Cloud Computing MiniProject 3

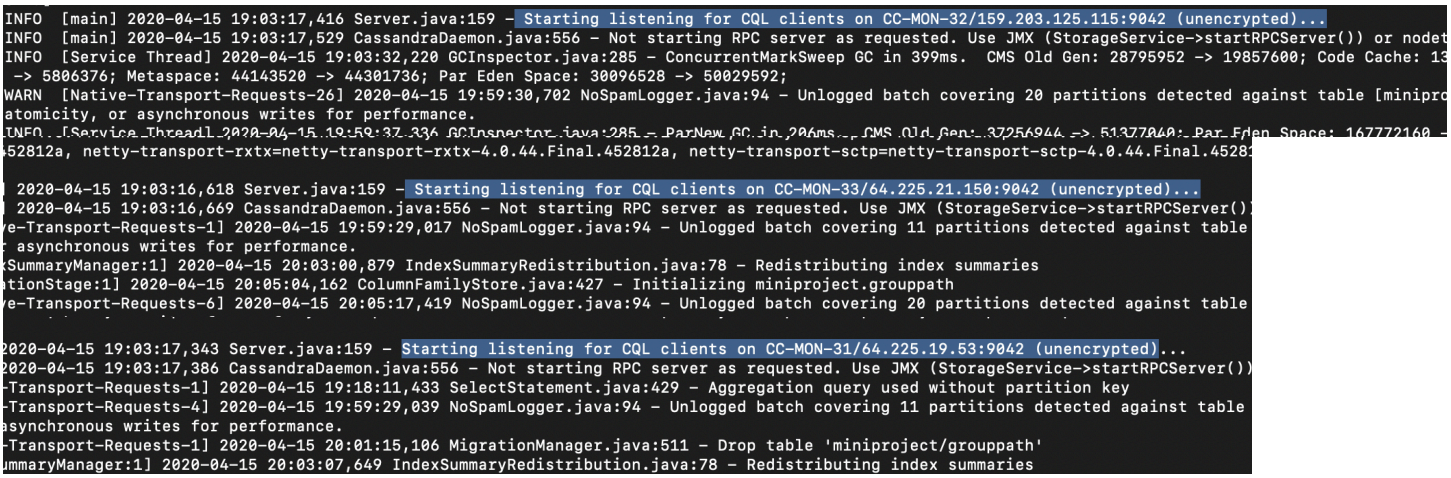
Ruoxi Zhang ruz39@pitt.edu
Kenny Wu kew143@pitt.edu
Zhanghaoxiang Yin zhy77@pitt.edu

Part 1. Setting up Cassandra

Make sure to shut down all the Hadoop services. Then on every node, stop Cassandra service first.
Use command `cassandra -Rf` to start Cassandra process on all the nodes.

```
student@CC-MON-33:~$ service cassandra stop
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to stop 'cassandra.service'.
Authenticating as: ,,, (student)
Password:
==== AUTHENTICATION COMPLETE ===
student@CC-MON-33:~$ cassandra -Rf
```

This line should appears on all the terminals.



Use `nodetool status` to check the Cassandra cluster status:

```
student@CC-MON-31:~$ nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load       Tokens       Owns (effective)  Host ID                               Rack
UN  64.225.19.53     234.4 MiB  256          100.0%            58d5b4a5-0525-4197-8815-ab654e1afffa  rack1
UN  64.225.21.150    234.83 MiB  256          100.0%            93f7fadf-3399-4ce6-8887-9250ac8e40e1  rack1
UN  159.203.125.115  113.84 MiB  256          100.0%            76774991-be23-4a96-96e0-e82961cf7aeb  rack1
```

