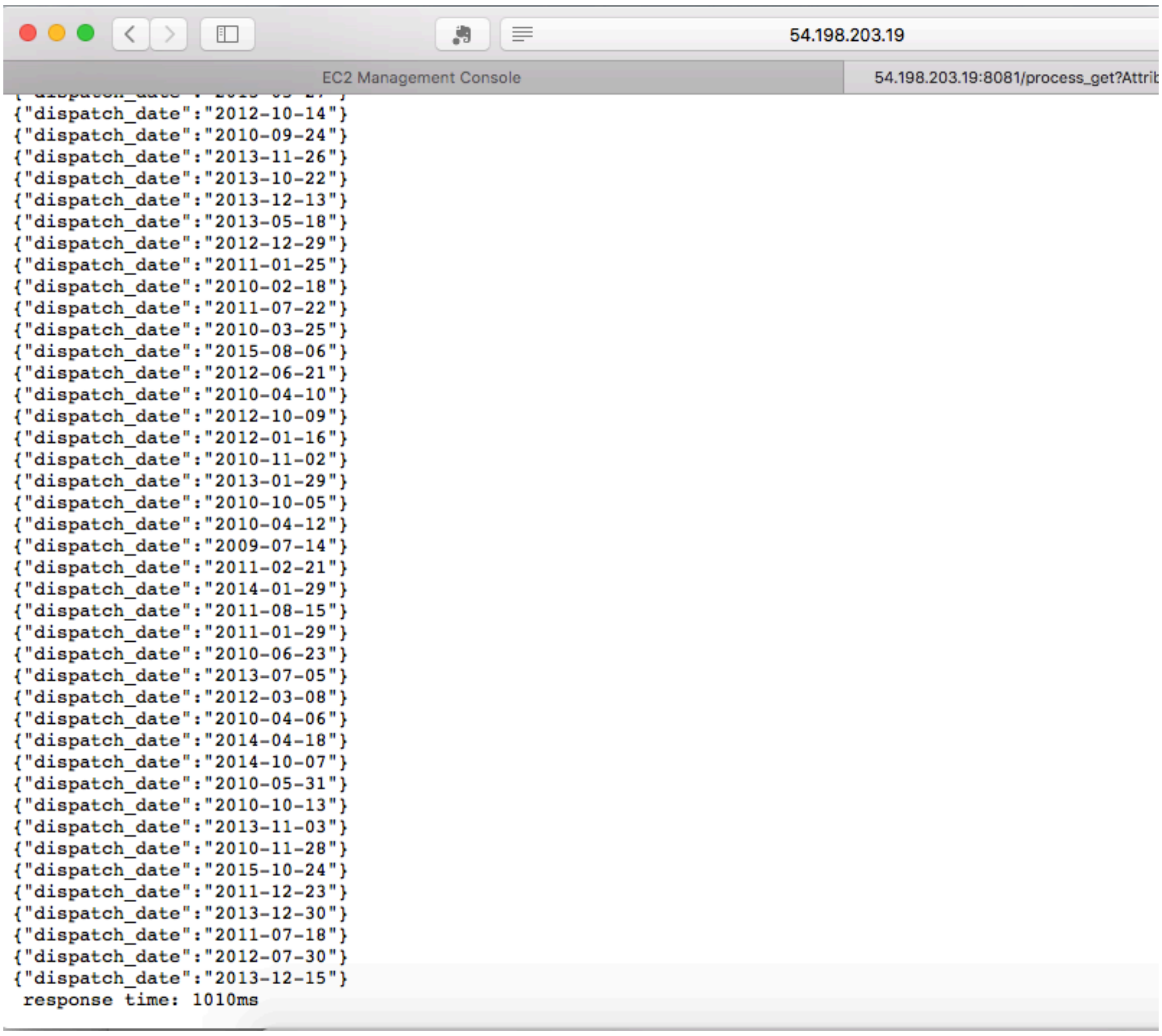
Attributes: Text_General_Code = 'Thef

Query

Query_Attributes: Dispatch_Date

Submit

- Application query result and response time can be seen from the picture below



54.198.203.19

EC2 Management Console | 54.198.203.19:8081/process_get?Attrib

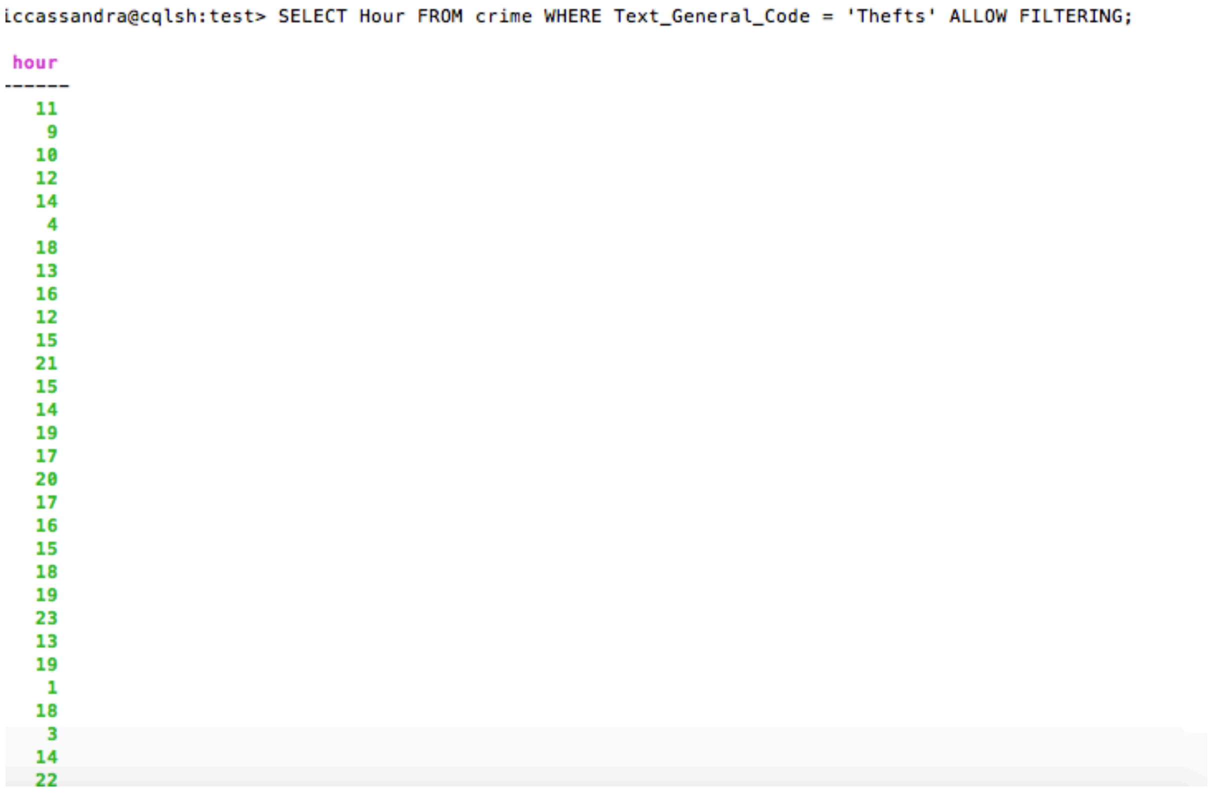
```
{"dispatch_date":"2012-10-14"}
{"dispatch_date":"2010-09-24"}
{"dispatch_date":"2013-11-26"}
{"dispatch_date":"2013-10-22"}
{"dispatch_date":"2013-12-13"}
{"dispatch_date":"2013-05-18"}
{"dispatch_date":"2012-12-29"}
{"dispatch_date":"2011-01-25"}
{"dispatch_date":"2010-02-18"}
{"dispatch_date":"2011-07-22"}
{"dispatch_date":"2010-03-25"}
{"dispatch_date":"2015-08-06"}
{"dispatch_date":"2012-06-21"}
{"dispatch_date":"2010-04-10"}
{"dispatch_date":"2012-10-09"}
{"dispatch_date":"2012-01-16"}
{"dispatch_date":"2010-11-02"}
{"dispatch_date":"2013-01-29"}
{"dispatch_date":"2010-10-05"}
{"dispatch_date":"2010-04-12"}
{"dispatch_date":"2009-07-14"}
{"dispatch_date":"2011-02-21"}
{"dispatch_date":"2014-01-29"}
{"dispatch_date":"2011-08-15"}
{"dispatch_date":"2011-01-29"}
{"dispatch_date":"2010-06-23"}
{"dispatch_date":"2013-07-05"}
{"dispatch_date":"2012-03-08"}
{"dispatch_date":"2010-04-06"}
{"dispatch_date":"2014-04-18"}
{"dispatch_date":"2014-10-07"}
{"dispatch_date":"2010-05-31"}
{"dispatch_date":"2010-10-13"}
{"dispatch_date":"2013-11-03"}
{"dispatch_date":"2010-11-28"}
{"dispatch_date":"2015-10-24"}
{"dispatch_date":"2011-12-23"}
{"dispatch_date":"2013-12-30"}
{"dispatch_date":"2011-07-18"}
{"dispatch_date":"2012-07-30"}
{"dispatch_date":"2013-12-15"}
 response time: 1010ms
```

