**项目实例笔记2014**

**辽宁沈阳**

**吴洪泽**

**二○一四年**

**修订历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** | **审核人** |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1总体介绍 5](#__RefHeading__711_637971582)

[2安装部署 5](#__RefHeading__107_157057738)

[约定 5](#__RefHeading__2145_1892713713)

[ssh相互登录 5](#__RefHeading__2147_1892713713)

[安装R和rsutdio 5](#__RefHeading__2149_1892713713)

[安装R时的错误及解决 5](#__RefHeading__2151_1892713713)

[mount系统安装包作为repos 6](#__RefHeading__2153_1892713713)

[公共javacp配置 6](#__RefHeading__2155_1892713713)

[配置hive 6](#__RefHeading__2157_1892713713)

[采用mysql存储metadata 6](#__RefHeading__2159_1892713713)

[Scp到另外一台机器 7](#__RefHeading__2161_1892713713)

[相关脚本 7](#__RefHeading__2163_1892713713)

[脚本学习过程 8](#__RefHeading__2165_1892713713)

[问题 8](#__RefHeading__2167_1892713713)

[Hadoop:slave datanode连接master出现问题 8](#__RefHeading__2169_1892713713)

[Hadoop:运行wordcount出现Bad connect ack with firstBadLink as 8](#__RefHeading__2171_1892713713)

[Hive:使用mysql时提示Unable to instantiate org.apache.hadoop.hive.metastore.HiveMetaStoreClient 9](#__RefHeading__2173_1892713713)

[3导入数据:sqoop1.99/sqoop2 9](#__RefHeading__2175_1892713713)

[实例：导入 9](#__RefHeading__109_157057738)

[创建数据库连接 9](#__RefHeading__2177_1892713713)

[创建任务 9](#__RefHeading__2179_1892713713)

[执行任务 9](#__RefHeading__2181_1892713713)

[查看运行结果 10](#__RefHeading__2183_1892713713)

[遇到的问题 10](#__RefHeading__2185_1892713713)

[Sqoop不能获得hadoop的任务反馈 10](#__RefHeading__111_157057738)

[对照表 10](#__RefHeading__713_637971582)

[词汇表 11](#__RefHeading__715_637971582)

[4导入数据： sqoop-1.4.4. hadoop-2.0.4 11](#__RefHeading__2187_1892713713)

[实例：直接导入表到hive 11](#__RefHeading__2189_1892713713)

[保护密码 11](#__RefHeading__2191_1892713713)

[定义options-file 11](#__RefHeading__2193_1892713713)

[应用options-file，列出所有表 12](#__RefHeading__2195_1892713713)

[应用options-file，直接导入到hive：-m 1 12](#__RefHeading__2197_1892713713)

[应用options-file，直接导入到hive：--split-by 12](#__RefHeading__2199_1892713713)

[检验导入结果，hive 13](#__RefHeading__2201_1892713713)

[实际应用：导入相关表到hive 13](#__RefHeading__2203_1892713713)

[其他,重新导入时数据清理： 13](#__RefHeading__2205_1892713713)

[特别注意事项 14](#__RefHeading__2207_1892713713)

[错误处理： 14](#__RefHeading__2209_1892713713)

[5预处理:hive 14](#__RefHeading__2211_1892713713)

[安装部署hive 14](#__RefHeading__2213_1892713713)

[配置文件 14](#__RefHeading__2215_1892713713)

[Hello hive 15](#__RefHeading__2217_1892713713)

[电力2014-辽宁沈阳 15](#__RefHeading__2219_1892713713)

# 总体介绍

# 安装部署

## 约定

|  |
| --- |
| /etc/hosts |
| 10.160.2.105 master-hadoop  10.160.2.106 slave-hadoop-01 |

## Ssh相互登录

**ssh无密登陆**需要特别特别注意: 在ssh无密码登陆时,**用户主目录的权限必须是700**, **~/.ssh/authorized\_keys的权限必须是600**.

(我遇到的问题,

1. 既不是/etc/selinux/config配置的问题

<http://blog.csdn.net/w13770269691/article/details/16342353>

centos6.4对网络管理相当严格，需要在原来的基础上关闭selinux, 到/etc/selinux/config下，把SELINUX=enforcing修改为SELINUX=disabled，重启机器，ok

2. 也不是仅仅是 authorized\_keys的权限问题;

<http://sjsky.iteye.com/blog/1123184>

查看详细信息: ssh -v master-hadoop

查看系统日志: sudo tail /var/log/secure -n 30

3. 而是用户住目录的权限问题!

<http://serverfault.com/questions/55343/cant-get-ssh-public-key-authentication-to-work>)

|  |
| --- |
| http://serverfault.com/questions/55343/cant-get-ssh-public-key-authentication-to-work |
| Check that your Centos machine has:  RSAAuthentication yes  PubkeyAuthentication yes  in sshd\_config  and ensure that you have proper permission on the centos machine's ~/.ssh/ directory.  chmod 700 ~/.ssh/  chmod 600 ~/.ssh/\*  should do the trick. |

|  |
| --- |
| 本地配置。先用ssh登录主机，然后在ssh-keygen。 |
| [hadoop@dsjsdxm-test1 ~]$ ssh-keygen -t rsa  Generating public/private rsa key pair.  Enter file in which to save the key (/home/hadoop/.ssh/id\_rsa):  Enter passphrase (empty for no passphrase):  Enter same passphrase again:  Your identification has been saved in /home/hadoop/.ssh/id\_rsa.  Your public key has been saved in /home/hadoop/.ssh/id\_rsa.pub.  The key fingerprint is:  e6:21:45:83:26:dc:67:03:af:e6:90:f2:c4:a0:b4:c0 hadoop@dsjsdxm-test1  The key's randomart image is:  +--[ RSA 2048]----+  | . ..oo |  |. o +o+. |  |.E. o oo. |  |o..o . o |  |... = + S |  | + + + . |  | . . . |  | |  | |  +-----------------+  [hadoop@dsjsdxm-test1 ~]$ cd ~/.ssh  [hadoop@dsjsdxm-test1 .ssh]$ ls  id\_rsa id\_rsa.pub  [hadoop@dsjsdxm-test1 .ssh]$ cp id\_rsa.pub authorized\_keys  [hadoop@dsjsdxm-test1 .ssh]$ ls -al authorized\_keys  -rw-r--r--. 1 hadoop hadoop 402 Jul 4 09:55 authorized\_keys  [hadoop@dsjsdxm-test1 .ssh]$ chmod 600 authorized\_keys  [hadoop@dsjsdxm-test1 .ssh]$ ls -al authorized\_keys  -rw-------. 1 hadoop hadoop 402 Jul 4 09:55 authorized\_keys  [hadoop@dsjsdxm-test1 .ssh]$ |

|  |
| --- |
| 相互登录 |
| 拷贝对方机器的authorized\_keys内容，追加到本地authorized\_keys。每个机器一行记录。 |

