

大数据数据库系统

6.2 Hive的安装

6.2 Hive的安装

◆ 主要内容

6.2.1 Hive的安装

6.2.2 配置Hive元数据存储为MySQL

6.2 Hive的安装

◆ 一、下载并解压Hive安装包

1、首先需要下载Hive安装包文件，下载地址：

<http://www.apache.org/dyn/closer.cgi/hive/>

- 提供的是“apache-hive-3.1.2-bin.tar.gz”文件，将其拷贝到linux下，假定为./目录下

6.2 Hive的安装

2、打开一个linux终端，输入：

- `sudo tar -zxvf ./apache-hive-3.1.2-bin.tar.gz -C /usr/local`

解压压缩包到/usr/local中，sudo表示使用超级用户权限执行命令

tar -zxvf 表示解压，-C后跟的是解压缩到目标路径

```
hadoop@ubuntu:~$ ls
apache-hive-3.1.2-bin.tar.gz  Documents  Music      Public      Videos
Desktop                      Downloads  Pictures   Templates   workspace
hadoop@ubuntu:~$ sudo tar -zxvf ./apache-hive-3.1.2-bin.tar.gz -C /usr/local
[sudo] password for hadoop:
```

执行之后，要求输入密码，输入了正确密码后回车，开始解压

```
apache-hive-3.1.2-bin/hcatalog/share/webhcat/ivr/lib/hive-webhcat-3.1.2.jar
apache-hive-3.1.2-bin/hcatalog/share/webhcat/ivr/lib/wadl-resourcedoc-doclet-1.4.jar
apache-hive-3.1.2-bin/hcatalog/share/webhcat/ivr/lib/xercesImpl-2.9.1.jar
apache-hive-3.1.2-bin/hcatalog/share/webhcat/ivr/lib/xml-apis-1.3.04.jar
apache-hive-3.1.2-bin/hcatalog/share/webhcat/ivr/lib/commons-exec-1.1.jar
apache-hive-3.1.2-bin/hcatalog/share/webhcat/ivr/lib/jul-to-slf4j-1.7.10.jar
apache-hive-3.1.2-bin/hcatalog/share/webhcat/java-client/hive-webhcat-java-client-3.1.2.jar
hadoop@ubuntu:~$
```

6.2 Hive的安装

3、进入刚刚解压的目录/usr/local/下，查看并修改apache-hive-3.1.2-bin目录名为hive，输入命令如下：

- `cd /usr/local/`
- `sudo mv apache-hive-3.1.2-bin hive`

```
hadoop@ubuntu:/usr/local$ ls
apache-hive-3.1.2-bin  eclipse  games  hbase  lib  sbin  src
bin                   etc      hadoop  include  man  share
hadoop@ubuntu:/usr/local$ sudo mv apache-hive-3.1.2-bin hive
```

4、修改hive目录的权限，输入

- `sudo chown -R hadoop hive`

将当前hive目录的拥有者改为“hadoop”，如果用户名为其他则将“hadoop”改成对应的用户名

-R表示处理指定目录以及其子目录下的所有文件

6.2 Hive的安装

5、配置环境变量，为了方便使用，把hive命令加入到环境变量中去
使用vim编辑器打开.bashrc文件，命令如下：

- vim ~/.bashrc

在该文件最前面一行添加如下内容：

```
export HIVE_HOME=/usr/local/hive
```

```
export HADOOP_HOME=/usr/local/hadoop
```

```
export PATH=$PATH:$HIVE_HOME/bin
```

```
export PATH=$PATH:$HADOOP_HOME/sbin
```

这里的HADOOP_HOME表示hadoop的安装路径，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

A terminal window with a dark background and light-colored text. The title bar shows standard window controls and menu items: '终端' (Terminal), '文件(F)' (File), '编辑(E)' (Edit), '查看(V)' (View), '搜索(S)' (Search), '终端(T)' (Terminal), and '帮助(H)' (Help). The terminal content shows the output of running 'cat ~/.bashrc'. The first line is a comment: ' ~/.bashrc: executed by bash(1) for non-login shells.'. This is followed by several 'export' statements for environment variables: HIVE_HOME, PATH (updated with HIVE_HOME/bin), HADOOP_HOME, JAVA_HOME, JRE_HOME, CLASSPATH, and PATH (updated with JAVA_HOME/bin). The final two lines are comments: '# see /usr/share/doc/bash/examples/startup-files (in the' and '# for examples'.

