School of Engineering and Computer Science

SWEN 432
Advanced Database Design and Implementation

Assignment 2

Model Solutions

Due date: Monday 01 May at 23:59

The objective of this assignment is to test your understanding of Cassandra Cloud Database Management System and your ability to apply this knowledge. The Assignment is worth 5.0% of your final grade. The Assignment is marked out of 100.

You will need to use Cassandra to answer a number of assignment questions. Cassandra has been already installed on our school system. There is an Instruction for using Cassandra on our lab workstations given at the end of the Assignment.

Overview

In lectures, we discussed Cassandra architecture, consistency levels, and repair mechanisms in detail. In this assignment, you are going to perform a number of experiments on these Cassandra features using `ccm` and `nodetool`.

`single_dc` Cluster [66 marks]

Question 1. [2 marks] Use `ccm` to make a single data center Cassandra cluster having 5 nodes. Call it `single_dc`. Start the cluster and run the `ccm ring` command. Save the output of the ring command for future use and show it in the answer to the question.

ANSWER

```
[~] % ccm create -n 5 single_dc
Current cluster is now: single_dc
dunsheas-deli: [~] % ccm start

dunsheas-deli: [~] % ccm node1 ring

Datacenter: datacenter1
==========
Address   Rack Status State   Load            Token
                                        5534023222112865484
127.0.0.1 rack1 Up   Normal  106.18 KB -9223372036854775808
127.0.0.2 rack1 Up   Normal  104.86 KB -5534023222112865485
127.0.0.3 rack1 Up   Normal  104.86 KB -1844674407370955162
127.0.0.4 rack1 Up   Normal   86.15 KB  1844674407370955161
127.0.0.5 rack1 Up   Normal   68.76 KB  5534023222112865484
```

Question 2. [14 marks] Consider the `casssandra.yaml` file of node1.

a) [2 marks] What is the setting of the `endpoint_snitch` property?

ANSWER

```
SimpleSnitch
```

b) [6 marks] What is the value of the `initial_token` property, which Cassandra component has calculated it, and is there any relationship between `initial_token` property value and the output of the `ccm node1 ring` command?

ANSWER

```
Initial_token: -9223372036854775808
Calculated by the Cassandra default partitioner
Initial token is the same as Token of the node with IP
Address 127.0.0.1 in the output of the ccm ring command.
Represents the largest token value to store on node
127.0.0.1
```

c) [2 marks] What is the setting of the `partitioner` property?

ANSWER

```
org.apache.cassandra.dht.Murmur3Partitioner
```

d) [4 marks] What is the setting of the `rpc_address` property and is there any relationship between `rpc_address` property value and the output of the `ccm node1 ring` command?

ANSWER

```
127.0.0.1
The same as the IP Address of the node1 in the output of
the ccm ring command
```

Question 3. [2 marks] Consider the `casssandra.topology.properties` file of node1 and comment on the relationship between file's content and the output of the `ccm node1 ring` command.

ANSWER

```
There is no relationship.
The file casssandra.topology.properties represents the file
of the PropertyFileSnitch. Since the current cluster has
SimpleSnitch, the file casssandra.topology.properties
contains default values that have no relationship with
SimpleSnitch.
```

Question 4. [8 marks]

a) [3 marks] Connect to `cqlsh` prompt and create a keyspace with the name `ass2`. Replication strategy should be simple, and the replication factor equal 3. In your answer, show your keyspace declaration.

ANSWER

```
[~] % ccm node1 cqlsh
Connected to single_dc at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.1.1 | CQL spec 3.3.1 | Native
protocol v4]
Use HELP for help.
cqlsh> create keyspace ass2 with replication = {'class':
'SimpleStrategy', 'replication_factor': 3};
```

b) [5 marks] The following files:

```
table_declarations.cql
data_point_data.txt
driver_data_txt
time_table_data.txt
```

`vehicle_data.txt`

are given on the course Assignments page. The file `table_declarations.cql` contains create table statements, while the other files contain comma separated table data. Use these files, and `SOURCE` and `COPY cqlsh` commands to implement a version of the train time table data base. In your answer show the results of running the `cqlsh` command `describe tables` and of running `select` statements on each table for a row of your choice.

