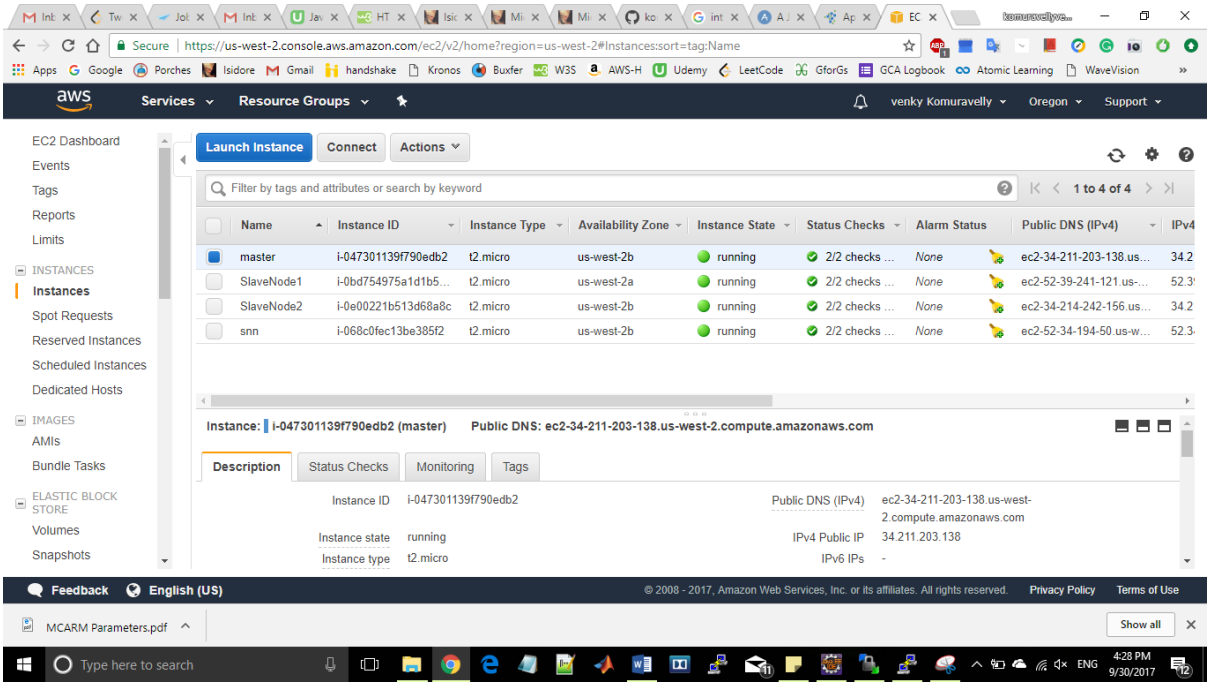
Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060

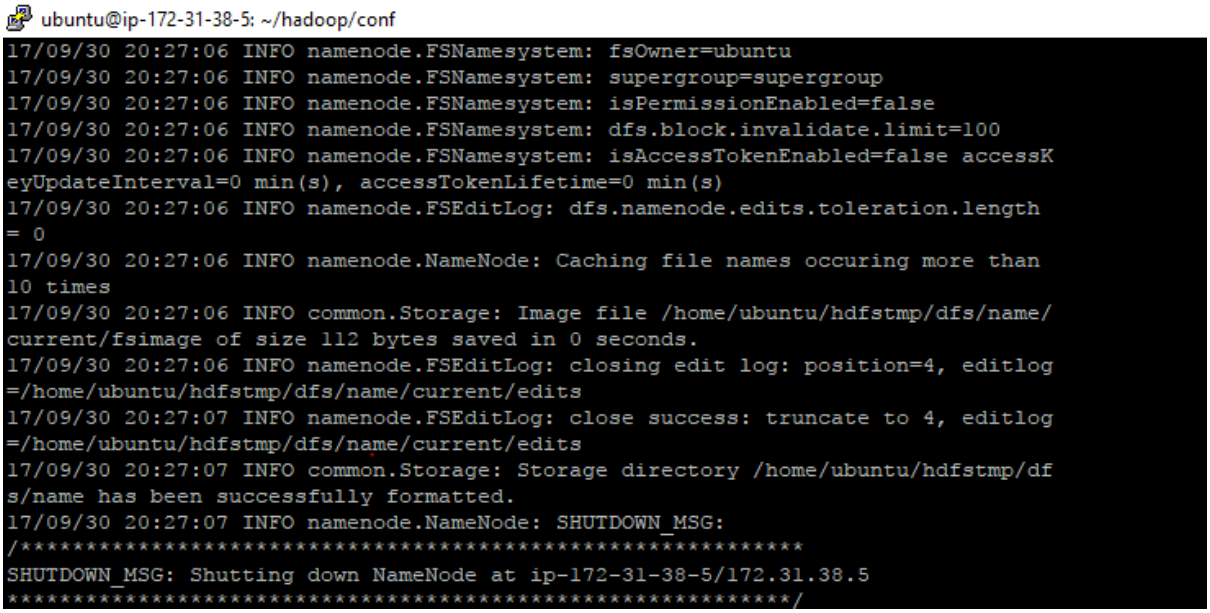
Assignment-2 In-Map Combining Methods

For the following assignment I used Stock Overflow and Google as resources.

Started the AWS-Instances: -

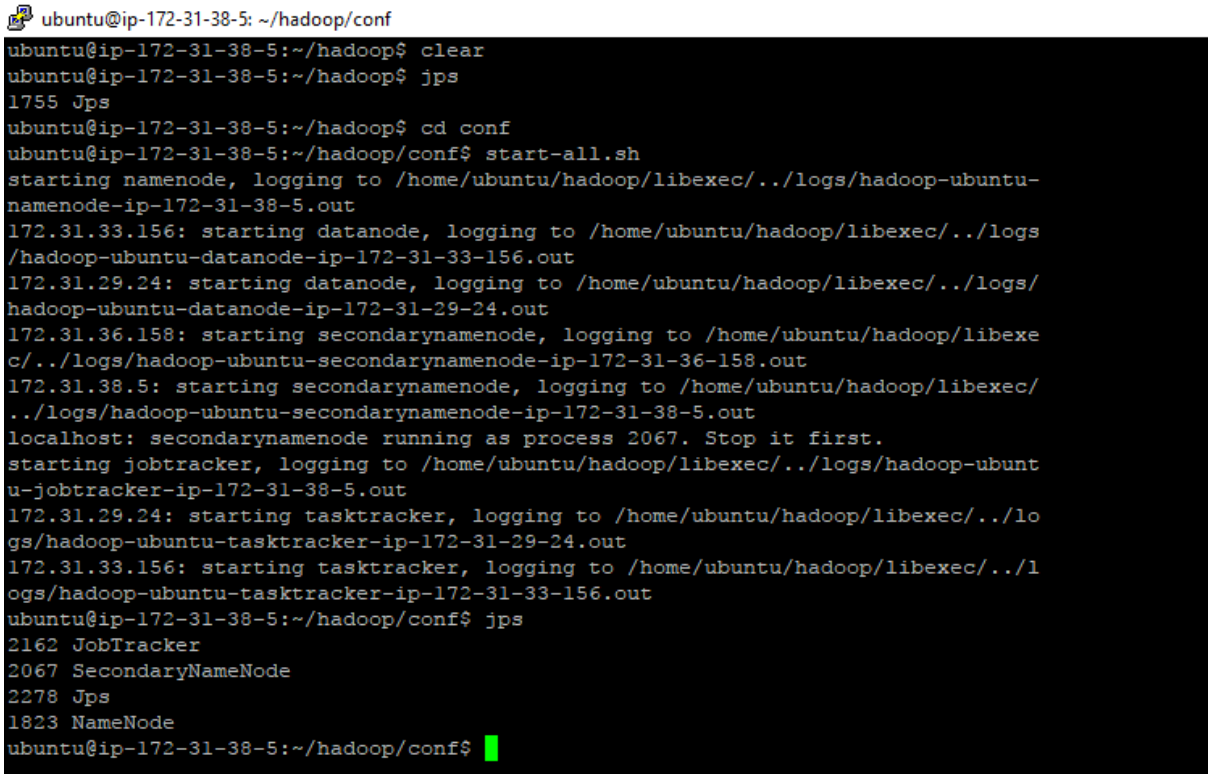


Formatted the Name Node space:-

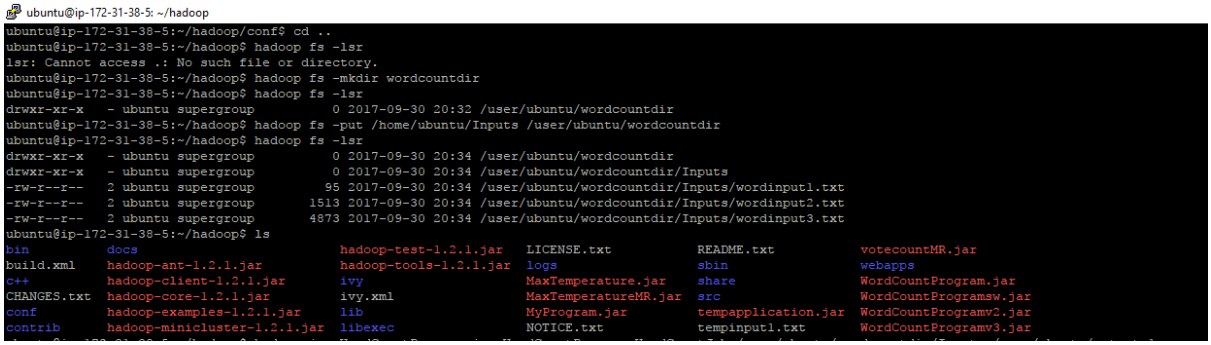


Now, Started the Hadoop by using start-all.sh command as shown below. By giving the JPS command we can confirm that Name node, Secondary Name node ,Job tracker has started on Master Instance.

Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060



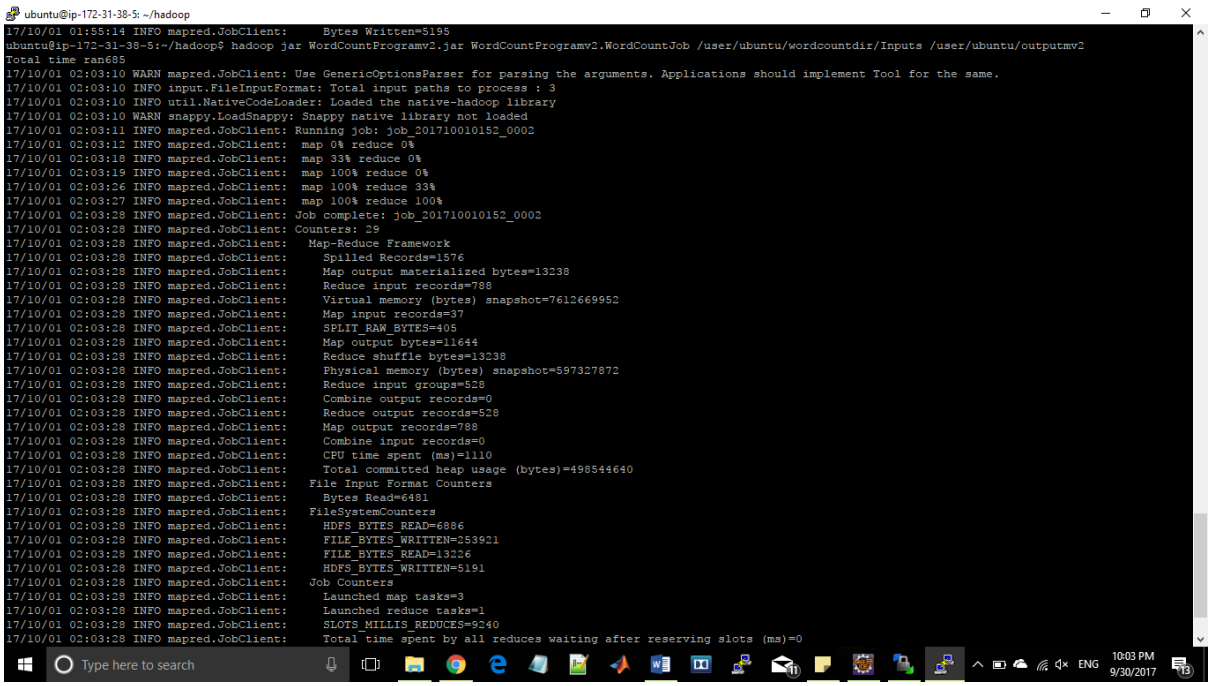
Created wordcountdir in HDFS system using *hadoop fs -mkdir wordcountdir* Then copied Input Folder from Ubuntu File System to HDFS created folder. This, we can see by *hadoop fs -lsr* . Jar files we can see by ls command in hadoop folder.

