School of Engineering and Computer Science

SWEN 432
Advanced Database Design and Implementation

Assignment 5

Due date: Sunday 08 June at 23:59 pm

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.

The Assignment has been written by Aaron Morton from The Last Pickle.

1. Overview

Acme Corporation sells an iPhone application that works as a data recorder for cars and trucks. The application uses mobile data connections to send information to Acme servers in real time. The information includes measurements such as the location and speed of the vehicle. They have recently received Venture funding and are looking to expand globally and grow by 100x. They have selected Cassandra as the database to use for this project. Your job is to design the database and advise on its use.

2. Application Requirements

Users

Users will create an account through the iPhone application. Each account has the following information:

1. `email_address`
   a. variable length unicode string
   b. email address for the user
   c. e.g. aaron@thelastpickle.com

2. `password`
    a. variable length unicode string
    b. password entered by the user, hashed on the iPhone
    c. e.g. "12ew5"

Vehicles

Once the user has an account they may register one or more vehicles. Each vehicle registration has the following information:

1. `email_address`:
    a. variable length unicode string
    b. email address of the user
2. `vehicle_id`
    a. variable length unicode string
    b. vehicle registration plate number, assumed to be globally unique
    c. e.g. "ABC123"
3. `colour`
    a. variable length unicode string
    b. selected from a list in the iPhone application
    c. e.g. red, white, black,…
4. `type`
    a. variable length unicode string
    b. selected from a list in the iPhone application
    c. e.g. sedan, hatch back, van, truck.

Data Points

After it has been installed and configured the application will automatically record information when the phone is in the vehicle. It knows when it is in the vehicle by bluetooth pairing, for now assume it is always in the vehicle. Each sampling of position and other information is considered a Data Point. The application records the time with the Data Point and either immediately sends it to the server or stores it to be sent later if there is no connection. The frequency of the Data Points depends on the speed of the vehicle. At most the application will sample once per second and no samples are created when stationary.

Each Data Point has the following information:

1. `sequence`
   a. timestamp
   b. time to the millisecond when the iPhone application created the Data Point
   c. e.g. "2011-02-03T04:05:00+0000"
2. `latitude`
   a. double
   b. used with `longitude` for position
   c. e.g. -41.288142
3. `longitude`
   a. double
   b. used with `latitude` for position
   c. e.g. 174.776676
4. `speed`
   a. double
   b. speed in kilometers per hour
   c. e.g. 37.5

Metrics

Acme corp. wants to understand who are the most prolific users of the application so they can charge them later. No decision has been made on how the information will be analyzed, only that they want to collect it. There are two metrics they want to track:

1. For each user, the number of vehicles they have registered.

2. For each user, the number of data points they have recorded each day.

User Interface

Users will be asked to log into the iPhone application, when this happens the iPhone will hash the password and send it to the server for checking.

Users may see information about the vehicles they have registered through the iPhone application. The application presents the users with the following information:

* A list of the vehicles they have registered.

* For each vehicle a list of the days it has been active.

* For each vehicle the most recent location and speed for a given day.

* For each vehicle a track of the locations for a specified time span for a given day.

Consistency Requirements

The product team has agreed to the following Consistency requirements:

1. Reading User and Vehicle data must be strongly consistent.

2. Reading Data Point may be eventually consistent.

Infrastructure Requirements

Initially the iPhone application will only be available in the USA, where the servers will be deployed to a single Amazon Web Services (AWS) Region. Over time the application may be available in other countries, and the servers will be deployed to other AWS Regions. At all times there should only be one global database for the application.

The initial deployment of the application will involve:

1. One AWS region which will be called `us-east-1`

2. A total of 6 nodes.

3. Using Cassandra 2.0.7

Availability Requirements

The infrastructure team has agreed to the following Availability requirements:

1. The cluster must provide Strong Consistency for 100% of the data when one node is down.

1. The cluster must provide Strong Consistency for 100% of the data when one Availability Zone is down.

Assignment

Question 1. [15 marks] List the read and write requests the application requires using plain English.

Question 2. **[20 marks]** Create data model using CQL 3 statements that supports the requirements.

a. **[5 marks]** For the Keyspace include the replication strategy and the minimal value of the Replication Factor that will support Availability Requirements above.

b. **[15 marks]** For the Table definitions include any non default property settings.

Question 3. **[20 marks]** Provide CQL3 statements to support each of the application requests specified in question 1.

Question 4. **[10 marks]** Identify in plain English the *Consistency Level* the developers should use for each of CQL3 statements (identified in question 3) to achieve the consistency requirements.

Question 5. **[15 marks]** Provide an example of a Strong Consistency insert for User data succeeding and then failing in the `cqlsh` application with a 6 node cluster. For both the succeeding and failing case provide the `INSERT` statement, the Consistency Level, and the state of the cluster via the `nodetool status`. For the purposes of this question all nodes may be in the same Cassandra Data Centre and Rack.