ANSWER

```
cqlsh:ass2> source 'table_declarations.cql';
cqlsh:ass2> describe tables;
data_point driver vehicle time_table

copy driver (driver_name, email, password, mobile,
current_position, skill) from 'driver_data.txt';
Starting copy of ass2.driver with columns ['driver_name',
'email', 'password', 'mobile', 'current_position',
'skill'].
6 rows imported in 0.229 seconds.

cqlsh:ass2> select * from driver where driver_name =
‘fred’;
 driver_name | current_position | email                |
mobile   | password | skill
-------------+------------------+----------------------+---
-------+----------+-------------------------------------
        fred |            Taita |   fred@ecs.vuw.ac.nz |
2799797 |     f00f |            {'Ganz Mavag', 'Guliver'}
(1 rows)

cqlsh:ass2> copy vehicle (vehicle_id, status, type) from
'vehicle_data.cql';
Starting copy of ass2.vehicle with columns ['vehicle_id',
'status', 'type'].
6 rows imported in 0.180 seconds.

cqlsh:ass2> select * from vehicle where vehicle_id =
‘KW3300’;
 vehicle_id | status       | type
------------+--------------+------------
     KW3300 |   Wellington |    Matangi
 (1 rows)
```

```
cqlsh:ass2> copy time_table (line_name, service_no, stop,
time, latitude, longitude, distance) from
'time_table_data.txt';
Starting copy of ass2.time_table with columns ['line_name',
'service_no', 'stop', 'time', 'latitude', 'longitude',
'distance'].
30 rows imported in 0.247 seconds.

cqlsh:ass2> select * from time_table limit 1;
 line_name        | service_no | time | distance | latitude
| longitude | stop
------------------+------------+------+----------+--------
+-----------+-------------
          Melling |          3 |  807 |     13.7 | -41.2036
|  174.9054 |      Melling

cqlsh:ass2> copy data_point (line_name, service_no, date,
sequence, longitude, latitude, speed) from
'data_point_data.txt';
Starting copy of ass2.data_point with columns ['line_name',
'service_no', 'date', 'sequence', 'longitude', 'latitude',
'speed'].
5 rows imported in 0.213 seconds.

cqlsh:ass2> select * from data_point limit 1;
 line_name       | service_no | date     | sequence
         | latitude | longitude | speed
-----------------+------------+----------+-----------------
---------+----------+-----------+-------
 Hutt Valey Line |          2 | 20160326 | 2016-03-25
21:07:40+0000 | -41.2012 |       175 |  70.1
```

Question 5. [10 marks] To answer this question, you will need to use the `getendpoints nodetool` command.

a) [1 mark] Find the nodes storing data of driver pavle. In your answer, show the output of the `getendpoints nodetool` command. Let us call these nodes node_a, node_b, and node_c.

ANSWER

```
dunsheas-deli: [~] % ccm node1 nodetool getendpoints -- ass2
driver pavle
127.0.0.1
127.0.0.2
127.0.0.3
```

b) [3 marks] Connect to `cqlsh` prompt using a node that is not in the set {node_a, node_b, node_c}. Set the consistency level to `ALL` and read data of the driver pavle. Stop node_a, connect to `cqlsh`, set the consistency level to `ALL` and read pavle's data again. What have you learned?

ANSWER

Consistency ALL requires all replica nodes to respond. If any replica node is not available, Cassandra will throw an "Unavailable exception" message.

c) [3 marks] With node_a still being stopped, set the consistency level to `QOURUM` and read pavle's data. Stop node_b, connect to `cqlsh`, set the consistency level to `QUORUM` and read pavle's data again. What have you learned.

ANSWER

Consistency QUORUM requires majority of replica nodes to respond. If less than majority of nodes are available, Cassandra will throw an "Unavailable exception" message.

d) [3 marks] With node_a and node_b still being stopped, set the consistency level to `ONE` and read pavle's data. Stop node_c, connect to `cqlsh`, and read pavle's data again. What have you learned.

ANSWER

Consistency ONE requires at least one replica node to respond. If none nodes are available, Cassandra will throw an "Unavailable exception" message.

Question 6. [15 marks] You are asked to find those nodes of the `single_dc` Cassandra cluster that store replicas of driver eileen. Very soon you realized that all `ccm` commands and `nodetool` commands, including `ccm start, ccm stop, ccm status, ccm nodei cqlsh` and so on, work properly except the command

```
ccm nodei nodetool getendpoints ass2 driver eileen
```

Despite that, you have devised a procedure to find the nodes requested. In your answer, describe the procedure and show how you have applied it.