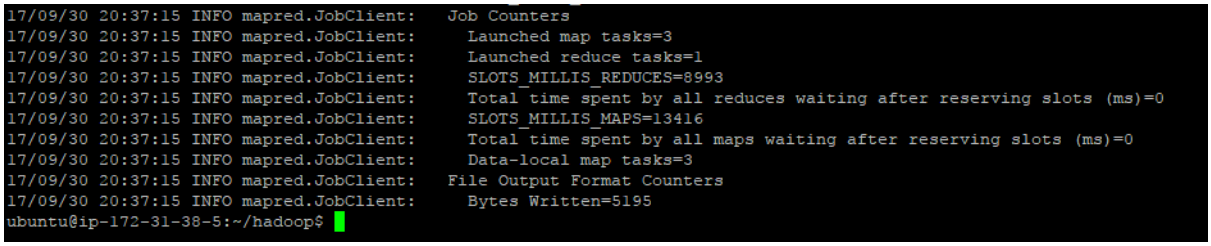


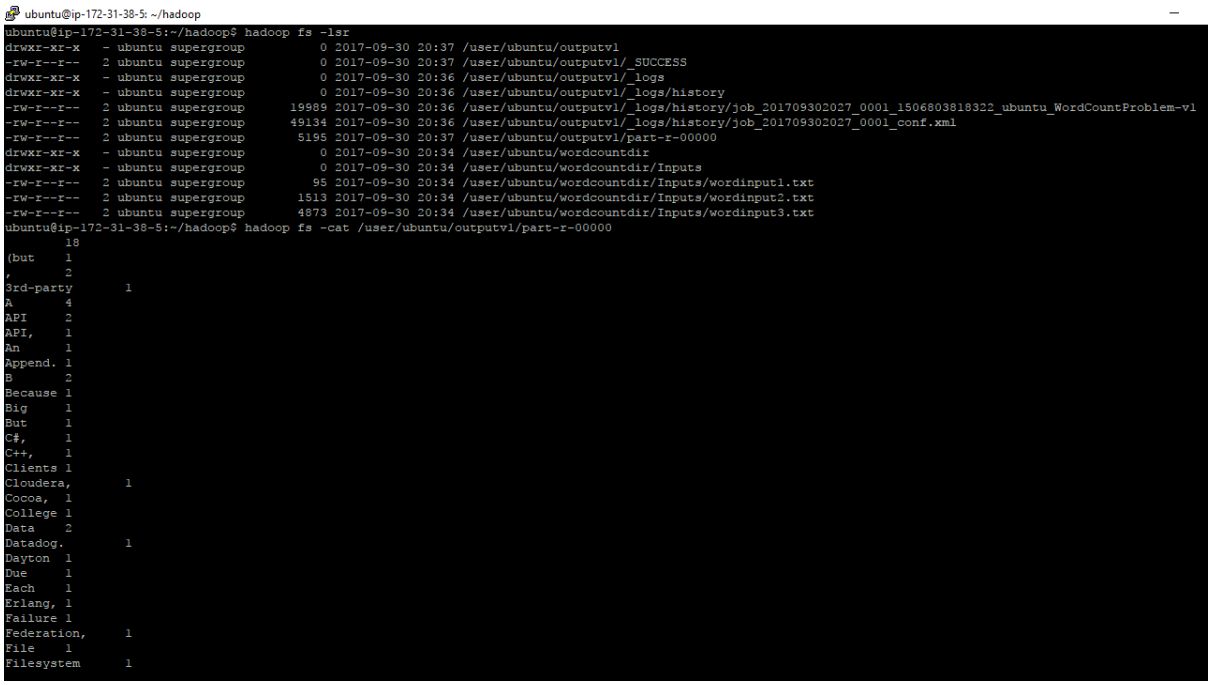
Here we ran version1 of the system by giving the Folder containing the input files as input and storing the output in outputv1.

Note:: Comparison between Version1,2,3 is shown in the last slide.

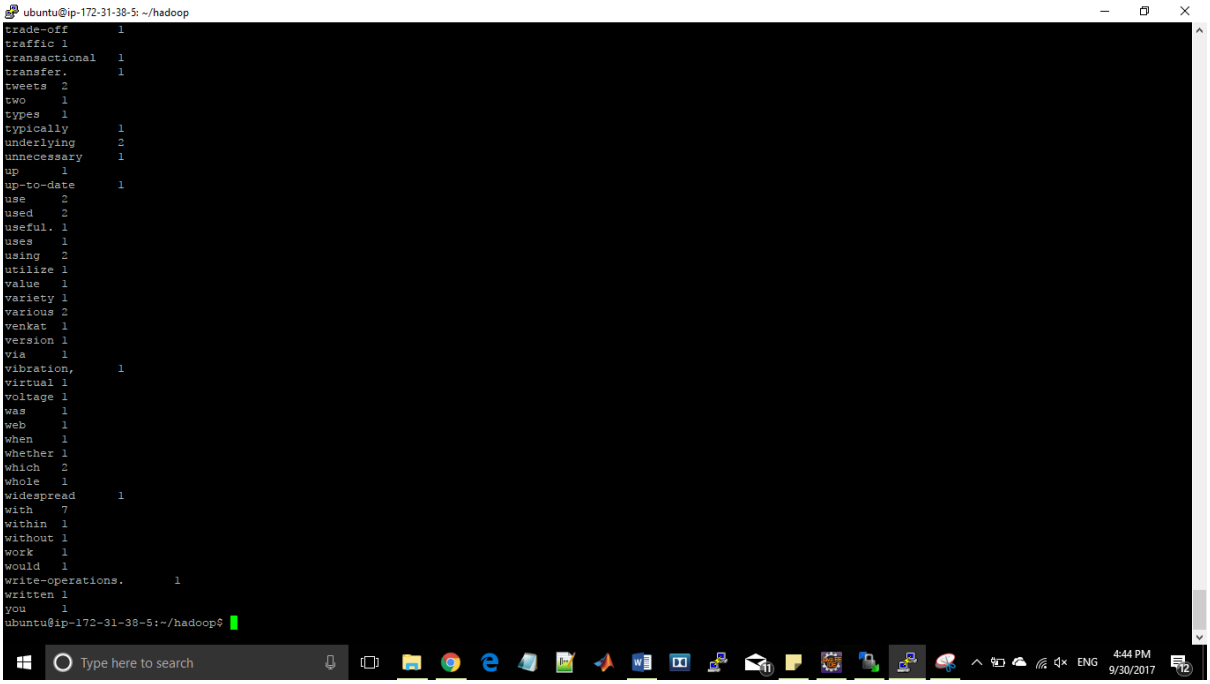
Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060



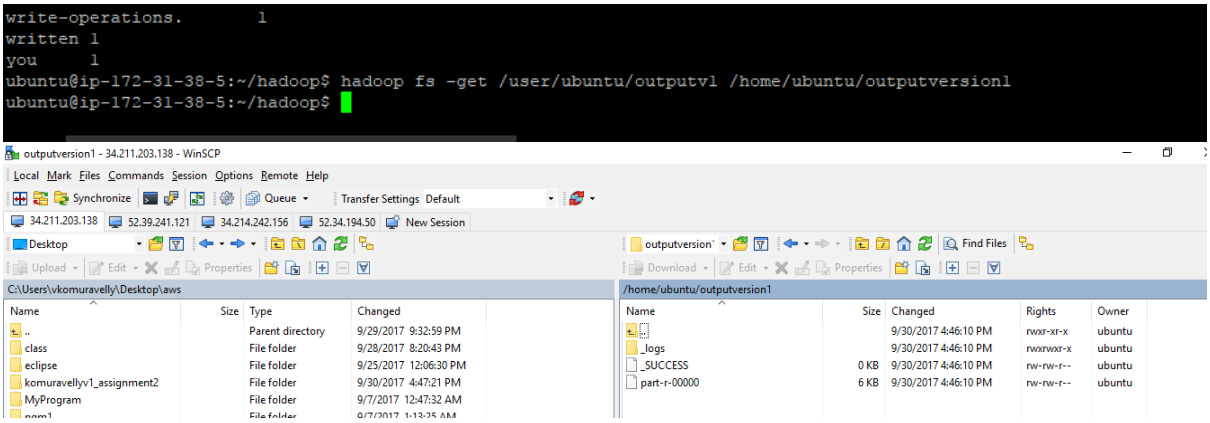




Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060



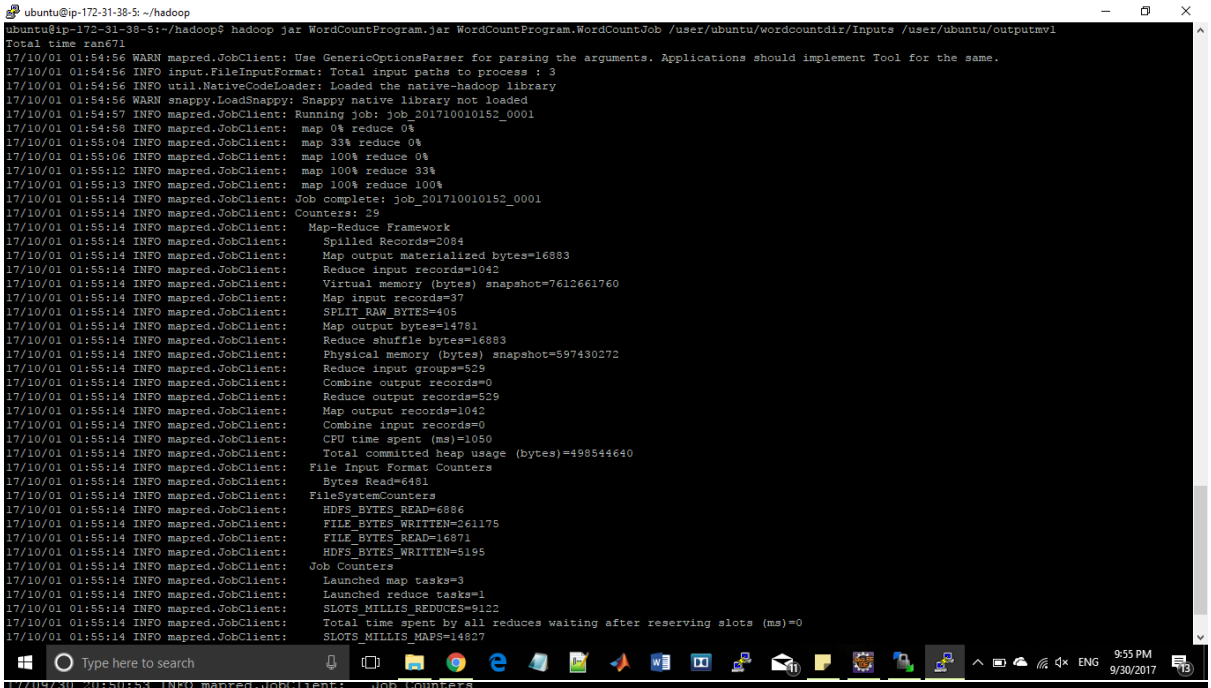
Copied the output file created to Ubuntu system by using get command.

