Hadoop Lab

CSE 328

Assignment-4


```
Gyanendra Kr. Shukla
CSE 1
191112040
```

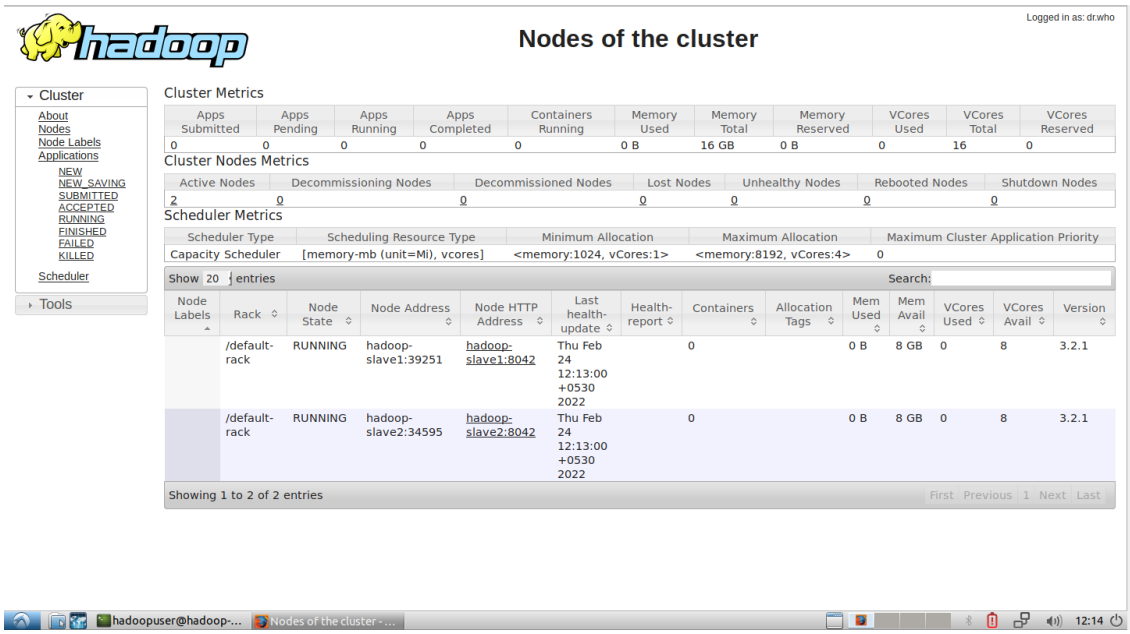

1. Compile and Run the given java file

I followed these steps to compile and run the word count map reduce file on hadoop-

1. ssh to `localhost` and switch to `hadoopuser` using `su hadoopuser`.

