一、任务要求

1.1 目的

由市场使用情况来看，豆瓣网的影音社区是行业中做的比较好的，将影视评分做成几乎成了行业标准。从影评的角度来说，豆瓣网的影评内容发布门槛较低，任何人都可以在豆瓣上发表自己的短评或者长评，形成百花齐放但良莠不齐的影评内容。而我们做这个项目的初衷即将将豆瓣评论中的关键字，点赞数，评论及数量等信息呈现在用户面前，给用户该电影的直观评价。

1.2 内容和要求

（一)数据爬取：通过爬虫对豆瓣电影网的数据进行爬取并获取相关数据集；

（二）部署 Hadoop 环境：配置服务器；熟悉 Hadoop 三大组件并学会使用 HDFS 系统的 shell 命令以及 Java 的 Mapreduce 编程；

（三）部署 Hive：熟悉 ODS 层、DWD 层、DWS 层和 ADS 层的相关作用，并对数据集进行预处理，去除脏数据；

（四）安装 FinBI，完成基本配置和数据库的链接

（五）数据可视化：基于 Hive 处理后的数据，我们使用 FinBI 软件对数据进行可视化处理。

二、环境配置

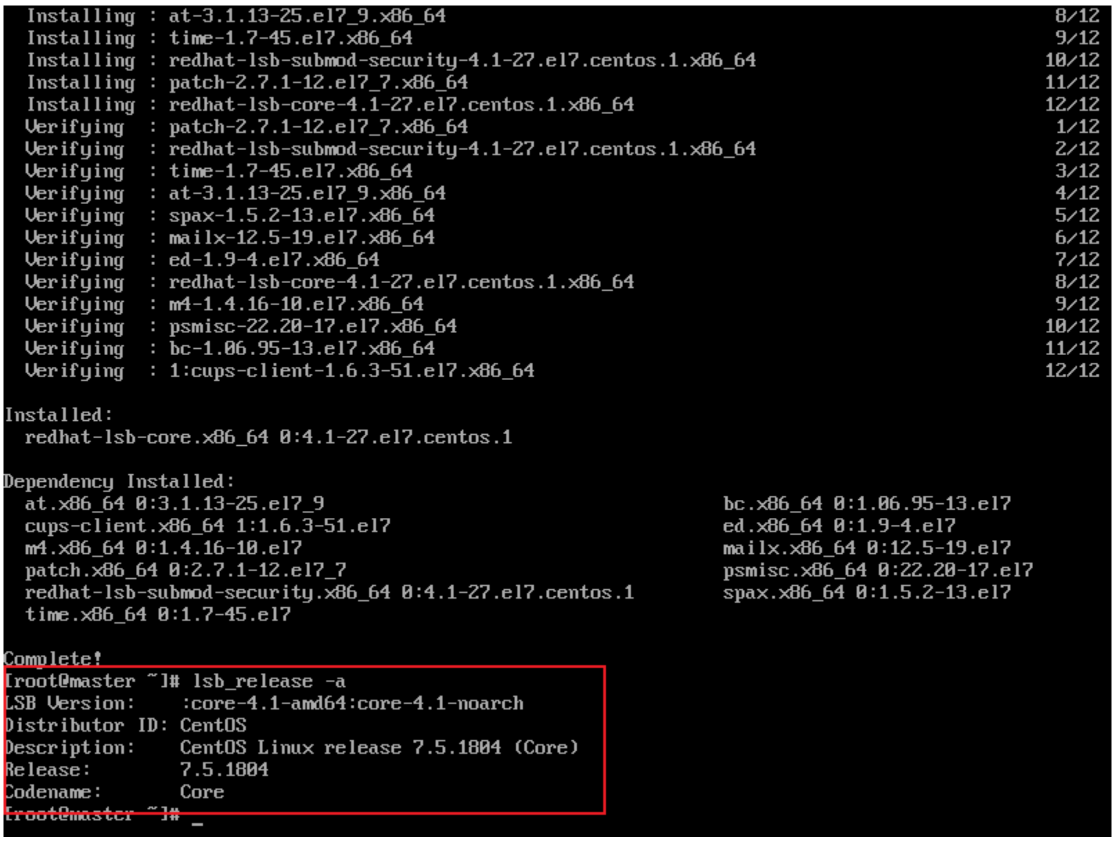
2.1 环境搭建

2.1.1 服务器配置和 Hadoop 集群搭建

为更好的实现 Hadoop 分布式系统搭建，本项目所需 3 台主机，分别为主节点 master，节点一 node1，节点二 node2。

（一）部署虚拟机基础环境

安装 VMware Workstation Pro，搭建 Linux 虚拟机，系统采用 Centos7



```
  Installing : at-3.1.13-25.el7_9.x86_64                                      8/12
  Installing : time-1.7-45.el7.x86_64                                         9/12
  Installing : redhat-lsb-submod-security-4.1-27.el7.centos.1.x86_64         10/12
  Installing : patch-2.7.1-12.el7_7.x86_64                                   11/12
  Installing : redhat-lsb-core-4.1-27.el7.centos.1.x86_64                    12/12
  Verifying  : patch-2.7.1-12.el7_7.x86_64                                    1/12
  Verifying  : redhat-lsb-submod-security-4.1-27.el7.centos.1.x86_64          2/12
  Verifying  : time-1.7-45.el7.x86_64                                         3/12
  Verifying  : at-3.1.13-25.el7_9.x86_64                                      4/12
  Verifying  : spax-1.5.2-13.el7.x86_64                                       5/12
  Verifying  : mailx-12.5-19.el7.x86_64                                       6/12
  Verifying  : ed-1.9-4.el7.x86_64                                            7/12
  Verifying  : redhat-lsb-core-4.1-27.el7.centos.1.x86_64                     8/12
  Verifying  : m4-1.4.16-10.el7.x86_64                                        9/12
  Verifying  : psmisc-22.20-17.el7.x86_64                                    10/12
  Verifying  : bc-1.06.95-13.el7.x86_64                                      11/12
  Verifying  : 1:cups-client-1.6.3-51.el7.x86_64                             12/12

Installed:
  redhat-lsb-core.x86_64 0:4.1-27.el7.centos.1

Dependency Installed:
  at.x86_64 0:3.1.13-25.el7_9                                 bc.x86_64 0:1.06.95-13.el7
  cups-client.x86_64 1:1.6.3-51.el7                           ed.x86_64 0:1.9-4.el7
  m4.x86_64 0:1.4.16-10.el7                                   mailx.x86_64 0:12.5-19.el7
  patch.x86_64 0:2.7.1-12.el7_7                               psmisc.x86_64 0:22.20-17.el7
  redhat-lsb-submod-security.x86_64 0:4.1-27.el7.centos.1     spax.x86_64 0:1.5.2-13.el7
  time.x86_64 0:1.7-45.el7

Complete!
[root@master ~]# lsb_release -a
LSB Version:    :core-4.1-amd64:core-4.1-noarch
Distributor ID: CentOS
Description:    CentOS Linux release 7.5.1804 (Core)
Release:        7.5.1804
Codename:       Core
[root@master ~]# _
```

图 1：系统版本号

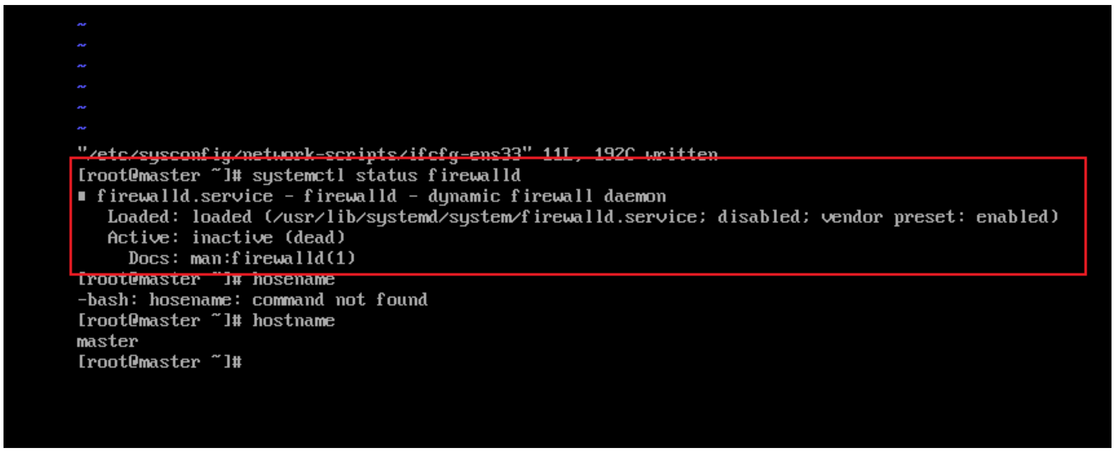
（二）配置主节点

1、修改主机名，并关闭防火墙：

关闭防火墙命令：systemctl stop firewalld

取消防火墙自启命令：systemctl disable firewalld

查询防火墙状态命令：systemctl status firewalld



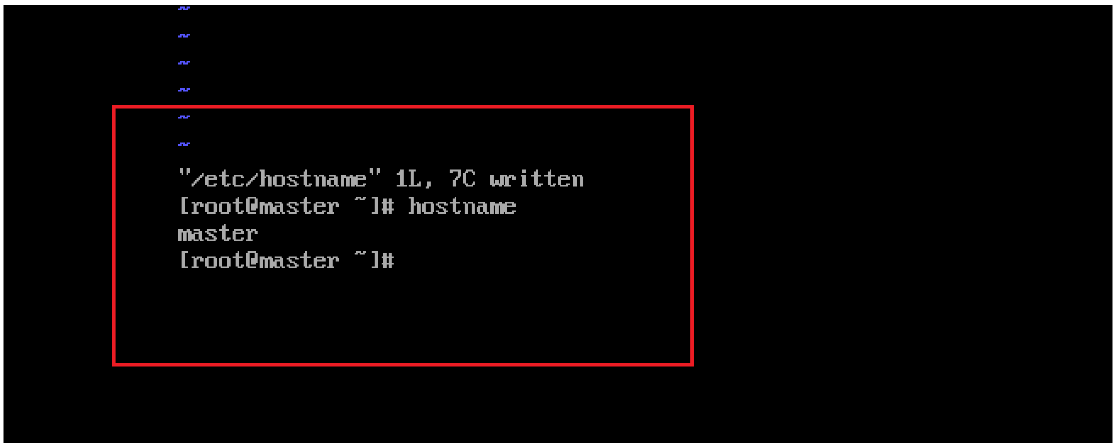
```
~
~
~
~
~
~
"/etc/sysconfig/network-scripts/ifcfg-ens33" 11L, 192C written
[root@master ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)
[root@master ~]# hosename
-bash: hosename: command not found
[root@master ~]# hostname
master
[root@master ~]#
```

图 2：查询防火墙状态

修改主机名：

vim /etc/hostname

三台主机节点分别为 master、node1、node2

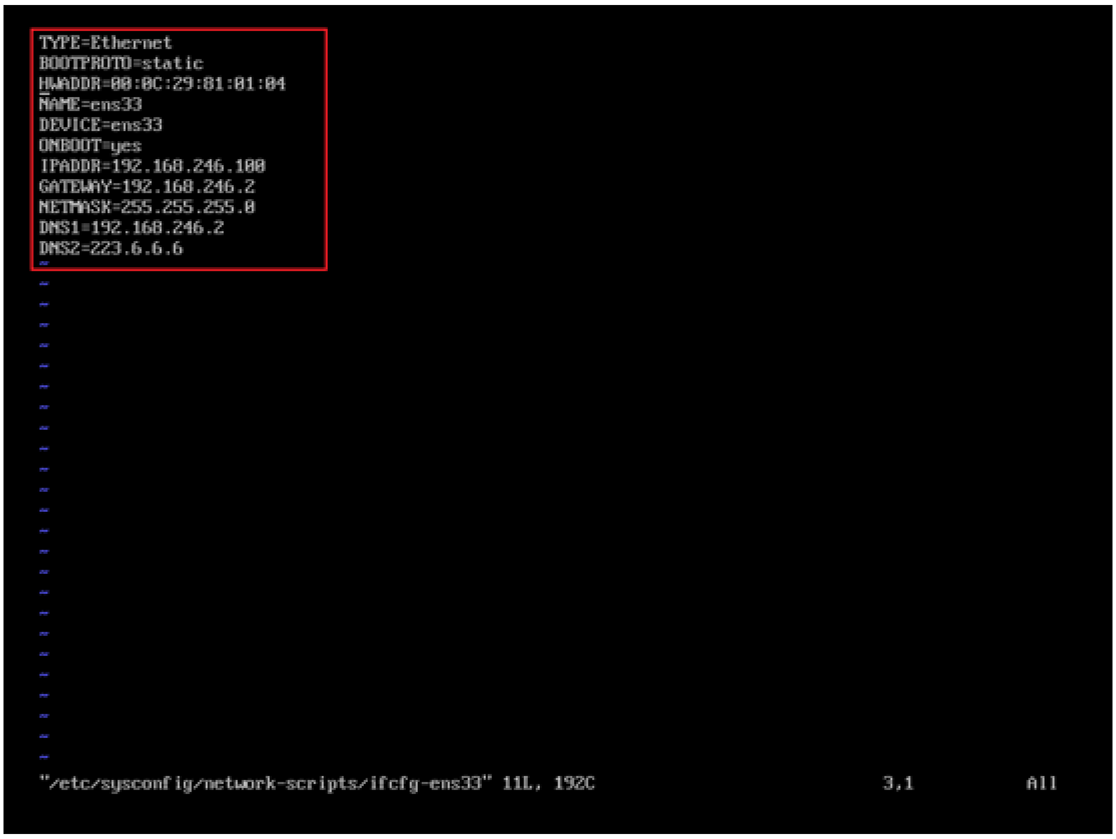


```
~
~
~
~
~
~
"/etc/hostname" 1L, 7C written
[root@master ~]# hostname
master
[root@master ~]#
```

图 3：主机名称

2、编辑网络配置文件，重启网络服务：

```
vim /etc/sysconfig/network-scripts/ifcfg-ens33
```



```
TYPE=Ethernet
BOOTPROTO=static
HWADDR=00:0C:29:01:01:04
NAME=ens33
DEVICE=ens33
ONBOOT=yes
IPADDR=192.168.246.100
GATEWAY=192.168.246.2
NETMASK=255.255.255.0
DNS1=192.168.246.2
DNS2=223.6.6.6
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/sysconfig/network-scripts/ifcfg-ens33" 11L, 192C                    3,1           All
```

图 3：网络配置文件修改后内容

关闭 NetworkManager，并取消开机自启命令：

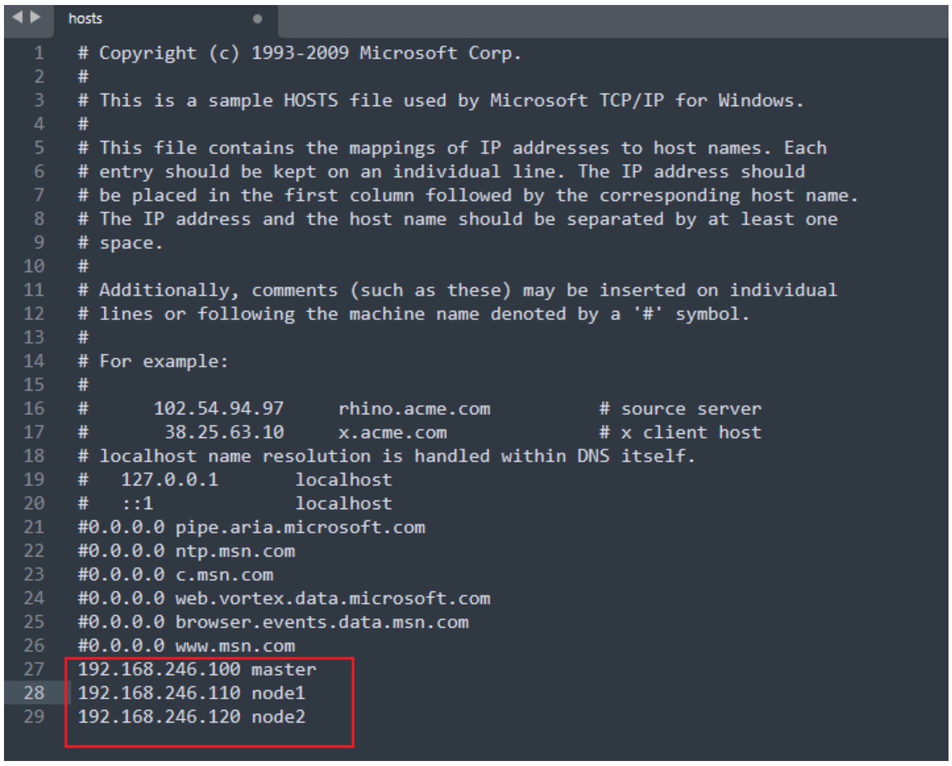
```
systemctl stop NetworkManager
systemctl disable NetworkManager
```

