INFSCI 2750 – Cloud Computing

Mini Project 3 – Working with Apache Cassandra

Charu Sreedharan (chs263@pitt.edu)

Lakshmi Ravichandran (lar146@pitt.edu)

Sanzil Madye (ssm59@pitt.edu)

Part 1 – Setting up Cassandra

Download and install Cassandra with Debian package in master node (159.65.43.106) and slave nodes (68.183.153.175, 68.183.53.213) using the following commands -

```
echo "deb http://www.apache.org/dist/cassandra/debian 311x main" \
| sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list
curl https://www.apache.org/dist/cassandra/KEYS | sudo apt-key add -
sudo apt-get update
sudo apt-get install Cassandra
```

Alter configuration fie (Cassandra.yaml) on every node

```
nano /etc/cassandra/cassandra.yaml
```

#Change in both master and slave nodes
- seeds: "master, slave1, slave2"

```
  # seeds is actually a comma-delimited list of addresses.
  # Ex: "<ip1>,<ip2>,<ip3>"
  - seeds: "159.65.43.106,68.183.153.175,68.183.53.213"
```

Increase the read timeout period
read_request_timeout_in_ms: 600000

master node
listen_address: CC-AM-12

```
# Setting listen_address to 0.0.0.0 is always wrong.
#
listen_address: 159.65.43.106
```

rpc_address: CC-AM-12

```
# set broadcast_rpc_address to a value other than 0.0.0.0.
#
# For security reasons, you should not expose this port to the internet.  Firewall it if needed.
rpc_address: 159.65.43.106
```

slave nodes
listen_address: CC-AM-13/CC-AM-14
rpc_address: CC-AM-13/CC-AM-14

Stop Cassandra service using –

service cassandra stop

```
student@CC-AM-12:/etc/cassandra$ service cassandra stop
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to stop 'cassandra.service'.
Authenticating as: ,,, (student)
Password:
==== AUTHENTICATION COMPLETE ===
```

Start Cassandra cluster using the following command on each node –

sudo Cassandra -Rf

```
22, 741346159395265493 8, 746840969381363418, 755693482782496012 3, 761341842319859029 7, 762476387736076741 0, 767175161784785825 7, 774049249897497840 9, 77585724
38503227, 789151854955775438 5, 796646752035064619, 802279249540942096 3, 802367768832055892, 817836859787478422 2, 827903598981193211 3, 828451530353069968 0, 8317
4607591643122, 835693564002198186 4, 841304482424264962 2, 851879644365911183 0, 851911314641570872 6, 855565552381453590 9, 881190142191974329 2, 883602589262458200
9008572171704011325, 902355757784769408, 916024582534291764 5, 916891976325332342, 921551522546846775 2]
INFO  [main] 2019-04-01 02:17:53,625 StorageService.java:1483 - JOINING: Finish joining ring
INFO  [main] 2019-04-01 02:17:53,646 SecondaryIndexManager.java:509 - Executing pre-join tasks for: CFS(Keyspace='patient', ColumnFamily='exam')
INFO  [main] 2019-04-01 02:17:53,704 StorageService.java:2327 - Node /159.65.43.106 state jump to NORMAL
INFO  [main] 2019-04-01 02:17:53,720 Gossiper.java:1684 - Waiting for gossip to settle...
INFO  [main] 2019-04-01 02:18:01,723 Gossiper.java:1715 - No gossip backlog; proceeding
INFO  [main] 2019-04-01 02:18:01,949 NativeTransportService.java:70 - Netty using native Epoll event loop
INFO  [main] 2019-04-01 02:18:02,014 Server.java:155 - Using Netty Version: [netty-buffer=netty-buffer-4.0.44.Final.452812a, netty-codec=netty-codec-4.0.44.Fin
-haproxy=netty-codec-haproxy-4.0.44.Final.452812a, netty-codec-http=netty-codec-http-4.0.44.Final.452812a, netty-codec-socks=netty-codec-socks-4.0.44.Final.452
y-common-4.0.44.Final.452812a, netty-handler=netty-handler-4.0.44.Final.452812a, netty-tcnative=netty-tcnative-1.1.33.Fork26.142ecbb, netty-transport=netty-tra
812a, netty-transport-native-epoll=netty-transport-native-epoll-4.0.44.Final.452812a, netty-transport-rxtx=netty-transport-rxtx-4.0.44.Final.452812a, netty-tra
port-sctp-4.0.44.Final.452812a, netty-transport-udt=netty-transport-udt-4.0.44.Final.452812a]
INFO  [main] 2019-04-01 02:18:02,015 Server.java:156 - Starting listening for CQL clients on /159.65.43.106:9042 (unencrypted)...
INFO  [main] 2019-04-01 02:18:02,080 CassandraDaemon.java:556 - Not starting RPC server as requested. Use JMX (StorageService->startRPCServer()) or nodetool (e
t
```

