Hive源码分析(一)-源码调试

Hive 可分为 exec (hive-exec，主要对应源码里的ql目录) 和 metastore 两部分，其中exec对外有两种接口方式 CLIDriver 和 HiveServer2的ThriftCLIService。前者对应的就是直接执行hive命令的command line，后者对应就是thrift直连或jdbc的连接方式，因此这里其实有个知识点：hive命令的执行是不经过hiveserver2的服务的，虽然底层代码是共用的。而metastore对外主要是以Thrift方式为主，并实现了一个客户端 HiveMetaStoreClient。

# 环境准备

Windows10 + JDK 1.8 + Maven 3.5 + Hadoop + Hive + eclipse

1. 在本地部署好Hadoop并启动HDFS服务（参考：《01\_环境准备/01-Windows 环境搭建 Hadoop 3.1.2.docx》），并且确保HDFS的目录/tmp 和 /user/hive 目录hive用户有可写权限。
2. 安装并启动Hive（参考：《01\_环境准备/02-Windows 环境搭建 Hive 3.1.1 .docx》）
3. MySQL并不是必须的，如果你不关注metastore的存储，则无需指定相关的metastore配置，Hive会自动选择和初始化嵌入式数据库derby作为metastore的存储。
4. IDE：Idea或Eclipse，本文用的是eclipse。
5. 检查是否已安装Maven 和 JDK 1.7+。

# 代码准备

1. Hive官网或对应发行版的官网（HDP、CDH）下载对应版本的Hive源码。
2. Hive官网或对应发行版的官网（HDP、CDH）下载对应版本的Hive运行版。需要运行版的原因是用hive传统方式启动需要hive/lib/下的依赖jar。
3. 先对源码打包一次，防止导入IDE时太慢。命令如下：

mvn clean package -DskipTests (注意：windows下编译不成功，建议在linux下编译)

执行编译的目的在于，确保过程中生成的代码(Thrift)已经生成，这样导入IDE就不会出现有些类找不到的情况。

1. 初次打包没什么问题可以跳过测试，不然要挺久的。然后大陆连apache central repo有时可能会超时，超时请在不同时间段多试几次或使用梯子。
2. 源码maven项目导入IDE
3. 在Hive-cli项目中的BuildPaht中增加hadoop和hive的配置文件目录。在项目属性对话框中的Java Build Path 通过点击【Add External Class Folder】增加hadoop和Hived的目录。
4. CliDriver.java 右键 Debug As -> config

# 编译报错

1. 报错如下：

[root@slave1 apache-hive-3.1.1-src]# mvn clean package -DskipTests

[INFO] Scanning for projects...

Downloading: http://maven.aliyun.com/nexus/content/groups/public/org/apache/apache/18/apache-18.pom

[ERROR] [ERROR] Some problems were encountered while processing the POMs:

[FATAL] Non-resolvable parent POM for org.apache.hive:hive:3.1.1: Could not transfer artifact org.apache:apache:pom:18 from/to alimaven (http://maven.aliyun.com/nexus/content/groups/public/): sun.security.validator.ValidatorException: PKIX path validation failed: java.security.cert.CertPathValidatorException: validity check failed and 'parent.relativePath' points at wrong local POM @ line 17, column 11

@

[ERROR] The build could not read 1 project -> [Help 1]

[ERROR]

[ERROR] The project org.apache.hive:hive:3.1.1 (/opt/source/apache-hive-3.1.1-src/pom.xml) has 1 error

[ERROR] Non-resolvable parent POM for org.apache.hive:hive:3.1.1: Could not transfer artifact org.apache:apache:pom:18 from/to alimaven (http://maven.aliyun.com/nexus/content/groups/public/): sun.security.validator.ValidatorException: PKIX path validation failed: java.security.cert.CertPathValidatorException: validity check failed and 'parent.relativePath' points at wrong local POM @ line 17, column 11: NotBefore: Fri May 31 11:37:01 HKT 2019 -> [Help 2]

[ERROR]

[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.

