

## Assignment- 1

**AIM:** Write a code in JAVA for a implement WordCount application that counts the number of occurrence of each word in a given input set using the Hadoop MapReduce framework on local-standalone set-up.

### WordCount.java

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

    public static class TokenizerMapper
        extends Mapper<Object, Text, Text, IntWritable>{

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context
            ) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class IntSumReducer
        extends Reducer<Text,IntWritable,Text,IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values,
                          Context context
                          ) throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
```

```

Configuration conf = new Configuration();
Job job = Job.getInstance(conf, "word count");
job.setJarByClass(WordCount.class);
job.setMapperClass(TokenizerMapper.class);
job.setCombinerClass(IntSumReducer.class);
job.setReducerClass(IntSumReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
System.exit(job.waitForCompletion(true) ? 0: 1);
}
}

```

## **Input: input.txt**

WordCount example reads text files and counts how often words occur. The input is text files, and the output is text files, each line of which contains a word and the count of how often it occurred, separated by a tab.

Map Reduce Project that works on weather data and process it, the final outcome of the project can be processed further to find similarities on different weather stations

The Shadow was an American pulp magazine published by Street & Smith from 1931 to 1949. Each issue contained a novel about The Shadow, a mysterious crime-fighting figure who spoke the line "Who knows what evil lurks in the hearts of men? The Shadow knows" in radio broadcasts of stories from Street & Smith's Detective Story Magazine. For the first issue, dated April 1931, Walter Gibson wrote the lead novel.

## **Output: output file (part-r-00000)**

```

part-r-00000
~/Desktop/Wordcountexp
11 For 1
12 Gibson 1
13 Magazine. 1
14 Map 1
15 Project 1
16 Reduce 1
17 Shadow 2
18 Shadow, 1
19 Smith 1
20 Smith's 1
21 Story 1
22 Street 2|
23 The 4
24 Walter 1
25 WordCount 1
26 a 4
27 about 1
28 an 1
29 and 4
30 be 1
31 broadcasts 1
32 by 2
33 can 1
34 contained 1
35 contains 1
36 count 1
37 counts 1
38 crime-fighting 1
39 data 1
40 dated 1
41 different 1
42 each 1
43 evil 1
44 example 1
45 figure 1
46 files 2
47 files, 1

```

## **Wordcount Steps to run:**

1. Starting Hadoop

**\$ start-all.sh**

2. Made A folder “wordcountexp” and write WordCount.java code.

3. Create new folder for input data.
4. Add input text file in the input data folder.
5. Create new folder to hold java class files.
6. Set HADOOP\_CLASSPATH environment variable.

```
$ export HADOOP_CLASSPATH=$(hadoop classpath)
```

7. Create a directory on HDFS

```
$ hdfs dfs -mkdir /WordCountTut
```

```
$ hdfs dfs -mkdir /WordCountTut/Input
```

8. Upload the input file (device) to that directory.

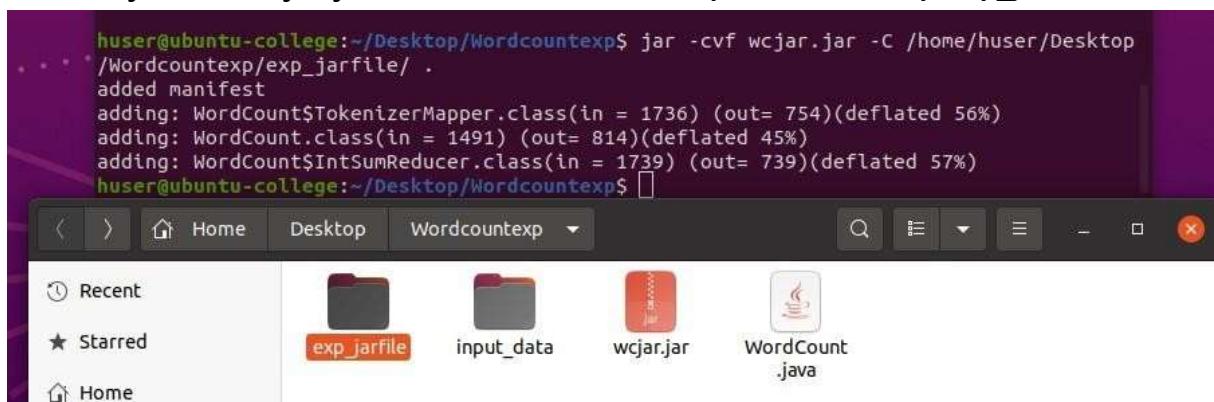
```
$ hdfs dfs -put '/home/huser/Desktop/Wordcountexp/input_data/input.txt'  
/WordCountTut/Input
```

9. Compile the java code:

```
$ javac -classpath $(HADOOP_CLASSPATH) -d '/home/huser/Desktop/Wordcountexp/  
exp_jarfile' /home/huser/Desktop/Wordcountexp/.java
```

10. Creation .jar file of classes:

```
$ jar -cvf wcjar.jar -C '/home/huser/Desktop/Wordcountexp/exp_jarfile/ .
```



11. Running the jar file on Hadoop

```
$ hadoop jar wcjar.jar WordCount /WordCountTut/Input /WordCountTut/Output
```

```
huser@ubuntu-college:~/Desktop/Wordcountexp$ hadoop jar wcjar.jar WordCount /WordCount  
Tut/Input /WordCountTut/Output  
2022-04-11 23:21:14,369 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0  
.1:8032  
2022-04-11 23:21:17,291 WARN mapreduce.JobResourceUploader: Hadoop command-line option  
parsing not performed. Implement the Tool interface and execute your application with  
ToolRunner to remedy this.  
2022-04-11 23:21:17,535 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding f  
or path: /tmp/hadoop-yarn/staging/huser/.staging/job_1649697057322_0001  
2022-04-11 23:21:18,170 INFO input.FileInputFormat: Total input files to process : 1  
2022-04-11 23:21:18,286 INFO mapreduce.JobSubmitter: number of splits:1  
2022-04-11 23:21:18,812 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_16  
49697057322_0001  
2022-04-11 23:21:18,813 INFO mapreduce.JobSubmitter: Executing with tokens: []  
2022-04-11 23:21:18,813 Peak Reduce Physical memory (bytes)=167215104 VM not found  
2022-04-11 23:21:18,813 Peak Reduce Virtual memory (bytes)=2533072896  
1 Shuffle Errors  
BAD_ID=0  
CONNECTION=0  
IO_ERROR=0  
WRONG_LENGTH=0  
WRONG_MAP=0  
WRONG_REDUCE=0  
File Input Format Counters  
Bytes Read=800  
File Output Format Counters  
Bytes Written=858
```

## 12. Check output on localhost:9870 /localhost:50070

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

### Browse Directory

/WordCountTut/Output

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rW-r--r--	huser	supergroup	0 B	Apr 11 23:23	1	128 MB	_SUCCESS
-rw-r--r--	huser	supergroup	858 B	Apr 11 23:23	1	128 MB	part-r-00000

Showing 1 to 2 of 2 entries Previous 1 Next

Hadoop, 2021.

### Browse Directory

/WordCountTut

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	huser	supergroup	0 B	Apr 11 23:06	0	0 B	Input
drwxr-xr-x	huser	supergroup	0 B	Apr 11 23:23	0	0 B	Output

Showing 1 to 2 of 2 entries Previous 1 Next

Hadoop, 2021.

File information - part-r-00000

Download Head the file (first 32K) Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073741886  
Block Pool ID: BP-1388353168-127.0.1.1-1647528100285  
Generation Stamp: 1062  
Size: 858  
Availability:

- ubuntu-college

File contents

```
a 4
about 1
an 1
and 4
be 1
broadcasts 1
by 2
can 1
contained 1
contains 1
count 1
counts 1
```