## 安装R和rsutdio

请参见” history.安装R和rstudio.20140704.txt”

### 安装R时的错误及解决

|  |
| --- |
| **configure: error: --with-readline=yes (default) and headers/libs are not available**  [hadoop@slave-hadoop-01 R-rc-20140702]$ ./configure --enable-R-shlib  checking for history\_truncate\_file... no  configure: error: --with-readline=yes (default) and headers/libs are not available |
| 安装readline-devel  sudo yum install readline-devel |

### mount系统安装包作为repos

|  |
| --- |
| mount系统安装包作为repos |
| sudo mount -t iso9660 -o loop /home/hadoop/pkg-bak/CentOS-6.5-x86\_64-bin-DVD1.iso /media/CentOS  其他相关命令  320 cd /etc/  321 ls  322 cd yum.repos.d/  323 ls  324 pwd  325 ls  326 pwd  327 mkdir bak.etc\_yum.repos.d  328 mkdir ~/bak.etc\_yum.repos.d  329 sudo mv \* ~/bak.etc\_yum.repos.d/  330 cp ~/bak.etc\_yum.repos.d/CentOS-Media.repo  331 cp ~/bak.etc\_yum.repos.d/CentOS-Media.repo .  332 sudo cp ~/bak.etc\_yum.repos.d/CentOS-Media.repo .  333 ls  334 pwd  335 vi CentOS-Media.repo  [hadoop@master-hadoop pkg-bak]$ cat /etc/yum.repos.d/CentOS-Media.repo  # CentOS-Media.repo  #  # This repo can be used with mounted DVD media, verify the mount point for  # CentOS-6. You can use this repo and yum to install items directly off the  # DVD ISO that we release.  #  # To use this repo, put in your DVD and use it with the other repos too:  # yum --enablerepo=c6-media [command]  #  # or for ONLY the media repo, do this:  #  # yum --disablerepo=\\* --enablerepo=c6-media [command]    [c6-media]  name=CentOS-$releasever - Media  baseurl=file:///media/CentOS/  file:///media/cdrom/  file:///media/cdrecorder/  gpgcheck=1  enabled=1  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6 |

**特别注意: 启动rstudio时，需要设置 export LANG=C。原因，不知道**。

~~其他时候，又需要改回来。 export LANG=zh\_us.UTF-8~~

## 公共javacp配置

|  |
| --- |
| Linux“/opt/java\_cp”目录下存放各种驱动 |
| CLASSPATH=/opt/java\_cp/mysql-connector-java-5.1.27.jar:/opt/java\_cp/ojdbc6.jar:/opt/java\_cp/ojdbc6\_g.jar |

## 配置hive

### 采用mysql存储metadata

|  |
| --- |
| 在mysql中创建hive用户 |
| create user 'hive'@'%' indentified by 'hive';  grant all privileges on \*.\* to 'hive'@'%' with grant option;  grant all privileges on \*.\* to 'root'@'%' with grant option;  flush privileges;  修改/etc/mysql/my.cnf，调整其绑定的ip  注释掉 # bind-address = 127.0.0.1 |
| **创建hive数据库** |
| #使用hive用户登录  mysql -u hive -phive -h master-hadoop  ~~# 创建hive数据库---也可以由hive自动创建~~  ~~create database hive;~~  ~~show databases;~~ |

|  |
| --- |
| 在mysql中创建sqoop用户 |
| create user 'qoop'@'%' indentified by 'sqoop';  grant all privileges on \*.\* to 'sqoop'@'%' with grant option;  grant all privileges on \*.\* to 'root'@'%' with grant option;  flush privileges; |
| 创建sqoop数据库 |
| create database db\_sqoop;  show databases; |

## Scp到另外一台机器

|  |
| --- |
| Linux“/opt/java\_cp”目录下存放各种驱动 |
| CLASSPATH=/opt/java\_cp/mysql-connector-java-5.1.27.jar:/opt/java\_cp/ojdbc6.jar:/opt/java\_cp/ojdbc6\_g.jar |

|  |
| --- |
| scp环境变量并使之有效 |
| scp ~/.bashrc hadoop@slave-hadoop-01:~/.bashrc  ssh hadoop@slave-hadoop-01 "source ~/.bashrc"  # 做不到远程使用sudo执行命令。还有别的法子吗？  #ssh hadoop@slave-hadoop-01 "sudo usermod -a -G root hadoop"  # 拷贝目录到远程 （需要事先在目标机器上保证hadoop加入root组，且/opt可读写）  # 添加hadoop用户到root用户组 sudo usermod -a -G root hadoop  # root用户组可以具有/opt的写权限 sudo chmod g+w /opt  # 按目录拷贝是很慢的，建议使用压缩后再解压的方式  # ----------------按目录拷贝  # scp -r /opt/hadoop hadoop@slave-hadoop-01:/opt  # ----------------压缩后再解压的方式 |

|  |
| --- |
| **自动拷贝所有** |
| cd ~/dev-sw/  ./exec-myshell-scp.sh |
| **针对某一个单独执行** |
| export VAR\_SCP=zookeeper  /home/hadoop/dev-sw/myshell-scp.sh  export VAR\_SCP=不存在的VAR\_SCP变量 |

#### 相关脚本

|  |
| --- |
| myshell-scp.sh  存放在/home/hadoop/dev-sw/myshell-scp.sh |
| #! /bin/bash  # export VAR\_SCP=hadoop  cd /opt; tar -zcf ${VAR\_SCP}.tar.gz ${VAR\_SCP}  scp /opt/${VAR\_SCP}.tar.gz hadoop@slave-hadoop-01:/opt  ssh hadoop@slave-hadoop-01 "cd /opt; tar -zxf ${VAR\_SCP}.tar.gz"  ssh hadoop@slave-hadoop-01 "rm /opt/${VAR\_SCP}.tar.gz"  cd /opt; rm ${VAR\_SCP}.tar.gz  export VAR\_SCP=不存在的VAR\_SCP变量 |

|  |
| --- |
| exec-myshell-scp.sh  存放在/home/hadoop/dev-sw/myshell-scp.sh |
| #! /bin/bash  export VAR\_SCP=hadoop  /home/hadoop/dev-sw/myshell-scp.sh  export VAR\_SCP=jdk  /home/hadoop/dev-sw/myshell-scp.sh  export VAR\_SCP=mahout  /home/hadoop/dev-sw/myshell-scp.sh  export VAR\_SCP=java\_cp  /home/hadoop/dev-sw/myshell-scp.sh  export VAR\_SCP=scala  /home/hadoop/dev-sw/myshell-scp.sh  export VAR\_SCP=spark  /home/hadoop/dev-sw/myshell-scp.sh  export VAR\_SCP=zookeeper  /home/hadoop/dev-sw/myshell-scp.sh  export VAR\_SCP=不存在的VAR\_SCP变量 |

