DataSet1: <http://norvig.com/big.txt>

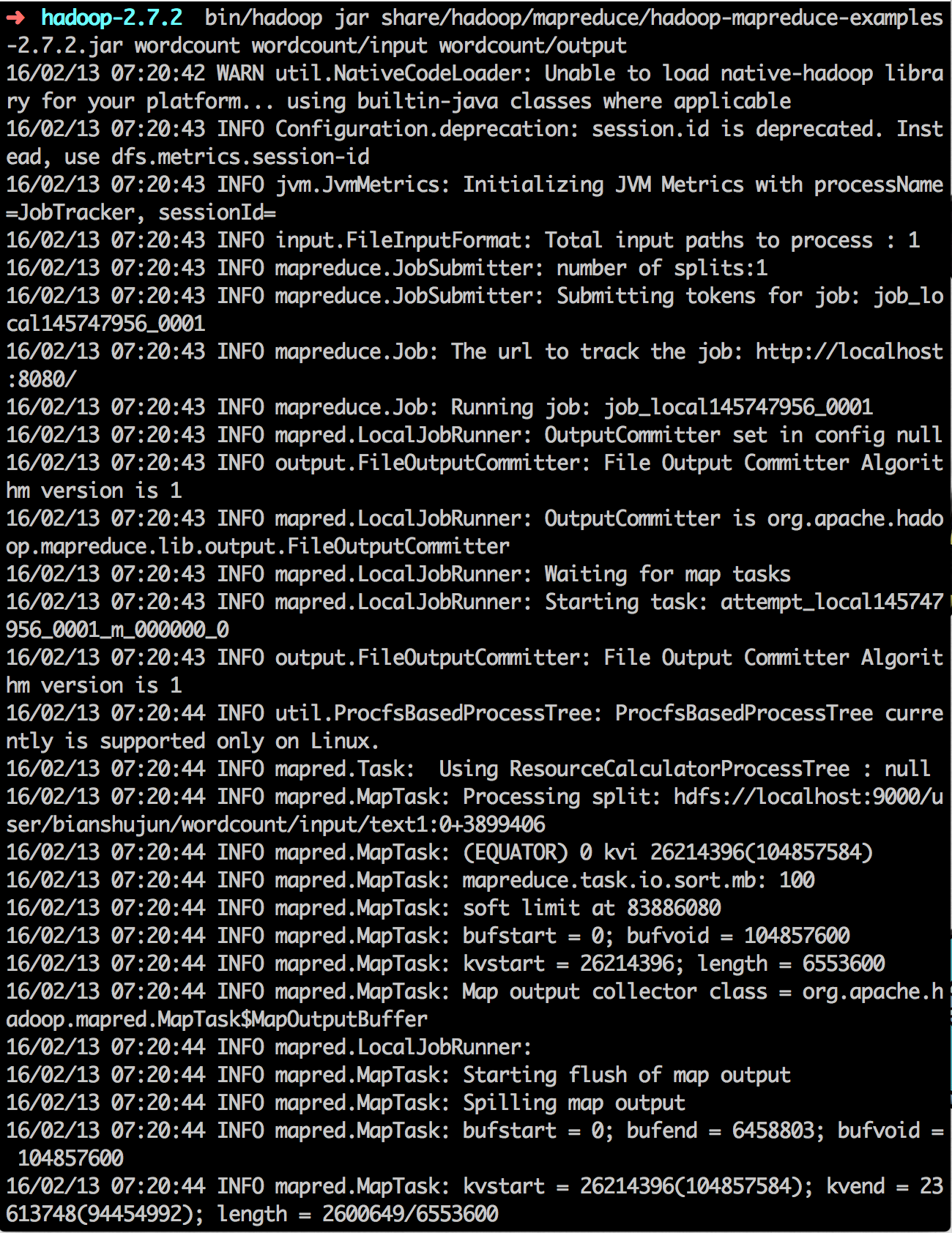
DataSet2: <http://www.math.sjsu.edu/~foster/dictionary.txt>

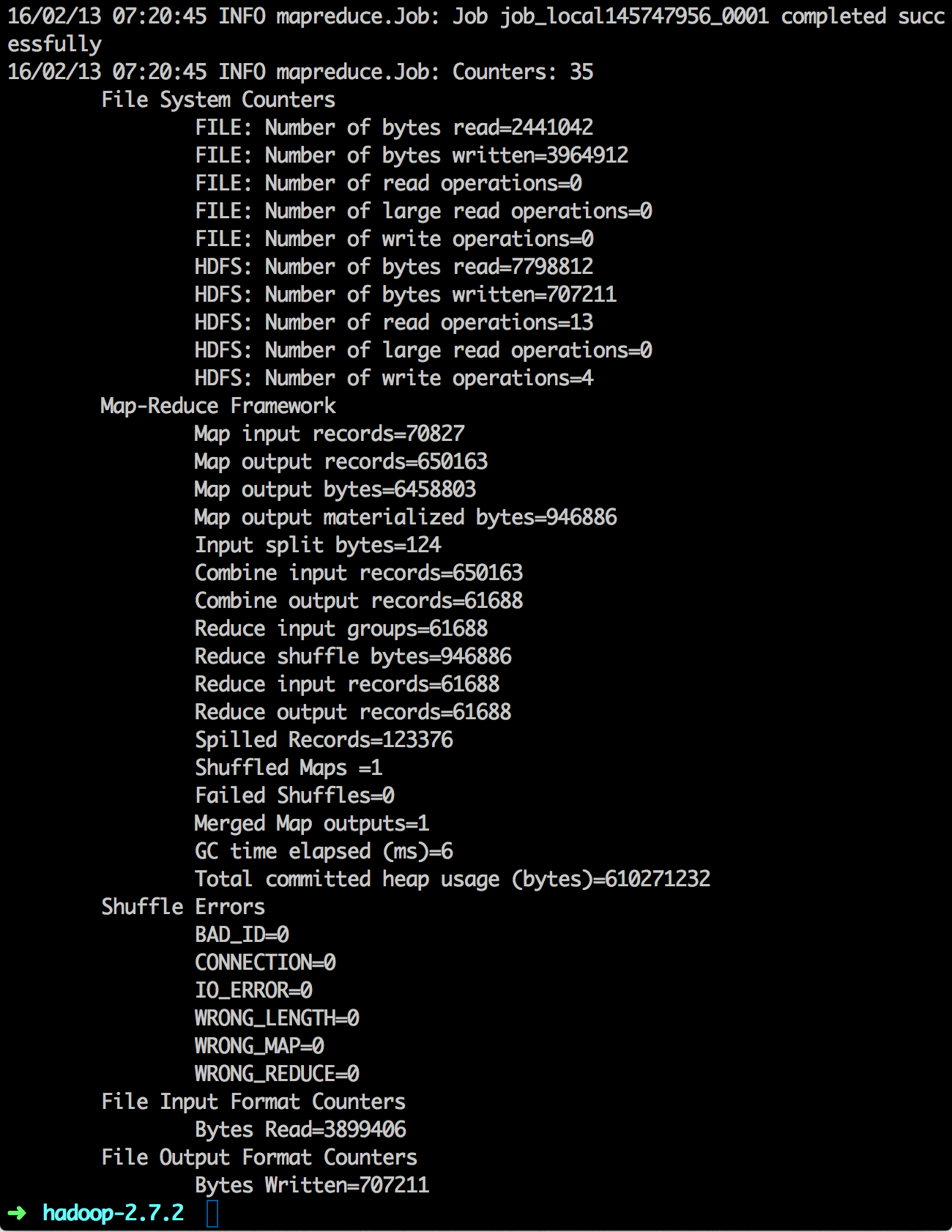
MapReduce Function1: wordcount

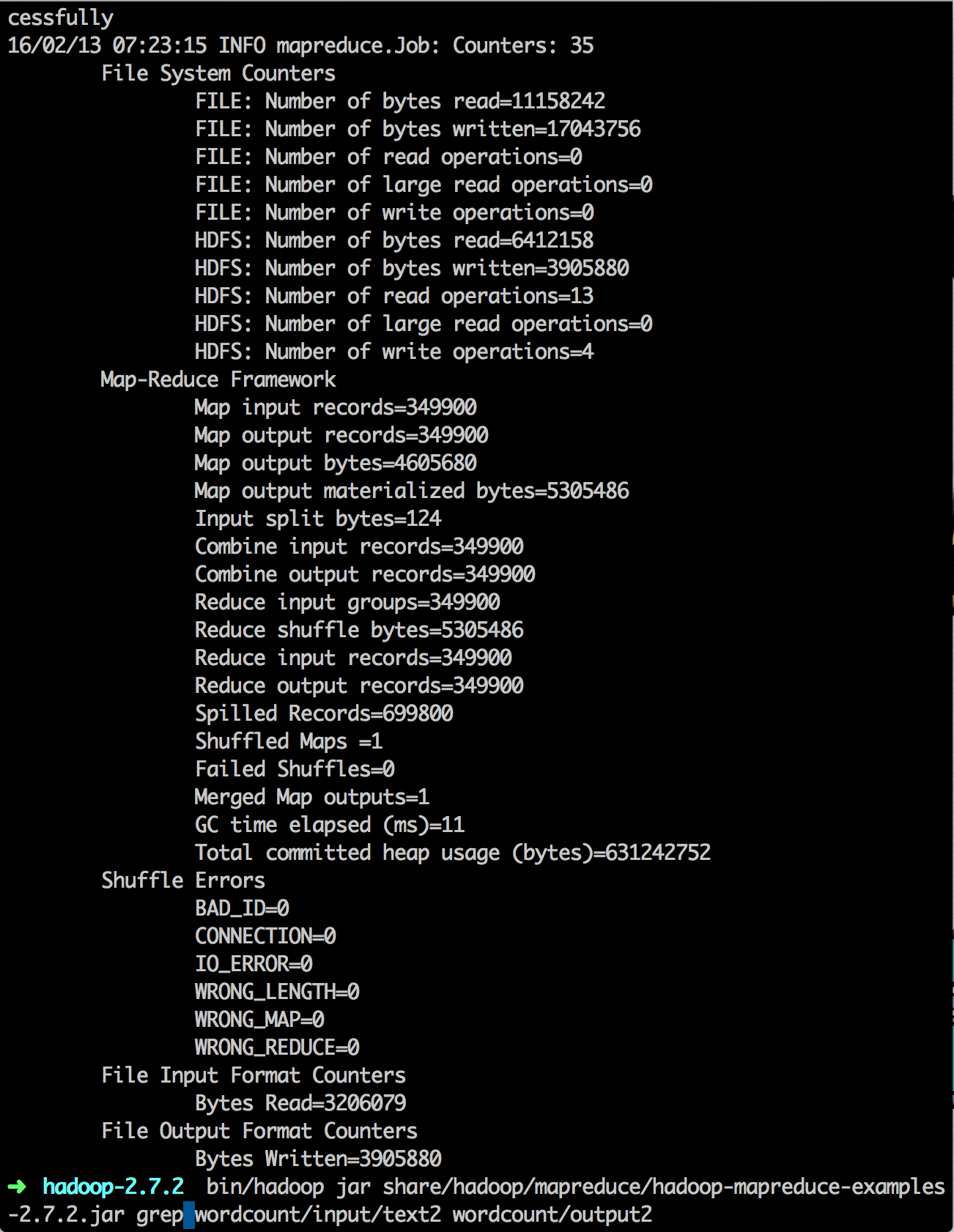
MapReduce Function2: grep

