**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);

}

}

STEPS:-

1.hadoop version (To check if hadoop is installed)

2.javac -version(To check if java c is running)

3.On the desktop create a folder named WordCountTutorial

3.1 In the folder create a document named wordcount.java and copy the code in that

3.2 Create new folder inside as input\_data and in that create a document named as input.txt

3.3 Now inside the text document write

Hello World

Hello

World

HI

RMDSTIC

RMDSTIC

RMDSTIC

This is a Pracrical Examination

3.4 Create a new folder Tutorial\_classes in WordCountTutorial (To hold the java class files)

4. Go to Terminal

4.1 Type command :-

1. export HADOOP\_CLASSPATH=$(hadoop classpath) ……………………………………………………………….(To set HADOOP\_CLASSPATH environment variable)

2.echo $HADOOP\_CLASSPATH

……………………………………………………………….(To make sure It has been set correctly)

3.hadoop fs -mkdir /WordCountTutorial

……………………………………………………………….(Create a Directory on HDFS)

4. hadoop fd -mkdir /WordCountTutorial/Input

………………………………………………………………(create a directory inside HDFS for the input)

5.Go to browser and search localhost:50070 (To check it)

5.1Once the page has been loaded go to “utilities”->”Browse the file system”->”WordCountTutorial”->”Input”

6. Go to terminal

6.1 hadoop fs -put ‘path of input.txt file’ /WordCountTutorial/Input

……………………………………………………………….(To upload the input file to that i.e HDFS directory)

7.Go to the webpage again and repeat the steps from 5 ………..(To check if the file is uploaded to the site)

8. cd /home/username/Desktop/WordCountTutorial …………(Change the current directory to the tutorial directory)

9. java -classpath ${HADOOP\_CLASSPATH} -d ‘copy path for tutorial\_classes’ ‘copy path for WordCount.java file’ …………………………(To compile the java code)

10.Check if the files are generated in the tutorial \_classes folder

11.” jar -cvf firstTutorial.jar -C tutorial\_classes/ . ” (To put the output files in one jar file)

12.Now you must have the .jar file inside WordCountTutorial folder

13.hadoop jar ‘ copy path for firstTutorial.jar file/document’ WordCount /WordCountTutorial/Input /WordCountTutorial/Output …………..(Run the jar file on Hadoop)

14.hadoop dfs -cat /WordCountTutorial/Output/\* …………(for Output)

15. Congratulations your project is completed.