

第 3 章

Hadoop 的安装和使用

Hadoop 是一个开源的、可运行于大规模集群上的分布式计算平台,它主要包含分布式并行编程模型 MapReduce 和分布式文件系统 HDFS 等功能,已在业内得到广泛的应用。借助于 Hadoop,程序员可以轻松地编写分布式并程序,将其运行于计算机集群上,完成海量数据的存储与处理分析。

本章首先简要介绍 Hadoop 的发展情况;然后,阐述安装 Hadoop 之前的一些必要准备工作;最后,介绍安装 Hadoop 的具体方法,包括单机模式、伪分布式模式以及分布式模式。

3.1 Hadoop 简介

Hadoop 是 Apache 软件基金会旗下的一个开源分布式计算平台,为用户提供了系统底层细节透明的分布式基础架构。Hadoop 是基于 Java 语言开发的,具有很好的跨平台特性,并且可以部署在廉价的计算机集群中。Hadoop 的核心是 HDFS 和 MapReduce。

Apache Hadoop 版本分为 3 代,分别是 Hadoop 1.0、Hadoop 2.0 和 Hadoop 3.0。第一代 Hadoop 包含 0.20.x、0.21.x 和 0.22.x 三大版本,其中,0.20.x 最后演化成 1.0.x,变成了稳定版,而 0.21.x 和 0.22.x 则增加了 HDFS HA 等重要新特性。第二代 Hadoop 包含 0.23.x 和 2.x 两大版本,它们完全不同于 Hadoop 1.0,是一套全新的架构,均包含 HDFS Federation 和 YARN(Yet Another Resource Negotiator)两个系统。Hadoop 2.0 是基于 JDK 1.7 开发的,而 JDK 1.7 在 2015 年 4 月已停止更新,于是 Hadoop 社区基于 JDK 1.8 重新发布一个新的 Hadoop 版本,也就是 Hadoop 3.0。因此,到了 Hadoop 3.0 以后,JDK 版本的最低依赖从 1.7 变成了 1.8。Hadoop 3.0 中引入了一些重要的功能和优化,包括 HDFS 可擦除编码、多名称节点支持、任务级别的 MapReduce 本地优化、基于 cgroup 的内存和磁盘 I/O 隔离等。本书采用 Hadoop 3.3.5。

除了免费开源的 Apache Hadoop 以外,还有一些商业公司推出 Hadoop 的发行版。2008 年,Cloudera 成为第一个 Hadoop 商业化公司,并在 2009 年推出第一个 Hadoop 发行版。此后,很多大公司也加入了做 Hadoop 产品化的行列,如 MapR、Hortonworks、星环等。2018 年 10 月,Cloudera 和 Hortonworks 宣布合并。一般而言,商业化公司推出的 Hadoop 发行版也是以 Apache Hadoop 为基础,但是前者比后者具有更好的易用性、更多的功能以及更高的性能。

3.2 安装 Hadoop 前的准备工作

本节介绍安装 Hadoop 之前的一些准备工作,包括创建 hadoop 用户、更新 APT、安装 SSH 和安装 Java 环境等。

3.2.1 创建 hadoop 用户

本书全部采用 hadoop 用户登录 Linux 系统,并为 hadoop 用户增加了管理员权限。在前面的“第2章 Linux 系统的安装和使用”内容中,已经介绍了 hadoop 用户创建和增加权限的方法,一定按照该方法创建 hadoop 用户,并且使用 hadoop 用户登录 Linux 系统,然后再开始下面的学习内容。本书所有学习内容,都是采用 hadoop 用户登录 Linux 系统。

3.2.2 更新 APT

本书第2章介绍了 APT 软件作用和更新方法,为了确保 Hadoop 安装过程顺利进行,建议按照第2章介绍的方法,用 hadoop 用户登录 Linux 系统后打开一个终端,执行下面命令更新 APT 软件:

```
$ sudo apt-get update
```

3.2.3 安装 SSH

SSH 是 Secure Shell 的缩写,是建立在应用层和传输层基础上的安全协议。SSH 是目前较可靠、专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SSH 最初是 UNIX 系统上的一个程序,后来又迅速扩展到其他操作平台。SSH 由客户端和服务端的软件组成:服务端是一个守护进程,它在后台运行并响应来自客户端的连接请求;客户端包含 ssh 程序以及像 scp(远程复制)、slogin(远程登录)、sftp(安全文件传输)等其他的应用程序。

为什么在安装 Hadoop 之前要配置 SSH 呢?这是因为,Hadoop 名称节点(NameNode)需要启动集群中所有机器的 Hadoop 守护进程,这个过程需要通过 SSH 登录来实现。Hadoop 并没有提供 SSH 输入密码登录的形式,因此,为了能够顺利登录集群中的每台机器,需要将所有机器配置为“名称节点可以无密码登录它们”。

Ubuntu 默认已安装了 SSH 客户端,因此,这里还需要安装 SSH 服务端,在 Linux 的终端中执行以下命令:

```
$ sudo apt-get install openssh-server
```

安装后,可以使用如下命令登录本机:

```
$ ssh localhost
```

执行该命令后会出现图 3-1 所示的提示信息(SSH 首次登录提示),输入 yes,然后按提

示输入密码 `hadoop`,就登录到本机了。

```
hadoop@DBLab-XMU:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is a9:28:e0:4e:89:40:a4:cd:75:8f:0b:8b:57:79:67:86.
Are you sure you want to continue connecting (yes/no)? yes
```

图 3-1 SSH 登录提示信息

这里在理解上会有一点“绕弯”。也就是说,原本登录进入 Linux 系统以后,就是在本机上,这时,在终端中输入的每条命令都是直接提交给本机去执行,然后,又在本机上使用 SSH 方式登录到本机,这时,在终端中输入的命令,是通过 SSH 方式提交给本机处理。如果换成包含两台独立计算机的场景,SSH 登录会更容易理解。例如,有两台计算机 A 和 B 都安装了 Linux 系统,计算机 B 上安装了 SSH 服务端,计算机 A 上安装了 SSH 客户端,计算机 B 的 IP 地址是 59.77.16.33,在计算机 A 上执行命令 `ssh 59.77.16.33`,就实现了通过 SSH 方式登录计算机 B 上面的 Linux 系统,在计算机 A 的 Linux 终端中输入的命令,都会提交给计算机 B 上的 Linux 系统执行,也就是说,在计算机 A 上操作计算机 B 中的 Linux 系统。现在只有一台计算机,就相当于计算机 A 和 B 都在同一台机器上,所以理解起来就会有点“绕弯”。

