

# 实 验 报 告 册

2023 学 年 春 季 学 期

专业班级： 计科 6 班

学 号： 2020204331

姓 名： 王博文

指导教师： 纪俊

青岛大学计算机科学技术学院

实验项目	Hive 数据库操作实例	日 期 期	2023.6.5
实验环境	LINUX	成绩评定	
<p>实验目的和要求：</p> <p>搭建 hive 并实现 wordcount 计算单词个数的操作</p> <p>参考网站：<a href="#">Hive3.1.2 安装指南 厦大数据库实验室博客 (xmu.edu.cn)</a>  <a href="#">Ubuntu 安装 MySQL 及常用操作 厦大数据库实验室博客 (xmu.edu.cn)</a></p>			

实验内容或结果:

1. 安装 hive, 我安装的是 mysql-5.7, hive-3.1.2

1. 下载并解压 Hive 安装包

首先需要下载 Hive 安装包文件, Hive 官网下载地址

也可以直接点击[这里](#)从百度云盘下载软件(提取码: ziyu)。进入百度网盘后, 进入“软件”目录, 找到 apache-hive-3.1.2-bin.tar.gz 文件, 下载到本地。

sudo tar -zxvf ./apache-hive-3.1.2-bin.tar.gz -C /usr/local # 解压到/usr/local 中

cd /usr/local/

sudo mv apache-hive-3.1.2-bin hive # 将文件夹名改为 hive

sudo chown -R dblab:dblab hive # 修改文件权限

Shell 命令

注意, 上面的 dblab:dblab 是用户组 and 用户名, 如果你当前使用用户名 hadoop 登录了 Linux 系统, 则把 dblab 替换成 hadoop。

2. 配置环境变量

为了方便使用, 我们把 hive 命令加入到环境变量中去, 请使用 vim 编辑器打开 .bashrc 文件, 命令如下:

vim ~/.bashrc

Shell 命令

在该文件最前面一行添加如下内容:

export HIVE\_HOME=/usr/local/hive

export PATH=\$PATH:\$HIVE\_HOME/bin

export HADOOP\_HOME=/usr/local/hadoop

HADOOP\_HOME 需要被配置成你机器上 Hadoop 的安装路径, 比如这里是安装在 /usr/local/hadoop 目录。

保存退出后, 运行如下命令使配置立即生效:

source ~/.bashrc

Shell 命令

3. 修改 /usr/local/hive/conf 下的 hive-site.xml

执行如下命令:

cd /usr/local/hive/conf

mv hive-default.xml.template hive-default.xml

Shell 命令

上面命令是将 hive-default.xml.template 重命名为 hive-default.xml;

然后, 使用 vim 编辑器新建一个配置文件 hive-site.xml, 命令如下:

cd /usr/local/hive/conf

vim hive-site.xml

Shell 命令

在 hive-site.xml 中添加如下配置信息:

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>javax.jdo.option.ConnectionURL</name>

```

<value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true</value>
  <description>JDBC connect string for a JDBC metastore</description>
</property>
<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>com.mysql.jdbc.Driver</value>
  <description>Driver class name for a JDBC metastore</description>
</property>
<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>hive</value>
  <description>username to use against metastore database</description>
</property>
<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>hive</value>
  <description>password to use against metastore database</description>
</property>
</configuration>

```

然后，按键盘上的“ESC”键退出 vim 编辑状态，再输入:wq，保存并退出 vim 编辑器。

## 二、安装并配置 mysql

这里我们采用 MySQL 数据库保存 Hive 的元数据，而不是采用 Hive 自带的 derby 来存储元数据。

1.Ubuntu 下 mysql 的安装请参考：Ubuntu 安装 MySQL

2.下载 mysql jdbc 包,下载地址

```
tar -zxvf mysql-connector-java-5.1.40.tar.gz #解压
```

```
cp mysql-connector-java-5.1.40/mysql-connector-java-5.1.40-bin.jar
/usr/local/hive/lib #将 mysql-connector-java-5.1.40-bin.jar 拷贝到/usr/local/hive/lib
目录下
```

Shell 命令

3. 启动并登陆 mysql shell

```
service mysql start #启动 mysql 服务
```

```
mysql -u root -p #登陆 shell 界面
```

Shell 命令

4. 新建 hive 数据库。

```
mysql> create database hive; #这个 hive 数据库与 hive-site.xml 中
localhost:3306/hive 的 hive 对应，用来保存 hive 元数据
```

```
mysql
```

5. 配置 mysql 允许 hive 接入：

```
mysql> grant all on *.* to hive@localhost identified by 'hive'; #将所有数据
库的所有表的所有权限赋给 hive 用户，后面的 hive 是配置 hive-site.xml 中配置
的连接密码
```

```
mysql> flush privileges; #刷新 mysql 系统权限关系表
```

```
mysql
```

6. 启动 hive

启动 hive 之前，请先启动 hadoop 集群。

```
start-dfs.sh #启动 Hadoop 的 HDFS
```

```
hive #启动 hive
```

Shell 命令

注意，我们这里已经配置了 PATH，所以，不要把 start-all.sh 和 hive 命令的路径加上。如果没有配置 PATH，请加上路径才能运行命令，比如，本教程 Hadoop 安装目录是“/usr/local/hadoop”，Hive 的安装目录是“/usr/local/hive”，因此，启动 hadoop 和 hive，也可以使用下面带路径的方式：

```
cd /usr/local/hadoop #进入 Hadoop 安装目录
```

```
./sbin/start-dfs.sh
```

```
cd /usr/local/hive
```

```
./bin/hive
```

```
cd /usr/local/hive
```

```
./bin/schematool -dbType mysql -initSchema
```

