## 二、实验项目内容

1 Spark 单机版搭建

1.1 准备工作 ：

1. 配置用户
2. 配置 SSH
3. 配置 yum 源
4. 配置 Java 环境
5. 安装 python

1.2 Hadoop 安装

1.3 Spark 安装

1.4 测试

2 Hadoop+Spark 分布式环境搭建

2.1 准备工作：

1. 修改主机名
2. 修改 host
3. SSH 互相免密

2.2 Hadoop 集群配置：

1. master 节点配置
2. slave 节点配置
3. 集群启动测试

2.3 Spark 集群配置：

1. Spark 配置
2. 启动 Spark 集群

## 三、实验过程或算法（源程序）

## 1. spark 单机版环境配置

**首先记录小组成员各自服务器的内网 IP&公网 IP：**

| 主机名 | 内网 IP | 外网 IP |
|---|---|---|
| master | 192.168.0.251 | 1.94.33.146 |
| slave01 | 192.168.0.130 | 1.94.11.79 |
| slave02 | 192.168.0.204 | 1.94.37.155 |
| slave04 | 192.168.0.200 | 60.204.193.83 |

注：slave03 在中途配置过程中被挂掉了，所以 slave04 替换 slave03.

### 1.1 准备工作

1.1.1 配置用户

按照实验指导书的步骤创建 hadoop 用户，并设置密码。并使用 vim 设置如下权限，使得 hadoop 能够拥有等同 root 的权限。（Hadoop 的分布式文件系统中，需要给 Hadoop 一个与 root 相同的权限（all），这是为了确保 Hadoop 能够执行必要的文件系统操作，比如读取、写入和执行等。Hadoop 是一个分布式系统，它将文件存储在多个节点上，这些节点可能由不同的用户或组拥有。为了让 Hadoop 可以正常操作这些文件，它需要拥有足够的权限来管理这些文件系统。如果没有这些权限，Hadoop 就无法进行必要的文件操作，如存储数据、创建目录、删除文件等。

```
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
hadoop  ALL=(ALL)       ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
```

切换到 hadoop 用户下：

```
[root@slave01 ~]# su hadoop
[hadoop@slave01 root]$
```

此时可以成功使用 su 方法切换到 hadoop 用户下，hadoop 用户配置成功。

1.1.2 配置 SSH

因为集群、单节点模式都需要用到 ssh 登陆。同时每次登陆 ssh 都要输入密码是件蛮麻烦的事 ，我们可以通过生成公钥配置来免密码登陆。

生成密匙指令：

```
[hadoop@master root]$ ssh localhost
```

授权指令：

```
[hadoop@master .ssh]$ cat id_rsa.pub >> authorized_keys
```

修改权限指令：

```
[hadoop@master .ssh]$ chmod 600 ./authorized_keys
```

上述操作完成后则成功使用公匙配置来进行 SSH 登录，不需要再输入密码，如下截图为成功配置后的测试截图：

```
[hadoop@master .ssh]$ ssh localhost   # ssh登陆
Last login: Sun Oct 22 21:19:25 2023 from 127.0.0.1

        Welcome to Huawei Cloud Service
```

可以看到无需输入密码成功登录。

### 1.1.3 配置 yum 源

本实验按照指导书使用阿里源进行下载。指令截图分别如下：

切换 yum 仓库

```
[hadoop@master /]$ cd /etc/yum.repos.d/
```

备份下载原 repo 文件

```
[hadoop@master /]$ sudo mv CentOS-Base.repo CentOS-Base.repo.backup
```

下载 repo 文件

```
[hadoop@master /]$ sudo wget -O /etc/yum.repos.d/CentOS-7.repo http://mirrors.aliyun.com/repo/Centos-7.repo
```

将其设置为默认 repo 文件

```
[hadoop@master /]$ sudo mv  CentOS-7.repo CentOS-Base.repo
```

生成缓存

```
[hadoop@slave01 ~]$ yum clean all
Loaded plugins: fastestmirror
Cleaning repos: base epel extras updates
Cleaning up list of fastest mirrors
Other repos take up 229 M of disk space (use --verbose for details)
[hadoop@slave01 ~]$ yum makecache
```