#### 脚本学习过程

|  |
| --- |
| Scp命令与脚本 |
| scp ~/.bashrc hadoop@slave-hadoop-01:~/.bashrc  ssh hadoop@slave-hadoop-01 "source ~/.bashrc"  # 做不到远程使用sudo执行命令。还有别的法子吗？  #ssh hadoop@slave-hadoop-01 "sudo usermod -a -G root hadoop"  # 拷贝目录到远程 （需要事先在目标机器上保证hadoop加入root组，且/opt可读写）  # 添加hadoop用户到root用户组 sudo usermod -a -G root hadoop  # root用户组可以具有/opt的写权限 sudo chmod g+w /opt  # 按目录拷贝是很慢的，建议使用压缩后再解压的方式  # ----------------按目录拷贝  # scp -r /opt/hadoop hadoop@slave-hadoop-01:/opt  # ----------------压缩后再解压的方式  # 压缩  cd /opt; tar -zcf hadoop.tar.gz hadoop;  # 传输  scp /opt/hadoop.tar.gz hadoop@slave-hadoop-01:/opt  ## 解压缩  #ssh hadoop@slave-hadoop-01 "cd /opt; tar -zxf hadoop.tar.gz"  #ssh hadoop@slave-hadoop-01 "cd /opt; tar -zxf hadoop.tar.gz; rm hadoop.tar.gz;"  ## 删除远程包  #ssh hadoop@slave-hadoop-01 "rm /opt/hadoop.tar.gz"  # 删除本地包  #cd /opt; rm hadoop.tar.gz  # ----------------压缩后再解压的方式  # /home/hadoop/dev-sw/myschshell.sh  #! /bin/bash  # export VAR\_SCP=hadoop  cd /opt; tar -zcf ${VAR\_SCP}.tar.gz ${VAR\_SCP}  scp /opt/${VAR\_SCP}.tar.gz hadoop@slave-hadoop-01:/opt  ssh hadoop@slave-hadoop-01 "cd /opt; tar -zxf ${VAR\_SCP}.tar.gz"  ssh hadoop@slave-hadoop-01 "rm /opt/${VAR\_SCP}.tar.gz"  cd /opt; rm ${VAR\_SCP}.tar.gz  export VAR\_SCP=不存在的VAR\_SCP变量  export VAR\_SCP=hadoop  /home/hadoop/dev-sw/myschshell.sh  export VAR\_SCP=jdk  /home/hadoop/dev-sw/myschshell.sh  export VAR\_SCP=mahout  /home/hadoop/dev-sw/myschshell.sh  export VAR\_SCP=java\_cp  /home/hadoop/dev-sw/myschshell.sh  export VAR\_SCP=scala  /home/hadoop/dev-sw/myschshell.sh  export VAR\_SCP=spark  /home/hadoop/dev-sw/myschshell.sh  export VAR\_SCP=zookeeper  /home/hadoop/dev-sw/myschshell.sh |

## 问题

#### Hadoop:slave datanode连接master出现问题

|  |
| --- |
| **日志** |
| ady tried 0 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1 SECONDS)  2014-07-04 13:56:11,381 INFO org.apache.hadoop.ipc.Client: Retrying connect to server: master-hadoop/10.160.2.105:9000. Already tried 1 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1 SECONDS)  2014-07-04 13:56:12,382 INFO org.apache.hadoop.ipc.Client: Retrying connect to server: master-hadoop/10.160.2.105:9000. Already tried 2 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1 SECONDS)  2014-07-04 13:56:13,384 INFO org.apache.hadoop.ipc.Client: Retrying connect to server: master-hadoop/10.160.2.105:9000. Already tried 3 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1 SECONDS)  2014-07-04 13:56:14,385 INFO org.apache.hadoop.ipc.Client: Retrying connect to server: master-hadoop/10.160.2.105:9000. Already tried 4 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1 SECONDS)  2014-07-04 13:56:15,387 INFO org.apache.hadoop.ipc.Client: Retrying connect to server: master-hadoop/10.160.2.105:9000. Already tried 5 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1 SECONDS)  2014-07-04 13:56:16,388 INFO org.apache.hadoop.ipc.Client: Retrying connect to server: master-hadoop/10.160.2.105:9000. Already tried 6 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1 SECONDS) |
| **诊断** |
| 查看端口状态：  [hadoop@master-hadoop hadoop]$ lsof -i:9000  COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME  java 20894 hadoop 167u IPv4 354363 0t0 TCP master-hadoop:cslistener (LISTEN)  java 20894 hadoop 177u IPv4 355132 0t0 TCP master-hadoop:cslistener->master-hadoop:33306 (ESTABLISHED)  java 21017 hadoop 177u IPv4 355131 0t0 TCP master-hadoop:33306->master-hadoop:cslistener (ESTABLISHED)  使用nmap测试端口是否打开：（在master-hadoop机器上执行）  [hadoop@master-hadoop hadoop]$ nmap master-hadoop -p 9000  Starting Nmap 5.51 ( http://nmap.org ) at 2014-07-04 14:09 CST  Nmap scan report for master-hadoop (10.160.2.105)  Host is up (0.000052s latency).  PORT STATE SERVICE  9000/tcp open cslistener  Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds  使用nmap测试端口是否打开：（在slave-hadoop-01机器上执行）  [hadoop@slave-hadoop-01 ~]$ nmap master-hadoop -p 9000  Starting Nmap 5.51 ( http://nmap.org ) at 2014-07-04 14:24 CST  Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn  Nmap done: 1 IP address (0 hosts up) scanned in 0.04 seconds |
| 关闭防火墙后解决：  在master-hadoop:  sudo service iptables stop  在slave-hadoop-01:  nmap master-hadoop -p 9000  查看hadoop日志： |

