PROJECT 1: CONTENT RETRIEVAL, STORAGE AND ANALYSIS FOUNDATIONS OF DATA-INTENSIVE COMPUTING

Purpose:

1. To understand the components and core technologies related to content retrieval, storage and data-intensive computing analysis
2. To explore designing and implementing data-intensive (Big-data) computing solutions using MapReduce (MR) programming model on Hadoop cluster hosted by Center for Computational Research (CCR).
3. To visualize the data using appropriate tools,
4. Prepare a detailed project report that documents this project.

Problem Statement:

In order to aggregate interesting data and also to keep up with the "trends" we will aggregate data from Twitter. (Why Twitter?) We will collect Twitter data for different ranges of dates (week-range, month-range). Aggregated raw data needs to be cleansed to some extend before analyzing it using Big-data methods. We will analyze the data for (i) simple wordcount (so as to get used to MR) (ii) counts only on the #tagged words (iii) counts only on @xyz words and (iii) some more deeper analysis such as what is trending this week vs. last week? vs last month? and clustering of the tweets by dominant theme, etc.

Preparation before lab:

1. Review your Java/Python language skills by working on the sample application that will be given to you.
2. Understand the MR model and modeling your data as <key, value> store.
3. You must have a clear understanding of a client-server system operation and also three-tier application development.
4. Run the *wordcount* program on the virtual machine on your laptop and make sure you understand the code for the Mapper, Reducer, and the main MR job configuration.

Application Architecture

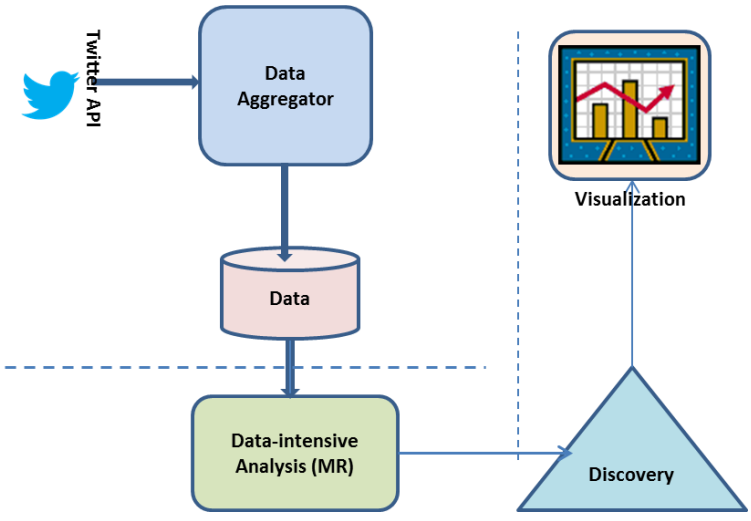

Figure 1: System Architecture for Data-intensive Analyzer

Project Implementation Details and Steps:

1. Data aggregator: Twitter Developer API is available at https://dev.twitter.com/docs You can it to understand the details of what is available. However there are many aggregators already available online. You are allowed to use any of these. I will let you find one of these. (I have used one written in Python.)
2. Clean the unwanted details from the tweets and save them in a set of regular files in a designated directory. How many tweets to collect? Initially as you will develop the prototype with smaller Big-data (!) so that you can get the aggregator working without any problem. Then you will scale it up to getting a large set. What is considered a large set? I would say 200,000 – 500,000 tweets.
3. Now design and implement the various MR workflows to extract various information from the data. (i) simple wordcount (ii) trends (iii)#tag counts (iv)@xyz counts etc.
4. Render the discovered knowledge using appropriate visualization methods.
5. The basic MR algorithms can be improved for performance using the knowledge about the data. For example, typically it is not a best practice to have more than two #tags in a tweet. So once you locate 2 #tags, you can move on to the next tweet.
6. Document the complete process and all the MR source code and the screen shots of visualization.
7. You could also create a simple HTML5 (Javascript) web access to the knowledge base you created. (optional)

Project Deliverables:

1. A report providing all the details of the project:
    a. Data format and source
    b. Amount of data collected and details
    c. Aggregator details, from the scratch or used exiting one, modified etc.
    d. MR (mapper, reducer pseudo code)
    e. Configuration of the cluster used
    f. Outputs: different outputs, and visualizations
2. Tar bundle and a README so that I can run the program repeating your MR deployment and execution.

    More details will be given as necessity arises.

Submission Details:

submit_cse487 files separated by space
submit_cse587 files separated by space