### 1.1.4 配置 java 服务器

安装 JDK：

```
[hadoop@master /]$ sudo yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel
```

配置环境变量，使用 vim 编辑器配置：

```
[hadoop@master ~]$ vim ~/.bashrc
```

在 bashrc 文件中加入单独一行路径，指向 jdk 安装位置：

```
# User specific aliases and functions
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
```

再执行 source ~/.bashrc 生效环境变量即可。

测试：配置成功可以输出 java 版本：

```
[hadoop@master ~]$ echo $JAVA_HOME     # 检验变量值
/usr/lib/jvm/java-1.8.0-openjdk
[hadoop@master ~]$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
```

1.1.5 安装 python

使用 yum 拉去 python3 安装包：

```
[hadoop@master ~]$ yum list python3
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
Installed Packages
python3.x86_64                                          3.6.8-19.el7_9
Available Packages
python3.i686                                            3.6.8-19.el7_9
```

执行如下指令进行安装，输入 hadoop 密码执行管理员安装操作即可：

```
[hadoop@master ~]$ sudo yum install python3.x86_64
```

完成后查看 python：

```
[hadoop@master ~]$ python
Python 2.7.5 (default, Jun 20 2023, 11:36:40)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

**1.2 hadoop 安装**（本实验在指导书上描述为只需要一台 master 服务器上执行，但是实际我们在此遇到了问题，[将在后续的遇到问题并解决中详细描述] 并发现作为 slave 的主机仍然需要执行该操作）

使用北理工镜像网站进行安装

（ps:原来指导书上的网址已经失效，新的网址及命令行如下：

sudo wget -O hadoop-2.10.1.tar.gz http://mirrors.bit.edu.cn/apache/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz --no-check-certificate 使用--no-check-certificate 可以避开验证直接下载）

```
[hadoop@master ~]$ sudo wget -O hadoop-2.10.1.tar.gz http://mirrors.bit.edu.cn/apache/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz  --no-check-certificate
```

然后根据指导书的步骤解压，修改文件：

sudo tar -zxf hadoop-2.8.5.tar.gz -C /usr/local

sudo tar -zxf hadoop-2.10.1.tar.gz -C /usr/local

cd /usr/local/　　#  切换到解压目录下

sudo mv ./hadoop-2.8.5/ ./hadoop

sudo chown -R hadoop:hadoop ./hadoop　#  修改文件权限

最后进行测试：

```
[hadoop@master ~]$ cd /usr/local/hadoop      # 切换到hadoop目录下
[hadoop@master hadoop]$ ./bin/hadoop version      # 输出hadoop版本号
Hadoop 2.10.1
```

可以看到成功转入了 hadoop 的文件下，并输出了版本号 2.10.1

## 2. Hadoop+Spark 分布式环境搭建：

### 2.1 准备工作

#### 1) 修改主机名

在 master 服务器上运行命令，编辑修改为 master，并使用 sudo reboot 重启：sudo vim /etc/hostname


在 slave 服务器上运行命令，编辑修改为 slave01/slave02/slave04，并使用 sudo reboot 重启：

```
[root@twt4507 ~]# sudo vim /etc/hostname
[root@twt4507 ~]# su hadoop
[hadoop@twt4507 root]$ sudo vim /etc/hostname
[sudo] password for hadoop:
[hadoop@twt4507 root]$ sudo vim /etc/hostname
[hadoop@twt4507 root]$ sudo reboot
Connection closing...Socket close.

Connection closed by foreign host.

Disconnected from remote host(华为云服务器) at 20:12:34.

Type `help' to learn how to use Xshell prompt.
[C:\~]$
Reconnecting in 30 seconds. Press any key to exit local shell.
.............................