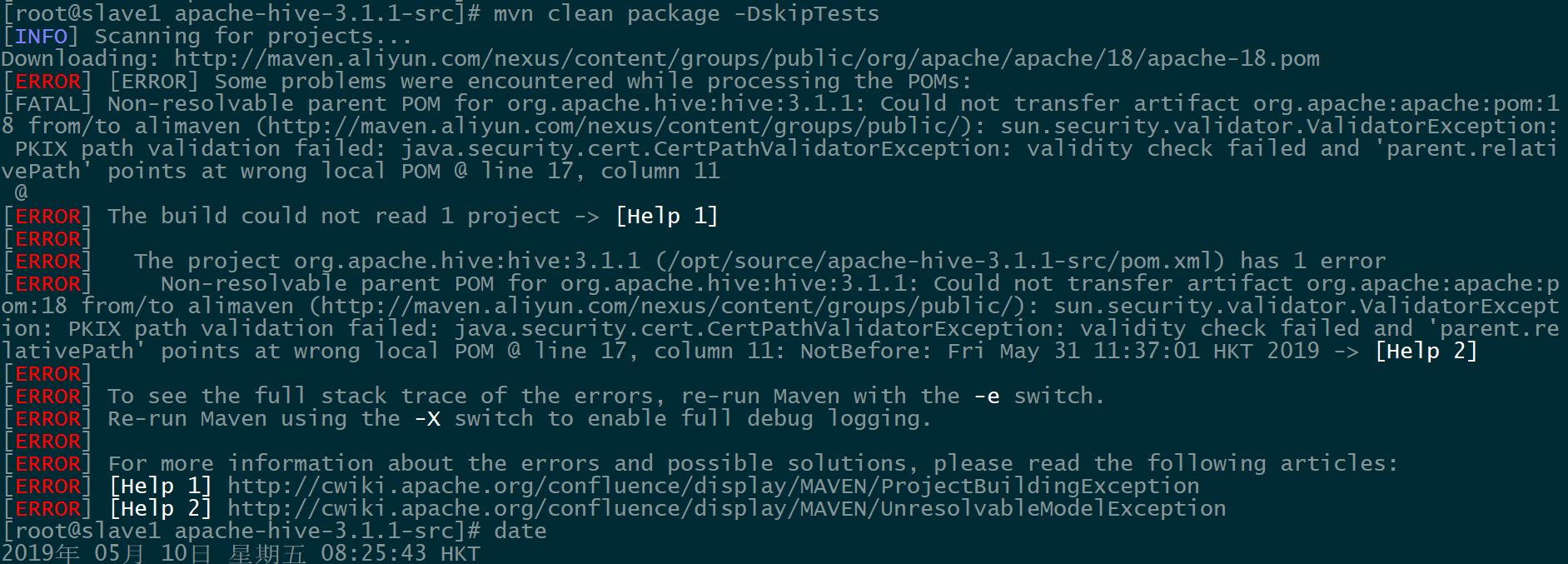
[ERROR] Re-run Maven using the -X switch to enable full debug logging.

[ERROR]

[ERROR] For more information about the errors and possible solutions, please read the following articles:

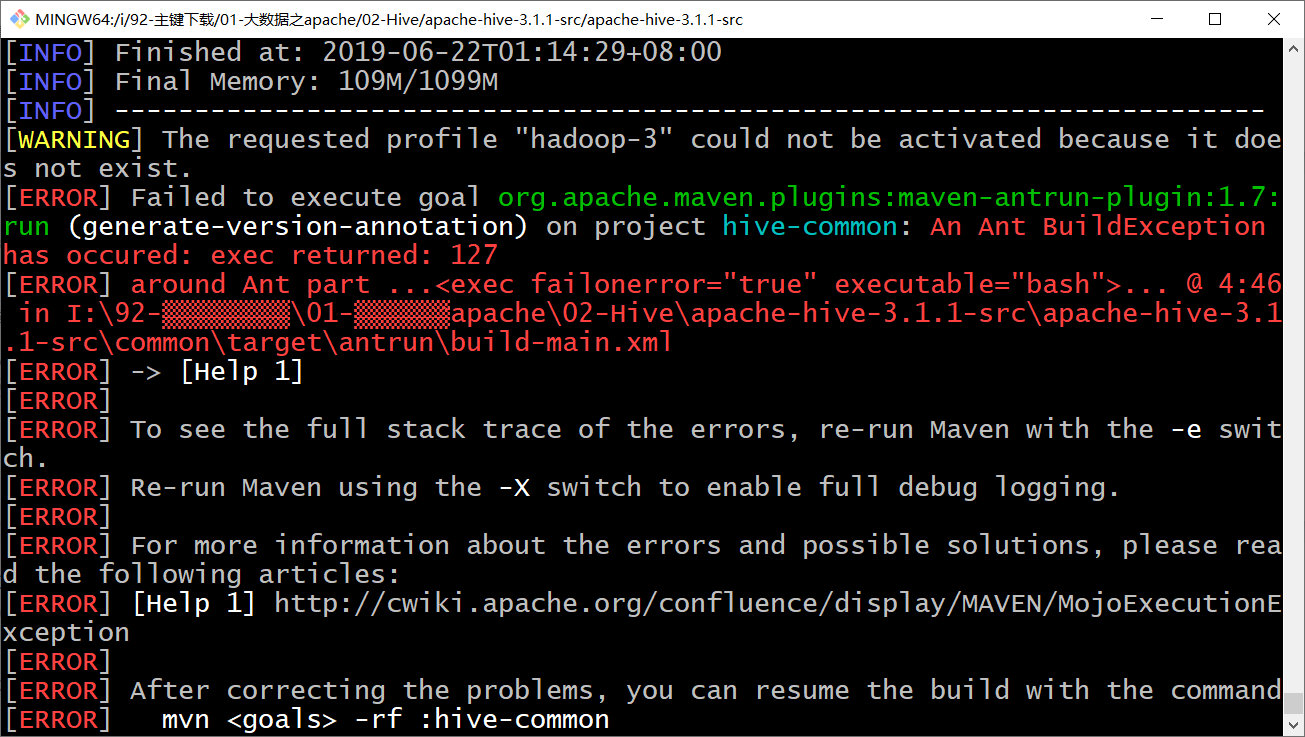
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/ProjectBuildingException

