目　录

第 4 章　ZooKeeper / 59

第 5 章　Hive / 79

第 6 章　Hadoop I/O 流操作 / 97

第 7 章 MapReduce 开发 / 131

第 8 章 Hadoop 数据建模 / 159

第 9 章 Hadoop 应用开发——点击流量 / 185

第 1 章 初识 Hadoop

学习目标

1. 了解 Hadoop 的概念。
2. 掌握 Hadoop 的基本安装。
3. 掌握 Hadoop 的环境安装。
4. 掌握 Hadoop 的模式配置。

知识导图

笔记

本章导读

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，已经在业内广泛地应用。借助于 Hadoop, 程序员可以轻松地编写分布式并行程序，并将其运行于计算机集群上，完成海量数据的存储与处理。

1.1 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 的核心是分布式文件系统（Hadoop distributed file system，HDFS）和 MapReduce。

Hadoop 被公认为行业大数据标准开源软件，在分布式环境下提供了海量数据的处理能力。几乎所有主流厂商都围绕 Hadoop 提供开发工具、开源软件、商业化工具和技术服务，如谷歌、微软、思科、淘宝等，都支持 Hadoop。

Apache Hadoop 版本有三代：第一代 Hadoop 称为 Hadoop 1.0；第二代 Hadoop 称为 Hadoop 2.0；第三代称为 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 HDFSHA 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 Hadoop1.0，是一套全新的架构，均包含 HDFS Federation 和 YARN（yet another resource negotiator）两个组件。第三代 Hadoop 基于 JDK 1.8 开发（由于 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，这直接迫使 Hadoop 社区基于 JDK 1.8 重新发布一个新的 Hadoop 版本，即 Hadoop 3.0）。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 NameNode 支持、MR Native Task 优化、YARN 基于 cgroup 的内存和磁盘 IO 隔离、YARN container resizing 等。

除了免费开源的 Apache Hadoop 以外，还有一些商业公司推出了 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 Hadoop 产品化的行列，如 MapR、Hortonworks、星环等。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、更多的功能和更高的性能。

Apache Hadoop 版本下载：

（1）各版本说明：http://hadoop.apache.org/releases.html。

（2）下载稳定版：找到一个镜像，下载 stable 文件夹下的版本。

（3）Hadoop 最全版本：http://svn.apache.org/repos/asf/hadoop/common/branches/，可直接导入到 eclipse 中。

常见的 Hadoop 不同版本的下载地址：

http://archive.apache.org/dist/hadoop/core/

http://archive.cloudera.com/cdh/3/

http://archive.cloudera.com/cdh4/cdh/4/

1.2 Hadoop 基础安装

笔记

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，没有采用分布式文件系统 HDFS。

（2）伪分布式模式。存储采用分布式文件系统 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。存储采用分布式文件系统 HDFS，而且，HDFS 的名称节点和数据节点不同。

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。

1.2.1 下载安装文件

可以到 Hadoop 官网（https://mirrors.cnnic.cn/apache/hadoop/common/）下载安装文件。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-3.3.0.tar.gz-C/usr/local# 解压到U/usr/local目录中
$cd/usr/local/
$sudo mv./hadoop-3.3.0/./hadoop          # 将文件夹名改为hadoop
$sudo chown-Rhadoop./hadoop  A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
```

1.2.2 单机模式配置

Hadoop 的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/hadoopjar./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.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：

笔记

```
$cd/usr/local/hadoop
$mkdir input
$cp./etc/hadoop/".xml./input# 将配置文件复制到 input 目录下
$./bin/hadoopjar./share/hadoop/mapreduce/hadoop-mapreduce-examples-".jargrep./%#('#(, $9"[&-
8.]+, $c&(./output/"        # 查看运行结果
```

执行成功后的结果如图 1-1 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 出现的次数。

图 1-1　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
```

1.2.3 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中的文件。

1. 修改配置文件

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 hdf-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>JAbase for other temporary directories.</description>
</property>
<property>
```

```
<name〉fs.defaultFS</name〉
<value>hdfs://localhost:9000</value〉
</property〉
</configuration〉
```

笔记

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，同时，必须配置 fs.defaultFS 这个参数，用于指定 HDFS 的访问地址，其中，9000 是端口。

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。因为在分布式文件系统 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，所以设置 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。

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/hdfsnamenode-format
```

