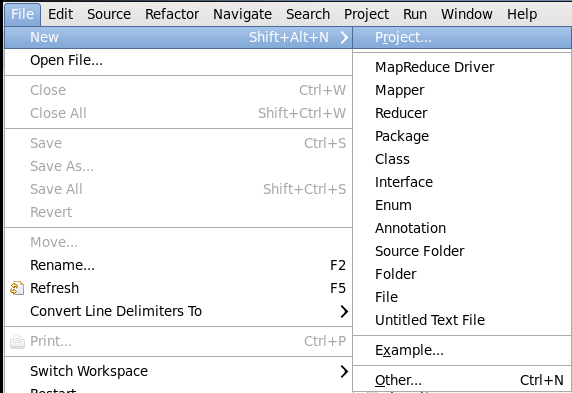
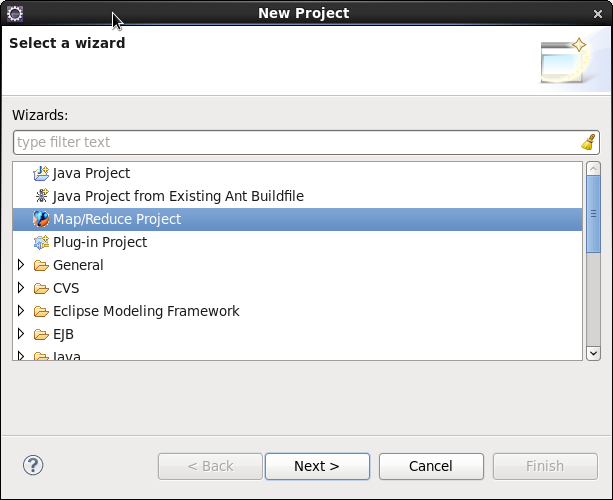
**8. 在Eclipse中运行“Word Count” MapReduce程序**

（1）在 Eclipse 中创建 “WordCount” MapReduce项目

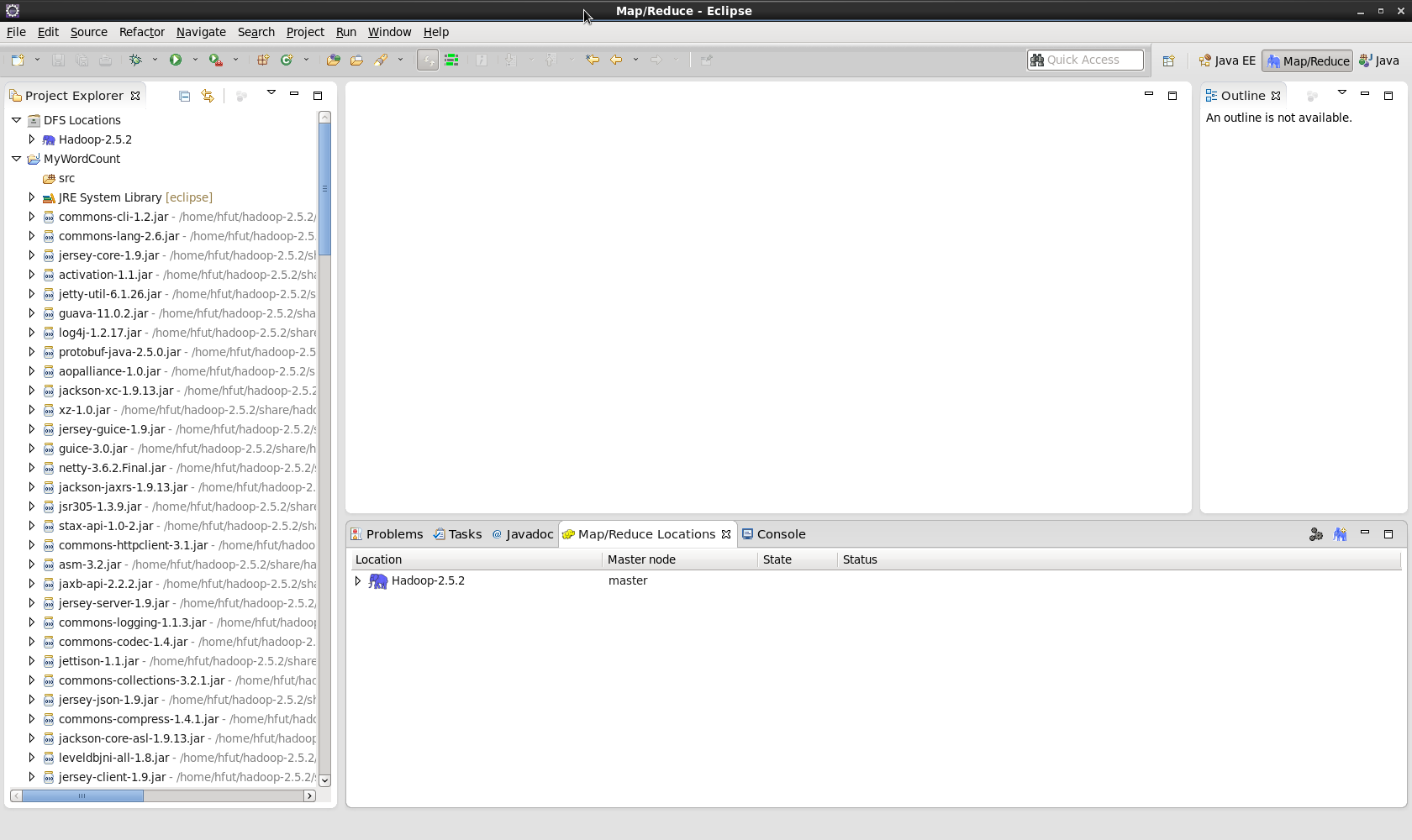
点击 File 菜单，选择 New -> Project…：



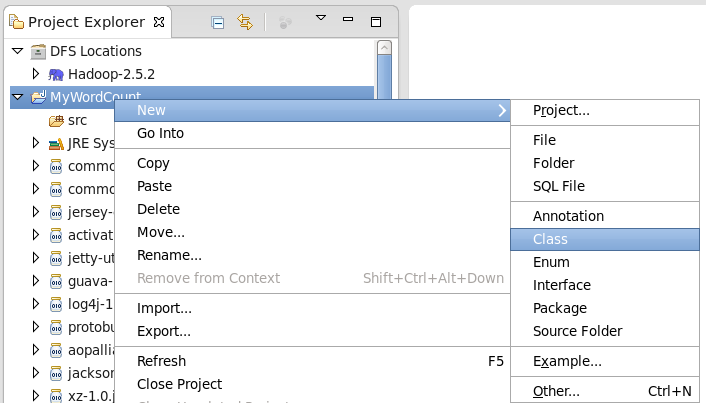
选择 Map/Reduce Project，点击 Next：



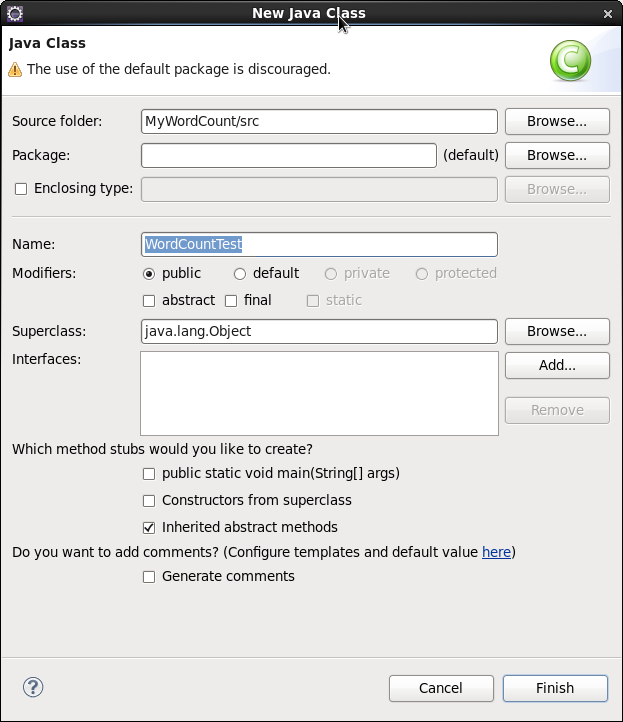
填写Project name 为 MyWordCount，点击Finish创建项目。