但是,这样登录需要每次输入密码,所以,需要配置成 SSH 无密码登录会比较方便,而且,在 Hadoop 集群中,名称节点要登录某台机器(数据节点)时,也不可能人工输入密码,所以,也需要设置成 SSH 无密码登录。

首先输入命令 `exit` 退出刚才的 SSH,就回到了原先的终端窗口;然后可以利用 `ssh-keygen` 生成密钥,并将密钥加入授权中,命令如下:

```
$ cd ~/.ssh/ #若没有该目录,先执行一次 ssh localhost
$ ssh-keygen -t rsa #会有提示,按 Enter 键即可
$ cat ./id_rsa.pub >> ./authorized_keys #加入授权
```

此时,再执行 `ssh localhost` 命令,无须输入密码就可以直接登录了,如图 3-2 所示。

```
hadoop@ubuntu:~$ ssh localhost
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

31 packages can be updated.
2 updates are security updates.

Last login: Mon Jun 12 00:24:04 2023 from 127.0.0.1
```

图 3-2 SSH 登录后的提示信息

3.2.4 安装 Java 环境

由于 Hadoop 本身是使用 Java 语言编写的,因此 Hadoop 的开发和运行都需要 Java 的支持,对于 Hadoop 3.3.5 而言,要求使用 JDK 1.8 或者更新的版本。

访问 Oracle 官网(<https://www.oracle.com/technetwork/java/javase/downloads>)下载 JDK 1.8 安装包。或者也可以访问本书官网,进入“下载专区”,在“软件”目录下找到文件 `jdk-8u371-linux-x64.tar.gz` 并将其下载到本地。这里假设下载得到的 JDK 安装文件保存在

Ubuntu 系统的/home/hadoop/Downloads/目录下。

执行如下命令创建/usr/lib/jvm 目录用来存放 JDK 文件：

```
$cd /usr/lib  
$sudo mkdir jvm
```

#创建/usr/lib/jvm 目录用来存放 JDK 文件

执行如下命令对安装文件进行解压缩：

```
$cd ~  
$cd Downloads  
$sudo tar -zxvf ./jdk-8u371-linux-x64.tar.gz -C /usr/lib/jvm
```

#进入 hadoop 用户的主目录

下面继续执行如下命令，设置环境变量：

```
$vim ~/.bashrc
```

上面命令使用 vim 编辑器打开了 hadoop 这个用户的环境变量配置文件，在这个文件的开头位置，添加如下几行内容：

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_371  
export JRE_HOME=${JAVA_HOME}/jre  
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib  
export PATH=${JAVA_HOME}/bin:$PATH
```

保存.bashrc 文件并退出 vim 编辑器。然后，继续执行如下命令让.bashrc 文件的配置立即生效：

```
$source ~/.bashrc
```

这时，可以使用如下命令查看是否安装成功：

```
$java -version
```

如果能够在屏幕上返回如下信息，则说明安装成功：

```
java version "1.8.0_371"  
Java(TM) SE Runtime Environment (build 1.8.0_371-b11)  
Java HotSpot(TM) 64-Bit Server VM (build 25.371-b11, mixed mode)
```

至此，就成功安装了 Java 环境。下面就可以进入 Hadoop 的安装。

3.3 安装 Hadoop

Hadoop 包括 3 种安装模式。

- (1) 单机模式：只在一台机器上运行，存储是采用本地文件系统，没有采用 HDFS。
- (2) 伪分布式模式：存储采用 HDFS，但是 HDFS 的名称节点和数据节点都在同一台

机器上。

(3) 分布式模式：存储采用 HDFS, 而且 HDFS 的名称节点和数据节点位于不同机器上。

本节介绍 Hadoop 的具体安装方法, 包括下载安装文件、单机模式配置、伪分布式模式配置、分布式模式配置等。

3.3.1 下载安装文件

本书采用的 Hadoop 版本是 3.3.5, 可以到 Hadoop 官网下载安装文件(<https://archive.apache.org/dist/hadoop/core/hadoop-3.3.5/>), 或者也可以到本书官网的“下载专区”中下载安装文件, 单击进入下载专区后, 在“软件”这个文件夹中, 找到文件 `hadoop-3.3.5.tar.gz`, 下载到本地。由于本书全部采用 hadoop 用户登录 Linux 系统, 所以, `hadoop-3.3.5.tar.gz` 文件会被保存到 `/home/hadoop/Downloads/` 目录下。

下载完安装文件以后, 需要对文件进行解压。按照 Linux 系统使用的默认规范, 用户安装的软件一般都是存放在 `/usr/local/` 目录下。使用 hadoop 用户登录 Linux 系统, 打开一个终端, 执行如下命令:

```
$ sudo tar -zxvf ~/Downloads/hadoop-3.3.5.tar.gz -C /usr/local          #解压到/usr/local 中
$ cd /usr/local/
$ sudo mv ./hadoop-3.3.5/ ./hadoop                                  #将文件夹名改为 hadoop
$ sudo chown -R hadoop:hadoop ./hadoop                             #修改文件权限
```

Hadoop 解压后即可使用, 可以输入如下命令来检查 Hadoop 是否可用, 成功则会显示 Hadoop 版本信息:

```
$ cd /usr/local/hadoop
$ ./bin/hadoop version
```

3.3.2 单机模式配置

Hadoop 默认模式为非分布式模式(本地模式), 无须进行其他配置即可运行。Hadoop 附带了丰富的例子, 运行如下命令可以查看所有例子:

```
$ cd /usr/local/hadoop
$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar
```

上述命令执行后, 会显示所有例子的简介信息, 包括 `grep`、`join`、`wordcount` 等。这里选择运行 `grep` 例子, 可以先在 `/usr/local/hadoop` 目录下创建一个文件夹 `input`, 并复制一些文件到该文件夹下; 然后, 运行 `grep` 程序, 将 `input` 文件夹中的所有文件作为 `grep` 的输入, 让 `grep` 程序从所有文件中筛选出符合正则表达式 `dfs[a-z.]` 的单词, 并统计单词出现的次数; 最后, 把统计结果输出到 `/usr/local/hadoop/output` 文件夹中。完成上述操作的具体命令如下:

