|  |
| --- |
| package employee; |
|  |  |
|  | import java.io.IOException; |
|  | import java.util.\*; |
|  | import org.apache.hadoop.fs.Path; |
|  | import org.apache.hadoop.conf.\*; |
|  | import org.apache.hadoop.io.\*; |
|  | import org.apache.hadoop.mapred.\*; |
|  | import org.apache.hadoop.util.\*; |
|  |  |
|  | public class Program2 { |
|  |  |
|  | //MAPPER CODE |
|  |  |
|  | public static class Map extends MapReduceBase implements 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, OutputCollector < Text, IntWritable > output, Reporter reporter) throws IOException { |
|  | String myvalue = value.toString(); |
|  | String[] tokens = myvalue.split(","); |
|  |  |
|  | output.collect(new Text("Cumulative years of service done for the company by all the employees: "), |
|  | new IntWritable(Integer.parseInt(tokens[1))); |
|  | } |
|  | } |
|  |  |
|  | //REDUCER CODE |
|  | public static class Reduce extends MapReduceBase implements Reducer < Text, IntWritable, Text, IntWritable > { |
|  | public void reduce(Text key, Iterator < IntWritable > values, OutputCollector < Text, IntWritable > output, Reporter reporter) throws IOException { //{little: {1,1}} |
|  | int count = 0 ; |
|  | while(values.hasNext()) { |
|  | count += values.next().get(); |
|  | } |
|  | output.collect(key, new IntWritable(count)); |
|  |  |
|  | } |
|  | } |
|  |  |
|  | //DRIVER CODE |
|  | public static void main(String[] args) throws Exception { |
|  | JobConf conf = new JobConf(Prog2.class); |
|  | conf.setJobName("Prog2"); |
|  | conf.setOutputKeyClass(Text.class); |
|  | conf.setOutputValueClass(IntWritable.class); |
|  | conf.setMapperClass(Map.class); |
|  | conf.setCombinerClass(Reduce.class); |
|  | conf.setReducerClass(Reduce.class); |
|  | conf.setInputFormat(TextInputFormat.class); |
|  | conf.setOutputFormat(TextOutputFormat.class); |
|  | FileInputFormat.setInputPaths(conf, new Path(args[0])); |
|  | FileOutputFormat.setOutputPath(conf, new Path(args[1])); |
|  | JobClient.runJob(conf); |
|  | } |
|  | } |