#### Hadoop:运行wordcount出现Bad connect ack with firstBadLink as

|  |
| --- |
| **日志：**  hadoop jar hadoop-mapreduce-examples-2.2.0.jar wordcount words-data result/wordcount01 |
| [hadoop@master-hadoop mapreduce]$  [hadoop@master-hadoop mapreduce]$ hadoop jar hadoop-mapreduce-examples-2.2.0.jar wordcount words-data result/wordcount01  14/07/04 14:48:04 INFO client.RMProxy: Connecting to ResourceManager at master-hadoop/10.160.2.105:8032  14/07/04 14:48:05 INFO hdfs.DFSClient: Exception in createBlockOutputStream  java.io.IOException: Bad connect ack with firstBadLink as 10.160.2.106:50010  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.createBlockOutputStream(DFSOutputStream.java:1166)  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.nextBlockOutputStream(DFSOutputStream.java:1088)  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:514)  14/07/04 14:48:05 INFO hdfs.DFSClient: Abandoning BP-679147928-10.160.2.105-1404448880214:blk\_1073741847\_1023  14/07/04 14:48:05 INFO hdfs.DFSClient: Excluding datanode 10.160.2.106:50010  14/07/04 14:48:05 INFO input.FileInputFormat: Total input paths to process : 1  14/07/04 14:48:05 INFO hdfs.DFSClient: Exception in createBlockOutputStream  java.io.IOException: Bad connect ack with firstBadLink as 10.160.2.106:50010  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.createBlockOutputStream(DFSOutputStream.java:1166)  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.nextBlockOutputStream(DFSOutputStream.java:1088)  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:514)  14/07/04 14:48:05 INFO hdfs.DFSClient: Abandoning BP-679147928-10.160.2.105-1404448880214:blk\_1073741849\_1025  14/07/04 14:48:05 INFO hdfs.DFSClient: Excluding datanode 10.160.2.106:50010  14/07/04 14:48:05 INFO hdfs.DFSClient: Exception in createBlockOutputStream  java.io.IOException: Bad connect ack with firstBadLink as 10.160.2.106:50010  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.createBlockOutputStream(DFSOutputStream.java:1166)  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.nextBlockOutputStream(DFSOutputStream.java:1088)  at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:514)  14/07/04 14:48:05 INFO hdfs.DFSClient: Abandoning BP-679147928-10.160.2.105-1404448880214:blk\_1073741851\_1027  14/07/04 14:48:05 INFO hdfs.DFSClient: Excluding datanode 10.160.2.106:50010  14/07/04 14:48:05 INFO mapreduce.JobSubmitter: number of splits:1  14/07/04 14:48:05 INFO Configuration.deprecation: user.name is deprecated. Instead, use mapreduce. |
|  |
| 防火墙的问题： |
| 关闭slave上的防火墙：  [hadoop@slave-hadoop-01 logs]$ sudo service iptables stop  [sudo] password for hadoop:  iptables：将链设置为政策 ACCEPT：filter [确定]  iptables：清除防火墙规则： [确定]  iptables：正在卸载模块： [确定]  [hadoop@slave-hadoop-01 logs]$ |
|  |

#### Hive:使用mysql时提示Unable to instantiate org.apache.hadoop.hive.metastore.HiveMetaStoreClient

|  |
| --- |
| **日志：** |
| [hadoop@master-hadoop logs-tmp]$ hive  14/07/04 15:31:31 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces  14/07/04 15:31:31 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize  14/07/04 15:31:31 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce.reduce.speculative  14/07/04 15:31:31 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.node  14/07/04 15:31:31 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive  14/07/04 15:31:31 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack  14/07/04 15:31:31 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.maxsize  14/07/04 15:31:31 INFO Configuration.deprecation: mapred.committer.job.setup.cleanup.needed is deprecated. Instead, use mapreduce.job.committer.setup.cleanup.needed  14/07/04 15:31:31 WARN conf.HiveConf: DEPRECATED: hive.metastore.ds.retry.\* no longer has any effect. Use hive.hmshandler.retry.\* instead  Logging initialized using configuration in file:/opt/hive/apache-hive-0.13.1-bin/conf/hive-log4j.properties  Exception in thread "main" java.lang.RuntimeException: java.lang.RuntimeException: Unable to instantiate org.apache.hadoop.hive.metastore.HiveMetaStoreClient  at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:346)  at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:681)  at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:625)  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)  at java.lang.reflect.Method.invoke(Method.java:606)  at org.apache.hadoop.util.RunJar.main(RunJar.java:212)  Caused by: java.lang.RuntimeException: Unable to instantiate org.apache.hadoop.hive.metastore.HiveMetaStoreClient  at org.apache.hadoop.hive.metastore.MetaStoreUtils.newInstance(MetaStoreUtils.java:1412)  at org.apache.hadoop.hive.metastore.RetryingMetaStoreClient.<init>(RetryingMetaStoreClient.java:62)  at org.apache.hadoop.hive.metastore.RetryingMetaStoreClient.getProxy(RetryingMetaStoreClient.java:72)  at org.apache.hadoop.hive.ql.metadata.Hive.createMetaStoreClient(Hive.java:2453)  at org.apache.hadoop.hive.ql.metadata.Hive.getMSC(Hive.java:2465)  at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:340)  ... 7 more  Caused by: java.lang.reflect.InvocationTargetException  at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) |
| 进入debug模式运行hive：  hive -hiveconf hive.root.logger=DEBUG,console |
| 原因：需要将mysql的驱动拷贝到$HIVE\_HOME/lib |
| Caused by: org.datanucleus.exceptions.NucleusException: Attempt to invoke the "BoneCP" plugin to create a ConnectionPool gave an error : The specified datastore driver ("com.mysql.jdbc.Driver") was not found in the CLASSPATH. Please check your CLASSPATH specification, and the name of the driver.  at org.datanucleus.store.rdbms.ConnectionFactoryImpl.generateDataSources(ConnectionFactoryImpl.java:259)  at org.datanucleus.store.rdbms.ConnectionFactoryImpl.initialiseDataSources(ConnectionFactoryImpl.java:131)  at org.datanucleus.store.rdbms.ConnectionFactoryImpl.<init>(ConnectionFactoryImpl.java:85)  ... 64 more  Caused by: org.datanucleus.store.rdbms.connectionpool.DatastoreDriverNotFoundException: The specified datastore driver ("com.mysql.jdbc.Driver") was not found in the CLASSPATH. Please check your CLASSPATH specification, and the name of the driver.  at org.datanucleus.store.rdbms.connectionpool.AbstractConnectionPoolFactory.loadDriver(AbstractConnectionPoolFactory.java:58)  at org.datanucleus.store.rdbms.connectionpool.BoneCPConnectionPoolFactory.createConnectionPool(BoneCPConnectionPoolFactory.java:54)  at org.datanucleus.store.rdbms.ConnectionFactoryImpl.generateData |

