安装 Hive

1. 内嵌模式配置

a)

```
[root@BigData201916071072 local]# ls
apache-hive-1.2.1-bin.tar.gz  etc    hadoop                  hadoop-2.7.7.tar.gz  lib    libexec  share
bin                           games  hadoop-2.7.7-src.tar.gz  include              lib64  sbin     src
[root@BigData201916071072 local]# tar zxvf apache-hive-1.2.1-bin.tar.gz -C /usr/local
apache-hive-1.2.1-bin/NOTICE
apache-hive-1.2.1-bin/LICENSE
apache-hive-1.2.1-bin/README.txt
apache-hive-1.2.1-bin/RELEASE_NOTES.txt
apache-hive-1.2.1-bin/examples/files/emp.txt
apache-hive-1.2.1-bin/examples/files/type_evolution.avro
```

b) 更改文件名

```
[root@BigData201916071072 local]# mv apache-hive-1.2.1-bin hive
[root@BigData201916071072 local]# ls
apache-hive-1.2.1-bin.tar.gz  etc    hadoop                  hadoop-2.7.7.tar.gz  include  lib64  sbin   src
bin                           games  hadoop-2.7.7-src.tar.gz  hive                 lib      libexec  share
[root@BigData201916071072 local]#
```

c) 配置环境变量、启动 Hive 并查看 hive 版本

```
[root@BigData201916071072 local]# vi /etc/profile
[root@BigData201916071072 local]#  89L, 2451C written
[root@BigData201916071072 local]# source /etc/profile
[root@BigData201916071072 local]# hive

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-1.2.1.jar!/hive-log4j.properties
hive>
```

```
[root@BigData201916071072 local]# hive --version
Hive 1.2.1
Subversion git://localhost.localdomain/home/sush/dev/hive.git -r 243e7c1ac39cb7ac8b65c5bc6988f5cc3162f558
Compiled by sush on Fri Jun 19 02:03:48 PDT 2015
From source with checksum ab480aca41b24a9c3751b8c023338231
```

2. 本地模式配置

2.1 安装 mysql

a) 将 mysql 安装包解压并安装相应组件

```
[root@BigData201916071072 local]# sudo rpm -ivh mysql-community-server-5.7.22-1.el6.x86_64.rpm  --force --nodep
警告: mysql-community-server-5.7.22-1.el6.x86_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 5072e1f5: NOKEY
准备中...                          ################################# [100%]
正在升级/安装...
   1:mysql-community-server-5.7.22-1.e##### ( 16%)
```

b) 登录 mysql

```
[root@BigData201916071072 ~]# sudo mysqld_safe --skip-grant-tables &
[1] 13796
[root@BigData201916071072 ~]# 2019-06-15T07:03:59.575807Z mysqld_safe Logging to '/var/log/mysqld.lo
2019-06-15T07:03:59.614006Z mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql

[root@BigData201916071072 ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.22 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, 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.
```

c) 修改密码

```
mysql> select user from mysql.user;
+---------------+
| user          |
+---------------+
| mysql.session |
| mysql.sys     |
| root          |
+---------------+
3 rows in set (0.00 sec)
```

```
mysql> update mysql.user set authentication_string=password('&Shieshuyuan21') where user='root';
Query OK, 1 row affected, 1 warning (0.00 sec)
```

（后续因为符号&在 xml 中会出现问题，我就把密码改成 123456@Hzd 了）

d) 创建 hive-site.xml 并根据上述 mysql 的用户密码修改相应的值

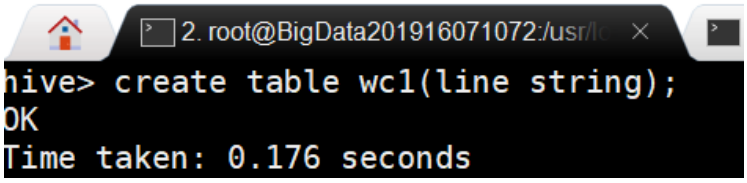
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>javax.jdo.option.ConnectionURL</name>
        <value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true</value>
        <description>JDBC 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>root</value>
        <description>username to use against metastore database</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionPassword</name>
        <value>123456@Hzd</value>
        <description>password to use against metastore database</description>
    </property>
</configuration>
~
```

3. 运行 hive 实例

a) 上传 hadoop/mydata.txt 到 hdfs 的/user/root/file/中