Question 6. **[15 marks]** Provide an example of an Eventual Consistency insert for a Data Point succeeding with one node down and then failing in the `cqlsh` application with a 6 node cluster. For both the succeeding and failing case provide the `INSERT` statement, the Consistency Level, and the state of the cluster via the `nodetool status`. For the purposes of this question all nodes may be in the same Cassandra Data Centre and Rack.

Question 7. **[5 marks]** Describe how many AWS Availability Zones the cluster should be deployed to and how many nodes per Zone to achieve the Availability Requirements.

***Note*:** When creating a cluster on your workstation the default data centre name will be `datacenter1`. This may be used in the schema rather than the name of the AWS Region.

References

* CQL3 language reference from Data Stax
`http://www.datastax.com/documentation/cql/3.1/cql/cql_intro_c.html`

* CQL3 formal language definition
`https://github.com/apache/cassandra/blob/cassandra-2.0/doc/cql3/CQL.textile`

* CCM (already installed on campus) `https://github.com/pcmanus/ccm`

Sample Data

The following sample data can be used for testing:

Users:

email_address, password

'fred@fred.com', '1234'

'bob@bob.com', '5678'

'jane@jane.com, '9012'

Vehicles:

email_address, vehicle_id, colour, type

'fred@fred.com', 'abc123', 'red', 'sedan',

'fred@fred.com', 'def123', 'blue', 'truck',

'bob@bob.com', 'ghi123', 'white', 'van'

'jane@jane.com, 'jkl123', 'black, 'hatch back'

Data Points:

sequence, latitude, longitude, speed

2014-05-26 10:49:40+1200, -39.968, 174.8, 30

2014-05-26 10:49:10+1200, -40.66, 176.5, 34

2014-05-26 10:48:40+1200, -41.3, 176.06, 38

2014-05-26 10:48:10+1200, -39.523, 175.89, 37

2014-05-26 10:47:40+1200, -40.081, 175.44, 34

2014-05-26 10:47:10+1200, -40.478, 174.8, 36

2014-05-26 10:46:40+1200, -40.597, 176.87, 35

2014-05-26 10:46:10+1200, -41.3, 176.23, 35

2014-05-26 10:45:40+1200, -39.798, 175.59, 35

2014-05-26 10:45:10+1200, -40.496, 175.06, 32

2014-05-26 10:44:40+1200, -41.3, 174.87, 34

2014-05-26 10:44:10+1200, -39.394, 174.8, 33

2014-05-26 10:43:40+1200, -39.399, 176.58, 32

2014-05-26 10:43:10+1200, -39.911, 176.55, 29

2014-05-26 10:42:40+1200, -40.27, 175.94, 33

2014-05-26 10:42:10+1200, -40.469, 175.73, 37

2014-05-26 10:41:40+1200, -40.9, 174.8, 37

2014-05-26 10:41:10+1200, -41.3, 176.42, 36

2014-05-26 10:40:40+1200, -39.518, 175.56, 35

2014-05-26 10:40:10+1200, -40.45, 174.8, 39

2014-05-26 10:39:40+1200, -40.565, 176.97, 36

2014-05-26 10:39:10+1200, -41.3, 176.31, 33

2014-05-26 10:38:40+1200, -39.041, 175.62, 30

2014-05-26 10:38:10+1200, -39.074, 174.8, 30

2014-05-26 10:37:40+1200, -39.357, 176.91, 27

2014-05-26 10:37:10+1200, -39.737, 176.34, 27

2014-05-26 10:36:40+1200, -40.686, 175.49, 26

2014-05-26 10:36:10+1200, -41.3, 174.8, 23

2014-05-26 10:35:40+1200, -39.696, 176.81, 20

2014-05-26 10:35:10+1200, -40.164, 176.61, 24

2014-05-26 10:34:40+1200, -41.099, 176.43, 28

2014-05-26 10:34:10+1200, -41.3, 175.86, 32

2014-05-26 10:33:40+1200, -39.015, 175.83, 36

2014-05-26 10:33:10+1200, -39.325, 174.95, 33

2014-05-26 10:32:40+1200, -39.937, 174.93, 33

2014-05-26 10:32:10+1200, -40.118, 174.8, 30

2014-05-26 10:31:40+1200, -40.953, 176.93, 34

2014-05-26 10:31:10+1200, -41.3, 175.95, 34

2014-05-26 10:30:40+1200, -39.206, 175.59, 34

2014-05-26 10:30:10+1200, -39.833, 175.55, 31

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 command, and Cassandra message to it from the console pane. Do not submit contents of any tables.

Using Cassandra ccm on a Workstation

At the command line you need to type:
`% need ccm`
to set up the environment.

Scripts available to use:
`% ccm_create` (to create and start a cluster of 3 nodes called `training`)
`% ccm_stop` (to stop the cluster)
`% ccm_remove` (to remove the cluster and it's data)
`% ccm_cli` (to start the Thrift terminal connected to node 1 in the cluster)
`% ccm_cqlsh` (to start the CQL terminal connected to node 1 in the cluster)
`% ccm_` (the standard and the most important script that gives all `ccm` options)

This 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.