Graduation Project Report

# (1) Progress done:

1. Yarn/Twill.
2. Query Builder.
3. Avro + Parquet file formats.
4. SeeDB implementation.

# (2) Details of each point:

1. Implemented a hello world Yarn App to run on the HDFS using Twill API, also the application could now creates a new file on the HDFS. Now it is ready if want to use it in our application if we want to parse the semi structured into parquet or avro file format on the HDFS and not local.
2. A JDBC API ready to be connected to Impala to execute and retrieve queries from Impala. This will be used in the backend.
3. Writing Avro files and parquet which Impala uses to execute its queries. Impala has a custom command "LoadData" which takes an avro or parquet file and load the data in it into Impala table. If we used Twill to parse the semi structured files, we will use "LoadData" command to load the data in the required table. On the other hand we could insert into Impala table while we are parsing the semi-structured file on the go, Instead of using Twill and writing into one of the formats. Also we want to see whether parquet is faster or avro so we document why we used one of them.
4. Implementing SeeDB, now we recommend visualizations to the user, but it is limited to bar charts we used multiple distance metrics to document which is better to use. We also did some optimization to the algorithm which is shared based optimization where we execute only one query each time not two queries, we combined the two queries into one query so we save the time our backend goes and access Impala database.

# (3) Project Flow:

1. First the user will upload the data he needs to visualize, these data will be in the form of CVS or Jason, These data will be changed to a structure format readable by Impala which is Avro or Parquet. We have two choice here (A) Create an empty parquet file and insert the data in it on the go will we are parsing the file, we will not use YARN or the hdfs here. (B) Parse the file and change into Parquet file first using YARN and saving this file on the HDFS, then using the "LoadData" command to insert the data into the table we will use later on our visualizations.
2. The user will enter some keywords in the web UI to query on, he will specify the x-axis / the y-axis and the aggregate function in the y-axis. Also he must enter attributes to filter the data and the recommendation visualizations will change according to these attributes. The backend which will Java will take those input and by using the query builder [ JDBC API ] and execute the queried on Impala and return back the main visualization which the user specifies and a group of recommendations which will be interested to him.

# (4) To Do:

1. Java web server Backend.
2. Front End.
3. Installing multi-node Impala on the cluster.
4. More optimized Parser: CSV, and parsing Jason.
5. Optimization on our recommendation: ex. sampling.
6. Figuring a way to visualize scatter plot rather than just bar charts.