|  |
| --- |
|  |
| import java.io.\*; |
|  | | import org.apache.hadoop.io.DoubleWritable; |
|  | | import org.apache.hadoop.io.Text; |
|  | | import org.apache.hadoop.io.LongWritable; |
|  | | import org.apache.hadoop.mapreduce.Job; |
|  | | import org.apache.hadoop.mapreduce.Mapper; |
|  | | import org.apache.hadoop.mapreduce.Reducer; |
|  | | import org.apache.hadoop.conf.\*; |
|  | | import org.apache.hadoop.fs.\*; |
|  | | import org.apache.hadoop.mapreduce.lib.input.\*; |
|  | | import org.apache.hadoop.mapreduce.lib.output.\*; |
|  | |  |
|  | |  |
|  | | public class vehicle { |
|  | |  |
|  | | public static class MapClass extends Mapper<LongWritable,Text,Text, DoubleWritable> |
|  | | { |
|  | | public void map(LongWritable key, Text value, Context context)throws IOException, InterruptedException |
|  | | { |
|  | | try{ |
|  | | String[] str = value.toString().split(","); |
|  | | long val = Long.parseLong(str[1]); |
|  | | context.write(new Text(str[0]),new DoubleWritable(val)); |
|  | |  |
|  | | } |
|  | | catch(Exception e) |
|  | | { |
|  | | System.out.println(e.getMessage()); |
|  | | } |
|  | | } |
|  | | } |
|  | |  |
|  | | public static class ReduceClass extends Reducer<Text,DoubleWritable,Text,Text> |
|  | | { |
|  | | //private FloatWritable result = new FloatWritable(); |
|  | |  |
|  | | public void reduce(Text key, Iterable<DoubleWritable> values,Context context) throws IOException, InterruptedException { |
|  | | int totcount = 0, offencount=0, per=0; |
|  | |  |
|  | | for (DoubleWritable count : values) |
|  | | { |
|  | | totcount++; |
|  | | if(count.get() >65) |
|  | | { |
|  | | offencount++; |
|  | | } |
|  | |  |
|  | | } |
|  | | per= ((offencount\*100)/totcount); |
|  | | String percentvalue= String.format("%d", per); |
|  | | String valwithsign= percentvalue + "%"; |
|  | |  |
|  | | //result.set(per); |
|  | |  |
|  | |  |
|  | |  |
|  | | context.write(key, new Text(valwithsign)); |
|  | |  |
|  | | } |
|  | | } |
|  | | public static void main(String[] args) throws Exception { |
|  | | Configuration conf = new Configuration(); |
|  | | //conf.set("name", "value") |
|  | | Job job = Job.getInstance(conf, "Vehicle"); |
|  | | job.setJarByClass(vehicle.class); |
|  | | job.setMapperClass(MapClass.class); |
|  | | //job.setCombinerClass(ReduceClass.class); |
|  | | job.setReducerClass(ReduceClass.class); |
|  | | //job.setNumReduceTasks(2); |
|  | | job.setMapOutputKeyClass(Text.class); |
|  | | job.setMapOutputValueClass(DoubleWritable.class); |
|  | | job.setOutputKeyClass(Text.class); |
|  | | job.setOutputValueClass(Text.class); |
|  | | FileInputFormat.addInputPath(job, new Path(args[0])); |
|  | | FileOutputFormat.setOutputPath(job, new Path(args[1])); |
|  | | System.exit(job.waitForCompletion(true) ? 0 : 1); |
|  | | } |
|  | | } |