```
$ cd /usr/local/hadoop
$ mkdir input
$ cp ./etc/hadoop/*.xml ./input          #将配置文件复制到 input 目录下
$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar
  grep ./input ./output 'dfs[a-z.]+'
$ cat ./output/*                        #查看运行结果
```

执行成功后,如图 3-3 所示,作业的相关信息会输出在屏幕上,输出的结果是符合正则表达式的,单词 dfsadmin 出现了 1 次。

```
hadoop@ubuntu:/usr/local/hadoop$ cat ./output/*
1
dfsadmin
```

图 3-3 grep 程序运行结果

需要注意的是,Hadoop 默认不会覆盖结果文件,因此,再次运行上面实例会提示出错。如果要再次运行,需要先使用如下命令把 output 文件夹删除:

```
$ rm -r ./output
```

3.3.3 伪分布式模式配置

Hadoop 可以在单个节点(一台机器)上以伪分布式的方式运行,同一个节点既作为名称节点(NameNode),也作为数据节点(DataNode),读取的是分布式文件系统 HDFS 中的文件。

1. 修改配置文件

需要配置相关文件,才能够让 Hadoop 在伪分布式模式下顺利运行。Hadoop 的配置文件位于 /usr/local/hadoop/etc/hadoop/ 中,进行伪分布式模式配置时,需要修改两个配置文件,即 core-site.xml 和 hdfs-site.xml。

可以使用 vim 编辑器打开 core-site.xml 文件,它的初始内容如下:

```
<configuration>
</configuration>
```

修改以后,core-site.xml 文件的内容如下:

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

在上面的配置文件中, `hadoop.tmp.dir` 用于保存临时文件, 若没有配置 `hadoop.tmp.dir` 这个参数, 则默认使用的临时目录为 `/tmp/hadoo-hadoop`, 而这个目录在 Hadoop 重启时有可能被系统清理掉, 导致一些意想不到的问题, 因此, 必须配置这个参数。 `fs.defaultFS` 这个参数, 用于指定 HDFS 的访问地址, 其中, 9000 是端口号。

同样, 需要修改配置文件 `hdfs-site.xml`, 修改后的内容如下:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/data</value>
  </property>
</configuration>
```

在 `hdfs-site.xml` 文件中, `dfs.replication` 这个参数用于指定副本的数量, 因为在 HDFS 中, 数据会被冗余存储多份, 以保证可靠性和可用性。但是, 由于这里采用伪分布式模式, 只有一个节点, 只可能有 1 个副本, 因此设置 `dfs.replication` 的值为 1。 `dfs.namenode.name.dir` 用于设定名称节点的元数据的保存目录, `dfs.datanode.data.dir` 用于设定数据节点的数据保存目录, 这两个参数必须设定, 否则后面会出错。

配置文件 `core-site.xml` 和 `hdfs-site.xml` 的内容, 也可以直接到本书官网的“下载专区”下载, 位于“代码”目录下的“第 3 章”子目录下的“伪分布式模式配置”子目录中。

需要指出的是, Hadoop 的运行方式 (如运行在单机模式下还是运行在伪分布式模式下), 是由配置文件决定的, 启动 Hadoop 时会读取配置文件, 然后根据配置文件来决定运行在什么模式下。因此, 如果需要从伪分布式模式切换回单机模式, 只需要删除 `core-site.xml` 中的配置项即可。

2. 执行名称节点格式化

修改配置文件以后, 要执行名称节点的格式化, 命令如下:

```
$ cd /usr/local/hadoop
$ ./bin/hdfs namenode -format
```

如果格式化成功, 会看到 `successfully formatted` 的提示信息 (见图 3-4)。

如果在执行这一步时提示错误信息 `Error: JAVA_HOME is not set and could not be found`, 则说明之前设置 `JAVA_HOME` 环境变量的时候, 没有设置成功, 按前面的教程先设置好 `JAVA_HOME` 变量, 否则, 后面的过程都无法顺利进行。


```
2023-06-16 20:15:26,505 INFO common.Storage: Storage directory /usr/local/hadoop/tmp/dfs/name has been successfully formatted.
2023-06-16 20:15:26,562 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt_000000000000000000 using no compression
2023-06-16 20:15:26,764 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt_000000000000000000 of size 401 bytes saved in 0 seconds.
2023-06-16 20:15:26,788 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-06-16 20:15:26,840 INFO namenode.FSNamesystem: Stopping services started for active state
2023-06-16 20:15:26,841 INFO namenode.FSNamesystem: Stopping services started for standby state
2023-06-16 20:15:26,847 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2023-06-16 20:15:26,847 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
*****/
```

图 3-4 执行名称节点格式化后的提示信息

3. 启动 Hadoop

执行下面命令启动 Hadoop:

```
$ cd /usr/local/hadoop
$ ./sbin/start-dfs.sh #start-dfs.sh 是个完整的可执行文件,中间没有空格
```

如果出现图 3-5 所示的 SSH 提示,输入 yes 即可:

```
hadoop@DBLab-XMU:/usr/local/hadoop$ sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-localhost.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-datanode-localhost.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is a9:28:e0:4e:89:40:a4:cd:75:8f:0b:8b:57:79:67:86.
Are you sure you want to continue connecting (yes/no)? yes
```

图 3-5 启动 Hadoop 后的提示信息

启动时可能会出现如下警告信息:

```
WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable WARN
```

这个警告提示信息可以忽略,并不会影响 Hadoop 正常使用。

如果启动 Hadoop 时遇到输出非常多 ssh: Could not resolve hostname xxx 的异常情况,如图 3-6 所示。

```
hadoop@vm: /usr/local/hadoop
library: ssh: Could not resolve hostname library: Name or service not known
which: ssh: Could not resolve hostname which: Name or service not known
disabled: ssh: Could not resolve hostname disabled: Name or service not known
warning:: ssh: Could not resolve hostname warning:: Name or service not known
stack: ssh: Could not resolve hostname stack: Name or service not known
```

