Hudi数据湖笔记

目录:

Hudi 下载地址

1. Github 下载（推荐）

1.1 最新0.9.0 SNAPSHOT 版本

https://github.com/apache/hudi/tree/master

拉取0.9.0 SNAPSHOT 源码

```
$ git clone https://github.com/apache/hudi.git
```

1.2 最新Release 0.8.0 branches 版本

https://github.com/apache/hudi/tree/release-0.8.0

拉取指定分支源码

```
$ git clone -b release-0.8.0 git@github.com:apache/hudi.git
```

2. Apache Dist下载

下载地址：https://archive.apache.org/dist/hudi/0.8.0/

修改源码pom.xml文件添加阿里云仓库

```
<repositories>
    <!-- 添加如下仓库地址 -->
    <repository>
        <id>maven-ali</id>
        <url>http://maven.aliyun.com/nexus/content/groups/public//</url>
        <releases>
            <enabled>true</enabled>
        </releases>
        <snapshots>
            <enabled>true</enabled>
            <updatePolicy>always</updatePolicy>
            <checksumPolicy>fail</checksumPolicy>
        </snapshots>
    </repository>
    <!-- 添加仓库地址 -->
    <repository>
      <id>Maven Central</id>
      <name>Maven Repository</name>
      <url>https://repo.maven.apache.org/maven2</url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
    </repository>
    <repository>
      <id>cloudera-repo-releases</id>
      <url>https://repository.cloudera.com/artifactory/public/</url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
    </repository>
    <repository>
      <id>confluent</id>
      <url>https://packages.confluent.io/maven/</url>
    </repository>
  </repositories>
```

Building Apache Hudi from source

Prerequisites for building Apache Hudi:

- Unix-like system (like Linux, Mac OS X)
- Java 8 (Java 9 or 10 may work)
- Git
- Maven

```
# Checkout code and build
# Hudi 源码下载到 C:\Users\49921\Desktop\
# windows 下进入PowerShell , 进入Hudi 目录下
```

```
PS C:\Users\49921\Desktop> cd hudi
PS C:\Users\49921\Desktop\hudi> pwd
Path
----
C:\Users\49921\Desktop\hudi

# 执行mvn 打包命令 , 默认hudi 0.8.0 基于hadoop 2.7.3 ， spark 2.4.4
PS C:\Users\49921\Desktop\hudi> mvn clean package -DskipTests

# Start comm4.4
spark-2.4.4-bin-hadoop2.7/bin/spark-shell \
  --jars `ls packaging/hudi-spark-bundle/target/hudi-spark-bundle_2.11-*.*.*-
SNAPSHOT.jar` \
  --conf 'spark.serializer=org.apache.spark.serializer.KryoSerializer'
```

打包之后的jar包位置：C:\Users\49921\Desktop\hudi-release-0.8.0\hudi-release-0.8.0\packaging

具体hudi与spark集成jar 位于： C:\Users\49921\Desktop\hudi-release-0.8.0\hudi-release-0.8.0\packaging\hudi-spark-bundle\target

To build the Javadoc for all Java and Scala classes:

```
# Javadoc generated under target/site/apidocs
mvn clean javadoc:aggregate -Pjavadocs
```

Build with Scala 2.12

The default Scala version supported is 2.11. To build for Scala 2.12 version, build using `scala-2.12` profile

```
mvn clean package -DskipTests -Dscala-2.12
```

Build with Spark 3.0.0

The default Spark version supported is 2.4.4. To build for Spark 3.0.0 version, build using `spark3` profile

Hadoop version supported is 2.7.3

```
mvn clean package -DskipTests -Dspark3
```

Build with Sugon Cluster Version

hadoop 3.1.1
zookeeper 3.4.6
hive 3.1.0
hbase 2.1.7 1.2.3
spark 2.3.4
kafka 2.4.1

presto ?