Connecting to 1.94.37.155:22...
```

**2) 修改 host**

各自服务器的内网和公网 IP 已记录如上，在 master 上运行命令，编辑 hosts 文件，得到结果：

```
[hadoop@master root]$ sudo vim /etc/hosts

192.168.0.251   master
1.94.11.79      slave01
1.94.37.155     slave02
60.204.193.83   slave04
```

在 slave 上运行命令，编辑 hosts 文件，得到结果：

```
[hadoop@slave02 root]$ sudo vim /etc/hosts

1.94.33.146     master
192.168.0.204   slave02
127.0.0.1       localhost
```

**3) SSH 互相免密**

在 master 上 scp 传递公钥，并且输入 slave01@hadoop 用户密码，运行命令,在 master 主机上进行测试

```
[hadoop@master ~]$ scp ~/.ssh/id_rsa.pub hadoop@slave01:/home/had
p/
The authenticity of host 'slave01 (1.94.11.79)' can't be establis
d.
ECDSA key fingerprint is SHA256:YW4A9/2a+9TQe6oAaLRoR7Mdzl5on+NGs
f7Y/F2JU.
ECDSA key fingerprint is MD5:ac:da:ad:d6:16:e7:86:f9:3a:bf:bf:35:
:bc:a9:c3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'slave01,1.94.11.79' (ECDSA) to the li
 of known hosts.
hadoop@slave01's password:
id_rsa.pub                    100%  395   184.3KB/s   00:00
[hadoop@master ~]$ ssh slave01
Last login: Sat Oct 14 20:22:48 2023 from 127.0.0.1

        Welcome to Huawei Cloud Service
```

## 2.2 Hadoop 集群配置

**1) master 节点配置**

切换目录：cd /usr/local/hadoop/etc/hadoop

修改文件 slaves

修改文件 core-site.xml

```
  <configuration>
      <property>
          <name>hadoop.tmp.dir</name>
          <value>/usr/local/hadoop/tmp</value>
          <description>Abase for other temporary directories.</de
scription>
      </property>
      <property>
          <name>fs.defaultFS</name>
          <value>hdfs://master:9000</value>
      </property>
  </configuration>
```

- 修改 hdfs-site.xml

```
  <configuration>
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
    <property>
        <name>mapred.job.tracker</name>
        <value>master:9001</value>
    </property>
    <property>
        <name>dfs.namenode.http-address</name>
        <value>master:50070</value>
    </property>
  </configuration>
"hdfs-site.xml" 32L, 1091C                    32,3          Bot
```

- 修改 mapred-site.xml.template

```
  <configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
  </configuration>
"mapred-site.xml" 24L, 866C                   24,3          Bot
```

- 修改 yarn-site.xml

```
 <configuration>
  <!-- Site specific YARN configuration properties -->
      <property>
          <name>yarn.nodemanager.aux-services</name>
          <value>mapreduce_shuffle</value>
      </property>
      <property>
          <name>yarn.resourcemanager.hostname</name>
          <value>master</value>
      </property>
  </configuration>
"yarn-site.xml" 25L, 944C                     25,3          Bot
```

```
[hadoop@master ~]$ cd /usr/local/hadoop/etc/hadoop
[hadoop@master hadoop]$ vim slaves
[hadoop@master hadoop]$ vim core-site.xml
[hadoop@master hadoop]$ vim hdfs-site.xml
[hadoop@master hadoop]$ cp mapred-site.xml.template mapred-site.x
[hadoop@master hadoop]$ vim mapred-site.xml
[hadoop@master hadoop]$ vim yarn-site.xml
```

**3) slave 节点配置**

传送已修改的配置文件，在 master 节点上使用如下命令将 yarn-site.xml 等发送到所有从机上：

```
[hadoop@master ~]$ cd /usr/local/hadoop/etc/hadoop/
[hadoop@master hadoop]$ # on master
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/core-sit
xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
e01:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/yarn-site.xml hadoop@slave01:/us
core-site.xml                 100% 1089   541.7KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/hdfs-sit
xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
hdfs-site.xml                 100% 1091   491.8KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/mapred-s
e.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
mapred-site.xml               100%  866   396.7KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/yarn-sit
xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
yarn-site.xml                 100%  944   437.2KB/s   00:00
```

设置文件权限：检查文件变更，通过 cat 命令检查 slave 上的相关文件是否变更：

