Reducer

import java.io.IOException;

import org.apache.hadoop.io.FloatWritable;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

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

@Override

protected void reduce(Text key, Iterable<IntWritable> values,

Reducer<Text, IntWritable, Text, IntWritable>.Context context)

throws IOException, InterruptedException {

int sum=0;

for (IntWritable val : values){

sum+=val.get();

}

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

}

}

==============================

Mapper

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class MyMap extends Mapper<IntWritable, Text, Text, IntWritable> {

@Override

protected void map(IntWritable key, Text value,

Mapper<IntWritable, Text, Text, IntWritable>.Context context)

throws IOException, InterruptedException {

String line = value.toString();

StringTokenizer str = new StringTokenizer(line, ",");

while(str.hasMoreTokens()){

String word = str.nextToken();

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

}

}

}

===================================

import java.io.IOException;

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.SequenceFile;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.JobClient;

import org.apache.hadoop.mapreduce.Job;

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

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

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

public class Driver {

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

Job job = new Job();

job.setJarByClass(Driver.class);

job.setJobName("MyfirstHadopp Program");

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

job.setMapperClass(MyMap.class);

job.setReducerClass(MyReduce.class);

Path outputdir = new Path(args[1]);

FileSystem fs = FileSystem.get(new Configuration());

fs.delete(outputdir, true );

job.setInputFormatClass(SequenceFileInputFormat.class);

FileInputFormat.setInputPaths(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job,outputdir);

job.waitForCompletion(true);

}

}