前一篇文章“[Apache Spark学习：将Spark部署到Hadoop 2.2.0上](http://developer.51cto.com/art/201401/426591.htm)”介绍了如何使用Maven编译生成可直接运行在Hadoop 2.2.0上的Spark jar包，而本文则在此基础上，介绍如何利用Eclipse构建Spark集成开发环境。

**（1） 准备工作**

在正式介绍之前，先要以下软硬件准备：

软件准备：

[Eclipse Juno版本（4.2版本）](http://www.eclipse.org/juno/)，可以直接点击这里下载：[Eclipse 4.2](http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/junosr2)

Scala 2.9.3版本，Window安装程序可以直接点击这里下载：[Scala 2.9.3](http://www.scala-lang.org/files/archive/scala-2.9.3.msi" \t "_blank)

Eclipse Scala IDE插件，可直接点击这里下载：[Scala IDE(for Scala 2.9.x and Eclipse Juno)](http://download.scala-ide.org/sdk/e38/scala29/stable/update-site.zip" \t "_blank)

硬件准备

装有Linux或者Windows操作系统的机器一台

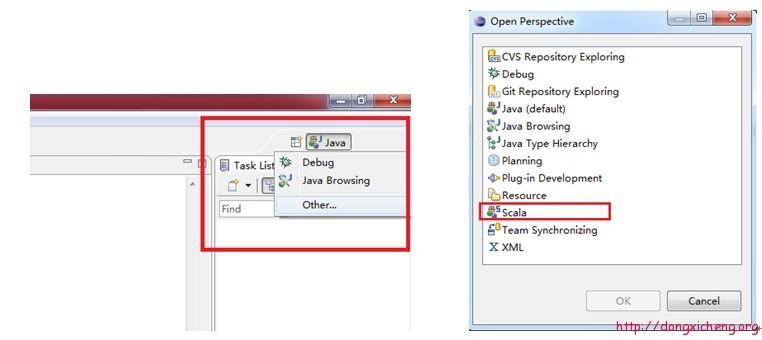
**（2） 构建Spark集成开发环境**

我是在windows操作系统下操作的，流程如下：

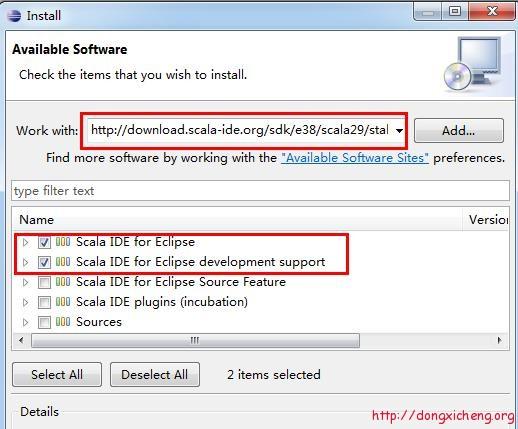
步骤1：安装scala 2.9.3：直接点击安装即可。

步骤2：将Eclipse Scala IDE插件中features和plugins两个目录下的所有文件拷贝到Eclipse解压后对应的目录中

步骤3：重新启动Eclipse，点击eclipse右上角方框按钮，如下图所示，展开后，点击“Other….”，查看是否有“Scala”一项，有的话，直接点击打开，否则进行步骤4操作。

[](http://s8.51cto.com/wyfs02/M02/09/F5/wKioL1LLbLWS6mm1AABwGtkWax0050.jpg)

步骤4：在Eclipse中，依次选择“Help” –> “Install New Software…”，在打开的卡里填入[http://download.scala-ide.org/sdk/e38/scala29/stable/site](http://download.scala-ide.org/sdk/e38/scala29/stable/site" \t "_blank)，并按回车键，可看到以下内容，选择前两项进行安装即可。（由于步骤3已经将jar包拷贝到eclipse中，安装很快，只是疏通一下）安装完后，重复操作一遍步骤3便可。

