1. Program one: Counting words.
2. **import** java.io.IOException;
3. **import** java.util.StringTokenizer;
5. **import** org.apache.hadoop.conf.Configuration;
6. **import** org.apache.hadoop.fs.Path;
7. **import** org.apache.hadoop.io.IntWritable;
8. **import** org.apache.hadoop.io.Text;
9. **import** org.apache.hadoop.mapreduce.Job;
10. **import** org.apache.hadoop.mapreduce.Mapper;
11. **import** org.apache.hadoop.mapreduce.Reducer;
12. **import** org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
13. **import** org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
15. **public** **class** WordCount {
17. **public** **static** **class** TokenizerMapper
18. **extends** Mapper<Object, Text, Text, IntWritable>{
20. **private** **final** **static** IntWritable one = **new** IntWritable(1);
21. **private** Text word = **new** Text();
23. **public** **void** map(Object key, Text value, Context context
24. ) **throws** IOException, InterruptedException {
25. StringTokenizer itr = **new** StringTokenizer(value.toString());
26. **while** (itr.hasMoreTokens()) {
27. word.set(itr.nextToken());
28. context.write(word, one);
29. }
30. }
31. }
33. **public** **static** **class** IntSumReducer
34. **extends** Reducer<Text,IntWritable,Text,IntWritable> {
35. **private** IntWritable result = **new** IntWritable();
37. **public** **void** reduce(Text key, Iterable<IntWritable> values,
38. Context context
39. ) **throws** IOException, InterruptedException {
40. **int** sum = 0;
41. **for** (IntWritable val : values) {
42. sum += val.get();
43. }
44. result.set(sum);
45. context.write(key, result);
46. }
47. }
49. **public** **static** **void** main(String[] args) **throws** Exception {
50. Configuration conf = **new** Configuration();
51. Job job = Job.getInstance(conf, "word count");
52. job.setJarByClass(WordCount.**class**);
53. job.setMapperClass(TokenizerMapper.**class**);
54. job.setCombinerClass(IntSumReducer.**class**);
55. job.setReducerClass(IntSumReducer.**class**);
56. job.setOutputKeyClass(Text.**class**);
57. job.setOutputValueClass(IntWritable.**class**);
58. FileInputFormat.addInputPath(job, **new** Path(args[0]));
59. FileOutputFormat.setOutputPath(job, **new** Path(args[1]));
60. System.exit(job.waitForCompletion(**true**) ? 0 : 1);
61. }
62. }