SWEN432 Assignment 2

Songbo Wu
ID :300422252

Question 1.

arthur: [A2] % ccm create -n 5 single_dc;
ccm node1 ring ;

Datacenter: datacenter1
==========

Address	Rack	Status	State	Load	Owns	Token
						5534023222112865484
127.0.0.1	rack1	Up	Normal	78.39 KiB	40.00%	-9223372036854775808
127.0.0.2	rack1	Up	Normal	98.98 KiB	40.00%	-5534023222112865485
127.0.0.3	rack1	Up	Normal	104.13 KiB	40.00%	-1844674407370955162
127.0.0.4	rack1	Up	Normal	98.98 KiB	40.00%	1844674407370955161
127.0.0.5	rack1	Up	Normal	98.97 KiB	40.00%	5534023222112865484

Question 2.

a) endpoint_snitch: SimpleSnitch
b)initial_token: -9223372036854775808.
partitioner calculated it.
initial_token in casssandra.yaml is in the node1 127.0.0.1 row Token column of the ccm node1 ring output table
c) partitioner: org.apache.cassandra.dht.Murmur3Partitioner
d) rpc_address: 127.0.0.1 is in the node1 row Address column of the ccm node1 ring output table

Question 3.

If there is only one data center, then the information between the file and the ccm ring output is irrelevant.
If there are mulitply data centers, the ip address in the ccm ring output is the ip address in the casssandra.topology.properties file.

Question 4.

source '/home/wusong3/SWEN432/A2/A2Q4.cql';
a)

```
DROP KEYSPACE IF EXISTS ass2 ;
CREATE KEYSPACE IF NOT EXISTS  ass2 WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 3};
USE ass2;
```

b)

```
create table driver (driver_name text, email text, password text, mobile int,
current_position text, skill set<text>, primary key (driver_name))
with compaction = {'class': 'LeveledCompactionStrategy'} and dclocal_read_repair_chance = 1;

create index driver_skill_idx on driver (values(skill));
create index driver_current_position_idx on driver (current_position);

create table vehicle (vehicle_id text, status text, type text, primary key (vehicle_id))
with compaction = {'class': 'LeveledCompactionStrategy'};

create index veh_stat_idx on vehicle (status);

create table time_table (line_name text, service_no int, time int, distance double, latitude double, longitude
double, stop text,
primary key ((line_name, service_no), time)) with clustering order by (time desc);

create table data_point (line_name text, service_no int, date int, sequence timestamp, longitude double, latitude
double,
speed double, primary key ((line_name, service_no, date), sequence)) with clustering order by (sequence
desc);



COPY driver FROM   '/home/wusong3/SWEN432/A2/driver_data.csv' ;
COPY vehicle FROM   '/home/wusong3/SWEN432/A2/vehicle_data.csv';
COPY time_table FROM   '/home/wusong3/SWEN432/A2/time_table_data.csv';
COPY data_point FROM  '/home/wusong3/SWEN432/A2/data_point_data.csv';


describe tables;
select * from driver limit 1 ;
select * from vehicle limit 1 ;
select * from time_table limit 1 ;
select * from data_point limit 1 ;

time_table  data_point  driver  vehicle
```

```
 driver_name | current_position | email              | mobile  | password | skill
-------------+------------------+--------------------+---------+----------+--------------------------
        fred |            Taita | fred@ecs.vuw.ac.nz | 2799797 |     f00f | {'Ganz Mavag', 'Guliver'}

(1 rows)

 vehicle_id | status     | type
------------+------------+---------
     KW3300 | Wellington | Matangi

(1 rows)

 line_name | service_no | time | distance | latitude | longitude | stop
-----------+------------+------+----------+----------+-----------+---------
   Melling |          3 |  807 |     13.7 | -41.2036 |  174.9054 | Melling

(1 rows)

 line_name       | service_no | date     | sequence                       | latitude | longitude | speed
-----------------+------------+----------+--------------------------------+----------+-----------+-------
 Hutt Valey Line |          2 | 20160326 | 2016-03-25 21:07:40.000000+0000 | -41.2012 |       175 |  70.1
```

Question 5:

a)

```
arthur: [A2] % ccm node1 nodetool getendpoints ass2 driver pavle

127.0.0.1
127.0.0.2
127.0.0.3
```

b)

```
arthur: [A2] % ccm node4 cqlsh
Connected to single_dc at 127.0.0.4:9042.

 driver_name | current_position | email                 | mobile | password | skill
-------------+------------------+-----------------------+--------+----------+-------------------------------------
       pavle |       Upper Hutt | pmogin@ecs.vuw.ac.nz | 213344 |     pm33 | {'Ganz Mavag', 'Guliver', 'Matangi'}
```

ccm node1 stop ;

arthur: [A2] % ccm node4 cqlsh ;
Connected to single_dc at 127.0.0.4:9042.