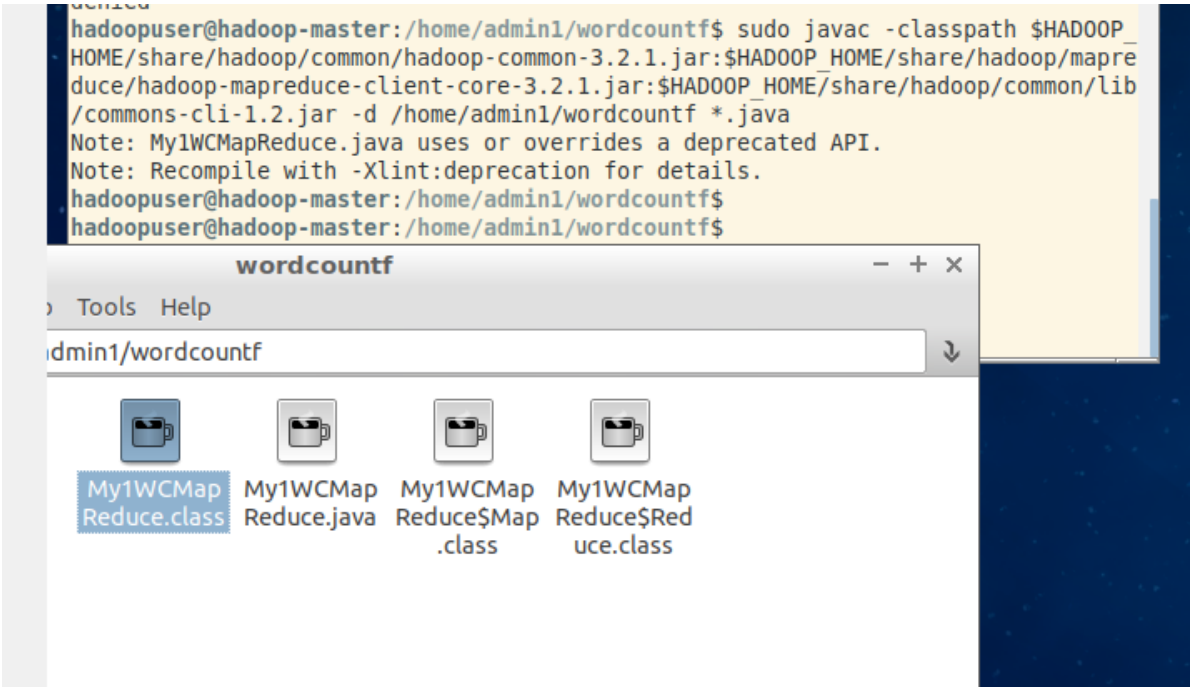
```
admin1@hadoop-master:~$ su  hadoopuser
Password:
hadoopuser@hadoop-master:/home/admin1$ cd /usr/local/hadoop/sbin/
hadoopuser@hadoop-master:/usr/local/hadoop/sbin$ bash start-dfs.sh
Starting namenodes on [hadoop-master]
Starting datanodes
Starting secondary namenodes [hadoop-master]
hadoopuser@hadoop-master:/usr/local/hadoop/sbin$ bash start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoopuser@hadoop-master:/usr/local/hadoop/sbin$ jps
3280 Jps
2723 SecondaryNameNode
2982 ResourceManager
2475 NameNode
hadoopuser@hadoop-master:/usr/local/hadoop/sbin$ hdfs dfs -mkdir /user
```

2. Start `dfs` and `yarn` using `bash start-dfs.sh` and `bash start-yarn.sh` respectively.



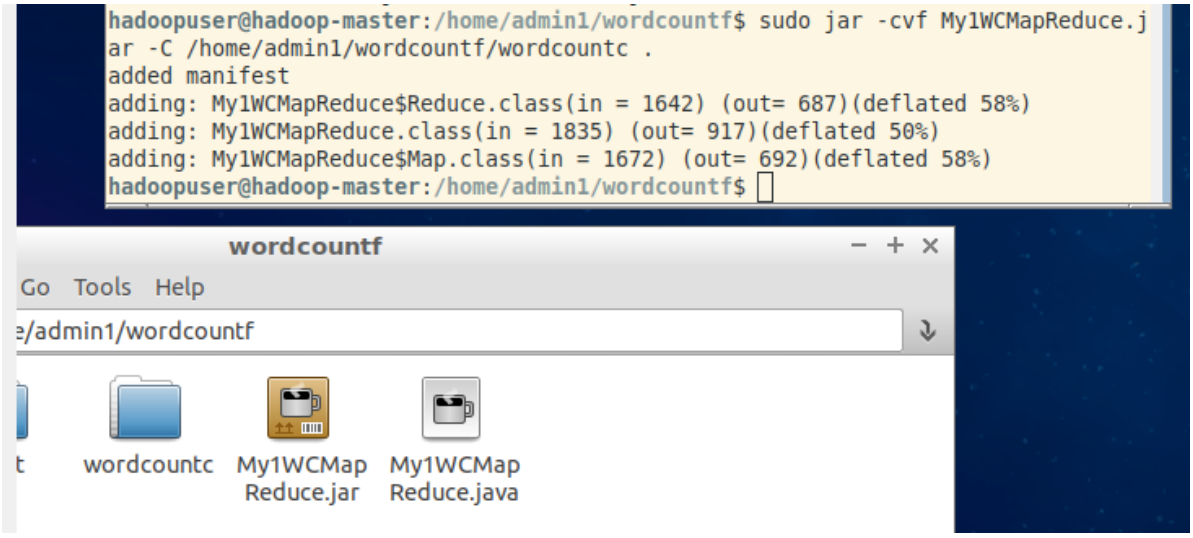
3. Move the input data to hdfs using `bin/hdfs dfs -put /home/admin1/input user/inputdata`.
4. cd to the directory containing the wordcount java file.
5. To compile the java program, run -

```
javac -classpath $HADOOP_HOME/share/common/hadoop-common-3.2.1.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-core-3.2.1.jar:$HADOOP_HOME/share/hadoop/common/lib/commons-cli-1.2.jar -d /home/hadoopuser/wordcount *.java
```



6. Three `.class` files will be created in the same directory. Move them to a new directory.
7. Convert the `.class` files to `.jar` files using

```
jar -cvf My1WCMapReduce.jar -C /home/hadoopuser/wordcount/wordcountf .
```



8. `cd` to hadoop installation directory with `cd /usr/local/hadoop`.
9. Execute the jar file with

```
bin/hadoop jar /home/hadoopuser/wordcountf/My1WCMapReduce.jar My1WCMapReduce
/user/inputdata outputwc
```

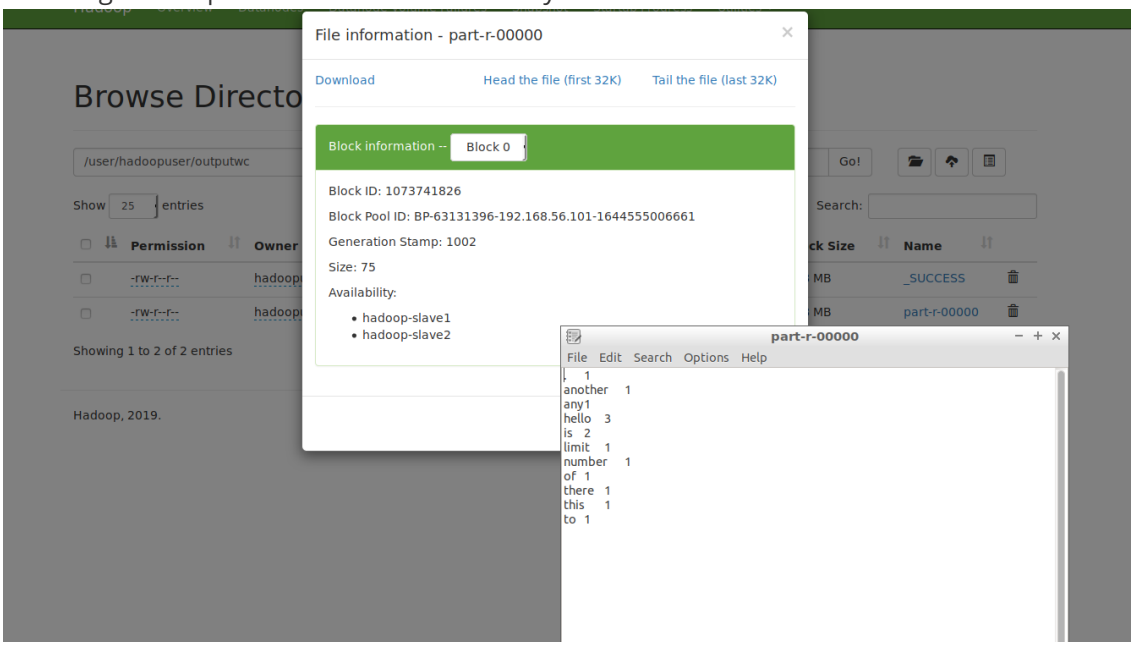
```
hadoopuser@hadoop-master:/home/admin1/wordcountf$ cd /usr/local/hadoop
hadoopuser@hadoop-master:/usr/local/hadoop$ bin/hadoop jar /home/admin1/wordcountf/My1WCMapReduce.jar My1WCMapReduce /user/inputdata
outputwc
2022-02-24 12:38:53,184 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2022-02-24 12:38:55,343 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2022-02-24 12:38:55,346 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2022-02-24 12:38:57,473 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool inte
rface and execute your application with ToolRunner to remedy this.
2022-02-24 12:38:58,381 INFO input.FileInputFormat: Total input files to process : 1
2022-02-24 12:38:58,454 INFO mapreduce.JobSubmitter: number of splits:1
2022-02-24 12:39:00,528 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1840031340_0001
2022-02-24 12:39:00,532 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-02-24 12:39:01,550 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2022-02-24 12:39:01,551 INFO mapreduce.Job: Running job: job_local1840031340_0001
2022-02-24 12:39:01,757 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2022-02-24 12:39:01,789 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2022-02-24 12:39:01,796 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2022-02-24 12:39:01,810 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2022-02-24 12:39:02,758 INFO mapreduce.Job: Job job_local1840031340_0001 running in uber mode : false
2022-02-24 12:39:02,802 INFO mapreduce.Job:  map 0% reduce 0%
2022-02-24 12:39:03,248 INFO mapred.LocalJobRunner: Waiting for map tasks
2022-02-24 12:39:03,249 INFO mapred.LocalJobRunner: Starting task: attempt_local1840031340_0001_m_000000_0