如果格式化成功，会看到“ has been successfully formatted”和“ Exiting with status 0”的提示信息，如图 1-2 所示。若为“Exiting with status 1”，则表示出现错误。

笔记

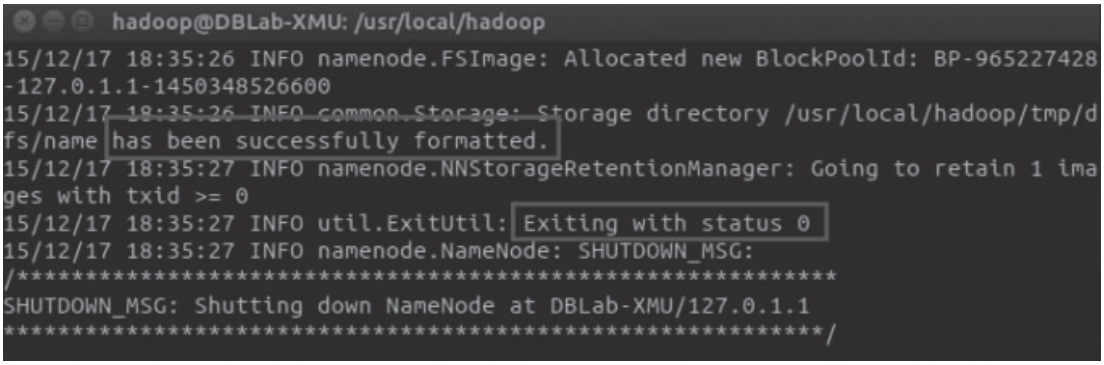

图 1-2　执行名称节点格式化后的提示信息

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”，则说明设置 JAVA_HOME 环境变量时没有设置成功，要先设置好 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
```

如果出现如下 SSH 提示，输入 yes 即可：

```
hadoop@DBLab-XMU:/usr/local/hadoop$sbin/start-dfs.sh
Starting nane nodeson[localhost]
localhost:starting namenode,loggingto/usr/local/hadoop/logs/hadoop-hadoop-nalocalhost:starting
datanode,loggingto/usr/local/hadoop/logs/hadoop-hadoop-daStartingsecondarynanenodes[0.0.0.0]
The authenti city of host10.0.0.0(9.9.0.9)can't be established.
ECDSAkey._finqerprintisa9:28:e9:4e:89:40:a4:cd:75:8f:0b:8b:57:79:67:86.
Are you sure you want to continue connecting(yes/no)?!yes
```

启动时可能会出现如下警告信息：

```
WAR Nutil.Native Code Loader:Unable to load native-hadoop library for your
platform***usingbuiltin-java classes wher eappli cable 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:Coul dnot resolve hostname xxx”的异常情况，如图 1-3 所示，这并不是 SSH 的问题，可以通过设置 Hadoop 环境变量来解决。

