School of Engineering and Computer Science

SWEN 432
Advanced Database Design and Implementation

Assignment 2

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 node1 ring` command. Save the output of the ring command for future use and show it in the answer to the question.

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

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

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?

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

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?

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.

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.

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.

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.

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?

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

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

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.

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.

`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.
   ```

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.

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.

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.

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.

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.

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.

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.