```
[hadoop@slave02 ~]$ sudo chown -R hadoop /usr/local/hadoop
[sudo] password for hadoop:
Sorry, try again.
[sudo] password for hadoop:
[hadoop@slave02 ~]$ cd /usr/local/hadoop/etc/hadoop/
[hadoop@slave02 hadoop]$ ls
core-site.xml  hdfs-site.xml  mapred-site.xml  yarn-site.xml
[hadoop@slave02 hadoop]$ cat /usr/local/hadoop/etc/hadoop/core-site.xml
<?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
```

输出中含有上一步中修改后的信息，则确认正确。比如，core-site.xml 文件输出如下：

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

  <configuration>
      <property>
          <name>hadoop.tmp.dir</name>
          <value>/usr/local/hadoop/tmp</value>
          <description>Abase for other temporary directories.</description>
      </property>
      <property>
          <name>fs.defaultFS</name>
          <value>hdfs://master:9000</value>
      </property>
  </configuration>
```

**4) 集群启动测试**

**Spark 配置**

- 切换配置目录：

```
[hadoop@master root]$ cd /usr/local/spark/conf
[hadoop@master conf]$ vim slaves
```

- 配置 slaves 文件(cp slaves.template slaves)，将 localhost 替换为以下：

```
slave01
slave02
slave04
~
                                              21,1          Bot
```

- 配置 spark-env.sh 文件：cp spark-env.sh.template spark-env.sh

```
[hadoop@master conf]$ vim spark-env.sh

#!/usr/bin/env bash
export SPARK_DIST_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classp
ath)
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export SPARK_MASTER_IP=192.168.0.251
```

- 复制 Spark 文件到各个 slave 节点

```
[hadoop@master conf]$ cd ~
[hadoop@master ~]$ scp ./spark.master.tar.gz slave01:/home/hadoop
```

- 节点替换文件，在 slave 上：

  sudo rm -rf /usr/local/spark/

  sudo tar -zxf /home/hadoop/spark.master.tar.gz -C /usr/local

  sudo chown -R hadoop /usr/local/spark

**启动 Spark 集群**

- 启动 hadoop 集群

```
[hadoop@master ~]$ cd /usr/local/hadoop/
[hadoop@master hadoop]$ sbin/start-all.sh
```

- 启动 master 节点

```
[hadoop@master hadoop]$ cd /usr/local/spark/
[hadoop@master spark]$ sbin/start-master.sh
```

• 启动所有 slave 节点

接着在 master 和各个 slave 节点运行 jps 命令，可得到相关正确结果截图，见第四部分实验结果及分析。

```
[hadoop@master root]$ cd /usr/local/hadoop
[hadoop@master hadoop]$ sbin/start-alevs.sh
```

web UI 查看，得到的结果见第四部分实验结果及分析。

## 四、实验结果及分析和（或）源程序调试过程

### 1. Spark 单机版搭建

#### 1.1 准备工作

##### 1.1.1 配置用户

配置 Hadoop 用户，并成功切换到 Hadoop 用户下：

```
[root@master ~]# su hadoop
[hadoop@master root]$
```

##### 1.1.2 配置 SSH

生成公钥配置来免密码登陆，登陆时不用输入密码，直接登陆，配置正确：

```
[hadoop@master ~]$ ssh localhost
Last login: Sun Oct 22 21:01:56 2023 from 127.0.0.1

        Welcome to Huawei Cloud Service
```

##### 1.1.3 配置 yum 源

##### 1.1.4 配置 Java 环境

正确配置 java，可以看到输出 java 版本号：

```
[hadoop@master ~]$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
```

##### 1.1.5 安装 python

输出 python 版本号，正确安装：

```
[hadoop@master ~]$ python
Python 2.7.5 (default, Jun 20 2023, 11:36:40)
```

### 1.2 Hadoop 安装

切换到 Hadoop 路径下并查看 Hadoop 版本号，能够正确输出说明 Hadoop 安装正确：

```
[hadoop@master spark]$ cd /usr/local/hadoop
[hadoop@master hadoop]$ ./bin/hadoop version
Hadoop 2.10.1
```