图 3-6 Hadoop 启动后的错误提示信息

这个并不是 SSH 的问题,可以通过设置 Hadoop 环境变量来解决。首先,按组合键 Ctrl+C 中断启动过程;然后,使用 vim 编辑器打开文件 ~/.bashrc,在文件最上边的开始位置增加如下两行内容(设置过程与 JAVA_HOME 变量一样,其中,HADOOP_HOME 为 Hadoop 的安装目录):

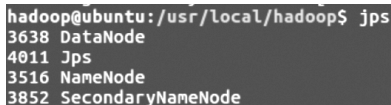
```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
```

保存该文件以后,务必要执行命令 source ~/.bashrc 使变量设置生效;然后,再次执行命令 ./sbin/start-dfs.sh 启动 Hadoop。

Hadoop 启动完成后,可以通过命令 jps 来判断是否成功启动,命令如下:

```
$ jps
```

若成功启动,则会列出如下进程(见图 3-7): NameNode、DataNode 和 SecondaryNameNode。如果看不到 SecondaryNameNode 进程,运行命令 ./sbin/stop-dfs.sh 关闭 Hadoop 相关进程;然后,再次尝试启动。如果看不到 NameNode 或 DataNode 进程,则表示配置不成功,仔细检查之前步骤,或通过查看启动日志排查原因。



```
hadoop@ubuntu:/usr/local/hadoop$ jps
3638 DataNode
4011 Jps
3516 NameNode
3852 SecondaryNameNode
```

图 3-7 Hadoop 启动成功以后的进程

通过 start-dfs.sh 命令启动 Hadoop 以后,就可以运行 MapReduce 程序处理数据,此时是对 HDFS 进行数据读写,而不是对本地文件进行读写。

4. Hadoop 无法正常启动的解决方法

一般可以通过查看启动日志来排查原因。启动时屏幕上会显示类似如下的信息:

```
DBLab-XMU: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-DBLab-XMU.out
```

其中,DBLab-XMU 对应的是机器名(你的机器名可能不是这个名称),不过,实际上启动日志信息是记录在下面这个文件中:

```
/usr/local/hadoop/logs/hadoop-hadoop-namenode-DBLab-XMU.log
```

所以,应该查看这个扩展名为.log 的文件,而不是.out 文件。此外,每一次的启动日志都是追加在日志文件之后,所以,需要拉到日志文件的最后面查看,根据日志记录的时间信息,就可以找到某次启动的日志信息。

当找到属于本次启动的一段日志信息以后,出错的提示信息一般会出现在最后面,通常是写着 Fatal、Error、Warning 或者 Java Exception 的地方。可以在网上搜索一下出错信息,寻找一些相关的解决方法。

如果执行 jps 命令后,找不到 DataNode 进程,则表示数据节点启动失败,可尝试如下的方法(注意这时删除 HDFS 中原有的所有数据,如果原有的数据很重要,不要这样做,不过

对于初学者而言,通常这个时候不会有重要数据):

```
$ ./sbin/stop-dfs.sh           #关闭
$ rm -r ./tmp                  #删除 tmp 文件,注意这时删除 HDFS 中原有的所有数据
$ ./bin/hdfs namenode -format  #重新格式化名称节点
$ ./sbin/start-dfs.sh          #重启
```

5. 使用 Web 界面查看 HDFS 信息

Hadoop 成功启动后,可以在 Linux 系统中(不是 Windows 系统)打开一个浏览器,在地址栏输入地址 `http://localhost:9870`(见图 3-8),就可以查看名称节点和数据节点信息,还可以在线查看 HDFS 中的文件。

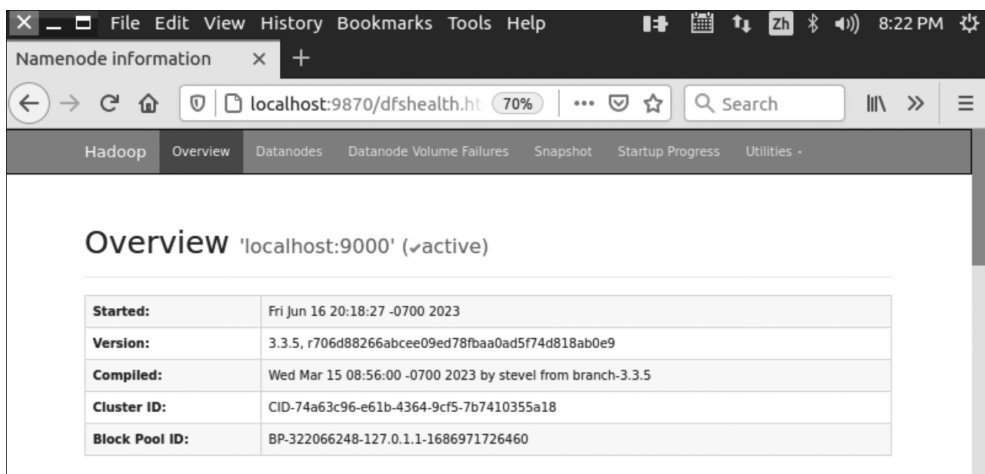


图 3-8 HDFS 的 Web 管理界面

6. 运行 Hadoop 伪分布式实例

上面的单机模式中,grep 例子读取的是本地数据,伪分布式模式下,读取的则是分布式文件系统 HDFS 上的数据。要使用 HDFS,首先需要在 HDFS 中创建用户目录(本书全部统一采用 hadoop 用户名登录 Linux 系统),命令如下:

```
$ cd /usr/local/hadoop
$ ./bin/hdfs dfs -mkdir -p /user/hadoop
```

上面的命令是分布式文件系统 HDFS 的操作命令,会在“第 4 章 HDFS 操作方法基础编程”中做详细介绍,目前只需要按照命令操作即可。

接着需要把本地文件系统的 `/usr/local/hadoop/etc/hadoop` 目录中的所有 XML 文件作为输入文件,复制到分布式文件系统 HDFS 中的 `/user/hadoop/input` 目录中,命令如下:

```
$ cd /usr/local/hadoop
$ ./bin/hdfs dfs -mkdir input           #在 HDFS 中创建 hadoop 用户对应的 input 目录
$ ./bin/hdfs dfs -put ./etc/hadoop/*.xml input  #把本地文件复制到 HDFS 中
```

