Hadoop源码编译与调试

虽然在运行Hadoop的时候可以打印出大量的运行日志，但是很多时候只通过打印这些日志是不能很好地跟踪Hadoop各个模块的运行状况。这时候编译与调试Hadoop源码就得派上场了。这也就是今天本文需要讨论的。

编译Hadoop源码

先说说怎么编译Hadoop源码，本文主要介绍在Linux环境下用Maven来编译Hadoop。在编译Hadoop之前，我们需要准备好编译环境：

- 安装好1.6或以上的JDK;
- 安装Maven，被做好相应的配置;
- 安装ProtocolBuffer2.5.0，MapReduce和HDFS用protocol buffer来压缩和交换数据;如何安装请参见：《在CentOS下编译安装Protobuf类库》
- 如果你是第一次编译，则需要保证电脑能够网络连接，主要用于获取所有Maven和Hadoop依赖库;
- 如果你需要查找bug，则需要安装Findbugs;
- 如果要生成文档则需要安装Forrest 0.8;
- 如果编译native code则需要安装Autoto;

上面的前提条件1-4是必须的，5-7是可选的。编译的环境准备好之后，那我们就可以去编译Hadoop源码了。本文以编译Hadoop2.2.0源码为例，进行说明。从官方下载下来的Hadoop源码一般是包含以下几个Maven工程：

hadoop-assemblies
hadoop-client
hadoop-common-project
hadoop-dist
hadoop-hdfs-project
hadoop-mapreduce-project
hadoop-maven-plugins
hadoop-minicluster
hadoop-project
hadoop-project-dist
hadoop-tools
hadoop-yarn-project

上面各个功能模块的含义已经超出本文的主题，所以我不打算介绍。将上面的源码放在一个地方，比如，我放在/home/wyp/hadoop文件夹中，下面提供几种Hadoop源码的编译方式及说

明

如果不需要native code、忽略测试用例和文档，可以用下面的命令创建二进制分发版：
[wyp@date52 /home/wyp/hadoop]$ mvn package -Pdist -DskipTests -Dtar

创建二进制分发版，带native code和文档：
[wyp@date52 /home/wyp/hadoop]$ mvn package -Pdist,native,docs -DskipTests -Dtar

创建源码分发版
[wyp@date52 /home/wyp/hadoop]$ mvn package -Psrc -DskipTests

创建二进制带源码分发版，带native code和文档：
[wyp@date52 /home/wyp/hadoop]$ mvn package ₩
-Pdist,native,docs,src -DskipTests -Dtar

创建本地版web页面，放在/tmp/hadoop-site
[wyp@date52 /home/wyp/hadoop]$ mvn clean site; mvn ₩
site:stage -DstagingDirectory=/tmp/hadoop-site

上面提供了几种Hadoop源码的编译方式（参考文档：http://svn.apache.org/repos/asf/hadoop/common/trunk/BUILDING.txt），大家可以根据自己的需求选择不同的编译方式，但是这里我推荐第一种编译方式。我们可以将编译好的模块覆盖掉${HADOOP_HOME}/share/hadoop目录中对应模块里面的jar文件，然后重启Hadoop集群则新的编译包将生效。

远程调试Hadoop

远程调试对应用程序开发十分有用，那如何调试Hadoop源码？这里介绍如何用IDE远程调试Hadoop源码。本文以IntelliJ IDEA作为IDE，以调试Jobhistory WEB UI代码为例进行说明。

第一步：在启动Hadoop历史服务器进程之前在终端加入以下环境配置：

[wyp@date52 /home/wyp/hadoop]$ export HADOOP_OPTS="-Xdebug ₩
-Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8888"

这里对上面的几个参数进行说明：
-Xdebug 启用调试特性
-Xrunjdwp 启用JDWP实现，包含若干子选项：
transport=dt_socket JPDA front-end和back-end之间的传输方法。dt_socket表示使用套接字传输。
address=8888 JVM在8888端口上监听请求，这个设定为一个不冲突的端口即可。
server=y y表示启动的JVM是被调试者。如果为n，则表示启动的JVM是调试器。

suspend=y y表示启动的JVM会暂停等待，直到调试器连接上才继续执行。suspend=n，则JVM不会暂停等待。

第二步：启动Jobhistory进程

```
[wyp@date52 /home/wyp/hadoop]$ ${HODOOP_HOME}/sbin/mr-jobhistory-daemon.sh ₩
                                start historyserver
starting historyserver, logging to /home/wyp/Downloads/          ₩
                    hadoop/logs/mapred-wyp-historyserver-master.out
Listening for transport dt_socket at address: 8888
```

上面的Listening for transport dt_socket at address: 8888表明jobhistory已经在端口为8888启动了远程调试。

第三步：打开IntelliJ IDEA，找到hadoop-2.2.0-src₩hadoop-mapreduce-project₩hadoop-mapreduce-client₩hadoop-mapreduce-client-hs₩src₩main₩java₩org₩apache₩hadoop₩mapreduce₩v2₩hs₩webapp₩HsController.java类，在里面设置一些断点，然后依次选择菜单 Run->Run...->Edit Configurations...->选择左上角的+号->Remote，这时右边将会出现一个Configuration页面进行远程调试配置，请在Host和Port文本框里面输入jobhistory服务所在主机的IP及刚刚的8888端口，然后选择OK。这时候IDE进入了远程调试模式，你可以和普通的调试一样调试Hadoop源码。调试其他的Hadoop源码道理和上面的一样，这里就不一一列举了。

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: 【】（）