(4) Service endpoint URL of web services
http://54.198.203.19:8081/process_get?Attribute=Text_General_Code+%3D+%27Thefts%27+&Query_Attribute=Dispatch_Date+

**Query 4:**
(1)Explanation of the query:
List all the hour when thefts occurred.
(2) Query and query result for CQL
- CQL syntax of the queries :

SELECT Hour FROM crime WHERE Text_General_Code = 'Thefts' ALLOW FILTERING;

- Results returned from the query via cqlsh:

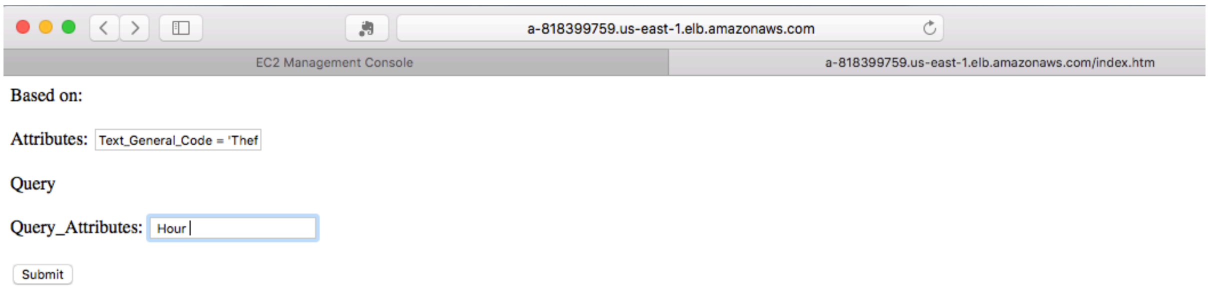


```
iccassandra@cqlsh:test> SELECT Hour FROM crime WHERE Text_General_Code = 'Thefts' ALLOW FILTERING;

 hour
------
   11
    9
   10
   12
   14
    4
   18
   13
   16
   12
   15
   21
   15
   14
   19
   17
   20
   17
   16
   15
   18
   19
   23
   13
   19
    1
   18
    3
   14
   22
```

(3)Query and query result for web application

This query is based on the condition “Text_General_Code = 'Thefts'” and the query attribute for result is “Hour”.



a-818399759.us-east-1.elb.amazonaws.com
EC2 Management Console
a-818399759.us-east-1.elb.amazonaws.com/index.htm

Based on:

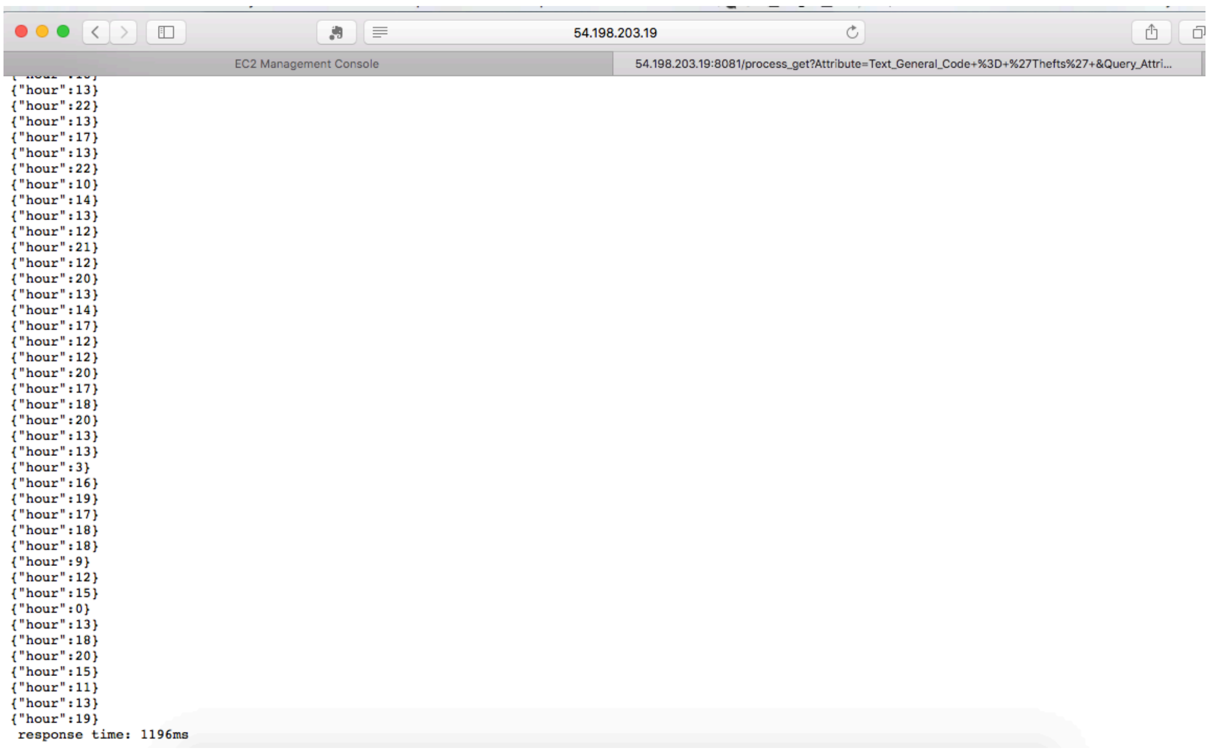
Attributes: Text_General_Code = 'Thef

Query

Query_Attributes: Hour

Submit

- Application query result and response time can be seen from the picture below


54.198.203.19
EC2 Management Console
54.198.203.19:8081/process_get?Attribute=Text_General_Code+%3D+%27Thefts%27+&Query_Attri...

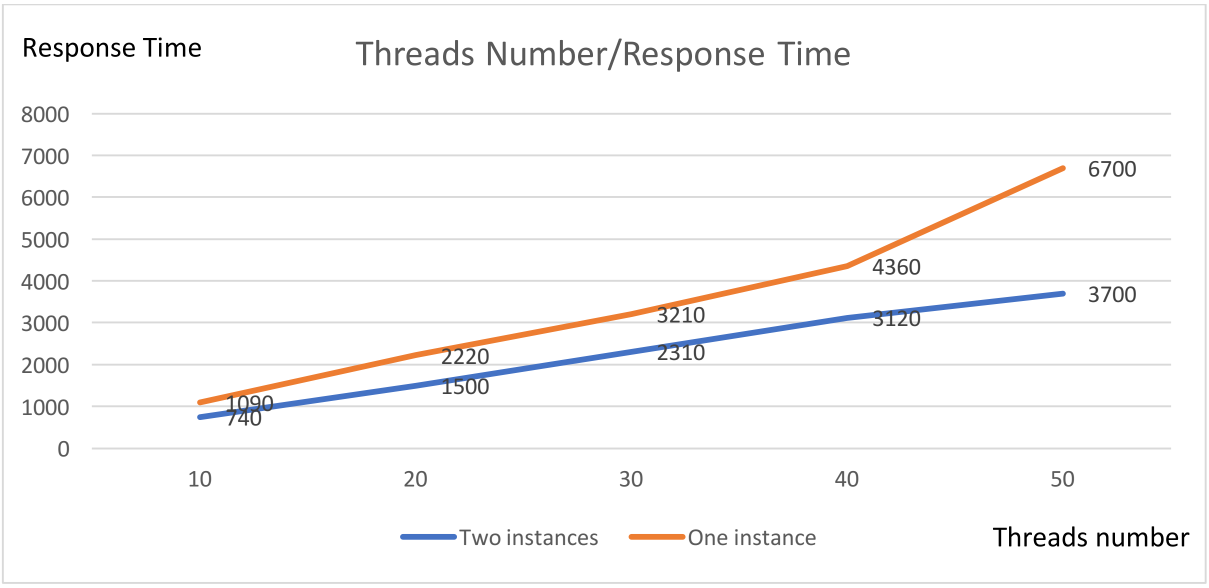
```
{"hour":13}
{"hour":22}
{"hour":13}
{"hour":17}
{"hour":13}
{"hour":22}
{"hour":10}
{"hour":14}
{"hour":13}
{"hour":12}
{"hour":21}
{"hour":12}
{"hour":20}
{"hour":13}
{"hour":14}
{"hour":17}
{"hour":12}
{"hour":12}
{"hour":20}
{"hour":17}
{"hour":18}
{"hour":20}
{"hour":13}
{"hour":13}
{"hour":3}
{"hour":16}
{"hour":19}
{"hour":17}
{"hour":18}
{"hour":18}
{"hour":9}
{"hour":12}
{"hour":15}
{"hour":0}
{"hour":13}
{"hour":18}
{"hour":20}
{"hour":15}
{"hour":11}
{"hour":13}
{"hour":19}
 response time: 1196ms
```