cqlsh> consistency all ; use ass2 ; select * from driver where driver_name ='pavle' ;
Consistency level set to ALL.
NoHostAvailable:

Read Consistency Level ALL provides the lowest availabity of all the levels. If one node is down, the data still can not be read.

c)
cqlsh:ass2> consistency quorum ; select * from driver where driver_name ='pavle' ;
Consistency level set to QUORUM.

 driver_name | current_position | email | mobile | password | skill
-------------+------------------+----------------------+--------+----------+-----------------------------------
       pavle | Upper Hutt | pmogin@ecs.vuw.ac.nz | 213344 | pm33 | {'Ganz Mavag', 'Guliver', 'Matangi'}

(1 rows)

ccm node2 stop ;

cqlsh:ass2> consistency quorum ; use ass2 ; select * from driver where driver_name ='pavle' ;
Improper consistency command.
NoHostAvailable:

QUORUM is the prerequisite for the strong consistency which requires at least two node up in this case.
Since node1 and node2 are the down. It can not meet the quorum in this case.

d)
cqlsh:ass2> consistency one ; use ass2 ; select * from driver where driver_name ='pavle' ;

Consistency level set to ONE.

```
 driver_name | current_position | email                | mobile | password | skill
-------------+------------------+----------------------+--------+----------+-------------------------------------
       pavle |       Upper Hutt | pmogin@ecs.vuw.ac.nz | 213344 |     pm33 | {'Ganz Mavag', 'Guliver', 'Matangi'}

(1 rows)
```

ccm node3 stop ;

cqlsh:ass2> consistency one ; use ass2 ; select * from driver where driver_name ='pavle' ;
Consistency level set to ONE.
NoHostAvailable:

Because driver pavle is in node1 , node2 , node3 . If all these three nodes are down, it can not meet the consistency level one which needs at least one
node up to provide the highest availability of all the levels

Question 6

procedure:

ccm start ; ccm node1 stop ; ccm status;
cqlsh:ass2> consistency all ; select * from driver where driver_name = 'eileen' ;
Consistency level set to ALL.
NoHostAvailable:
ccm start ; ccm node2 stop ; ccm status;
cqlsh:ass2> consistency all ; select * from driver where driver_name = 'eileen' ;
Consistency level set to ALL.
NoHostAvailable:
ccm start ; ccm node3 stop ; ccm status;
cqlsh:ass2> consistency all ; select * from driver where driver_name = 'eileen' ;
Consistency level set to ALL.
NoHostAvailable:
ccm start ; ccm node4 stop ; ccm status;
cqlsh:ass2> consistency all ; select * from driver where driver_name = 'eileen' ;

Consistency level set to ALL.
NoHostAvailable:
ccm start ; ccm node5 stop ; ccm status;
cqlsh:ass2> consistency all ; select * from driver where driver_name = 'eileen' ;
Consistency level set to ALL.
NoHostAvailable:

since return NoHostAvailable in all case, 'eileen' does not EXIST in the database.

Question 7

arthur: [A2] % ccm node4 stop ;

use ass2 ; insert into driver (driver_name, password) values ('james', '7007');

arthur: [A2] % ccm node5 stop ; ccm node1 stop ; ccm node4 start ;
arthur: [A2] % ccm node3 cqlsh

cqlsh> use ass2 ; select driver_name, password from driver where driver_name = 'james';

 driver_name | password
-------------+----------
       james | 7007

By default , consistency level is one , which is eventually consistency.

The hinted handoff Cassandra mechanism is used in this case.
When node4 is down, hint and data stores into coordinator node3 system.hints table .
when node4 is up , data stored in system.hints table goes back to node4.

The hinted handoff is applied because there are enough available replica nodes to satisfy the requested consistency level of one .

Question 9:

arthur: [A2] % ccm create -n 5:4 -s multi_dc ;

arthur: [A2] % ccm node1 ring ;

Datacenter: dc1
==========

Address	Rack	Status	State	Load	Owns	Token
						5534023222112865484
127.0.0.1	r1	Up	Normal	97.39 KiB	25.00%	-9223372036854775808
127.0.0.2	r1	Up	Normal	103.22 KiB	20.00%	-5534023222112865485
127.0.0.3	r1	Up	Normal	77.7 KiB	20.00%	-1844674407370955162
127.0.0.4	r1	Up	Normal	96.65 KiB	20.00%	1844674407370955161
127.0.0.5	r1	Up	Normal	103.22 KiB	20.00%	5534023222112865484

Datacenter: dc2
==========

Address	Rack	Status	State	Load	Owns	Token
						4611686018427388004
127.0.0.6	r1	Up	Normal	83.2 KiB	20.00%	-9223372036854775708
127.0.0.7	r1	Up	Normal	65.2 KiB	25.00%	-4611686018427387804
127.0.0.8	r1	Up	Normal	78.35 KiB	25.00%	100
127.0.0.9	r1	Up	Normal	83.85 KiB	25.00%	4611686018427388004

Question 10:

arthur: [A2] % ccm create -n 5:4 -s multi_dc ;
endpoint_snitch: org.apache.cassandra.locator.PropertyFileSnitch

arthur: [A2] % ccm create -n 5 single_dc;
endpoint_snitch: SimpleSnitch

it is different because the SimpleSnitch one is using cluster with one Datacenter while the other one is using multipul Datacenter.

Question 11:

/home/wusong3/cassandra_training/multi_dc/node1/conf/cassandra-topology.properties :
default=dc1:r1
127.0.0.1=dc1:r1
127.0.0.2=dc1:r1
127.0.0.3=dc1:r1
127.0.0.4=dc1:r1
127.0.0.5=dc1:r1
127.0.0.6=dc2:r1
127.0.0.7=dc2:r1
127.0.0.8=dc2:r1
127.0.0.9=dc2:r1

The ip address in ccm node1 ring output are the same as the ip address showed in the cassandra-topology.properties. Both showing the fact that
127.0.0.1 ~ 127.0.5 belong to datacenter1 rack1.
127.0.0.6 ~ 127.0.9 belong to datacenter2 rack1.

But the ccm node1 ring output shows more information like Status ,State , Load , Owns , Token about each ip address.

Question 12:

DROP KEYSPACE IF EXISTS ass2 ;
CREATE KEYSPACE IF NOT EXISTS ass2 WITH replication = {'class':'NetworkTopologyStrategy', 'dc1':3, 'dc2':3}};

Question 13:

USE ass2;
create table driver (driver_name text, email text, password text, mobile int,
current_position text, skill set<text>, primary key (driver_name))
with compaction = {'class': 'LeveledCompactionStrategy'} and dclocal_read_repair_chance = 1;

create index driver_skill_idx on driver (values(skill));

```
create index driver_current_position_idx on driver (current_position);

create table time_table (line_name text, service_no int, time int, distance double, latitude double, longitude
double, stop text,
primary key ((line_name, service_no), time)) with clustering order by (time desc);

COPY driver FROM   '/home/wusong3/SWEN432/A2/driver_data.csv' ;
COPY time_table FROM   '/home/wusong3/SWEN432/A2/time_table_data.csv';


describe tables;
select * from driver limit 1 ;
select * from time_table limit 1 ;


cqlsh>  source '/home/wusong3/SWEN432/A2/A2Q4.cql';
Using 7 child processes

Starting copy of ass2.driver with columns [driver_name, current_position, email, mobile, password, skill].
Processed: 6 rows; Rate:       3 rows/s; Avg. rate:       5 rows/s
6 rows imported from 1 files in 1.185 seconds (0 skipped).
Using 7 child processes

Starting copy of ass2.time_table with columns [line_name, service_no, time, distance, latitude, longitude, stop].
Processed: 30 rows; Rate:      50 rows/s; Avg. rate:      75 rows/s
30 rows imported from 1 files in 0.400 seconds (0 skipped).

time_table  driver


 driver_name | current_position | email              | mobile  | password | skill
-------------+------------------+--------------------+---------+----------+--------------------------
        fred |            Taita | fred@ecs.vuw.ac.nz | 2799797 |     f00f | {'Ganz Mavag', 'Guliver'}

(1 rows)

 line_name | service_no | time | distance | latitude | longitude | stop
-----------+------------+------+----------+----------+-----------+---------
   Melling |          3 |  807 |     13.7 | -41.2036 |  174.9054 | Melling
```

(1 rows)

