CSC:591 - Data Intensive Computing:
Project Status Report

GROUP NUMBER 11:

TweetSmart
(Smart Twitter Notification System)

- TweetSmart is a data-intensive computing system for dynamically determining the top 5 interests of each user and notifying them about the most recent updates regarding those interests.
- A data intensive solution is implemented to query this distributed database containing historical and streaming data to calculate those interest values with a few parameters.
- This framework will help to analyze topics of interest over time, locations, age of users, etc. and enable smarter notifications and targeting of advertisements.

Team members:

Saurabh Shanbhag, sshanbh2
Sayali Godbole, ssgodbol
Aishwarya Sundararajan, asundar2

DELIVERABLES

1. **Design:**
   - Designing the components of system.
   - Determination of the amount of data (likes * usernames) to be processed and stored.
   - Determine number of nodes for distributed data storage and retrieval.
2. **Getting and Storing Twitter Data:**
   - Mining streaming data from Twitter and storing it in the Amazon Dynamo/Cassandra database.
3. **Using Hadoop/ZooKeeper for Computation:**
   - Distributing the data efficiently and computing top interests (Mapreduce) for each user.
4. **Implementation:**
   - Getting Top Interests: Retrieving and updating top interests continuously.
   - Watcher for new posts: determining users to be notified when a person (top interest) posts content.
5. **Displaying results/Notification:**
   - Notifying user on new interesting post.

STATUS

1. **Design:**
   - Finalized Data Structure.
   - Finalized amount of data and number of nodes (15000 users).
2. **Getting and Storing Twitter Data:**
   - Using tweepy library, retrieved data about usernames and likes.
3. **Using Hadoop/ZooKeeper for Computation:**
   - Implemented prototype in python to compute top five interests of a user (to be converted to map and reduce jobs).
4. **Implementation:**
   - Scaling to multiple users is in progress.
   - Watcher: *pending.*
5. **Notifying users***: pending*

ISSUES

- Getting active usernames for the sample data.
- Determining optimal intervals for API requests for new posts and likes.
- Access issues for notifications.
- Parallelizing/distributing the work among nodes to get faster computation.
- Determining the right hosting server from available options. (VCL/AWS)
- Validating that the users are notified of the right content by cross-checking the Cassandra database for their topics of interest.
- Dealing with the time-lag where new topics of interest are added for a specified user in the database while content about those topics need to be notified to the users simultaneously.