复制完成后,可以通过如下命令查看 HDFS 中的文件列表:

```
$ ./bin/hdfs dfs -ls input
```

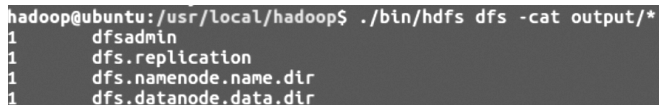
执行上述命令以后,可以看到 input 目录下的文件信息。现在就可以运行 Hadoop 自带的 grep 程序,命令如下:

```
$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar grep input output 'dfs[a-z.]+'
```

运行结束后,可以通过如下命令查看 HDFS 中的 output 文件夹中的内容:

```
$ ./bin/hdfs dfs -cat output/*
```

执行结果如图 3-9 所示。



```
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -cat output/*
1      dfsadmin
1      dfs.replication
1      dfs.namenode.name.dir
1      dfs.datanode.data.dir
```

图 3-9 在 Hadoop 伪分布式模式下运行 grep 的结果

需要强调的是,Hadoop 运行程序时,输出目录不能存在,否则会提示如下错误信息:

```
org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory
hdfs://localhost:9000/user/hadoop/output already exists
```

因此,若要再次执行 grep 程序,需要执行如下命令删除 HDFS 中的 output 文件夹:

```
$ ./bin/hdfs dfs -rm -r output    #删除 output 文件夹
```

7. 关闭 Hadoop

如果要关闭 Hadoop,可以执行下面命令:

```
$ cd /usr/local/hadoop
$ ./sbin/stop-dfs.sh
```

下次启动 Hadoop 时,无须进行名称节点的初始化(否则会出错),也就是说,不要再次执行 hdfs namenode -format 命令,每次启动 Hadoop 只需要直接运行 start-dfs.sh 命令即可。

8. 配置 PATH 变量

前面在启动 Hadoop 时,都要加上命令的路径,例如,./sbin/start-dfs.sh 这个命令中就带上了路径,实际上,通过设置 PATH 变量,就可以在执行命令时,不用带上命令本身所在的路径。例如,打开一个 Linux 终端,在任何一个目录下执行 ls 命令时,都没有带上 ls 命令的路径,实际上,执行 ls 命令时,是执行 /bin/ls 这个程序,之所以不需要带上路径,是因为

Linux 系统已经把 `ls` 命令的路径加入到 `PATH` 变量中,当执行 `ls` 命令时,系统是根据 `PATH` 这个环境变量包含的目录位置,逐一进行查找,直至在这些目录位置下找到匹配的 `ls` 程序(若没有匹配的程序,则系统会提示该命令不存在)。

知道了这个原理以后,同样可以把 `start-dfs.sh`、`stop-dfs.sh` 等命令所在的目录 `/usr/local/hadoop/sbin`,加入环境变量 `PATH` 中,这样,以后在任何目录下都可以直接使用命令 `start-dfs.sh` 启动 Hadoop,不用带上命令路径。具体操作方法是,首先使用 `vim` 编辑器打开 `~/.bashrc` 这个文件;然后,在这个文件的最前面位置加入如下单独一行:

```
export PATH=$PATH:/usr/local/hadoop/sbin
```

在后面的学习过程中,如果要继续把其他命令的路径也加入 `PATH` 变量中,也需要继续修改 `~/.bashrc` 这个文件。当后面要继续加入新的路径时,只要用英文冒号(;)隔开,把新的路径加到后面即可,例如,如果要继续把 `/usr/local/hadoop/bin` 路径增加到 `PATH` 中,只要继续追加到后面,如下所示:

```
export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin
```

添加后,执行命令 `source ~/.bashrc` 使设置生效。设置生效后,在任何目录下启动 Hadoop,都只要直接输入 `start-dfs.sh` 命令即可。同理,停止 Hadoop,也只需要在任何目录下输入 `stop-dfs.sh` 命令即可。

3.3.4 分布式模式配置

当 Hadoop 采用分布式模式部署和运行时,存储采用分布式文件系统 HDFS,而且, HDFS 的名称节点和数据节点位于不同机器上。这时,数据就可以分布到多个节点上,不同数据节点上的数据计算可以并行执行,这时的 MapReduce 分布式计算能力才能真正发挥作用。

这里使用 3 个节点(两台物理机器)来搭建集群环境,主机名分别为 `hadoop01`、`hadoop02` 和 `hadoop03`。3 个节点上的 Hadoop 组件分布如表 3-1 所示。

表 3-1 Hadoop 集群组件分布

名 称	hadoop01	hadoop02	hadoop03
HDFS	NameNode DataNode	DataNode	DataNode SecondaryNameNode
YARN	ResourceManager NodeManager	NodeManager	NodeManager

1. 安装虚拟机

此前,在 2.2.3 节安装 Linux 虚拟机这部分内容中,已经安装了虚拟机 `hadoop01`,按照相同的方法再安装另外两个虚拟机 `hadoop02` 和 `hadoop03`,或者也可以采用“克隆”虚拟机的方式快速生成两个新的虚拟机。由于 `hadoop02` 和 `hadoop03` 是从节点,不需要安装很多的软件,所以,配置可以比 `hadoop01` 低一些,例如,对于 `hadoop02` 和 `hadoop03` 而言,内存只

需要配置 4GB,磁盘只需要配置 20GB。

安装好虚拟机 hadoop02 和 hadoop03 以后,参照 3.2.2 节的方法,首先创建 hadoop 用户,然后使用 hadoop 用户登录 Linux 系统,安装 SSH 服务端,并安装 Java 环境。

2. 网络配置

在 Ubuntu 中,在 hadoop01 节点上执行如下命令修改主机名:

```
$ sudo vim /etc/hostname
```

执行上面命令后,就打开了/etc/hostname 这个文件,这个文件里面记录了主机名。因此,打开这个文件以后,里面就只有 ubuntu 这一行内容,可以直接删除,并修改为 hadoop01 (注意是区分大小写的),然后,保存退出 vim 编辑器,这样就完成了主机名的修改,需要重启 Linux 系统才能看到主机名的变化。