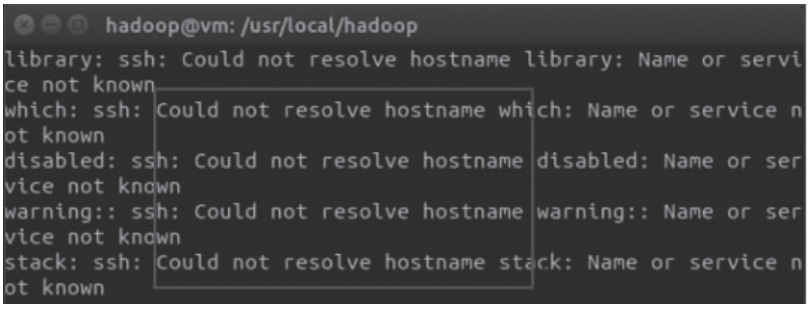

图 1-3　Hadoop 启动后的错误提示信息

笔记

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：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,loggingto/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，应该查看这个后缀为 .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 JavaException 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：这会删除 HDFS 中原有的所有数据，如果原有的数据很重要，请不要这样做，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：这会删除 HDFS 中原有的所有数据
$./bin/hdfsnamenode-format   # 重新格式化名称节点
$./sbin/start-dfs.sh  # 重启
```

笔记

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“ localhost:50070”（见图 1-4 ）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。

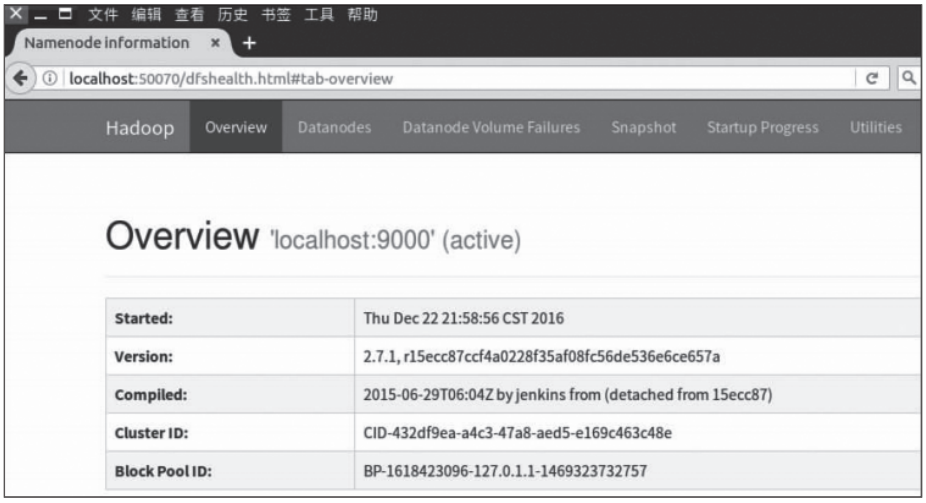

图 1-4　HDFS 的 Web 管理界面

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/hdfsdfs-mkdir-'/user/hadoop
```

上面的命令是分布式文件系统 HDFS 的操作命令，会在后面的章节中做详细介绍，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/hdfsdfs-mkdir input# 在 HDFS 中创建 hadoop 用户对应的 input 目录
$./bin/hdfsdfs-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/hdfsdfs -ls input
```

执行上述命令以后，可以看到 input 目录下的文件信息（见图 1-5 ）。现在就可以运行 Hadoop 自带的 grep 程序。

```
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 1-5　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.File Already Exists Exception: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，可以执行如下命令：

```
$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，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 1s 命令时，系统是根据 PATH 这个环境变量中包含的目录位置，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：

```
$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。

1.2.4 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

笔记

笔记

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。

为了降低分布式模式部署的难度，本书简单地使用两个节点（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 上启动 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，这里不再详述，下面主要介绍步骤（5）(6）。

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，才能实现多个节点互连，如图 1-6 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 冲突。

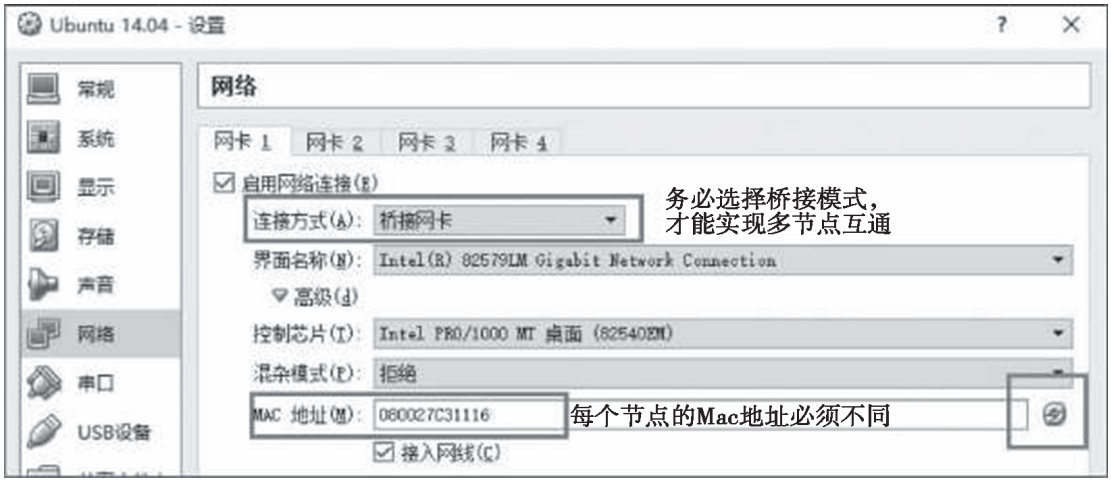

图 1-6　网络连接方式设置

网络配置完成以后，可以查看一下机器的 IP 地址，可以使用 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，比较容易区分当前是对哪台机器进行操作。在 Ubuntu 中，在 Master 节点上执行

如下命令修改主机名：

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

笔记

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。例如，若安装系统时设置的主机名是 dblab-VirtualBox，则打开这个文件以后，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）。然后保存并退出 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，不会与 Slave 节点产生混淆。

然后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 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 1-7 所示：

图 1-7　修改 IP 和主机名映射关系后的效果

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

笔记

加如下两条 IP 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。然后，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。

```
$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 通的话，会显示如图 1-8 所示的结果。

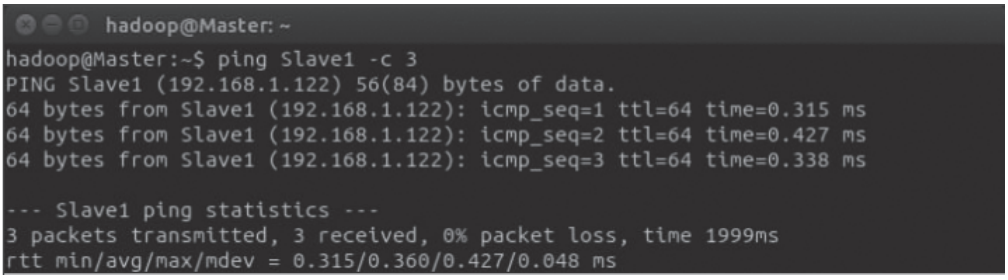

图 1-8　使用 ping 命令的效果

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 节点的公钥，如果之前已经生成过公钥，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。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*# 删除之前生成的公钥（如果已经存在）
$ssh-keygen-trsa    # 执行该命令后，遇到提示信息，一直按 Etner 键就可以
```

为了让 Master 节点能够无密码 SSH 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 Master 节点上将公钥传输到 Slavel 节点：

```
scp-/.ssh/id_rsa.pubhadoop@Slave1:/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 Slavel 上 hadoop 用户的密码，输入完成后会提示传输完毕，如图 1-9 所示。

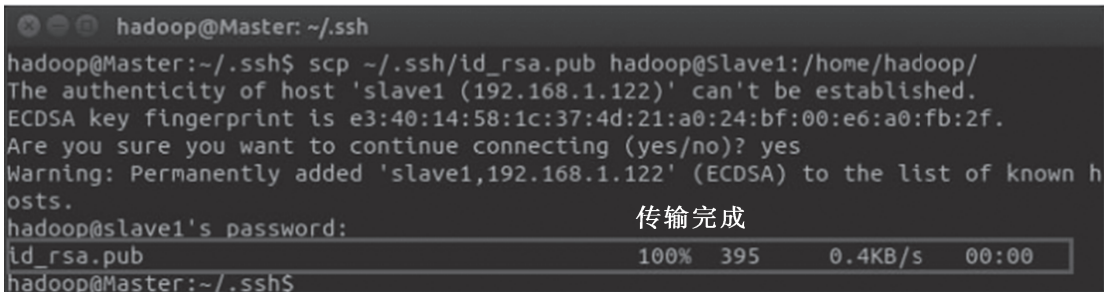

图 1-9　执行 scp 命令的效果

接着在 Slavel 节点上将 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
```