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
 ~/.bashrc: executed by bash(1) for non-login shells.
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_HOME=/usr/local/hadoop
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_162
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
# see /usr/share/doc/bash/examples/startup-files (in the
# for examples
```


6.2 Hive的安装

6、用vim打开.profile文件，命令如下

- `vim ~/.profile`

在文件开头加上：

`source ~/.bashrc`

```
# ~/.profile: executed by the command interpreter when login
# This file is not read by bash(1), if ~/.bashrc exists.
# see /usr/share/doc/bash/examples/startup-files for more details
# the files are located in the bash-doc package

# the default umask is set in /etc/profile
# for ssh logins, install and configure the sshd_config file
#umask 022

# if running bash
source ~/.bashrc
if [ -n "$BASH_VERSION" ]; then
```

保存并退出vim编辑器，输入

- `source ~/.bashrc`

使修改立即生效，接着重启shell

6.2 Hive的安装

7、进入/usr/local/hive/conf目录下，使用ls观察里面的文件

```
hadoop@ubuntu:/usr/local$ cd hive
hadoop@ubuntu:/usr/local/hive$ cd conf
hadoop@ubuntu:/usr/local/hive/conf$ ls
beeline-log4j2.properties.template      ivysettings.xml
hive-default.xml.template                llap-cli-log4j2.properties.template
hive-env.sh.template                    llap-daemon-log4j2.properties.template
hive-exec-log4j2.properties.template    parquet-logging.properties
hive-log4j2.properties.template
hadoop@ubuntu:/usr/local/hive/conf$
```

➤ 有一个hive-default.xml.template文件，该文件为hive的配置文件hive-default.xml的模板文件，将该文件进行重命名为hive-default.xml，输入命令：

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

hive-default.xml是hive的**关键配置**，所以一般不要直接修改，拷贝hive-default.xml为**hive-site.xml**文件，将修改的内容在这个里面配置

6.2 Hive的安装

8、由于conf目录下没有hive-site.xml文件，因此，我们新建一个hive-site.xml文件，输入命令：

- `vim hive-site.xml`

并在 hive-site.xml中添加如下信息，见下页，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

```
<?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?createDatabaseIf
      NotExist=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>
```

6.2 Hive的安装

◆ 添加的四个配置含义

javax.jdo.option.ConnectionURL

表示元数据存储方式及位置

javax.jdo.option.ConnectionDriverName

MySQL的JDBC driver的名字

javax.jdo.option.ConnectionUserName

连接MySQL使用的用户名

javax.jdo.option.ConnectionPassword

连接MySQL使用的密码

以上两项均设置为“hive”

Config Param	Config Value	Comment
javax.jdo.option.ConnectionURL	jdbc:mysql://<host name>/<database name>? createDatabaseIfNotExist=true	metadata is stored in a MySQL server
javax.jdo.option.ConnectionDriverName	com.mysql.jdbc.Driver	MySQL JDBC driver class
javax.jdo.option.ConnectionUserName	<user name>	user name for connecting to MySQL server
javax.jdo.option.ConnectionPassword	<password>	password for connecting to MySQL server

6.2.2 配置Hive元数据存储为MySQL

◆ HIVE的元数据存储

进入HIVE的安装目录，/usr/local/hive下，查看里面的文件

```
drwxrwxr-x. 3 atguigu atguigu 4096 10月 24 10:35 bin
drwxrwxr-x. 2 atguigu atguigu 4096 10月 24 10:35 conf
-rw-rw-r--. 1 atguigu atguigu 22187 10月 24 10:48 derby.log
drwxrwxr-x. 4 atguigu atguigu 4096 10月 24 10:35 examples
drwxrwxr-x. 7 atguigu atguigu 4096 10月 24 10:35 hcatalog
drwxrwxr-x. 2 atguigu atguigu 4096 10月 24 10:35 jdbc
drwxrwxr-x. 4 atguigu atguigu 20480 10月 24 10:38 lib
-rwxr-xr-x. 1 atguigu atguigu 20798 8月 23 2019 LICENSE
drwxrwxr-x. 5 atguigu atguigu 4096 10月 24 10:48 metastore_db
-rwxr-xr-x. 1 atguigu atguigu 230 8月 23 2019 NOTICE
-rwxr-xr-x. 1 atguigu atguigu 2469 8月 23 2019 RELEASE_NOTES.txt
drwxrwxr-x. 4 atguigu atguigu 4096 10月 24 10:35 scripts
```

可以看到里面有个derby.log文件

实际上HIVE的元数据是存储在derby数据库中的

6.2.2 配置Hive元数据存储为MySQL

◆ Derby数据库

HIVE默认采用Hive自带的derby来存储元数据

Apache Derby是一个完全用java编写的数据库，Derby是一个Open source的产品，基于Apache License 2.0分发。

由于Derby是由纯java开发的，因此也与生俱来具有了java的跨平台性，可以很好地工作在各种操作系统上。Apache Derby非常小巧，核心部分derby.jar只有2M，所以既可以做为单独的数据库服务器使用，也可以内嵌在应用程序中使用。



官方网站：<http://db.apache.org/derby/>

6.2.2 配置Hive元数据存储为MySQL

◆ Derby来储元数据的缺陷

HIVE默认采用Hive自带的derby来存储元数据

打开终端1输入命令：

- `cd /usr/local/hive/bin`
- `hive`

等待进入hive