Run the below command in a new session to check status of the Cassandra cluster –

nodetool status

```
student@CC-AM-12: ~
student@CC-AM-12:~$ nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address         Load       Tokens       Owns (effective)  Host ID                               Rack
UN  68.183.53.213   539.49 MiB  256         100.0%            d34e2b71-7a7c-4c44-bde9-08768b8830c4  rack1
DN  68.183.153.175  121.12 MiB  256         100.0%            e4985be7-ab9a-499e-996a-f5bfc89e29dc  rack1
UN  159.65.43.106   534.75 MiB  256         100.0%            22df6076-d996-428e-b172-b00982821c51  rack1

student@CC-AM-12:~$
```

To start CQL shell on the cluster -

```
student@CC-AM-12:~$ cqlsh 159.65.43.106
Connected to Test Cluster at 159.65.43.106:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh>
```

```
student@CC-AM-12:~$ cqlsh 159.65.43.106  --request-timeout=600
Connected to Test Cluster at 159.65.43.106:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> SELECT cluster_name, listen_address FROM system.local;

 cluster_name | listen_address
--------------+----------------
 Test Cluster |  159.65.43.106

(1 rows)
cqlsh>
```

Part 2 – Importing access_log data into Cassandra

LoadAccessLog.java is the java file to import access_log data file into cassandra. The java file is uploaded in course web in the source code folder. The project is created as a maven project. In the pom.xml file we add the dependencies like datastax cassandra java driver.

```
<dependencies>
        <dependency>
          <groupId>com.datastax.cassandra</groupId>
          <artifactId>cassandra-driver-core</artifactId>
          <version>3.6.0</version>
        </dependency>

        <dependency>
         <groupId>com.datastax.cassandra</groupId>
         <artifactId>cassandra-driver-mapping</artifactId>
         <version>3.6.0</version>
        </dependency>

        <dependency>
          <groupId>org.slf4j</groupId>
          <artifactId>slf4j-api</artifactId>
          <version>1.7.25</version>
        </dependency>

        <dependency>
          <groupId>org.slf4j</groupId>
          <artifactId>slf4j-simple</artifactId>
          <version>1.7.25</version>
          <scope>test</scope>
        </dependency>

        <dependency>
          <groupId>com.google.guava</groupId>
          <artifactId>guava</artifactId>
          <version>21.0</version>
        </dependency>

        <dependency>
          <groupId>io.netty</groupId>
          <artifactId>netty-all</artifactId>
          <version>4.1.20.Final</version>
        </dependency>

        <dependency>
          <groupId>com.codahale.metrics</groupId>
          <artifactId>metrics-core</artifactId>
          <version>3.0.2</version>
        </dependency>
</dependencies>
```

In the java file, LoadFile (String filepath) loads the file access_log from the file location path /home/student/hadoop/input and returns a buffer reader object. CreateKeySpaceTable() creates keyspace miniproject3 and tables fulllog, ip, url using CQL create queries. InsertValuesToTables(Session curSession, BufferedReader buffRead) inserts values into the

above created tables by reading access_log file line by line and splits the attributes based on regular expression match. 256 outstanding asynchronous queries can be run at the same time, but it can be modified in the configuration and java files.

Then, export the project as a runnable jar file. To run the jar file in cassandra -

Java -jar MiniProject3.jar

```
student@CC-AM-12:~/hadoop/input$ java -jar MiniProject3.jar
java.io.BufferedReader@66d3c617
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinde
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder
inserted 20000
inserted 40000
inserted 60000
inserted 80000
inserted 100000
inserted 120000
inserted 140000
inserted 160000
inserted 180000
inserted 200000
inserted 220000
```

```
inserted 4240000
inserted 4260000
inserted 4280000
inserted 4300000
inserted 4320000
inserted 4340000
inserted 4360000
inserted 4380000
inserted 4400000
inserted 4420000
inserted 4440000
inserted 4460000
inserted 4477813
Total run time to load webserver log: 1459 seconds
```

