# Wordcount

Programs that count the one-word and two-word frequency of occurrence of words within an input file as well as the one-word frequency of occurrence of words given by a list in the distributed cache.

# Programs

For all programs, the wordcount, mapper, and reducer class are in a single java file.

## HW1/part1/WordCount.java

This program counts the number of one-word occurrences in the files located in the specified input folder. The mapper maps all these words with an occurrence of 1 and the reducer sums up the occurrences from the maps of the same words.

## HW1/part2/WordCount.java

This program counts the number of double-word occurrences in the files located in the specified input folder. The mapper loops through all the words and emits the previous word and the current word with an occurrence of 1. This effectively maps all adjacent words from the files. The reducer than performs the same actions as in *HW1/part1/WordCount.java*.

## HW1/part3/WordCount.java

This program counts the number of one-word occurrences in the files located in the specified input folder. ToolRunner is used to configure the job and to configure the mapper. The word-patterns.txt file is given to the different data nodes using DistributedCache and the words in word-patterns.txt is stored in a Set object in the Mapper class. The mapper then only emits words that match the words in the word patterns Set. The reducer than performs the same actions as in *HW1/part1/WordCount.java*.

# How to Run

## One-Word occurrence

hadoop jar wordcount.jar HW1.WordCount /user/ec2-user/input /user/ec2-user/output

## Double-Word occurrence

hadoop jar wordcount.jar HW1.WordCount /user/ec2-user/input /user/ec2-user/output

## One-Word occurrence using word-patterns.txt and DistributedCache

hadoop jar wordcount.jar HW1.WordCount /user/ec2-user/input /user/ec2-user/output /user/ec2-user/word-patterns.txt

# Results

## WordCount – one-word occurrence

The output is a single file showing pairs of <word, occurrences> corresponding to the word and the number of times it has been seen in the output as shown in Fig. 3.

## WordCount – two-word occurrence

The output is a single file showing triples of <word1, word2, occurrences> where word1 and word2 are adjacent to each other in the file and occurrences is the number of times these two words appear together as shown in Fig. 4.

## WordCount – one-word occurrence of words in word-patterns.txt in distributed cache

The output is in the same format as that of *WordCount – one-word occurrence*. Only the words from the input files that appear in word-patterns.txt are mapped. This is shown in Fig. 5.

# References

MapReduce Tutorial - <https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html>

