|  |
| --- |
|  |
| import java.io.\*; |
|  | import java.util.\*; |
|  | import org.apache.hadoop.io.IntWritable; |
|  | 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 wordcount { |
|  |  |
|  | public static class MapClass extends Mapper<LongWritable,Text,Text,IntWritable> |
|  | { |
|  | private final static IntWritable one= new IntWritable(1); |
|  | private Text word = new Text(); |
|  | public void map(LongWritable 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 ReduceClass 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(); |
|  | conf.set("mapreduce.output.textoutputformat.seperator"," ,"); |
|  | Job job = Job.getInstance(conf, "Word Count"); |
|  | job.setJarByClass(wordcount.class); |
|  | job.setMapperClass(MapClass.class); |
|  | //job.setCombinerClass(ReduceClass.class); |
|  | job.setReducerClass(ReduceClass.class); |
|  | //job.setNumReduceTasks(2); |
|  | job.setMapOutputKeyClass(Text.class); |
|  | job.setMapOutputValueClass(IntWritable.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); |
|  | } |
|  | } |
|  |  |