要注意观察主机名修改前后的变化。在修改主机名之前,如果用 hadoop 登录 Linux 系统,打开终端,进入 Shell 命令提示符状态,会显示如下内容:

```
hadoop@ ubuntu:~$
```

修改主机名并且重启 Linux 系统之后,用 hadoop 登录 Linux 系统,打开终端,进入 Shell 命令提示符状态,会显示如下内容:

```
hadoop@ hadoop01:~$
```

同理,按照相同的方法,把虚拟机 hadoop02 和 hadoop03 中的主机名分别修改为 hadoop02 和 hadoop03,并重启 Linux 系统。

然后,使用 ifconfig 命令获取每台虚拟机的 IP 地址,具体命令如下:

```
$ ifconfig
```

图 3-10 给出 ifconfig 命令的执行效果,从中可以看到,hadoop01 的 IP 地址是 192.168.91.128 (你的机器的 IP 地址可能和这个不同)。同理,可以查询到 hadoop02 的 IP 地址是 192.168.91.129 (你的机器的 IP 地址可能和这个不同),hadoop03 的 IP 地址是 192.168.91.130 (你的机器的 IP 地址可能和这个不同)。

```
hadoop@hadoop01:~$ ifconfig
ens33: flags=4096<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    link/ether 00:0c:29:62:3d:16 brd ff:ff:ff:ff:ff:ff
    inet addr:192.168.91.128 bcast:192.168.91.255 mask:255.255.255.0
    inet6 addr: fe80::a9cf:39a6:336a:7118/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:189 errors:0 dropped:0 overruns:0 frame:0
    TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:31562 (31.5 KB) TX bytes:11012 (11.0 KB)
```

图 3-10 ifconfig 命令执行效果

需要注意的是,建议将每台机器的 IP 地址设置为固定 IP 地址,不要使用动态分配 IP 地址,否则,每次重启系统以后有可能 IP 地址会动态变化,导致后面搭建的集群无法连接。

下面介绍把机器的 IP 地址设置为固定 IP 地址的方法。

在 Ubuntu 系统中新建一个终端,执行如下命令查询网关地址:

```
$ netstat -nr
```

查询结果如图 3-11 所示,从图中可以看到,网关是 192.168.91.2。

```
hadoop@hadoop01:~$ netstat -nr
Kernel IP routing table
Destination    Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0        192.168.91.2   0.0.0.0         UG      0 0        0 ens33
169.254.0.0    0.0.0.0        255.255.0.0     U       0 0        0 ens33
192.168.91.0   0.0.0.0        255.255.255.0   U       0 0        0 ens33
```

图 3-11 查询网关地址

单击 Ubuntu 系统界面右上角的齿轮(见图 3-12),在弹出的菜单中选择 System Settings 命令。

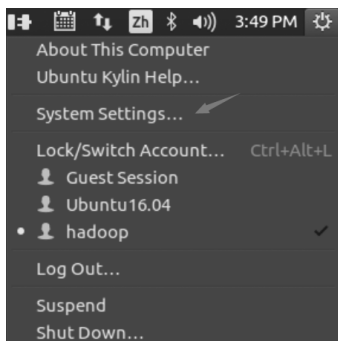


图 3-12 打开系统设置

在出现的界面中(见图 3-13),单击 Network 按钮。

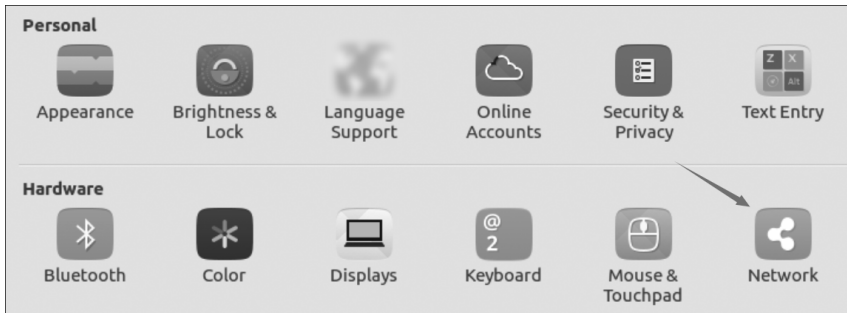


图 3-13 打开网络设置

在出现的界面中(见图 3-14),单击 Options 按钮。

在出现的界面中(见图 3-15),在 Method 右边的下来列表中选择 Manual,然后,把 Address 设置为 192.168.91.128,把 Netmask 设置为 255.255.255.0,把 Gateway 设置为 192.168.91.2,把 DNS servers 设置为 114.114.114.114,最后单击界面底部的 Save 按钮。然后,重新启动 Ubuntu 系统,IP 地址就生效了。注意,系统重启以后,一定要使用 ifconfig 命令检查 IP 地址是否已经设置成功,同时,也要在 Ubuntu 中打开一个浏览器,测试一下是否