重启网络服务命令：

```
systemctl restart network
```

3、添加关系映射：

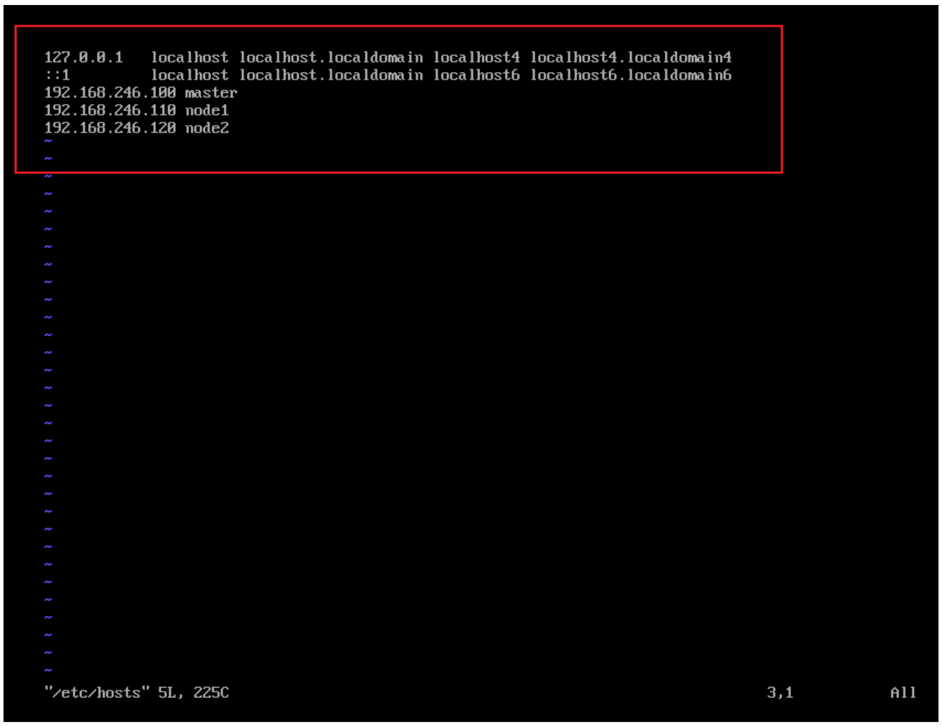
Windows 下：C:\Windows\System32\drivers\etc\hosts



```
hosts
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host
# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost
#0.0.0.0 pipe.aria.microsoft.com
#0.0.0.0 ntp.msn.com
#0.0.0.0 c.msn.com
#0.0.0.0 web.vortex.data.microsoft.com
#0.0.0.0 browser.events.data.msn.com
#0.0.0.0 www.msn.com
192.168.246.100 master
192.168.246.110 node1
192.168.246.120 node2
```

图 4：windows 下 hosts 文件修改后内容

Linux 下：vim /etc/hosts



```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.246.100 master
192.168.246.110 node1
192.168.246.120 node2
~
"/etc/hosts" 5L, 225C                                          3,1          All
```

图 3：hosts 文件修改后内容

4、安装 JDK

查询是否安装 Java 软件：rpm -qa|grep ″java″

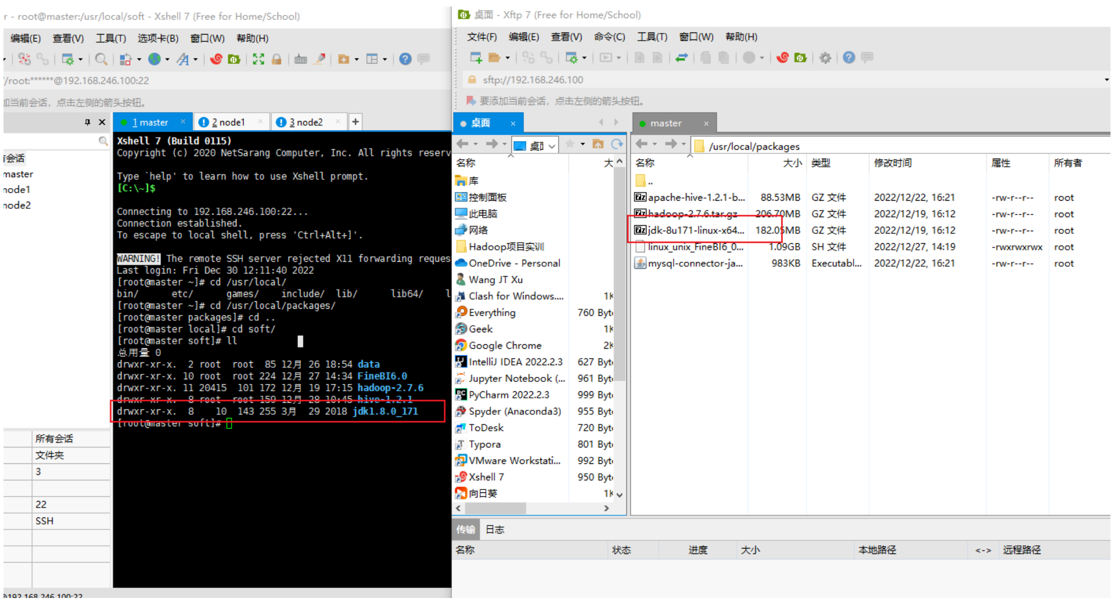
删除自带 JDK：rpm -e 文件名—nodeps

创建文件夹 mkdir -p /usr/local/soft

mkdir -p /usr/local/packages

使用远程链接软件 Xshell 和 XFPT 将 JDK 文件上传至 package 目录，并解压至 soft 目录下：

tar -zxvf jdk-8u171-linux-x64.tar.gz -C /usr/local/soft/



```
Xshell 7 (Build 0115)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserv

Type `help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 192.168.246.100:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

WARNING! The remote SSH server rejected X11 forwarding reques
Last login: Fri Dec 30 12:11:40 2022
[root@master ~]# cd /usr/local/
bin/      etc/      games/    include/  lib/      lib64/    l
[root@master ~]# cd /usr/local/packages/
[root@master packages]# cd ..
[root@master local]# cd soft/
[root@master soft]# ll
总用量 0
drwxr-xr-x.  2 root  root  85 12月 26 18:54 data
drwxr-xr-x. 10 root  root 224 12月 27 14:34 FineBI6.0
drwxr-xr-x. 11 20415  101 172 12月 19 17:15 hadoop-2.7.6
drwxr-xr-x.  8 root  root 150 12月 28 10:45 hive-1.2.1
drwxr-xr-x.  8    10  143 255 3月  29 2018 jdk1.8.0_171
[root@master soft]#
```

/usr/local/packages

名称	大小	类型	修改时间	属性	所有者
..					
apache-hive-1.2.1-b...	88.53MB	GZ 文件	2022/12/22, 16:21	-rw-r--r--	root
hadoop-2.7.6.tar.gz	206.70MB	GZ 文件	2022/12/19, 16:12	-rw-r--r--	root
jdk-8u171-linux-x64...	182.05MB	GZ 文件	2022/12/19, 16:12	-rw-r--r--	root
linux_unix_FineBI6_0...	1.09GB	SH 文件	2022/12/27, 14:19	-rwxrwxrwx	root
mysql-connector-ja...	983KB	Executabl...	2022/12/22, 16:21	-rw-r--r--	root

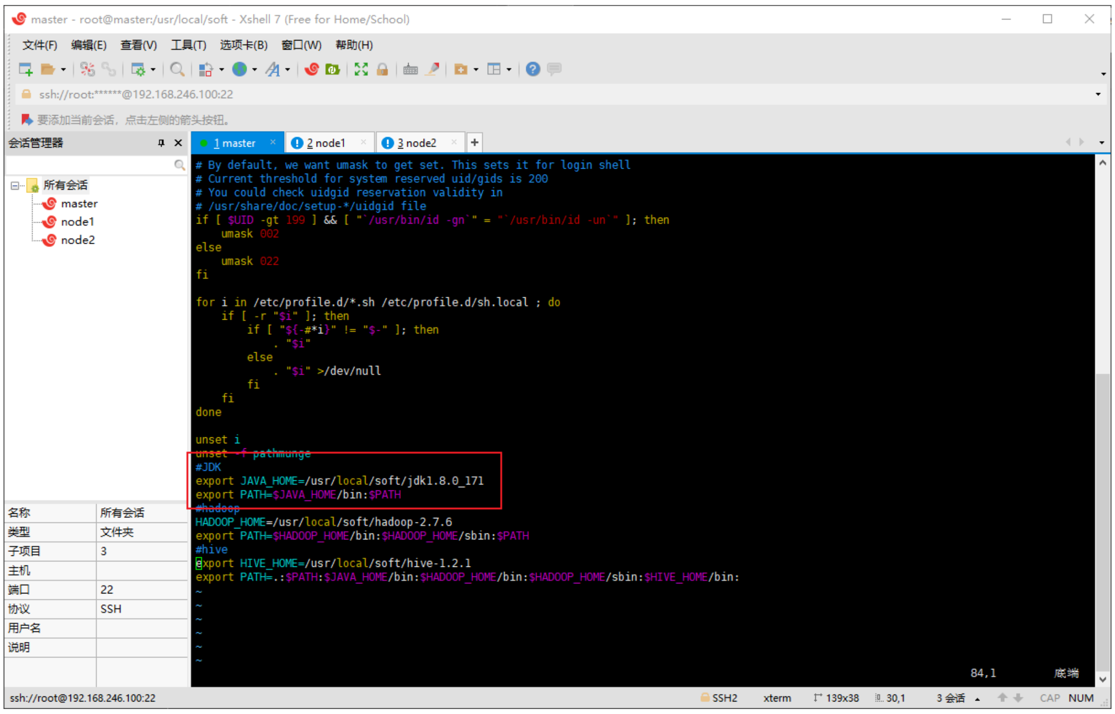
图 4：安装 JDK

配置环境变量，修改 profile 文件：vim /etc/profile

```
vim /etc/profile
#JDK
export JAVA_HOME=/usr/local/soft/jdk1.8.0_171
export PATH=$JAVA_HOME/bin:$PATH
```

刷新资源，使环境变量生效

```
source /etc/profile
```


master - root@master:/usr/local/soft - Xshell 7 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root:******@192.168.246.100:22
要添加当前会话，点击左侧的箭头按钮。
会话管理器
1 master
2 node1
3 node2
所有会话
master
node1
node2
ssh://root@192.168.246.100:22
SSH2
xterm
139x38
30,1
3 会话
CAP
NUM

名称	所有会话
类型	文件夹
子项目	3
主机	
端口	22
协议	SSH
用户名	
说明	

```
# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
#JDK
export JAVA_HOME=/usr/local/soft/jdk1.8.0_171
export PATH=$JAVA_HOME/bin:$PATH
#hadoop
HADOOP_HOME=/usr/local/soft/hadoop-2.7.6
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
#hive
export HIVE_HOME=/usr/local/soft/hive-1.2.1
export PATH=.:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin:
~
~
~
~
~
~
                                                                        84,1          底端
```

图 5：JDK 环境变量配置

（三）克隆主节点

根据 VMware Workstation Pro 功能特性和 Hadoop 分布式系统搭建的需要，将此台虚拟机额外克隆出两台当作另外两个节点，克隆完成后需要修改主机名分别为 node1、node2，并且关闭防火墙以及其自启功能，命令与步骤与上述一致。

配置 ssh 免密登录:

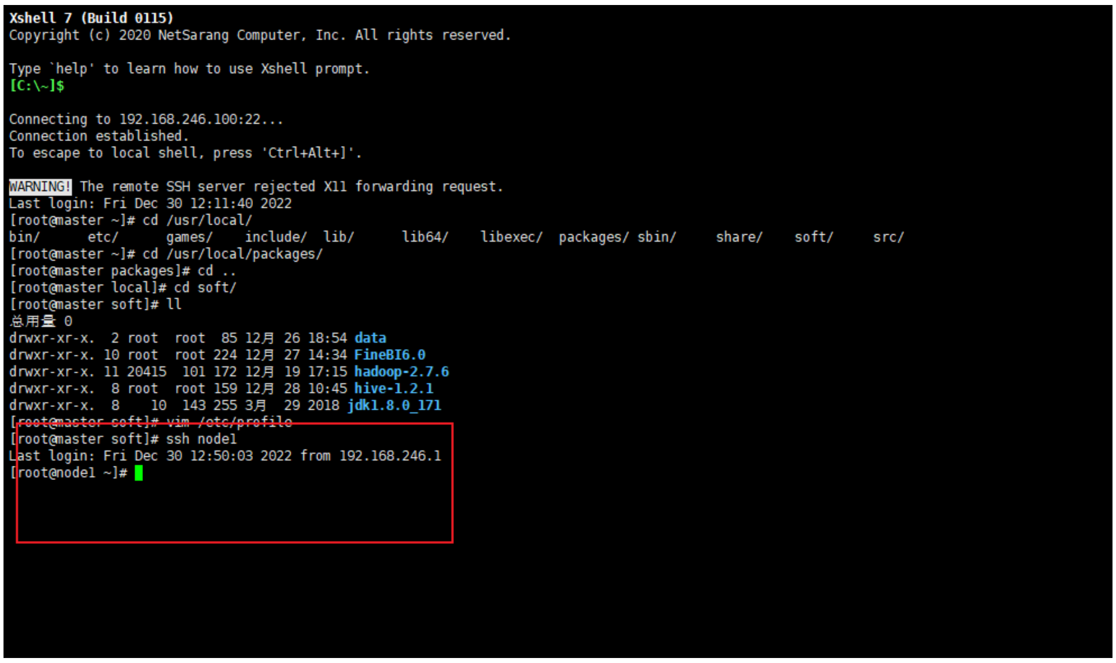
生成密钥：ssh-keygen -t rsa

ssh-copy-id master

ssh-copy-id node1

ssh-copy-id node2

测试 master 主机登录 node1 节点：ssh node1



```
Xshell 7 (Build 0115)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type `help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 192.168.246.100:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Fri Dec 30 12:11:40 2022
[root@master ~]# cd /usr/local/
bin/      etc/      games/    include/  lib/      lib64/    libexec/  packages/ sbin/     share/    soft/     src/
[root@master ~]# cd /usr/local/packages/
[root@master packages]# cd ..
[root@master local]# cd soft/
[root@master soft]# ll
总用量 0
drwxr-xr-x.  2 root  root  85 12月 26 18:54 data
drwxr-xr-x. 10 root  root 224 12月 27 14:34 FineBI6.0
drwxr-xr-x. 11 20415  101 172 12月 19 17:15 hadoop-2.7.6
drwxr-xr-x.  8 root  root 159 12月 28 10:45 hive-1.2.1
drwxr-xr-x.  8    10  143 255 3月  29 2018 jdk1.8.0_171
[root@master soft]# vim /etc/profile
[root@master soft]# ssh node1
Last login: Fri Dec 30 12:50:03 2022 from 192.168.246.1
[root@node1 ~]# 
```

图 6：ssh 免密登录

（四）部署 Hadoop 环境

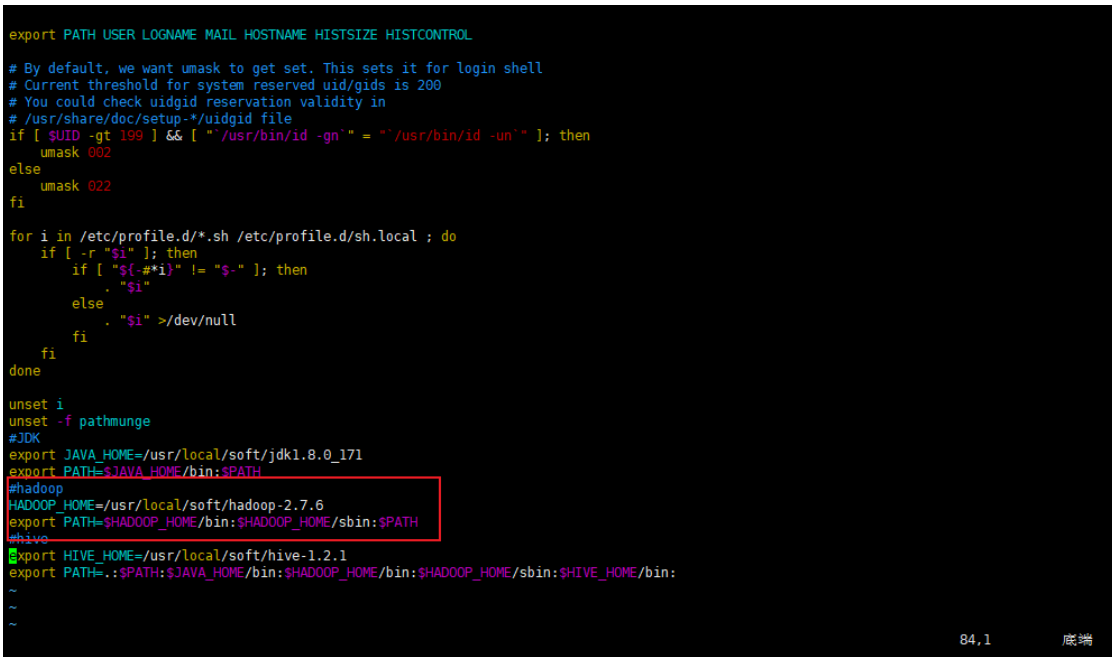
利用 XFTP 上传 hadoop 安装包至/usr/local/packages，并解压安装包至soft/目录下：tar -zxvf hadoop-2.7.6.tar.gz -C /usr/local/soft/