2022-02-24 12:39:03,678 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2022-02-24 12:39:03,684 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2022-02-24 12:39:04,042 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2022-02-24 12:39:04,065 INFO mapred.MapTask: Processing split: hdfs://hadoop-master:9000/user/inputdata/inputf.txt:0+69
2022-02-24 12:39:05,662 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2022-02-24 12:39:05,663 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2022-02-24 12:39:05,663 INFO mapred.MapTask: soft limit at 83886080
2022-02-24 12:39:05,663 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2022-02-24 12:39:05,663 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2022-02-24 12:39:06,321 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2022-02-24 12:39:09,506 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
```

```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=138
                HDFS: Number of bytes written=75
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=6
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=12
                Map output records=14
                Map output bytes=124
                Map output materialized bytes=158
                Input split bytes=116
                Combine input records=0
                Combine output records=0
                Reduce input groups=11
                Reduce shuffle bytes=158
                Reduce input records=14
                Reduce output records=11
                Spilled Records=28
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=331
                Total committed heap usage (bytes)=244187136
        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=69
        File Output Format Counters
                Bytes Written=75
hadoopuser@hadoop-master:/usr/local/hadoop$
```

10. Checkin the output using `bin/hdfs dfs -cat outputwc/*`

```
hadoopuser@hadoop-master:/usr/local/hadoop$ bin/hdfs dfs -cat outputwc/*
2022-02-24 12:56:00,794 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
.       1
another 1
any     1
hello   3
is      2
limit   1
number  1
of      1
there   1
this    1
to      1
hadoopuser@hadoop-master:/usr/local/hadoop$
```

11. Checking the output in the hdfs web directory interface.



2. Compile and Run the example jar file

Since all the examples in the hadoop are already in `jar` format, we can directly execute them.

