1. Copy Jar 1 & Jar 2 to Hadoop Cluster Hosted on AWS

scp -i ~/Downloads/emrkeypair.pem -r JAR2 hadoop@ec2-13-126-180-80.ap-south-1.compute.amazonaws.com:

scp -i ~/Downloads/emrkeypair.pem -r JAR1 hadoop@ec2-13-126-180-80.ap-south-1.compute.amazonaws.com:

1. Get Wikipedia documents stored at AWS S3 Bucket.

wget 'https://s3.amazonaws.com/aws-logs-642790757236-us-east-1/data/WikiFull.txt'

hdfs dfs -put WikiFull.txt

1. hdfs dfs -put stopwords.txt

4.1 RUN JAR1 by following command

time hadoop jar wc.jar WordCount WikiFull.txt output1 stopwords.txt

(Java Code for map.py and reducer.py)

4.2 RUN JAR2 with map and map3

time hadoop jar hadoop-streaming-3.1.1.jar -file map3.py -mapper map3.py -file reduce2.py -reducer reduce2.py -input WikiFull.txt -output output2

* + 1. word count map.py: This is the mapper function implemented as a Python script. It reads in a stream of key, value pairs from stdin and emits another set of key/value pairs where the key is the word and the value is the number of occurrences. Modify this mapper to discard useless words, such as articles, useless verbs(am, is, are), and prepositions.

stopwords.txt contains a list of such useless words,one in each line. The modified mapper should filter out these useless words and then emit the key value pair.

4.2.2 map3.py includes the combiner. Instead of emitting the word frequency to be 1, perform a local frequency counting and then emit the partial frequency. Rename this mapper file to be mapcombiner.py

4.2.3. word count reduce.py and reduce2.py: These are two versions of the reducer function, which reads in the stream of key/value pairs and aggregates the count for each word.

**Change no. of mappers in one cluster**

1. nano /etc/hadoop/conf/mapred-site.xml
2. Change following parameters

mapred.tasktracker.map.tasks.maximum

Mapred.tasktracker.reduce.tasks.maximum