JAVA PROGRAM

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);

}

}

COMMANDS

hadoop version

2 javac -version

3 export HADOOP\_CLASSPATH=$(hadoop classpath)

4 echo $HADOOP\_CLASSPATH

5 hadoop fs -mkdir /wordCountTutorial

6 start-all.sh

7

8 hadoop fs -mkdir hdfs://localhost:9000/wordCountTutorial

9 hadoop fs -mkdir hdfs://localhost:9000/wordCountTutorial/input

10 hadoop fs -put '/home/hadoop/input\_data/input.txt' /wordCountTutorial/input

11 ls

12 javac -classpath ${HADOOP\_CLASSPATH} -d '/home/hadoop/tutorial\_classes' '/home/hadoop/WordCount.java'

13 jar -cvf firstTutorial.jar -C tutorial\_classes/ .

14 hadoop jar '/home/hadoop/firstTutorial.jar' WordCount /wordCountTutorial/input /wordCountTutorial/output

15 hadoop dfs -cat /wordCountTutorial/output/\*

16 hadoop fs -cat hdfs://localhost:9000/wordCountTutorial/output/\*

17 hadoop fs -cat hdfs://localhost:9000/wordCountTutorial/output

18 hadoop dfs -cat /wordCountTutorial/output/\*

19 hadoop hdfs -cat /wordCountTutorial/output/\*

20 hadoop dfs -ls /wordCountTutorial/output/\*

21 hadoop dfs -cat /wordCountTutorial/output/\_SUCCESS

22 hadoop dfs -cat /wordCountTutorial/output/part-r-00000