# Screenshots



Fig. 1. Starting the Hadoop cluster

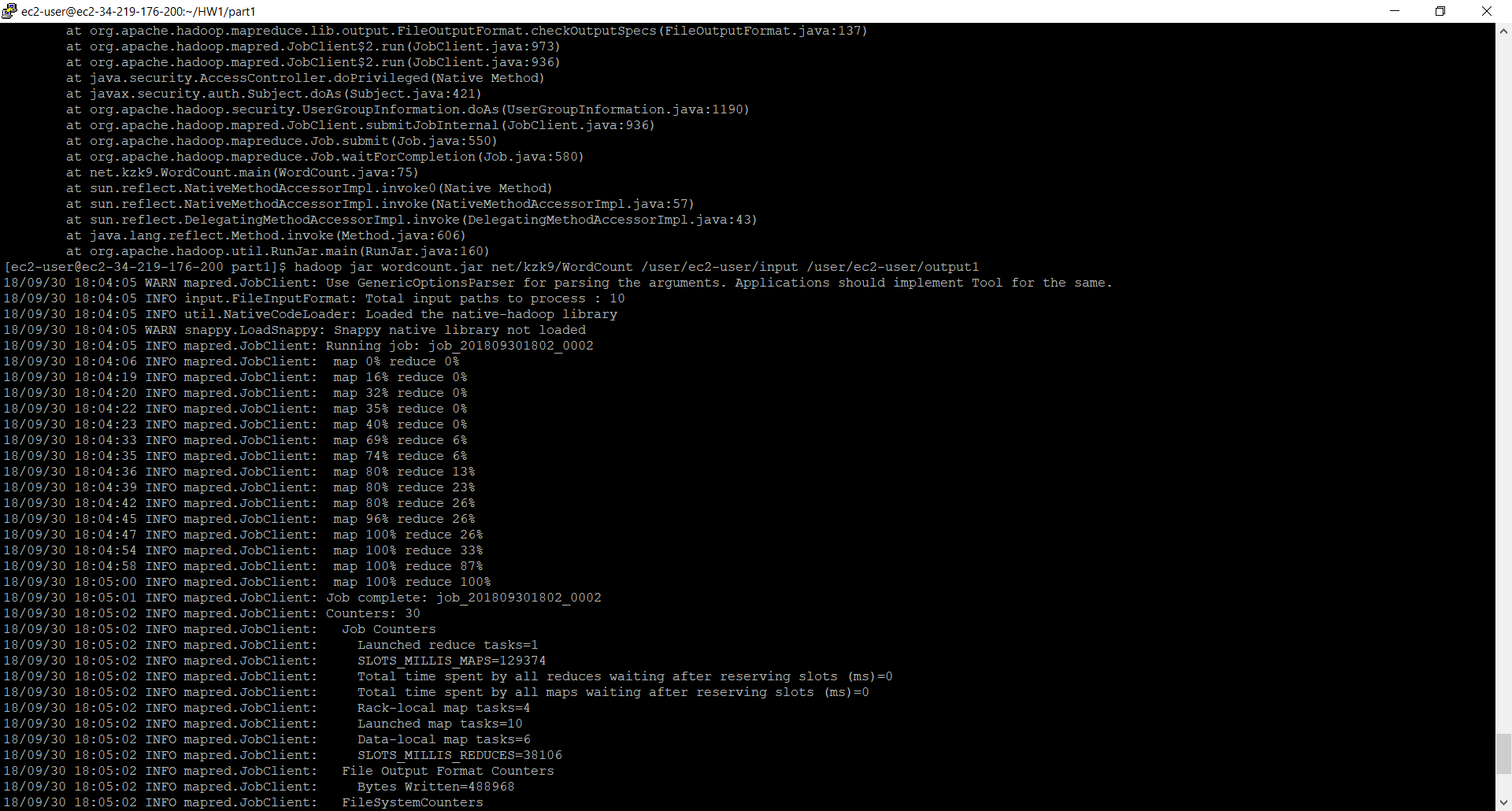


Fig. 2. Running