Shell 命令

执行后，再启动 Hive，应该就正常了。

启动进入 Hive 的交互式执行环境以后，会出现如下命令提示符：

```
hive>
```

```
hive
```

可以在里面输入 SQL 语句，如果要退出 Hive 交互式执行环境，可以输入如下命令：

```
hive>exit;
```

2. 由于 hive 是部署在 hdfs 上的组件，故打开 hive 之前要打开 hdfs 和 yarn 框架来调度资源，而 hive 产生的元数据存储到 mysql 中，故打开 hdfs 和 yarn 之后，且在打开 hive 之前要打开 mysql。

3. 创建名为 hh 的数据库并使用

```
hive> create database hh;
OK
Time taken: 0.298 seconds
hive> use hh;
OK
Time taken: 0.129 seconds
```

```
hive> show tables;
OK
Time taken: 0.138 seconds
```

4. 创建名为 docs 的表格，运行结果如下：

```
hive> create table docs(line string);
OK
Time taken: 1.014 seconds
```

5. 将圣经这个文件中的数据装载进 docs 表中

6. 最后一步，将各词汇装进 word\_count 表中，以空格划分

```
create table word_count as

select word, count(1) as count from

(select explode(split(line,' '))as word from docs) w

group by word

order by word;
```

```
Query ID = hadoop_20230605154831_62faf241-4610-48f6-8d11-a85516e590db
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2023-06-05 15:48:40,103 Stage-1 map = 0%,  reduce = 0%
2023-06-05 15:48:44,127 Stage-1 map = 100%,  reduce = 0%
2023-06-05 15:48:47,251 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local932625123_0001
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2023-06-05 15:48:49,321 Stage-2 map = 0%,  reduce = 0%
2023-06-05 15:48:50,345 Stage-2 map = 100%,  reduce = 0%
```

7. 执行完成后，用 select 语句查看结果如下

```
select * from word_count;
```

♦No,?replied	1	
♦No,?♦No.?Anything		1
♦No.?2	1	
♦No.?John	1	
♦No.♦♦	1	
♦None	2	
♦Not	8	
♦Nothing		2
♦Now	27	
♦Number	1	
♦O	54	
♦O,	1	
♦Obey	3	
♦Of	1	
♦Offer	1	
♦Oh	1	
♦Oh,	10	
♦Old	1	
♦On	11	
♦Once	3	
♦One	4	
♦Only	5	
♦Open	2	
♦Or	1	
♦Order	2	
♦Otherwise		2
♦Our	14	
♦Out	2	
♦Pasture		1
♦Paul	2	
♦Pay	3	
♦Peace	9	
♦Peace, 2		
♦Peace,?when		1
♦Peoples		1
♦Perform		1
♦Perhaps		4

```

♦sinners?Galatians      1
♦sky.?And              1
♦so      9
♦sons    2
♦stand   1
♦super-apostles,?even   1
♦super-apostles.?2     1
♦surely  1
♦take    1
♦tear    1
♦that    5
♦the     17
♦their   1
♦then    2
♦there   1
♦therefore      1
♦they     5
♦this     4
♦though  1
♦throw    2
♦to       1
♦twenty  1
♦understand     1
♦unless  3
♦until    2
♦we       19
♦what     9
♦when    10
♦where    3
♦who      3
♦why      6
♦will     2
♦with     3
♦woman,?for      1
♦you      15
♦your     9
♦♦Do      1
♦♦Love    2
Time taken: 0.687 seconds, Fetched: 61818 row(s)

```



思考或体会：

这个 hive 的问题也还是不少的

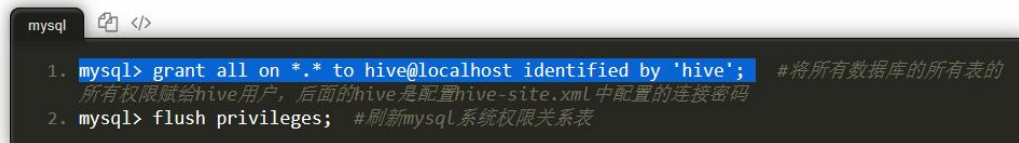
1.在刚开始装 mysql 时，用的是教程里的操作，但我的 ubuntu 是 20.4 的，执行指令 `sudo apt-get update` #更新软件源

`sudo apt-get install mysql-server` 后，我看到我下的版本是 8.0，但当时也就没多想，以为一般软件都能向下兼容，所以就继续做了下去。

教程中的有一步要改变编码方式，进入 `mysql -u root -p` (mysql shell) 内部，但需要密码。装 8.0 时不会像 5.7 一样弹出来问你密码，他的密码是随机的，所以需要进去查看密码然后更改他，这是我完成此目的的网站 [https://blog.csdn.net/m0\\_70885101/article/details/127414184](https://blog.csdn.net/m0_70885101/article/details/127414184)

2.换成了自己的密码后就继续按照教程走，把 mysql jdbc 解压，直到做这步

#### 5. 配置mysql允许hive接入：



```
mysql <img alt="copy icon" data-bbox="225 382 238 392"/> <img alt="code icon" data-bbox="242 382 258 392"/>
1. mysql> grant all on *.* to hive@localhost identified by 'hive'; #将所有数据库的所有表的
   所有权限赋给hive用户，后面的hive是配置hive-site.xml中配置的连接密码
2. mysql> flush privileges; #刷新mysql系统权限关系表
```

发现怎样都运行失败，因此我往回倒退，一步一步寻找问题，最终发现是版本的问题，因为是用指令下载的，所以就上网搜索

彻底删除 mysql

<https://blog.csdn.net/sixiangfy/article/details/128374171>

最终把 mysql 彻底删除，然后重新安装 5.7 版本的 mysql

<https://blog.csdn.net/liuhuango123/article/details/128264867>

备注：

无