[ERROR] [Help 2] http://cwiki.apache.org/confluence/display/MAVEN/UnresolvableModelException



解决方法：更新linux 系统时间

1. 报错如下图所示：



解决方式：应在linux环境下编译

1. 报错如下：

Plugin execution not covered by lifecycle configuration: org.apache.avro:avro-maven-plugin:1.7.6:protocol (execution: default, phase: generate-test-sources) pom.xml /hive-hbase-handler line 246 Maven Project Build Lifecycle Mapping Problem

解决方式1：pom.xml文件中加入如下配置

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.eclipse.m2e</groupId>

<artifactId>lifecycle-mapping</artifactId>

<version>1.0.0</version>

<configuration>

<lifecycleMappingMetadata>

<pluginExecutions>

<pluginExecution>

<pluginExecutionFilter>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<versionRange>[1.7,)</versionRange>

<goals>

<goal>run</goal>

</goals>

</pluginExecutionFilter>

<action>

<ignore />

</action>

</pluginExecution>

<pluginExecution>

<pluginExecutionFilter>

<groupId>org.apache.avro</groupId>

<artifactId>avro-maven-plugin</artifactId>

<versionRange>1.7.6</versionRange>

<goals>

<goal>protocol</goal>

</goals>

</pluginExecutionFilter>

<action>

<ignore />

</action>

</pluginExecution>

</pluginExecutions>

</lifecycleMappingMetadata>

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

解决方法2: 菜单栏 Window -> Preferences -> Maven -> Lifecycle Mapping -> Change mapping file location -> lifecycle-mapping-metadata.xml

<?xml version="1.0" encoding="UTF-8" >

<lifecycleMappingMetadata>

<pluginExecutions>

<pluginExecution>

<pluginExecutionFilter>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<goals>

<goal>run</goal>

</goals>

<versionRange>[1.7,)</versionRange>

</pluginExecutionFilter>

<action>

<ignore />

</action>

</pluginExecution>

<pluginExecution>

<pluginExecutionFilter>

<groupId>org.apache.avro</groupId>

<artifactId>avro-maven-plugin</artifactId>

<goals>

<goal>protocol</goal>

</goals>

<versionRange>1.7.6</versionRange>

</pluginExecutionFilter>

<action>

<ignore />

</action>

</pluginExecution>

</pluginExecutions>

</lifecycleMappingMetadata>

# 调试exec

1. 在Hive运行版目录下执行 hive --service metastore 启动 metastore
2. 将本地Hadoop的core-site.xml放到 hive 源码目录的 src/main/resources 下，重新打包（如有必要）
3. debug模式运行hive-cli中的CLIDriver类，此时控制台可见hive command line，可以输入命令然后再ql的相应代码中打断点进行单步或拦截。

# 调试metastore

1. 以debug模式启动HiveMetastore
2. 另外写testcase使用HiveMetastoreClient类进行交互访问
3. 修改/hive-cli/src/java/org/apache/hadoop/hive/cli/CliDriver.java

在构造方法CliDriver中加入:

// 解决在Eclipse下运行时，控制台无法接受输入，输入命令、回车，系统没反应

System.*setProperty*("jline.WindowsTerminal.directConsole", "false");

// 是否打印调试信息

System.*setProperty*("jline.internal.Log.debug", "false");