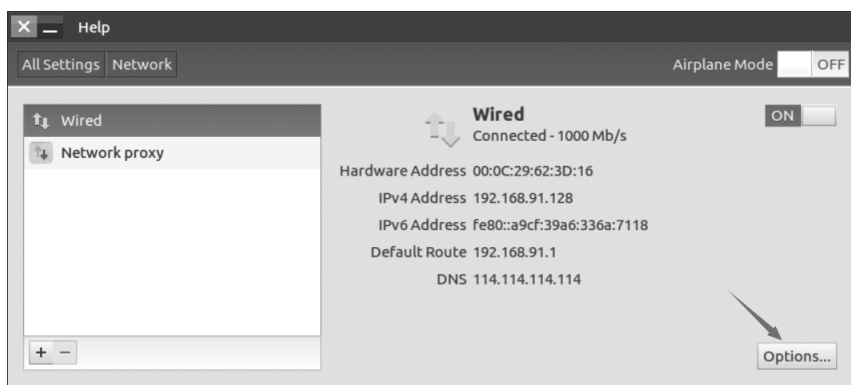


图 3-14 单击 Options 按钮

可以正常访问网络(如访问百度网站)。

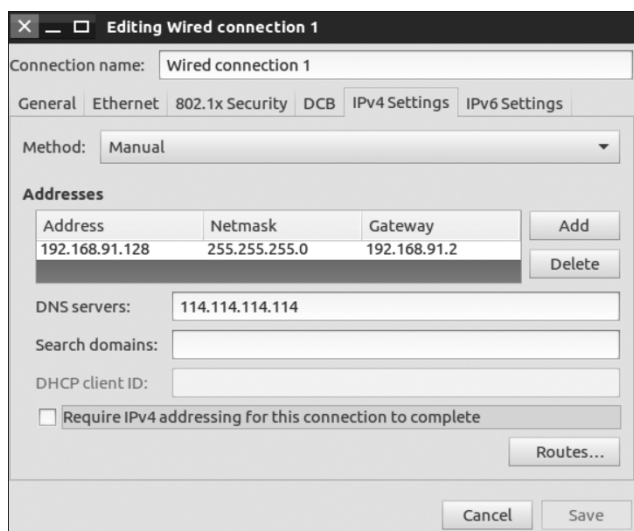


图 3-15 IP 地址设置界面

然后,在 hadoop01 中,执行如下命令打开并修改/etc/hosts 文件:

```
$ sudo vim /etc/hosts
```

可以在 hosts 文件中增加如下 3 条 IP 和主机名映射关系:

```
192.168.91.128    hadoop01
192.168.91.129    hadoop02
192.168.91.130    hadoop03
```

修改后的效果如图 3-16 所示。

需要注意的是,一般 hosts 文件中只能有一个 127.0.0.1,其对应主机名为 localhost,如果有多余 127.0.0.1 映射,应删除,特别是不能存在 127.0.0.1 hadoop01 这样的映射记录。修改后需要重启 Linux 系统。

上面完成了 hadoop01 的配置,接下来要继续完成对其他节点的配置修改。参照上面的

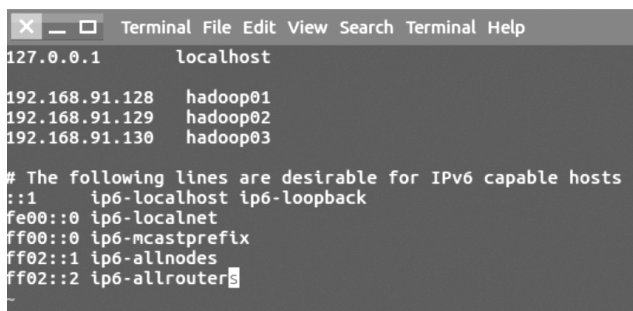


图 3-16 修改 IP 和主机名映射关系后的效果

方法,分别到 hadoop02 和 hadoop03 中,在 hosts 文件中增加如下 3 条 IP 和主机名映射关系:

```
192.168.91.128 hadoop01
192.168.91.129 hadoop02
192.168.91.130 hadoop03
```

修改完成以后,重新启动 Linux 系统。

需要在各个节点上都执行如下命令,测试是否相互 ping 得通,如果 ping 不通,后面就无法顺利配置成功。在每个节点上都执行下面 3 条命令:

```
$ ping hadoop01 -c 3          # 只 ping 3 次就会停止,否则要按组合键 Ctrl+C 中断 ping 命令
$ ping hadoop02 -c 3
$ ping hadoop03 -c 3
```

例如,在 hadoop01 节点上执行 ping 命令,如果 ping 通的话,会显示图 3-17 所示的结果。

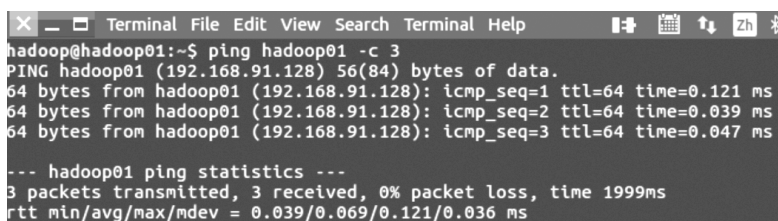


图 3-17 使用 ping 命令的效果

3. SSH 无密码登录节点

必须要让 hadoop01 节点可以 SSH 无密码登录到各个节点上(包括 hadoop01 可以 SSH 无密码登录自己)。

首先需要再次确认 3 个节点上已经安装了 SSH 服务端,如果之前没有安装,需要执行如下命令安装 SSH 服务端(如果此前已经安装就不用重复安装):

```
$ sudo apt-get install openssh-server
```

然后,生成 hadoop01 节点的公钥,如果之前已经生成过公钥(在 3.3.3 节安装伪分布式模式的 Hadoop 时生成过一次公钥),必须要删除原来生成的公钥,重新生成一次,因为前面对主机名进行了修改。在 hadoop01 节点执行如下命令:

```
$ cd ~/.ssh #如果没有该目录,先执行一次 ssh localhost
$ rm ./id_rsa* #删除之前生成的公钥(如果已经存在)
$ ssh-keygen -t rsa #执行该命令后,遇到提示信息,一直接 Enter 键就可以
```

为了让 hadoop01 节点能够无密码 SSH 登录本机,需要在 hadoop01 节点上执行如下命令:

```
$ cat ./id_rsa.pub >> ./authorized_keys
```

完成后可以执行命令 `ssh hadoop01` 来验证一下,可能会遇到提示信息,只要输入 yes 即可,测试成功后,执行 `exit` 命令返回原来的终端。

接下来,在 hadoop01 节点将上公钥传输到 hadoop02 和 hadoop03 节点:

```
$ scp ~/.ssh/id_rsa.pub hadoop@hadoop02:/home/hadoop/
$ scp ~/.ssh/id_rsa.pub hadoop@hadoop03:/home/hadoop/
```

上面的命令中,scp 是 secure copy 的简写,用于在 Linux 下进行远程复制文件,类似于 cp 命令,不过,cp 只能在本机中复制。执行 scp 时会要求输入 hadoop02 和 hadoop03 节点上 hadoop 用户的密码,输入完成后会提示传输完毕,如图 3-18 所示。传输完成以后,在 hadoop02 和 hadoop03 节点上的 /home/hadoop 目录下就可以看到文件 id_rsa.pub 了。

```
hadoop@hadoop01:~$ scp ~/.ssh/id_rsa.pub hadoop@hadoop02:/home/hadoop/
The authenticity of host 'hadoop02 (192.168.91.129)' can't be established.
ECDSA key fingerprint is SHA256:9k8cB6HLwyrHyueVwM3RP06FAxXa6MoyUj8EXTqKRwY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop02,192.168.91.129' (ECDSA) to the list of known hosts
hadoop@hadoop02's password:
```

