Hadoop map reduce example

这个例子会使用一个example数据集来执行一个简单的分类汇总任务(根据不同的产品名称计算总销售额)，关于如何运行这个例子请参见本地调试

首先我们新建一个样本csv文件，将其保存为`testData.csv`并在运行前拖入项目目录下的`\input`文件夹，这里使用的例子中样本csv如下：

```
productName,area,salesRevenue
hadoop,south china,1000
spark,south china,1000
hadoop,middle china,1000
hadoop,north china,1000
spark,middle china,1000
```

Map reduce的程序主要分为三个部分`Mapper`，`Reducer`和`Driver`(main函数在这里)，分别如下:

Mapper

```
// The following is a map process
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class Mapper extends MapReduceBase implements
org.apache.hadoop.mapred.Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1000);

    public void map(LongWritable key, Text value, OutputCollector<Text,
IntWritable> output, Reporter reporter){

        String valueString = value.toString();
        String[] oneRecord = valueString.split(",");
        try{
            int salesRevenue = Integer.parseInt(oneRecord[2]);  // 3rd column is
sales revenue
            output.collect(new Text(oneRecord[0]), one); // 1st column is the
product name
        } catch (NumberFormatException | IOException e) {
            e.printStackTrace();
        }
    }
}
```

Reducer

```
// The following is the reducer
import java.io.IOException;
```

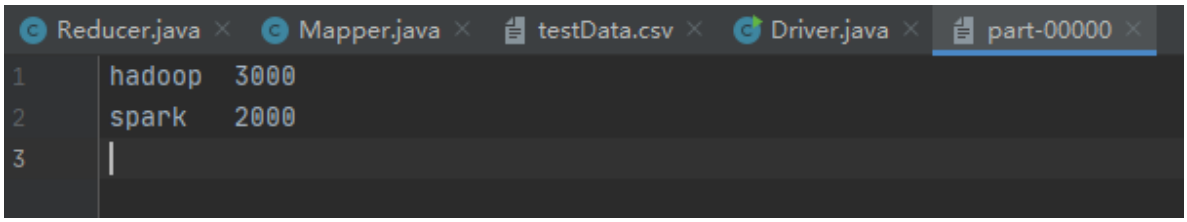
```
import java.util.*;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class Reducer extends MapReduceBase implements
org.apache.hadoop.mapred.Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text t_key, Iterator<IntWritable> values,
OutputCollector<Text,IntWritable> output, Reporter reporter) throws IOException
{
        int oneProductRevenue = 0;
        for (Iterator<IntWritable> it = values; it.hasNext(); ) {
            IntWritable value = it.next();
            oneProductRevenue += value.get();
        }
        output.collect(t_key, new IntWritable(oneProductRevenue));

    }
}
```

