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);

}

}

Assuming environment variables are set as follows:

export JAVA\_HOME=/usr/java/default

export PATH=${JAVA\_HOME}/bin:${PATH}

export HADOOP\_CLASSPATH=${JAVA\_HOME}/lib/tools.jar

Compile WordCount.java and create a jar:

**$ bin/hadoop com.sun.tools.javac.Main WordCount.java**

**$ jar cf wc.jar WordCount\*.class**

Assuming that:

* /user/joe/wordcount/input - input directory in HDFS
* /user/joe/wordcount/output - output directory in HDFS

Sample text-files as input:

**$ bin/hadoop fs -ls /user/joe/wordcount/input/**

**/user/joe/wordcount/input/file01**

**/user/joe/wordcount/input/file02**

**$ bin/hadoop fs -cat /user/joe/wordcount/input/file01**

**Hello World Bye World**

**$ bin/hadoop fs -cat /user/joe/wordcount/input/file02**

**Hello Hadoop Goodbye Hadoop**

Run the application:

**$ bin/hadoop jar wc.jar WordCount /user/joe/wordcount/input /user/joe/wordcount/output**

Output:

**$ bin/hadoop fs -cat /user/joe/wordcount/output/part-r-00000**

Bye 1

Goodbye 1

Hadoop 2

Hello 2

World 2