NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. VIIIth Sem

Subject Code: CS-484

Subject Name: Cloud Computing

Submitted By:

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

Q. Write and Run Word Count Mapreduce.

(Reference: https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html)

➔ **Filename: 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);
  }
}
```

Foldername/Filename: /input/file01

Hello World Bye World

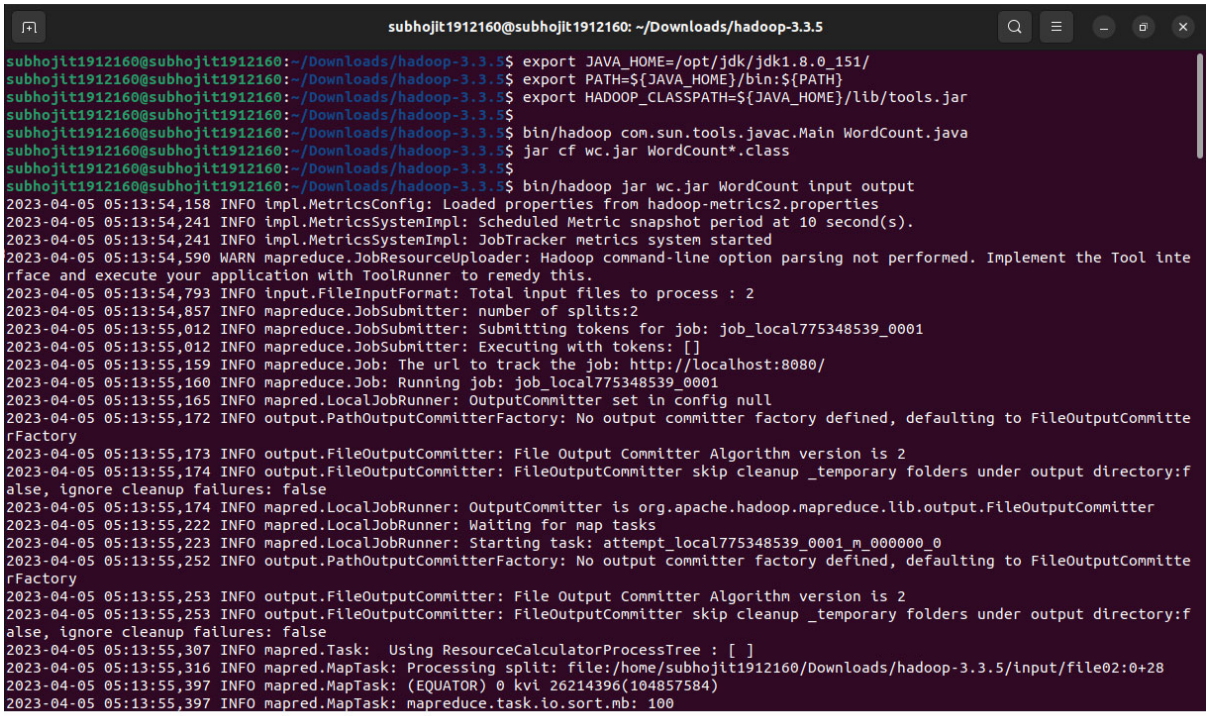
Foldername/Filename: /input/file02

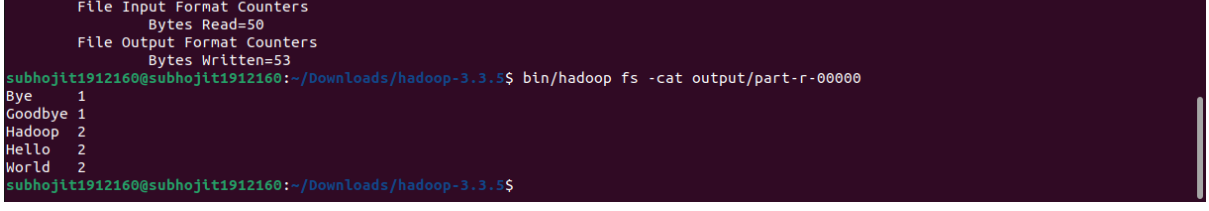
Hello Hadoop Goodbye Hadoop

Execution:

```
$ bin/hadoop com.sun.tools.javac.Main WordCount.java
$ jar cf wc.jar WordCount*.class
$ bin/hadoop jar wc.jar WordCount input output
$ $ bin/hadoop fs -cat output/part-r-00000
```

Output:


subhojit1912160@subhojit1912160: ~/Downloads/hadoop-3.3.5
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$ export JAVA_HOME=/opt/jdk/jdk1.8.0_151/
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$ export PATH=${JAVA_HOME}/bin:${PATH}
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$ export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$ bin/hadoop com.sun.tools.javac.Main WordCount.java
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$ jar cf wc.jar WordCount*.class
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$ bin/hadoop jar wc.jar WordCount input output
2023-04-05 05:13:54,158 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2023-04-05 05:13:54,241 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2023-04-05 05:13:54,241 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2023-04-05 05:13:54,590 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2023-04-05 05:13:54,793 INFO input.FileInputFormat: Total input files to process : 2
2023-04-05 05:13:54,857 INFO mapreduce.JobSubmitter: number of splits:2
2023-04-05 05:13:55,012 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local775348539_0001
2023-04-05 05:13:55,012 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-04-05 05:13:55,159 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2023-04-05 05:13:55,160 INFO mapreduce.Job: Running job: job_local775348539_0001
2023-04-05 05:13:55,165 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2023-04-05 05:13:55,172 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2023-04-05 05:13:55,173 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-04-05 05:13:55,174 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2023-04-05 05:13:55,174 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2023-04-05 05:13:55,222 INFO mapred.LocalJobRunner: Waiting for map tasks
2023-04-05 05:13:55,223 INFO mapred.LocalJobRunner: Starting task: attempt_local775348539_0001_m_000000_0
2023-04-05 05:13:55,252 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2023-04-05 05:13:55,253 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-04-05 05:13:55,253 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2023-04-05 05:13:55,307 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
2023-04-05 05:13:55,316 INFO mapred.MapTask: Processing split: file:/home/subhojit1912160/Downloads/hadoop-3.3.5/input/file02:0+28
2023-04-05 05:13:55,397 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2023-04-05 05:13:55,397 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100


Map input records=2
Map output records=8
Map output bytes=82
Map output materialized bytes=85
Input split bytes=254
Combine input records=8
Combine output records=6
Reduce input groups=5
Reduce shuffle bytes=85
Reduce input records=6
Reduce output records=5
Spilled Records=12
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=59
Total committed heap usage (bytes)=727711744
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=50
File Output Format Counters
Bytes Written=53
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$ bin/hadoop fs -cat output/part-r-00000
Bye 1
Goodbye 1
Hadoop 2
Hello 2
World 2
subhojit1912160@subhojit1912160:~/Downloads/hadoop-3.3.5$