### 1.3 Spark 安装

启动 Spark，可以看到启动的界面和 Spark 版本号：

```
/usr/local/spark/python/pyspark/context.py:317: FutureWarning: Python 3.7 support is deprecated in Spark 3.4.
  warnings.warn("Python 3.7 support is deprecated in Spark 3.4.", FutureWarning)
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.4.1
      /_/

Using Python version 3.6.8 (default, Jun 20 2023 11:53:23)
Spark context Web UI available at http://master:4040
Spark context available as 'sc' (master = local[*], app id = local-1697979259422).
SparkSession available as 'spark'.
```

### 1.4 测试

进行简单测试，可以正确输出，说明 Spark 正确安装：

```
>>> 8*2+5
21
```

## 2. Hadoop+Spark 分布式环境搭建

### 2.1 准备工作

#### 2.1.1 修改主机名

四台服务器一台作为 master，三台作为 slave。修改 slave 从机为 01、02、04：

```
[hadoop@master /]$
```

```
[hadoop@slave02 hadoop]$
```

#### 2.1.2 修改 host

master 上修改编辑 host 文件，添加 master 内网 ip 和 slave 的外网 ip；在 slave 上添加 master 的外网 ip 和 slave 的内网 ip。

#### 2.1.3 SSH 互相免密

将 master 的 id_rsa.pub 分别传递给三台 slave 主机，实现 master 主机免密码登陆 slave01、slave02 、slave04：

```
[hadoop@master /]$ ssh slave01
Last login: Sun Oct 22 19:49:14 2023 from 127.0.0.1

        Welcome to Huawei Cloud Service
```

切换回 master 主机，需要输入密码：

```
[hadoop@slave01 ~]$ ssh master
hadoop@master's password:
Last login: Sun Oct 22 20:43:51 2023

        Welcome to Huawei Cloud Service
```

## 2.2 Hadoop 集群配置

### 2.2.1 master 节点配置

修改修改 master 主机上 hadoop 配置文件，修改 slaves 文件，将集群中三台 slave 机加入：

```
[hadoop@master hadoop]$ vim slaves

slave01
slave02
slave04
~
```

并修改文件 core-site.xml、hdfs-site.xml、mapred-site.xml、yarn-site.xml。

### 2.2.2 slave 节点配置

在 master 上节点上传送已修改的配置文件，将 yarn-site.xml 修改后的文件等发送到从机 slave01、02、04 上，可以看到 master 主机成功发送：

```
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/core-sit
xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
e01:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/yarn-site.xml hadoop@slave01:/us
core-site.xml                 100% 1089   541.7KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/hdfs-sit
xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
hdfs-site.xml                 100% 1091   491.8KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/mapred-s
e.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
mapred-site.xml               100%  866   396.7KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/yarn-sit
xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
yarn-site.xml                 100%  944   437.2KB/s   00:00
```

slave 主机正确接收（展示 slave02 接收情况），通过 cat 命令检查 slave 上的相关文件是否变更，输出中正确含有上一步中修改后的信息：

```
[hadoop@slave02 hadoop]$ ls
core-site.xml  hdfs-site.xml  mapred-site.xml  yarn-site.xml
```

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

  <configuration>
      <property>
          <name>hadoop.tmp.dir</name>
          <value>/usr/local/hadoop/tmp</value>
          <description>Abase for other temporary directories.</description>
      </property>
      <property>
          <name>fs.defaultFS</name>
          <value>hdfs://master:9000</value>
      </property>
  </configuration>
```

### 2.2.3 集群启动

在 master 上启动集群，输入 jps 命令查看，master 节点上出现下面四个进程，配置正确：

```
[hadoop@master hadoop]$ jps
13584 SecondaryNameNode
14024 Jps
13754 ResourceManager
13374 NameNode
```

slave 节点上出现如下三个进程，配置正确：

```
[hadoop@slave01 ~]$ jps
8372 Jps
8231 NodeManager
8120 DataNode
```

```
[hadoop@slave02 hadoop]$ jps
3314 NodeManager
3206 DataNode
3431 Jps
```

```
[hadoop@slave04 ~]$ jps
31510 NodeManager
31401 DataNode
31644 Jps
```

## 2.3 Spark 集群配置

### 2.3.1 配置 Spark

成功复制 Spark 文件到各个 slave 节点，在 slave 节点上可以正确接收到

spark 文件：