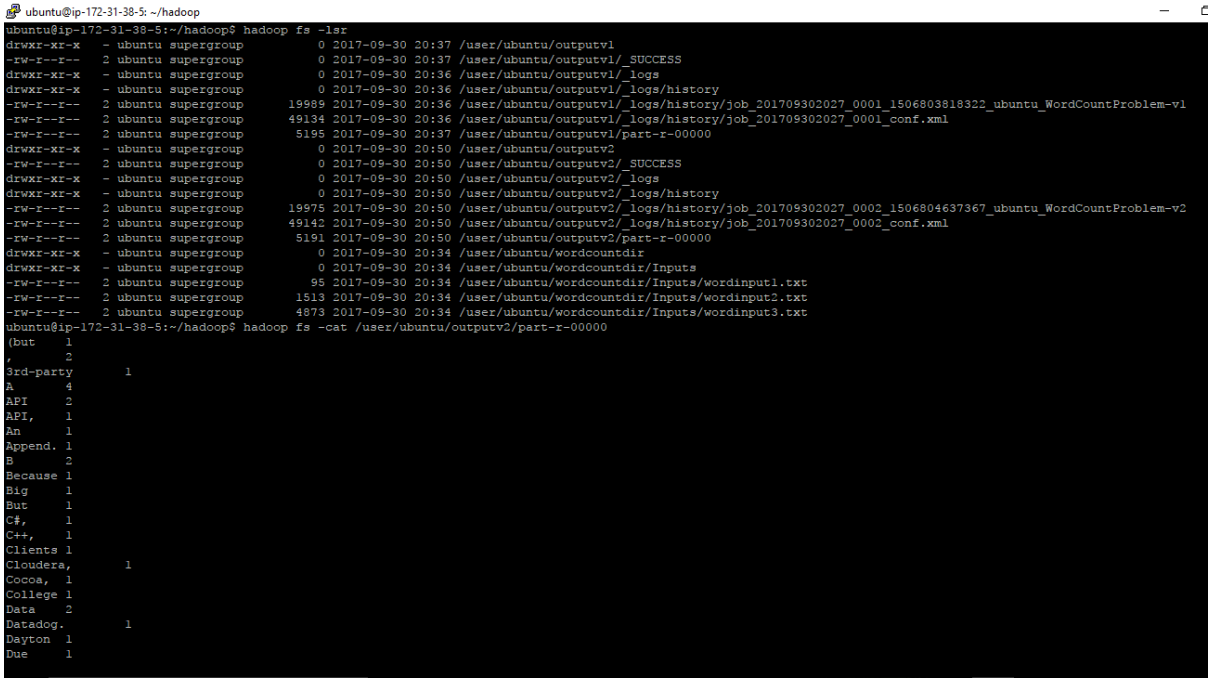


Version-2

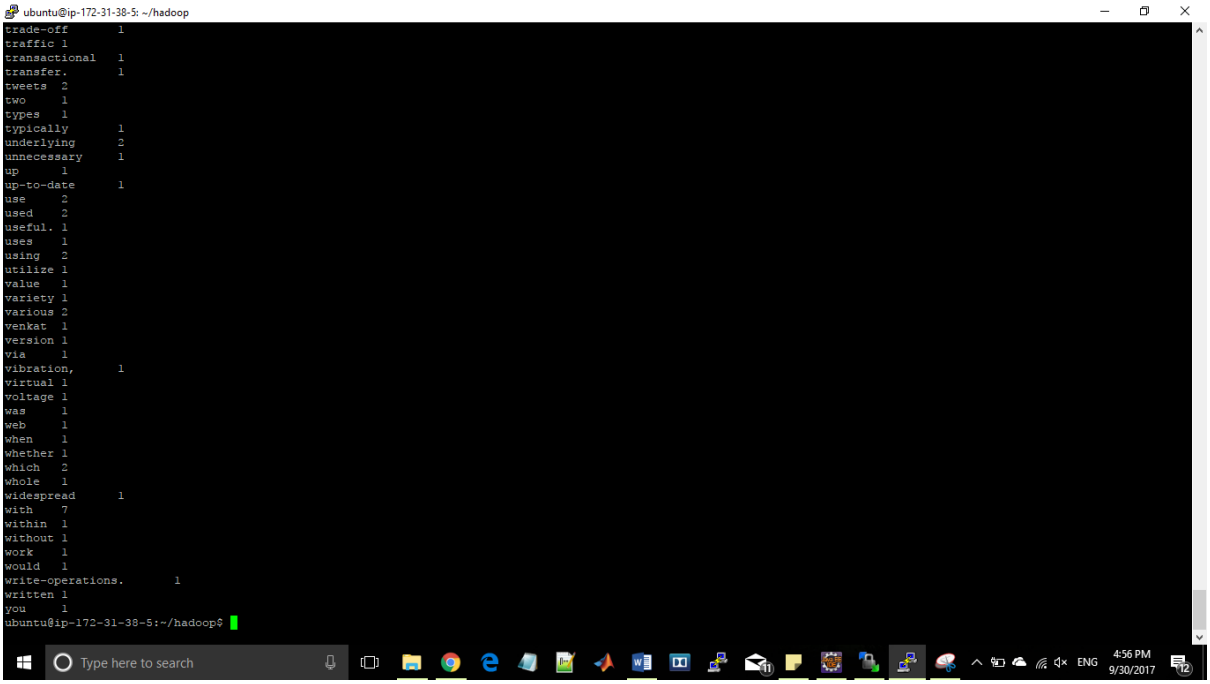
Like same as version 1, Version 2 is ran.But in version 2 I used HashMap to store the map output instead of sending the map output directly to reducer. By using HashMap method I made the Key as unique so that key-value pairs are reduced. Adding the value for the same key makes few number of key-value pairs.