Total run time to load the webserver log data is 1459 seconds

```
cqlsh:miniproject3> exit
student@CC-AM-12:~/hadoop/input$ clear
student@CC-AM-12:~/hadoop/input$ cqlsh 159.65.43.106
Connected to Test Cluster at 159.65.43.106:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> DESCRIBE keyspaces;

miniproject3  system_schema  system              system_traces
patient       system_auth    system_distributed

cqlsh> USE miniproject3 ;
cqlsh:miniproject3> DESCRIBE tables;

url  ip  fulllog
```

The resulting table in CQLSH shell after importing access_log data

```
cqlsh:miniproject3> SELECT * FROM fulllog LIMIT 10;

 id      | ip            | url                                                  | identity | method | protocol | size    | status | time                      | username
---------+---------------+------------------------------------------------------+----------+--------+----------+---------+--------+---------------------------+----------
 1792034 | 10.167.188.164 |             /images/filmpics/0000/0975/ShinjukuDVD2D.jpg |     null |    GET | HTTP/1.1 | 3107362 |    200 | 07/Jan/2011:09:49:37 -0800 |     null
 3607449 | 10.198.238.249 |         /images/newspics/0000/0373/Atrociousweb_thumb.jpg |     null |    GET | HTTP/1.1 |    null |    304 | 24/Aug/2011:02:25:04 -0700 |     null
  302602 |  10.163.155.84 |                                     /download.php?id=90 |     null |    GET | HTTP/1.1 | 1089536 |    200 | 08/Mar/2010:18:51:09 -0800 |     null
 3819940 | 10.142.203.173 |                                 /assets/css/combined.css |     null |    GET | HTTP/1.0 |    6112 |    200 | 16/Sep/2011:07:42:45 -0700 |     null
 2301876 |  10.205.174.82 |                                             /index.php |     null |    GET | HTTP/1.1 |   18931 |    200 | 12/Mar/2011:07:12:14 -0800 |     null
  531141 |     10.219.4.6 |           /images/filmmediablock/290/HelgiBj%C3%AErns.jpg |     null |    GET | HTTP/1.1 |  502868 |    200 | 05/May/2010:08:04:43 -0700 |     null
 3472067 | 10.115.218.237 |                         /assets/js/javascript_combined.js |     null |    GET | HTTP/1.1 |   20404 |    200 | 11/Aug/2011:15:17:44 -0700 |     null
 2119753 |   10.122.217.3 |           /images/filmpics/0000/4291/Monsters6_thumb.jpg |     null |    GET | HTTP/1.1 |   26737 |    200 | 21/Feb/2011:08:19:32 -0800 |     null
 1416569 |   10.82.64.235 | /images/filmpics/0000/2563/deadcert_20091114_0151crop.jpg |     null |    GET | HTTP/1.1 |  105006 |    200 | 24/Oct/2010:19:30:32 -0700 |     null
 1817764 | 10.103.214.246 |                                 /assets/css/combined.css |     null |    GET | HTTP/1.1 |    6112 |    200 | 12/Jan/2011:11:01:43 -0800 |     null

(10 rows)
```

```
cqlsh:miniproject3> SELECT * FROM ip LIMIT 10;

 ip              | count
-----------------+-------
 10.226.129.213 |     2
  10.207.147.18 |    13
  10.217.21.189 |    14
 10.142.189.149 |     1
 10.126.208.138 |     1
  10.232.73.246 |     1
   10.68.57.243 |    24
  10.140.232.61 |     1
  10.140.203.33 |     1
  10.10.191.185 |     1

(10 rows)
```

```
cqlsh:miniproject3> SELECT * FROM url LIMIT 10;

 url                                                                        | count
----------------------------------------------------------------------------+-------
 /database/fullDetails.php?height=600&modal=true&id=163&random=1306336880267 |     1
                                        /downloadSingle.php?id=2085&fid=345 |    48
                              /SH/shanghai/360_bid/3_etid/28_did/15_ps/1_stid/ |     1
                /images/filmpics/0000/2155/SBX481_InvisibleTarget_DVD_lge.jpg |    71
 /database/fullDetails.php?height=600&modal=true&id=134&random=1314117502102 |     1
                                          /release-schedule/?p=28&l=&rpp=10 |     1
                                              /assets/img/about-us-logo.png |  3157
                          /displaytitle.php?id=546%27%20aND%20%278%27%3D%278 |     1
                        /images/filmmediablock/295/TaiChiMaster_2DSleeve.jpg |    76
                         /2010/02/dead-wizard-always-wins/?replytocom=113343 |     2

(10 rows)
cqlsh:miniproject3>
```