配置环境变量

```
#Hadoop
HADOOP_HOME=/usr/local/soft/hadoop-2.7.6
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

刷新资源，使环境变量生效

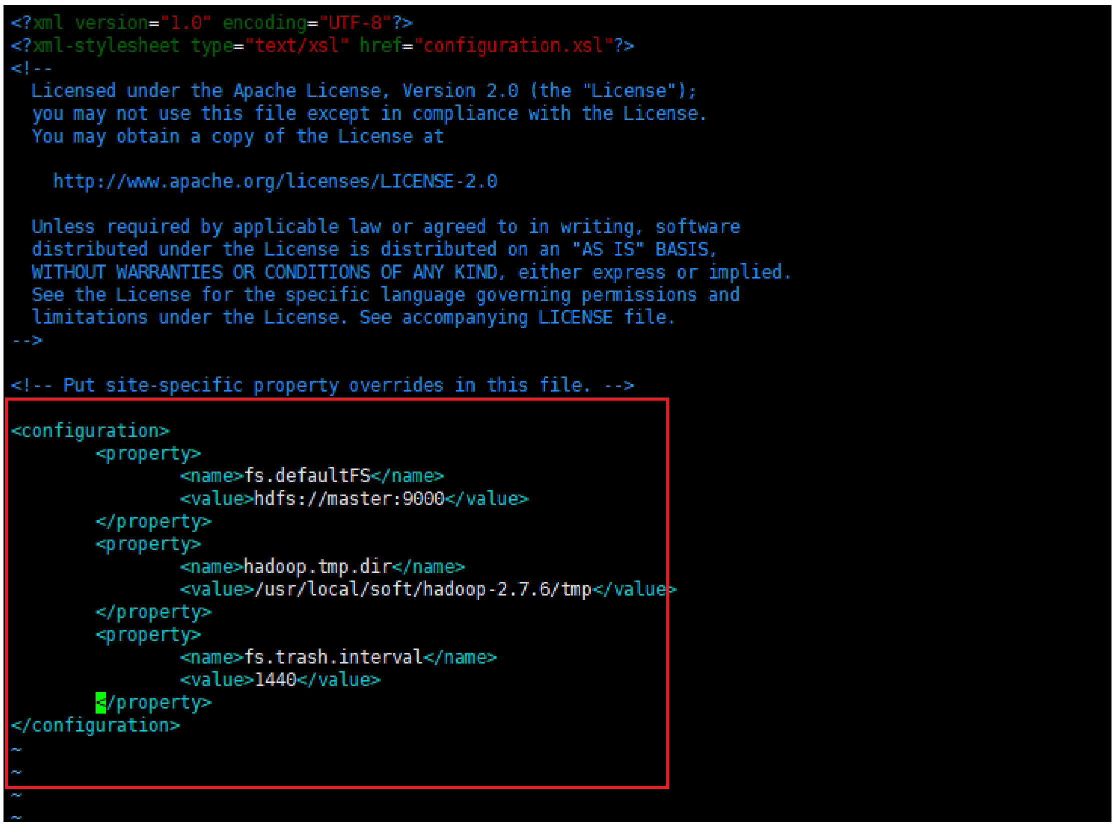
```
source /etc/profile
```



```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
#JDK
export JAVA_HOME=/usr/local/soft/jdk1.8.0_171
export PATH=$JAVA_HOME/bin:$PATH
#hadoop
HADOOP_HOME=/usr/local/soft/hadoop-2.7.6
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
#hive
export HIVE_HOME=/usr/local/soft/hive-1.2.1
export PATH=.:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin:
~
~
~
                                                                      84,1          底端
```

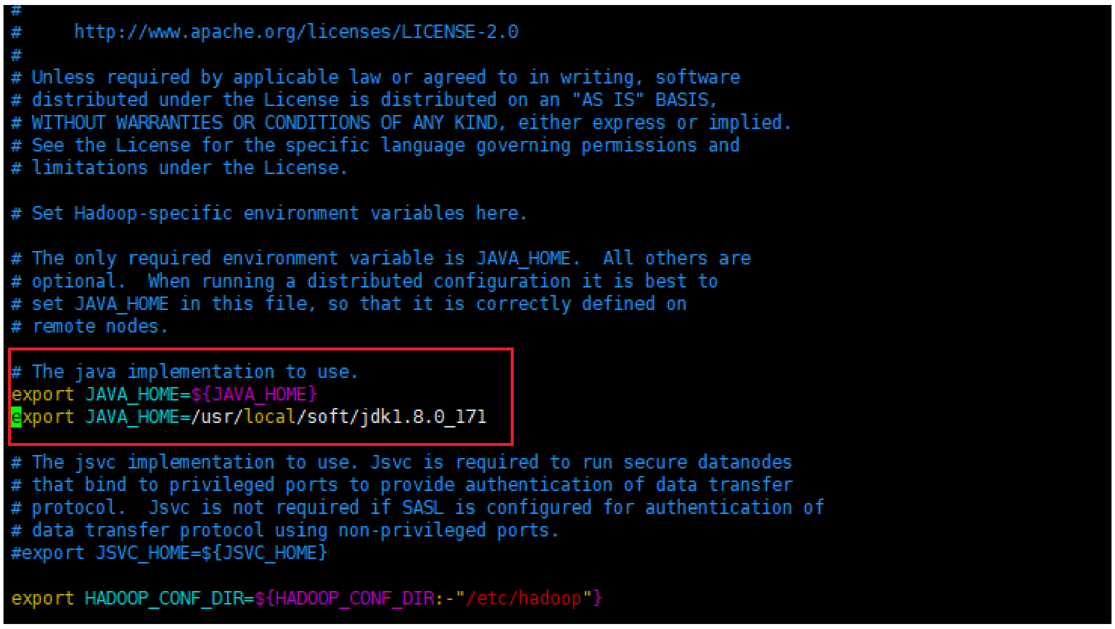
图 7：Hadoop 环境变量

完全分布式运行模式的集群搭建需要修改 6 个与 Hadoop 相关的配置文件，分别为：core-site.xml、hadoop-env.sh、hdfs-site.xml、mapred-site.xml、yarn-site.xml、slaves。图 2-8 到图 2-13 将分别展示各文件修改截图。



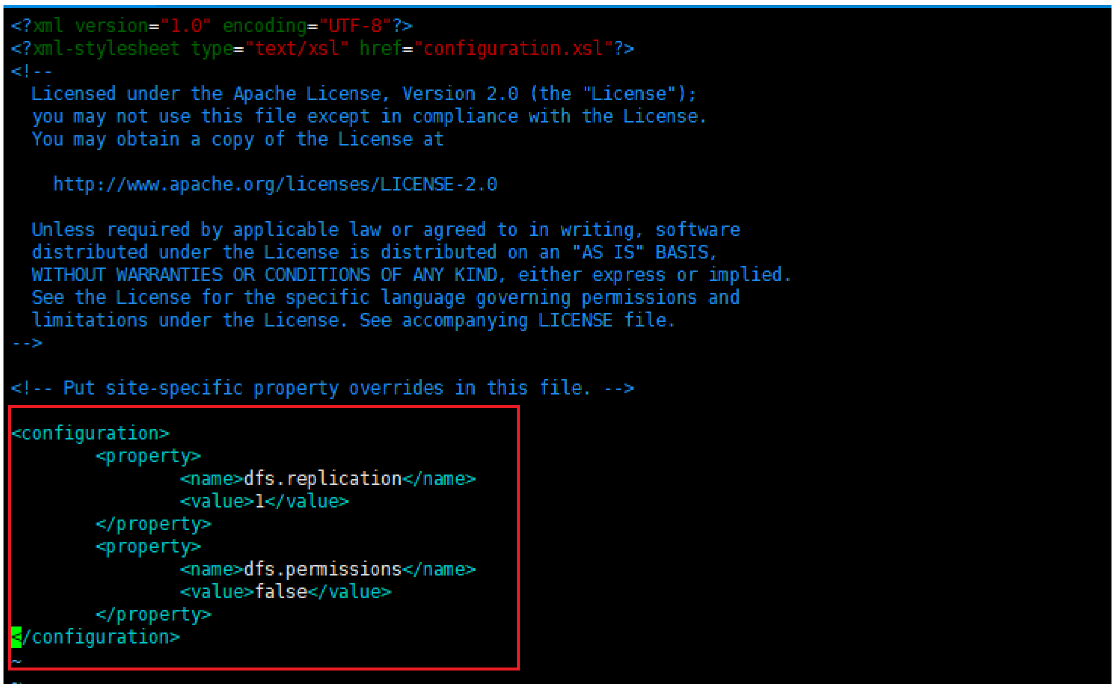
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://master:9000</value>
        </property>
        <property>
                <name>hadoop.tmp.dir</name>
                <value>/usr/local/soft/hadoop-2.7.6/tmp</value>
        </property>
        <property>
                <name>fs.trash.interval</name>
                <value>1440</value>
        </property>
</configuration>
~
~
~
~
```

图 8：修改 core-site.xml 文件



```
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/usr/local/soft/jdk1.8.0_171

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}
```

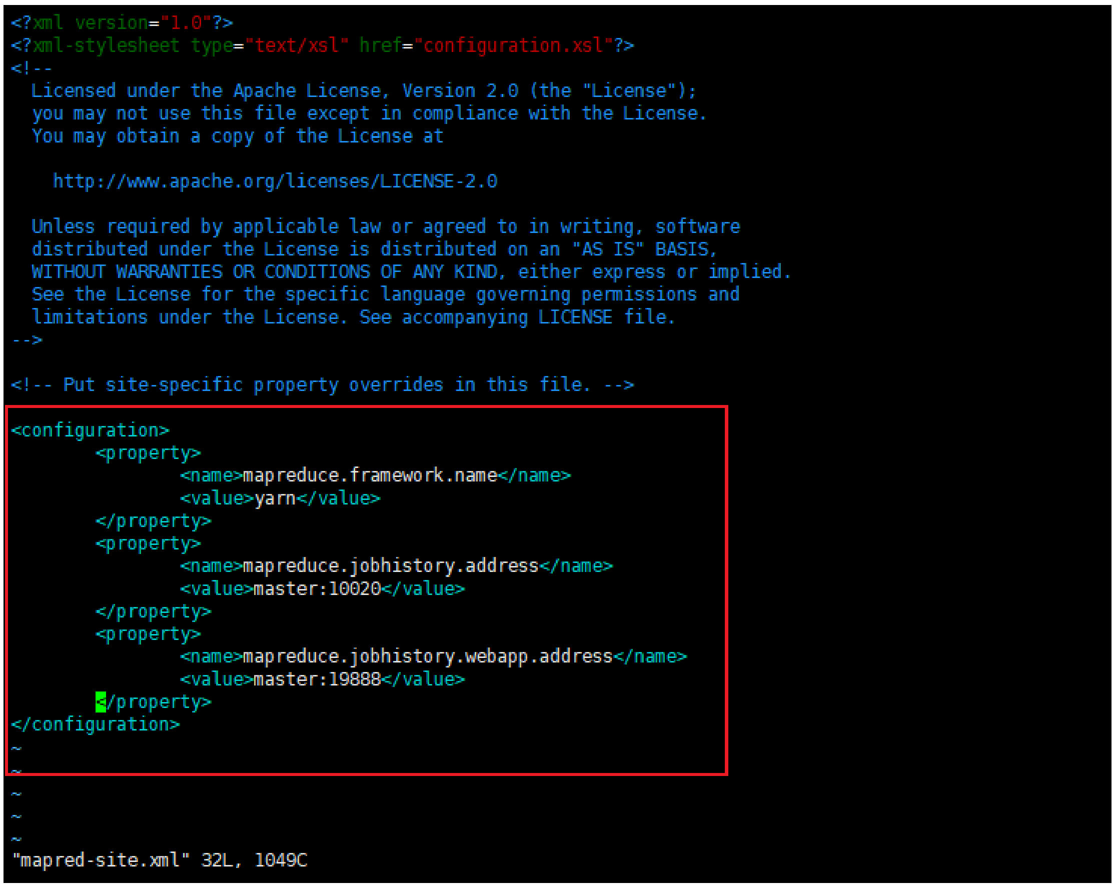
图 9：修改 hadoop-env.sh 文件



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>dfs.replication</name>
                <value>1</value>
        </property>
        <property>
                <name>dfs.permissions</name>
                <value>false</value>
        </property>
</configuration>
~
```

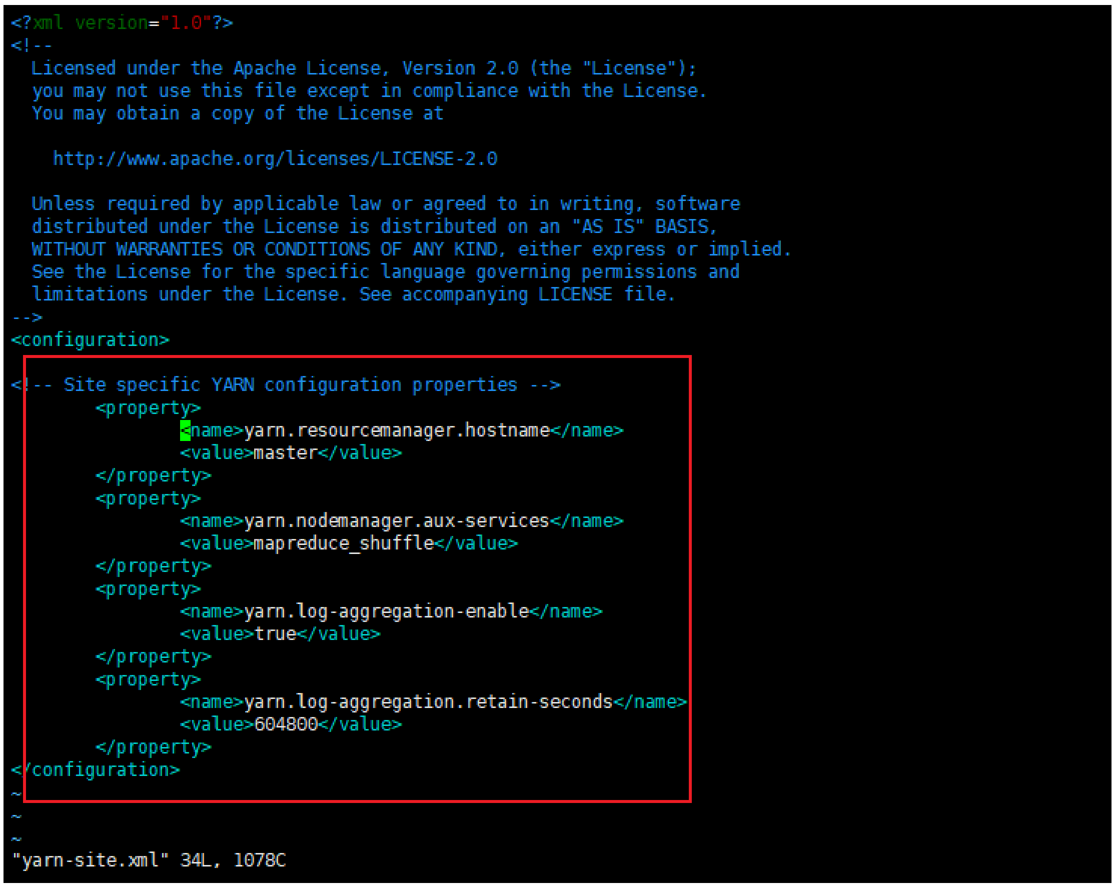
图 10：修改 hdfs-site.xml 文件

重命名文件：cp mapred-site.xml.template mapred-site.xml



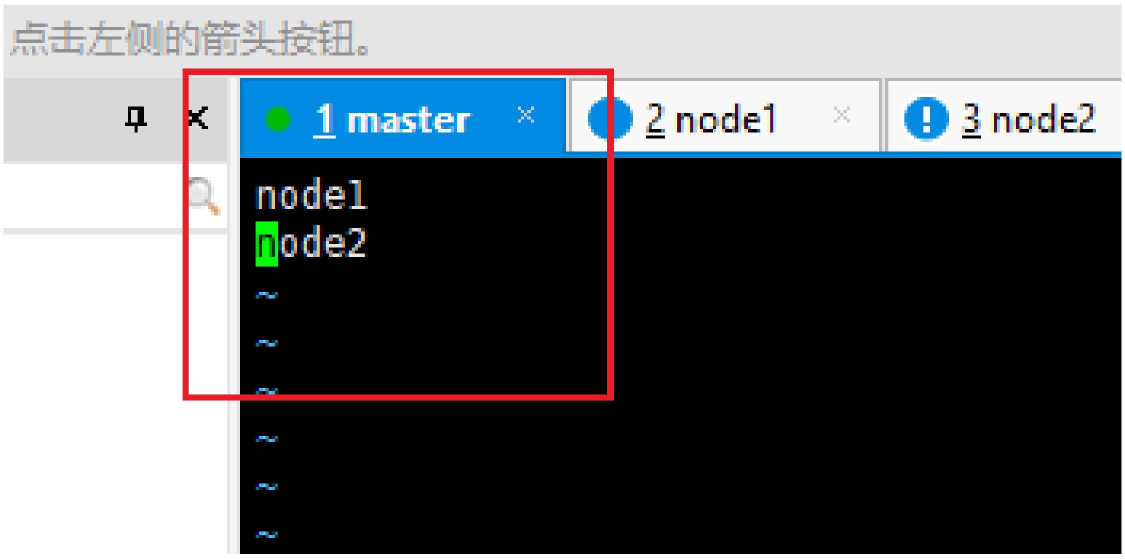
```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>mapreduce.framework.name</name>
                <value>yarn</value>
        </property>
        <property>
                <name>mapreduce.jobhistory.address</name>
                <value>master:10020</value>
        </property>
        <property>
                <name>mapreduce.jobhistory.webapp.address</name>
                <value>master:19888</value>
        </property>
</configuration>
~
~
~
~
~
"mapred-site.xml" 32L, 1049C
```

图 11：修改 mapred-site.xml 文件



```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
        <property>
                <name>yarn.resourcemanager.hostname</name>
                <value>master</value>
        </property>
        <property>
                <name>yarn.nodemanager.aux-services</name>
                <value>mapreduce_shuffle</value>
        </property>
        <property>
                <name>yarn.log-aggregation-enable</name>
                <value>true</value>
        </property>
        <property>
                <name>yarn.log-aggregation.retain-seconds</name>
                <value>604800</value>
        </property>
</configuration>
~
~
~
"yarn-site.xml" 34L, 1078C
```

图 12：修改 yarn-site.xml 文件