```
[root@BigData201916071072 hadoop]# ls
bin  include  lib      LICENSE.txt  mydata.txt  README.txt  share
etc  input    libexec  logs         NOTICE.txt  sbin        tmp
[root@BigData201916071072 hadoop]# ./bin/hdfs dfs -mkdir -p /user/root/file
[root@BigData201916071072 hadoop]# ./bin/hdfs dfs -put ./mydata.txt /user/root/file/
[root@BigData201916071072 hadoop]#
```

b) 在 hive 中创建表 wc1



c) 将 hdfs 中保存 mydata.txt 载入到 wc1 表

```
hive> load data inpath '/user/root/file/mydata.txt' overwrite into table w
Loading data to table default.wc1
Table default.wc1 stats: [numFiles=1, numRows=0, totalSize=26, rawDataSize
OK
Time taken: 0.241 seconds
hive> select * from wc1;
OK
hello word
hello big data
Time taken: 0.055 seconds, Fetched: 2 row(s)
```

d) 使用 HQL 运行 count 函数：

```
hive> select word,count(*) from( select explode(split(line,' ')) as word from wc1) t g
roup by word;
Query ID = root_20190615214256_041d2207-d91f-4a65-ba6f-44a9c27369e0
Total jobs = 1
Launching Job 1 out of 1
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>
Starting Job = job_1560605406223_0002, Tracking URL = http://BigData201916071072.lab.b
eihangsoft.cn:8088/proxy/application_1560605406223_0002/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1560605406223_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-06-15 21:43:04,424 Stage-1 map = 0%,  reduce = 0%
2019-06-15 21:43:09,692 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.66 sec
2019-06-15 21:43:14,940 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.15 sec
MapReduce Total cumulative CPU time: 3 seconds 150 msec
Ended Job = job_1560605406223_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.15 sec   HDFS Read: 7387 HDFS Wri
te: 28 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 150 msec
OK
big     1
data    1
hello   2
word    1
Time taken: 19.733 seconds, Fetched: 4 row(s)
```

e) 将上一步查询语句返回的结果保存到 wordnum 中

```
hive> create table wordnumber as select word,count(*) from( select explode(split(line,' ')) as word from wc1) t group by word
```

```
hive> select * from wordnumber;
OK
big     1
data    1
hello   2
word    1
Time taken: 0.044 seconds, Fetched: 4 row(s)
```

f) 在 mysql 中的 hive 数据库中查看在相应表的信息

```
mysql> select * from TBLS;
+--------+-------------+-------+------------------+-------+-----------+-------+-----------+---------------+--------------------+
| TBL_ID | CREATE_TIME | DB_ID | LAST_ACCESS_TIME | OWNER | RETENTION | SD_ID | TBL_NAME  | TBL_TYPE      | VIEW_EXPANDED_TE
| VIEW_ORIGINAL_TEXT |
+--------+-------------+-------+------------------+-------+-----------+-------+-----------+---------------+--------------------+
|      1 |  1560589883 |     1 |                0 | root  |         0 |     1 | words     | MANAGED_TABLE | NULL
| NULL               |
|      6 |  1560591000 |     1 |                0 | root  |         0 |     6 | words1    | MANAGED_TABLE | NULL
| NULL               |
|     11 |  1560591288 |     1 |                0 | root  |         0 |    11 | wordscount | MANAGED_TABLE | NULL
| NULL               |
|     16 |  1560591897 |     1 |                0 | root  |         0 |    16 | mywords   | MANAGED_TABLE | NULL
| NULL               |
|     21 |  1560598317 |     1 |                0 | root  |         0 |    21 | wc        | MANAGED_TABLE | NULL
| NULL               |
|     26 |  1560605962 |     1 |                0 | root  |         0 |    26 | wc1       | MANAGED_TABLE | NULL
| NULL               |
|     27 |  1560607152 |     1 |                0 | root  |         0 |    27 | wordnumber | MANAGED_TABLE | NULL
| NULL               |
+--------+-------------+-------+------------------+-------+-----------+-------+-----------+---------------+--------------------+
7 rows in set (0.00 sec)
```

遇到的问题及解决方案：

1. Hadoop 中 jline 的版本过旧，把/hive/lib/jline-2.12.jar 替换原来的 jline-*.jar，替换后如下：

