实验：Hive 安装部署及实验

一、实验说明

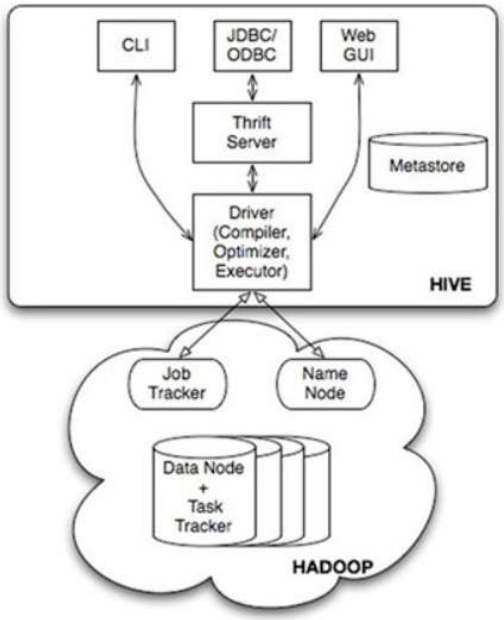
1. 实验环境：Windows10， VMware Workstation 12 Pro，ubuntu-14.04.4-serveramd64.iso，jdkhadoop 2.6.0，XShell 5
2. 实验目的：了解和掌握 Hive 的原理，部署以及使用，完成数据文件的导入与建表实验。
3. 实验文件：
   apache-hive-1.2.0-bin.tar.gz，mysql-connector-java-5.1.44-bin.jar，data.zip

二、关于 Hive

（1） 介绍

Hive 是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具，可以用来**进行数据提取转化加载（ETL）**，这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。它定义了简单的类 SQL 查询语言，称为 HQL，允许熟悉 SQL 的用户查询数据。

（2） 原理



（3） 用户接口

CLI：就是 Shell 命令行；

JDBC：这个是 Hive 的 java 接口，与通常的数据库类似；

WebGUI：网页界面。

（4） 驱动组件（Driver）

Hive 的编译、解析、优化转化为 MapReduce 任务提交给 Hadoop 进行分派和执行相应的任务。

（5） 元数据组件（Metatore）

存储着 hive 的元数据信息，包括表名、列、分区和属性等。默认数据库为 Derby，为了更健壮，一般使用 Mysql 来代替。另外，MetaStore 分为服务端和客户端，服务端提供给客户端对本地 RDBMS 的访问服务。

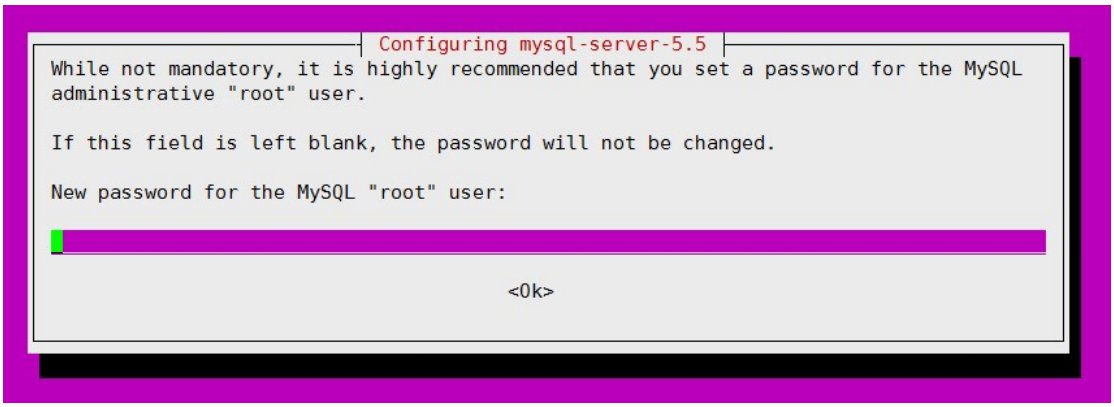
三、安装 Mysql

1. 安装 mysql

（1） 使用以下命令在 Linux 系统下安装 Mysql：

```
sudo apt-get install mysql-server
```

（2） 设置数据库密码



2. 配置 Mysql

（1） 登录 Mysql 的终端：

```
mysql –uroot –p
```

使用初始化时设置的 root 密码登录

（2） 新增 hive 用户，并给于权限：

```
mysql> create user 'hive' identified by 'hive';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on *.* to 'hive' with grant option;
Query OK, 0 rows affected (0.00 sec)

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

mysql>
```