```
node1
node2
~
~
~
~
~
~
```

图 13：修改 slaves 文件

分发 Hadoop 文件到 node1 和 node2 节点

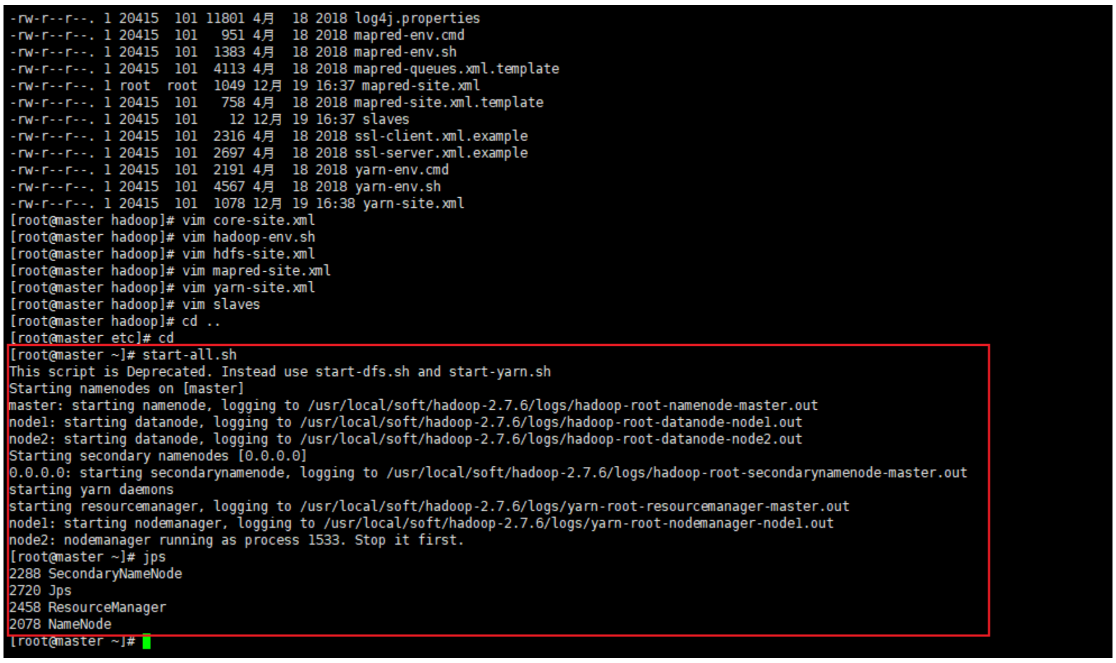
cd /usr/local/soft/

scp -r hadoop-2.7.6/ node1:\`pwd\`

scp -r hadoop-2.7.6/ node2:\`pwd\`

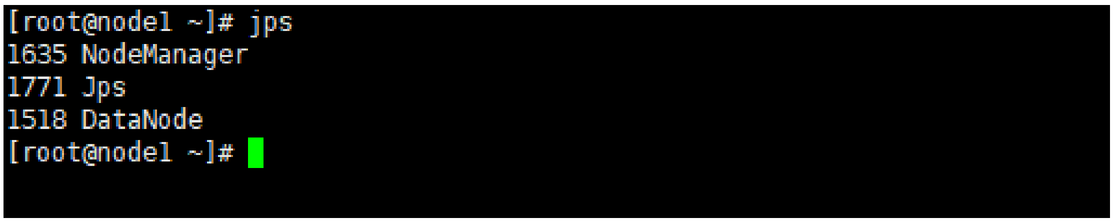
格式化 namenode（第一次启动的时候需要执行）：hdfs namenode -format

启动 hadoop 集群：start-all.sh，并查看进程：jps

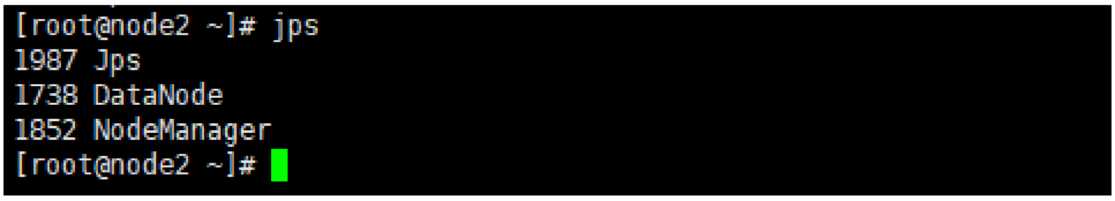


```
-rw-r--r--. 1 20415  101 11801 4月   18 2018 log4j.properties
-rw-r--r--. 1 20415  101   951 4月   18 2018 mapred-env.cmd
-rw-r--r--. 1 20415  101  1383 4月   18 2018 mapred-env.sh
-rw-r--r--. 1 20415  101  4113 4月   18 2018 mapred-queues.xml.template
-rw-r--r--. 1 root  root  1049 12月  19 16:37 mapred-site.xml
-rw-r--r--. 1 20415  101   758 4月   18 2018 mapred-site.xml.template
-rw-r--r--. 1 20415  101    12 12月  19 16:37 slaves
-rw-r--r--. 1 20415  101  2316 4月   18 2018 ssl-client.xml.example
-rw-r--r--. 1 20415  101  2697 4月   18 2018 ssl-server.xml.example
-rw-r--r--. 1 20415  101  2191 4月   18 2018 yarn-env.cmd
-rw-r--r--. 1 20415  101  4567 4月   18 2018 yarn-env.sh
-rw-r--r--. 1 20415  101  1078 12月  19 16:38 yarn-site.xml
[root@master hadoop]# vim core-site.xml
[root@master hadoop]# vim hadoop-env.sh
[root@master hadoop]# vim hdfs-site.xml
[root@master hadoop]# vim mapred-site.xml
[root@master hadoop]# vim yarn-site.xml
[root@master hadoop]# vim slaves
[root@master hadoop]# cd ..
[root@master etc]# cd
[root@master ~]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-namenode-master.out
node1: starting datanode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-datanode-node1.out
node2: starting datanode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-datanode-node2.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-secondarynamenode-master.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/soft/hadoop-2.7.6/logs/yarn-root-resourcemanager-master.out
node1: starting nodemanager, logging to /usr/local/soft/hadoop-2.7.6/logs/yarn-root-nodemanager-node1.out
node2: nodemanager running as process 1533. Stop it first.
[root@master ~]# jps
2288 SecondaryNameNode
2720 Jps
2458 ResourceManager
2078 NameNode
[root@master ~]#
```

图 14：启动 Hadoop，并查看进程



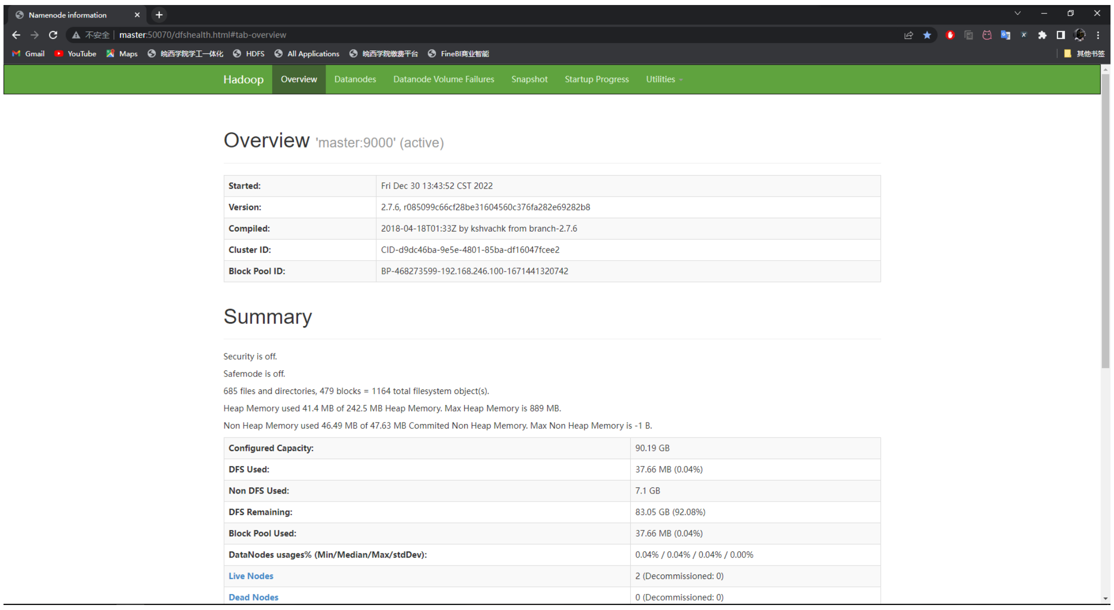
```
[root@node1 ~]# jps
1635 NodeManager
1771 Jps
1518 DataNode
[root@node1 ~]#
```



```
[root@node2 ~]# jps
1987 Jps
1738 DataNode
1852 NodeManager
[root@node2 ~]#
```

图 15：node1 和 node2 节点进程

http://master:50070


Namenode information

master:50070/dfshealth.html#tab-overview

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Overview 'master:9000' (active)

Started:	Fri Dec 30 13:43:52 CST 2022
Version:	2.7.6, r085099c66cf28be31604560c376fa282e69282b8
Compiled:	2018-04-18T01:33Z by kshvachk from branch-2.7.6
Cluster ID:	CID-d9dc46ba-9e5e-4801-85ba-df16047fcee2
Block Pool ID:	BP-468273599-192.168.246.100-1671441320742

Summary

Security is off.

Safemode is off.

685 files and directories, 479 blocks = 1164 total filesystem object(s).

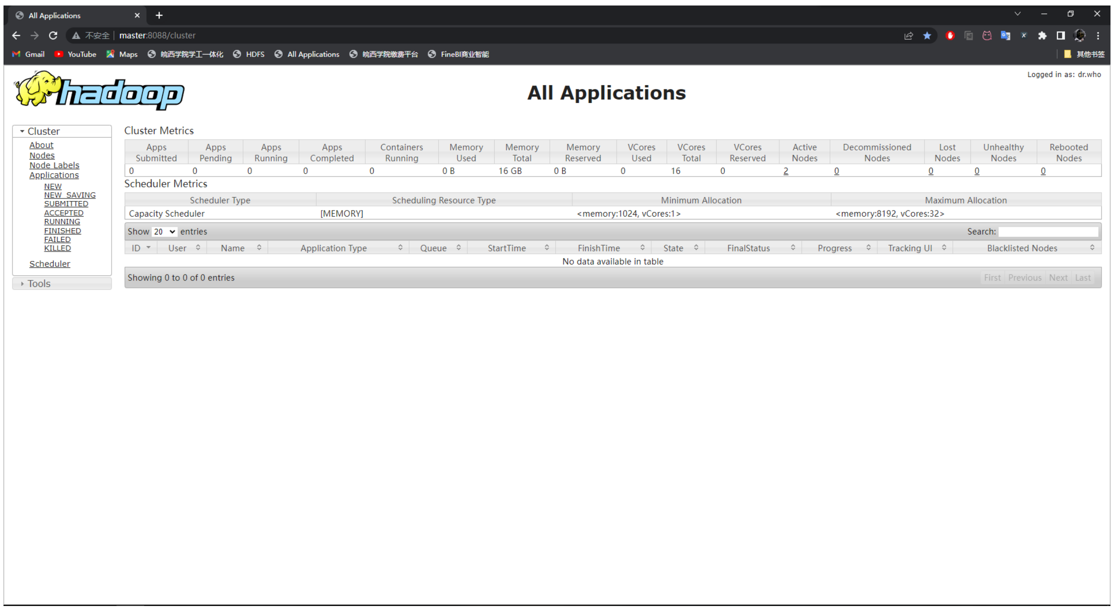
Heap Memory used 41.4 MB of 242.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 46.49 MB of 47.63 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	90.19 GB
DFS Used:	37.66 MB (0.04%)
Non DFS Used:	7.1 GB
DFS Remaining:	83.05 GB (92.08%)
Block Pool Used:	37.66 MB (0.04%)
DataNodes usages% (Min/Median/Max/stdDev):	0.04% / 0.04% / 0.04% / 0.00%
Live Nodes	2 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)

图 16：访问 HDFS 的 WEB 界面

http://master:8088/cluster


All Applications

master:8088/cluster

Logged in as: dr.who

All Applications

Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	16 GB	0 B	0	16	0	2	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:32>

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
No data available in table											

Showing 0 to 0 of 0 entries First Previous Next Last

图 17：访问 YARN 的 WEB 界面

2.1.2 任务环境部署

（一）部署 MySQL

1、下载 yum Repository：

wget -i -c http://dev.mysql.com/get/mysql57-community-release-el7-10.noarch.rpm

2、安装 yum Repository：

yum -y install mysql57-community-release-el7-10.noarch.rpm

3、升级 GPG：

rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022

4、安装 MySQL5.7 服务：

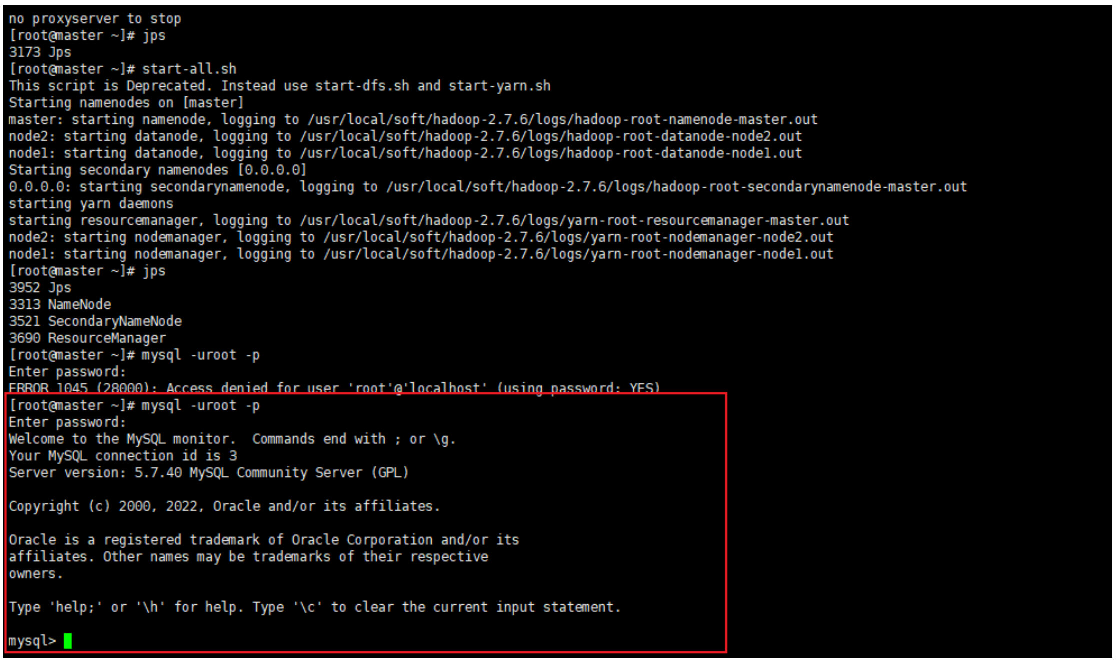
yum -y install mysql-community-server

5、开机自启动：systemctl enable mysqld.service

6、启动 MySQL：systemctl start mysqld.service

在以上设置设置好后，我们还需要获取 MySQL 临时密码，通过临时密码进入 MySQL，然后对 MySQL 进一步设置，修改密码；本设备已经提前设置好

7、获取临时密码：grep ″password″ /var/log/mysqld.log



```
no proxyserver to stop
[root@master ~]# jps
3173 Jps
[root@master ~]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-namenode-master.out
node2: starting datanode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-datanode-node2.out
node1: starting datanode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-datanode-node1.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-secondarynamenode-master.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/soft/hadoop-2.7.6/logs/yarn-root-resourcemanager-master.out
node2: starting nodemanager, logging to /usr/local/soft/hadoop-2.7.6/logs/yarn-root-nodemanager-node2.out
node1: starting nodemanager, logging to /usr/local/soft/hadoop-2.7.6/logs/yarn-root-nodemanager-node1.out
[root@master ~]# jps
3952 Jps
3313 NameNode
3521 SecondaryNameNode
3690 ResourceManager
[root@master ~]# mysql -uroot -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
[root@master ~]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