右键点击MyWordCount 项目，选择New -> Class：



在 Name 处填写 WordCountTest。



将如下 WordCountTest 的代码复制到该WordCountTest.java中。

import java.io.IOException;

import java.util.Iterator;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

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;

import org.apache.hadoop.util.GenericOptionsParser;

public class WordCountTest {

public WordCountTest() {

}

public static void main(String[] args) throws Exception {

Configuration conf = new Configuration();

String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();

if(otherArgs.length < 2) {

System.err.println("Usage: wordcount <in> [<in>...] <out>");

System.exit(2);

}

Job job = Job.getInstance(conf, "word count test");

job.setJarByClass(WordCountTest.class);

job.setMapperClass(WordCountTest.TokenizerMapper.class);

job.setCombinerClass(WordCountTest.IntSumReducer.class);

job.setReducerClass(WordCountTest.IntSumReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

for(int i = 0; i < otherArgs.length - 1; ++i) {

FileInputFormat.addInputPath(job, new Path(otherArgs[i]));

}

FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));

System.exit(job.waitForCompletion(true)?0:1);

}

public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

private IntWritable result = new IntWritable();

public IntSumReducer() {

}

public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable, Text, IntWritable>.Context context) throws IOException, InterruptedException {

int sum = 0;

IntWritable val;

for(Iterator itr = values.iterator(); itr.hasNext(); sum += val.get()) {

val = (IntWritable)itr.next();

}

this.result.set(sum);

context.write(key, this.result);

}

}

public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {

private static final IntWritable one = new IntWritable(1);

private Text word = new Text();

public TokenizerMapper() {

}

public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context context) throws IOException, InterruptedException {

StringTokenizer itr = new StringTokenizer(value.toString());

while(itr.hasMoreTokens()) {

this.word.set(itr.nextToken());

context.write(this.word, one);

}

}

}

}

（2）将Hadoop配置文件添加到“WordCount” MapReduce项目

将log4j.properties 复制到 WordCount 项目下的 src 文件夹（~/workspace/WordCount/src）中：

[hfut@master ~]$ cp ~/hadoop-2.5.2/etc/hadoop/log4j.properties ~/workspace/MyWordCount/src

log4j 用于记录程序的输出日记，需要 log4j.properties 这个配置文件，如果没有复制该文件到项目中，运行程序后在 Console 面板中会出现警告提示：

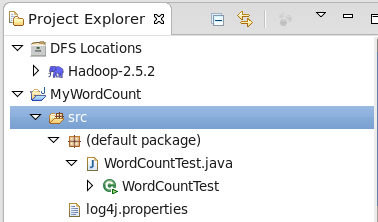
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.lib.MutableMetricsFactory).

log4j:WARN Please initialize the log4j system properly.

log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

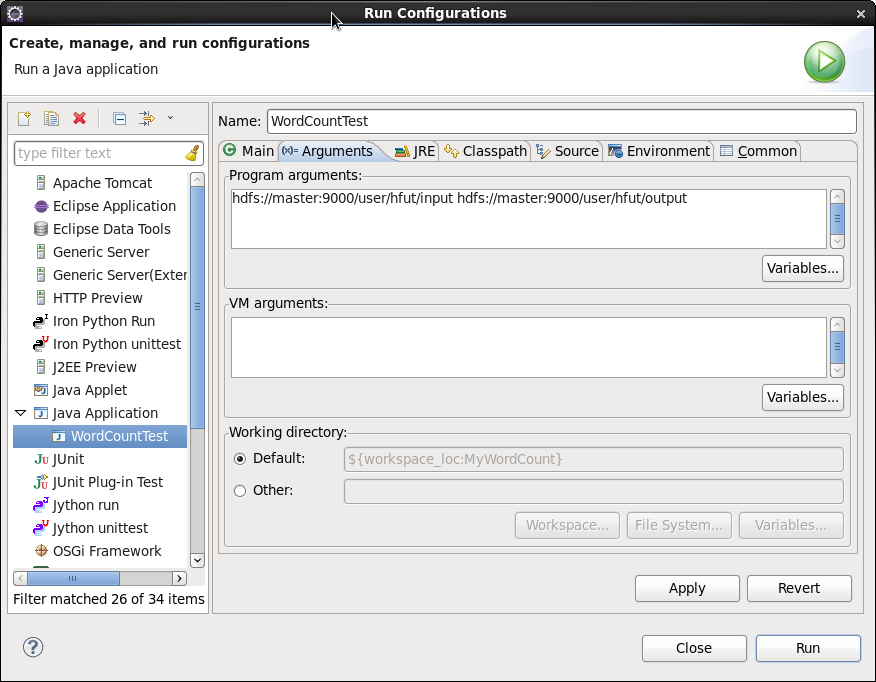
虽然不影响程序的正确运行的，但程序运行时无法看到任何提示消息（只能看到出错信息）。

