|  |  |
| --- | --- |
|  | |
| |  | | --- | | import java.io.BufferedReader; | |  | import java.io.FileReader; | |  | import java.io.IOException; | |  | import java.util.HashMap; | |  | import java.util.Map; | |  | import java.net.URI; | |  |  | |  | 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 totalqty\_statewise { | |  |  | |  |  | |  | public static class MyMapper extends Mapper<LongWritable,Text, Text, Text> | |  | { | |  |  | |  |  | |  | private Map<String, String> abMap = 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 =DistributedCache.getCacheFiles(context.getConfiguration()); | |  | URI[] files = context.getCacheFiles(); // getCacheFiles returns null | |  |  | |  | Path p = new Path(files[0]); | |  |  | |  |  | |  | if (p.getName().equals("store\_master.txt")) { | |  | BufferedReader reader = new BufferedReader(new FileReader(p.toString())); | |  | String line = reader.readLine(); | |  | while(line != null) { | |  | String[] tokens = line.split(","); | |  | String state = tokens[2]; | |  | String store\_id=tokens[0]; | |  | abMap.put(store\_id,state); | |  | line = reader.readLine(); | |  | } | |  | reader.close(); | |  | } | |  |  | |  |  | |  | if (abMap.isEmpty()) { | |  | throw new IOException("MyError:Unable to load store\_master- data."); | |  | } | |  |  | |  | } | |  |  | |  |  | |  | protected void map(LongWritable key, Text value, Context context) | |  | throws java.io.IOException, InterruptedException { | |  |  | |  |  | |  | String row = value.toString(); | |  | String[] tokens = row.split(","); | |  | String storeid = tokens[0]; | |  | String state = abMap.get(storeid); | |  | outputKey.set(row); | |  | outputValue.set(state); | |  | 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(totalqty\_statewise.class); | |  | job.setJobName("Map Side Join of POS1 and POS2 ans master\_data"); | |  | job.setMapperClass(MyMapper.class); | |  | job.addCacheFile(new Path("store\_master.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); | |  |  | |  |  | |  | } | |  | } | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |