|  |  |
| --- | --- |
|  | |
| import java.io.BufferedReader; |
|  | import java.io.FileReader; |
|  | import java.io.IOException; |
|  | import java.net.URI; |
|  | import java.util.HashMap; |
|  | import java.util.Map; |
|  |  |
|  | import org.apache.hadoop.conf.Configuration; |
|  | import org.apache.hadoop.fs.Path; |
|  | import org.apache.hadoop.io.LongWritable; |
|  | import org.apache.hadoop.io.Text; |
|  | import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  | import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; |
|  | import org.apache.hadoop.mapreduce.Job; |
|  | import org.apache.hadoop.mapreduce.Mapper; |
|  |  |
|  | public class MyDC { |
|  |  |
|  |  |
|  | public static class MyMapper extends Mapper<LongWritable,Text, Text, Text> { |
|  |  |
|  |  |
|  | private Map<String, String> abMap = new HashMap<String, String>(); |
|  | private Map<String, String> abMap1 = new HashMap<String, String>(); |
|  | private Text outputKey = new Text(); |
|  | private Text outputValue = new Text(); |
|  |  |
|  | protected void setup(Context context) throws java.io.IOException, InterruptedException{ |
|  |  |
|  | super.setup(context); |
|  |  |
|  | URI[] files = context.getCacheFiles(); // getCacheFiles returns null |
|  |  |
|  | Path p = new Path(files[0]); |
|  |  |
|  | Path p1 = new Path(files[1]); |
|  |  |
|  | if (p.getName().equals("salary.txt")) { |
|  | BufferedReader reader = new BufferedReader(new FileReader(p.toString())); |
|  | String line = reader.readLine(); |
|  | while(line != null) { |
|  | String[] tokens = line.split(","); |
|  | String emp\_id = tokens[0]; |
|  | String emp\_sal = tokens[1]; |
|  | abMap.put(emp\_id, emp\_sal); |
|  | line = reader.readLine(); |
|  | } |
|  | reader.close(); |
|  | } |
|  | if (p1.getName().equals("desig.txt")) { |
|  | BufferedReader reader = new BufferedReader(new FileReader(p1.toString())); |
|  | String line = reader.readLine(); |
|  | while(line != null) { |
|  | String[] tokens = line.split(","); |
|  | String emp\_id = tokens[0]; |
|  | String emp\_desig = tokens[1]; |
|  | abMap1.put(emp\_id, emp\_desig); |
|  | line = reader.readLine(); |
|  | } |
|  | reader.close(); |
|  | } |
|  |  |
|  |  |
|  | if (abMap.isEmpty()) { |
|  | throw new IOException("MyError:Unable to load salary data."); |
|  | } |
|  |  |
|  | if (abMap1.isEmpty()) { |
|  | throw new IOException("MyError:Unable to load designation data."); |
|  | } |
|  |  |
|  | } |
|  |  |
|  |  |
|  | protected void map(LongWritable key, Text value, Context context) |
|  | throws java.io.IOException, InterruptedException { |
|  |  |
|  |  |
|  | String row = value.toString(); |
|  | String[] tokens = row.split(","); |
|  | String emp\_id = tokens[0]; |
|  | String salary = abMap.get(emp\_id); |
|  | String desig = abMap1.get(emp\_id); |
|  | String sal\_desig = salary + "," + desig; |
|  | outputKey.set(row); |
|  | outputValue.set(sal\_desig); |
|  | context.write(outputKey,outputValue); |
|  | } |
|  | } |
|  |  |
|  |  |
|  | public static void main(String[] args) |
|  | throws IOException, ClassNotFoundException, InterruptedException { |
|  |  |
|  | Configuration conf = new Configuration(); |
|  | conf.set("mapreduce.output.textoutputformat.separator", ","); |
|  | Job job = Job.getInstance(conf); |
|  | job.setJarByClass(MyDC.class); |
|  | job.setJobName("Map Side Join"); |
|  | job.setMapperClass(MyMapper.class); |
|  | job.addCacheFile(new Path("salary.txt").toUri()); |
|  | job.addCacheFile(new Path("desig.txt").toUri()); |
|  | job.setNumReduceTasks(0); |
|  | job.setMapOutputKeyClass(Text.class); |
|  | job.setMapOutputValueClass(Text.class); |
|  |  |
|  | FileInputFormat.addInputPath(job, new Path(args[0])); |
|  | FileOutputFormat.setOutputPath(job, new Path(args[1])); |
|  |  |
|  | job.waitForCompletion(true); |
|  |  |
|  |  |
|  | } |
|  | } |