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> 
```

图 18：MySQL 成功安装并启动界面

8、关闭密码复杂验证：

set global validate_password_policy=0;

set global validate_password_length=1;

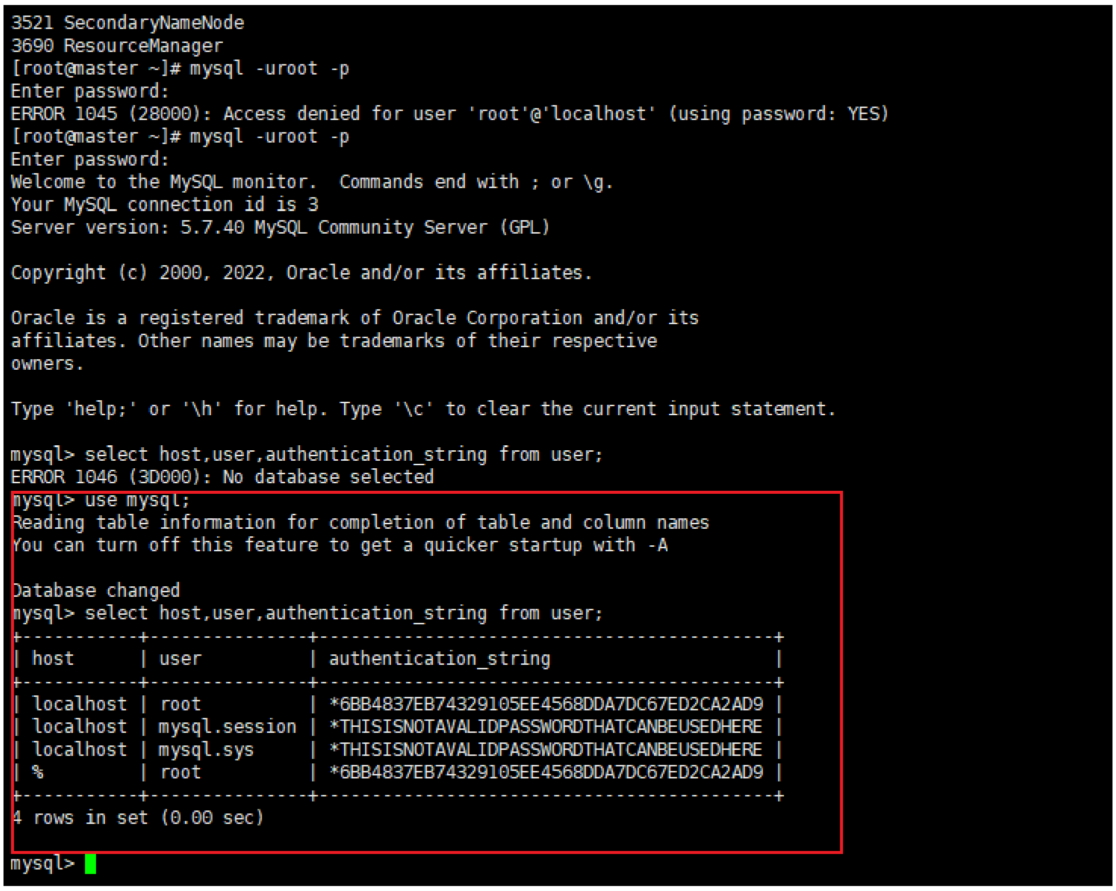
9、设置密码：alter user user() identified by "123456";

10、修改权限：use mysql;

GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY '123456' WITH GRANTOPTION; --修改权限

flush privileges; --刷新权限

select host,user,authentication_string from user; --查看权限



```
3521 SecondaryNameNode
3690 ResourceManager
[root@master ~]# mysql -uroot -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
[root@master ~]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

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> select host,user,authentication_string from user;
ERROR 1046 (3D000): No database selected
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select host,user,authentication_string from user;
+-----------+---------------+-------------------------------------------+
| host      | user          | authentication_string                     |
+-----------+---------------+-------------------------------------------+
| localhost | root          | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| localhost | mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| localhost | mysql.sys     | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| %         | root          | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
+-----------+---------------+-------------------------------------------+
4 rows in set (0.00 sec)

mysql>
```

图 19：MySQL 修改权限

（二）部署 Hive 数据库

前提：启动 Hadoop：start-all.sh

1、解压上传的文件至 soft/

tar -zxvf /usr/local/soft/module/apache-hive-1.2.1-bin.tar.gz -C /usr/local/soft/

2、修改目录名称为 hive-1.2.1：mv apache-hive-1.2.1-bin/ hive-1.2.1

3、备份配置文件：cd /usr/local/soft/hive-1.2.1/conf

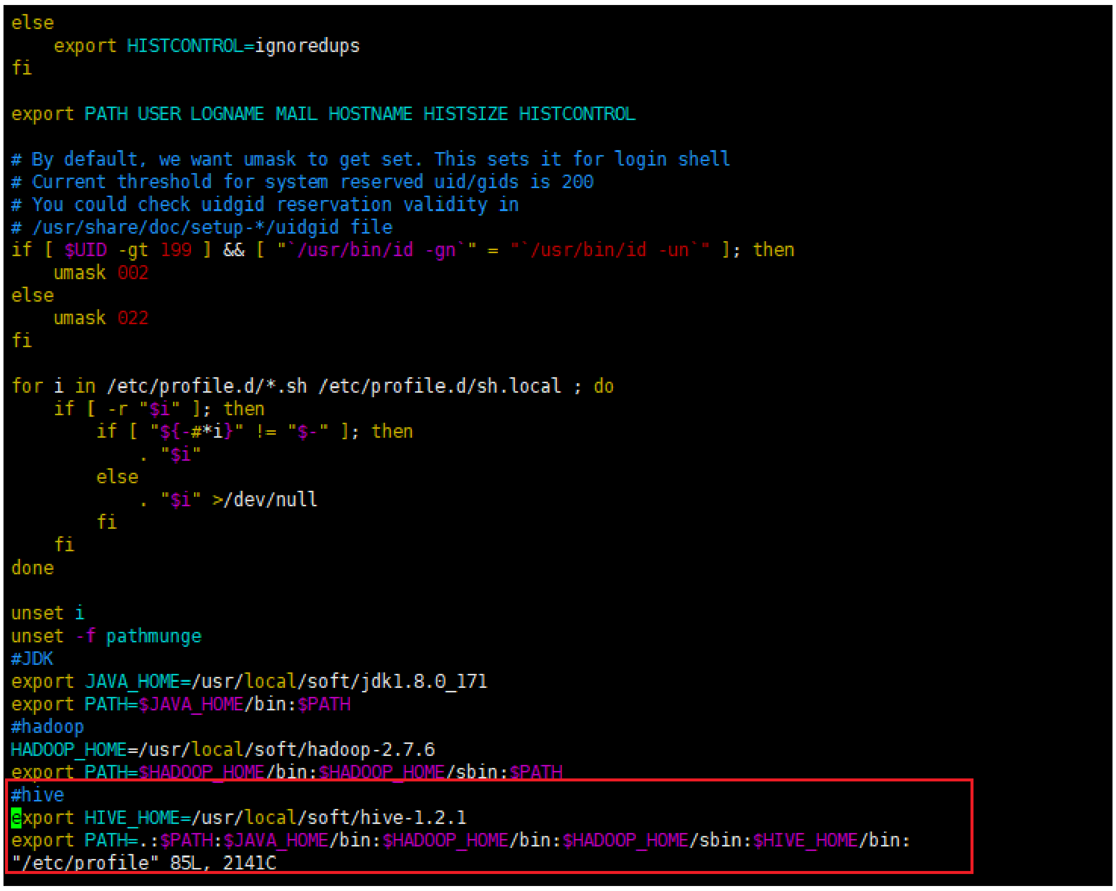
cp hive-env.sh.template hive-env.sh

cp hive-default.xml.template hive-site.xml

4、修改配置文件：vim hive-env.sh

export HIVE_HOME=/usr/local/soft/hive-1.2.1

export PATH=.:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin:



```
else
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
#JDK
export JAVA_HOME=/usr/local/soft/jdk1.8.0_171
export PATH=$JAVA_HOME/bin:$PATH
#hadoop
HADOOP_HOME=/usr/local/soft/hadoop-2.7.6
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
#hive
export HIVE_HOME=/usr/local/soft/hive-1.2.1
export PATH=.:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin:
"/etc/profile" 85L, 2141C
```

图 20：Hive 环境变量配置

5、修改 hive-site.xml 对应位置的配置参数

```
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:mysql://master:3306/hive?characterEncoding=UTF-8&amp;
createDatabaseIfNotExist=true&amp;useSSL=false</value>
</property>
<property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>com.mysql.jdbc.Driver</value>
</property>
<property>
<name>javax.jdo.option.ConnectionUserName</name>
<value>root</value>
</property>
<property>
<name>javax.jdo.option.ConnectionPassword</name>
<value>123456</value>
</property>
<property>
<name>hive.querylog.location</name>
<value>/usr/local/soft/hive-1.2.1/tmp</value>
</property>
<property>
<name>hive.exec.local.scratchdir</name>
<value>/usr/local/soft/hive-1.2.1/tmp</value>
</property>
<property>
<name>hive.downloaded.resources.dir</name>
<value>/usr/local/soft/hive-1.2.1/tmp</value>
</property>
<property>
<name>hive.cli.print.header</name>
<value>true</value>
<description>Whether to print the names of the columns in query
output.</description>
</property>
<property>
<name>hive.cli.print.current.db</name>
<value>true</value>
<description>Whether to include the current database in the Hive
prompt.</description>
</property>
```

6、复制 mysql 连接工具包到 hive/lib

cd /usr/local/soft/hive-1.2.1

cp /usr/local/moudle/mysql-connector-java-5.1.49.jar /usr/local/soft/hive-1.2.1/lib/

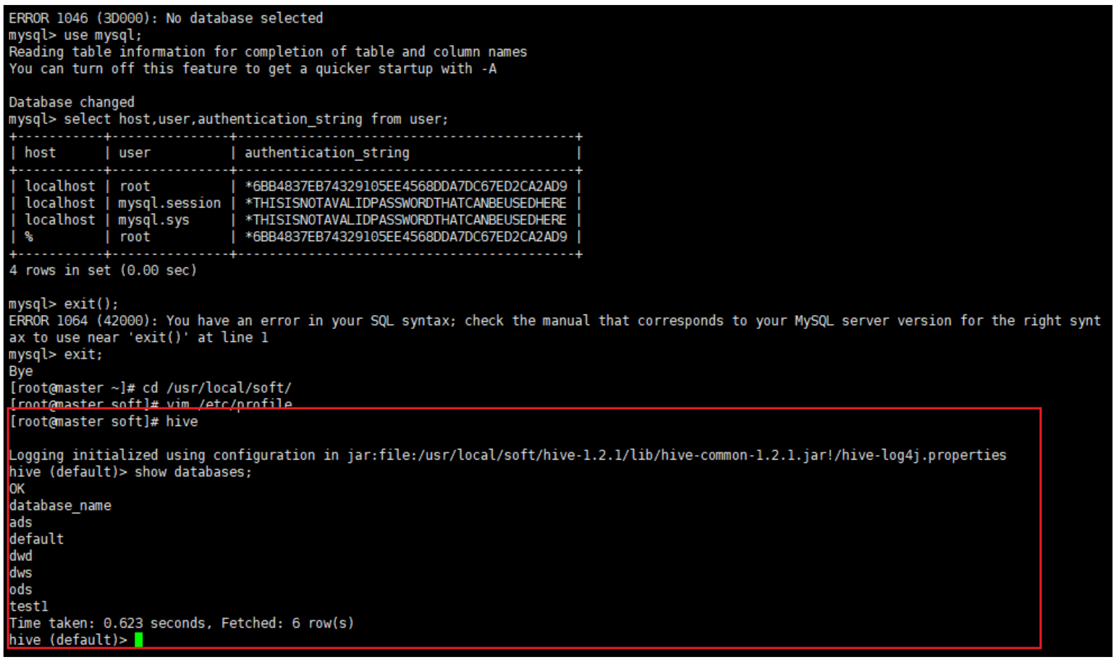
7、删除 hadoop 中自带的 jline-2.12.jar 位置在 /usr/local/soft/hadoop-2.7.6/share/hadoop/yarn/lib/jline-2.12.jar

rm -rf /usr/local/soft/hadoop-2.7.6/share/hadoop/yarn/lib/jline-2.12.jar

8、把 hive 自带的 jline-2.12.jar 复制到 hadoop 中 hive 中所在位置 /usr/local/soft/hive-1.2.1/lib/jline-2.12.jar

cp /usr/local/soft/hive-1.2.1/lib/jline-2.12.jar /usr/local/soft/hadoop-2.7.6/share/hadoop/yarn/lib/

9、验证、启动 Hive，并创建 ODS、DWD、DWS、ADS 四层数据库



```
ERROR 1046 (3D000): No database selected
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select host,user,authentication_string from user;
+-----------+---------------+-------------------------------------------+
| host      | user          | authentication_string                     |
+-----------+---------------+-------------------------------------------+
| localhost | root          | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| localhost | mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| localhost | mysql.sys     | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| %         | root          | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
+-----------+---------------+-------------------------------------------+
4 rows in set (0.00 sec)

mysql> exit();
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right synt
ax to use near 'exit()' at line 1
mysql> exit;
Bye
[root@master ~]# cd /usr/local/soft/
[root@master soft]# vim /etc/profile
[root@master soft]# hive