复制完成后，务必右键点击 WordCount 选择 refresh 进行刷新（不会自动刷新，需要手动刷新），可以看到文件结构如下所示：

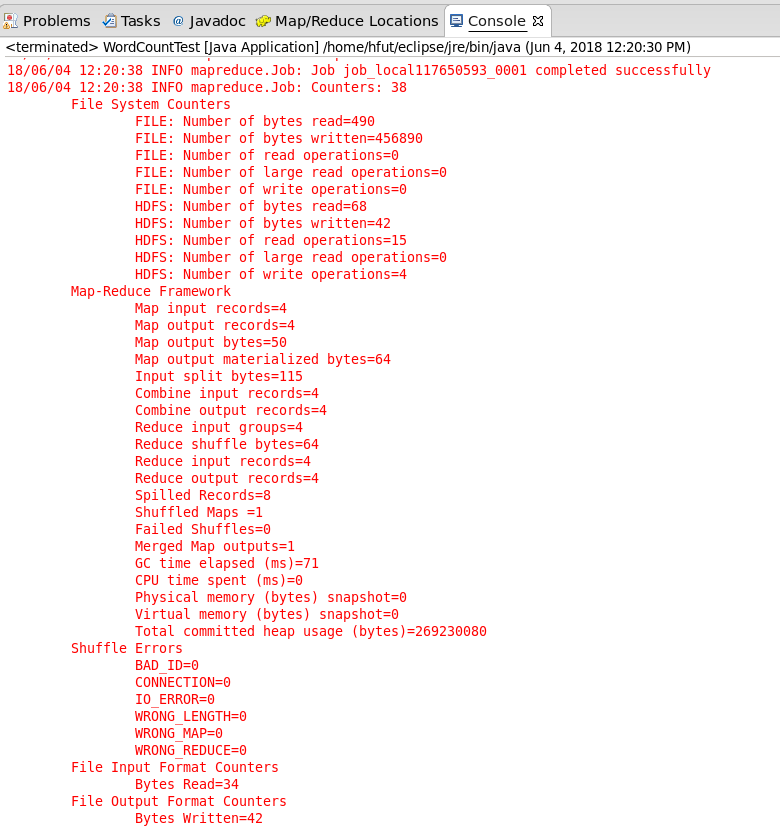


（3）通过Eclipse运行“MyWordCount” MapReduce项目

点击工具栏中的 Run 图标，或者右键点击 Project Explorer 中的 WordCountTest.java，选择 Run As -> Run on Hadoop，就可以运行 MapReduce 程序了。不过由于没有指定参数，运行时会提示 “Usage: wordcount”，需要通过Eclipse设定一下运行参数。

右键点击刚创建的 WordCount.java，选择 Run As -> Run Configurations，在此处可以设置运行时的相关参数（如果 Java Application 下面没有 WordCount，那么需要先双击 Java Application）。切换到 “Arguments” 栏，在 Program arguments 处填写 “hdfs://master:9000/user/hfut/input hdfs://master:9000/user/hfut/output” 就可以了。

点击Run运行程序，可以看到运行成功的提示。



刷新 DFS Location 后也能看到输出的 output 文件夹。双击part-r-00000文件，可以看到程序运行的结果。

