|  |
| --- |
|  |
| |  | | --- | | import java.io.\*; | |  | import org.apache.hadoop.io.FloatWritable; | |  | 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 average { | |  |  | |  | public static class MapClass extends Mapper<LongWritable,Text,Text, FloatWritable> | |  | { | |  | private final static FloatWritable score= new FloatWritable(); | |  |  | |  | public void map(LongWritable key, Text value, Context context) | |  | throws IOException, InterruptedException { | |  | double myscore=0.00; | |  |  | |  | try { | |  | String[] str = value.toString().split(","); | |  | myscore= Float.parseFloat(str[2]); | |  | score.set((float)myscore); | |  | context.write(new Text(str[3]),score); | |  |  | |  | } | |  | catch(Exception e) | |  | { | |  | System.out.println(e.getMessage()); | |  | } | |  |  | |  | } | |  | } | |  |  | |  |  | |  |  | |  | public static class ReduceClass extends Reducer<Text,FloatWritable,Text,FloatWritable> | |  | { | |  |  | |  | private FloatWritable result = new FloatWritable(); | |  |  | |  |  | |  | public void reduce(Text key, Iterable<FloatWritable> values,Context context) | |  | throws IOException, InterruptedException { | |  | double sum =0.00, total=0.00; | |  | double avg=0.00; | |  | for(FloatWritable val:values) | |  | { | |  | sum += (double)val.get(); | |  | total++; | |  | } | |  | avg= sum/total; | |  | result.set((float) avg); | |  | context.write(key,result); | |  | } | |  | } | |  |  | |  | public static void main(String[] args) throws Exception { | |  | Configuration conf = new Configuration(); | |  | //conf.set("name", "value") | |  | Job job = Job.getInstance(conf, "average"); | |  | job.setJarByClass(average.class); | |  | job.setMapperClass(MapClass.class); | |  | //job.setCombinerClass(ReduceClass.class); | |  | job.setReducerClass(ReduceClass.class); | |  | //job.setNumReduceTasks(2); | |  | job.setMapOutputKeyClass(Text.class); | |  | job.setMapOutputValueClass(FloatWritable.class); | |  | job.setOutputKeyClass(Text.class); | |  | job.setOutputValueClass(FloatWritable.class); | |  | FileInputFormat.addInputPath(job, new Path(args[0])); | |  | FileOutputFormat.setOutputPath(job, new Path(args[1])); | |  | System.exit(job.waitForCompletion(true) ? 0 : 1); | |  | } | |  | } | |  |  | |  |  | |  |  | |  |  | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |