**DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION**

*For a detailed explanation please watch the video included in our submission*

**OVERVIEW:**

In this lab, we aggregated data from two different sources - research based and opinion based - for the same keyword. The two datasets were processed using classic big data analytics using Hadoop VM and the outcomes were compared using WordCloud.

**TOPIC SELECTION:**

For aggregating data, we chose topics that are of current interest in the United States, and which yielded a good amount of information. Initially, we chose “Gun Control” as our keyword, and later also selected “North Korea” as another keyword.

**DATA COLLECTION:**

Data was collected from two different sources, namely New York Times news articles using NYTimesArticles API and Twitter, using RTweet API respectively. The New York Times articles were collected using Python Language, while Twitter tweets were collected using R language. Initial data collected was for “Gun Control” over only one day. This is our small dataset. For a larger dataset, data was collected over one week. The large dataset was done for both topics – “Gun Control” and “North Korea”.

**WORD COUNT:**

After importing the VM appliance for Hadoop infrastructure, the aggregated data was loaded into the VM in two directories – NewsData and TwitterData. The mapper and reducer, coded in Python, was executed on all of the datasets. The mapper took the raw aggregated datasets, cleaned and parsed them into words, and also removed stop words. Cleaning was done by splitting the data based on spaces and some characters that appeared in our datasets. Stop words were removed using the nltk library. The Mapper emits out <useful words,1>. The reducer the counts and aggregates the useful words and outputs a sorted word list in decreasing order of word count. The useful words are stored in NewsData -> NewsWords and TwitterData -> TwitterWords.

**CO-OCCURRENCE:**

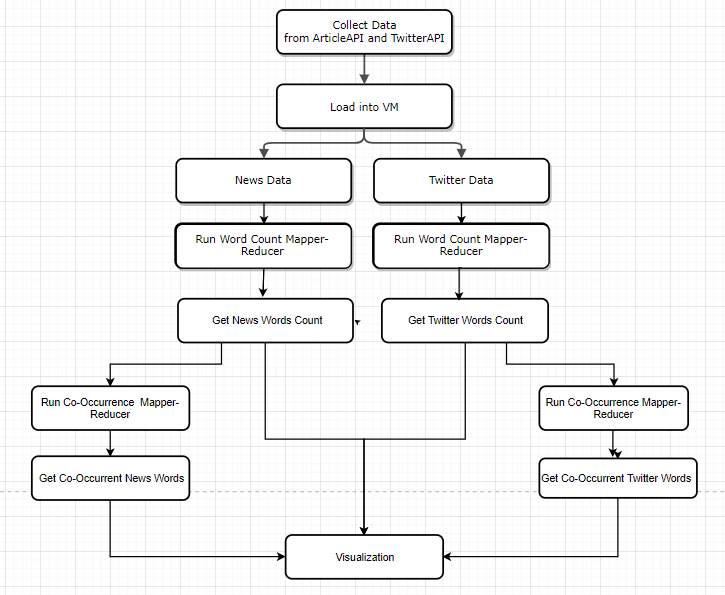
From the output of Word Count Reducer for the small dataset of both news data and twitter data for “Gun Control”- we chose the top twenty words from each set and ran the Co-occurrence Map-Reduce for each of these words. The neighbor to select the co-occurring word for an NY Article is one whole article and for Twitter tweets is one tweet. The mapper takes the top twenty words and emits a pair <word, co-occurring word>. The reducer emits the count for each pair of co-occurring words.

**VISUALIZATION:**

A web page was created for visualizing and comparing all these outputs. We fed the results into the web page by embedding d3.js code in it. The page has replaceable word clouds to compare News Data and Twitter data for either small dataset (only “Gun Control” topic collected over one day) or large dataset (both “Gun Control” and “North Korea” topics collected over one week). We also visualized co-occurrence in the small dataset by displaying a table with the count of a word’s co-occurrences when that particular word is selected in a drop-down.

The output files from the reducer output is transformed into json objects for ease of Html access. This transformation can be found in the R notebook submitted.

**BLOCK DIAGRAM:**



**REFERENCES:**

* http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/
* <https://chandramanitiwary.wordpress.com/2012/08/19/map-reduce-design-patterns-pairs-stripes/>
* wvengen’s d3.js cloud sample
* <http://www.pythonforbeginners.com/python-on-the-web/web-scraping-with-beautifulsoup/>
* <http://dlab.berkeley.edu/blog/scraping-new-york-times-articles-python-tutorial>