**PROJECT PHASE II REPORT**

**PRINCIPLES OF BIG DATA**

**CS 5590 PB**

**DONE BY,**

**DEEPTHI JONNALAGADDA**

**MANOJ KUMAR MARDHANASETTI**

**PREETHI REDDY PESARU**

1. **INTRODUCTION:**

We are using MongoDB as our No SQL tool. MongoDB is a Document oriented database. It helped us in using JSON documents which made the data integration process easier and faster.

The main features that made us choose MongoDB are: schema free database, it supports Ad hoc queries , replication, aggregation, load balancing.

**Mongo Shell**:

The mongo shell is an interactive JavaScript shell that is included with all MongoDB distributions.

All the commands supported by MongoDB can be issued through the shell, including administrative operations. The mongo shell is a popular way to interact with MongoDB for ad hoc operations, key value queries, range queries, Geospatial queries, text search, aggregation framework and map reduce queries.

1. **ARCHITECTURE:**

Client (Front-End visualization)

Server

MongoDB

1. **IMPLEMENTATION:**

Firstly, we collected the tweets based on some keywords and then imported the data to MongoDB using the tool Eclipse IDE. Using the mongo-java-deriver 2.9.3 we connected our mongo shell to the eclipse. We dragged the data from mongodb to the front end application using eclipse IDE. Here, we have executed different queries on our twitter data and brought the output in the form of different visualisations.

* Our twitter data size is 1.953GB.
* The name of our Database in MongoDB is “BigDataProject” and the collection is “PB”.
* We connected using the localhost with the port number #27017.

1. **Queries:**

We connected the Eclipse IDE tool to the MongoDb with the help of mongo-java-driver -2.9.3

a) **Counting the number of tweets based on a key word:**

With the available data in the collected tweets, we used the keywords like Windows, Android, Blackberry, IOS, Mac, Linux and Ubuntu. With those keywords we ran the java program in Eclipse IDE and visualized the data in the form of a pie chart. We used the find() to count the data.

b) **Finding the number of tweets based on a Language using Aggregate():**

There are several languages in the collected data. We used the aggregation operation to group the languages and sorted them based on the number of counts. Here the language "en"- English has the highest count followed by the other languages. We visualized the output with the help of a bar graph.

**c) Sorting the tweets based on country using a keyword:**

We tried to sort the particular tweet keyword based on the country. Here we used find() and with the help of projection we displayed the tweets. The user has to input the name of the country and the keyword he wish to search for. The output will visualize the tweets.

**d) Finding the followers count, retweet count and other details of a particular user:**

By entering the screen name of the user, one can find the followers count, retweet count, favorites count, statuses count and friends count. We used the find () to find the counts and projected those counts.

**e) Tweets ranging between the dates:**

The user needs to enter the starting date and the ending date to find the tweets ranging between those. This is done with the help of "$gt" and "$lt".

1. **Testing:**

We had done some manual testing in the Eclipse IDE and then compared the results with the Mongo shell.

1. **References:**

* <http://www.mongodb.org/>
* <http://www.infoworld.com/article/2608083/application-development/do-twitter-analysis-the-easy-way-with-mongodb.html?page=3>
* <http://tweettracker.fulton.asu.edu/tda/TwitterDataAnalytics.pdf>
* <http://en.wikipedia.org/wiki/MongoDB>
* <http://cs.brynmawr.edu/Courses/cs110/fall2003/Applets/PieChart/PieChart.html>
* <http://info.mongodb.com/rs/mongodb/images/MongoDB_Architecture_Guide.pdf>