|  |  |
| --- | --- |
|  | |
| |  | | --- | | import java.io.\*; | |  |  | |  | import org.apache.hadoop.io.\*; | |  | import org.apache.hadoop.mapreduce.\*; | |  | import org.apache.hadoop.conf.\*; | |  |  | |  | import org.apache.hadoop.fs.\*; | |  |  | |  | import org.apache.hadoop.mapreduce.lib.input.\*; | |  | import org.apache.hadoop.mapreduce.lib.output.\*; | |  |  | |  | import org.apache.hadoop.util.\*; | |  |  | |  | public class POS\_partition extends Configured implements Tool | |  | { | |  | //Map class | |  |  | |  | public static class MapClass extends Mapper<LongWritable,Text,Text,Text> | |  | { | |  | public void map(LongWritable key, Text value, Context context) | |  | { | |  | try{ | |  | String[] str = value.toString().split(","); | |  | String itemid=str[1]; | |  | String qty=str[2]; | |  | String state=str[4]; | |  | String myrow = qty + "," + state; | |  | context.write(new Text(itemid), new Text(myrow)); | |  | } | |  | catch(Exception e) | |  | { | |  | System.out.println(e.getMessage()); | |  | } | |  | } | |  | } | |  |  | |  | //Reducer class | |  |  | |  | public static class ReduceClass extends Reducer<Text,Text,Text,IntWritable> | |  | { | |  | private Text outputKey = new Text(); | |  | private IntWritable result = new IntWritable(); | |  |  | |  | public void reduce(Text key, Iterable <Text> values, Context context) throws IOException, InterruptedException | |  | { | |  |  | |  | int sum = 0; | |  | String myState = ""; | |  | for (Text val : values) { | |  |  | |  | String [] str = val.toString().split(","); | |  | sum += Integer.parseInt(str[0]); | |  | myState = str[1]; | |  | } | |  | String mykey = myState + ',' + key.toString(); | |  | outputKey.set(mykey); | |  | result.set(sum); | |  | context.write(outputKey, result); | |  | } | |  | } | |  |  | |  | //Partitioner class | |  |  | |  | public static class CaderPartitioner extends | |  | Partitioner < Text, Text > | |  | { | |  | @Override | |  | public int getPartition(Text key, Text value, int numReduceTasks) | |  | { | |  | String[] str = value.toString().split(","); | |  | if(str[1].equals("MAH")) | |  | { | |  | return 0; | |  | } | |  | else | |  | { | |  | return 1 ; | |  | } | |  | } | |  | } | |  |  | |  |  | |  | public int run(String[] arg) throws Exception | |  | { | |  |  | |  |  | |  | Configuration conf = new Configuration(); | |  | Job job = Job.getInstance(conf); | |  | job.setJarByClass(POS\_partition.class); | |  | job.setJobName("State Wise Item Qty sales"); | |  | FileInputFormat.setInputPaths(job, new Path(arg[0])); | |  | FileOutputFormat.setOutputPath(job,new Path(arg[1])); | |  |  | |  | job.setMapperClass(MapClass.class); | |  |  | |  | job.setMapOutputKeyClass(Text.class); | |  | job.setMapOutputValueClass(Text.class); | |  |  | |  | //set partitioner statement | |  |  | |  | job.setPartitionerClass(CaderPartitioner.class); | |  | job.setReducerClass(ReduceClass.class); | |  | job.setNumReduceTasks(2); | |  | job.setInputFormatClass(TextInputFormat.class); | |  |  | |  | job.setOutputFormatClass(TextOutputFormat.class); | |  | job.setOutputKeyClass(Text.class); | |  | job.setOutputValueClass(IntWritable.class); | |  |  | |  | System.exit(job.waitForCompletion(true)? 0 : 1); | |  | return 0; | |  | } | |  |  | |  | public static void main(String ar[]) throws Exception | |  | { | |  | ToolRunner.run(new Configuration(), new POS\_partition(),ar); | |  | System.exit(0); | |  | } | |  | } | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |