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，已经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行程序，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；其次，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

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等重要的新特性。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JDK1.8重新发布一个新的Hadoop版本，即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 3.1.3。

除了免费开源的Apache Hadoop以外，还有一些商业公司推出Hadoop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。在第 2 章已经介绍了 hadoop 用户创建和增加权限的方法，一定要按照该方法先创建 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。

3.2.2 更新 APT

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 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，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如图 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 首次登录提示信息

这里在理解上会有一点“绕弯”。也就是说，原本登录 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，即在计算机 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”。

由于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。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 LTS (GNU/Linux 4.4.0-171-generic
 x86_64)

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

399 packages can be updated.
19 updates are security updates.

Last login: Sun Jan 26 10:59:05 2020 from 192.168.20.1
```

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 3.1.3 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，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-8u162-linux-x64.tar.gz 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 ~                # 进入 hadoop 用户的主目录
$cd Downloads
$sudo tar -zxvf ./jdk-8u162-linux-x64.tar.gz -C /usr/lib/jvm
```

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，在这个文件的开头添加如下几行内容：

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_162
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，可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_162"
Java(TM) SE Runtime Environment (build 1.8.0_162-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.162-b12, 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。

3.3.1　下载安装文件

本书采用的 Hadoop 版本是 3.1.3，可以到 Hadoop 官网（http://mirrors.cnnic.cn/apache/hadoop/common/）下载安装文件，也可以到本书官网的“下载专区”中下载安装文件，进入“下载专区”后，在“软件”目录下找到文件 hadoop-3.1.3.tar.gz，下载到本地。下载的方法是，在 Linux 系统中（不是在 Windows 系统中），打开浏览器，一般自带了火狐（FireFox）浏览器。打开浏览器后，访问本书官网，下载 hadoop-3.1.3.tar.gz。火狐浏览器默认会把下载文件都保存到当前用户的下载目录，由于本书全部采用 hadoop 用户登录 Linux 系统，所以，hadoop-3.1.3.tar.gz 文件会被保存到“/home/hadoop/下载/”目录下。

需要注意的是，如果是在 Windows 系统下载安装文件 hadoop-3.1.3.tar.gz，则需要通过 FTP 软件上传到 Linux 系统的“/home/hadoop/下载/”目录下，这个目录是本书所有安装文件的中转站。

下载完安装文件后，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 -zxf ～/下载/hadoop-3.1.3.tar.gz -C /usr/local    #解压到/usr/local 目录中
$cd /usr/local/
$sudo mv ./hadoop-3.1.3/ ./hadoop        #将文件夹名改为 hadoop
$sudo chown -R 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 的默认模式为本地模式（非分布式模式），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-3.1.3.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-3.1.3
.jar grep ./input ./output 'dfs[a-z.]+'
$cat ./output/*                  #查看运行结果
```

执行成功后结果如图3-3所示，输出了作业的相关信息，输出的结果是符合正则表达式的单词dfsadmin出现了1次。

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/目录下，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，所以，只可能有一个副本，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的保存目录，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”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）。

```
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = hadoop/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:  version = 3.1.3
************************************************************
*****/
......
2020-01-08 15:31:35,677 INFO common.Storage: Storage dir
ectory /usr/local/hadoop/tmp/dfs/name has been successfu
lly formatted.
......
/************************************************************
*****
SHUTDOWN_MSG: Shutting down NameNode at hadoop/127.0.1.1
************************************************************
*****/
```

图 3-4　执行名称节点格式化后的提示信息

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 环境变量时，没有设置成功，要按前面的内容介绍先设置好 JAVA_HOME 变量，否则，后面的过程都无法顺利进行。

3. 启动 Hadoop

执行下面命令启动 Hadoop：

```
$cd /usr/local/hadoop
$./sbin/start-dfs.sh    #start-dfs.sh是一个完整的可执行文件，中间没有空格
```

如果出现如图 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-na
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-da
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。

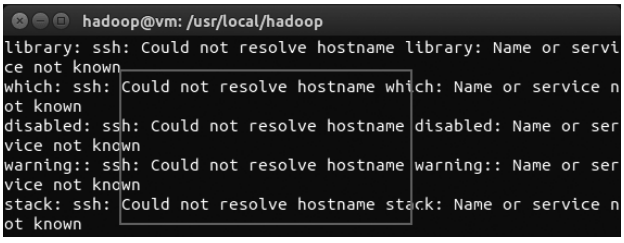

图 3-6　Hadoop 启动后的错误提示信息

这并不是 SSH 的问题，可以通过设置 Hadoop 环境变量来解决。首先，按 Ctrl＋C 键中断启动过程；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，务必先执行命令 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 判断是否成功启动，命令如下：

```
$jps
```

若成功启动，则会列出进程 NameNode、DataNode 和 SecondaryNameNode。如果看不到 SecondaryNameNode 进程，先运行命令./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，仔细检查之前的步骤，或通过查看启动日志排查原因。

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数据读写。

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，应该查看后缀为.log的文件，而不是后缀.out的文件。此外，每次的启动日志都是追加到日志文件后，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：

```
$./sbin/stop-dfs.sh              #关闭
$rm -r ./tmp                     #删除 tmp 文件，注意：这会删除 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
```

注意：这会删除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，通常这时不会有重要数据。

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3-7）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。

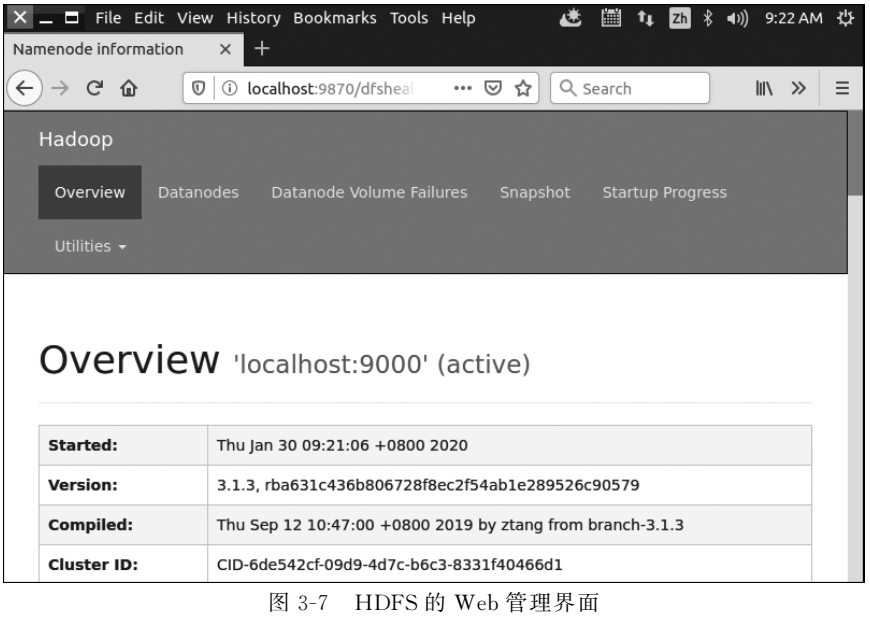

图3-7 HDFS的Web管理界面

6. 运行 Hadoop 伪分布式实例

在上面的单机模式中，grep 例子读取的是本地数据，在伪分布式模式下，读取的则是 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 HDFS 的操作命令，会在第 4 章做详细介绍，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 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-3.1.3
.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 3-8 所示。

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

图 3-8　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 命令中就带上了路径，实际上，通过配置 PATH 变量，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 PATH 变量中，当执行 ls 命令时，系统是根据 PATH 环境变量中包含的目录逐一进行查找，直至在这些目录下找到匹配的 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 文件，然后在这个文件的最前面加入如下单独一行：

```
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，例如：

```
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 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，MapReduce 分布式计算能力才能真正发挥作用。

为了降低分布式模式部署的难度，本书简单使用两个节点（两台物理机器）来搭建集群环境：一台机器作为 Master 节点，局域网 IP 地址为 192.168.1.121；另一台机器作为 Slave 节点，局域网 IP 地址为 192.168.1.122。由 3 个以上节点构成的集群，也可以采用类似的方法完成安装部署。

Hadoop 集群的安装配置大致包括以下 6 个步骤。

（1）选定一台机器作为 Master。

（2）在 Master 节点上创建 hadoop 用户、安装 SSH 服务端、安装 Java 环境。

（3）在 Master 节点上安装 Hadoop，并完成配置。

（4）在其他 Slave 节点上创建 hadoop 用户、安装 SSH 服务端、安装 Java 环境。

（5）将 Master 节点上的/usr/local/hadoop 目录复制到其他 Slave 节点上。

（6）在 Master 节点上开启 Hadoop。

上述这些步骤中，关于如何创建 hadoop 用户、安装 SSH 服务端、安装 Java 环境、安装 Hadoop 等过程，已经在前面介绍伪分布式模式配置时做了详细介绍，按照之前介绍的方法完成步骤(1)～(4)，这里不再赘述。在完成步骤(1)～(4)的操作以后，才可以继续进行下面的操作。

1. 网络配置

假设集群所用的两个节点（机器）都位于同一个局域网内。如果两个节点使用的是虚拟机方式安装的 Linux 系统，那么两者都需要更改网络连接方式为“桥接网卡”模式（可以参考第 2 章介绍的网络连接设置方法），才能实现多个节点互连，如图 3-9 所示。此外，一定要确保各个节点的 MAC 地址不能相同，否则会出现 IP 地址冲突。在第 2 章曾介绍过采用导入虚拟机镜像文件的方式安装 Linux 系统，如果是采用这种方式安装 Linux 系统，则有可能出现两台机器的 MAC 地址是相同的，因为一台机器复制了另一台机器的配置。因此，需要改变机器的 MAC 地址，如图 3-9 所示，可以单击界面右边的“刷新”按钮随机生成 MAC 地址，这样就可以让两台机器的 MAC 地址不同了。

网络配置完成以后，可以查看机器的 IP 地址（可以使用在第 2 章介绍过的 ifconfig 命令查看）。本书在同一个局域网内部的两台机器的 IP 地址分别是 192.168.1.121 和 192.168.1.122。

由于集群中有两台机器需要设置，所以，在接下来的操作中，一定要注意区分 Master 节点和 Slave 节点。为了便于区分 Master 节点和 Slave 节点，可以修改各个节点的主机名，这样，在 Linux 系统中打开一个终端以后，在终端窗口的标题和命令行中都可以看到主机名，就比较容易区分当前是对哪台机器进行操作。在 Ubuntu 中，可在 Master 节点上执行如下命令修改主机名：

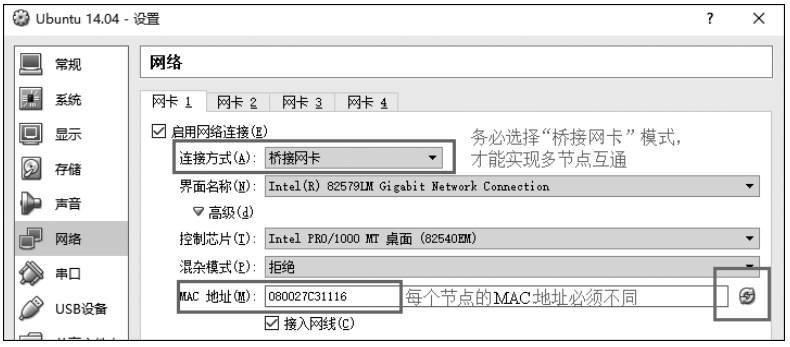

图 3-9 网络连接方式设置

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

执行上面命令后，就打开了/etc/hostname 文件，这个文件里面记录了主机名，例如，本书在第 2 章安装 Ubuntu 系统时，设置的主机名是 dblab-VirtualBox，因此，打开这个文件以后，里面就只有 dblab-VirtualBox 这一行内容，可以直接删除，并修改为 Master（注意是区分大小写的）；再保存并退出 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 用户登录 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@dblab-VirtualBox:~$
```

修改主机名并且重启系统之后，用 hadoop 用户登录 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@Master:~$
```

可以看出，这时就很容易辨认出当前是处于 Master 节点上进行操作，不会和 Slave 节点产生混淆。

再执行如下命令打开并修改 Master 节点中的/etc/hosts 文件：

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

可以在 hosts 文件中增加如下两条 IP 地址和主机名映射关系：

```
192.168.1.121   Master
192.168.1.122   Slave1
```

修改后的效果如图 3-10 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 Master 这样的映射记录。修改

图 3-10　修改 IP 地址和主机名映射关系后的效果

后需要重启 Linux 系统。

上面完成了 Master 节点的配置，接下来要继续完成对其他 Slave 节点的配置修改。本书只有一个 Slave 节点，主机名为 Slave1。参照上面的方法，把 Slave 节点上的/etc/hostname 文件中的主机名修改为 Slave1，同时，修改/etc/hosts 的内容，在 hosts 文件中增加如下两条 IP 地址和主机名映射关系：

```
192.168.1.121   Master
192.168.1.122   Slave1
```

修改完成以后，重新启动 Slave 节点的 Linux 系统。

这样就完成了 Master 节点和 Slave 节点的配置，需要在各个节点上都执行如下命令，测试是否可以相互 ping 通，如果 ping 不通，后面就无法顺利配置成功：

```
$ping Master -c 3       #只 ping 3 次就会停止，否则要按 Ctrl+C 键中断 ping 命令
$ping Slave1 -c 3
```

例如，在 Master 节点上 ping Slave1，如果 ping 通，会显示如图 3-11 所示的结果。

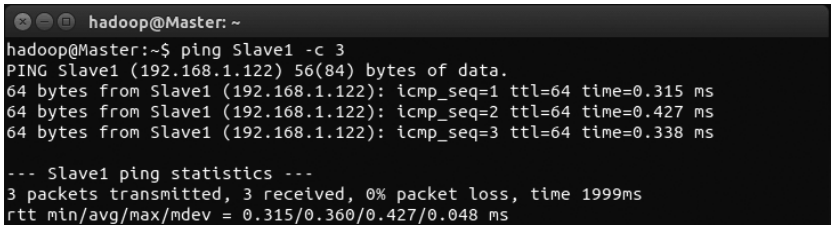

图 3-11　使用 ping 命令的效果

2. SSH 无密码登录节点

必须要让 Master 节点可以 SSH 无密码登录到各个 Slave 节点上。首先，生成 Master 节点的公匙，如果之前已经生成过公匙，必须要删除原来生成的公匙，重新生成一次，因为前面对主机名进行了修改。具体命令如下：

```
$cd ~/.ssh              #如果没有该目录，先执行一次 ssh localhost
$rm ./id_rsa*           #删除原来生成的公匙(如果已经存在)
$ssh-keygen -t rsa      #执行该命令后，遇到提示信息，一直按 Etner 键即可
```

为了让 Master 节点能够 SSH 无密码登录本机，需要在 Master 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 Master 来验证一下，可能遇到提示信息，只要输入 yes 即可，测试成功后，执行 exit 命令返回原来的终端。

其次，在 Master 节点将 Master 公匙传输到 Slave1 节点：

```
$scp ~/.ssh/id_rsa.pub hadoop@Slave1:/home/hadoop/
```

上面的命令中，scp 是 secure copy 的简写，用于在 Linux 系统下进行远程复制文件，类似于 cp 命令，但是，cp 只能在本机中复制。执行 scp 命令时会要求输入 Slave1 上 hadoop 用户的密码，输入完成后会提示传输完毕，如图 3-12 所示。

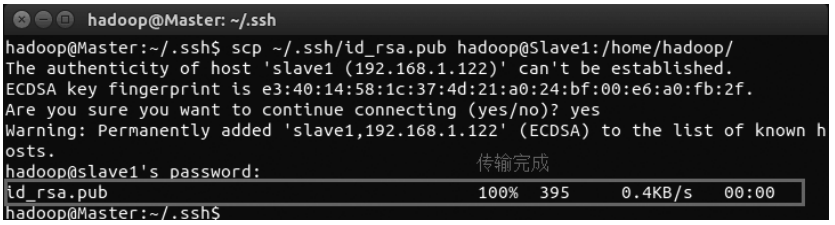

图 3-12 执行 scp 命令的效果

最后，在 Slave1 节点上将 SSH 公匙加入授权：

```
$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
```

如果有其他 Slave 节点，也要执行将 Master 公匙传输到 Slave 节点以及在 Slave 节点上加入授权这两步操作。

这样，在 Master 节点上就可以 SSH 无密码登录到各个 Slave 节点，可在 Master 节点上执行如下命令进行检验：

```
$ssh Slave1
```

执行该命令的效果如图 3-13 所示。

3. 配置 PATH 变量

在前面的伪分布式模式配置中，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 Master 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，在该文件最上面加入下面一行内容：

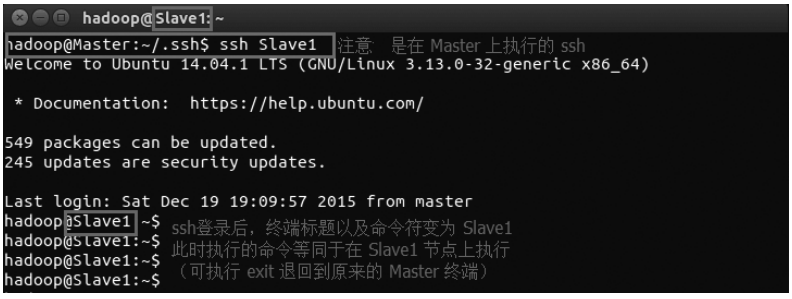

图 3-13　ssh 命令执行效果

```
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。

4. 配置集群/分布式环境

在配置集群/分布式环境时，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、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，默认为 localhost（即把本机作为数据节点），所以，在伪分布式模式配置时，就采用了这种默认的配置，使得节点既作为名称节点也作为数据节点。在进行分布式模式配置时，可以保留 localhost，让 Master 节点同时充当名称节点和数据节点，也可以删掉 localhost 这行，让 Master 节点仅作为名称节点使用。