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

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

```
$ssh Slavel
```

执行该命令的效果如图 1-10 所示。

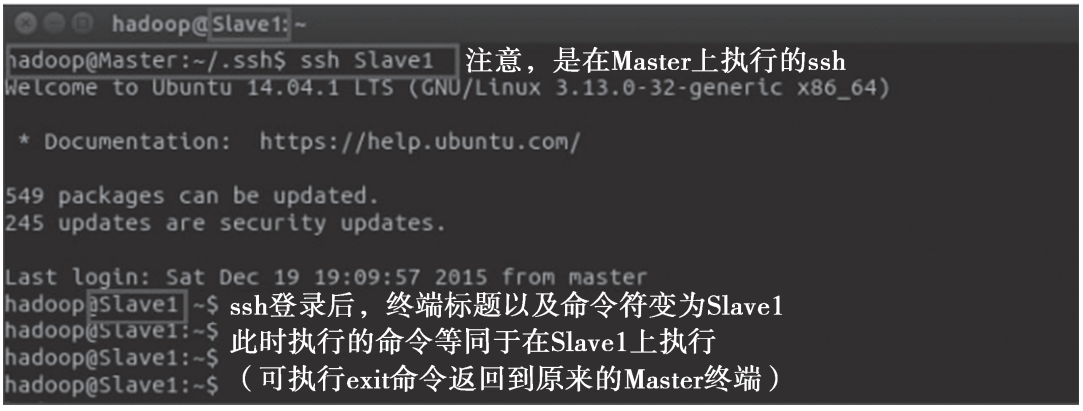

图 1-10　ssh 命令执行效果

3. 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 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。

笔记

笔记

1.3 Hadoop 应用场景

Hadoop 最早来自 Google 的三大论文，Nutch 的开发人员完成了相应的开源实现 HDFS 和 MAPREDUCE，并从 Nutch 中剥离成为独立项目 Hadoop。经过演化，Hadoop 的组件又多出一个 yarn（mapreduce+ yarn + hdfs），而且，Hadoop 外围产生了越来越多的工具组件，形成一个庞大的 Hadoop 生态体系。

在大数据背景下，Apache Hadoop 已经逐渐成为一种标签，业界对于这一开源分布式技术的了解也在不断加深。 Hadoop 的最大应用场景当然是它的“发源地”，像 Google 这样的大型互联网搜索引擎，以及 Yahoo 专门的广告分析系统。Hadoop 平台发挥作用的领域是互联网行业，用来改善分析性能并提高扩展性。其实 Hadoop 的应用场景远不止这一点，深入挖掘的话你会发现 Hadoop 能够在许多地方发挥巨大的作用。

（1）在线旅游。目前全球范围内 80% 的在线旅游网站都是在使用 Cloudera 公司提供的 Hadoop 发行版，其中 SearchBI 网站曾经报道过的 Expedia 也在其中。

（2）移动数据。Cloudera 运营总监称，美国有 70% 的智能手机数据服务背后都是由 Hadoop 来支撑的，也就是说，包括数据的存储以及无线运营商的数据处理等，都是在利用 Hadoop 技术。

（3）电子商务。这一场景应该是非常确定的，eBay 就是最大的实践者之一。国内的电商在 Hadoop 技术上也是储备颇为雄厚的。

（4）能源开采。美国 Chevron 公司是全美第二大石油公司，他们的 IT 部门主管介绍了 Chevron 使用 Hadoop 的经验。他们利用 Hadoop 进行数据的收集和处理，其中这些数据是海洋的地震数据，以便于他们找到油矿的位置。

（5）节能。另外一家能源服务商 Opower 也在使用 Hadoop，为消费者提供节约电费的服务，其中对用户电费单进行了预测分析。

（6）基础架构管理。这是一个非常基础的应用场景，用户可以用 Hadoop 从服务器、交换机以及其他的设备中收集并分析数据。

（7）图像处理。创业公司 Skybox Imaging 使用 Hadoop 来存储并处理图片数据，从卫星中拍摄的高清图像中探测地理变化。

（8）诈骗检测。这个场景用户接触得比较少，一般金融服务或者政府机构会用到。利用 Hadoop 来存储所有的客户交易数据，包括一些非结构化的数据，能够帮助机构发现客户的异常活动，预防欺诈行为。

（9）IT 安全。除企业 IT 基础机构的管理之外，Hadoop 还可以用来处理机器生成数据以便甄别来自恶意软件或者网络中的攻击。

（10）医疗保健。医疗行业也会用到 Hadoop，像 IBM 的 Watson 就会使用 Hadoop 集群作为其服务的基础，包括语义分析等高级分析技术。医疗机构可以利用语义分析为患者提供医护人员，并协助医生更好地为患者进行诊断。