Rewrite Cassandra DataStore Implementation

Student Name: Madhawa Gunasekara
Student Email: madhawa30@gmail.com
JIRA Issues: https://issues.apache.org/jira/browse/GORA-497

Introduction

Apache Gora is an object to datastore mapping data persistence framework similar to Object-relational mapping frameworks such as Hibernate, JPA, however, Apache Gora stands out from these by providing an easy-to-use in-memory data bean model and more importantly data store specific mappings which allow to utilize full data model of non-relational/NoSQL databases. Gora currently supports persistent datastores that write to HBase, Cassandra, MongoDB, DynamoDB. Gora uses Avro as serialization mechanism.

Definition of the Problem

Apache Cassandra is widely used open-source distributed NoSQL database management system. Cassandra has been designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra is one of the widely used databases in the industry. Apache Gora Cassandra module is written based on thrift connector (Hector connector) but currently, Cassandra developers have removed the thrift connector and have moved into CQL (Cassandra Query Language), which is very much similar to SQL. So the Cassandra community has migrated to CQL. Therefore the latest Cassandra databases servers which use CQL 3,4 versions are not compatible with gora cassandra module. Therefore current gora cassandra module is little bit out dated.

CQL performance is much better than thrift (Hector) [1]

Proposed Method

To enhance the usability of CQL Feature we should support multiple serializations, like native Cassandra serialization, Avro, kryo serializations to the users. So I'm hoping to support multiple serializations as well. I will use https://github.com/datastax/java-driver driver for the development since this driver is the widely used java cassandra driver.

```java
public class CassandraStore<K, T extends PersistentBase> extends DataStoreBase<K, T> {

public enum SerializerType {
    AVRO("AVRO"), NATIVE("NATIVE"), BINARY("BINARY");

    String val;
    SerializerType(String v){
        this.val = v;
    }
}
```

There are new properties has been introduced by the cassandra developers when connecting to a cassandra database, So willing to support those as well [2],[3].

We can use an enhanced version of current gora cassandra mapping.
<gora-otd>
<keyspace name="Employee" host="localhost" placement_strategy="org.apache.cassandra.locator.SimpleStrategy" replication_factor="1" cluster="Gora Cassandra Test Cluster">
    <family name="p" gc_grace_seconds="5" />
    <family name="f" gc_grace_seconds="5" />
    <family name="sc" type="super" />
</keyspace>
<class name="org.apache.gora.examples.generated.Employee" keyClass="java.lang.String" keyspace="Employee">
    <field name="name" family="p" qualifier="info:nm" ttl="10" />
    <field name="dateOfBirth" family="p" qualifier="info:db" ttl="10" />
    <field name="ssn" family="p" qualifier="info:sn" ttl="10" />
    <field name="salary" family="p" qualifier="info:sl" ttl="10" />
</class>
</gora-otd>

Deliverables

1. Implementing cassandra module to support CQL
2. Writing Integration tests to cover basic scenarios
3. Documenting the usage and the functionality of the module

Project Timeline

<table>
<thead>
<tr>
<th>Activity</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community Bonding Period : familiar with gora project, Finalize the requirements and design</td>
<td>04-05-2017</td>
<td>29-05-2017</td>
</tr>
<tr>
<td>Iteratively develop the prototype by adding functionality to connect with cassandra server with multiple serialization support</td>
<td>30-05-2017</td>
<td>18-06-2016</td>
</tr>
<tr>
<td>Testing the functionality of the current implementation and improve the functionality</td>
<td>19-06-2017</td>
<td>25-06-2017</td>
</tr>
<tr>
<td>Midterm Evaluations : Submit the work done so far and discuss.</td>
<td>26-06-2017</td>
<td>30-06-2017</td>
</tr>
<tr>
<td>Complete the gora-cassandra module and test the functionality</td>
<td>01-07-2017</td>
<td>20-07-2017</td>
</tr>
<tr>
<td>Documenting the usage of gora-cassandra module</td>
<td>21-07-2017</td>
<td>28-07-2017</td>
</tr>
<tr>
<td>Final Evaluations : Submit the completed project with documentation</td>
<td>29-07-2017</td>
<td>29-08-2017</td>
</tr>
</tbody>
</table>

Availability
I will be able to allocate more than 40 hours of work for this project in the coding period. In the weekdays I will be working in the evening and in weekends full time. Also it is worth to mention that within the community bonding period I can work full time implementation of this project. Currently I’m having some exams till April 22nd.

Additional information

I’m Madhawa Gunasekara, a final year student studying Engineering in Electronics, Telecommunication and Computing at IESL College Of Engineering, Sri Lanka. I’m an open source enthusiast and I use many of open source products for my personal use and for academic purposes. Research areas that I’m interested are SOA, Web services and Distributed systems.

I also have open source industry level training experience with WSO2 (an open source middleware company) organization where I was an Intern for more than 7 months. I’m also WSO2 Committer, I have hands on experience some of their matured open source projects such as WSO2 ESB, WSO2 Data Analytics Server and WSO2 Data Service server etc. I am fully adapted to open source software model at the industry level.

My motivation in taking part in this year GSoC is to get the chance to actively participate in a prominent open source project of Gora.

My open source contributions can be found in github account[7]

Gora Contributions

![GORA-498 - Adding MongoDB Authentications](https://github.com/madhawa-gunasekara)

References

[7] [https://github.com/madhawa-gunasekara](https://github.com/madhawa-gunasekara)