|  |  |
| --- | --- |
|  | |
| import java.io.IOException; | |
|  | import java.util.TreeMap; |
|  | import org.apache.hadoop.conf.Configuration; |
|  | import org.apache.hadoop.fs.Path; |
|  | import org.apache.hadoop.io.LongWritable; |
|  | import org.apache.hadoop.io.NullWritable; |
|  | import org.apache.hadoop.io.Text; |
|  | import org.apache.hadoop.mapreduce.Job; |
|  | import org.apache.hadoop.mapreduce.Mapper; |
|  | import org.apache.hadoop.mapreduce.Reducer; |
|  | import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  | import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; |
|  |  |
|  |  |
|  | public class top5 { |
|  | public static class Top5Mapper extends |
|  | Mapper<LongWritable, Text, NullWritable, Text> { |
|  | private TreeMap<Integer, Text> repToRecordMap = new TreeMap<Integer, Text>(); |
|  |  |
|  | public void map(LongWritable key, Text value, Context context |
|  | ) throws IOException, InterruptedException { |
|  | String record = value.toString(); |
|  | String[] parts = record.split(","); |
|  | int myKey = Integer.parseInt(parts[2]); |
|  | repToRecordMap.put(myKey, new Text(value)); |
|  | if (repToRecordMap.size() > 5) { |
|  | repToRecordMap.remove(repToRecordMap.firstKey()); |
|  | } |
|  | } |
|  | protected void cleanup(Context context) throws IOException, |
|  | InterruptedException { |
|  | // Output our 5 records to the reducers with a null key |
|  | for (Text t : repToRecordMap.values()) { |
|  | context.write(NullWritable.get(), t); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | public static class Top5Reducer extends |
|  | Reducer<NullWritable, Text, NullWritable, Text> { |
|  | private TreeMap<Integer, Text> repToRecordMap = new TreeMap<Integer, Text>(); |
|  |  |
|  | public void reduce(NullWritable key, Iterable<Text> values, |
|  | Context context) throws IOException, InterruptedException { |
|  | for (Text value : values) { |
|  | String record = value.toString(); |
|  | String[] parts = record.split(","); |
|  | int myKey = Integer.parseInt(parts[2]); |
|  | repToRecordMap.put(myKey, new Text(value)); |
|  | if (repToRecordMap.size() > 5) { |
|  | repToRecordMap.remove(repToRecordMap.firstKey()); |
|  | } |
|  | } |
|  | for (Text t : repToRecordMap.descendingMap().values()) { |
|  | // Output our five records to the file system with a null key |
|  | context.write(NullWritable.get(), t); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | public static void main(String[] args) throws Exception { |
|  |  |
|  | Configuration conf = new Configuration(); |
|  | Job job = Job.getInstance(conf, "Top 5 Records"); |
|  | job.setJarByClass(top5.class); |
|  | job.setMapperClass(Top5Mapper.class); |
|  | job.setReducerClass(Top5Reducer.class); |
|  | job.setOutputKeyClass(NullWritable.class); |
|  | job.setOutputValueClass(Text.class); |
|  | FileInputFormat.addInputPath(job, new Path(args[0])); |
|  | FileOutputFormat.setOutputPath(job, new Path(args[1])); |
|  | System.exit(job.waitForCompletion(true) ? 0 : 1); |
|  | } |
|  | } |