# 导入数据:sqoop1.99/sqoop2

## 实例：导入

### 创建数据库连接

|  |
| --- |
| sqoop:000> show connector  +----+------------------------+---------+------------------------------------------------------+  | Id | Name | Version | Class |  +----+------------------------+---------+------------------------------------------------------+  | 1 | generic-jdbc-connector | 1.99.3 | org.apache.sqoop.connector.jdbc.GenericJdbcConnector |  +----+------------------------+---------+------------------------------------------------------+  sqoop:000> create connection --cid 1  Creating connection for connector with id 1  Please fill following values to create new connection object  Name: oracle  Connection configuration  JDBC Driver Class: oracle.jdbc.OracleDriver  JDBC Connection String: jdbc:oracle:thin:@10.160.8.22:1521/epmpln  Username: JTYDFX\_SELECT  Password: \*\*\*\*\*\*\*\*\*\*  JDBC Connection Properties:  There are currently 0 values in the map:  entry#  Security related configuration options  Max connections: 10  New connection was successfully created with validation status FINE and persistent id 1  sqoop:000>  sqoop:000> create job  Required argument --xid is missing.  sqoop:000> show connection  +----+--------+-----------+---------+  | Id | Name | Connector | Enabled |  +----+--------+-----------+---------+  | 1 | oracle | 1 | true |  +----+--------+-----------+---------+ |

### 创建任务

|  |
| --- |
| sqoop:000> show connection  +----+--------+-----------+---------+  | Id | Name | Connector | Enabled |  +----+--------+-----------+---------+  | 1 | oracle | 1 | true |  +----+--------+-----------+---------+  sqoop:000> create job --xid 1 --type import  Creating job for connection with id 1  Please fill following values to create new job object  Name: bigdata\_user\_info\_s01  Database configuration  Schema name:  Table name: bigdata\_user\_info\_s01  Table SQL statement:  Table column names:  Partition column name: 用电类别  Nulls in partition column:  Boundary query:  Output configuration  Storage type:  0 : HDFS  Choose: 0  Output format:  0 : TEXT\_FILE  1 : SEQUENCE\_FILE  Choose: 0  Compression format:  0 : NONE  1 : DEFAULT  2 : DEFLATE  3 : GZIP  4 : BZIP2  5 : LZO  6 : LZ4  7 : SNAPPY  Choose: 0  Output directory: ds-oracle/epmpln/bigdata\_user\_info\_s01  Throttling resources  Extractors:  Loaders:  New job was successfully created with validation status FINE and persistent id 1  sqoop:000>  sqoop:000> show job  +----+-----------------------+--------+-----------+---------+  | Id | Name | Type | Connector | Enabled |  +----+-----------------------+--------+-----------+---------+  | 1 | bigdata\_user\_info\_s01 | IMPORT | 1 | true |  +----+-----------------------+--------+-----------+---------+ |

### 执行任务

|  |
| --- |
| sqoop:000> start job -j 1  Exception has occurred during processing command  Exception: org.apache.sqoop.common.SqoopException Message: CLIENT\_0001:Server has returned exception  sqoop:000> set option --name verbose --value true  Verbose option was changed to true  sqoop:000> create job --xid 1 --type import  Creating job for connection with id 1  Please fill following values to create new job object  Name: bigdata\_user\_info\_s01  Database configuration  Schema name:  Table name: bigdata\_user\_info\_s01  Table SQL statement:  Table column names:  Partition column name: 用电类别  Nulls in partition column:  Boundary query:  Output configuration  Storage type:  0 : HDFS  Choose: 0  Output format:  0 : TEXT\_FILE  1 : SEQUENCE\_FILE  Choose: 0  Compression format:  0 : NONE  1 : DEFAULT  2 : DEFLATE  3 : GZIP  4 : BZIP2  5 : LZO  6 : LZ4  7 : SNAPPY  Choose: 0  Output directory: ds-oracle/epmpln/bigdata\_user\_info\_s01  Throttling resources  Extractors:  Loaders:  New job was successfully created with validation status FINE and persistent id 2  sqoop:000>  sqoop:000> show job  +----+-----------------------+--------+-----------+---------+  | Id | Name | Type | Connector | Enabled |  +----+-----------------------+--------+-----------+---------+  | 1 | bigdata\_user\_info\_s01 | IMPORT | 1 | true |  | 2 | bigdata\_user\_info\_s01 | IMPORT | 1 | true |  +----+-----------------------+--------+-----------+---------+  sqoop:000> start job -j 2  Submission details  Job ID: 2  Server URL: http://localhost:12000/sqoop/  Created by: hadoop  Creation date: 2014-06-23 11:31:21 CST  Lastly updated by: hadoop  External ID: job\_1403228186343\_0005  http://dsjtest1.dsjsdxm.com:8088/proxy/application\_1403228186343\_0005/  2014-06-23 11:31:21 CST: BOOTING - Progress is not available |

### 查看运行结果

|  |
| --- |
| sqoop:000> status job --jid 2  Submission details  Job ID: 2  Server URL: http://localhost:12000/sqoop/  Created by: hadoop  Creation date: 2014-06-23 11:31:21 CST  Lastly updated by: hadoop  External ID: job\_1403228186343\_0005  http://dsjtest1.dsjsdxm.com:19888/jobhistory/job/job\_1403228186343\_0005  2014-06-23 12:06:37 CST: SUCCEEDED  Counters:  org.apache.hadoop.mapreduce.JobCounter  SLOTS\_MILLIS\_MAPS: 7371  TOTAL\_LAUNCHED\_MAPS: 1  OTHER\_LOCAL\_MAPS: 1  org.apache.hadoop.mapreduce.lib.input.FileInputFormatCounter  BYTES\_READ: 0  org.apache.hadoop.mapreduce.lib.output.FileOutputFormatCounter  BYTES\_WRITTEN: 2879091  org.apache.hadoop.mapreduce.TaskCounter  MAP\_INPUT\_RECORDS: 0  MERGED\_MAP\_OUTPUTS: 0  PHYSICAL\_MEMORY\_BYTES: 167645184  SPILLED\_RECORDS: 0  FAILED\_SHUFFLE: 0  CPU\_MILLISECONDS: 4260  COMMITTED\_HEAP\_BYTES: 137887744  VIRTUAL\_MEMORY\_BYTES: 733609984  MAP\_OUTPUT\_RECORDS: 20288  SPLIT\_RAW\_BYTES: 183  GC\_TIME\_MILLIS: 82  org.apache.hadoop.mapreduce.FileSystemCounter  FILE\_READ\_OPS: 0  FILE\_WRITE\_OPS: 0  FILE\_BYTES\_READ: 0  FILE\_LARGE\_READ\_OPS: 0  HDFS\_BYTES\_READ: 183  FILE\_BYTES\_WRITTEN: 89761  HDFS\_LARGE\_READ\_OPS: 0  HDFS\_BYTES\_WRITTEN: 3550790  HDFS\_READ\_OPS: 4  HDFS\_WRITE\_OPS: 2  org.apache.sqoop.submission.counter.SqoopCounters  ROWS\_READ: 20288  Job executed successfully  [hadoop@dsjtest1 ~]$ hdfs dfs -ls ds-oracle  14/06/23 12:08:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  Found 1 items  drwxr-xr-x - hadoop supergroup 0 2014-06-23 11:31 ds-oracle/epmpln  [hadoop@dsjtest1 ~]$ hdfs dfs -ls ds-oracle/epmpln  14/06/23 12:08:57 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  Found 1 items  drwxr-xr-x - hadoop supergroup 0 2014-06-23 11:31 ds-oracle/epmpln/bigdata\_user\_info\_s01  [hadoop@dsjtest1 ~]$ hdfs dfs -ls ds-oracle/epmpln/bigdata\_user\_info\_s01  14/06/23 12:09:10 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  Found 2 items  -rw-r--r-- 2 hadoop supergroup 0 2014-06-23 11:31 ds-oracle/epmpln/bigdata\_user\_info\_s01/\_SUCCESS  -rw-r--r-- 2 hadoop supergroup 3550790 2014-06-23 11:31 ds-oracle/epmpln/bigdata\_user\_info\_s01/part-m-00000  [hadoop@dsjtest1 ~]$ |

## 遇到的问题

### Sqoop不能获得hadoop的任务反馈

|  |  |
| --- | --- |
| **现象** | 2014-06-19 09:18:15,357 ERROR server.SqoopProtocolServlet [org.apache.sqoop.server.SqoopProtocolServlet.doGet(SqoopProtocolServlet.java:53)] Exception in GET http://master-hadoop:12000/sqoop//v1/submission/notification/4  org.apache.sqoop.common.SqoopException: MAPREDUCE\_0003:Can't get RunningJob instance  at org.apache.sqoop.submission.mapreduce.MapreduceSubmissionEngine.status(MapreduceSubmissionEngine.java:295)  at org.apache.sqoop.framework.JobManager.update(JobManager.java:529)  at org.apache.sqoop.framework.JobManager.status(JobManager.java:519)  at org.apache.sqoop.handler.SubmissionRequestHandler.handleNotification(SubmissionRequestHandler.java:92)  at org.apache.sqoop.handler.SubmissionRequestHandler.handleEvent(SubmissionRequestHandler.java:79)  at org.apache.sqoop.server.v1.SubmissionServlet.handleGetRequest(SubmissionServlet.java:39)  at org.apache.sqoop.server.SqoopProtocolServlet.doGet(SqoopProtocolServlet.java:48)  at javax.servlet.http.HttpServlet.service(HttpServlet.java:617)  at javax.servlet.http.HttpServlet.service(HttpServlet.java:723)  ….....  Caused by: java.io.IOException: java.net.ConnectException: Call From master-hadoop/127.0.0.1 to 0.0.0.0:10020 failed on connection exception: java.net.ConnectException: Connection refused; For more details see: http://wiki.apache.org/hadoop/ConnectionRefused  at org.apache.hadoop.mapred.ClientServiceDelegate.invoke(ClientServiceDelegate.java:331)  at org.apache.hadoop.mapred.ClientServiceDelegate.getJobStatus(ClientServiceDelegate.java:416)  at org.apache.hadoop.mapred.YARNRunner.getJobStatus(YARNRunner.java:522)  at org.apache.hadoop.mapreduce.Cluster.getJob(Cluster.java:183)  at org.apache.hadoop.mapred.JobClient$2.run(JobClient.java:580)  at org.apache.hadoop.mapred.JobClient$2.run(JobClient.java:578)  at java.security.AccessController.doPrivileged(Native Method)  at javax.security.auth.Subject.doAs(Subject.java:415) |
| **诊断** | 10020端口不能访问, 10020端口是jobhistory端口.  查看hadoop服务器端口是否打开:netstat -an |grep 10020  端口的确尚未打开 |
| **解决** | 启动jobhistory服务:  $HADOOP\_PREFIX/sbin/mr-jobhistory-daemon.sh start historyserver --config $HADOOP\_CONF\_DIR |
| **小结/疑问** | 为什么jobhistory需要再次启动? |

## 对照表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Hadoop** | **spark** | **传统数据仓库** | **未知的** |
| 数据抽取 | sqoop |  |  |  |
| 数据存储 | HDFS,HBase |  | RDBMS(Oracle, MySQL, SQL Server, Teradata, and Netezza  ) |  |
| 数据处理 | Hive,Pig, Impala | shark |  |  |
| 分析引擎 | 执行引擎:MapReduce |  |  |  |
|  | SQL-oriented analytics  :  Cloudera  Impala and Dremel-style analytic engines |  |  |  |
|  | 统计分析: R |  |  |  |
| 调度框架 | YARN |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 词汇表

|  |  |
| --- | --- |
| Sqoop | Apache Sqoop, short for “SQL to Hadoop,”  was  created to perform bidirectional data transfer between Hadoop and almost any external  structured datastore. |
| 集成商 | Open Source:Apache Bigtop  Commercial: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 导入数据： sqoop-1.4.4. hadoop-2.0.4

## 实例：直接导入表到hive

### 保护密码

|  |
| --- |
| echo 你的密码 > sqoop.password |
| #将sqoop.password上传到HDFS，并将权限修改为400  hadoop dfs -put sqoop.password /user/$USER/sqoop.password  hdfs dfs -chown 400 /user/$USER/sqoop.password |
| 在脚本中运用  --password-file /user/$USER/sqoop.password |

### 定义options-file

|  |
| --- |
| --options-file： 列出所有表list-tables-oracle.sqoop |
| list-tables  --connect  jdbc:oracle:thin:@10.160.8.22:1521/epmpln  --username  JTYDFX\_SELECT  #--password  --password-file  /user/hadoop/sqoop.password  --verbose  # sqoop --options-file list-tables-oracle.sqoop |

|  |
| --- |
| --options-file： 导入到HDFS: import-oracle.sqoop |
| import  --connect  jdbc:oracle:thin:@10.160.8.22:1521/epmpln  --username  JTYDFX\_SELECT  #--password  --password-file  /user/hadoop/sqoop.password  --verbose  # sqoop --options-file import-oracle.sqoop --table XXXX |

|  |
| --- |
| --options-file： 导入到hive: hive-import-oracle.sqoop |
| import  --connect  jdbc:oracle:thin:@10.160.8.22:1521/epmpln  --username  JTYDFX\_SELECT  #--password  --password-file  /user/hadoop/sqoop.password  --hive-import  # 临时目录  --warehouse-dir  /user/hadoop/tmp-for-sqoop-hive  --verbose  # sqoop --options-file import-oracle.sqoop --table XXXX |

