**Hadoop Exercise -7**

**Question) Implement a simple map-reduce code for the wordcount problem using Java. Create the jar files and run the code using HDFS.**

1. The following is the directory structure in my Ubuntu WSL:

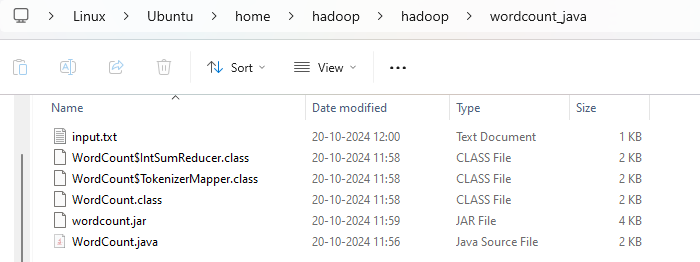
/home/hadoop/hadoop/

├── wordcount\_java/

├── input.txt/

├── WordCount.java/

├── all the wordcount classes/



1. The following is the structure in HDFS (or) Browser Directory:

/hadoop/ (user)

├── hadoop/ (parent folder)

├── wordcount\_java/

├── input/

├── input.txt/

├── output/

├── \_SUCCESS/

├── part-r-00000/

1. Start All Hadoop Daemons

A screenshot of a computer program

Description automatically generated

1. Setting up the directory structure

A black background with white text

Description automatically generated

1. Write the WordCount java program



import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.\*;

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<LongWritable, Text, Text, IntWritable> {

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

private Text word = new Text();

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

String[] words = value.toString().split("\\s+");

for (String str : words) {

word.set(str);

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]));

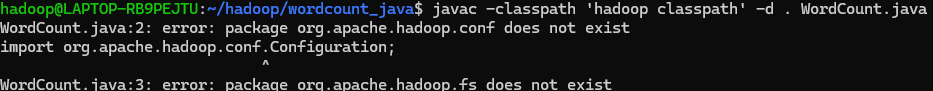
System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

1. Compile the Java Code

NOTE: THE BELOW COMMAND GIVES ERROR BECAUSE OF THE APOSTROPHE STYLE



USE THE BELOW COMMAND WITH THE CORRECT APOSTROPHE



1. Create the JAR file



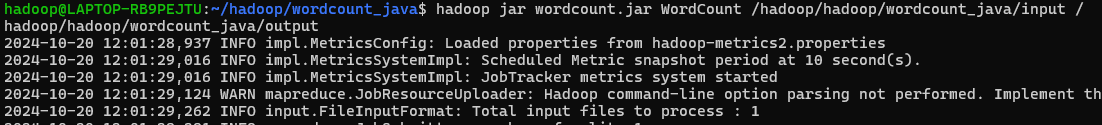
1. Create the input directory in HDFS



A screenshot of a computer

Description automatically generated

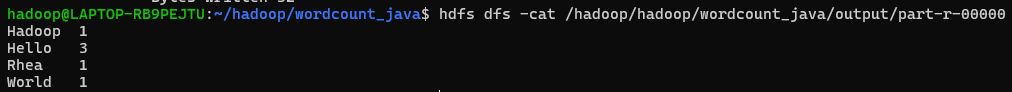
1. Run the MapReduce Job



A screenshot of a computer

Description automatically generated

1. View the output



A screenshot of a computer

Description automatically generated

1. Stop all Daemons

A screen shot of a computer program

Description automatically generated