四、Hive 安装与配置

1. 安装 hive

（1） 将 apache-hive-1.2.0-bin.tar.gz 解压在/usr/local 目录下

```
sudo tar –zxvf apache-hive-1.2.0-bin.tar.gz –C /usr/local
```

（2） 重命名文件夹为 hive 文件夹，并将其权限修改成 hadoop

```
mv /usr/local/ apache-hive-1.2.0-bin /usr/local/hive
```

```
sudo chown -R hadoop:hadoop /usr/local/hive
```

```
hadoop@master:/usr/local$ ll
total 52
drwxr-xr-x 13 root   root   4096 Nov  8 18:28 ./
drwxr-xr-x 10 root   root   4096 Oct 12 01:15 ../
drwxr-xr-x  2 root   root   4096 Feb 18  2016 bin/
drwxr-xr-x  2 root   root   4096 Feb 18  2016 etc/
drwxr-xr-x  2 root   root   4096 Feb 18  2016 games/
drwxr-xr-x 12 hadoop hadoop 4096 Oct 26 03:15 hadoop/
drwxr-xr-x  8 hadoop hadoop 4096 Nov  8 17:49 hive/
drwxr-xr-x  2 root   root   4096 Feb 18  2016 include/
drwxr-xr-x  3 root   root   4096 Oct 26 01:28 jvm/
drwxr-xr-x  4 root   root   4096 Oct 12 01:18 lib/
lrwxrwxrwx  1 root   root      9 Oct 12 01:15 man -> share/man/
drwxr-xr-x  2 root   root   4096 Feb 18  2016 sbin/
drwxr-xr-x  6 root   root   4096 Oct 12 01:23 share/
drwxr-xr-x  2 root   root   4096 Feb 18  2016 src/
```

（3） 把 mysql 的 jdbc 的驱动 mysql-connector-java-5.1.44-bin.jar 拷贝到 \usr\local\hive\lib 目录下

```
cp mysql-connector-java-5.1.44-bin.jar /usr/local/hive/lib
```

2. 配置环境变量

配置 HIVE_HOME 以及 PATH 的内容

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

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

可以直接在原来的 setenv.sh 文件中追加如下内容：

```
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/u

export JAVA_HOME=/usr/local/jvm/jdk1.8.0_60
export HADOOP_HOME=/usr/local/hadoop
export HIVE_HOME=/usr/local/hive
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib:$(${HADOOP_HOME}/bin/hadoop classp
export PATH=$PATH:${JAVA_HOME}/bin:${HIVE_HOME}/bin

# parameters for hadoop jar
```

注意用冒号隔开

同样的使用 source 命令使环境变量生效

```
source setenv.sh
```

3. Hive 的配置

说明：hive 有关于 metastore 具有三种配置，分别为内嵌模式、本地元存储以及远程元存储，本实验在 namenode 上配置服务端，datanode 上配置为客户端，在 datanode 进行 hql 时通过远程模式访问在 namenode 的元数据。（实际场景 metastore 的数据库可以在任何节点，以减轻 namenode 的压力）

（1） 在 /usr/local/hive/conf 目录下创建 hive-site.xml 文件：

```
sudo vi /usr/local/hive/conf/hive-site.xml
```

（2）在 server 端配置 hive-stie.xml（在同目录下有个带 template 后缀的 xml 文件，记录了 hive 可修改的各个属性，我们在 hive-site.xml 文件中配置相关所需属性即可生效）

```
<?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/db_hive?createDatabaseIfNotExist=true</value>
        </property>
        <property>
                <name>javax.jdo.option.ConnectionDriverName</name>
                <value>com.mysql.jdbc.Driver</value>
        </property>
        <property>
                <name>javax.jdo.option.ConnectionUserName</name>
                <value>hive</value>
        </property>
        <property>
                <name>javax.jdo.option.ConnectionPassword</name>
                <value>hive</value>
        </property>
        <property>
                <name>hive.metastore.warehouse.dir</name>
                <value>/hive/warehouse</value>
        </property>
        <property>
                <name>hive.metastore.local</name>
                <value>true</value>
        </property>
</configuration>
```

