DATA WAREHOUSING SOLUTION USING APACHE SPARK

Our project is to create a data warehouse using apache spark and related technologies. We are creating a data warehouse for movies and ratings. We have used movielens 20M dataset containing 20 million movie ratings for creating a data warehouse like environment. Implementation of full-fledged data warehouse operations is a task that requires a lot of data and computational resources and therefore, some basic operations have been implemented for demo.

Dataset:
The dataset used for building the warehouse is movielens 20M dataset. The dataset contains 4 CSV files of which, 2(movies.csv and ratings.csv) are useful for this project. Some stats about the data:

- 20 million ratings
- 465,000 tag applications
- 27,000 movies
- 138,000 users

Use cases:

- Ad-hoc queries like:
    - find best rated movies by year
    - find best rated movies by genre
    - mix of the above two
    - draw time vs rating graph for a movie
    - movie recommendation

Workflow:

- Entire data resides in Hadoop filesystem (about 550MB)
- Querying is performed from the browser with the help of REST API calls. Each query has a REST URL associated with it.
- Each REST API calls hits a python function which is mediated by python flask.
- For each query, the data is read from HDFS and converted to spark's own resilient distributed datasets(RDDs). RDDs support querying out of the box and are supported by various other spark features(MLib, graphX).
- For graph queries, graphX has been used.
- For the movie recommendation engine, spark MLib has been used. Spark MLlib library for Machine Learning provides a Collaborative Filtering implementation by using Alternating Least Squares.

- The model was trained using smaller dataset to learn ALS parameters and then the final model was created after training using the entire dataset.

Setting up the environment:
Download apache spark from http://spark.apache.org/downloads.html. We are using spark version 1.5.2 pre-built version for hadoop 2.6.

Untar the .tar file and ensure that spark works by kickstarting spark shell. (instructions for starting spark shell can be found online)

Download dataset from grouplens website and unzip. Place this folder where your spark is setup.

Download the code from the github repo https://github.com/ChilupuriAnilReddy/Cloud-Major-Project-Team-18. The scripts folder contains the code for the project. Place this along with spark and dataset folders.

Enter the spark folder and run this command:

```
./bin/spark-submit Cloud-Major-Project-Team-18/Scripts/flaskprogram.py
ml-20m/movies.csv ml-20m/ratings.csv
```