|  |  |
| --- | --- |
|  | |
|  | |  |
|  | import java.io.\*; |
|  |  | import org.apache.hadoop.io.IntWritable; |
|  |  | import org.apache.hadoop.io.Text; |
|  |  | import org.apache.hadoop.io.LongWritable; |
|  |  | import org.apache.hadoop.mapreduce.Job; |
|  |  | import org.apache.hadoop.mapreduce.Mapper; |
|  |  | import org.apache.hadoop.mapreduce.Reducer; |
|  |  | import org.apache.hadoop.conf.\*; |
|  |  | import org.apache.hadoop.fs.\*; |
|  |  | import org.apache.hadoop.mapreduce.lib.input.\*; |
|  |  | import org.apache.hadoop.mapreduce.lib.output.\*; |
|  |  |  |
|  |  |  |
|  |  | public class stringsearch { |
|  |  |  |
|  |  | public static class MapClass extends Mapper<LongWritable,Text,Text,IntWritable> |
|  |  | { |
|  |  | private final static IntWritable one= new IntWritable(1); |
|  |  | private Text sentence = new Text(); |
|  |  | public void map(LongWritable key, Text value, Context context) |
|  |  | throws IOException,InterruptedException { |
|  |  | String mysearchtext= context.getConfiguration().get("mytext"); |
|  |  | String line= value.toString(); |
|  |  | if(mysearchtext !=null) |
|  |  | { |
|  |  | if(line.contains(mysearchtext)) |
|  |  | { |
|  |  | sentence.set(line); |
|  |  |  |
|  |  | context.write(sentence, one); |
|  |  | } |
|  |  | } |
|  |  | } |
|  |  | } |
|  |  | public static class ReduceClass extends Reducer<Text,IntWritable,Text, IntWritable> |
|  |  | { |
|  |  | private IntWritable result = new IntWritable(); |
|  |  |  |
|  |  | public void reduce(Text key, Iterable<IntWritable> values,Context context) |
|  |  | throws IOException, InterruptedException { |
|  |  | int sum = 0; |
|  |  |  |
|  |  | for (IntWritable val : values) |
|  |  | { |
|  |  | sum += val.get(); |
|  |  |  |
|  |  | } |
|  |  |  |
|  |  | result.set(sum); |
|  |  | context.write(key, result); |
|  |  |  |
|  |  | } |
|  |  | } |
|  |  | public static void main(String[] args) throws Exception { |
|  |  | Configuration conf = new Configuration(); |
|  |  | if(args.length > 2) |
|  |  | { |
|  |  | conf.set("mytext", args[2]); |
|  |  | } |
|  |  | Job job = Job.getInstance(conf,"stringsearch"); |
|  |  | job.setJarByClass(stringsearch.class); |
|  |  | job.setMapperClass(MapClass.class); |
|  |  | //job.setCombinerClass(ReduceClass.class); |
|  |  | job.setReducerClass(ReduceClass.class); |
|  |  | //job.setNumReduceTasks(2); |
|  |  | job.setMapOutputKeyClass(Text.class); |
|  |  | job.setMapOutputValueClass(IntWritable.class); |
|  |  | job.setOutputKeyClass(Text.class); |
|  |  | job.setOutputValueClass(IntWritable.class); |
|  |  | FileInputFormat.addInputPath(job, new Path(args[0])); |
|  |  | FileOutputFormat.setOutputPath(job, new Path(args[1])); |
|  |  | System.exit(job.waitForCompletion(true) ? 0 : 1); |
|  |  | } |
|  |  | } |
|  |  |  |