the WordCount program

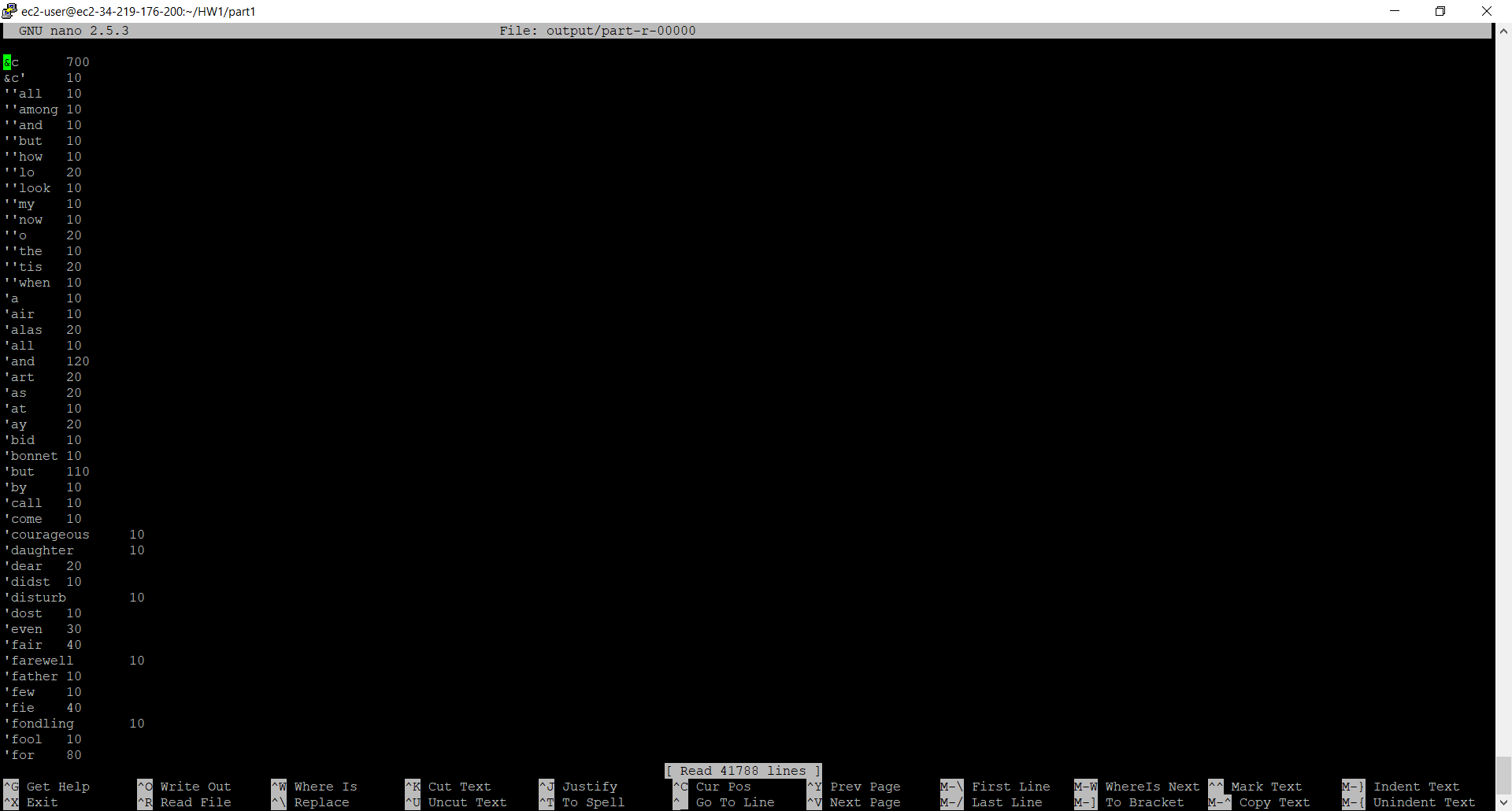


Fig. 3. Output of the one word occurrences

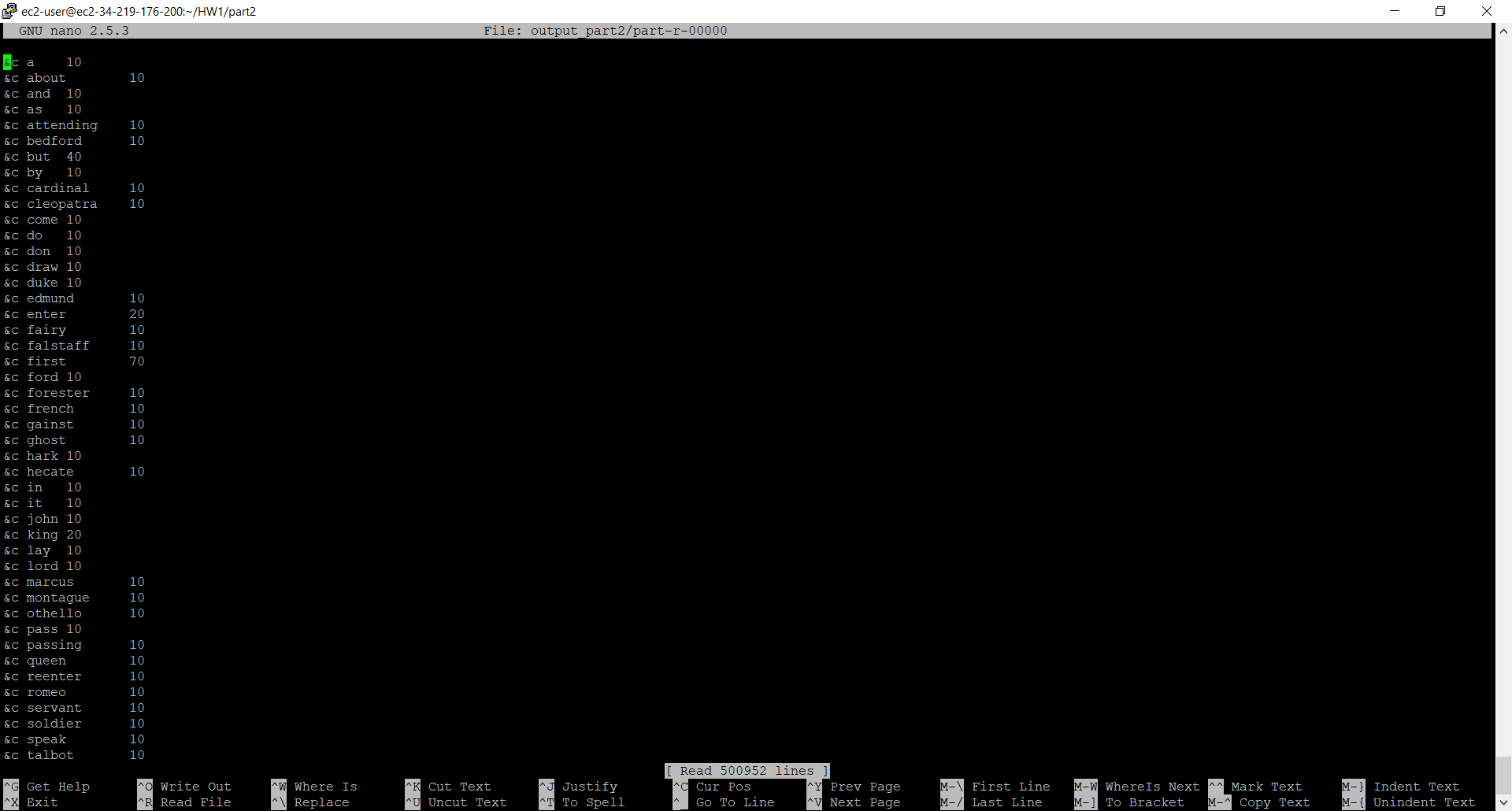


Fig. 4. Output of the two-word occurrences

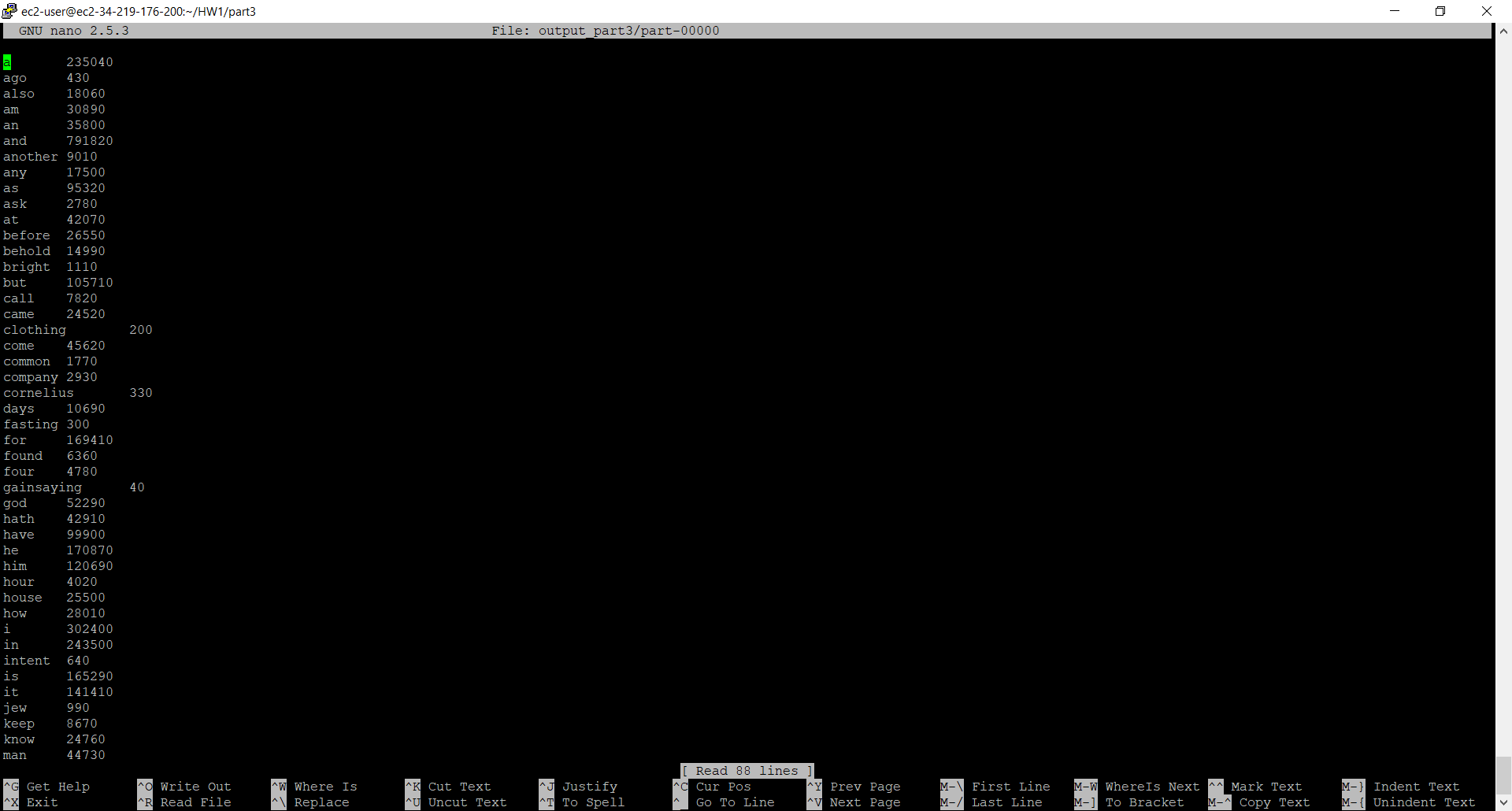


Fig. 5. Output of the one-word occurrences of the words in word-patterns.txt from the distributed cache