Part 3: Operate Data in Cassandra

1. **Problem 1**

We got the result by running the CQL (Cassandra Query Language) query below in CQLSH:

Query:

SELECT count(*) FROM miniproject3.fulllog WHERE url='/assets/img/release-schedule-logo.png' ALLOW FILTERING;

```
student@CC-AM-12: ~
cqlsh:miniproject3> SELECT count(*) FROM miniproject3.fulllog WHERE url='/assets/img/release-schedule-logo.png' ALLOW FILTERING;

 count
-------
 24292

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:miniproject3>
```

As the screenshot shows, the website /assets/img/release-schedule-logo.png was accessed **24292** times.

2. **Problem 2**

We got the result by running the CQL query below in CQLSH:

Query:

SELECT * FROM ip where ip ='10.207.188.188';

```
student@CC-AM-12: ~
cqlsh:miniproject3> select * from ip  where ip = '10.207.188.188';

 ip             | count
----------------+-------
 10.207.188.188 |   398

(1 rows)
cqlsh:miniproject3>
```

As can be seen in the screenshot, **398** hits were made to the IP: 10.207.188.188

3. **Problem 3**

LogQuestion3.java file is the source code for Problem 3.

We used the JAVA driver of cassandra to do this question.

We can execute this program using the below command:

java -cp hadoop/input/cloud3-0.0.1-SNAPSHOT-jar-with-dependencies.jar com.pitt.cloudcomputing.LogQuestion3

```
student@CC-AM-12: ~
student@CC-AM-12:~$ java -cp hadoop/input/cloud3-0.0.1-SNAPSHOT-jar-with-dependencies.jar com.pitt.cloudcomputing.LogQuestion3
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
The website /assets/css/combined.css was hit 117348 times.
Total running time in seconds: 4.17s
student@CC-AM-12:~$
```

The total running time of this java code is 4.17 seconds.

In the java code, we select the entire data from the url table in the miniproject3 keyspace and find the website with the maximum hits by iterating over the resultant result set.

We verified our output by running the 2 queries below in CQLSH. As Cassandra doesn't support subqueries, we had to use 2 queries instead of a single nested subquery.

Test Query:

SELECT max(count) FROM url;

SELECT * FROM url WHERE count=117348 ALLOW FILTERING;

```
student@CC-AM-12: ~
cqlsh:miniproject3> SELECT max(count) FROM url;

 system.max(count)
-------------------
            117348

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:miniproject3> SELECT * FROM url WHERE count=117348 ALLOW FILTERING;

 url                       | count
---------------------------+--------
 /assets/css/combined.css | 117348

(1 rows)
cqlsh:miniproject3>
```

So, the website **/assets/css/combined.css** was hit the most with hit count = **117348**

4. **Problem 4**

LogQuestion4.java file is the source code for Problem 3.

We used the JAVA driver of cassandra to do this question.

We can execute this program using the below command:

java -cp hadoop/input/cloud3-0.0.1-SNAPSHOT-jar-with-dependencies.jar com.pitt.cloudcomputing.LogQuestion4

```
student@CC-AM-12: ~
student@CC-AM-12:~$ java -cp hadoop/input/cloud3-0.0.1-SNAPSHOT-jar-with-dependencies.jar com.pitt.cloudcomputing.LogQuestion4
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
The IP 10.216.113.172 was hit 158614 times.
Total running time in seconds: 6.777s
student@CC-AM-12:~$
```

The total running time of this java code is 6.777 seconds.

In the java code, we select the entire data from the ip table in the miniproject3 keyspace and find the IP with the maximum hits by iterating over the resultant result set.

We verified our output by running the 2 queries below in CQLSH. As Cassandra doesn't support subqueries, we had to use 2 queries instead of a single nested subquery.

Test Query:

SELECT max(count) FROM ip;

SELECT * FROM ip WHERE count=158614 ALLOW FILTERING;

```
student@CC-AM-12: ~
cqlsh:miniproject3> SELECT max(count) FROM ip;

 system.max(count)
-------------------
            158614

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:miniproject3> SELECT * FROM ip WHERE count=158614 ALLOW FILTERING;

 ip             | count
----------------+--------
 10.216.113.172 | 158614

(1 rows)
cqlsh:miniproject3>
```

So, **10.216.113.172** is the most accessed IP, with number of hits = **158614**