Driver

```
// The following is the driver

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;

public class Driver {
    public static void main(String[] args) {
        JobClient myClient = new JobClient();
        // Create a configuration object for the job
        JobConf jobConf = new JobConf(Driver.class);

        // Set a name of the Job
        jobConf.setJobName("ProductSalesRevenue");

        // Specify data type of output key and value
        jobConf.setOutputKeyClass(Text.class);
        jobConf.setOutputValueClass(IntWritable.class);

        // Specify names of Mapper and Reducer Class
        jobConf.setMapperClass(Mapper.class);
        jobConf.setReducerClass(Reducer.class);

        // Specify formats of the data type of Input and output
        jobConf.setInputFormat(TextInputFormat.class);
        jobConf.setOutputFormat(TextOutputFormat.class);

        // Set input and output directories using command line arguments,
        //arg[0] = name of input directory on HDFS, and arg[1] =  name of output
directory to be created to store the output file.
        FileInputFormat.setInputPaths(jobConf, new Path(args[0]));
        FileOutputFormat.setOutputPath(jobConf, new Path(args[1]));
```

```
        myClient.setConf(jobConf);
        try {
            // Run the job
            JobClient.runJob(jobConf);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Result



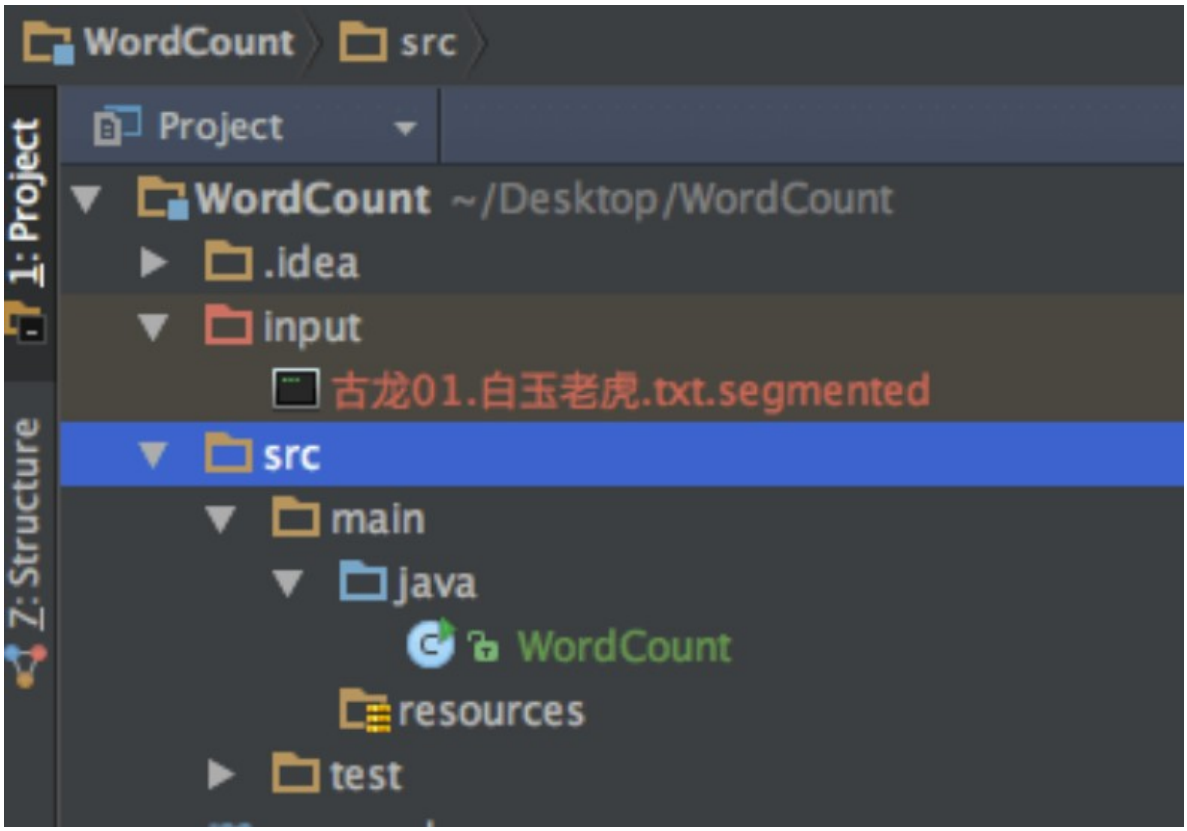
在本地调试map reduce

Material: IntelliJ IDEA.

Note: the screenshot is only for reference!

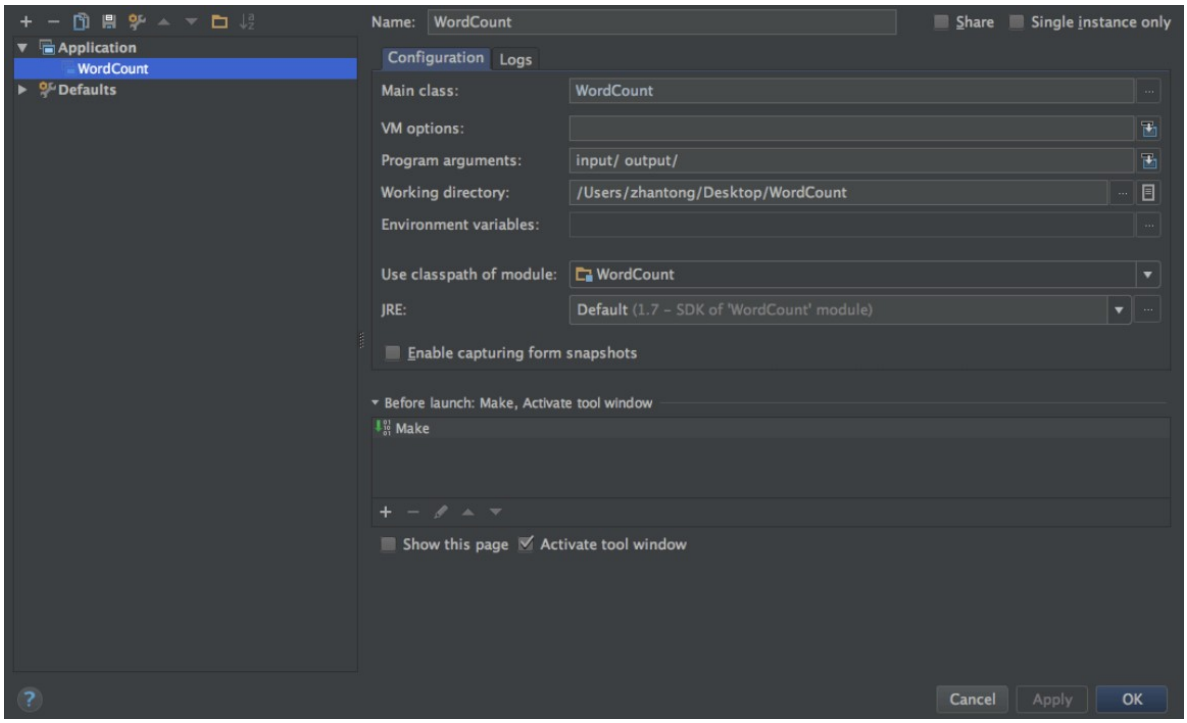
Reference

1. Suppose you have a started a java project, create a folder called `input` in the same level of the `src` folder, put your piece of data into the `input` folder.



2. Select `Run`--`Edit Configurations`, click `+` to add new `Application`, set the `Main class` to be your main class, set `Program arguments` to be `input/ output/`.

   Note: you don't need to create `output` folder since it will be created automatically and you need to clear the content in `output` folder if there is any before running the codes.



3. Once you run the program, the program will create the `output` folder automatically and you can check `part-r-00000` for results directly in the IntelliJ IDEA.