

## 用Maven编译Spark 1.0.0源码以错误解决

写在前面的话，最近发现有很多网站转载我博客的文章，这个我都不介意的，但是这些网站转载我博客都将文章的出处去掉了，直接变成自己的文章了！！我强烈谴责他们，鄙视那些转载文章去掉出处的人！所以为了防止这些，我以后发表文章的时候，将会在文章里面加入一些回复之后才可见的内容！！请大家不要介意，本博客Hadoop相关的文章都是原创，每篇文章都花了很长时间编写，这些都是无偿的，所以转载本文章请标明出处：

本篇博客地址：[《用Maven编译Spark 1.0.0源码以错误解决》](https://www.iteblog.com/archives/1038.html)：

<https://www.iteblog.com/archives/1038.html>

博客地址：[过往记忆](https://www.iteblog.com)：<https://www.iteblog.com>

目前Spark已经更新到1.0.0了，在本博客的[《Spark 1.0.0于5月30日正式发布》](#)中已经介绍了Spark 1.0.0的一些新特性。我们可以看到Spark 1.0.0带来了许多很不错的感受。本篇文章来介绍如何用Maven编译Spark 1.0.0源码。步骤主要如下：

一、先去Spark官网下载好源码。

```
# wget http://d3kbcqa49mib13.cloudfront.net/spark-1.0.0.tgz
# tar -zxf spark-1.0.0.tgz
```

二、设置MAVEN\_OPTS参数

在编译Spark的时候Maven需要很多内存，否则会出现类似下面的错误信息：

```
Exception in thread "main" java.lang.OutOfMemoryError: PermGen space
at org.apache.maven.cli.MavenCli.execute(MavenCli.java:545)
at org.apache.maven.cli.MavenCli.doMain(MavenCli.java:196)
at org.apache.maven.cli.MavenCli.main(MavenCli.java:141)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at org.codehaus.plexus.classworlds.launcher.Launcher.launchEnhanced(Launcher.java:290)
at org.codehaus.plexus.classworlds.launcher.Launcher.launch(Launcher.java:230)
at org.codehaus.plexus.classworlds.launcher.Launcher.mainWithExitCode(Launcher.java:409)
at org.codehaus.plexus.classworlds.launcher.Launcher.main(Launcher.java:352)
```

解决方法是：

```
export MAVEN_OPTS="-Xmx2g -XX:MaxPermSize=512M -XX:ReservedCodeCacheSize=512m"
```

三、 Cannot run program "javac": java.io.IOException:

如果编译的过程出现以下错误，请设置一下Java path。

```
[ERROR] Failed to execute goal net.alchim31.maven:scala-maven-plugin:3.1.6:
compile (scala-compile-first) on project spark-core_2.10: wrap:
java.io.IOException: Cannot run program "javac": java.io.IOException:
error=2, No such file or directory -> [Help 1]
```

四、 Please set the SCALA\_HOME

这个错误很明显没有设置SCALA\_HOME，去下载一个scala，然后设置一下即可。

```
[ERROR] Failed to execute goal org.apache.maven.plugins:
maven-antrun-plugin:1.7:run (default) on project spark-core_2.10:
An Ant BuildException has occurred: Please set the SCALA_HOME
(or SCALA_LIBRARY_PATH if scala is on the path) environment
variables and retry.
[ERROR] around Ant part ...<fail message="Please set the SCALA_HOME
(or SCALA_LIBRARY_PATH if scala is on the path) environment variables
and retry.">... @ 6:126 in spark-1.0.0/core/target/antrun/build-main.xml
```

五、选择相应的Hadoop和Yarn版本

因为不同版本的HDFS在协议上是不兼容的，所以如果你想用你的Spark从HDFS上读取数据，那么你就选择相应版本的HDFS来编译Spark，这个可以在编译的时候通过设置hadoop.version来选择，默认情况下，Spark是用Hadoop 1.0.4版本。

Hadoop version	Profile required
0.23.x	hadoop-0.23
1.x to 2.1.x	(none)
2.2.x	hadoop-2.2
2.3.x	hadoop-2.3
2.4.x	hadoop-2.4

(1)、对于Apache Hadoop 1.x、Cloudera CDH的mr1发行版，这些版本没有YARN，所以我们可以用下面的命令来编译Spark

```
# Apache Hadoop 1.2.1
mvn -Dhadoop.version=1.2.1 -DskipTests clean package

# Cloudera CDH 4.2.0 with MapReduce v1
mvn -Dhadoop.version=2.0.0-mr1-cdh4.2.0 -DskipTests clean package

# Apache Hadoop 0.23.x
mvn -Phadoop-0.23 -Dhadoop.version=0.23.7 -DskipTests clean package
```

