Lab 5

Implement Wordcount program on Hadoop framework

Driver code

```
// Importing libraries import
java.io.IOException; import
org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapred.FileInputFor
mat; import
org.apache.hadoop.mapred.FileOutputFo
rmat; import
org.apache.hadoop.mapred.JobClient;
import
org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

   public int run(String[] args) throws
IOException {        if (args.length < 2) {
          System.out.println("Please give valid
inputs");            return -1;
      }

      JobConf conf = new JobConf(WCDriver.class);
conf.setJobName("WordCount");

      FileInputFormat.setInputPaths(conf, new Path(args[0]));
      FileOutputFormat.setOutputPath(conf, new Path(args[1]));

      conf.setMapperClass(WCMapper.class);
      conf.setReducerClass(WCReducer.class);

      conf.setMapOutputKeyClass(Text.class);
      conf.setMapOutputValueClass(IntWritable.class);

      conf.setOutputKeyClass(Text.class);
      conf.setOutputValueClass(IntWritable.class);

      JobClient.runJob(conf);
      return 0;
   }
```

```
    // Main Method
    public static void main(String[] args) throws
Exception {         int exitCode =
ToolRunner.run(new WCDriver(), args);
        System.out.println("Job Exit Code: " + exitCode);
    }
}
```

Mapper Code

```
// Importing libraries import
java.io.IOException; import
org.apache.hadoop.io.IntWritable;
import
org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapred.MapReduce
Base; import
org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements Mapper<LongWritable,
Text, Text, IntWritable> {

    // Map function
    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable>
output, Reporter reporter)             throws IOException {
        String line = value.toString();

        // Splitting the line on whitespace          for
(String word : line.split("\\s+")) {             if
(word.length() > 0) {
output.collect(new Text(word), new
IntWritable(1));
            }
        }
    }
}
```

Reducer Code

```
// Importing libraries
import
java.io.IOException;
import java.util.Iterator;
```

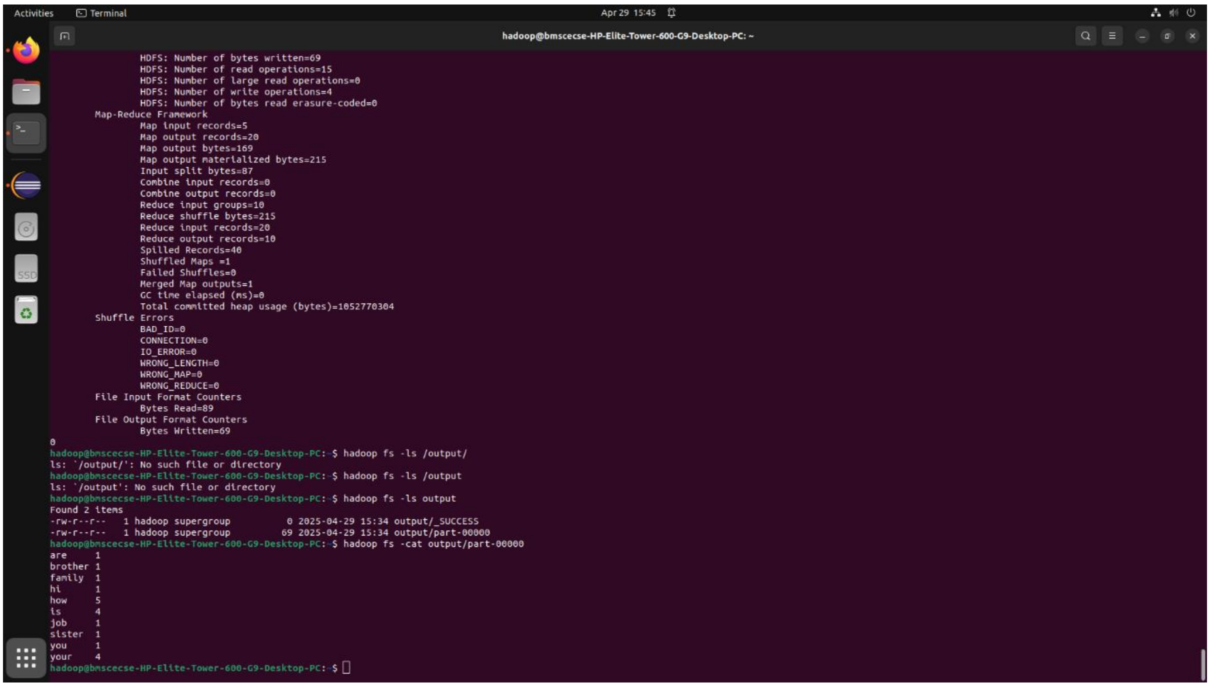
```
import
org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapred.MapReduce
Base; import
org.apache.hadoop.mapred.OutputColle
ctor; import
org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer<Text,
IntWritable, Text, IntWritable> {

    // Reduce function
    public void reduce(Text key, Iterator<IntWritable> values,
                        OutputCollector<Text,
IntWritable> output,                        Reporter
reporter) throws IOException {         int
count = 0;

        // Counting the frequency of each
word         while (values.hasNext()) {
            count += values.next().get();
        }

        output.collect(key, new IntWritable(count));
    }
}
```



```
                HDFS: Number of bytes written=69
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=5
                Map output records=20
                Map output bytes=169
                Map output materialized bytes=215
                Input split bytes=87
                Combine input records=0
                Combine output records=0
                Reduce input groups=10
                Reduce shuffle bytes=215
                Reduce input records=20
                Reduce output records=10
                Spilled Records=40
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=1052770304
        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=89
        File Output Format Counters
                Bytes Written=69
0
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /output/
ls: `/output/': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /output
ls: `/output': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls output
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-04-29 15:34 output/_SUCCESS
-rw-r--r--   1 hadoop supergroup         69 2025-04-29 15:34 output/part-00000
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -cat output/part-00000
are     1
brother 1
family  1
hi      1
how     5
is      4
job     1
sister  1
you     1
your    4
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ 
```