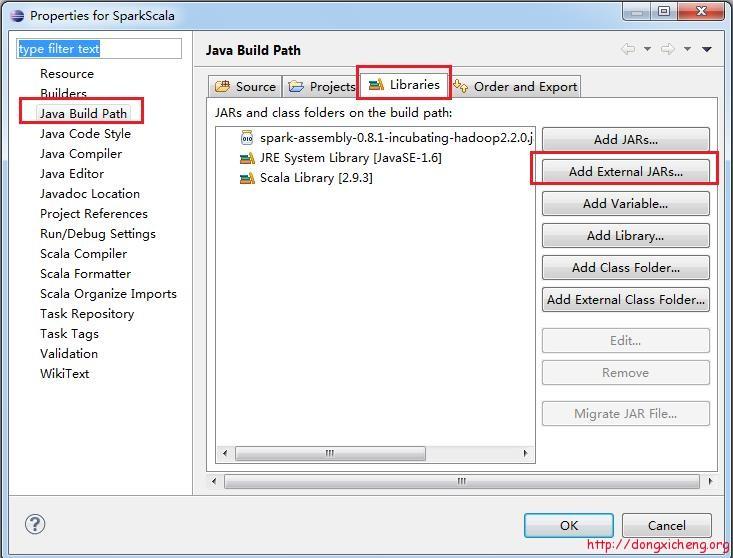
[](http://s9.51cto.com/wyfs02/M00/09/F5/wKioL1LLbLeg3MYCAACo4XwGR8U676.jpg)

**（3） 使用Scala语言开发Spark程序**

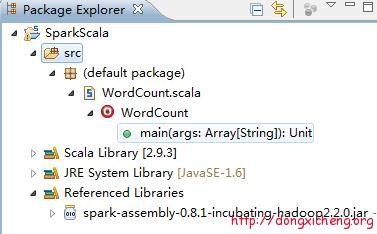
在eclipse中，依次选择“File” –>“New” –> “Other…” –>  “Scala Wizard” –> “Scala Project”，创建一个Scala工程，并命名为“SparkScala”。

右击“SaprkScala”工程，选择“Properties”，在弹出的框中，按照下图所示，依次选择“Java Build Path” –>“Libraties” –>“Add External JARs…”，导入文章“[Apache Spark：将Spark部署到Hadoop 2.2.0上](http://developer.51cto.com/art/201401/426591.htm)”中给出的

assembly/target/scala-2.9.3/目录下的spark-assembly-0.8.1-incubating- hadoop2.2.0.jar，这个jar包也可以自己编译spark生成，放在spark目录下的assembly/target/scala- 2.9.3/目录中。

[](http://s2.51cto.com/wyfs02/M01/09/F6/wKiom1LLbMWBDrv7AAEDzIESp6w695.jpg)

跟创建Scala工程类似，在工程中增加一个Scala Class，命名为：WordCount，整个工程结构如下：

[](http://s2.51cto.com/wyfs02/M02/09/F6/wKiom1LLbMfTW4lZAABIkBwHfyQ130.jpg)

WordCount就是最经典的词频统计程序，它将统计输入目录中所有单词出现的总次数，Scala代码如下：

1. **import** org.apache.spark.\_
2. **import** SparkContext.\_
3. object WordCount {
4. def main(args: Array[String]) {
5. **if** (args.length != 3 ){
6. println("usage is org.test.WordCount <master> <input> <output>")
7. **return**
8. }
9. val sc = **new** SparkContext(args(0), "WordCount",
10. System.getenv("SPARK\_HOME"), Seq(System.getenv("SPARK\_TEST\_JAR")))
11. val textFile = sc.textFile(args(1))
12. val result = textFile.flatMap(line => line.split("\\s+"))
13. .map(word => (word, 1)).reduceByKey(\_ + \_)
14. result.saveAsTextFile(args(2))
15. }
16. }

