WordCountDriver

import java.io.IOException;

import java.net.URI;

import java.net.URISyntaxException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

//import org.apache.hadoop.mapred.FileInputFormat;

//import org.apache.hadoop.mapred.FileInputFormat;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCountDriver {

public static void main(String[] args) throws IOException, ClassNotFoundException, InterruptedException, URISyntaxException

{

Configuration conf = new Configuration();

//FileSystem fs = FileSystem.get(conf);

//System.out.println(fs.getUri());

Job j = new Job();// getConf(), "Max temperature");

j.setJobName("My First Job");

j.setJarByClass(WordCountDriver.class );

j.setMapperClass(WordCountMapper.class );

//j.setCombinerClass(WordCountReducer.class);

//j.setPartitionerClass(WordCountPartitioner.class);

j.setReducerClass(WordCountReducer.class);

//j.setNumReduceTasks(0);

//j.setCombinerClass(WordCountReducer.class);

//j.setCombinerClass(WordCountReducer.class);

//

//j.setNumReduceTasks(0);

j.setOutputKeyClass(Text.class);

j.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(j, new Path(args[0]));

FileOutputFormat.setOutputPath(j, new Path(args[1]));

URI uri = new URI(args[1].toString());

FileSystem fs = FileSystem.get(uri, conf);

boolean x = fs.delete(new Path(uri),true);

int xxx = j.waitForCompletion(true) ? 0 : 1;

}

}

WordCountMapper

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.DoubleWritable;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class WordCountMapper extends Mapper<LongWritable, Text, Text, IntWritable>

{

@Override

protected void map(LongWritable key, Text value,

org.apache.hadoop.mapreduce.Mapper.Context context)

throws IOException, InterruptedException {

String inputstring = value.toString();

//StringTokenizer stk = new StringTokenizer(inputstring);

//while(stk.hasMoreTokens())

//{

// context.write(new Text(stk.nextToken()),new IntWritable(1));

//}

//for(int i =0; i<inputstring.split(" ").length;i++)

//{

// context.write(new Text(inputstring.split(" ")[i]),new IntWritable(1));

for(String x : inputstring.split(" "))

//how are you

//[0] how

//[1] are

//[2] you

{

context.write(new Text(x),new IntWritable(1));

// how 1

//are 1

//you 1

}

}

}

WordCountReducer

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.Reducer.Context;

public class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable>{

@Override

public void reduce(Text key, Iterable<IntWritable> values, Context context)

throws IOException, InterruptedException {

int y = 0;

for(IntWritable x : values)

{

y ++;

}

context.write(key, new IntWritable(y) );

}

}