ANSWER

There are two procedures presented, one for even, and the other for odd number of nodes in a cluster.

Assumptions:

- The cluster uses nodes from a single data center,
- The number of cluster nodes *m*,
- The replication factor is 3.

```
Procedure (m (≥ 4) and even):
  i = - 1
  do while select statement returns a row {
     i = i + 2
     ccm node(i) stop
     ccm node(i + 1) stop
     ccm node((i + 2)mod m) cqlsh
     consistency quorum;
     select * from driver where driver_name = “eileen”;
     ccm start
  }
  ccm node(i + 1) stop
  ccm node((i + 2)mod m) stop
  ccm node(i) cqlsh
  consistency quorum;
  select * from driver where driver_name = “eileen”;

  if select statement returns a row
  then
     nodes node(x), node(i), and node(i + 1) store
     driver Eileen, where x = i – 1 for i > 1, and x = m
     for x = 1
  else
     nodes node(i), node(i + 1), and node((i + 2)mod m)
     store driver Eileen


Procedure (m (≥ 3) and odd):
  i = - 1
  do while select statement returns a row {
     i = i + 2
     ccm node(i) stop
     ccm node((i + 1)mod m) stop
     ccm node((i + 2)mod m) cqlsh
     consistency quorum;
     select * from driver where driver_name = “eileen”;
     ccm start
  }
```

```
ccm node((i + 1)mod m) stop
ccm node((i + 2)mod m) stop
ccm node(i) cqlsh
consistency quorum;
select * from driver where driver_name = “eileen”;

if select statement returns a row
then
   nodes node(x), node(i), and node((i + 1)mod m) store
   driver Eileen, where x = i – 1 for i > 1, and x = m
   for x = 1
else
   nodes node(i), node((i + 1)mod m), and
   node((i + 2)mod m) store driver Eileen
```

Observe:

- The maximal number of iterations through the while loop is celling(*m* / 2)
- When the select statement throws an exception of the type Unavailable, it means:
  The two stopped nodes store the row required, and
  The third node is either the counter wise neighbor of the node(i) or the clock wise neighbor of the node(i + 1) for *m* even or the clock wise neighbor of the node((i + 1)mod *m*) for *m* odd.

Question 7. [15 marks] Assume the following situation:

1. The data of the driver james should be stored on node4, node5, and node1.
2. A client (say c0) connected to node3 and sent a request to write james’s data.
3. In the moment of running the statement

   ```
   insert into driver (driver_name, password) values
   (‘james’, ’7007’);
   ```

   node4 was down.
4. Writing succeeded.
5. In the next moment node5 and node1 went down and the node4 started.
6. A client (say c1) connected to `cqlsh` prompt via node3 and sent the following read statement:
7. `select driver_name, password from driver where driver_name = 'james';`
8. The read result was:

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

Repeat the experiment described above. Name and briefly explain Cassandra mechanism that made succeeding of the `select` statement above possible.

ANSWER

- The mechanism is hinted handoff
- The hinted handoff writes a hint and data to the coordinator node if some of *n* replicas are down or not replying
- The hinted handoff is applied only in the case when there are enough available replica nodes to satisfy the requested consistency level
- During a write operation, the coordinator node stores a hint about unavailable replica nodes in a local file and the actual data being written
- A hint indicates that a write needs to be replayed to one or more unavailable nodes
- After a coordinator node discovers from gossip that a node for which it holds hints has recovered, the node sends each hinted data to the recovered node

`multi_dc` Cluster [34 marks]

Question 9. [3 marks] Use `ccm` to make a Cassandra cluster spanning two datacenters. The cluster name shoud be `multi_dc`. Cassandra will automatically assign default names `dc1` and `dc2` to datacenters. The cluster `multi_dc` uses 5 nodes in `dc1` and 4 nodes in `dc2`. Start the cluster and run the `ccm ring` command. Save the output of the `ring` command for future use and show it in the answer to the question.