### 应用options-file，列出所有表

|  |
| --- |
| Linux命令行 |
| sqoop --options-file list-tables-oracle.sqoop |

### 应用options-file，直接导入到hive：-m 1

|  |
| --- |
| Linux命令行 |
| sqoop --options-file hive-import-oracle.sqoop --table BIGDATA\_VOLUME\_OF\_PRC\_S01 -m 1 |

### 应用options-file，直接导入到hive：--split-by

|  |
| --- |
| Linux命令行 |
| sqoop --options-file hive-import-oracle.sqoop --table BIGDATA\_ARC\_VOLUME\_PERM\_S01 --split-by 年月 |

### 检验导入结果，hive

|  |
| --- |
| Linux命令行 |
| hive |
| Hive命令行 |
| show tables;  select count(\*) from bigdata\_arc\_volume\_perm\_s01; |

## 实际应用：导入相关表到hive

|  |
| --- |
| Linux命令文件: hive-import-all.sh |
| #! /usr/bin/env bash  #查看一下表列表  sqoop --options-file list-tables-oracle.sqoop  #导入用户信息  sqoop --options-file hive-import-oracle.sqoop --table BIGDATA\_USER\_INFO\_S01 --split-by 行业  #月用电量  sqoop --options-file hive-import-oracle.sqoop --table BIGDATA\_ARC\_VOLUME\_PERM\_S01 --split-by 年月  #用户逐月缴费欠费信息  sqoop --options-file hive-import-oracle.sqoop --table BIGDATA\_A\_RCVBL\_FLOW\_PM\_S01 --split-by 年月  #违约用电次数  sqoop --options-file hive-import-oracle.sqoop --table BIGDATA\_POWER\_STEAL\_PERY\_S01 --split-by 年份  #用户逐年缴费方式变更次数  sqoop --options-file hive-import-oracle.sqoop --table bigdata\_change\_payType\_pY\_s01 --split-by 年份  #是否阶梯电价  sqoop --options-file hive-import-oracle.sqoop --table bigdata\_ts\_or\_prcScope\_s01 --split-by 是否分时,是否阶梯  #分时电价电量  sqoop --options-file hive-import-oracle.sqoop --table bigdata\_volume\_of\_ts\_s01 --split-by YM  #阶梯电价电量  sqoop --options-file hive-import-oracle.sqoop --table bigdata\_volume\_of\_prc\_s01 --split-by YM |
| Hive命令行 |
| chmod u+x hive-import-all.sh  ./hive-import-all.sh |

### 其他,重新导入时数据清理：

|  |
| --- |
| Linux命令行: HDFS临时目录清理 |
| cp hive-default.xml.template hive**-site**.xml |
| #查看hive warehouse目录下表  hdfs dfs -ls /user/hive/warehouse  #查看中间目录  hdfs dfs -ls /user/hadoop/tmp-for-sqoop-hive  **# 清空中间目录**  **hdfs dfs -rm -R /user/hadoop/tmp-for-sqoop-hive** |

|  |
| --- |
| Hive命令行: HIVE表清理 |
| cp hive-default.xml.template hive**-site**.xml |
| show tables;  drop table XXXXX; |

### 特别注意事项

1. **Oracle中的用户名和表名要大写**。否则会提示“Imported Failed: Attempted to generate class with no columns!”
2. 注意，若设置了统一的metastore\_db路径，则**在sqoop导入数据时不能登录hive**。因为默认的配置下，一个metastore\_db只允许一个hive登录。

|  |
| --- |
| 14/06/24 17:01:03 INFO hive.HiveImport: Caused by: java.sql.SQLException: Unable to open a test connection to the given database. JDBC url = jdbc:derby:;databaseName=/home/hadoop/data-hive-0.13/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: ------  14/06/24 17:01:03 INFO hive.HiveImport: java.sql.SQLException: Failed to start database '/home/hadoop/data-hive-0.13/metastore\_db' with class loader sun.misc.Launcher$AppClassLoader@4821e115, see the next exception for details. |

### 错误处理：

|  |
| --- |
| 导入bigdata\_rcvbl\_flow\_pM\_s98表时提示：  14/07/08 15:33:50 ERROR tool.ImportTool: Imported Failed: Attempted to generate class with no columns! |
| Oracle数据库的表名必须大写！ |
|  |