(4) service endpoint URL of web services
http://54.198.203.19:8081/process_get?Attribute=Text_General_Code+%3D+%27Thefts%27+&Query_Attribute=+Hour+

**Balancer DNS: a-818399759.us-east-1.elb.amazonaws.com**
**Instance 1:**
DNS: ec2-54-198-203-19.compute-1.amazonaws.com
IPv4 Public IP:54.198.203.19
**Instance 2:**
DNS: ec2-52-202-48-81.compute-1.amazonaws.com
IPv4 Public IP:52.202.48.81

## d. Test Results

The number of threads tested is 10, 20, 30, 40 and 50.
Test1: Launch exact the same two instances on the cloud
Test2: Launch only one instance on the could
The graph below shows the number of threads and the corresponding response time for both one instance and two instances with load balancer respectively.

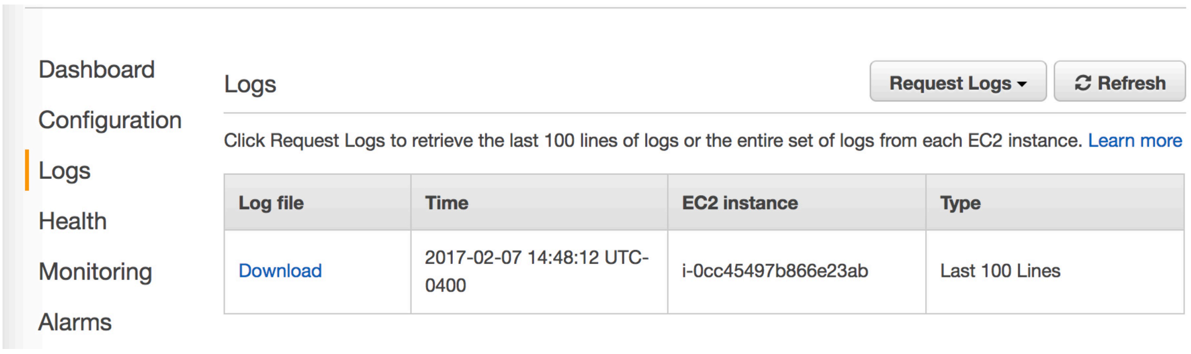


As can be seen from the chart above, we obtained two observations over these tests. As the number of the number of threads increases, the response time goes up as well. In addition, we found that with the help of load balancer, the response time of two instances case is shorter than one instance case no matter of the number of requests. It testified that load balancer can reasonably assign workload to different instances so as to relieve the traffic and get better performance and the more requests there are, the more obvious the effect of load balancer is.

## e. Summary

1.Before we uploaded our application on the web service, we had managed to test that application on localhost, which was convenient for coding.

2.If there are something wrong with the codes, we can also check the log file to find out where the errors are. The log files can be download from AWS where the picture below shows.



Click Request Logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. Learn more

| Log file | Time | EC2 instance | Type |
|---|---|---|---|
| Download | 2017-02-07 14:48:12 UTC-0400 | i-0cc45497b866e23ab | Last 100 Lines |

3. Actually, we also tried to use DevCenter to implement cloud dataset, but this tool seemed to occupy a lot of space of CPU and could slow the computer down. However, the benefit of this tool is that the query result can be shown in a table which is more readable than that in the terminal.