



0.1. 安装 Hive

第一节 简介

Hive 提供了简化 Mapreduce 查询的功能，可以输入类 SQL 语句（HQL 或 HiveQL）完成对 HDFS 中数据的查询。是非常重要的一个工具。

第二节 配置文件

hive-env.sh

```
HADOOP_HOME=/opt/hadoop
export HIVE_CONF_DIR=/home/zhangyu/hive/conf
```

hive-site.xml

```
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://master:3306/hive?
      createDatabaseIfNotExsit=true;
      characterEncoding=latin1</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>root</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>password</value>
  </property>
</configuration>
```

启动 Hive 之前，需要配置确保安装和配置好了 MySQL，否则会报拒绝连接错误。

Hive 启动步骤

```
# 更新软件源，如果必要的话
$ sudo apt update
# 安装mysql，设置root用户密码：password
$ sudo apt install mysql-server
$ mysql -u root -p password
mysql> create database hive;
# 由于配置的是远程访问
# 可能需要设置mysql访问的主机权限
mysql> use mysql;
mysql> select user, host from user;
mysql> flush privilege;
mysql> exit;
# 初次连接，使用schema-tool初始化mysql中Hive数据库中的表
$ schema-tool --initSchema
# 测试运行
$ hive
```

第三节 Hive 查询与 Shell 操作

Hive 定义了一套自己的 SQL，简称 HQL，它与关系型数据库的 SQL 略有不同，但支持了绝大多数的语句如 DDL、DML 以及常见的聚合函数、连接查询、条件查询。DDL 操作（数据定义语言）包括：Create、Alter、Show、Drop 等。

1. create database- 创建新数据库
2. alter database - 修改数据库
3. drop database - 删除数据库
4. create table - 创建新表
5. alter table - 变更（改变）数据库表
6. drop table - 删除表
7. create index - 创建索引（搜索键）
8. drop index - 删除索引
9. show table - 查看表

DML 操作（数据操作语言）包括：Load、Insert、Update、Delete、Merge。

1. load data - 加载数据
2. insert into - 插入数据
3. insert overwrite - 覆盖数据（insert ... values 从 Hive 0.14 开始可用。）
4. update table - 更新表（update 在 Hive 0.14 开始可用，并且只能在支持 ACID 的表上执行）
5. delete from table where id = 1; - 删除表中 ID 等于 1 的数据（delete 在 Hive 0.14 开始可用，并且只能在支持 ACID 的表上执行）
6. merge - 合并（MERGE 在 Hive 2.2 开始可用，并且只能在支持 ACID 的表上执行）

注意：频繁的 update 和 delete 操作已经违背了 Hive 的初衷。不到万不得已的情况，还是使用增量添加的方式最好。

```
$ wget http://192.168.1.100:60000/allfiles/hive3/buyer_log
$ wget http://192.168.1.100:60000/allfiles/hive3/buyer_favorite
$ hive
hive> create table buyer_log(id string,
buyer_id string,dt string,
ip string,opt_type string)
row format delimited fields
terminated by '\t' stored as textfile;
hive> create table buyer_favorite(
buyer_id string,goods_id string,dt string)
row format delimited fields
terminated by '\t' stored as textfile;
hive> select * from buyer_log limit 10;
hive> load data local inpath
'/data/hive3/buyer_log' into table buyer_log;
hive> load data local inpath
'/data/hive3/buyer_favorite' into table buyer_favorite;
hive> select * from buyer_log limit 10;
hive> select l.dt,f.goods_id from buyer_log l,buyer_favorite f
where l.buyer_id = f.buyer_id limit 10;
```

第四节 Hive JDBC 编程

使用 IDEA 开发 Hive JDBC 编程，用 Maven 管理项目依赖。

启动 Hive 服务，注意这里使用的不是 Cli:

启动 Hive 服务

```
$ hive --service hiveserver2 &
```

查看 Hive 服务

```
$ netstat -nptl | grep 10000  
# 出现监听进程，说明启动正常
```

启动 beeline，并尝试连接

```
$ beeline  
$ !connect jdbc:hive2://master:10000
```

在这里连接的时候总是报错，无法连接成功，查看后发现主要是授权的问题，可是明明用户名和密码都输入正确了。后来通过搜索查看到问题所在，修改 Hadoop 配置 `core-site.xml` 文件，添加如下内容取消权限检查，便能够成功进入了。

```
<property>  
  <name>hadoop.proxyuser.zhangyu.hosts</name>  
  <value>*</value>  
</property>  
<property>  
  <name>hadoop.proxyuser.zhangyu.groups</name>  
  <value>*</value>  
</property>
```

下面开始新建项目编程:

POM 文件依赖

```
<dependency>  
  <groupId>org.apache.hive</groupId>  
  <artifactId>hive-jdbc</artifactId>  
  <version>1.2.1</version>  
</dependency>  
<dependency>  
  <groupId>org.apache.hadoop</groupId>  
  <artifactId>hadoop-common</artifactId>  
  <version>3.1.2</version>  
</dependency>
```

注意这里的hive-jdbc版本用的是 1.2.1，而不是 Hive 的 3.1.1。因为发现使用 3.1.1 时，有依赖无法解决，一直飘红。所以找了一个使用量比较高的版本 1.2.1，因为测试比较简单，仅仅是输出打印，降低版本后能够正常运行了。

HiveDemo.java

```
import java.sql.*;

public class HiveDemo {
    public static void main(String[] args) {
        try {
            Connection connection = null;
            String driverName = "org.apache.hive.jdbc.HiveDriver";
            Class.forName(driverName);
            // arg1: connect url, arg2: username, arg3: password
            connection = DriverManager.getConnection("jdbc:hive2://
                master:10000/default", "root", "password");
            Statement statement = connection.createStatement();
            ResultSet resultSet = null;
            String sql = "select * from sogou_data1";
            System.out.println("Now running: " + sql);
            resultSet = statement.executeQuery(sql);
            while(resultSet.next()) {
                System.out.println(resultSet.getString(1) + '\t'
                    + resultSet.getString(2) + '\t'
                    + resultSet.getString(3) + '\t'
                    + resultSet.getString(4) + '\t'
                    + resultSet.getString(5) + '\t'
                );
            }
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

程序运行截图如下：

