Cassandra Browser Tool

(Using Cassandra and Python)

Group Number 13

Nitin Agarwal Prajit Patil Pushyami Srikanth R

PROBLEM STATEMENT :

Create a Cassandra Browser tool in Cassandra using python with the following functionalities.

1. **Keyspace Operations** :
   Implementing a browser tool for creating a keyspace, retrieving information about a keyspace and all keyspaces, retrieving single property about a keyspace and dropping keyspaces.

2. **Column Family Operations** :
   Implementing a browser tool for creating a column family, indexing a column family, get the description of a column family, dropping column families.

3. **Rows and Columns Operations** : Implementing a browser tool for inserting rows and columns, deleting rows and columns.

APPROACH TO THE PROBLEM :

1. Two of our team members (Prajit and Nitin) will work on the backend part of the project as administrators ie., implementing the functionalities in python and managing the database.

2. Other two team members (Shrikant and Pushyami) will work on the frontend part ie., developing the user interface and web layout of our project application listing out all the functionalities for the user to access and retrieve information from the database.

TOOLS TO BE USED :

1. **Cassandra 2.0.1** : The Apache Cassandra database provides high scalability and

high availability without compromising performance. Linear scalability and prove fault tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra's data modeloffers the convenience of column indexes with the performance of log structured updates, strong support for denormalization and materialized views, and powerful built in caching.
Basically this will provide as a platform for the database management server.

2. **Python Programming Language**

3. **Pycassa** : Python client API with features such as connection pooling, SuperColumn support, and a method to map existing classes to Cassandra column families.

4. **Webob** : WebOb is a library to create a request and response object. It's centered around the WSGI model. Requests are wrappers around the environment.
WSGI more specifically is made up of an *application* and a *server*. The application is a function that receives the request and produces the response. The server is the thing that calls the application function.

5. **Jinja2 Templates** : Jinja2 is one of the most used template engines for Python. We will be using to develop our User frontend interface.

DIVISION OF TASKS AND MODULES :

- **Cassandra Database Server**
  - Nitin and Prajit
- **Data Handler**
  - Prajit and Nitin Agarwal
- **Web User Interface**
  - Pushyami and Srikanth
- **Cassandra Client API**
  - Nitin and Prajit
- **Preparing Documentation**
  - Nitin, Prajit, Pushyami and Srikanth
- **Miscellaneous**
  - Prajit and Nitin

PLANS :

PHASE 2 Deliverables :

- Setting up Cassandra
- Developing the Architecture of our project application.

PHASE 3 Deliverables (Final) :

- Configuring Cassandra cluster with two to three nodes.
- Implementing the functionalities :
  - Create a keyspace
  - Retrieving information about a keyspace.
  - Retrieving information about all keyspaces.
  - Retrieve single property about a keyspace.
  - Dropping keyspace.
  - Creating column family.
  - Indexing column family.
  - Get description of column family.
  - Dropping column family.
  - Inserting rows and columns.
  - Deleting rows and columns.
- Web interface that will allow user to interact with the database server.

These are the basic functionalities we might extend our functionalities.

REFERENCES :

- Tutorial for the Python client API http://pycassa.github.io/pycassa/tutorial.html
- Cassandra Cluster Admin a GUI tool to help people administrate their Apache Cassandra cluster. https://github.com/sebgiroux/Cassandra-Cluster-Admin
- Helenos, a web based GUI Cassandra client that helps you to explore data and manage your schema. https://github.com/tomekkup/helenos
- DataStax OpsCenter shows how to you manage our data infrastructure. It allows administrators, architects and developers to manage, monitor and control even the most complex database workloads with point-and-click ease from a centralized web browser. http://www.datastax.com/what-we-offer/products-services/datastax-opscenter