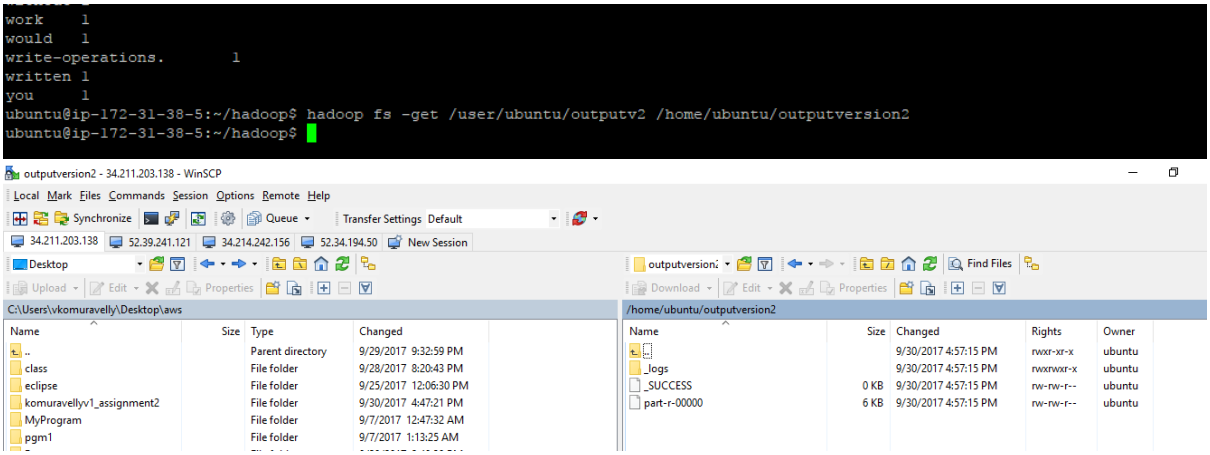
Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060





Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060

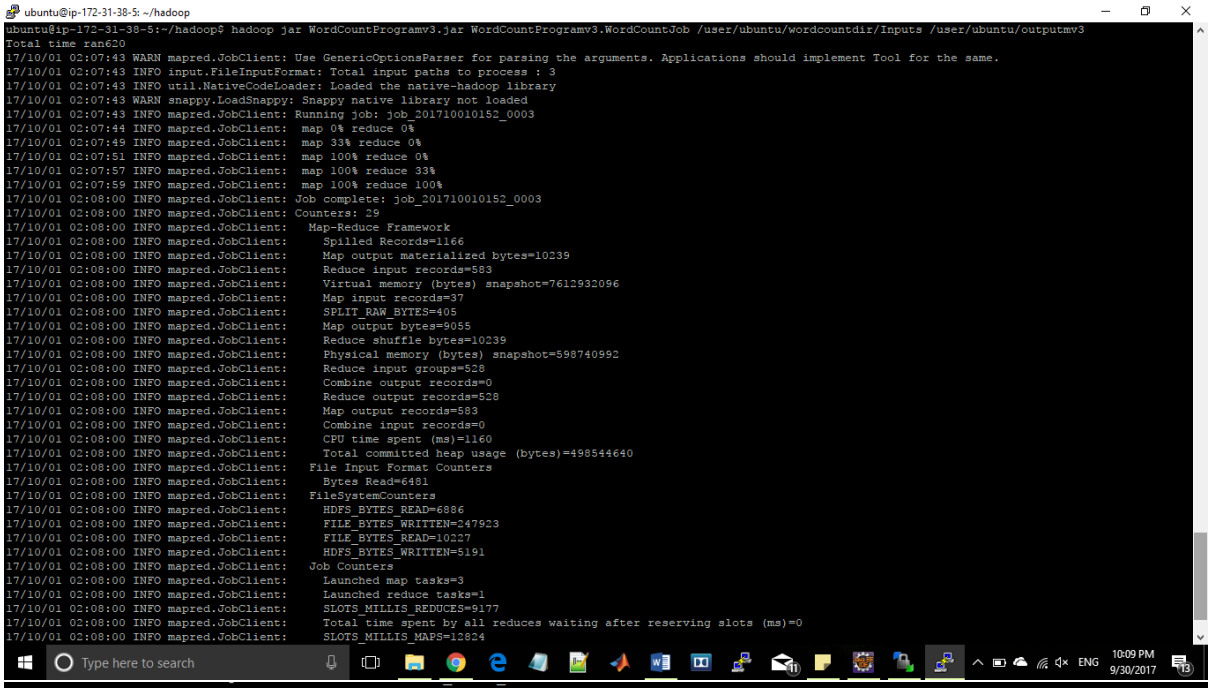


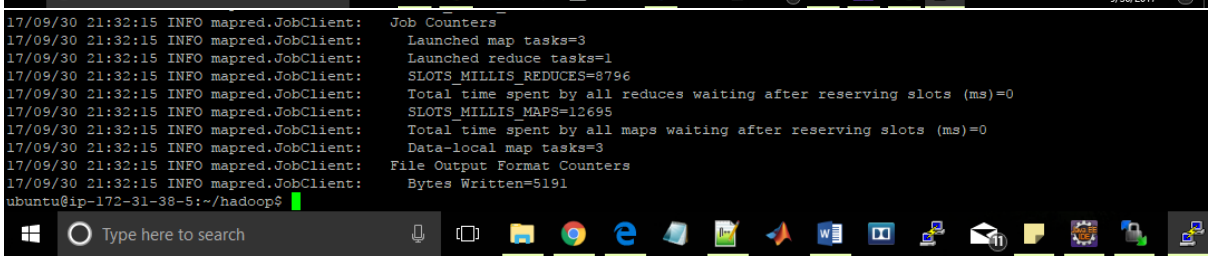


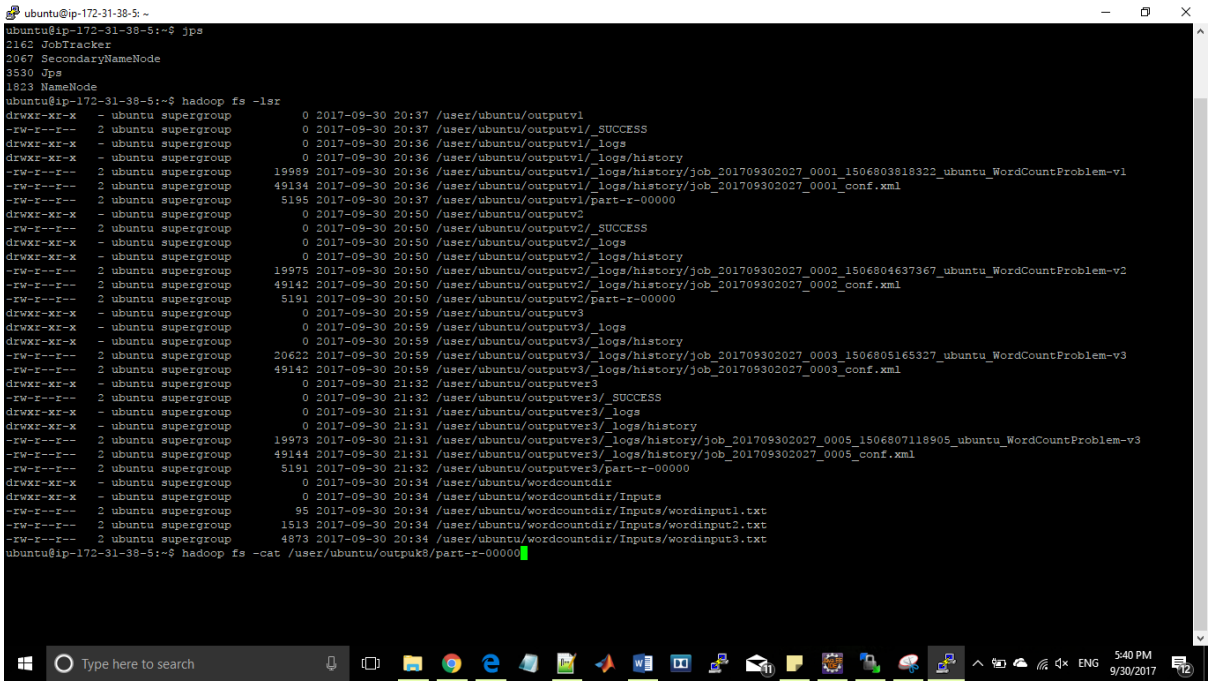
Version-3

In Version 3 , Using CleanUp Method, result of reduced key-value pairs from map (used HashMap method) are stored in MemoryBuffer. Same mapper is used for the next input chunk. In this way In total there will be few key-value pairs for the reducer to process.

