ASSIGNMENT 5 - WORDCOUNT USING MAPREDUCE AND HADOOP

40164521
Dhananjay Narayan

In my implementation, I have taken the input_file data from lorem ipsum. In the src folder, there are two files WordCount and StopWordList. The StopWords.java file contains the list of some of the stop words. The WordCount.java file contains the main logic to find the count of the words in the input_file.txt. I have also added the jar file of the application.
So, MapReduce is generally used to process huge amounts of data. As per the name, we can guess that there are 2 phases in the MapReduce algorithm - The Mapper & The Reducer.

Mapping phases involves processing the text file to generate key-value pairs using tokenization. Initially, the value for each of the keys is set as 1. So we have now split the text file into many numbers of words. For ex, if a line says “He is a Hero”. The mapper will generate 4 keys for each of the words with initial value count of 1.

In the Reduce phase, we group the keys generated chronologically from the mapping phase together. After this grouping, the values for each occurrence of the key is added up. One of the reduce tasks will be handling each of the occurrences of the word, through hashing. This helps in faster processing and saves time as the words are distributed and all of the task nodes are being run in parallel.

Command to run the jar file - hadoop jar MapReduceAssignmentDhananjay.jar WordCountAssignment.WordCount /input_dir /output_dir
Reading the output file- hadoop fs -cat /output_dir/*

Output Sample Snippet:

```
C:\hadoop>hadoop fs -cat /output_dir/*
a       1
ac      1
accumsan.       1
adipiscing      1
aenean  1
aliquam 2
amet    4
amet,   2
ante    2
arcu    2
at      6
auctor  1
augue   2
```