在Scala工程中，右击“WordCount.scala”，选择“Export”，并在弹出框中选择“Java” –> “JAR File”，进而将该程序编译成jar包，可以起名为“spark-wordcount-in-scala.jar”，我导出的jar包下载地址是 [spark-wordcount-in-scala.jar](http://kanboxshare.com/link/khGIHySIDoEg8SMP96C2PUVeLmQrHyI9BeZFGMOmRX9svM3UOu2dPT66Md4a0ZkQCKZeGiKJvHfUvM3wbLO4XeorAn3V2kvZ8V7Qq)。

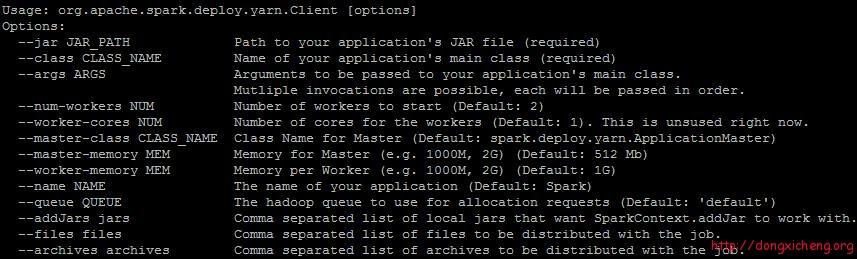
该WordCount程序接收三个参数，分别是master位置，HDFS输入目录和HDFS输出目录，为此，可编写run\_spark\_wordcount.sh脚本：

|  |
| --- |
| # 配置成YARN配置文件存放目录  export YARN\_CONF\_DIR=/opt/hadoop/yarn-client/etc/hadoop/  SPARK\_JAR=./assembly/target/scala-2.9.3/spark-assembly-0.8.1-incubating-hadoop2.2.0.jar \  ./spark-class org.apache.spark.deploy.yarn.Client \  –jar spark-wordcount-in-scala.jar \  –class WordCount \  –args yarn-standalone \  –args hdfs://hadoop-test/tmp/input \  –args hdfs:/hadoop-test/tmp/output \  –num-workers 1 \  –master-memory 2g \  –worker-memory 2g \  –worker-cores 2 |

需要注意以下几点：WordCount程序的输入参数通过“-args”指定，每个参数依次单独指定，第二个参数是HDFS上的输入目录，需要事先创建好，并上传几个文本文件，以便统计词频，第三个参数是HDFS上的输出目录，动态创建，运行前不能存在。

直接运行run\_spark\_wordcount.sh脚本即可得到运算结果。

在运行过程中，发现一个bug，org.apache.spark.deploy.yarn.Client有一个参数“–name”可以指定应用程序名称：

[](http://s4.51cto.com/wyfs02/M00/09/F5/wKioL1LLbLywmiZrAAC_vNqW42o213.jpg)

但是使用过程中，该参数会阻塞应用程序，查看源代码发现原来是个bug，该Bug已提交到Spark jira上：

1. // 位置：new-yarn/src/main/scala/org/apache/spark/deploy/yarn/ClientArguments.scala
2. case ("--queue") :: value :: tail =**>**
3. amQueue = value
4. args = tail
6. case ("--name") :: value :: tail =**>**
7. appName = value
8. args = tail //漏了这行代码，导致程序阻塞
10. case ("--addJars") :: value :: tail =**>**
11. addJars = value
12. args = tail

因此，大家先不要使用“–name”这个参数，或者修复这个bug，重新编译Spark。

**（4） 使用Java语言开发Spark程序**

方法跟普通的Java程序开发一样，只要将Spark开发程序包spark-assembly-0.8.1-incubating-hadoop2.2.0.jar作为三方依赖库即可。

**（5） 总结**

初步试用Spark On YARN过程中，发现问题还是非常多，使用起来非常不方便，门槛还是很高，远不如Spark On Mesos成熟。

原文链接：<http://dongxicheng.org/framework-on-yarn/spark-eclipse-ide/>