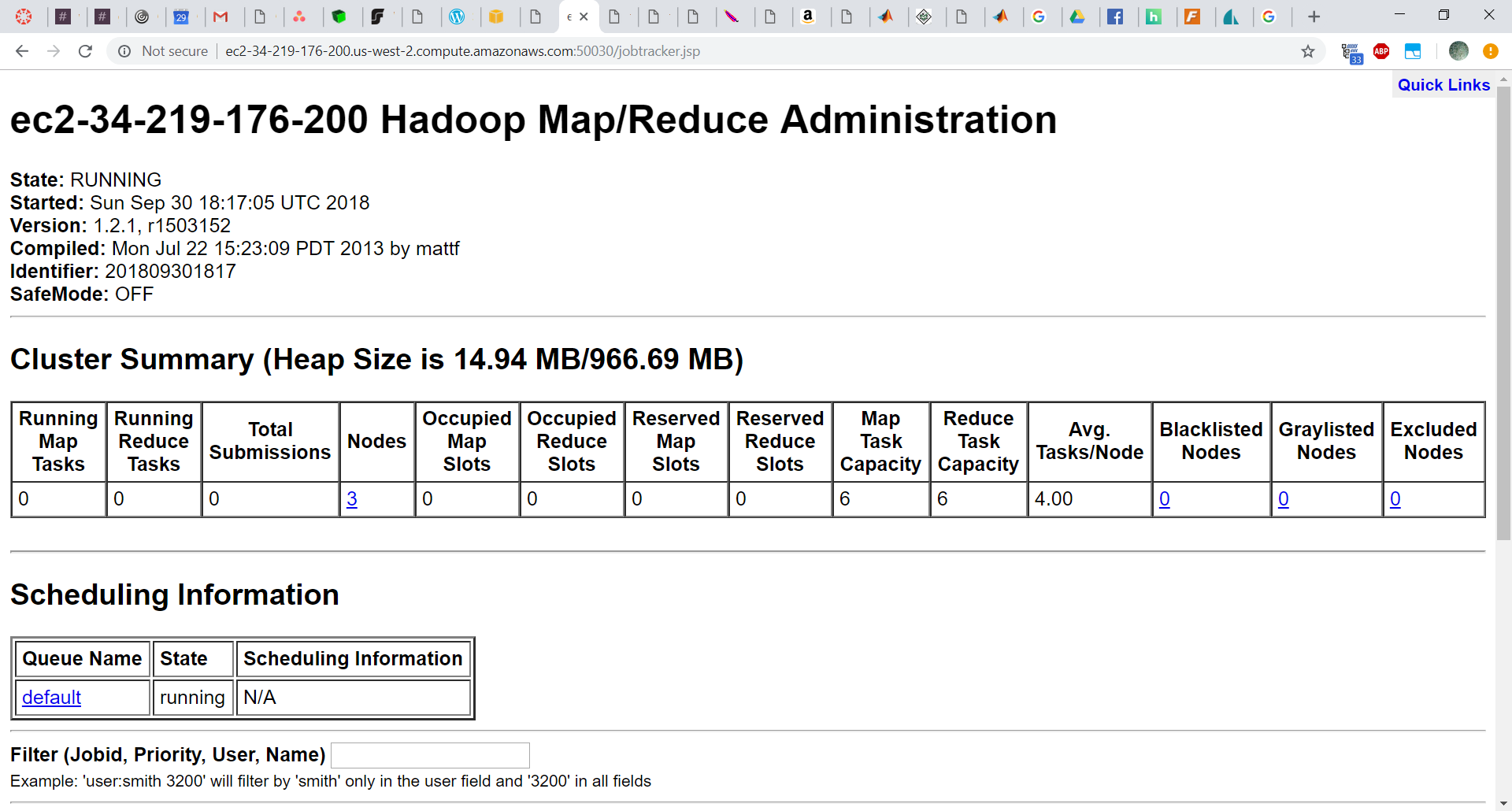


Fig. 6. Hadoop cluster showing 3 active nodes

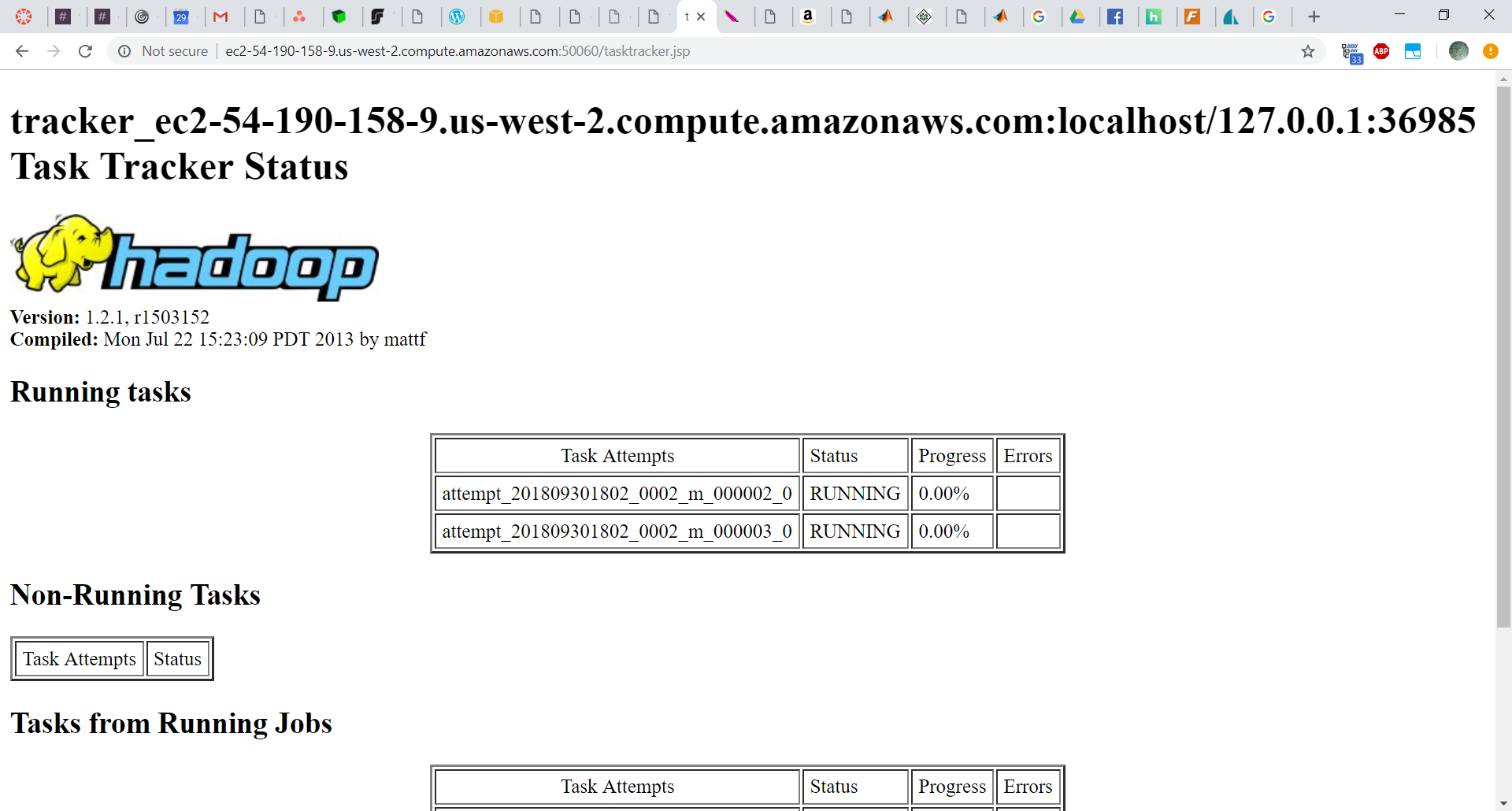