Part 2. Import Data into Cassandra

First modify the origin access_log data:

```
student@CC-MON-31:~/hadoop$ hdfs dfs -cat access_log.csv/*
2020-04-15 21:26:34,643 INFO sasl.SaslDataTransferClient: SASL enc
10.223.157.186,/,0
10.223.157.186,/favicon.ico,1
10.223.157.186,/,2
10.223.157.186,/assets/js/lowpro.js,3
10.223.157.186,/assets/css/reset.css,4
10.223.157.186,/assets/css/960.css,5
10.223.157.186,/assets/css/the-associates.css,6
10.223.157.186,/assets/js/the-associates.js,7
10.223.157.186,/assets/js/lightbox.js,8
10.223.157.186,/assets/img/search-button.gif,9
10.223.157.186,/assets/img/dummy/secondary-news-3.jpg,10
10.223.157.186,/assets/img/dummy/primary-news-1.jpg,11
10.223.157.186,/assets/img/dummy/primary-news-2.jpg,12
10.223.157.186,/assets/img/closelabel.gif,13
10.223.157.186,/assets/img/home-logo.png,14
10.223.157.186,/assets/img/dummy/secondary-news-2.jpg,15
10.223.157.186,/assets/img/loading.gif,16
10.223.157.186,/assets/img/dummy/secondary-news-4.jpg,17
10.223.157.186,/assets/img/home-media-block-placeholder.jpg,18
10.223.157.186,/assets/img/dummy/secondary-news-1.jpg,19
10.223.157.186,/assets/swf/home-media-block.swf,20
10.223.157.186,/,21
10.223.157.186,/assets/css/960.css,22
10.223.157.186,/assets/css/the-associates.css,23
10.223.157.186,/assets/js/lowpro.js,24
10.223.157.186,/assets/js/lightbox.js,25
10.223.157.186,/assets/css/reset.css,26
10.223.157.186,/assets/js/the-associates.js,27
10.223.157.186,/assets/img/dummy/secondary-news-4.jpg,28
10.223.157.186,/assets/img/search-button.gif,29
10.223.157.186,/assets/img/dummy/primary-news-1.jpg,30
10.223.157.186,/assets/img/dummy/secondary-news-3.jpg,31
10.223.157.186,/assets/img/home-media-block-placeholder.jpg,32
10.223.157.186,/assets/img/dummy/primary-news-2.jpg,33
10.223.157.186,/assets/img/dummy/secondary-news-1.jpg,34
10.223.157.186,/assets/img/home-logo.png,35
10.223.157.186,/assets/img/dummy/secondary-news-2.jpg,36
10.223.157.186,/assets/img/loading.gif,37
```

Enter `CQL` bash using the following commands.

```
student@CC-MON-31:~$ cqlsh --request-timeout=36000000 CC-MON-31
Connected to Test Cluster at CC-MON-31:9042.
[cqlsh 5.0.1 | Cassandra 3.11.6 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
```

Create a new KEYSPACE:

```
CREATE KEYSPACE miniproject WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '3'}
```

Enter the KEYSPACE:

```
cqlsh> use miniproject;
cqlsh:miniproject>
```

Create a new table for the access_log:

```
cqlsh:miniproject> DESC table access_log;

CREATE TABLE miniproject.access_log (
    id text PRIMARY KEY,
    address text,
    path text
) WITH bloom_filter_fp_chance = 0.01
```

Use `COPY` to import access_log.csv data into Cassandra table.

```
cqlsh:miniproject> COPY access_log(address, path, id) FROM '/home/student/access_log.csv/' WITH header=FALSE
```