```
[hadoop@master local]$ cd ~
[hadoop@master ~]$ scp ./spark.master.tar.gz slave01:/home/hadoop
spark.master.tar.gz             65%  187MB 128.0KB/s   13:15 ETA
```

### 2.3.2 启动 Spark 集群

启动 Spark 集群，先启动 Hadoop 集群，再启动 master 节点，在 master 上运行 jps 命令可以看到如下五个进程，说明配置成功：

```
[hadoop@master spark]$ jps
7316 SecondaryNameNode
7478 ResourceManager
7784 Master
8185 NameNode
8586 Jps
```

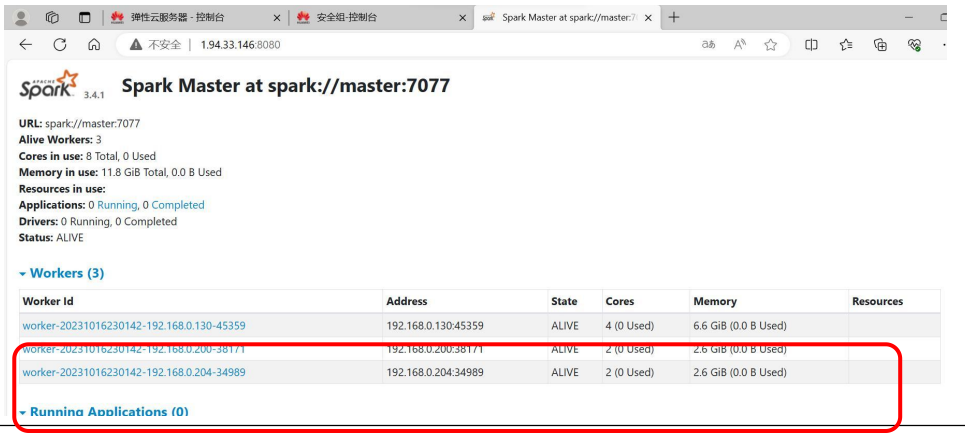
启动 slave 节点，在 slave 上运行 jps 命令后可看到如下五个进程，说明配置成功：

```
[hadoop@slave01 ~]$ jps
16233 Jps
15707 DataNode
15820 NodeManager
16013 Worker
```

```
[root@slave02 conf]# jps
4818 DataNode
5140 Worker
5414 Jps
4937 NodeManager
```

```
[hadoop@slave04 root]$ jps
31602 DataNode
32038 Worker
31787 NodeManager
3151 Jps
```

Web UI 查看，在 master 本地机器上，输入 1.94.33.146:8080 查看，出现如下界面，Web UI 正常显示三台 slave 机作为 workers，说明配置成功：




## 四、遇到的问题及解决方案

**1. 刚开始我们仅在 master 服务器上装了 hadoop 和 spark，但在后续 3.2.2 中配置集群时采用方法一出现了问题。**

### 3.1 Spark单机版搭建

> ⚠ 请注意，`3.1.1` 部分需在小组成员在**各自**云服务器上完成。`3.1.2~3.1.4` 小节只需在**一**台云服务器完成即可（作为master节点那台服务器）。

在进行Hadoop、Spark环境搭建前，我们需要进行一些准备工作。

• 对于方法一，如果 slave 节点上没有安装 hadoop，那么在 slave 节点上是没有/hadoop/etc/hadoop/目录路径的，直接采用命令传文件会提示路径不存在；即使自己新建该目录传文件后，也无法正常进行后续步骤，因为 slave 节点上没有 hadoop 的运行环境。因此采用方法一需要在 slave 节点上也按照 3.1.2 的步骤安装 hadoop.

**方法1：** 通过scp将上述变动文件发送至slave（可以大幅度减少传送时间）

1. 传送已修改的配置文件

   在master上节点上，使用如下命令将yarn-site.xml等发送到从机slave01上。

   发送给其它从机，如slave02，同理。

```
# on master
scp /usr/local/hadoop/etc/hadoop/core-site.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/hdfs-site.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/mapred-site.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/yarn-site.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
```

• 对于方法二，slave 节点上是不需要自己装 hadoop，但是经过尝试，这种方法确实很慢，很耗时间。

**方法2：** 压缩拷贝整个hadoop目录

- 在master节点上执行