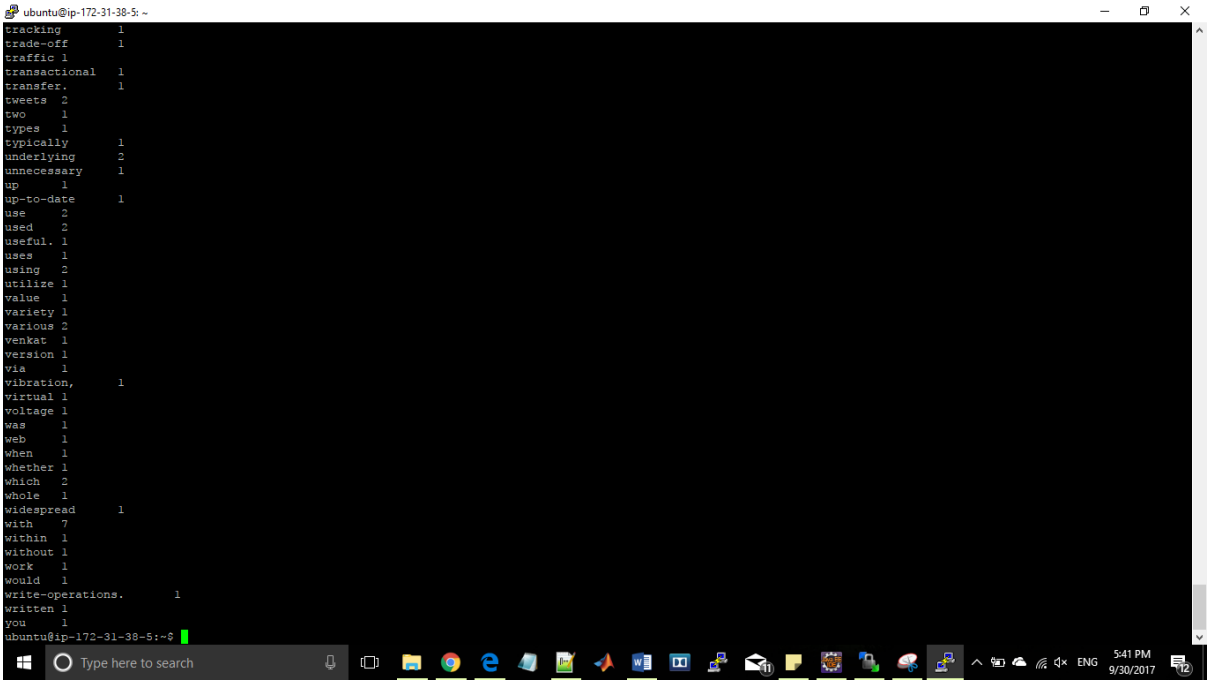
Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060





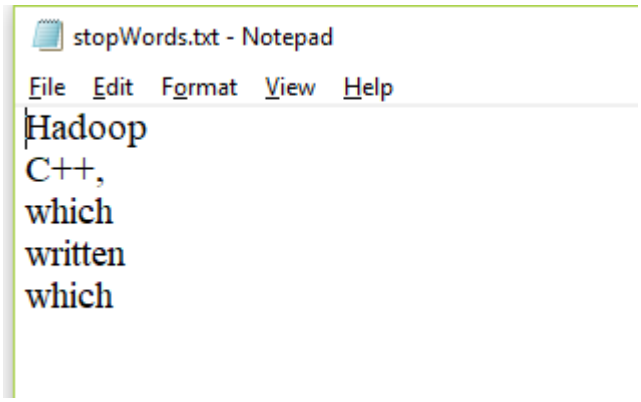


Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060

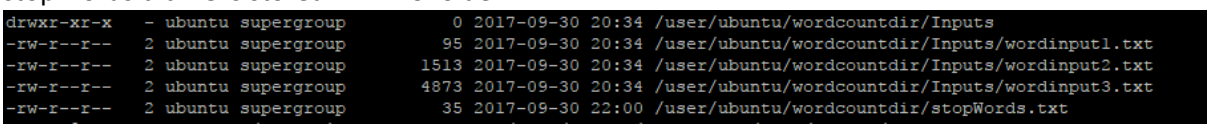


StopWords:-
From the output of version 2 , I used below keywords as stopwords and processed using distributed file cache API to remove these words from the output.

Reference: https://examples.javacodegeeks.com/enterprise-java/apache-hadoop/apache-hadoop-distributed-cache-example/

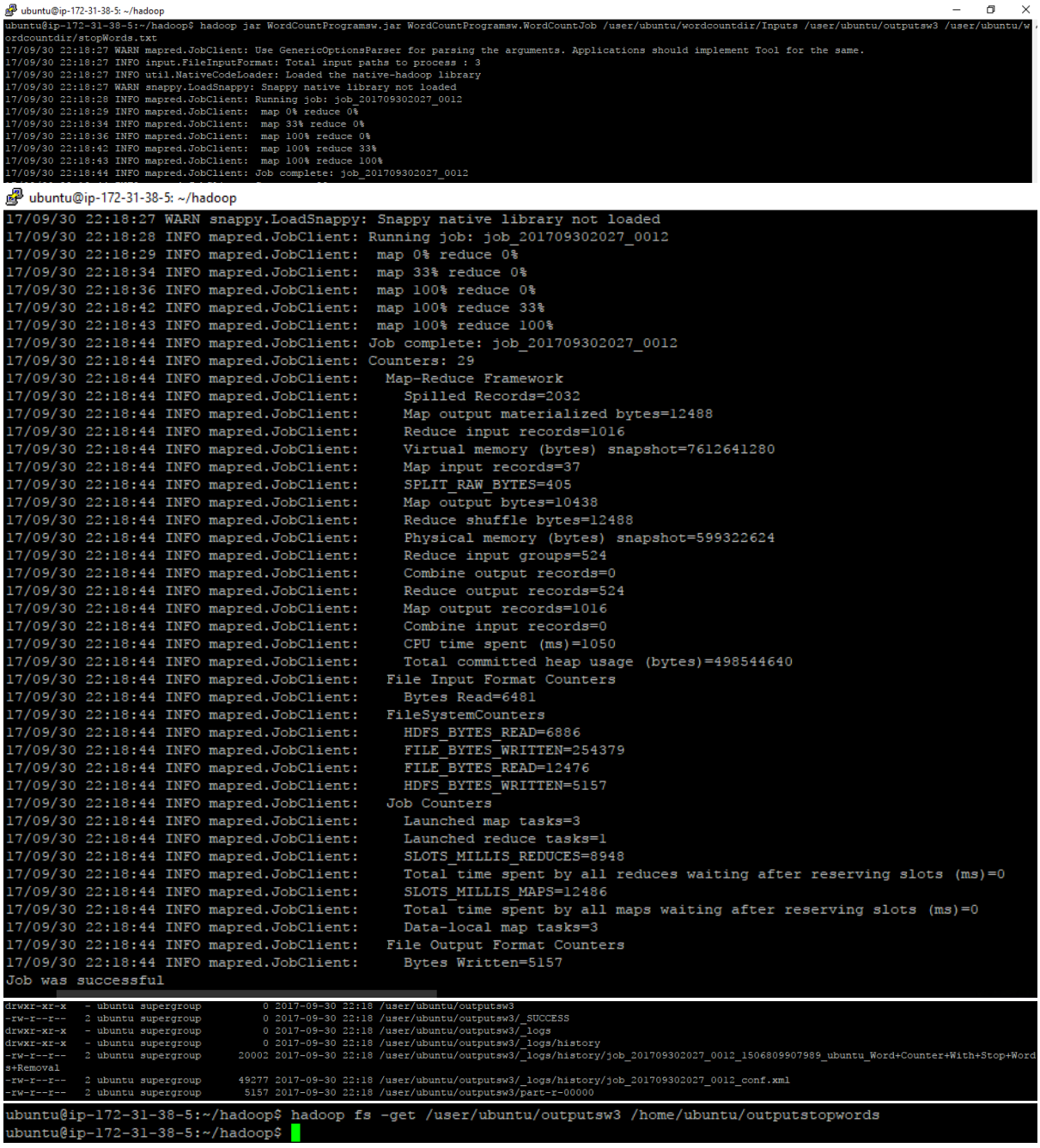


stopWords.txt file is stored in HDFS folder.