View the access_log table:

```
cqlsh:miniproject> select * from access_log;
```

id	address	path
8590845115	10.206.141.193	/images/filmmediablock/290/Harpoon_2d.JPG
8590373608	10.30.63.244	/images/filmpics/0000/0515/sleeve_3209_thumb.jpg
8590380182	10.111.7.41	/trailers/
8590820850	10.168.227.179	/images/frontpagepics/0000/0064/Darknessfront.jpg
17181044072	10.171.9.102	/images/newspics/0000/0377/Cannibalweb_thumb.jpg
8590803790	10.24.150.4	/images/filmmediablock/520/The_King_Maker_024.jpg
25770431565	10.210.31.172	/images/filmpics/0000/4903/DS_622E_sin%C2%A6%C2%A6_%C2%A6%C2%A6%C2%A6%C2%A6_%C2%A6%C3%BC_sin_%C2%BB-%C2%A6%C2%A6%C2%A6_%C2%A6__copy.jpg
544844	10.41.31.75	/images/filmmediablock/286/Sky_Crawlers_04.jpg
224505	10.216.113.172	/images/newspics/0000/0053/Homicidesleeve_thumb.jpg
17180483088	10.177.216.164	/images/filmpics/0000/5543/FCD477_Funhouse_Blu-Ray_s1_thumb.jpg
25770157391	10.179.11.74	/images/filmpics/0000/4941/pic2_thumb.jpg
8590467645	10.118.113.88	/images/filmpics/0000/3139/SBX476_Vanquisher_2d.jpg
8590156186	10.249.163.234	/images/filmmediablock/367/BiS6.jpg
25769899443	10.78.188.118	/images/newspics/0000/0379/TZHopperweb_thumb.jpg
8590787801	10.87.196.65	/images/filmmediablock/290/Harpoon_2d.JPG
8590382888	10.112.5.4	/displaytitle.php?id=296
8589935053	10.125.57.250	/trailers/index.php?o=d&r=d&l=2&go=Go
1084545	10.216.113.172	/images/filmmediablock/417/Inferno_BR_exploded.jpg
8590366174	10.174.165.242	/images/filmmediablock/283/Sooty2.jpg
8590832292	10.173.141.213	/assets/img/x.gif
25770612407	10.234.202.198	/images/filmpics/0000/4389/Blood_Simple_DVD_2D_thumb.jpg
123791	10.203.215.12	/assets/js/javascript_combined.js
17180005268	10.219.127.242	/images/filmpics/0000/1447/Homicide_3_thumb.JPG
8590774670	10.152.195.138	/images/filmpics/0000/4279/Monsters_Kaulder__Scoot_McNairy__6_thumb.jpg
17180276756	10.39.94.109	/images/filmpics/0000/4049/The_Pack_stills_024_thumb.jpg
911586	10.216.113.172	/database/images/sprite-aristo.png
17180315146	10.192.248.86	/

Part 3. Operate Data in Cassandra

Problem: 1. How many hits were made to the website item “/assets/img/release-schedule-logo.png”?

```
cqlsh:miniproject> select count(*) from access_log where path='/assets/img/release-schedule-logo.png' ALLOW FILTERING;

 count
-------
 24292

(1 rows)

Warnings :
Aggregation query used without partition key
```

24292 hits were made to that website item.

Problem: 2. How many hits were made from the IP: 10.207.188.188?

```
cqlsh:miniproject> select count(*) from access_log where address='10.207.188.188' ALLOW FILTERING;

 count
-------
   398

(1 rows)

Warnings :
Aggregation query used without partition key
```

398 hits were made from that IP.

Problem: 3. Which path in the website has been hit most? How many hits were made to the path?

Because Cassandra doesn't provide the group by function, we have to use User Defined Function.
The Group By Function

```
cqlsh:miniproject> CREATE OR REPLACE FUNCTION groupBy(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 = count + 1;
                ... state.put(type, count);
                ... return state;';
cqlsh:miniproject> CREATE OR REPLACE AGGREGATE groupBy_count(text)
                ... SFUNC groupBy
                ... STYPE map<text, int>
                ... INITCOND {};
cqlsh:miniproject> CAPTURE
Currently not capturing query output.
cqlsh:miniproject> CAPTURE '/home/student/groupBy_path.csv';
Now capturing query output to '/home/student/groupBy_path.csv'.
cqlsh:miniproject> select groupBy_count(path) from access_log;
```

Use `CAPTURE` command to export the query output to a new file 'groupBy_path.csv'. 'groupBy_path.csv' stores the data grouped by path.