```
cd /usr/local/
rm -rf ./hadoop/tmp      # 删除临时文件
rm -rf ./hadoop/logs/*   # 删除日志文件
# 压缩./hadoop文件，并重名为hadoop.master.tar.gz
tar -zcf ~/hadoop.master.tar.gz ./hadoop
```

## 2. hadoop 安装的北理工镜像站已经更换了：http://mirrors.bit.edu.cn/apache/hadoop/common/hadoop-2.10.1/

1. 下载

*北理工镜像站已经失效了*

> 为防止证书验证出现的下载错误，加上 `--no-check-certificate` ，相关讨论可见 issue#1

```
# 这里下载2.8.5版本，可能已失效，请去北理工镜像站，查看可下载的版本链接
# 建议下载版本低于3.0版本
sudo wget -O hadoop-2.8.5.tar.gz https://mirrors.cnnic.cn/apache/hadoop/common/hadoop-2.8.5/hadoop-2.8.5.tar.gz  --no-check-certificate
```

- `wget -O <指定下载文件名> <下载地址>`

如果直接使用该命令，可能会出现下载巨慢的问题，如会一直卡在这个界面(但实际上正在下载),一种解决方案就是在本地 windows 上下载后，然后本地上传到服务器(如 MobaXterm 是可以直接实现的)

```
[root@slave04 ~]# su hadoop
[hadoop@slave04 root]$ sudo wget -o hadoop-2.10.1.tar.gz https://mir
cate
[sudo] password for hadoop:
```

## 3. 在 3.2.3 spark 集群配置中---3 问题解决---重新启动集群中，第二行 sbin/start-slave.sh .....是有问题的

这会将 master 节点也加入 worker 中，得到如下图的效果(其中 192.168.0.251 是 master 节点却被加入到 worker 中了)，正确的应该是 sbin/start-slaves.sh.....[使用 sbin/stop-slave.sh 可以暂停 master 的 worker 身份]

- 重新启动集群

```
sbin/start-master.sh  # 先启动master
sbin/start-slave.sh spark://master内网ip:7077  # 指定master内网ip启动slaves节点
```

Status: ALIVE

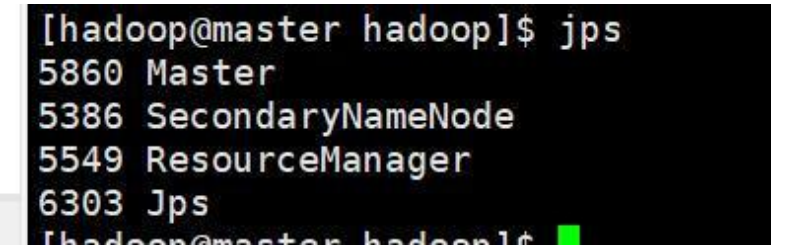
▾ Workers (4)

| Worker Id | Address | State | Cores | Memory | Reso |
|---|---|---|---|---|---|
| worker-20231016222123-192.168.0.251-39955 | 192.168.0.251:39955 | ALIVE | 2 (0 Used) | 2.6 GiB (0.0 B Used) | |
| worker-20231016222830-192.168.0.130-36195 | 192.168.0.130:36195 | ALIVE | 4 (0 Used) | 6.6 GiB (0.0 B Used) | |
| worker-20231016222830-192.168.0.200-37207 | 192.168.0.200:37207 | ALIVE | 2 (0 Used) | 2.6 GiB (0.0 B Used) | |
| worker-20231016222830-192.168.0.204-36063 | 192.168.0.204:36063 | ALIVE | 2 (0 Used) | 2.6 GiB (0.0 B Used) | |

▸ Running Applications (0)

▸ Completed Applications (0)

**4. 在 hadoop 集群正确启动的情况下再启用 spark 集群，出现了 master 节点缺乏 NameNode 进程的问题。**




按照以下步骤操作重启后得以解决：

> Q1：slave节点没有 DataNode 进程 / master节点没有 namenode 进程？

这个问题一般是由于在启动集群多次执行格式化命令：

```
bin/hdfs namenode -format
```