|  |
| --- |
| 导入BIGDATA\_RCVBL\_FLOW\_PM\_S98表时提示：  14/07/08 15:39:02 INFO hive.HiveImport: FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. MetaException(message:javax.jdo.JDODataStoreException: Add request failed : INSERT INTO `COLUMNS\_V2` (`CD\_ID`,`COMMENT`,`COLUMN\_NAME`,`TYPE\_NAME`,`INTEGER\_IDX`) VALUES (?,?,?,?,?)  14/07/08 15:39:02 INFO hive.HiveImport: at org.datanucleus.api.jdo.NucleusJDOHelper.getJDOExceptionForNucleusException(NucleusJDOHelper.java:451)  14/07/08 15:39:02 INFO hive.HiveImport: at org.datanucleus.api.jdo.JDOPersistenceManager.jdoMakePersistent(JDOPersistenceManager.java:732)  14/07/08 15:39:02 INFO hive.HiveImport: at org.datanucleus.api.jdo.JDOPersistenceManager.makePersistent(JDOPersistenceManager.java:752)  14/07/08 15:39:02 INFO hive.HiveImport: at org.apache.hadoop.hive.metastore.ObjectStore.createTable(ObjectStore.java:719)  14/07/08 15:39:02 INFO hive.HiveImport: NestedThrowablesStackTrace:  14/07/08 15:39:02 INFO hive.HiveImport: java.sql.BatchUpdateException: Duplicate entry '112-??' for key 'PRIMARY'  14/07/08 15:39:02 INFO hive.HiveImport: at com.mysql.jdbc.PreparedStatement.executeBatchSerially(PreparedStatement.java:2055)  14/07/08 15:39:02 INFO hive.HiveImport: at com.mysql.jdbc.PreparedStatement.executeBatch(PreparedStatement.java:1467)  14/07/08 15:39:02 INFO hive.HiveImport: at java.lang.reflect.Method.invoke(Method.java:606)  14/07/08 15:39:02 INFO hive.HiveImport: at org.apache.hadoop.util.RunJar.main(RunJar.java:212)  14/07/08 15:39:02 INFO hive.HiveImport: Caused by: com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintViolationException: Duplicate entry '112-??' for key 'PRIMARY'  14/07/08 15:39:02 INFO hive.HiveImport: at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)  14/07/08 15:39:02 INFO hive.HiveImport: at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:57)  14/07/08 15:39:02 INFO hive.HiveImport: ... 94 more  14/07/08 15:39:02 INFO hive.HiveImport: )  14/07/08 15:39:02 ERROR tool.ImportTool: Encountered IOException running import job: java.io.IOException: Hive exited with status 1  at org.apache.sqoop.hive.HiveImport.executeExternalHiveScript(HiveImport.java:385)  at org.apache.sqoop.hive.HiveImport.executeScript(HiveImport.java:335)  at org.apache.sqoop.hive.HiveImport.importTable(HiveImport.java:239)  at org.apache.sqoop.tool.ImportTool.importTable(ImportTool.java:425)  at org.apache.sqoop.tool.ImportTool.run(ImportTool.java:502)  at org.apache.sqoop.Sqoop.run(Sqoop.java:145)  at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)  at org.apache.sqoop.Sqoop.runSqoop(Sqoop.java:181)  at org.apache.sqoop.Sqoop.runTool(Sqoop.java:220)  at org.apache.sqoop.Sqoop.runTool(Sqoop.java:229)  at org.apache.sqoop.Sqoop.main(Sqoop.java:238) |
| mysql的字符集支持问题：  从mysql 的表中查询有乱码：  mysql> select \* from COLUMNS\_V2;  +-------+---------+-----------------+-----------+-------------+  | CD\_ID | COMMENT | COLUMN\_NAME | TYPE\_NAME | INTEGER\_IDX |  +-------+---------+-----------------+-----------+-------------+  | 91 | NULL | ?? | string | 1 |  | 91 | NULL | ?????? | double | 2 |  | 91 | NULL | ??id | double | 0 |  | 96 | NULL | ?? | string | 1 |  | 96 | NULL | ???????? | double | 2 |  | 96 | NULL | ??id | double | 0 |  | 101 | NULL | arrear\_trade | string | 4 |  mysql> show variables like "char%";  No connection. Trying to reconnect...  Connection id: 2  Current database: hive  +--------------------------+----------------------------+  | Variable\_name | Value |  +--------------------------+----------------------------+  | character\_set\_client | latin1 |  | character\_set\_connection | latin1 |  | character\_set\_database | latin1 |  | character\_set\_filesystem | binary |  | character\_set\_results | latin1 |  | character\_set\_server | latin1 |  | character\_set\_system | utf8 |  | character\_sets\_dir | /usr/share/mysql/charsets/ |  +--------------------------+----------------------------+  8 rows in set (0.00 sec)  mysql> show variables like "collation\_%";  +----------------------+-------------------+  | Variable\_name | Value |  +----------------------+-------------------+  | collation\_connection | latin1\_swedish\_ci |  | collation\_database | latin1\_swedish\_ci |  | collation\_server | latin1\_swedish\_ci |  +----------------------+-------------------+  3 rows in set (0.00 sec)  1.临时修改: alter database name character set utf8; ……  2.永久修改：修改mysql的配置my.cnf：路径在/etc/my.cnf或者/etc/mysql/my.cnf  [hadoop@master-hadoop etc]$ cat my.cnf  [mysqld]  datadir=/var/lib/mysql  socket=/var/lib/mysql/mysql.sock  user=mysql  # Disabling symbolic-links is recommended to prevent assorted security risks  symbolic-links=0  # utf-8  character-set-server=utf8  collation-server=utf8\_general\_ci  [mysql]  default-character-set=utf8  [client]  default-character-set=utf8  [mysqld\_safe]  log-error=/var/log/mysqld.log  pid-file=/var/run/mysqld/mysqld.pid |

# 预处理:hive

## 安装部署hive

### 配置文件

|  |
| --- |
| Linux命令行 |
| cp hive-default.xml.template hive**-site**.xml |
| <property>  <name>javax.jdo.option.ConnectionURL</name>  <value>jdbc:derby:;databaseName=/home/hadoop/data-hive-0.13/metastore\_db;create=true</value>  <description>JDBC connect string for a JDBC metastore</description>  </property> |

## Hello hive

|  |
| --- |
| Linux命令行 |
| echo 'X' > /tmp/dummy.txt  hive -e "CREATE TABLE dummy (value STRING); \  LOAD DATA LOCAL INPATH '/tmp/dummy.txt' \  OVERWRITE INTO TABLE dummy" |

|  |
| --- |
| Linux命令行(hadoop用户) |
| hive -S -e 'SELECT \* FROM dummy'  FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'dummy'  但在hive中，select语句可以执行成功。原因在于用户？ |

## 电力2014-辽宁沈阳

|  |
| --- |
| Hive命令行:创建表 |
| CREATE TABLE BIGDATA\_USER\_INFO\_S01  (  "用户ID" STRING,  "客户名称" STRING,  "电价代码" STRING,  "地市局编码" STRING,  "用电性质" STRING,  "用电类别" STRING,  "供电电压" STRING,  "合同容量" STRING,  "负荷类型" STRING,  "当前是否销户" STRING,  "城镇用户农村用户" STRING,  "出账周期" STRING,  "行业" STRING,  "高压低压" STRING,  "用户号" STRING  )  ROW FORMAT DELIMITED  FIELDS TERMINATED BY '\t';  FAILED: ParseException line 3:2 cannot recognize input near ''用户ID'' 'STRING' ',' in column specification  CREATE TABLE BIGDATA\_USER\_INFO\_S01  (  cons\_id STRING,  cons\_name STRING,  prc\_code STRING,  org\_no STRING,  volumn\_type STRING,  elec\_type\_code STRING,  volt\_code STRING,  contract\_cap STRING,  load\_attr\_code STRING,  status\_code STRING,  urban\_rural\_flag STRING,  period STRING,  trade\_code STRING,  cust\_type\_code STRING,  cons\_no STRING  )  ROW FORMAT DELIMITED  FIELDS TERMINATED BY '\t'; |

|  |
| --- |
| Hive命令行:装载数据,从HDFS |
| LOAD DATA INPATH ‘ds-oracle/epmpln/bigdata\_user\_info\_s01/\*’  OVERWRITE INTO TABLE BIGDATA\_USER\_INFO\_S01; |

|  |
| --- |
| Hive命令行:装载数据,从local |
| LOAD DATA LOCAL INPATH ‘~/workspace\_hadoop/bigdata\_user\_info\_s01/\*’  OVERWRITE INTO TABLE BIGDATA\_USER\_INFO\_S01; |