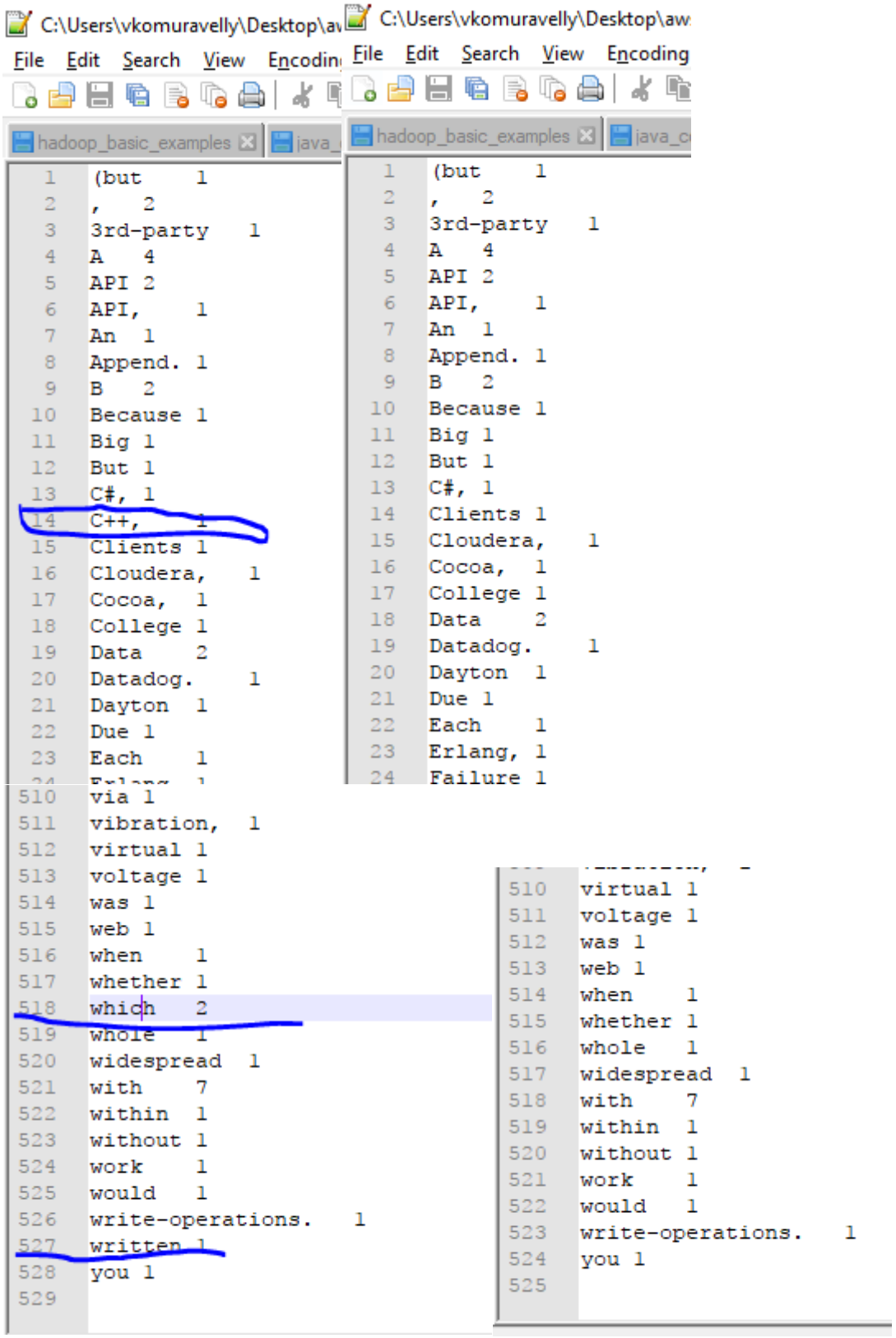
Ran the MapReduce program for stopWords below.

Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060



Here, We can see the words removed from output.

Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060

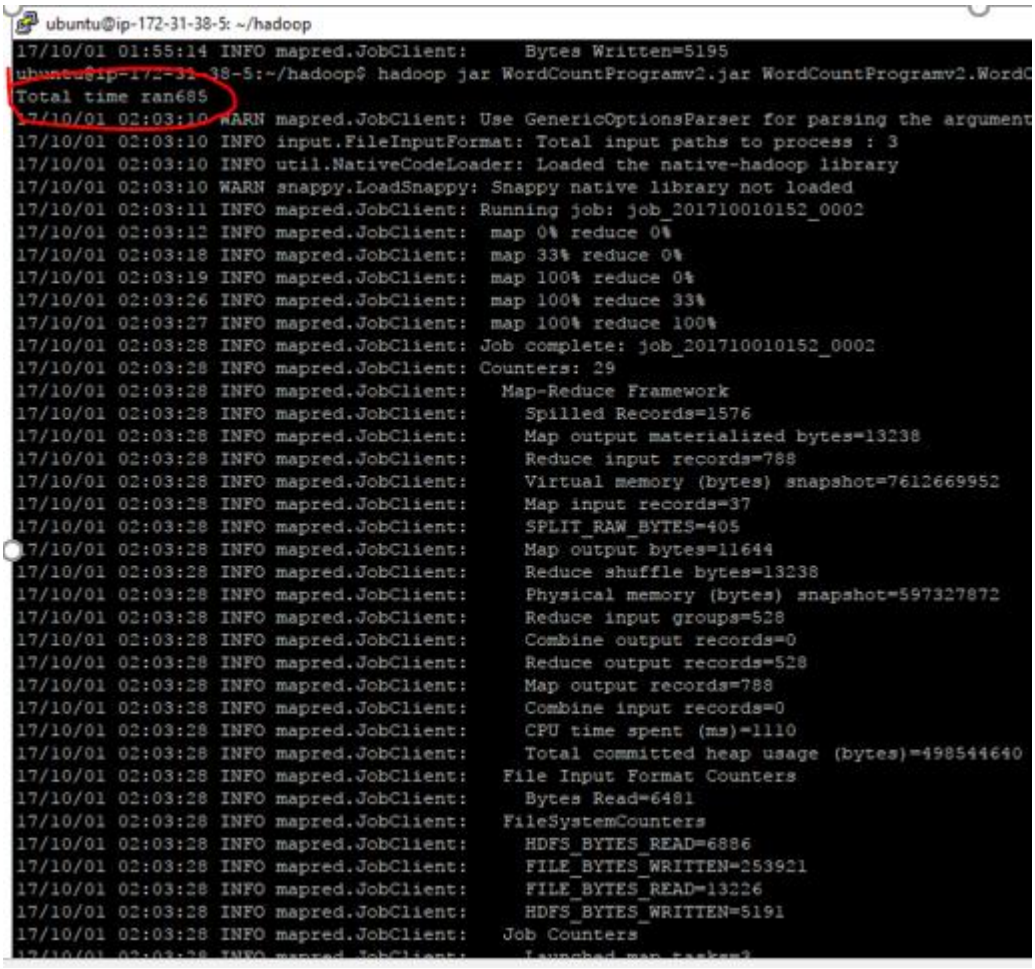


Comparison between v1,v2,v3.