图 3-18 执行 scp 命令的效果

接着在节点 hadoop02 和 hadoop03 上分别执行如下命令将 SSH 公钥加入授权:

```
$ mkdir ~/.ssh #如果不存在该文件夹需先创建,若已存在,则忽略本命令
$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
$ rm ~/id_rsa.pub #用完以后就可以删掉
```

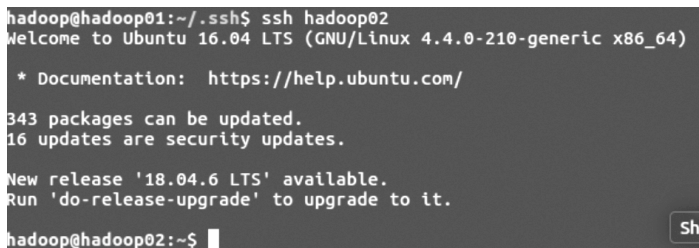
这样,在 hadoop01 节点上就可以无密码 SSH 登录到各个节点了(包括 hadoop01、hadoop02 和 hadoop03),可在 hadoop01 节点上执行如下命令进行检验:

```
$ ssh hadoop02
$ ssh hadoop03
```

执行该命令的效果如图 3-19 所示。

4. 下载安装文件

如果 hadoop01 节点上已经安装了 Hadoop(如之前安装过伪分布式的 Hadoop),需要



```
hadoop@hadoop01:~/.ssh$ ssh hadoop02
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

343 packages can be updated.
16 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

hadoop@hadoop02:~$
```

图 3-19 ssh 命令执行效果

先将其删除。

在 hadoop01 节点上下载 Hadoop 安装文件,并执行如下命令:

```
$ sudo tar -zxvf ~/Downloads/hadoop-3.3.5.tar.gz -C /usr/local
#解压到/usr/local 中
$ cd /usr/local/
$ sudo mv ./hadoop-3.3.5/ ./hadoop #将文件夹名改为 hadoop
$ sudo chown -R hadoop:hadoop ./hadoop #修改文件权限
```

5. 配置 PATH 变量

在前面的伪分布式安装内容中,已经介绍过 PATH 变量的配置方法。可以按照同样的方法进行配置,这样就可以在任意目录中直接使用 hadoop、hdfs 等命令了。如果还没有配置 PATH 变量,那么需要在 hadoop01 节点上进行配置。首先执行命令 `vim ~/.bashrc`,也就是使用 vim 编辑器打开 `~/.bashrc` 文件,然后,在该文件最上面的位置加入下面一行内容:

```
export PATH=$PATH:/usr/local/hadoop/bin:/usr/local/hadoop/sbin
```

保存后执行命令 `source ~/.bashrc`,使配置生效。

6. 配置集群/分布式环境

在配置集群/分布式模式时,需要修改 `/usr/local/hadoop/etc/hadoop` 目录下的配置文件,这里仅设置正常启动所必需的设置项,包括 `workers`、`core-site.xml`、`hdfs-site.xml`、`mapred-site.xml` 和 `yarn-site.xml` 共 5 个文件,更多设置项可查看官方说明。

(1) 修改文件 workers。

需要把所有数据节点的主机名写入该文件,每行一个。把 hadoop01 节点中的 `workers` 文件中原来的 `localhost` 删除,添加如下 3 行内容:

```
hadoop01
hadoop02
hadoop03
```

(2) 修改文件 core-site.xml。

把 hadoop01 节点中的 `core-site.xml` 文件修改为如下内容:

```

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop01:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:/usr/local/hadoop/tmp</value>
    <description>Abase for other temporary directories.</description>
  </property>
</configuration>

```

各个配置项的含义可以参考前面伪分布式模式时的介绍,这里不再赘述。

(3) 修改文件 hdfs-site.xml。

对于 Hadoop 的分布式文件系统 HDFS 而言,一般都是采用冗余存储,冗余因子通常为 3,也就是说,一份数据保存 3 份副本,所以,dfs.replication 的值设置为 3。把 hadoop01 节点中的 hdfs-site.xml 设置为如下内容:

```

<configuration>
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>hadoop03:50090</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/data</value>
  </property>
</configuration>

```

(4) 修改文件 mapred-site.xml。

hadoop01 节点中的 /usr/local/hadoop/etc/hadoop 目录下有一个 mapred-site.xml,把 mapred-site.xml 文件配置成如下内容:

```

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>

```

```

        <value>hadoop01:10020</value>
    </property>
    <property>
        <name>mapreduce.jobhistory.webapp.address</name>
        <value>hadoop01:19888</value>
    </property>
    <property>
        <name>yarn.app.mapreduce.am.env</name>
        <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
    </property>
    <property>
        <name>mapreduce.map.env</name>
        <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
    </property>
    <property>
        <name>mapreduce.reduce.env</name>
        <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
    </property>
</configuration>

```

(5) 修改文件 yarn-site.xml。

把 hadoop01 节点中的 yarn-site.xml 文件配置成如下内容：

```

<configuration>
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>hadoop01</value>
    </property>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>

```

上述 5 个文件全部配置完成以后,需要把 hadoop01 节点上的 /usr/local/hadoop 文件夹复制到各个节点上。如果之前已经运行过伪分布式模式,建议在切换到集群模式之前首先删除之前在伪分布式模式下生成的临时文件。具体来说,需要首先在 hadoop01 节点上执行如下命令：

```

$ cd /usr/local/hadoop
$ sudo rm -r ./tmp                                # 删除 Hadoop 临时文件
$ sudo rm -r ./logs/*                             # 删除日志文件
$ cd /usr/local
$ tar -zcf ~/hadoop.master.tar.gz ./hadoop        # 先压缩再复制
$ cd ~
$ scp ./hadoop.master.tar.gz hadoop02:/home/hadoop
$ scp ./hadoop.master.tar.gz hadoop03:/home/hadoop

```