Logging initialized using configuration in jar:file:/usr/local/soft/hive-1.2.1/lib/hive-common-1.2.1.jar!/hive-log4j.properties
hive (default)> show databases;
OK
database_name
ads
default
dwd
dws
ods
test1
Time taken: 0.623 seconds, Fetched: 6 row(s)
hive (default)>
```

图 21：Hive 环境成功部署

（三）部署 FinBI 可视化软件

1、在 FinBI 官网下载好 linux 版本的安装包之后，利用 XFTP 软件将文件传输至 packages 文件夹下，修改权限 chmod 777 linux_unix_FineBI6_0-CN.sh

2、执行安装命令：./linux_unix_FineBI6_0-CN.sh，接下来跟着系统提示即可完成。

3、FinBI 完成后，进入 bin 目录下，输入./finebi 即可启动完成，在浏览器中输入 http://localhost:37799/webroot/decision（需要把 localhost 修改成 master 的 ip 地址）

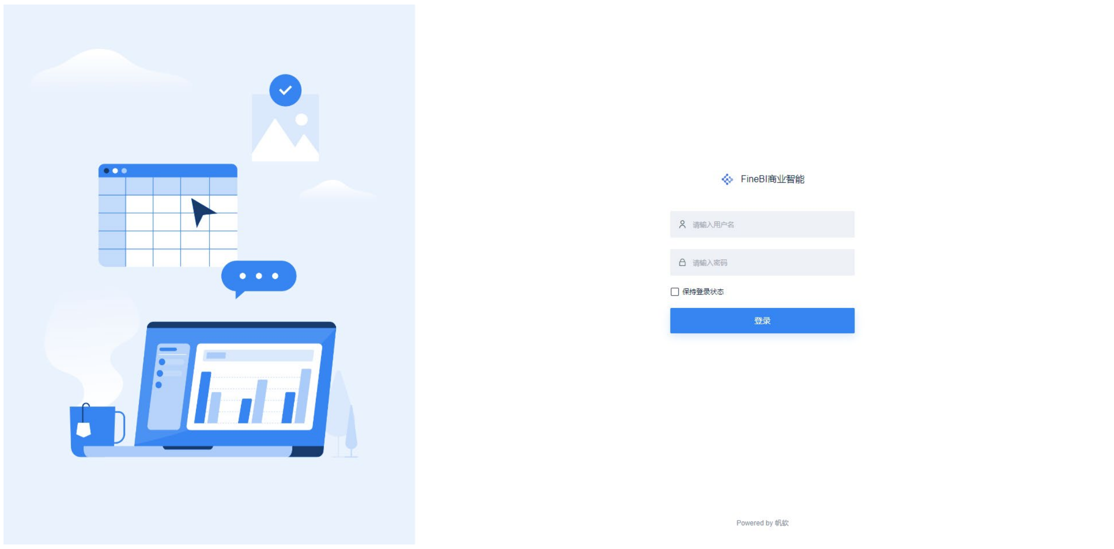
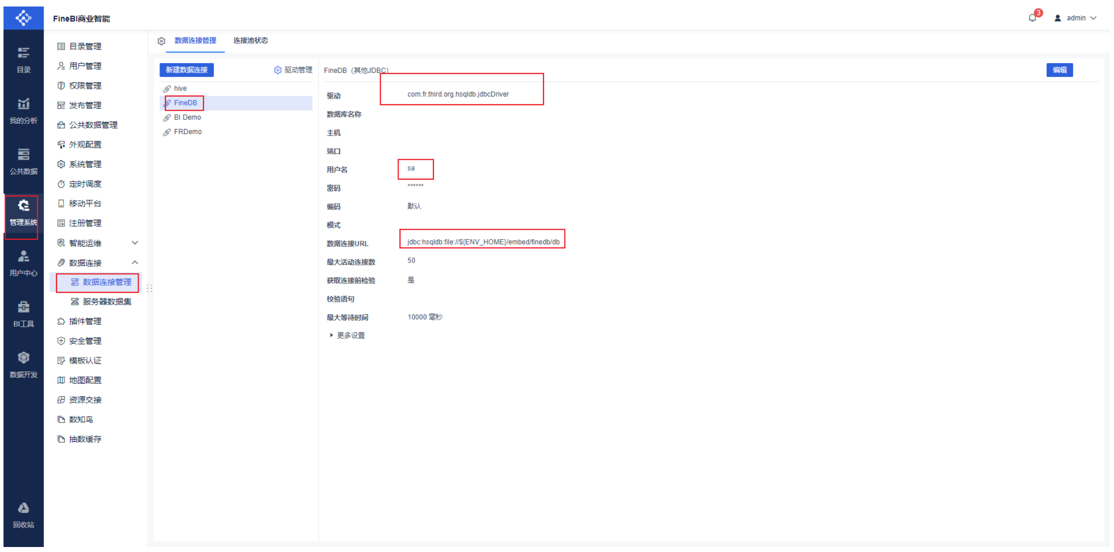


图 22：FinBI 软件成功部署

4、FinBI 部署完成后，进入系统，需要进行如下配置：图 23-26 将会展示

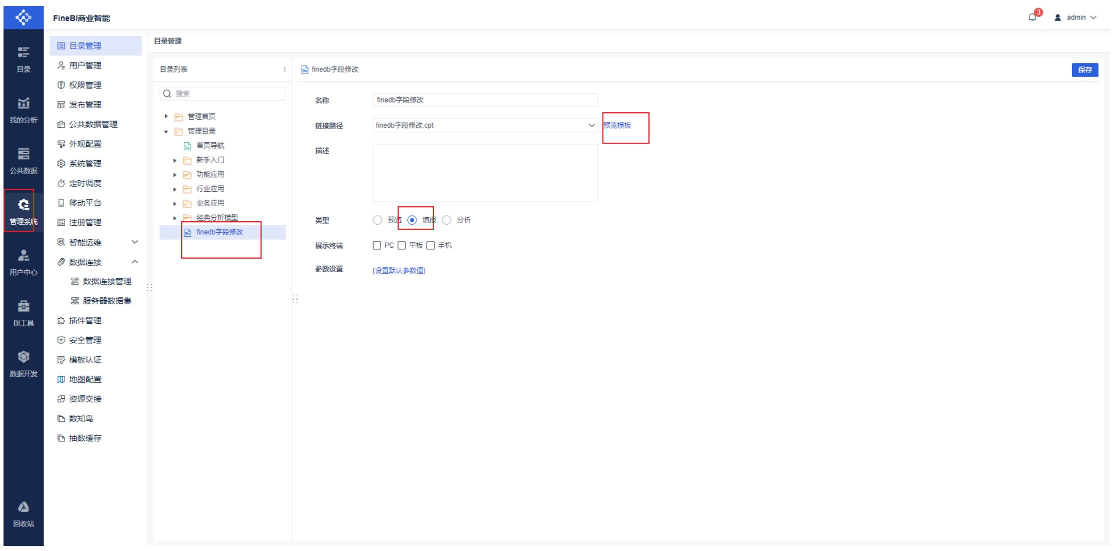


FineDB（其他JDBC）

驱动	com.fr.third.org.hsqldb.jdbcDriver
数据库名称	
主机	
端口	
用户名	sa
密码	******
编码	默认
模式	
数据连接URL	jdbc:hsqldb:file://${ENV_HOME}/embed/finedb/db
最大活动连接数	50
获取连接前检验	是
校验语句	
最大等待时间	10000 毫秒

▸ 更多设置

图 23：FinDB 链接



finedb字段修改

名称 finedb字段修改

链接路径 finedb字段修改.cpt 预览模板

描述

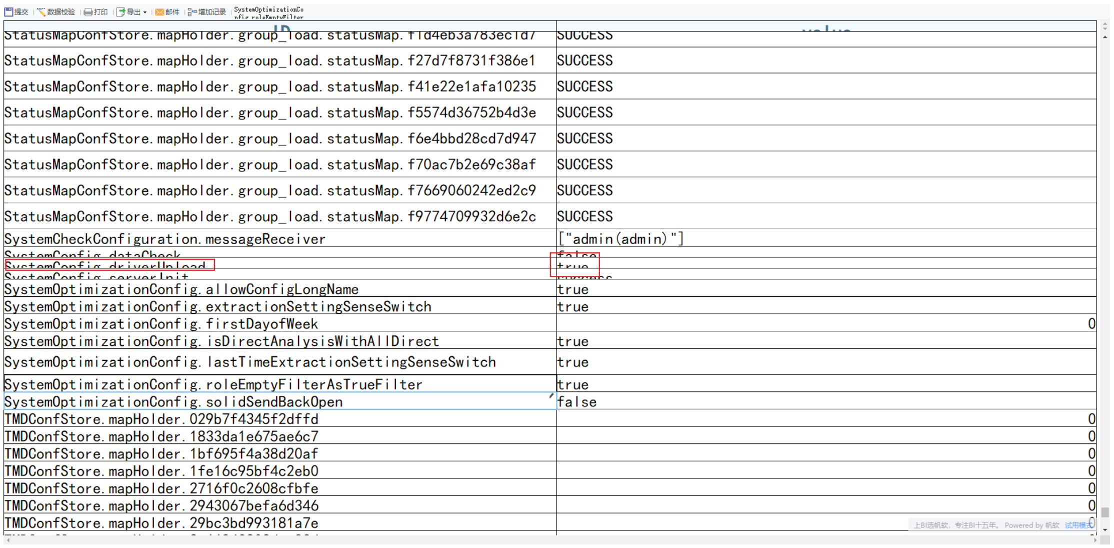
类型 预览 [✓ 填报] 分析

展示终端 ☐ PC ☐ 平板 ☐ 手机

参数设置 [设置默认参数值]

图 24：修改字段

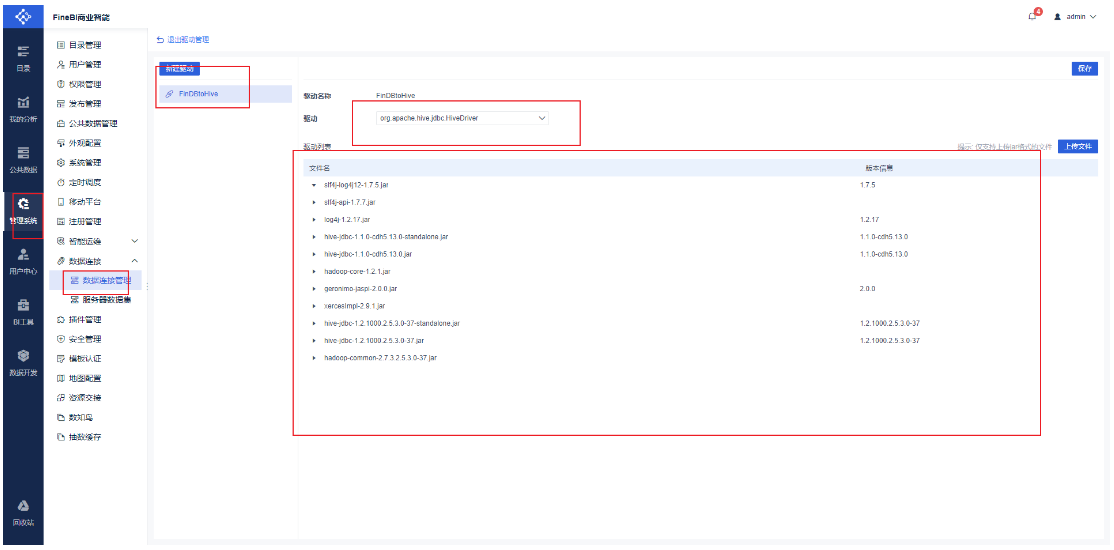
添加 SystemConfig.driverUpload 字段，并将值设置为 true



ID	value
StatusMapConfStore.mapHolder.group_load.statusMap.f1d4eb3a783ec1d7	SUCCESS
StatusMapConfStore.mapHolder.group_load.statusMap.f27d7f8731f386e1	SUCCESS
StatusMapConfStore.mapHolder.group_load.statusMap.f41e22e1afa10235	SUCCESS
StatusMapConfStore.mapHolder.group_load.statusMap.f5574d36752b4d3e	SUCCESS
StatusMapConfStore.mapHolder.group_load.statusMap.f6e4bbd28cd7d947	SUCCESS
StatusMapConfStore.mapHolder.group_load.statusMap.f70ac7b2e69c38af	SUCCESS
StatusMapConfStore.mapHolder.group_load.statusMap.f7669060242ed2c9	SUCCESS
StatusMapConfStore.mapHolder.group_load.statusMap.f9774709932d6e2c	SUCCESS
SystemCheckConfiguration.messageReceiver	["admin(admin)"]
SystemConfig.dataCheck	false
SystemConfig.driverUpload	true
SystemConfig.serverInit	[illegible]
SystemOptimizationConfig.allowConfigLongName	true
SystemOptimizationConfig.extractionSettingSenseSwitch	true
SystemOptimizationConfig.firstDayofWeek	0
SystemOptimizationConfig.isDirectAnalysisWithAllDirect	true
SystemOptimizationConfig.lastTimeExtractionSettingSenseSwitch	true
SystemOptimizationConfig.roleEmptyFilterAsTrueFilter	true
SystemOptimizationConfig.solidSendBackOpen	false
TMDConfStore.mapHolder.029b7f4345f2dffd	0
TMDConfStore.mapHolder.1833da1e675ae6c7	0
TMDConfStore.mapHolder.1bf695f4a38d20af	0
TMDConfStore.mapHolder.1fe16c95bf4c2eb0	0
TMDConfStore.mapHolder.2716f0c2608cfbfe	0
TMDConfStore.mapHolder.2943067befa6d346	0
TMDConfStore.mapHolder.29bc3bd993181a7e	0

上BI选帆软，专注BI十五年。Powered by 帆软 试用模式

图 25：修改 SystemConfig.driverUpload 字段



驱动名称 FinDBtoHive

驱动 org.apache.hive.jdbc.HiveDriver

驱动列表

提示: 仅支持上传jar格式的文件

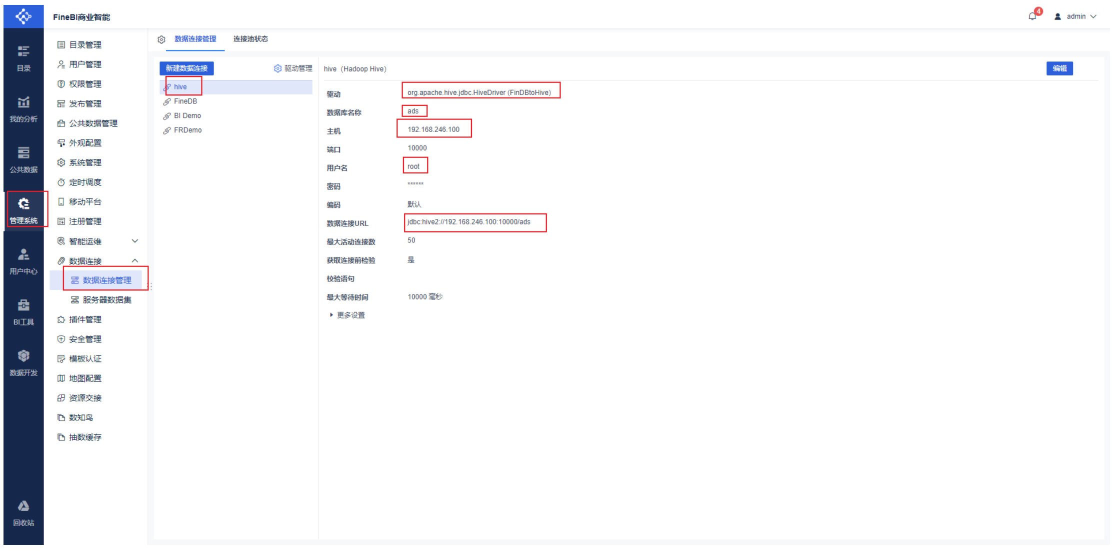
文件名	版本信息
slf4j-log4j12-1.7.5.jar	1.7.5
slf4j-api-1.7.7.jar	
log4j-1.2.17.jar	1.2.17
hive-jdbc-1.1.0-cdh5.13.0-standalone.jar	1.1.0-cdh5.13.0
hive-jdbc-1.1.0-cdh5.13.0.jar	1.1.0-cdh5.13.0
hadoop-core-1.2.1.jar	
geronimo-jaspi-2.0.0.jar	2.0.0
xercesImpl-2.9.1.jar	
hive-jdbc-1.2.1000.2.5.3.0-37-standalone.jar	1.2.1000.2.5.3.0-37
hive-jdbc-1.2.1000.2.5.3.0-37.jar	1.2.1000.2.5.3.0-37
hadoop-common-2.7.3.2.5.3.0-37.jar	

图 26：新建驱动管理，上传相关的 jar 包，

5、链接 hive 数据库

首先需要在 Hive 数据库里启动元数据服务：

```
hive --service metastore
hive --service hiveserver2
```



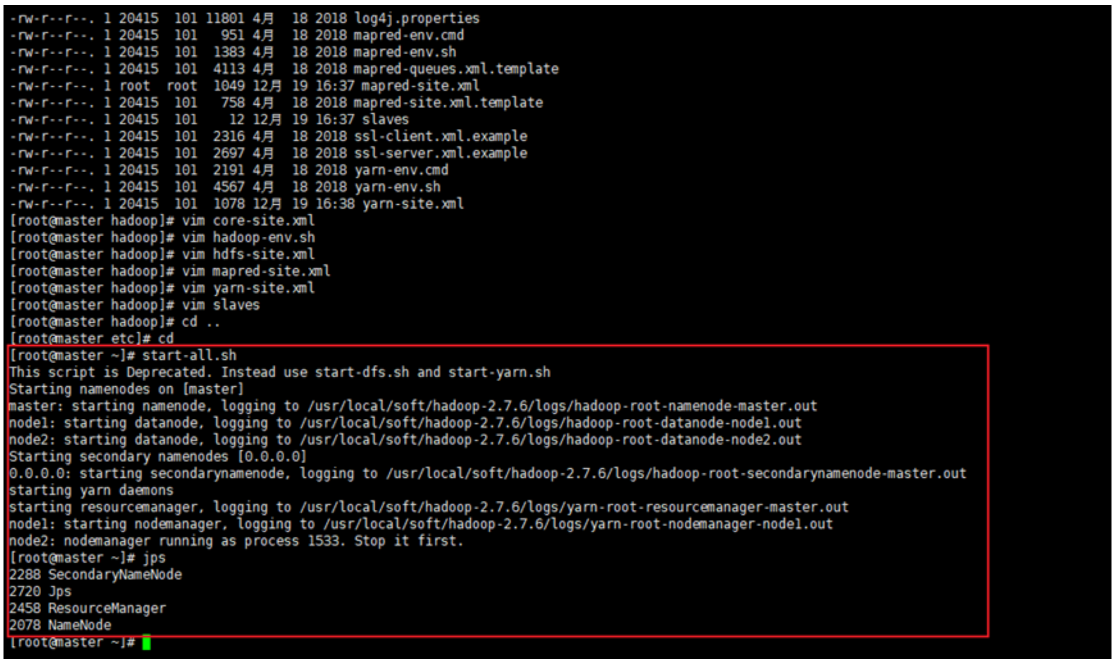
hive（Hadoop Hive）

驱动	org.apache.hive.jdbc.HiveDriver (FinDBtoHive)
数据库名称	ads
主机	192.168.246.100
端口	10000
用户名	root
密码	******
编码	默认
数据连接URL	jdbc:hive2://192.168.246.100:10000/ads
最大活动连接数	50
获取连接前检验	是
校验语句	
最大等待时间	10000 毫秒

图 27：链接 ADS 数据库

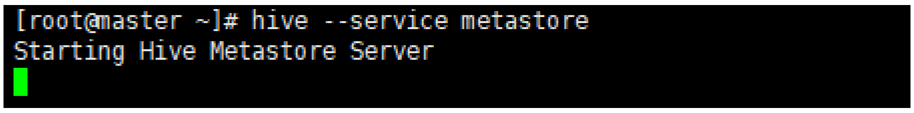
2.2 系统部署启动

2.2.2 Hadoop 集群的启动

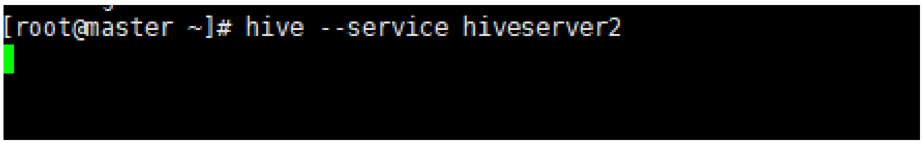


```
-rw-r--r--. 1 20415  101 11801 4月  18 2018 log4j.properties
-rw-r--r--. 1 20415  101   951 4月  18 2018 mapred-env.cmd
-rw-r--r--. 1 20415  101  1383 4月  18 2018 mapred-env.sh
-rw-r--r--. 1 20415  101  4113 4月  18 2018 mapred-queues.xml.template
-rw-r--r--. 1 root  root  1049 12月 19 16:37 mapred-site.xml
-rw-r--r--. 1 20415  101   758 4月  18 2018 mapred-site.xml.template
-rw-r--r--. 1 20415  101    12 12月 19 16:37 slaves
-rw-r--r--. 1 20415  101  2316 4月  18 2018 ssl-client.xml.example
-rw-r--r--. 1 20415  101  2697 4月  18 2018 ssl-server.xml.example
-rw-r--r--. 1 20415  101  2191 4月  18 2018 yarn-env.cmd
-rw-r--r--. 1 20415  101  4567 4月  18 2018 yarn-env.sh
-rw-r--r--. 1 20415  101  1078 12月 19 16:38 yarn-site.xml
[root@master hadoop]# vim core-site.xml
[root@master hadoop]# vim hadoop-env.sh
[root@master hadoop]# vim hdfs-site.xml
[root@master hadoop]# vim mapred-site.xml
[root@master hadoop]# vim yarn-site.xml
[root@master hadoop]# vim slaves
[root@master hadoop]# cd ..
[root@master etc]# cd
[root@master ~]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-namenode-master.out
node1: starting datanode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-datanode-node1.out
node2: starting datanode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-datanode-node2.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/soft/hadoop-2.7.6/logs/hadoop-root-secondarynamenode-master.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/soft/hadoop-2.7.6/logs/yarn-root-resourcemanager-master.out
node1: starting nodemanager, logging to /usr/local/soft/hadoop-2.7.6/logs/yarn-root-nodemanager-node1.out
node2: nodemanager running as process 1533. Stop it first.
[root@master ~]# jps
2288 SecondaryNameNode
2720 Jps
2458 ResourceManager
2078 NameNode
[root@master ~]#
```

2.2.3 Hive 数据库以及元数据服务的启动

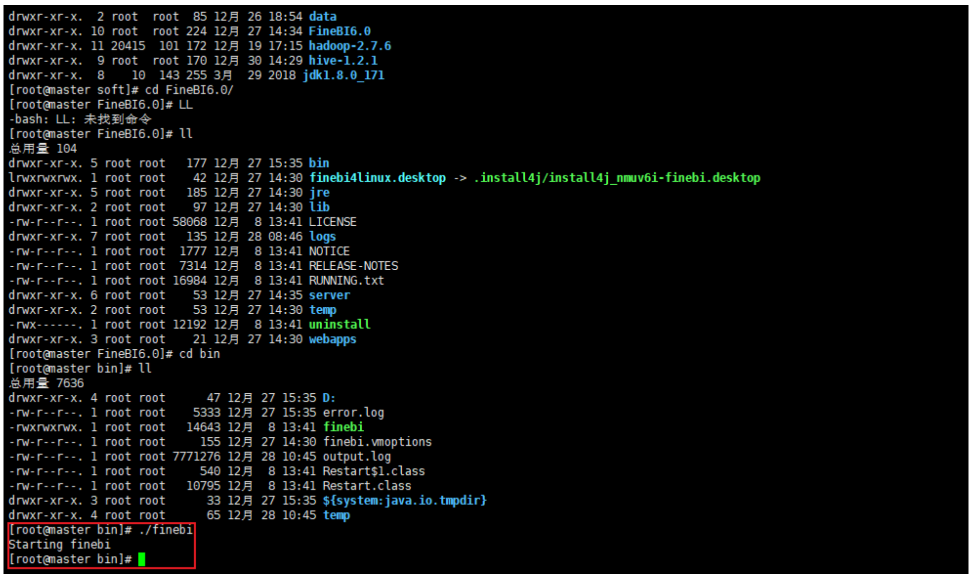