修改源码中POM文件中的对应组件Version， 尽量和上述版本一致

```
<!-- 修改如下组件版本即可 ， 其余保持不动-->
<properties>
    <kafka.version>2.4.1</kafka.version>
    <hadoop.version>3.1.1</hadoop.version>
    <hive.version>3.1.0</hive.version>
    <spark2.version>2.3.4</spark2.version>
    <spark3.version>3.0.0</spark3.version>
    <hbase.version>2.1.7</hbase.version>
</properties>
```

再次执行打包命令

```
mvn clean package -DskipTests
```

mvn 报错

1. hudi-hadoop-mr 缺少org.pentaho:pentaho-aggdesigner-algorithm:jar:5.1.5-jhyde 包

```
# 详细报错信息
[ERROR] Failed to execute goal on project hudi-hadoop-mr: Could not resolve
dependencies for project org.apache.hudi:hudi-hadoop-mr:jar:0.9.0-SNAPSHOT:
Could not find artifact org.pentaho:pentaho-aggdesigner-algorithm:jar:5.1.5-
jhyde in maven-ali (http://maven.aliyun.com/nexus/content/groups/public//) ->
[Help 1]
```

解决办法：

1. 手动下载此jar包

pentaho 中央仓库

下载地址1： https://public.nexus.pentaho.org/repository/proxy-public-3rd-party-release/org/pentaho/pentaho-aggdesigner-algorithm/5.1.5-jhyde/pentaho-aggdesigner-algorithm-5.1.5-jhyde.jar

MVN 公共仓库：

下载地址2：https://mvnrepository.com/artifact/org.pentaho/pentaho-aggdesigner-algorithm/5.1.5-jhyde

2. 将手动下载的jar移动到本地maven仓库地址中(推荐)

   需要先在hudi 目录下执行一次mvn clean package -DskipTests， 才能生存pentaho\pentaho-aggdesigner-algorithm\5.1.5-jhyde目录，之后再把jar包移进去

   本地maven仓库地址：C:\Users\49921.m2\repository\org\pentaho\pentaho-aggdesigner-algorithm\5.1.5-jhyde

3. 或者手动install 本地jar包到本地仓库中

   -Dfile 为本地jar路径

   mvn install:install-file -DgroupId=org.pentaho -DartifactId=pentaho-aggdesigner-algorithm -Dversion=5.1.5-jhyde -Dpackaging=jar -Dfile=C:/users/49921/Desktop/pentaho-aggdesigner-algorithm-5.15-jhyde.jar

mvn install:install-file将本地一个中央仓库没有的jar包，推到本地仓库

参考链接： https://www.cnblogs.com/daofree/p/12681855.html

2. window环境下注释掉 hudi-integ-test和hudi-integ-test-bundle

```
# 报错信息
[ERROR] Failed to execute goal org.codehaus.mojo:exec-maven-plugin:1.6.0:exec
(Setup HUDI_WS) on project hudi-integ-test: Command execution failed.: Cannot run
program "\bin\bash" (in directory "C:\Users\49921\Desktop\hudi-release-
0.8.0\hudi-release-0.8.0\hudi-integ-test"): CreateProcess error=2, 系统找不到指定的
文件。 -> [Help 1]
```

解决办法

POM 文件注释报错的相关模块

```
<modules>
   <module>hudi-common</module>
   <module>hudi-cli</module>
   <module>hudi-client</module>
   <module>hudi-hadoop-mr</module>
   <module>hudi-spark-datasource</module>
   <module>hudi-timeline-service</module>
   <module>hudi-utilities</module>
   <module>hudi-sync</module>
   <module>packaging/hudi-hadoop-mr-bundle</module>
   <module>packaging/hudi-hive-sync-bundle</module>
   <module>packaging/hudi-spark-bundle</module>
   <module>packaging/hudi-presto-bundle</module>
   <module>packaging/hudi-utilities-bundle</module>
   <module>packaging/hudi-timeline-server-bundle</module>
   <module>docker/hoodie/hadoop</module>
   <!--  如下两个test模块注释掉
   <module>hudi-integ-test</module>
   <module>packaging/hudi-integ-test-bundle</module>
   -->
   <module>hudi-examples</module>
   <module>hudi-flink</module>
   <module>packaging/hudi-flink-bundle</module>
 </modules>
```