ANSWER

```
[~] % ccm create -n 5:4 multi_dc
Current cluster is now: multi_dc

[~] % ccm node1 ring

Datacenter: dc1
==========
Address  Rack Status  State   Load             Token
                                         534023222112865484
127.0.0.1 r1    Up   Normal  81.04 KB  -9223372036854775808
127.0.0.2 r1    Up   Normal 104.85 KB  -5534023222112865485
127.0.0.3 r1    Up   Normal  81.58 KB  -1844674407370955162
127.0.0.4 r1    Up   Normal  80.69 KB   1844674407370955161
127.0.0.5 r1    Up   Normal  68.75 KB   5534023222112865484

Datacenter: dc2
==========
Address  Rack Status  State   Load              Token
                                        4611686018427388004
127.0.0.6 r1    Up   Normal  81.04 KB  -9223372036854775708
127.0.0.7 r1    Up   Normal  81.04 KB  -4611686018427387804
127.0.0.8 r1    Up   Normal  73.87 KB                 100
```

```
127.0.0.9 r1     Up   Normal  80.71 KB   4611686018427388004
```

Question 10. [4 marks] Consider the `casssandra.yaml` file of node1. What is the setting of the `endpoint_snitch` property? If you find it different to the setting in the case of the `single_dc` cluster, explain briefly why it is different.

ANSWER

The `endpoint_snitch` is set to `PropertyFileSnitch`. The `PropertyFileSnitch` is used in clusters spanning more than one data center or more racks, while the SimpleSnitch is used for clusters deployed in a single data center and single rack.

Since the cluster created in Q9 spans two data centers, Cassandra has automatically assigned the `PropertyFileSnitch` to the new cluster.

Question 11. [4 marks] Consider the `casssandra.topology.properties` file of node1 and comment on the relationship between file's content and the output of the `ccm node1 ring` command.

ANSWER

The `cassandra-topology.properties` file is used by the Network Topology Strategy as a snitch file. It contains complete data center and rack information of the cluster. In the case of a cluster using more than one data center or more than one rack, information in `cassandra-topology.properties` matches up with the output of the ring command.

Question 12. [2 marks] Create a `keyspace` with the name `ass2` having network topology replication strategy and a replication factor of 3 for both `dc1` and `dc2` datacenters. In your answer, show your keyspace declaration.

ANSWER

```
cqlsh> create keyspace ass2 with replication = {'class':
'NetworkTopologyStrategy', 'dc1': 3, 'dc2':3};
```

Question 13. [3 marks] Use `SOURCE` and `COPY cqlsh` commands and the following files:

`table_declarations.cql`

`driver_data_txt`

`time_table_data.txt`

to implement a version of the train time table data base. You need to populate only `driver` and `time_table` tables by data. In your answer show the results of running the `cqlsh` command `describe tables` and of running CQL `select` statements on `driver` and `time_table` for a row of your choice.

ANSWER

```
cqlsh> use ass2;

cqlsh:ass2> source 'table_declarations.cql';
cqlsh:ass2> describe tables;
data_point  time_table  driver  vehicle

cqlsh:ass2> copy driver from 'driver_data.txt';
6 rows imported in 0.379 seconds.

cqlsh:ass2> copy vehicle from 'vehicle_data.txt';
6 rows imported in 0.219 seconds.

cqlsh:ass2> copy time_table from 'time_table_data.txt';
30 rows imported in 0.334 seconds.

cqlsh:ass2> copy data_point from 'data_point_data.txt';
5 rows imported in 0.191 seconds
```

Question 14. [8 marks] Find nodes storing data of the driver pavle. Let these nodes be node_a, node_b, node_c, node_d, node_e, and node_f, where a < b < c < d < e < f.

i. [4 marks] Connect to `ass2` keyspace. Run the statement

   ```
   select driver_name, password from driver where
   driver_name = ‘pavle’;
   ```

   under consistency levels: `quorum, each_qourum,` and `local_quorum`. Run the select statement under consistency level `local_quorum` once for `dc1` being local, and once for `dc2` being local.

ii. [4 marks] Use `ccm` to stop node_e and node_f. Connect to `ass2` keyspace. Run the statement

   ```
   select driver_name, password from driver where
   driver_name = ‘pavle’;
   ```

   under consistency levels: `quorum, each_qourum,` and `local_quorum`. Run the `select` statement under consistency level `local_quorum` once for `dc1` being local, and once for `dc2` being local.