We have to do some transformation to convert 'groupBy_path.csv' into Cassandra-readable dataset:

```
student@CC-MON-31:~$ sed -i 's/,/\n/g' groupBy_address.csv
student@CC-MON-31:~$ sed -i 's/:/,/g' groupBy_address.csv
student@CC-MON-31:~$ sed -i 's/ //g' groupBy_address.csv
student@CC-MON-31:~$ nano groupBy_address.csv
```

These commands:
1. Replace commas with new lines.
2. Replace colons with commas.
3. Remove white-spaces.

Then we can create a new table 'groupPath' in Cassandra and copy the grouped data to this table.

```
Connected to Test Cluster at CC-MON-31:9042.
[cqlsh 5.0.1 | Cassandra 3.11.6 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use miniproject;
cqlsh:miniproject> create table groupPath(
               ... path text PRIMARY KEY,
               ... count int);
cqlsh:miniproject> COPY groupPath(path, count) FROM '/home/student/groupBy_path.csv' WITH header=FALSE;
Using 1 child processes

Starting copy of miniproject.grouppath with columns [path, count].
```

We have to define another UDF.

The Max Aggregation Function:

```
CREATE FUNCTION miniproject.maxcount(current int, next int)
    CALLED ON NULL INPUT
    RETURNS int
    LANGUAGE java
    AS $$ if (current==null) return next; else return Math.max(current, next);$$;

CREATE AGGREGATE miniproject.maxagg(int)
    SFUNC maxcount
    STYPE int;
```

Use the new defined function to select the max count in path-grouped data:

```
cqlsh:miniproject> SELECT maxAgg(count) FROM groupPath;

 miniproject.maxagg(count)
---------------------------
                    117348

(1 rows)

Warnings :
Aggregation query used without partition key
```

Using the max count, we can get the answer:

```
cqlsh:miniproject> SELECT path FROM groupPath where count = 117348 ALLOW FILTERING;

 path
--------------------------
 '/assets/css/combined.css'

(1 rows)
cqlsh:miniproject>
```

Path ‘/assets/css/combined.css’ has been hit most.
117348 hits was made to this path.

Problem: 4. Which IP accesses the website most? How many accesses were made by it?

Like Problem 3, we first use Group_By UDF to collect the dataset grouped by IP address.

```
cqlsh:miniproject> CAPTURE '/home/student/groupBy_address.csv'
Now capturing query output to '/home/student/groupBy_address.csv'.
cqlsh:miniproject> select groupBy_count(address) from access_log;
cqlsh:miniproject>
```

Then we modified the new csv file and copy it into a new table ‘groupAddress’.

```
Connected to Test Cluster at CC-MON-31:9042.
[cqlsh 5.0.1 | Cassandra 3.11.6 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use miniproject;
cqlsh:miniproject> create table groupAddress(
              ... address text PRIMARY KEY,
              ... count int);
cqlsh:miniproject> COPY groupAddress(address, count) FROM '/home/student/groupBy_address.csv' WITH header=FALSE;
Using 1 child processes

Starting copy of miniproject.groupaddress with columns [address, count].
Failed to import 1 rows: ParseError - Failed to parse 21} : invalid literal for int() with base 10: '21}',  giver
Failed to import 1 rows: ParseError - Invalid row length 0 should be 2,  given up without retries
Failed to process 2 rows; failed rows written to import_miniproject_groupaddress.err
Processed: 333924 rows; Rate:    5737 rows/s; Avg. rate:   10695 rows/s
333924 rows imported from 1 files in 31.224 seconds (0 skipped).
```

Use Max Aggregation function to get the max count:

```
cqlsh:miniproject> select maxAgg(count) from groupAddress;

 miniproject.maxagg(count)
---------------------------
                    158614

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:miniproject> select address from groupAddress where count=158614 allow filtering;

 address
-----------------
 '10.216.113.172'

(1 rows)
cqlsh:miniproject>
```

IP ’10.216.113.172’ accesses the website most.
158614 accesses were made by it.