

Hive安装

第01节 Hive安装部署

1. 安装前准备

由于Hive是一款基于Hadoop的数据仓库软件，通常部署运行在Linux系统之上。因此必须先保证服务器的基础环境正常，Hadoop环境正常运行，Hive不是分布式安装运行的软件，其分布式的特性主要借由Hadoop完成。包括分布式存储、分布式计算。

- 创建服务端目录用于存放Hive安装文件

```
# 用于存放安装包  
mkdir /opt/tools  
# 用于存放解压后的文件  
mkdir /opt/server
```

- 切换到/opt/tools目录，上传hive安装包

```
cd /opt/tools
```

- 共涉及到两个安装包，分别是apache-hive-3.1.2-bin.tar.gz与mysql-5.7.34-1.el7.x86_64.rpm-bundle.tar

2. 安装MySQL

前面提到Hive允许将元数据存储于本地或远程的外部数据库中，这种设置可以支持Hive的多会话生产环境，在本案例中采用MySQL作为Hive的元数据存储库。

- 卸载Centos7自带mariadb

```
# 查找  
rpm -qa|grep mariadb  
# mariadb-libs-5.5.52-1.el7.x86_64  
# 卸载  
rpm -e mariadb-libs-5.5.52-1.el7.x86_64 --nodeps
```

- 解压mysql

```
# 创建mysql安装包存放点  
mkdir /opt/server/mysql  
# 解压  
tar xvf mysql-5.7.34-1.el7.x86_64.rpm-bundle.tar -C /opt/server/mysql/
```

- 执行安装

```
# 安装依赖
yum -y install libaio
yum -y install libncurses*
yum -y install perl perl-devel
# 切换到安装目录
cd /opt/server/mysql/
# 安装
rpm -ivh mysql-community-common-5.7.34-1.el7.x86_64.rpm
rpm -ivh mysql-community-libs-5.7.34-1.el7.x86_64.rpm
rpm -ivh mysql-community-client-5.7.34-1.el7.x86_64.rpm
rpm -ivh mysql-community-server-5.7.34-1.el7.x86_64.rpm
```

- 启动Mysql

```
#启动mysql
systemctl start mysqld.service
#查看生成的临时root密码
cat /var/log/mysqld.log | grep password
```

```
[root@node01 mysql]# cat /var/log/mysqld.log | grep password
2021-11-09T10:39:41.579372Z 1 [Note] A temporary password is generated for root@localhost: d_he,HxRh1?x
[root@node01 mysql]#
```

- 修改初始的随机密码

```
# 登录mysql
mysql -u root -p
Enter password:      #输入在日志中生成的临时密码
# 更新root密码 设置为root
set global validate_password_policy=0;
set global validate_password_length=1;
set password=password('root');
```

- 授予远程连接权限

```
grant all privileges on *.* to 'root' @'%' identified by 'root';
# 刷新
flush privileges;
```

- 控制命令

```
#mysql的启动和关闭 状态查看
systemctl stop mysqld
systemctl status mysqld
systemctl start mysqld

#建议设置为开机自启动服务
systemctl enable mysqld
#查看是否已经设置自启动成功
systemctl list-unit-files | grep mysqld
```

3. Hive安装配置

- 解压安装包

```
# 切换到安装包目录
cd /opt/tools
# 解压到/root/server目录
tar -zxvf apache-hive-3.1.2-bin.tar.gz -C /opt/server/
```

- 添加mysql_jdbc驱动到hive安装包lib目录下

```
# 上传mysql-connector-java-5.1.38.jar
cd /opt/server/apache-hive-3.1.2-bin/lib
```

- 修改hive环境变量文件，指定Hadoop的安装路径

```
cd /opt/server/apache-hive-3.1.2-bin/conf
cp hive-env.sh.template hive-env.sh
vim hive-env.sh
# 加入以下内容
HADOOP_HOME=/opt/server/hadoop-3.1.0
```

- 新建 hive-site.xml 文件，内容如下，主要是配置存放元数据的 MySQL 的地址、驱动、用户名和密码等信息

```
vim hive-site.xml
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <!-- 存储元数据mysql相关配置 /etc/hosts -->
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://server:3306/hive?
createDatabaseIfNotExist=true&useSSL=false&useUnicode=true&characterEncoding=UTF-8</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>root</value>
  </property>
```

```
</configuration>
```