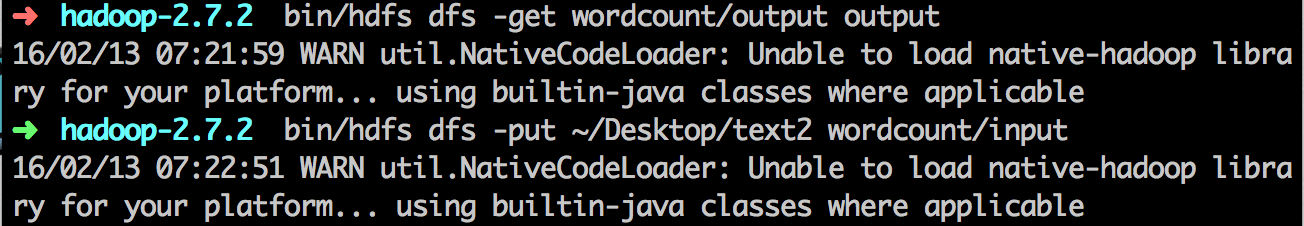
Some screenshots:

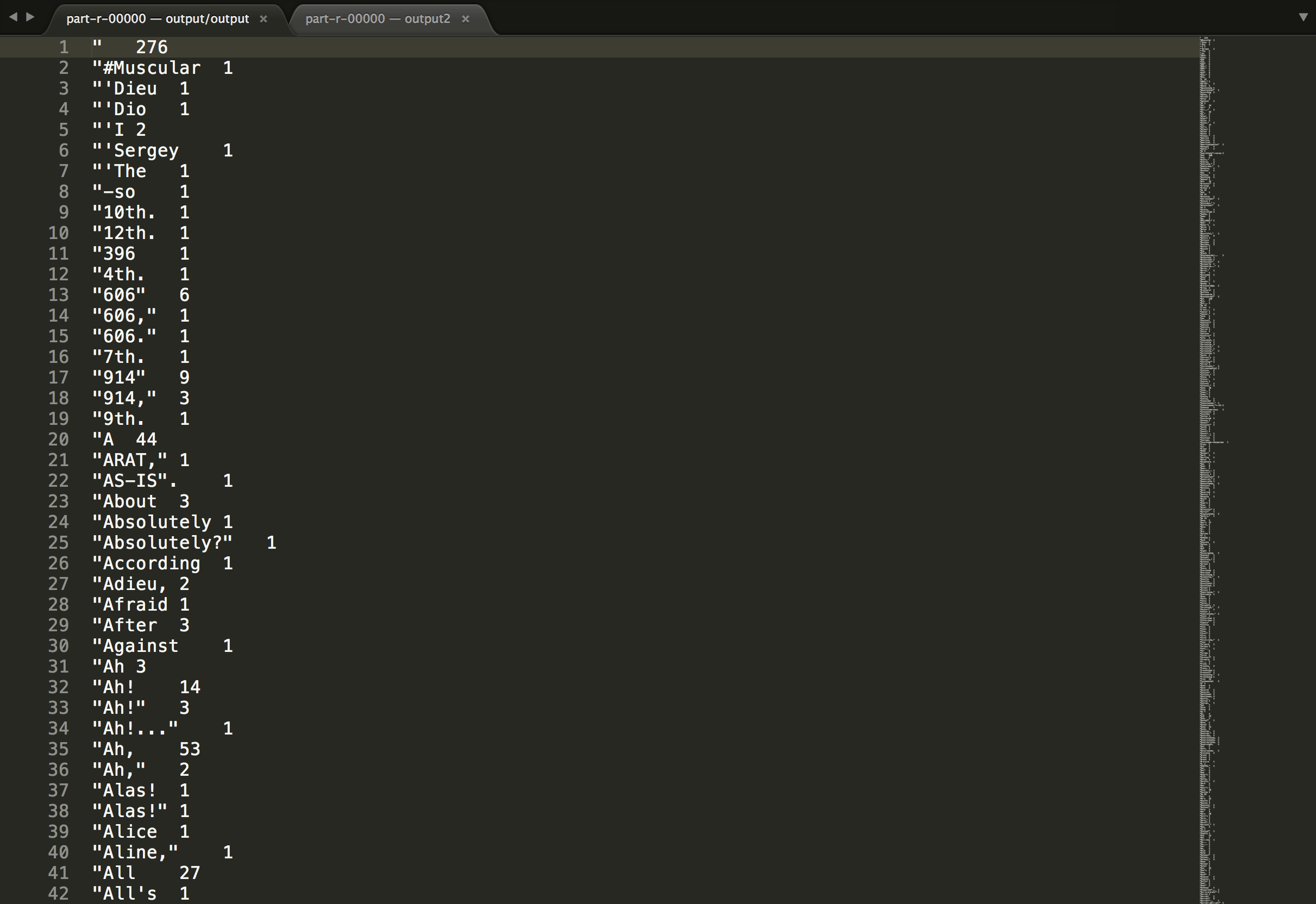
WordCount Part:

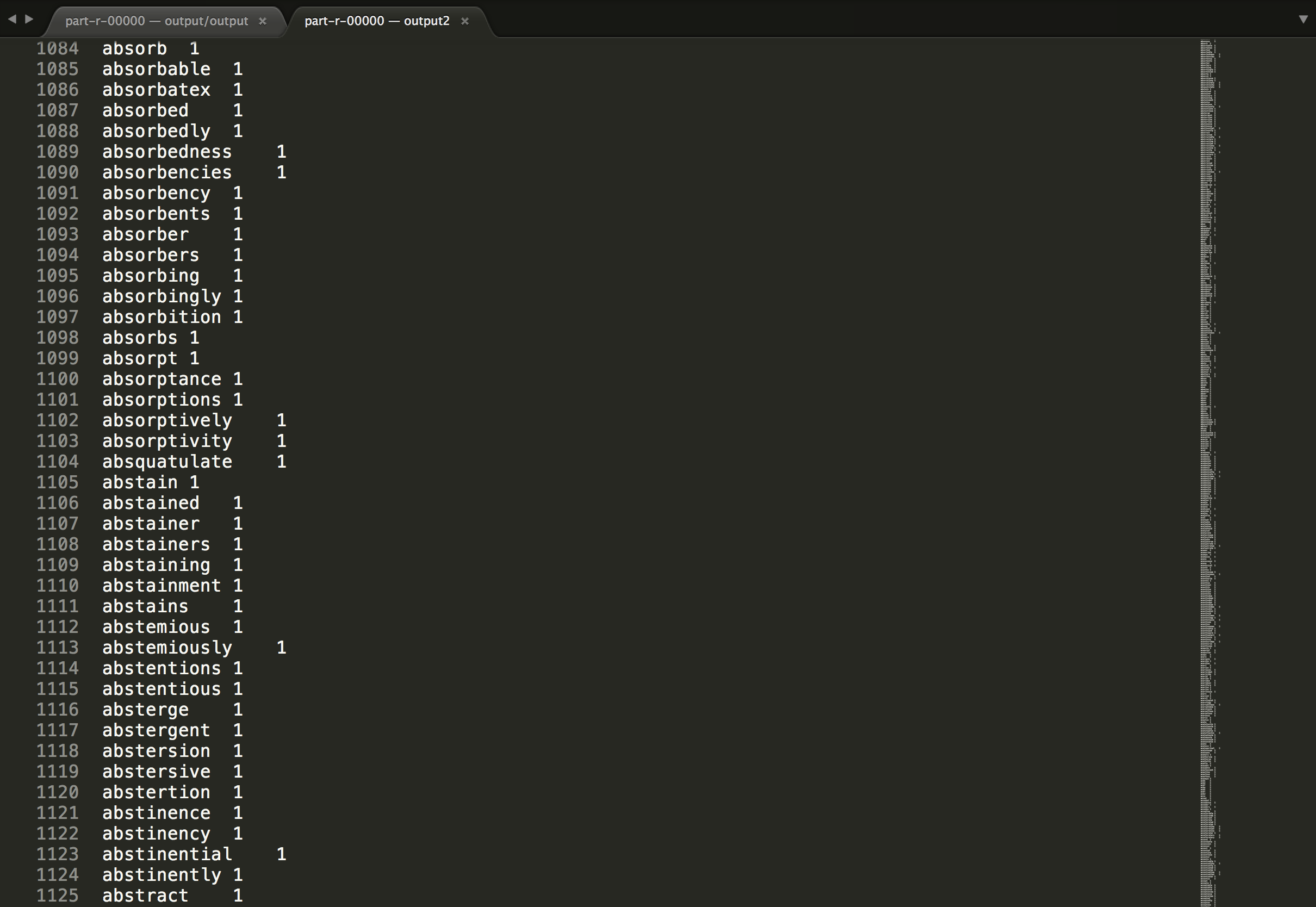




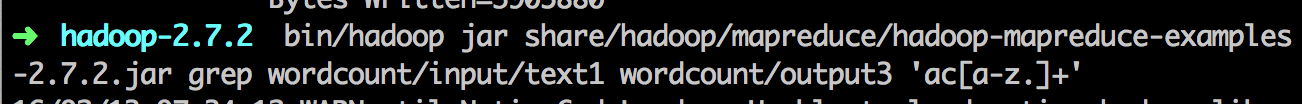


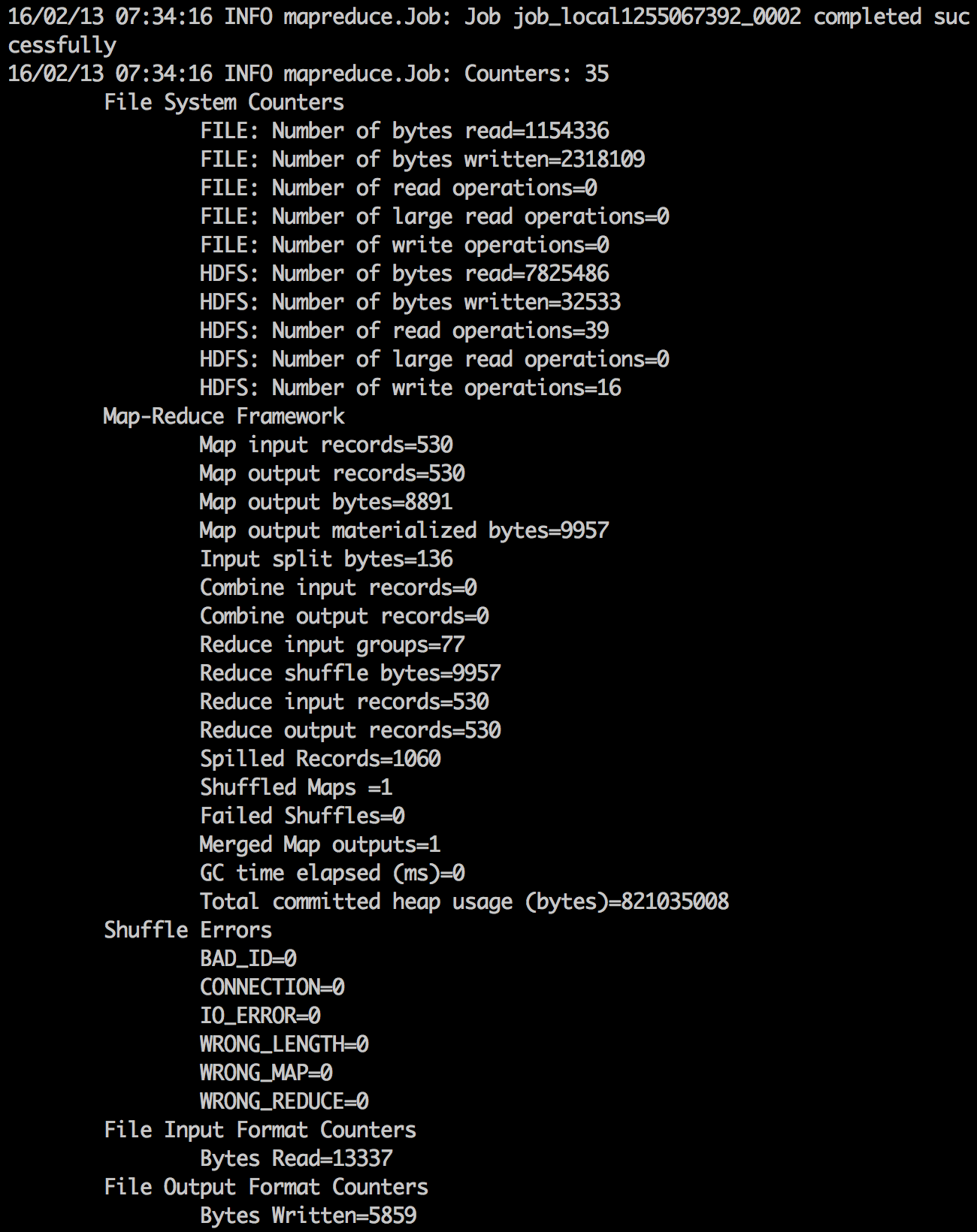