Fig. 7. Hadoop node starting tasks

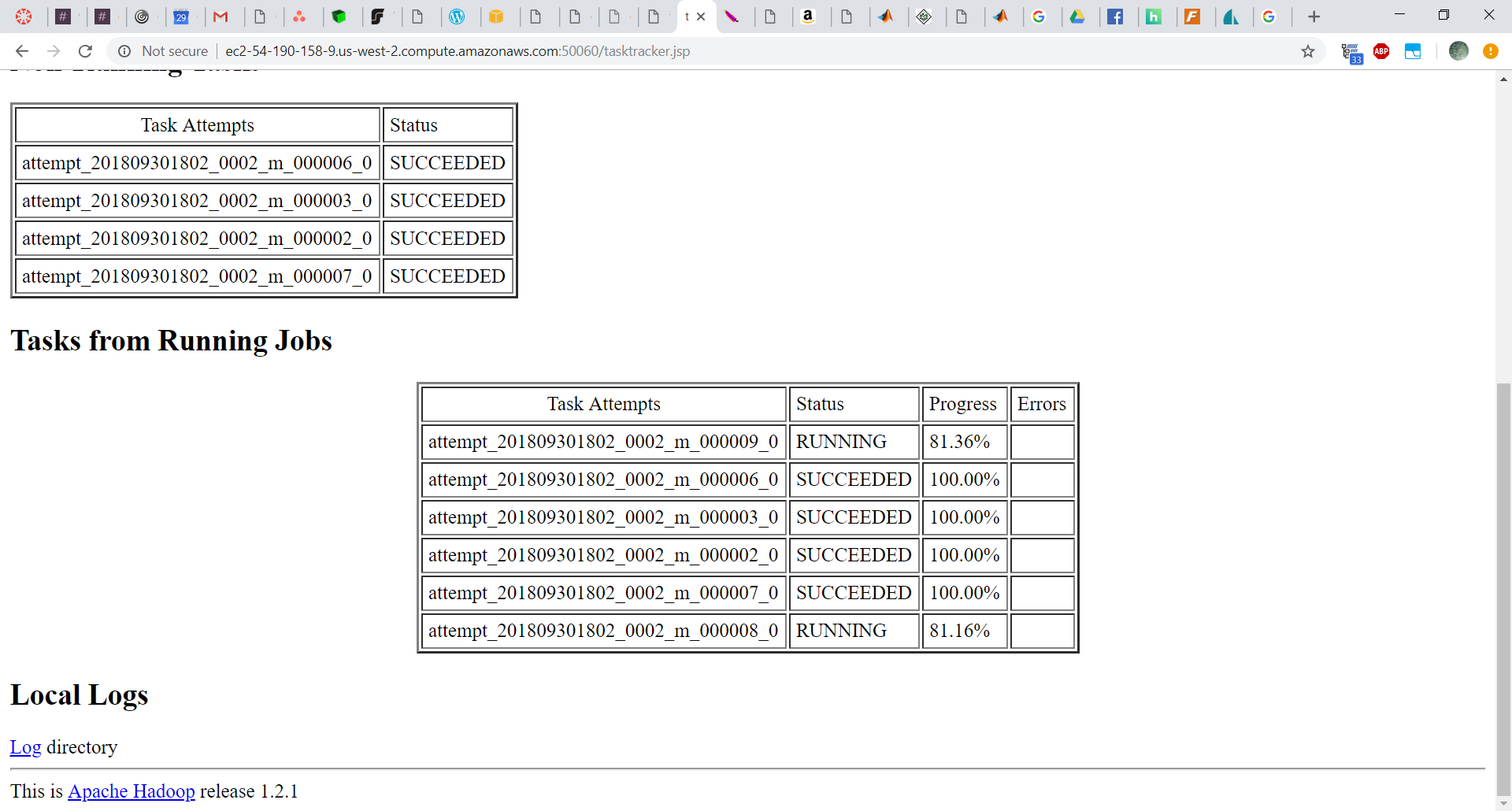


Fig. 8. Hadoop node with