```
[root@BigData201916071072 local]# find . -name jline*
./hadoop/lib/jline-2.12.jar
./hadoop/share/hadoop/kms/tomcat/webapps/kms/WEB-INF/lib/jline-2.12.jar
./hadoop/share/hadoop/httpfs/tomcat/webapps/webhdfs/WEB-INF/lib/jline-2.12.jar
./hive/lib/jline-2.12.jar
```

2. 在 hive 中运行 count 时，显示：

```
java.net.ConnectException: Call From BigData201916071072.lab.beihangsoft.cn/127.0.1.1
to 0.0.0.0:8032 failed on connection exception: java.net.ConnectException: 拒绝连接; F
or more details see:  http://wiki.apache.org/hadoop/ConnectionRefused
```

解决方法：./sbin/stop-all.sh，格式化 namenode，运行./sbin/start-all.sh

回答实验要求中的问题（若无则无需填写）：

1. Hive 有几种模式？区别在哪？

（1） 内嵌 Derby 模式：内嵌模式使用的是内嵌的 Derby 数据库来存储元数据，也不需要额外起 Metastore 服务。这个是默认的，配置简单，但是一次只能一个客户端连接，适用于用来实验，不适用于生产环境。

（2） Local 模式：采用外部数据库存储元数据本地元。存储不需要单独起 metastore 服务，用的是跟 hive 在同一个进程里的 metastore 服务。

（3） Remote 模式：使用外部数据库（如 MySQL、Postgres、Oracle、MS SQL Server）存储元数据。远程元存储需要单独起 metastore 服务，然后每个客户端都在配置文件里配置连接到该 metastore 服务。远程元存储的 metastore 服务和 hive 运行在不同的进程里。

2. Hive 各个模式下，元数据和数据分别存储在哪？

数据：都存储在 hdfs 下。

元数据：

内嵌 Derby 模式下：内嵌的 Derby 数据库

Local 模式：外部数据库（如 MySQL、Postgres、Oracle、MS SQL Server，实验中，我们使用的是 MySql）

Remote 模式：外部数据库

3.常用的 hive 命令有哪些？

（1） 进入 hive

option:

-H 帮助

-e 执行 hql 语句

-f 执行 hql 文件

-p 指定端口

-S 不打印日志

-i 从文件初始化 hql

（2） 2、查看所有表

show tables;

（3） 3、查看是否是分区表

show partitions tableName;

（4） 4、查看表前几条记录

select * from tableName limit 10;

（5） 5、查看表某一分区数据

select * from tableName where 分区字段=分区 limit 10;

（6） 6、创建表(例子)

create table if not exists tableName(name string,age int,sarlar bigint)row format delimited fields terminated by '\t' stored as textfile;

（7） 7、模糊查找表

show tables like '*tableName*';

（8） 8.创建视图

create view viewName select * from tableName;

（9） 9.删除表

drop table tableName;

（10） 就文本数据导入 hive 表

load data local inpath '/usr/data/students.txt' into table studentTable partition(classId=1)