```
[root@master ~]# hive --service metastore
Starting Hive Metastore Server
```



```
[root@master ~]# hive --service hiveserver2
```

2.2.4 FinBI 的启动



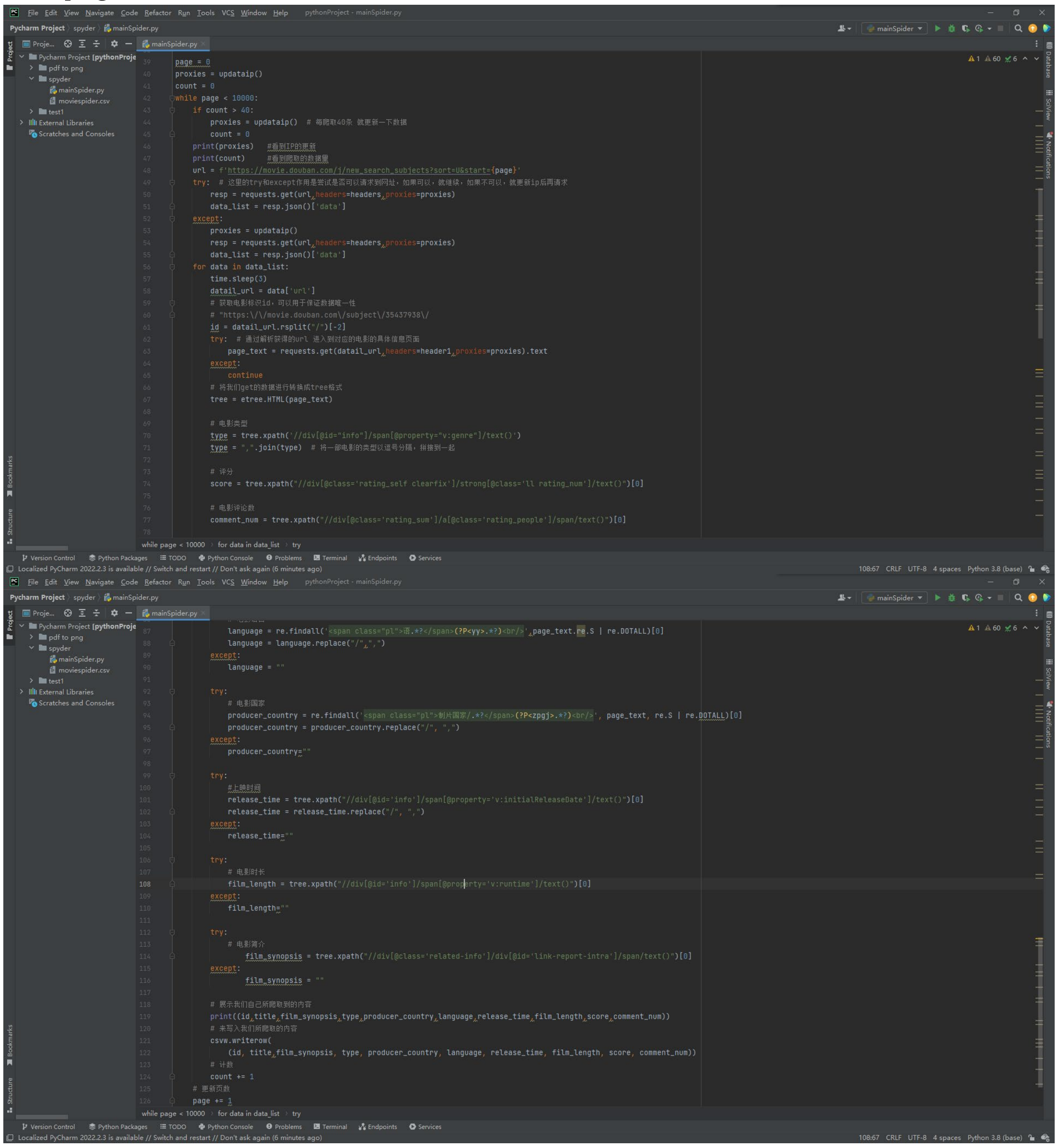
```
drwxr-xr-x.  2 root  root  85 12月 26 18:54 data
drwxr-xr-x. 10 root  root 224 12月 27 14:34 FineBI6.0
drwxr-xr-x. 11 20415  101 172 12月 19 17:15 hadoop-2.7.6
drwxr-xr-x.  9 root  root 170 12月 30 14:29 hive-1.2.1
drwxr-xr-x.  8    10  143 255 3月  29 2018 jdk1.8.0_171
[root@master soft]# cd FineBI6.0/
[root@master FineBI6.0]# LL
-bash: LL: 未找到命令
[root@master FineBI6.0]# ll
总用量 104
drwxr-xr-x. 5 root root   177 12月 27 15:35 bin
lrwxrwxrwx. 1 root root    42 12月 27 14:30 finebi4linux.desktop -> .install4j/install4j_nmuv6i-finebi.desktop
drwxr-xr-x. 5 root root   185 12月 27 14:30 jre
drwxr-xr-x. 2 root root    97 12月 27 14:30 lib
-rw-r--r--. 1 root root 58068 12月  8 13:41 LICENSE
drwxr-xr-x. 7 root root   135 12月 28 08:46 logs
-rw-r--r--. 1 root root  1777 12月  8 13:41 NOTICE
-rw-r--r--. 1 root root  7314 12月  8 13:41 RELEASE-NOTES
-rw-r--r--. 1 root root 16984 12月  8 13:41 RUNNING.txt
drwxr-xr-x. 6 root root    53 12月 27 14:35 server
drwxr-xr-x. 2 root root    53 12月 27 14:30 temp
-rwx------. 1 root root 12192 12月  8 13:41 uninstall
drwxr-xr-x. 3 root root    21 12月 27 14:30 webapps
[root@master FineBI6.0]# cd bin
[root@master bin]# ll
总用量 7636
drwxr-xr-x. 4 root root      47 12月 27 15:35 D:
-rw-r--r--. 1 root root    5333 12月 27 15:35 error.log
-rwxrwxrwx. 1 root root   14643 12月  8 13:41 finebi
-rw-r--r--. 1 root root     155 12月 27 14:30 finebi.vmoptions
-rw-r--r--. 1 root root 7771276 12月 28 10:45 output.log
-rw-r--r--. 1 root root     540 12月  8 13:41 Restart$1.class
-rw-r--r--. 1 root root   10795 12月  8 13:41 Restart.class
drwxr-xr-x. 3 root root      33 12月 27 15:35 ${system:java.io.tmpdir}
drwxr-xr-x. 4 root root      65 12月 28 10:45 temp
[root@master bin]# ./finebi
Starting finebi
[root@master bin]#
```

三、数据爬取

本项目爬取的数据是豆瓣电影网的数据，分别爬取了，电影的 id，类型，国家，评分，评论数，名称，年份，语言，上映时间，电影时长，电影简介。

实现代码如下：

```
import time
import requests
from lxml import etree
import re
import csv
from fake_useragent import UserAgent

ua = UserAgent()

fp = open('./moviespider.csv','a+',encoding='utf-8',newline="")
csvw = csv.writer(fp)

headers = {
    'User-Agent':ua.random
}
header1 = {
    'Host':'movie.douban.com',
    'Accept':'text/html, application/xhtml+xml, application/xml; q=0.9,
*/*; q=0.8',
    'Accept-Language':'zh-CN, zh; q=0.9',
    'X-Requested-with':'XMLHttpRequest',
    'Connection':'Keep-Alive',
    'User-Agent': ua.random
}

def updataip():
    inpurl =
'http://www.damaiip.cn/index.php?s=/front/user/getIPlist&xsn=d4f6e449
b0c71e52ad1017c186f023e3&osn=TC_NO167203722553612283&tiqu=1'
    ip = requests.get(inpurl).text  # 更新 IP 地址
    proxies = {'https':ip}
    return proxies   # 返回 ip 地址

page = 0
proxies = updataip()
count = 0
while page < 10000:
    if count > 40:
        proxies = updataip()  # 每爬取 40 条 就更新一下数据
        count = 0
    print(proxies)   #看到 IP 的更新
    print(count)     #看到爬取的数据量
    url =
f'https://movie.douban.com/j/new_search_subjects?sort=U&start={page}'
    try:  # 这里的 try 和 except 作用是尝试是否可以请求到网址,如果可以,
就继续，如果不可以，就更新 ip 后再请求
        resp = requests.get(url,headers=headers,proxies=proxies)
        data_list = resp.json()['data']
    except:
        proxies = updataip()
        resp = requests.get(url,headers=headers,proxies=proxies)
        data_list = resp.json()['data']
    for data in data_list:
        time.sleep(3)
        datail_url = data['url']
        # 获取电影标识 id，可以用于保证数据唯一性
        # "https:\/\/movie.douban.com\/subject\/35437938\/
        id = datail_url.rsplit("/")[-2]
        try:  # 通过解析获得的 url 进入到对应的电影的具体信息页面
            page_text =
requests.get(datail_url,headers=header1,proxies=proxies).text
        except:
            continue
        # 将我们 get 的数据进行转换成 tree 格式
        tree = etree.HTML(page_text)

        # 电影类型
        type =
tree.xpath('//div[@id="info"]/span[@property="v:genre"]/text()')
        type = ",".join(type)  # 将一部电影的类型以逗号分隔，拼接到一
起

        # 评分
        score = tree.xpath("//div[@class='rating_self
clearfix']/strong[@class='ll rating_num']/text()")[0]

        # 电影评论数
        comment_num =
tree.xpath("//div[@class='rating_sum']/a[@class='rating_people']/span
/text()")[0]

        # 电影名称
        title =
tree.xpath("//div[@id='content']/h1/span[@property='v:itemreviewed']/
text()")[0]

        # 电影年份
        year =
tree.xpath("//div[@id='content']/h1/span[@class='year']/text()")[0]

        try:
            # 电影语言
            language = re.findall('<span class="pl">
语.*?</span>(?P<yy>.*?)<br/>',page_text.re.S | re.DOTALL)[0]
            language = language.replace("/",",")
        except:
            language = ""

        try:
            # 电影国家
            producer_country = re.findall('<span class="pl">制片国家
/.*?</span>(?P<zpgj>.*?)<br/>', page_text, re.S | re.DOTALL)[0]
            producer_country = producer_country.replace("/", ",")
        except:
            producer_country=""

        try:
            #上映时间
            release_time =
tree.xpath("//div[@id='info']/span[@property='v:initialReleaseDate']/
text()")[0]
            release_time = release_time.replace("/", ",")
        except:
            release_time=""

        try:
            # 电影时长
            film_length =
tree.xpath("//div[@id='info']/span[@property='v:runtime']/text()")[0]
        except:
            film_length=""

        try:
            # 电影简介
                film_synopsis =
tree.xpath("//div[@class='related-info']/div[@id='link-report-intra']
/span/text()")[0]
        except:
                film_synopsis = ""
        # 展示我们自己所爬取到的内容
print((id,title,film_synopsis,type,producer_country,language,release_
time,film_length,score,comment_num))
        # 来写入我们所爬取的内容
        csvw.writerow(
            (id, title,film_synopsis, type, producer_country, language,
release_time, film_length, score, comment_num))
        # 计数
        count += 1
    # 更新页数
    page += 1
```



```
page = 0
proxies = updataip()
count = 0
while page < 10000:
    if count > 40:
        proxies = updataip()  # 每爬取40条 就更新一下数据
        count = 0
    print(proxies)   #看到IP的更新
    print(count)     #看到爬取的数据量
    url = f'https://movie.douban.com/j/new_search_subjects?sort=U&start={page}'
    try:  # 这里的try和except作用是尝试是否可以请求到网址，如果可以，就继续，如果不可以，就更新ip后再请求
        resp = requests.get(url,headers=headers,proxies=proxies)
        data_list = resp.json()['data']
    except:
        proxies = updataip()
        resp = requests.get(url,headers=headers,proxies=proxies)
        data_list = resp.json()['data']
    for data in data_list:
        time.sleep(3)
        datail_url = data['url']
        # 获取电影标识id，可以用于保证数据唯一性
        # "https:\/\/movie.douban.com\/subject\/35437938\/
        id = datail_url.rsplit("/")[-2]
        try:  # 通过解析获得的url 进入到对应的电影的具体信息页面
            page_text = requests.get(datail_url,headers=header1,proxies=proxies).text
        except:
            continue
        # 将我们get的数据进行转换成tree格式
        tree = etree.HTML(page_text)

        # 电影类型
        type = tree.xpath('//div[@id="info"]/span[@property="v:genre"]/text()')
        type = ",".join(type)  # 将一部电影的类型以逗号分隔，拼接到一起

        # 评分
        score = tree.xpath("//div[@class='rating_self clearfix']/strong[@class='ll rating_num']/text()")[0]

        # 电影评论数
        comment_num = tree.xpath("//div[@class='rating_sum']/a[@class='rating_people']/span/text()")[0]

```

```
            language = re.findall('<span class="pl">语.*?</span>(?P<yy>.*?)<br/>',page_text.re.S | re.DOTALL)[0]
            language = language.replace("/",",")
        except:
            language = ""

        try:
            # 电影国家
            producer_country = re.findall('<span class="pl">制片国家/.*?</span>(?P<zpgj>.*?)<br/>', page_text, re.S | re.DOTALL)[0]
            producer_country = producer_country.replace("/", ",")
        except:
            producer_country=""

        try:
            #上映时间
            release_time = tree.xpath("//div[@id='info']/span[@property='v:initialReleaseDate']/text()")[0]
            release_time = release_time.replace("/", ",")
        except:
            release_time=""

        try:
            # 电影时长
            film_length = tree.xpath("//div[@id='info']/span[@property='v:runtime']/text()")[0]
        except:
            film_length=""

        try:
            # 电影简介
                film_synopsis = tree.xpath("//div[@class='related-info']/div[@id='link-report-intra']/span/text()")[0]
        except:
                film_synopsis = ""

        # 展示我们自己所爬取到的内容
        print((id,title,film_synopsis,type,producer_country,language,release_time,film_length,score,comment_num))
        # 来写入我们所爬取的内容
        csvw.writerow(
            (id, title,film_synopsis, type, producer_country, language, release_time, film_length, score, comment_num))
        # 计数
        count += 1
    # 更新页数
    page += 1
```

图 28：PyCharm 代码截屏

四、数据分析

4.1 导入数据到 Hive 数据库的 ODS 层

```
create table ODS_DY(
    id bigint,
    title string,
    director string,
    movie_type string,
    producer_country string,
    language string,
    release_time string,
    film_length string,
    score double,
    comment_num string #这个字段建议用 bigint
)row format delimited fields terminated by ','
tblproperties("skip.header.line.count"="1");
#导入数据
load data local inpath '/usr/local/soft/data/moviesData2.txt' into table
ods_dy;
```

4.2 对原始数据进行处理，并放到 DWD 层 ，去除脏数据

```
create table dwd.dwd_dy like ods.ods_dy;
insert overwrite table dwd.dwd_dy
select id,
title,
director,
split(movie_type,"|")[0] as movie_type,
split(producer_country,"|")[0] as producer_country,
split(language,"|")[0] as language,
substring(release_time,1,4) as release_time,
film_length,
score,
comment_num
from ods.ods_dy where release_time is not null;
```

4.3 根据 DWD 表的格式，创建 DWS 层的表

```
create table dws.dws_dy like dwd.dwd_dy;
insert overwrite table dws.dws_dy
select * from dwd.dwd_dy;
```

4.4 对 DWS 层的数据进行指标分析，并把结果保存到 ADS 层

4.4.1 电影评分的均值（按照国家）

```
create table ads.xuwnag_country_avgscore(
producer_country string,
avg_score double
)row format delimited fields terminated by ',';
insert overwrite table ads.xuwnag_country_avgscore
select producer_country,round(avg(score),3) as country_avg_num
from dws.dws_dy
group by producer_country;
```

4.4.2 按照烂片率由高到低排序，烂片定义：评分低于 7.5 分；

好片率 = 这个类别评分大于 7.5 的电影总数量/这个类别的电影总数量

```
create table ads.xuwang_goodmovie(
movie_type string,
type_sc double
)row format delimited fields terminated by ',';
insert overwrite table ads.xuwang_goodmovie
select a.movie_type,round((a.count_num/b.count_num),4) as compare
from (
select m.movie_type,count(1) as count_num
from dws.dws_dy m
where m.score >= 7.5 and m.score < 10.0
group by m.movie_type
) a left join (
select w.movie_type,count(1) as count_num
from dws.dws_dy w
group by w.movie_type
) b on a.movie_type = b.movie_type
order by compare;
```

4.4.3 评价人数最多的前 50 部电影