```
Time taken: 0.04 seconds  
hive>
```

打开另一个终端，输入上述命令，也尝试进入hive

```
Caused by: java.sql.SQLException: Unable to open a test connection to the given database. JDBC url = jdbc:derby::databaseName=metastore_db;create=true, username = APP. Terminating connection pool (set lazyInit to true if you expect to start your database after your app). Original Exception: -----  
java.sql.SQLException: Failed to start database 'metastore_db' with class loader sun.misc.Launcher$AppClassLoader@2bb0bf9a, see the next exception for details.
```

```
Caused by: java.sql.SQLException: Another instance of Derby may have already booted the database
```

发现报错，提示：已经有一个实例启动了

6.2.2 配置Hive元数据存储为MySQL

◆ HIVE默认采用Hive自带的derby来存储元数据

缺点：

1、Derby存储使用文件存储：

- 同一时间只能启动一个数据库实例，实际应用中不可能只开一个实例，一张表可能有多个人分析

2、安全性不高，一旦文件出问题，元数据丢失，表等结构都丢失

6.2.2 配置Hive元数据存储为MySQL

◆ 目前HIVE所支持的元数据存放

Database	Minimum Supported Version	Name for Parameter Values
MySQL	5.6.17	mysql
Postgres	9.1.13	postgres
Oracle	11g	oracle
MS SQL Server	2008 R2	mssql

◆ 我们采用MySQL存储metastore

6.2.2 配置Hive元数据存储为MySQL

1、安装MySQL

使用以下命令即可进行mysql安装，注意安装前先更新一下软件源以获得最新版本：

- `sudo apt-get update` #更新软件源
- `sudo apt-get install mysql-server` #安装mysql

启动和关闭mysql服务器：

- `service mysql start`
- `service mysql stop`

6.2.2 配置Hive元数据存储为MySQL

确认是否启动成功，mysql节点处于LISTEN状态表示启动成功：

`sudo netstat -tap | grep mysql`

```
dblab@ubuntu:~$ sudo netstat -tap | grep mysql
tcp        0      0 0 localhost:mysql    *:*              LISTEN
6359/mysql
```

由于MySQL默认显示中文存在乱码的问题：

a. 编辑配置文件: `sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf`

b. 在[mysqld]下添加一行`character_set_server=utf8`

c. 重启MySQL服务: `service mysql restart`

```
[mysqld]
#
# * Basic Settings
#
user                = mysql
pid-file            = /var/run/mysqld/mysqld.pid
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
basedir             = /usr
datadir             = /var/lib/mysql
tmpdir              = /tmp
lc-messages-dir     = /usr/share/mysql
character_set_server=utf8
skip-external-locking
```

6.2.2 配置Hive元数据存储为MySQL

其他MySQL的安装方法和命令可详见

<http://dblab.xmu.edu.cn/blog/install-mysql/>

本课程主要介绍如何使用MySQL存储HIVE元数据

1、下载mysql jdbc 包，我们也提供，也可以从网站下载

<http://www.mysql.com/downloads/connector/j/>

2、解压mysql-connector-java-5.1.40.tar.gz，命令如下：

- tar -zxvf mysql-connector-java-5.1.40.tar.gz

```
hadoop@ubuntu:~$ sudo tar -zxvf mysql-connector-java-5.1.40.tar.gz  
[sudo] password for hadoop: 
```

到mysql-connector-java-5.1.40目录下，看到有个mysql-connector-java-5.1.40-bin.jar 文件