说明：ConnectionURL 属性用于设置 mysql 服务所在地址与端口，这里 mysql-server 在本地， hive.metastore.warehouse.dir 是在 HDFS 上的文件路径，hive.metastore.local 的值为 true 表示对 metastore 的访问为本地模式。

（3） 在 client 端配置 hive-site.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
        <property>
                <name>hive.metastore.warehouse.dir</name>
                <value>/hive/warehouse</value>
        </property>
        <property>
                <name>hive.metastore.local</name>
                <value>false</value>
        </property>
        <property>
                <name>hive.metastore.uris</name>
                <value>thrift://192.168.91.13:9083</value>
        </property>
</configuration>
~
```

说明：hive.metastore.uris 项指向提供数据库访问的 metastore 服务端，值须为 IP 地址。由于设置了 uris 的内容，因而对于 metastore 的访问默认为远程模式。另外若有多个远程元数据服务端。

五、运行 Hive

（1） 检查 jline 版本，hive 与 hadoop 的 jline 版本不对应可能导致运行错误，将 hive 上 jline 的 jar 包拷贝至 hadoop 的对应目录下：

```
cp /usr/local/hive/lib/jline-2.12.jar /usr/local/hadoop/share/hadoop/yarn/lib
```

（2） 更新 yarn-site.xml 配置

重要：为了使得 mapreduce 程序可以在各个节点提交，对各个节点的 yarn-site.xml 配置文件追加以下 property：

```
    <property>
            <name>yarn.resourcemanager.hostname</name>
            <value>192.168.91.13</value>
    </property>
    <property>
            <name>yarn.recourcemanager.address</name>
            <value>master:8032</value>
    </property>
    <property>
            <name>yarn.resourcemanager.schedulaer.address</name>
            <value>master:8030</value>
    </property>
    <property>
            <name>yarn.resourcemanager.resource-tracker.address</name>
            <value>master:8031</value>
    </property>
```

说明：其中 yarn.resourcemanager.hostname 配置的是运行 ResourceManager 进程所在的主机（master 节点）IP 地址。

（3） 启动 hadoop

```
/usr/local/hadoop/sbin/start-all.sh
```

（4） 初始化 Schema

```
/usr/local/hive/bin/./schematool –dbType mysql –initSchema
```

```
hadoop@master:/usr/local/hive/bin$ ./schematool --dbType mysql --initSchema
17/11/11 19:20:17 WARN conf.HiveConf: HiveConf of name hive.metastore.local does not exist
Metastore connection URL:        jdbc:mysql://localhost:3306/db_hive?createDatabaseIfNotExist=true
Metastore Connection Driver :    com.mysql.jdbc.Driver
Metastore connection User:       hive
Starting metastore schema initialization to 1.2.0
Initialization script hive-schema-1.2.0.mysql.sql
Initialization script completed
schemaTool completed
hadoop@master:/usr/local/hive/bin$
```

（5） 服务端启动 metastore 服务

```
hive –service metastore
```

注意 service 前是两个横杠

```
hadoop@master:~$ hive --service metastore
Starting Hive Metastore Server
17/11/12 17:25:56 WARN conf.HiveConf: HiveConf of name hive.metastore.local does not exist
```

（6） 客户端启动 hive:

```
hive
```

```
hadoop@slaver1:~$ hive
17/11/12 17:27:53 WARN conf.HiveConf: HiveConf of name hive.metastore.local does not e

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-1.
og4j.properties
hive>
```

六、实验：导入数据到 hive 数据仓库

说明：把数据转换成 hive 中的表的方式有四种，以下是源数据的格式：

```
10001   1003    c2sld54sfkid    paul
10002   1014    d5dsd32sdkif    rose
10005   1008    e0dlp92mklpi    harry
10203   1099    p3skd2911sie    bob
```

（1） 从本地导入

将 list1 和 list2 上传到客户端节点。