```
create table ads.xuwang_comment_num(
title string,
comment_num bigint
)row format delimited fields terminated by ',';
insert overwrite table ads.xuwang_comment_num
select title,comment_num
from dws.dws_dy
order by comment_num desc
limit 50;
```

4.5 数据可视化

通过上面的数据处理，以及指标分析后，我们将得到的数据存储到了 ADS 层，现在利用 FinBI 链接 Hive 数据库的 ADS 层。

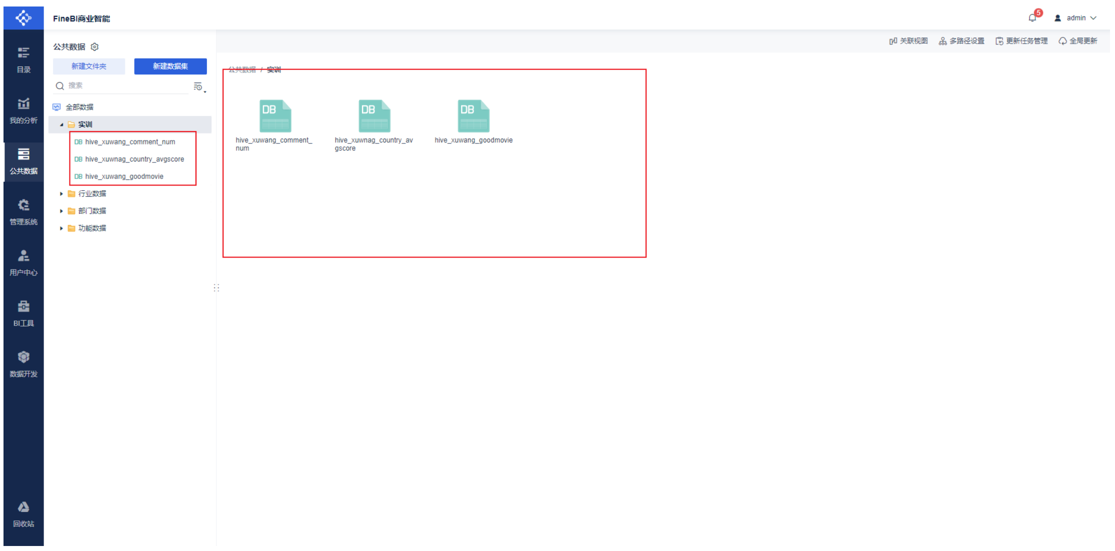


图 29：FinBI 链接 ADS 层获取数据并分析

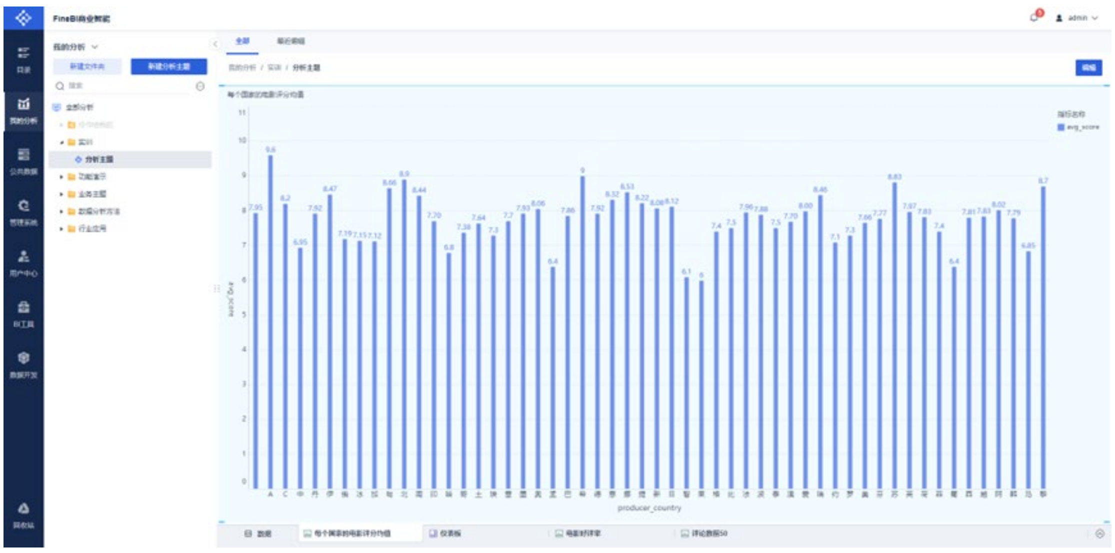


图 30：每个国家的电影评分均值

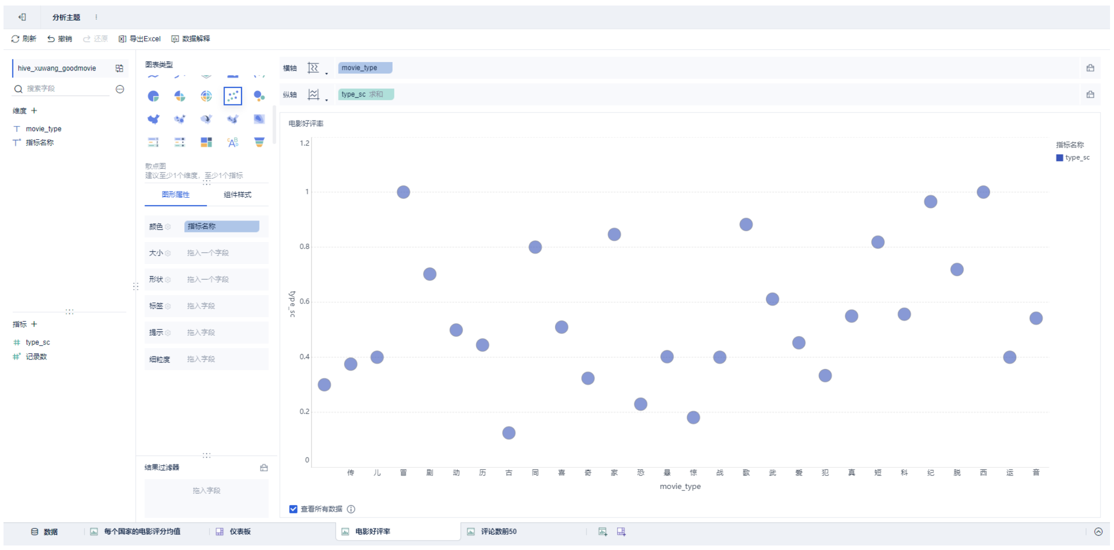
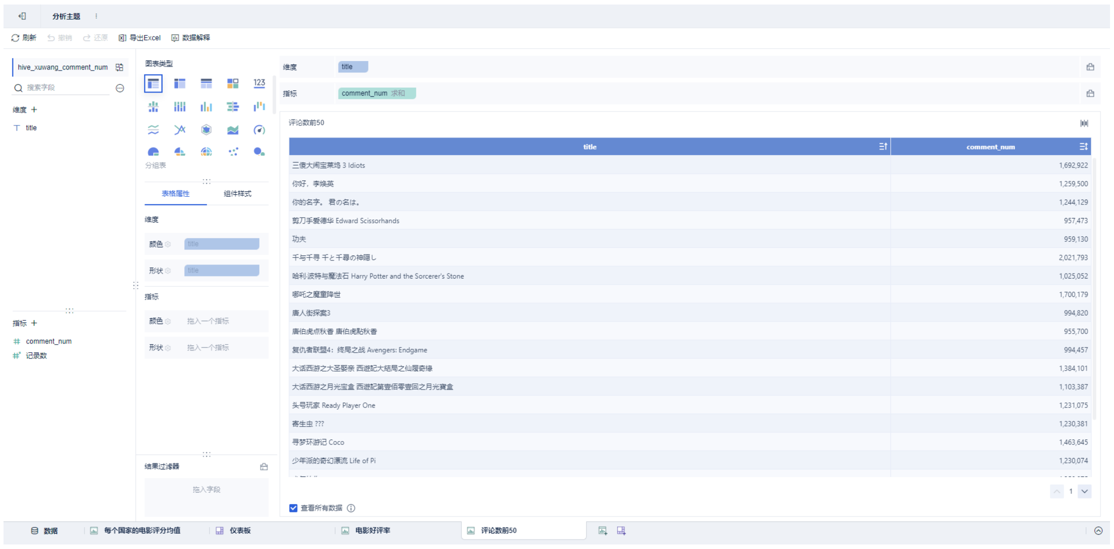
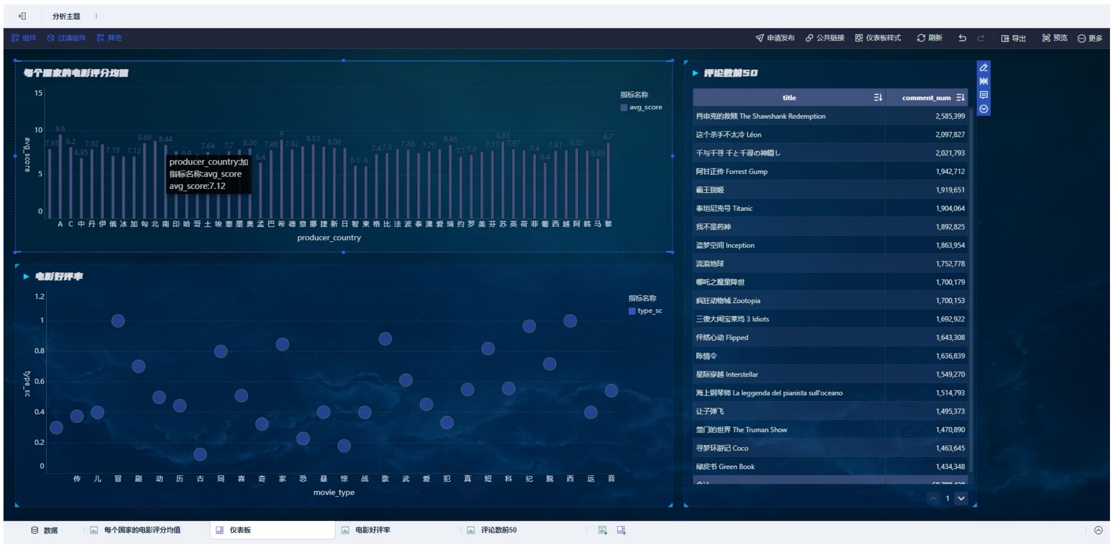


图 31：电影好评率



title	comment_num
三傻大闹宝莱坞 3 Idiots	1,692,922
你好，李焕英	1,259,500
你的名字。 君の名は。	1,244,129
剪刀手爱德华 Edward Scissorhands	957,473
功夫	959,130
千与千寻 千と千尋の神隠し	2,021,793
哈利·波特与魔法石 Harry Potter and the Sorcerer's Stone	1,025,052
哪吒之魔童降世	1,700,179
唐人街探案3	994,820
唐伯虎点秋香 唐伯虎點秋香	955,700
复仇者联盟4：终局之战 Avengers: Endgame	994,457
大话西游之大圣娶亲 西遊記大結局之仙履奇緣	1,384,101
大话西游之月光宝盒 西遊記第壹佰零壹回之月光寶盒	1,103,387
头号玩家 Ready Player One	1,231,075
寄生虫 ???	1,230,381
寻梦环游记 Coco	1,463,645
少年派的奇幻漂流 Life of Pi	1,230,074

图 32：评论数 TOP50



title	comment_num
肖申克的救赎 The Shawshank Redemption	2,585,399
这个杀手不太冷 Léon	2,097,827
千与千寻 千と千尋の神隠し	2,021,793
阿甘正传 Forrest Gump	1,942,712
霸王别姬	1,919,651
泰坦尼克号 Titanic	1,904,064
我不是药神	1,892,825
盗梦空间 Inception	1,863,954
流浪地球	1,752,778
哪吒之魔童降世	1,700,179
疯狂动物城 Zootopia	1,700,153
三傻大闹宝莱坞 3 Idiots	1,692,922
怦然心动 Flipped	1,643,308
陈情令	1,636,839
星际穿越 Interstellar	1,549,270
海上钢琴师 La leggenda del pianista sull'oceano	1,514,793
让子弹飞	1,495,373
楚门的世界 The Truman Show	1,470,890
寻梦环游记 Coco	1,463,645
绿皮书 Green Book	1,434,348

图 33：总仪表盘

五、项目源代码

5.1 爬虫源代码

```
import time
import requests
from lxml import etree
import re
import csv
from fake_useragent import UserAgent

ua = UserAgent()

fp = open('./moviespider.csv','a+',encoding='utf-8',newline="")
csvw = csv.writer(fp)

headers = {
    'User-Agent':ua.random
}
header1 = {
    'Host':'movie.douban.com',
    'Accept':'text/html, application/xhtml+xml, application/xml; q=0.9,
*/*; q=0.8',
    'Accept-Language':'zh-CN, zh; q=0.9',
    'X-Requested-with':'XMLHttpRequest',
    'Connection':'Keep-Alive',
    'User-Agent': ua.random
}

def updataip():
    inpurl                                                          =
'http://www.damaiip.cn/index.php?s=/front/user/getIPlist&xsn=d4f6e449
b0c71e52ad1017c186f023e3&osn=TC_NO16720372253612283&tiqu=1'
    ip = requests.get(inpurl).text  # 更新 IP 地址
    proxies = {'https':ip}
    return proxies   # 返回 ip 地址

page = 0
proxies = updataip()
count = 0
while page < 10000:
    if count > 40:
        proxies = updataip()  # 每爬取 40 条 就更新一下数据
        count = 0
    print(proxies)   #看到 IP 的更新
    print(count)     #看到爬取的数据量
    url =
f'https://movie.douban.com/j/new_search_subjects?sort=U&start={page}'
    try:  # 这里的 try 和 except 作用是尝试是否可以请求到网址,如果可以,
就继续，如果不可以，就更新 ip 后再请求
        resp = requests.get(url,headers=headers,proxies=proxies)
        data_list = resp.json()['data']
    except:
        proxies = updataip()
        resp = requests.get(url,headers=headers,proxies=proxies)
        data_list = resp.json()['data']
    for data in data_list:
        time.sleep(3)
        datail_url = data['url']
        # 获取电影标识 id，可以用于保证数据唯一性
        # "https:\/\/movie.douban.com\/subject\/35437938\/
        id = datail_url.rsplit("/")[-2]
        try:  # 通过解析获得的 url 进入到对应的电影的具体信息页面
            page_text =
requests.get(datail_url,headers=header1,proxies=proxies).text
        except:
            continue
        # 将我们 get 的数据进行转换成 tree 格式
        tree = etree.HTML(page_text)

        # 电影类型
        type =
tree.xpath('//div[@id="info"]/span[@property="v:genre"]/text()')
        type = ",".join(type)  # 将一部电影的类型以逗号分隔，拼接到一
起

        # 评分
        score = tree.xpath("//div[@class='rating_self
clearfix']/strong[@class='ll rating_num']/text()")[0]

        # 电影评论数
        comment_num =
tree.xpath("//div[@class='rating_sum']/a[@class='rating_people']/span
/text()")[0]

        # 电影名称
        title =
tree.xpath("//div[@id='content']/h1/span[@property='v:itemreviewed']/
text()")[0]

        # 电影年份
        year                                                                 =
tree.xpath("//div[@id='content']/h1/span[@class='year']/text()")[0]

        try:
            # 电影语言
            language       =       re.findall('<span       class="pl">
语.*?</span>(?P<yy>.*?)<br/>',page_text.re.S | re.DOTALL)[0]
            language = language.replace("/",",")
        except:
            language = ""

        try:
            # 电影国家
            producer_country = re.findall('<span class="pl">制片国家
/.*?</span>(?P<zpgj>.*?)<br/>', page_text, re.S | re.DOTALL)[0]
            producer_country = producer_country.replace("/", ",")
        except:
            producer_country=""

        try:
            #上映时间
            release_time                                                     =
tree.xpath("//div[@id='info']/span[@property='v:initialReleaseDate']/
text()")[0]
            release_time = release_time.replace("/", ",")
        except:
            release_time=""

        try:
            # 电影时长
            film_length                                                      =
tree.xpath("//div[@id='info']/span[@property='v:runtime']/text()")[0]
        except:
            film_length=""

        try:
            # 电影简介
                film_synopsis                                                =
tree.xpath("//div[@class='related-info']/div[@id='link-report-intra']
/span/text()")[0]
        except:
                film_synopsis = ""
        #   展  示  我  们  自  己  所  爬  取  到  的  内  容
print((id,title,film_synopsis,type,producer_country,language,release_
time,film_length,score,comment_num))
        # 来写入我们所爬取的内容
        csvw.writerow(
            (id, title,film_synopsis, type, producer_country, language,
release_time, film_length, score, comment_num))
        # 计数
        count += 1
    # 更新页数
page += 1
```

5.2 数据分析源代码

```
#电影评分的均值（按照国家）
create table ads.xuwnag_country_avgscore(
producer_country string,
avg_score double
)row format delimited fields terminated by ',';
insert overwrite table ads.xuwnag_country_avgscore
select producer_country,round(avg(score),3) as country_avg_num
from dws.dws_dy
group by producer_country;

#按照烂片率由高到低排序，烂片定义：评分低于 7.5 分；
#好片率 = 这个类别评分大于 7.5 的电影总数量/这个类别的电影总数量
create table ads.xuwang_goodmovie(
movie_type string,
type_sc double
)row format delimited fields terminated by ',';
insert overwrite table ads.xuwang_goodmovie
select a.movie_type,round((a.count_num/b.count_num),4) as compare
from (
select m.movie_type,count(1) as count_num
from dws.dws_dy m
where m.score >= 7.5 and m.score < 10.0
group by m.movie_type
) a left join (
select w.movie_type,count(1) as count_num
from dws.dws_dy w
group by w.movie_type
) b on a.movie_type = b.movie_type
order by compare;

#评价人数最多的前 50 部电影
create table ads.xuwang_comment_num(
title string,
comment_num bigint
)row format delimited fields terminated by ',';
insert overwrite table ads.xuwang_comment_num
select title,comment_num
from dws.dws_dy
order by comment_num desc
limit 50;
```