Question 14 :

arthur: [A2] % ccm node1 nodetool getendpoints ass2 driver pavle

127.0.0.1
127.0.0.6
127.0.0.2
127.0.0.7
127.0.0.3
127.0.0.8

i. ccm node1 nodetool getendpoints ass2 driver pavle

pranah: [~/SWEN432] % ccm node1 cqlsh ;
Connected to multi_dc 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> use ass2 ; consistency quorum;
Consistency level set to QUORUM.
cqlsh:ass2> select driver_name, password from driver where driver_name = 'pavle';

 driver_name | password
-------------+----------
       pavle | pm33

(1 rows)

cqlsh:ass2> use ass2 ; consistency each_quorum;
Consistency level set to EACH_QUORUM.
cqlsh:ass2> select driver_name, password from driver where driver_name = 'pavle';

 driver_name | password
-------------+----------
       pavle | pm33

(1 rows)

pranah: [~/SWEN432] % ccm node1 cqlsh ; -- dc1 being local.
cqlsh:ass2> use ass2 ; consistency local_quorum;
Consistency level set to LOCAL_QUORUM.
cqlsh:ass2> select driver_name, password from driver where driver_name = 'pavle';

 driver_name | password
-------------+----------
       pavle | pm33

(1 rows)

pranah: [~/SWEN432] % ccm node9 cqlsh ;   -- dc2 being local.
Connected to multi_dc at 127.0.0.9:9042.
[cqlsh 5.0.1 | Cassandra 3.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use ass2 ; consistency local_quorum;
Consistency level set to LOCAL_QUORUM.
cqlsh:ass2> select driver_name, password from driver where driver_name = 'pavle';

 driver_name | password
-------------+----------
       pavle | pm33

(1 rows)

ii,

arthur: [A2] % ccm node7 stop ; ccm node8 stop ;

pranah: [~/SWEN432] % ccm node1 cqlsh ;
Connected to multi_dc 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>  use ass2 ; consistency quorum ; select driver_name, password from driver where driver_name = 'pavle';
Consistency level set to QUORUM.

 driver_name | password
-------------+----------
       pavle |     pm33

(1 rows)


cqlsh:ass2>  use ass2 ; consistency each_quorum ; select driver_name, password from driver where
driver_name = 'pavle';
Consistency level set to EACH_QUORUM.
NoHostAvailable:




dc1 being local:
pranah: [~/SWEN432] % ccm node1 cqlsh  --  dc1 being local.
cqlsh:ass2>  use ass2 ; consistency local_quorum ; select driver_name, password from driver where
driver_name = 'pavle';
Consistency level set to LOCAL_QUORUM.

 driver_name | password
-------------+----------
       pavle |     pm33

(1 rows)


dc2 being local:
pranah: [~/SWEN432] % ccm node9 cqlsh ;     --  dc2 being local.
Connected to multi_dc at 127.0.0.9:9042.
[cqlsh 5.0.1 | Cassandra 3.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use ass2 ; consistency local_quorum ; select driver_name, password from driver where driver_name =
'pavle';
Consistency level set to LOCAL_QUORUM.
NoHostAvailable:
```

Under the consistency level EACH_QOURUMat, at least 2 nodes from each data centre has to respond. Since two nodes are down, so consistency each_qourum fails .

Question 15:

cqlsh:ass2> select token(line_name, service_no) from time_table where line_name = 'Hutt Valley Line' and service_no =2;

```
 system.token(line_name, service_no)
-------------------------------------
              2322329569350831795
              2322329569350831795
              2322329569350831795
              2322329569350831795
              2322329569350831795

(5 rows)


Datacenter: dc1
==========
Address    Rack       Status State   Load            Owns                Token
                                                                        5534023222112865484
127.0.0.1  r1         Up     Normal  97.39 KiB       25.00%              -9223372036854775808
127.0.0.2  r1         Up     Normal  103.22 KiB      20.00%              -5534023222112865485
127.0.0.3  r1         Up     Normal  77.7 KiB        20.00%              -1844674407370955162
127.0.0.4  r1         Up     Normal  96.65 KiB       20.00%              1844674407370955161
127.0.0.5  r1         Up     Normal  103.22 KiB      20.00%              5534023222112865484

Datacenter: dc2
==========
Address    Rack       Status State   Load            Owns                Token
                                                                        4611686018427388004
127.0.0.6  r1         Up     Normal  83.2 KiB        20.00%              -9223372036854775708
127.0.0.7  r1         Up     Normal  65.2 KiB        25.00%              -4611686018427387804
127.0.0.8  r1         Up     Normal  78.35 KiB       25.00%              100
127.0.0.9  r1         Up     Normal  83.85 KiB       25.00%              4611686018427388004
```

The token for target row is 2322329569350831795, according to this ccm ring, it belongs to 127.0.0.5 , 127.0.0.1 , 127.0.0.2 and 127.0.0.9, 127.0.0.6 ,127.0.0.7