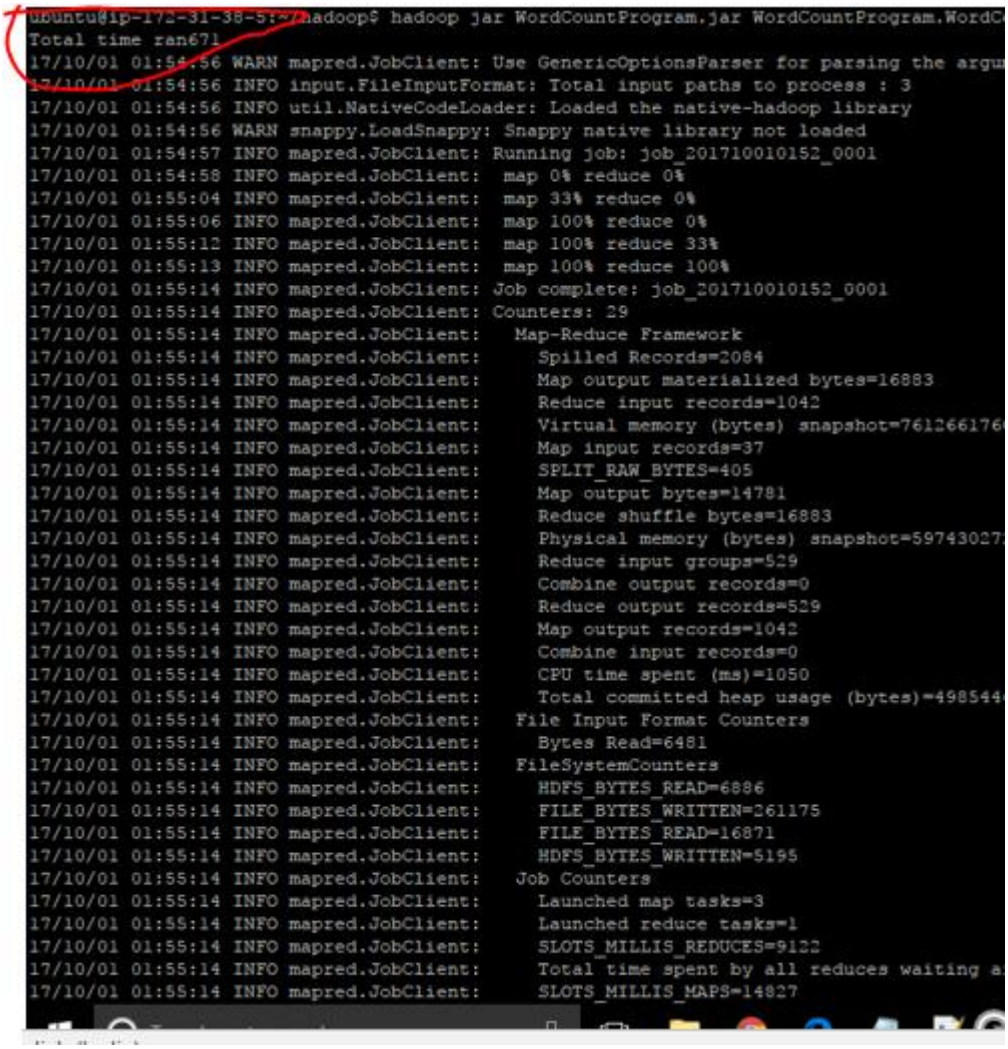
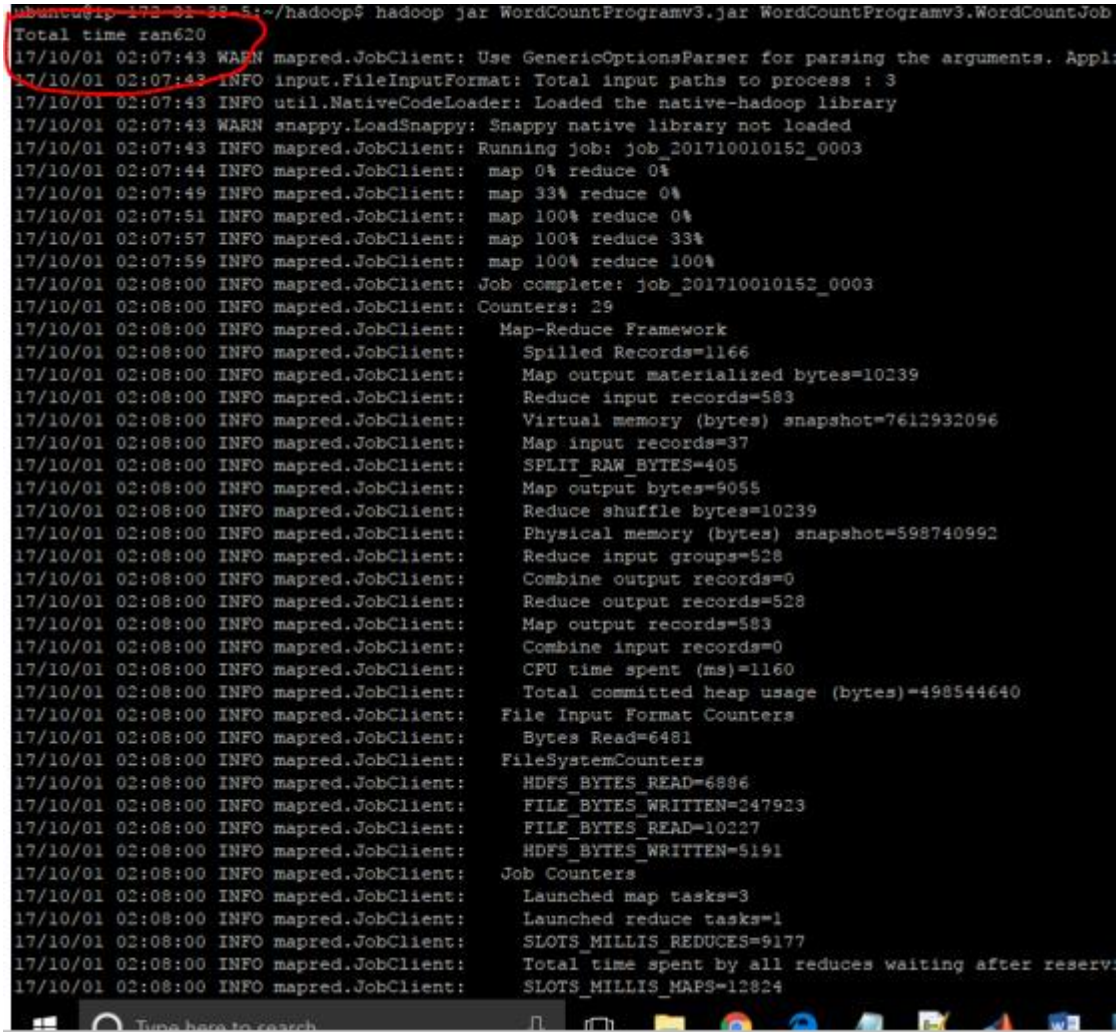
Execution times=

V1→685msec V2→671msec V3→620msec

Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060



Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060



Komuravelly Venkateshwarlu
Komuravellyv1@udayton.edu
1015237060