1. Move the `etc/hadoop/*.xml` files to input directory with `bin/hdfs dfs -put /etc/hadoop/* /user/inputdata`.

```
hadoopuser@hadoop-master:/usr/local/hadoop$ bin/hdfs dfs -put etc/hadoop/*.xml /user/inputdata
2022-02-24 12:50:53,216 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:50:53,942 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:50:54,177 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:50:54,425 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:50:54,581 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:50:54,809 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:50:55,001 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:50:55,175 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:50:55,396 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
```

2. Run the example using `bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar grep input output 'dfs[a-z.]+'

```
hadoopuser@hadoop-master:/usr/local/hadoop$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar grep /user/inpu
tdata output 'dfs[a-z.]+'
2022-02-24 12:53:06,364 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2022-02-24 12:53:07,364 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2022-02-24 12:53:07,365 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2022-02-24 12:53:08,553 INFO input.FileInputFormat: Total input files to process : 10
2022-02-24 12:53:08,707 INFO mapreduce.JobSubmitter: number of splits:10
2022-02-24 12:53:09,866 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2144738042_0001
2022-02-24 12:53:09,866 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-02-24 12:53:10,397 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2022-02-24 12:53:10,398 INFO mapreduce.Job: Running job: job_local2144738042_0001
2022-02-24 12:53:10,424 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2022-02-24 12:53:10,455 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2022-02-24 12:53:10,461 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2022-02-24 12:53:10,464 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2022-02-24 12:53:10,691 INFO mapred.LocalJobRunner: Waiting for map tasks
2022-02-24 12:53:10,692 INFO mapred.LocalJobRunner: Starting task: attempt_local2144738042_0001_m_000000_0
2022-02-24 12:53:10,832 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2022-02-24 12:53:10,834 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2022-02-24 12:53:10,932 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2022-02-24 12:53:10,938 INFO mapred.MapTask: Processing split: hdfs://hadoop-master:9000/user/inputdata/hadoop-policy.xml:0+11392
2022-02-24 12:53:11,259 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2022-02-24 12:53:11,260 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2022-02-24 12:53:11,260 INFO mapred.MapTask: soft limit at 83886080
2022-02-24 12:53:11,261 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2022-02-24 12:53:11,261 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2022-02-24 12:53:11,437 INFO mapreduce.Job: Job job_local2144738042_0001 running in uber mode : false
2022-02-24 12:53:11,529 INFO mapreduce.Job:  map 0% reduce 0%
2022-02-24 12:53:11,570 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2022-02-24 12:53:11,919 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2022-02-24 12:53:12,522 INFO mapred.LocalJobRunner:
2022-02-24 12:53:12,556 INFO mapred.MapTask: Starting flush of map output
2022-02-24 12:53:12,556 INFO mapred.MapTask: Spilling map output
```

3. Check the output using `bin/hdfs dfs -cat output/*`

```
hadoopuser@hadoop-master:/usr/local/hadoop$ bin/hdfs dfs -cat output/*
2022-02-24 12:55:29,000 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
1       dfsadmin
1       dfs.replication
1       dfs.namenode.name.dir
1       dfs.datanode.data.dir
hadoopuser@hadoop-master:/usr/local/hadoop$
```

The Map Reduce File used in 1st question -

```java
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

public class My1WCMapReduce
{
    public static class Map extends Mapper<LongWritable,Text,Text,IntWritable> {
        public void map(LongWritable key, Text value,Context context) throws
        IOException,InterruptedException{
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                value.set(tokenizer.nextToken());
                context.write(value, new IntWritable(1));
            }
        }
    }

    public static class Reduce extends Reducer<Text,IntWritable,Text,IntWritable> {
        public void reduce(Text key, Iterable<IntWritable> values,Context context)
            throws IOException,InterruptedException {
                int sum=0;
                for(IntWritable x: values) {
                sum+=x.get();
                }
                context.write(key, new IntWritable(sum));
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf= new Configuration();
        Job job = new Job(conf,"Our Word Count Program");
        job.setJarByClass(My1WCMapReduce.class);
```

```
        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        Path outputPath = new Path(args[1]);
        //Configuring the input/output path from the filesystem into the job
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        //deleting the output path automatically from hdfs so that we don't have to
        //delete it explicitly
        outputPath.getFileSystem(conf).delete(outputPath);
        //exiting the job only if the flag value becomes false
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```