- 初始化元数据库，当使用的 hive 是 1.x 版本时，可以不进行初始化操作，Hive 会在第一次启动的时候会自动进行初始化，但不会生成所有的元数据信息表，只会初始化必要的一部分，在之后的使用中用到其余表时会自动创建；
- 当使用的 hive 是 2以上版本时，必须手动初始化元数据库，初始化命令：

```
cd /opt/server/apache-hive-3.1.2-bin/bin
./schematool -dbType mysql -initSchema
```

```
Initialization script completed
schemaTool completed
[root@node01 bin]#
```

- 初始化成功会在mysql中创建74张表

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| hive |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> use hive;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_hive |
+-----+
| AUX_TABLE |
| BUCKETING_COLS |
| CDS |
| COLUMNS_V2 |
| COMPACTION_QUEUE |
| COMPLETED_COMPACTIONS |
| COMPLETED_TXN_COMPONENTS |
| CTLGS |
+-----+
```



```
console) --详细模式
```

交互式命令行

直接使用 Hive 命令，不加任何参数，即可进入交互式命令行。

执行SQL命令

在不进入交互式命令行的情况下，可以使用 hive -e 执行 SQL 命令。

```
hive -e 'select * from emp';
```

执行SQL脚本

用于执行的 sql 脚本可以在本地文件系统，也可以在 HDFS 上。

```
# 本地文件系统
hive -f /usr/file/simple.sql;
# HDFS文件系统
hive -f hdfs://node01:8020/tmp/simple.sql;
```

第02节 Hive简单使用

在hive中创建、切换数据库，创建表并执行插入数据操作，最后查询是否插入成功。

1. 基本操作

- 连接Hive

```
hive
```

- 数据库操作

```
create database test;--创建数据库
show databases;--列出所有数据库
use test;--切换数据库
```

- 表操作

```
-- 建表
create table t_student(id int,name varchar(255));
-- 插入一条数据
insert into table t_student values(1,"potter");
-- 查询表数据
select * from t_student;
```

在执行插入数据的时候，发现插入速度极慢，sql执行时间很长，花费了26秒，并且显示了MapReduce程序的进度

```

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-11-10 11:28:02,335 Stage-1 map = 0%, reduce = 0%
2021-11-10 11:28:08,523 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.86 sec
2021-11-10 11:28:13,634 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.81 sec
MapReduce Total cumulative CPU time: 2 seconds 810 msec
Ended Job = job_1636506894917_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://node01:8020/user/hive/warehouse/test.db/t_student/.hive-staging_hive_20211110_794399324094273757-1/-ext-10000
Loading data to table test.t_student
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 2.81 sec HDFS Read: 15839 HDFS Write: 242 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 810 msec
OK
Time taken: 26.419 seconds
hive>

```

2. 查看YARN及HDFS

登录Hadoop YARN观察是否有MapReduce程序执行，地址：<http://192.168.40.100:8088>，需要根据自己的服务器IP进行更换

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus
application_1636506894917_0001	root	insert into table t_stu...values(1,"potter") (Stage-1)	MAPREDUCE	default	0	Wed Nov 10 11:27:55 +0800 2021	Wed Nov 10 11:28:14 +0800 2021	FINISHED	SUCCEEDED

发现运行的任务名称就是所执行的SQL语句，任务的类型为MapReduce，最终状态为SUCCEEDED。

登录Hadoop HDFS浏览文件系统，根据Hive的数据模型，表的数据最终是存储在HDFS和表对应的文件夹下的。

地址：<http://192.168.40.100:9870/>，需要根据自己的服务器IP进行更换

/user/hive/warehouse/test.db/t_student

Go!

Show

25

entries

Search:

<input type="checkbox"/>	<div><div>⬇</div><div>⬆</div>Permission</div>	<div><div>⬆</div><div>⬇</div>Owner</div>	<div><div>⬆</div><div>⬇</div>Group</div>	<div><div>⬆</div><div>⬇</div>Size</div>	<div><div>⬆</div><div>⬇</div>Last Modified</div>	<div><div>⬆</div><div>⬇</div>Replication</div>	<div><div>⬆</div><div>⬇</div>Block Size</div>	<div><div>⬆</div><div>⬇</div>Name</div>	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	9 B	Nov 10 11:28	1	128 MB	000000_0	

Showing 1 to 1 of 1 entries

Previous

1

Next

3. 总结

- Hive SQL语法和标准SQL很类似,使得学习成本降低不少。
- Hive底层是通过MapReduce执行的数据插入动作,所以速度慢。
- 如果大数据集这么一条一条插入的话是非常不现实的，成本极高。
- Hive应该具有自己特有的数据插入表方式，结构化文件映射成为表。

附录

无