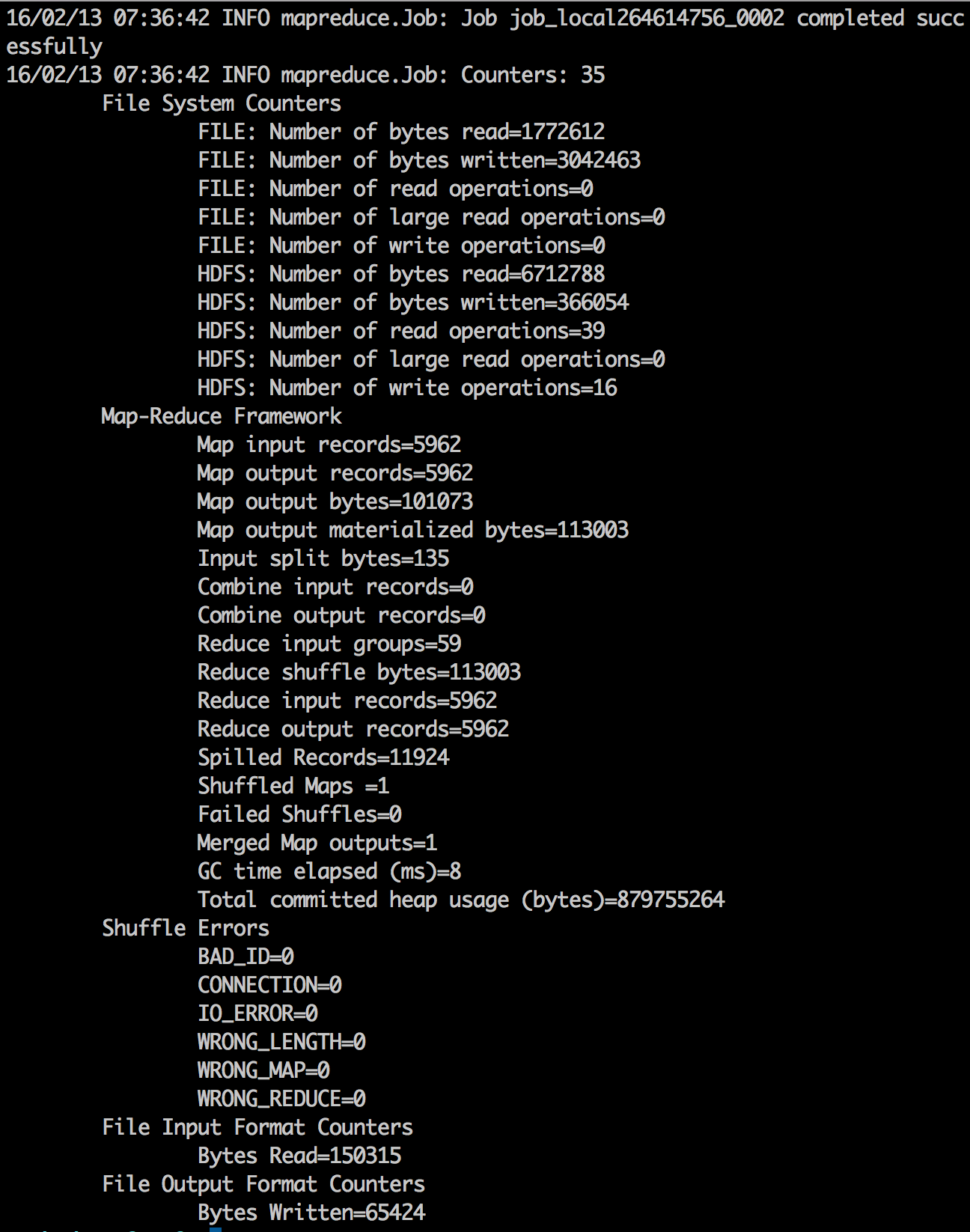




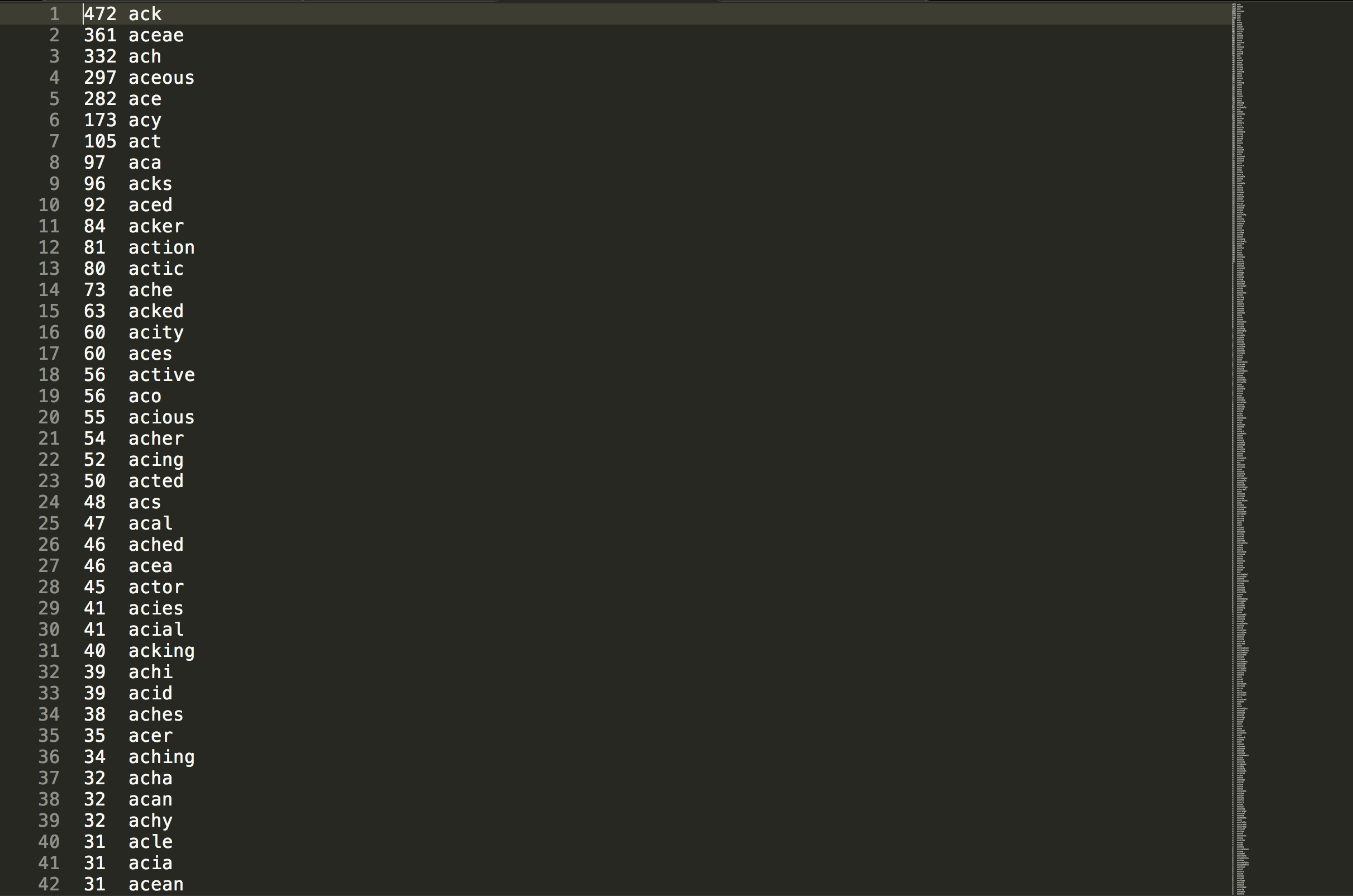
Grep Part:











Analysis and Feeling:

MapReduce is really fast! Doing wordcount and grep cost only few ms! Because I do not have enough storage in my computer, I only tested some MB files which is too small to mapreduce, which also cannot show the real power of MapReduce. But it still shows that MapReduce is one of the most popular programming model for big data analysis in Distributed and Parallel Computing Environment, especially for implementing parallel applications. And the huge size of file could be stored on the HDFS and can be analyzed with MapReduce. HDFS also bother me a lot at first, but finally I found it is quite similar to the normal Linux/Unix File System, which helps me a lot to understand the way HDFS works.

Although few data cannot show a lot of staff, I still can do some simple analysis. I found in the test that the filesize do have effect on the performance that with the filesize increasing the performance decreases. And also for the grep function, how many words are exactly searched also matters. Although I didn’t use multiple nodes in MapReduce, I do believe that as the number of nodes increases the execution time decreases and performance increases.

This is my first time to use mapreduce and run its function. I learned mapreduce function from scratch and to understand every command of mapreduce is not an easy thing. Through this experiment, I am confident that I could understand what is MapReduce and I could write some simple MapReduce code to implement the big data anaylsis!