```
hadoop@slaver1:~$ ls
apache-hive-1.2.0-bin.tar.gz  jdk-8u60-linux-x64.tar.gz  list2.txt
hadoop-2.6.0.tar.gz           list1.txt                  setenv.sh
```

登入 hive 客户端，创建 user_info 表

```
hive
create table user_info(uid int, did int, pwd string, uname string)
row format delimited
fields terminated by '\t'
lines terminated by '\n';
```

```
hadoop@slaver1:~$ hive
17/11/12 18:17:08 WARN conf.HiveConf: HiveConf of name hive.metastore.local doe

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-co
og4j.properties
hive> create table user_info(uid int, did int, pwd string, uname string)
    > row format delimited
    > fields terminated by '\t'
    > lines terminated by '\n';
OK
Time taken: 0.928 seconds
hive>
```

把数据加载到表

```
load data local inpath 'list*.txt' into table user_info;
```

```
hive> select * from user_info;
OK
10001   1003    c2sld54sfkid    paul
10002   1014    d5dsd32sdkif    rose
10005   1014    e0dlp92mklpi    harry
10203   1099    p3skd29llsie    bob
20004   2991    e2ifp12adlpi    alice
20132   1014    l8doo32haodp    jerry
50232   3022    d3sod41dsooo    smith
40001   1023    s3dfdlksdfj3    rose
```

（2） 从 HDFS 中导入

与本地的类似，只要把本地载入的命令中’local’去掉，输入路径即为 HDFS 上的路径。（略）

（3） 将查询结果插入到表

说明：这个实验将所有名为 rose 的记录插入到 account_rose 表中

先创建 account_rose 表

```
hive> create table account_rose(uid int, did int, pwd string)
    > row format delimited
    > fields terminated by '\t'
    > lines terminated by '\n';
OK
Time taken: 0.072 seconds
```

插入查询的数据到 account_rose 表中

```
hive> insert into table account_rose
    > select uid,did,pwd
    > from user_info
    > where uname='rose';
Query ID = hadoop_20171112201027_4f584faf-b0fc-4f33-9562-b32e68ef2e02
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1510484978137_0001, Tracking URL = http://master:8088/proxy/applicati
137_0001/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1510484978137_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-11-12 20:10:45,576 Stage-1 map = 0%,  reduce = 0%
2017-11-12 20:11:01,089 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.85 sec
MapReduce Total cumulative CPU time: 1 seconds 850 msec
Ended Job = job_1510484978137_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: hdfs://master:9000/hive/warehouse/account_rose/.hive-staging_hive_2017-1
_346_6125244884975429525-1/-ext-10000
Loading data to table default.account_rose
Table default.account_rose stats: [numFiles=1, numRows=2, totalSize=48, rawDataSize=46]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.85 sec   HDFS Read: 4062 HDFS Write: 124 SUCCE
Total MapReduce CPU Time Spent: 1 seconds 850 msec
OK
Time taken: 35.347 seconds
hive>
```

查看

```
hive> select * from account_rose;
OK
10002   1014    d5dsd32sdkif
40001   1023    s3dfdlksdfj3
Time taken: 0.117 seconds, Fetched: 2 row(s)
hive>
```

（4） 通过查询结果建表

说明：搜索所有在编号为 1014 部门（did=1014）的用户信息，创建为 dep1014 表：

```
hive> create table dep1014
    > as
    > select uid,did,uname
    > from user_info
    > where did='1014';
```

结果：

```
hive> select * from dep1014;
OK
10002   1014    rose
10005   1014    harry
20132   1014    jerry
Time taken: 0.069 seconds, Fetched: 3 row(s)
hive>
```

Tip:

1. 本实验的 hive 配置需要两个节点，如果自己机器撑不住的，可以使用单节点的本地元存储策略，网上有 hive-site.xml 文件的相关配置，自己搜下；
2. 出了异常检查下 hive 的配置，在运行 hive 前先确保 hadoop 正常运行，可以尝试在各个节点跑一下 wordcount；
3. 有问题举手问 TA