本书让 Master 节点仅作为名称节点使用，因此将 workers 文件中原来的 localhost 删除，只添加如下一行内容：

```
Slave1
```

2）修改文件 core-site.xml

把 core-site.xml 文件修改为如下内容：

```
<configuration>
      <property>
            <name>fs.defaultFS</name>
            <value>hdfs://Master: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 HDFS 而言，一般都是采用冗余存储，冗余因子通常为 3，即一份数据保存 3 份副本。但是，本书只有一个 Slave 节点作为数据节点，即集群中只有一个数据节点，数据只能保存一份，所以，dfs.replication 的值还是设置为 1。hdfs-site.xml 的具体内容如下：

```
<configuration>
        <property>
                <name>dfs.namenode.secondary.http-address</name>
                <value>Master:50090</value>
        </property>
        <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>
```

4）修改文件 mapred-site.xml

/usr/local/hadoop/etc/hadoop 目录下有一个 mapred-site.xml.template，需要修改文件名，把它重命名为 mapred-site.xml，然后把 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>Master:10020</value>
```

```
        </property>
        <property>
                <name>mapreduce.jobhistory.webapp.address</name>
                <value>Master: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 yarn-site.xml 文件配置成如下内容：

```
<configuration>
        <property>
                <name>yarn.resourcemanager.hostname</name>
                <value>Master</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 Master 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 Master 节点上执行如下命令：

```
$cd /usr/local
$sudo rm -r ./hadoop/tmp            #删除 Hadoop 临时文件
$sudo rm -r ./hadoop/logs/*         #删除日志文件
$tar -zcf ~/hadoop.master.tar.gz ./hadoop   #先压缩再复制
$cd ~
$scp ./hadoop.master.tar.gz Slave1:/home/hadoop
```

然后在 Slave1 节点上执行如下命令：

```
$sudo rm -r /usr/local/hadoop        #删掉原有的 Hadoop 文件(如果存在)
$sudo tar -zxf ~/hadoop.master.tar.gz -C /usr/local
$sudo chown -R hadoop /usr/local/hadoop
```

同样，如果有其他 Slave 节点，也要执行将 hadoop.master.tar.gz 传输到 Slave 节点以及在 Slave 节点解压文件的操作。

首次启动 Hadoop 集群时，需要先在 Master 节点执行名称节点的格式化（只需要执行这一次，后面再启动 Hadoop 时，不要再次格式化名称节点），命令如下：

```
$hdfs namenode -format
```

现在就可以启动 Hadoop 了，启动需要在 Master 节点上进行，执行如下命令：

```
$start-dfs.sh
$start-yarn.sh
$mr-jobhistory-daemon.sh start historyserver
```

通过命令 jps 可以查看各节点所启动的进程。如果已经正确启动，则在 Master 节点上可以看到 NameNode、ResourceManager、SecondaryNameNode 和 JobHistoryServer 进程，如图 3-14 所示。

在 Slave 节点可以看到 DataNode 和 NodeManager 进程，如图 3-15 所示。

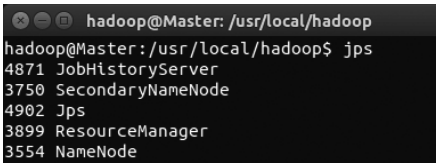

图 3-14 Master 节点上启动的进程

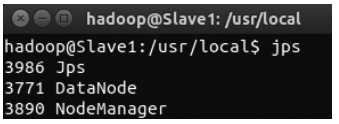

图 3-15 Slave 节点上启动的进程

缺少任一进程都表示出错。另外还需要在 Master 节点上通过命令 hdfs dfsadmin -report 查看数据节点是否正常启动，如果屏幕信息中的 Live datanodes 不为 0，则说明集群启动成功。由于本书只有一个 Slave 节点充当数据节点，因此，数据节点启动成功以后，会显示如图 3-16 所示的信息。

也可以在 Linux 系统的浏览器中输入地址 http://master:9870/，通过 Web 页面查看名称节点和数据节点的状态。如果不成功，可以通过启动日志排查原因。

这里再次强调，伪分布式模式和分布式模式切换时需要注意以下两点事项。

（1）从分布式模式切换到伪分布式模式时，不要忘记修改 workers 配置文件。

（2）在两者之间切换时，若遇到无法正常启动的情况，可以删除所涉及节点的临时文件夹，这样虽然之前的数据会被删掉，但能保证集群正确启动。所以，如果集群以前能启动，但后来启动不了，特别是数据节点无法启动，不妨试着删除所有节点（包括 Slave 节点）上的 /usr/local/hadoop/tmp 文件夹，再重新执行一次 hdfs namenode -format，再次启动即可。