```
hadoop@ubuntu:~$ ls  
apache-hive-3.1.2-bin.tar.gz  Music  Public  
Desktop  mysql-connector-java-5.1.40  Templates  
Documents  mysql-connector-java-5.1.40.tar.gz  Videos  
Downloads  Pictures  workspace
```

将其复制到/usr/local/hive/lib下，执行语句：

- cp mysql-connector-java-5.1.40/mysql-connector-java-5.1.40-bin.jar /usr/local/hive/lib

6.2.2 配置Hive元数据存储为MySQL

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

这里输入在mysql安装时设定的root密码，输入后按回车进入mysql

```
hadoop@ubuntu:/usr/local/hive/conf$ mysql -u root -p
Enter password: █
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.29-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

6.2.2 配置Hive元数据存储为MySQL

◆ 如果忘记了mysql的登录名和密码

- 1、可能mysql版本号为5.7以上，安装没提示要设置root密码，如下方法进行查看

```
hadoop@hadoop-virtual-machine:/etc/mysql$ sudo cat debian.cnf
# Automatically generated for Debian scripts. DO NOT TOUCH!
[client]
host      = localhost
user      = debian-sys-maint
password  = L81FXiWtF1nb1FmN
socket    = /var/run/mysqld/mysqld.sock
[mysql_upgrade]
host      = localhost
user      = debian-sys-maint
password  = L81FXiWtF1nb1FmN
socket    = /var/run/mysqld/mysqld.sock
```

◆ 重新设置密码：

1、使用查看到的用户名密码进入mysql；

2、输入如下指令：

```
mysql> update mysql.user set authentication_string=PASSWORD("123456") where User='root';
```

```
mysql> update mysql.user set plugin="mysql_native_password";
```

```
mysql> flush privileges;
```

```
mysql> quit;
```

3、先关闭服务再重启启动MySQL

```
sudo /etc/init.d/mysql stop
```

```
sudo /etc/init.d/mysql start
```

再用root用户和密码进去看看

```
mysql -u root -p
```

6.2.2 配置Hive元数据存储为MySQL

4、新建hive数据库，输入如下指令：

```
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

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> create database hive;  
Query OK, 1 row affected (0.00 sec)
```

```
<property>  
  <name>javax.jdo.option.ConnectionURL</name>  
  <value>jdbc:mysql://localhost:3306/hive?createDatabaseIf  
    NotExist=true</value>  
  <description>JDBC connect string for a JDBC  
    metastore</description>  
</property>
```

6.2.2 配置Hive元数据存储为MySQL

5、配置mysql允许hive接入，输入以下命令：

- 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> grant all on *.* to hive@localhost identified by 'hive';
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

6、刷新mysql系统权限关系表，输入指令：

- mysql> flush privileges;

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

```
<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>
```

6.2.2 配置Hive元数据存储为MySQL

6. 启动hive

启动hive之前，请先启动hadoop集群：

- start-all.sh

然后启动HIVE，输入如下命令：

- hive

```
Hive-on-MR is deprecated in Hive 2 and may not be available in the future. Please consider using a different execution engine (i.e. spark, tez) or your hive config file.
Hive Session ID = fcec8119-1dc8-406d-a504-dfb06c583e69
hive> █
```

如果没有出现错误信息，则表示成功启动。

- 我们这里已经配置了PATH，所以，可以不把start-all.sh和hive命令的路径加上
- start-all.sh所在的位置： /usr/local/hadoop/sbin/
- Hive所在位置： /usr/local/hive/bin

`vim ~/.bashrc`

在该文件最前面一行添加如下内容：

```
export HIVE_HOME=/usr/local/hive
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HIVE_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
```


6.2.2 配置Hive元数据存储为MySQL

◆ 启动Hive过程中，可能出现的错误

【错误1】

java.lang.NoSuchMethodError:

com.google.common.base.Preconditions.checkArgument

```
Exception in thread "main" java.lang.NoSuchMethodError: com.google.common.base.Preconditions.checkArgument(ZLjava/lang/String;Ljava/lang/Object;)V
    at org.apache.hadoop.conf.Configuration.set(Configuration.java:1357)
```

【原因】

com.google.common.base.Preconditions.checkArgument 这是因为hive内依赖的guava.jar和hadoop内的版本不一致造成的

【解决方法】

- 1.查看hadoop安装目录下share/hadoop/common/lib内guava.jar版本
- 2.查看hive安装目录下lib内guava.jar的版本 如果两者不一致，删除版本低的，并拷贝高版本的问题解决