In your answer to the question, show results of your experiments and describe briefly what you have learned.

ANSWER

Things learned:

- Locality is determined by the data center of the coordinator node.
- (Assuming replication factor 3) If all nodes of a data center are up, and two nodes belonging to the same replication set of the other data center are down, then statements having consistency level `quorum` succeed. Also statements coordinated by a node from the center having all nodes up succeed at consistency level `local quorum`. Statements having consistency level `each quorum` fail regardless of coordinator's locality. Statements coordinated by a node from the data center with two nodes down fail at the consistency level `local quorum`.

Question 15. [10 marks] You are asked to find those nodes of the `multi_dc` Cassandra cluster that store replicas of the `train_time` table row

```
 line_name        | service_no | time  | distance | latitude   | longitude  |    stop
------------------+------------+-------+----------+------------+------------+------------
 Hutt Valley Line |          2 | 1045 |      34.3 | -41.2865 | 174.7762 | Wellington
```

Very soon you realized that all `ccm` and `nodetool` commands, except `ccm nodei cqlsh`, do not work. So, you are unable to use: `ccm stop, ccm status, ccm start, ccm nodei ring` and so on, including the command

```
   ccm nodei nodetool getendpoints ass2 time_table <key>.
```

Despite that, you have devised a procedure to find the nodes requested. In your answer, describe the procedure and show how you have applied it.

Hint: Luckily, you have saved the output of the `ccm nodei ring` command and `cqlsh` prompt is still working.

ANSWER

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

 line_name        | service_no | system.token(line_name,
service_no)
------------------+------------+-----------------------------
----------
 Hutt Valley Line |          2 |
2322329569350831795

(1 rows)
```

Looking at the output of the ring command in question 9, it follows that the token `2322329569350831795` belongs to node5 in dc1 and to node9 in dc2. Accordingly, data of Hutt Valley Line service number 2 are stored on:

- node5, node1, and node2 in dc1, and
- node9, node6, and node7 in dc2.

What to hand in:

- All answers both electronically and as a hard copy.
- A statement of any assumptions you have made.
- Answers to the questions above, together with the listing and the result of each query. In your answers, copy your CQL or `ccm` or `nodetool` command, and Cassandra message to it from the console pane. Do not submit contents of any tables.
- Please do not submit any .odt, .zip, or similar files. Also, do not submit your files in toll directory trees. All files in the same directory is just fine.

Using Cassandra ccm on a Workstation

ccm stands for Cassandra Cluster Manager. This is a tool that creates Cassandra clusters on a local server and thus it simulates a Cassandra network.

t the command line you need to type:

```
[~] % need ccm
```

to set up the environment. You may want to insert `need ccm` into your `.cshrc` file and thus to avoid typing it repetitively whenever you log on.

The `ccm` tool supports a great number of commands. In the Assignment 1, you will need only a few of them. To see the available `ccm` commands, type

```
% ccm
```

Many `ccm` commands have options. To see available options of a command, type

```
% ccm <command> -h
```

When running a `ccm` command, do not use a `–v` or `--cassandra-version` option. The proper version of Cassandra is already installed on our school network.

To create a Cassandra cluster, use `ccm create –n <no_of_nodes> <clster_name>`.

To see available clusters and which one is the current (designated by *), use `ccm list`.

To switch to another cluster, use `ccm switch <cluster_name>`.

To see the status of the current cluster, use `ccm status`.

To start the current cluster, use `ccm start`.

To stop the current cluster, use `ccm stop`.

To open a CQL session, use `ccm nodei cqlsh`.

To exit, from `cqlsh`, type `exit`.

Note: `ccm` commands will not work on any `netbsd` computers but that should not be a problem as almost all computers that students have access to nowadays are `Linux` boxes.

Warning:

- In all deployments the same ports are assigned to server nodes. After finishing a session you have to do **ccm stop** to stop all servers of your deployment and release ports for other uses. Failing to do so, you will make trouble to other people (potentially including yourself) wanting to use the same workstation. Later, if you want to use the same deployment again, you just do `ccm start` and your deployment will resume functioning reliably.

- **You are strongly advised to use Cassandra from school lab workstations.** The school does not undertake any guarantees for using Cassandra from school servers. You may install and use Cassandra on your laptop, but the school does not undertake any responsibilities for the results you obtain.