completed and running tasks

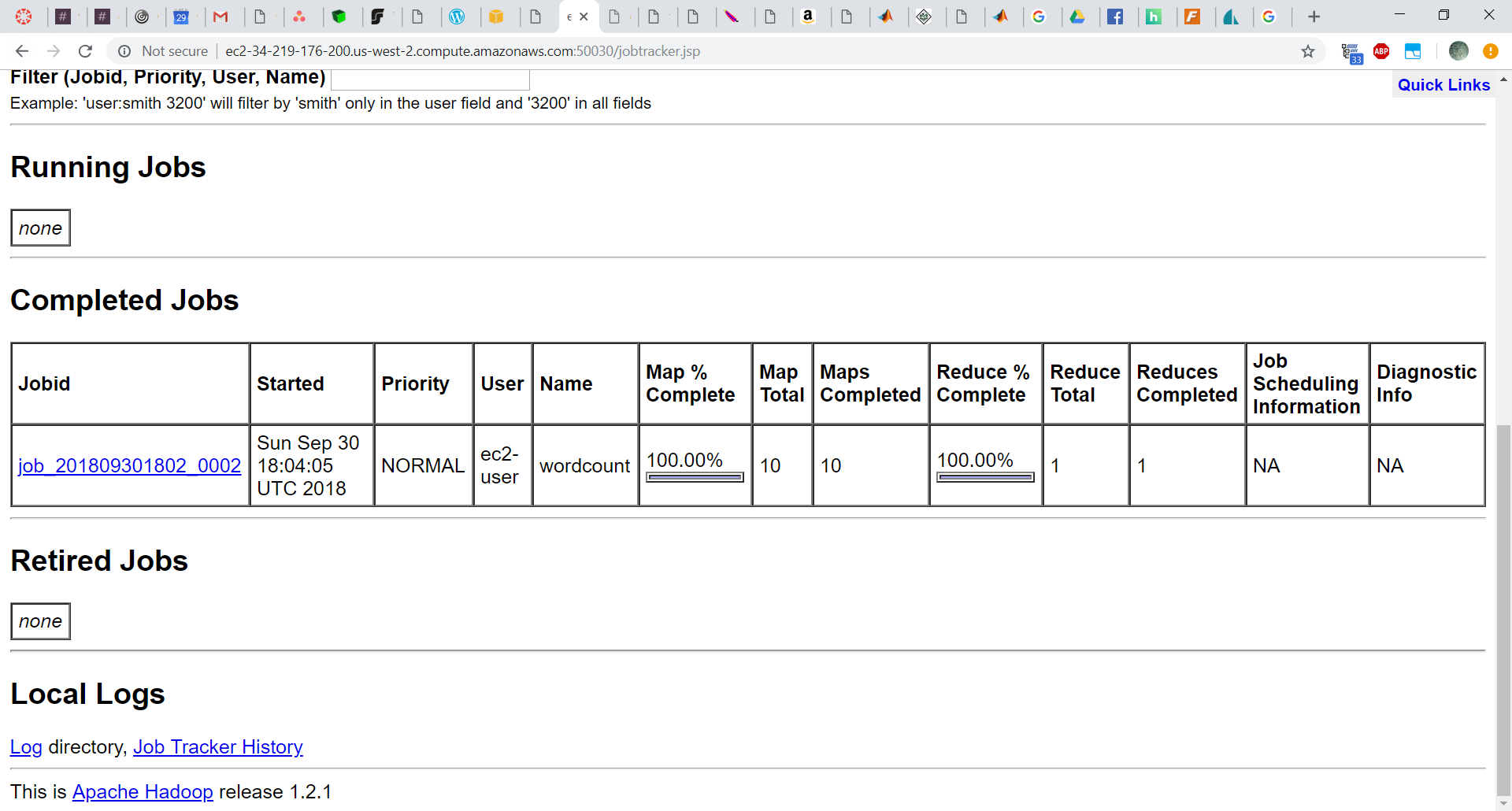


Fig. 9. Hadoop showing completed WordCount job