Distributed Computing and Big Data Technologies Assignment 1 Project Report

TEAM COMPOSITION:

Mr. Akshat Sehgal (UB Person Id – 50198939) – Pink Code (2 pm)

Ms. Saatchi Nandwani (UB Person Id -50207363), Batch – Blue Code (8 am)

CONTRIBUTIONS:

TASK	CONTRIBUTOR
VM & Hadoop DFS Setup	Saatchi/Akshat
Map Reduce Logic	Akshat
Gather inputs	Saatchi
Project Report	Saatchi/Akshat

HADOOP DFS SETUP AND YOUR ENVIRONMENT:

- We installed Oracle VM on our respective Windows machines and used the Hadoop VM image provided by the TA.
- The Linux VM provided to us had Hadoop 2.7.2 and Eclipse 3.1 installed on it.
- We setup Hadoop on the provided VM by providing our JAVA_HOME and HADOOP_CLASSPATH paths.
- We modified configuration files like core-site.xml. mapred-site.xml, hdfs-site.xml and yarn-site.xml to setup Hadoop as a single node.

DATA SETS:

We worked with text from some old historic newspaper articles. We obtained the data from the below URL:
http://chroniclingamerica.loc.gov/.

We worked on 39 articles, decided on the basis of last 2 digits of the person number (50198939).

To concatenate our text files, we used the below web utility:

http://www.ofoct.com/merge-text-files-online

DESIGN ISSUES:

While examining the input files, we noticed that a large chunk of the data was corrupted. Hence, we wanted to come up with a mapper program that could uniquely parse proper English words from the text. The idea was to exclude all special characters from the large input text files and then get a unique string token.

For example - 'Tuesday,' ; 'Tuesday.' ; 'Tuesday' should be treated as Tuesday and not as three separate words.
So we decided on using Java String class' split() using the regular expression '\\W+ ' that matches only to words(containing letter and numbers).

PRACTICAL EXPERIENCES:

We faced some blockers during the execution of this project. Below is a summary:

- Name node did not come up on starting the DFS:

  Frequency of issue - sometimes

  To solve this problem, we had to clear the file system and format the name node before finally starting the dfs again using the below sequence of commands:

  ```
  rm -Rf fs.defaultFS
  hadoop/bin name node -format
  sbin/start-dfs.sh
  ```

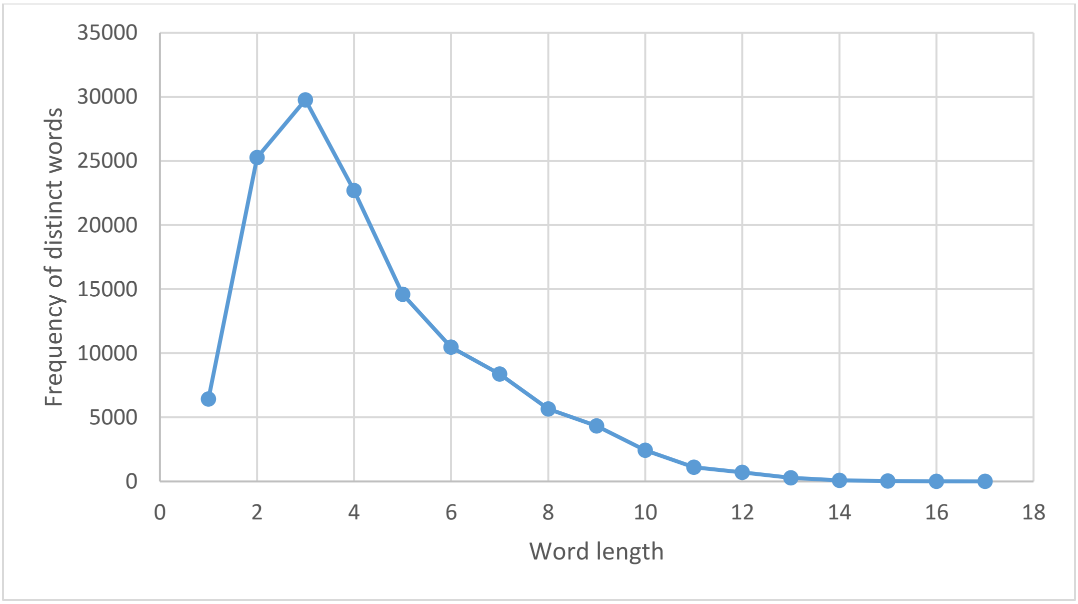
- Data node did not come up on starting the DFS:

  Frequency of issue – sometimes

  To solve this problem we cleaned the cache of our linux VM, then executed the above set of commands again.

EMPIRICAL RESULTS AND DECISIONS:

The given graph shows the relationship between Word Length and Frequency of each word. As per the corpus of data that we took for this experiment, the Range of the output word lengths that we were able to get is 16(max length=17, min length=1). The mode of the reported output is 3 letter words with a total frequency of 29778. The relationship between the Frequency of each word and the Word Length shows a positive linear association till the mode, after which the frequency shows an exponential decline.



OBSERVATIONS:

(A) How many Map and Reduce tasks did running Word Count on the newspaper page produce?

As per the Official Hadoop Documentation:
https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

Number of Map Tasks = Total Number of blocks of the input size

Number of Reduce Tasks = `0.95` or `1.75` multiplied by (*<no. of nodes>* * *<no. of maximum containers per node>*)

In hdfs-default.xml,

Minimum Block Size = 1048576 Bytes ~~ 1MB

And, size of one newspaper article < 1MB (~16-30KB).

Therefore, number of map tasks = 1

And number of reduce tasks = 1

(B) Suppose you took two newspaper pages and concatenated the text. Run MapReduce again on this new text. How many mappers and reducers are now used?

The answer will be same as (A) as long as the block size remains less than 1MB. And concatenating our input files doesn't exceed this limit.

(C) What is the link between the input size, the number of Map tasks and the size of a block on HDFS?

Number of Map Tasks = Total Number of blocks of the input size = Input Size/Block Size

(D) Can you edit the MapReduce program to display the total execution time of the job?

We have modified our program’s main function to find the total amount of time that the program takes to execute. However, the total execution time of the job is subject to core performance of the node in the Hadoop setup.