Project_group 12
Gunjan Singh (gs896)
Meghana Tumkur Narendra (mt1080)
Praveen Pinjala (pp813)
Shikha Vyaghra (sv629)

CS512 FINAL PROJECT PROPOSAL
WordCount and PageRank using MapReduce

- ## Overview:

  Our team desires to explore different applications of Map reduce. Map reduce is a framework of processing parallelizable problems across large datasets using a large number of computers(nodes). We will be using this framework for word count and Page ranking.

  1. **WordCount:** This is the simple application of Map reduce. This is helpful when we have a huge dataset and would like to count the frequency of words. This is achieved through the following functions:
     A. *Map Function* – It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pair).
     (Bus, Car, bus, car, train, car, bus) => ((Bus, 1), (Car,1), (bus,1), (car,1), (train,1),(car,1), (bus,1))

     B. *Reducer* – Takes the output from Map as an input and combines those data tuples into a smaller set of tuples.
     ((Bus,1), (Car,1), (bus, 1), (car,1), (train,1), (car,1), (bus,1)) => ((Bus, 3),(Car, 3), (train, 1))

  2. **PageRank:** This is an algorithm used by Google Search to rank web pages in their search engine results. PageRank is a way of measuring the importance of website pages. We will be implementing the same by using the MapReduce algorithm. The below are the steps we are planning to follow to achieve this:
     A. The Map function will be used to map the input text to a format which will be used by the reducer.
     B. The Reducer will take the input from the mapper and will calculate the page rank by using page rank formula.

- **Methodology:**

  As stated above, we will be using Map reduce methodology for both word count and Page Rank applications.

1. WordCount:

A. *Splitting* – The splitting parameter can be anything, e.g. splitting by space, comma, semicolon, or even by a new line ('\n').But we will be splitting it on the basis of space.

B. *Mapping* – We will map splitted keys as tuples of (Key, Value).

C. *Intermediate splitting* – the entire process in parallel on different clusters. In order to group them in "Reduce Phase" the similar KEY data should be on the same cluster.

D. *Reduce* – In this step we will perform grouping by phase.

E. *Combining* – The last phase where all the data (individual result set from each cluster) is combined together to form a result.

2. PageRank:

A. *Mapping* : The mapper will be used to produce the output for each node. Here each node refers to the weblink. The output will be the initial rank and other outlinks to the weblink.

B. *Reducer*: Receives for each node a list of values (the values can be either page ranks or the list of the outlinks), calculates the pagerank by using the formula (1 - d) + (d * sum(pageRank)). This step is repeated X times (X will be given in the argument).

- **Dataset:**

  In the case of WordCount, simple files will be used to store the test data.

  In the case of PageRank, we will be using the dataset in the format "NodeName NodeRanking OutNode1 OutNode2…"

- **Technology:**

  We will be using following technologies:

  1. Java
  2. Hadoop