【错误1】

java.lang.NoSuchMethodError:

com.google.common.base.Preconditions.checkArgument

解决:

- 1、进入/usr/local/hadoop/share/hadoop/common/lib下，找到guava.jar文件，观察版本，例下图为：27.0

```
error_prone_annotations-2.2.0.jar  
failureaccess-1.0.jar  
gson-2.2.4.jar  
guava-27.0-jre.jar  
hadoop-annotations-3.1.3.jar  
hadoop-auth-3.1.3.jar  
htrace-core4-4.1.0-incubating.jar
```

- 2、进入/usr/local/hive/lib下找到guava.jar文件，观察版本，例下图为19.0:

```
groovy-all-2.4.11.jar  
gson-2.2.4.jar  
guava-19.0.jar  
hbase-client-2.0.0-alpha4.jar  
hbase-common-2.0.0-alpha4.jar
```

- 3、将27.0的替换掉19.0，删掉19.0的版本

```
hadoop@ubuntu:/usr/local/hive/lib$ rm guava-19.0.jar  
hadoop@ubuntu:/usr/local/hive/lib$ sudo cp /usr/local/hadoop/share/hadoop/common  
/lib/guava-27.0-jre.jar ./  
[sudo] password for hadoop:
```

6.2.2 配置Hive元数据存储为MySQL

【错误2】 在启动Hive时，有可能会出现Hive metastore database is not initialized的错误，这里给出解决方案。

【解决方案】

解决方法：Hive现在包含一个用于 Hive Metastore 架构操控的脱机工具，名为 schematool.此工具可用于初始化当前 Hive 版本的 Metastore 架构。在终端执行如下命令：

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

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

6.2.2 配置Hive元数据存储为MySQL

【错误3】

org.datanucleus.store.rdbms.exceptions.MissingTableException:

Required table missing : "VERSION" in Catalog "" Schema "".

DataNucleus requires this table to perform its persistence operations.

```
2020-03-06 18:33:13,534 WARN DataNucleus.Query: Query for candidates of org.apache.hadoop.hive.metastore.model.MVersionTable and subclasses resulted in no possible candidates
Required table missing : "`VERSION`" in Catalog "" Schema "". DataNucleus requires this
table to perform its persistence operations. Either your MetaData is incorrect, or you
need to enable "datanucleus.schema.autoCreateTables"
org.datanucleus.store.rdbms.exceptions.MissingTableException: Required table missing :
```

【解决方案】

进入hive安装目录（比如/usr/local/hive），执行如下命令：

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

```
hadoop@ubuntu:/usr/local/hive$ ./bin/schematool -dbType mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/
```

6.2.2 配置Hive元数据存储为MySQL

◆ 再来测试下是否可以同时开启多个HIVE实例

- 1、建立一个表test，表只包含id字段，使用create table命令；

```
hive> create table test(id string);  
OK  
Time taken: 0.645 seconds
```



- 2、新开一个终端，建立一个id.txt文件，里面输入如下内容（两行：1002和1003），保存后退出：

```
1002  
1003
```

```
~  
~  
~  
~  
~  
~  
~
```


3、将id.txt文件放置于hdfs下的/usr/hive/warehouse/test/目录下，输入如下命令：

- Hadoop fs -put id.txt /usr/hive/warehouse/test

4、在hive中查看刚刚创建的test表的内容，发现已经有数据了

```
hive> select * from test;
OK
1002
1003
Time taken: 0.18 seconds,
hive> █
```

5、再新开一个终端，进入hive，查询刚刚我们创建的表，发现成功并且没有报错

```
[atguigu@hadoop102 hive]$ bin/hive
which: no hbase in (/usr/local/bin:
in:/opt/module/hadoop-3.1.3/bin:/op
/.local/bin:/home/atguigu/bin)
Hive Session ID = 7bddb8a6-7801-48e

Logging initialized using configura
hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2
ng a different execution engine (i.
Hive Session ID = e9764475-e254-464
hive> show tables;
OK
test
Time taken: 0.654 seconds, Fetched:
hive> select * from test;
OK
1002
1003
Time taken: 1.727 seconds, Fetched:
hive>
```


总结

- ◆ HIVE安装

- ◆ HIVE元数据

 - Derby的缺陷

- ◆ 配置Hive元数据存储为MySQL

- ◆ HIVE部分启动错误的解决