(2)、对于Apache Hadoop 2.x, 0.23.x，Cloudera CDH以及其它一些版本的Hadoop，它们都是带有YARN，所以你可以启用“yarn-alpha”或者“yarn”配置选项，并通过yarn.version来设置不同版本的YARN，可选择值如下：

YARN version	Profile required
0.23.x 到 2.1.x	yarn-alpha
2.2.x和之后版本	yarn

我们可以通过下面命令来编译Spark

```
# Apache Hadoop 2.0.5-alpha
mvn -Pyarn-alpha -Dhadoop.version=2.0.5-alpha -DskipTests clean package

# Cloudera CDH 4.2.0
mvn -Pyarn-alpha -Dhadoop.version=2.0.0-cdh4.2.0 -DskipTests clean package

# Apache Hadoop 0.23.x
mvn -Pyarn-alpha -Phadoop-0.23 -Dhadoop.version=0.23.7 -DskipTests clean package

# Apache Hadoop 2.2.X
mvn -Pyarn -Phadoop-2.2 -Dhadoop.version=2.2.0 -DskipTests clean package

# Apache Hadoop 2.3.X
mvn -Pyarn -Phadoop-2.3 -Dhadoop.version=2.3.0 -DskipTests clean package

# Apache Hadoop 2.4.X
mvn -Pyarn -Phadoop-2.4 -Dhadoop.version=2.4.0 -DskipTests clean package

# Different versions of HDFS and YARN.
mvn -Pyarn-alpha -Phadoop-2.3 -Dhadoop.version=2.3.0 -Dyarn.version=0.23.7
-DskipTests clean package
```

当然（1）我们也可以使用sbt来编译Spark，本博客的[《Spark 0.9.1源码编译》](#)有详细的介绍，大家可以去参考。

（2）、自己编译Spark可以学到许多东西，不过你完全可以去下载已经编译好的Spark，这完全由你自己去决定。

（3）、本文原文出自：[《用Maven编译Spark 1.0.0源码以错误解决》](#)：  
<https://www.iteblog.com/archives/1038.html>

（4）、在下载下来的Spark源码中的同一级目录下有个make-distribution.sh脚本，这个脚本可以打包Spark的发行包，make-distribution.sh文件其实就是调用了Maven进行编译的，可以通过下面的命令运行：

```
./make-distribution.sh --tgz -Phadoop-2.2 -Pyarn -DskipTests -Dhadoop.version=2.2.0
```

大量关于Hadoop、Spark的干货博客：[过往记忆](#)：<https://www.iteblog.com>

如果你看到下面的输出信息，那恭喜你，编译成功了！

```
[WARNING] See http://docs.codehaus.org/display/MAVENUSER/Shade+Plugin
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] Spark Project Parent POM ..... SUCCESS [2.172s]
[INFO] Spark Project Core ..... SUCCESS [3:14.405s]
[INFO] Spark Project Bagel ..... SUCCESS [22.606s]
[INFO] Spark Project GraphX ..... SUCCESS [56.679s]
[INFO] Spark Project Streaming ..... SUCCESS [1:14.616s]
[INFO] Spark Project ML Library ..... SUCCESS [1:31.366s]
[INFO] Spark Project Tools ..... SUCCESS [15.484s]
[INFO] Spark Project Catalyst ..... SUCCESS [1:13.788s]
[INFO] Spark Project SQL ..... SUCCESS [1:22.578s]
[INFO] Spark Project Hive ..... SUCCESS [1:10.762s]
[INFO] Spark Project REPL ..... SUCCESS [36.957s]
[INFO] Spark Project YARN Parent POM ..... SUCCESS [2.290s]
[INFO] Spark Project YARN Stable API ..... SUCCESS [38.067s]
[INFO] Spark Project Assembly ..... SUCCESS [23.663s]
[INFO] Spark Project External Twitter ..... SUCCESS [19.490s]
[INFO] Spark Project External Kafka ..... SUCCESS [24.782s]
[INFO] Spark Project External Flume Sink ..... SUCCESS [24.539s]
[INFO] Spark Project External Flume ..... SUCCESS [27.308s]
[INFO] Spark Project External ZeroMQ ..... SUCCESS [21.148s]
```

```
[INFO] Spark Project External MQTT ..... SUCCESS [2:00.741s]
[INFO] Spark Project Examples ..... SUCCESS [54.435s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 17:58.481s
[INFO] Finished at: Tue Sep 16 19:20:10 CST 2014
[INFO] Final Memory: 76M/1509M
[INFO] -----
```

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: **【】**（**）**