**环境**

本教程使用 **Ubuntu 18.04 64位** 作为系统环境（或者Ubuntu 14.04，Ubuntu16.04 也行，32位、64位均可），请自行安装系统（可参考[使用VirtualBox安装Ubuntu](http://dblab.xmu.edu.cn/blog/2760-2/" \t "_blank)）。

装好了 Ubuntu 系统之后，在安装 Hadoop 前还需要做一些必备工作。

**创建hadoop用户**

如果你安装 Ubuntu 的时候不是用的 “hadoop” 用户，那么需要增加一个名为 hadoop 的用户。

首先按 **ctrl+alt+t** 打开终端窗口，输入如下命令创建新用户 :

1. sudo useradd -m hadoop -s /bin/bash

Shell 命令

这条命令创建了可以登陆的 hadoop 用户，并使用 /bin/bash 作为 shell。

**sudo命令**

本文中会大量使用到sudo命令。sudo是ubuntu中一种权限管理机制，管理员可以授权给一些普通用户去执行一些需要root权限执行的操作。当使用sudo命令时，就需要输入您当前用户的密码.

**密码**

在Linux的终端中输入密码，终端是不会显示任何你当前输入的密码，也不会提示你已经输入了多少字符密码。而在windows系统中,输入密码一般都会以“\*”表示你输入的密码字符

**输入法中英文切换**

ubuntu中终端输入的命令一般都是使用英文输入。linux中英文的切换方式是使用键盘“shift”键来切换，也可以点击顶部菜单的输入法按钮进行切换。ubuntu自带的Sunpinyin中文输入法已经足够读者使用。

**Ubuntu终端复制粘贴快捷键**

在Ubuntu终端窗口中，复制粘贴的快捷键需要加上 shift，即粘贴是 ctrl+shift+v。

接着使用如下命令设置密码，可简单设置为 hadoop，按提示输入两次密码：

1. sudo passwd hadoop

Shell 命令

可为 hadoop 用户增加管理员权限，方便部署，避免一些对新手来说比较棘手的权限问题：

1. sudo adduser hadoop sudo

Shell 命令

最后注销当前用户（点击屏幕右上角的齿轮，选择注销），返回登陆界面。在登陆界面中选择刚创建的 hadoop 用户进行登陆。

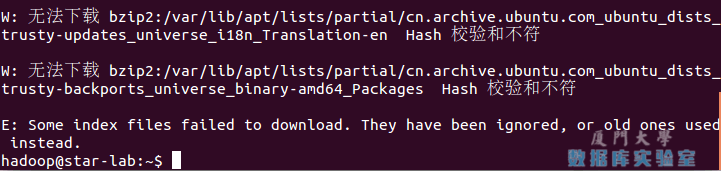
**更新apt**

用 hadoop 用户登录后，我们先更新一下 apt，后续我们使用 apt 安装软件，如果没更新可能有一些软件安装不了。按 ctrl+alt+t 打开终端窗口，执行如下命令：

1. sudo apt-get update

Shell 命令

若出现如下 “Hash校验和不符” 的提示，可通过更改软件源来解决。若没有该问题，则不需要更改。从软件源下载某些软件的过程中，可能由于网络方面的原因出现没法下载的情况，那么建议更改软件源。在学习Hadoop过程中，即使出现“Hash校验和不符”的提示，也不会影响Hadoop的安装。

Ubuntu更新软件源时遇到Hash校验和不符的问题

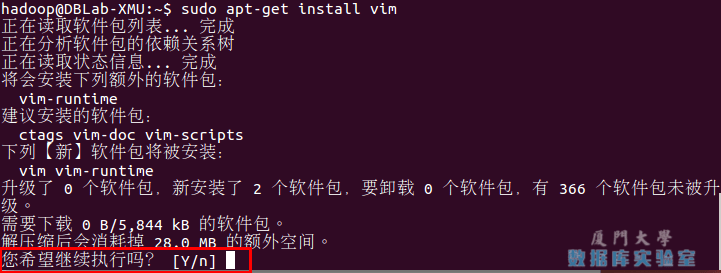
**点击查看：如何更改软件源**

后续需要更改一些配置文件，我比较喜欢用的是 vim（vi增强版，基本用法相同），建议安装一下（如果你实在还不会用 vi/vim 的，请将后面用到 vim 的地方改为 gedit，这样可以使用文本编辑器进行修改，并且每次文件更改完成后请关闭整个 gedit 程序，否则会占用终端）：

1. sudo apt-get install vim

Shell 命令

安装软件时若需要确认，在提示处输入 y 即可。

通过命令行安装软件

**点击查看：vim简单操作指南**

**安装SSH、配置SSH无密码登陆**

集群、单节点模式都需要用到 SSH 登陆（类似于远程登陆，你可以登录某台 Linux 主机，并且在上面运行命令），Ubuntu 默认已安装了 SSH client，此外还需要安装 SSH server：

1. sudo apt-get install openssh-server

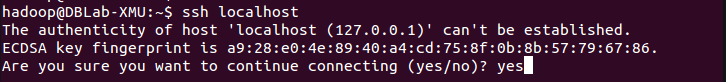
Shell 命令

安装后，可以使用如下命令登陆本机：

1. ssh localhost

Shell 命令

此时会有如下提示(SSH首次登陆提示)，输入 yes 。然后按提示输入密码 hadoop，这样就登陆到本机了。

SSH首次登陆提示

但这样登陆是需要每次输入密码的，我们需要配置成SSH无密码登陆比较方便。

首先退出刚才的 ssh，就回到了我们原先的终端窗口，然后利用 ssh-keygen 生成密钥，并将密钥加入到授权中：

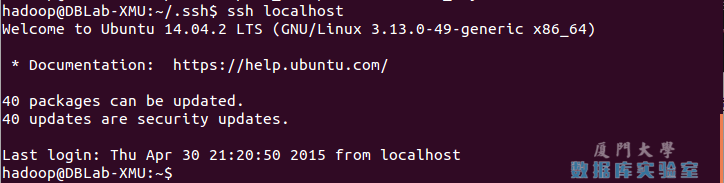
1. exit *# 退出刚才的 ssh localhost*
2. cd ~/.ssh/ *# 若没有该目录，请先执行一次ssh localhost*
3. ssh-keygen -t rsa *# 会有提示，都按回车就可以*
4. cat ./id\_rsa.pub >> ./authorized\_keys *# 加入授权*

Shell 命令

**~的含义**

在 Linux 系统中，~ 代表的是用户的主文件夹，即 “/home/用户名” 这个目录，如你的用户名为 hadoop，则 ~ 就代表 “/home/hadoop/”。 此外，命令中的 # 后面的文字是注释，只需要输入前面命令即可。

此时再用 ssh localhost 命令，无需输入密码就可以直接登陆了，如下图所示。

SSH无密码登录

**安装Java环境**

手动安装，推荐采用本方式

Hadoop3.1.3需要JDK版本在1.8及以上。需要按照下面步骤来自己手动安装JDK1.8。  
我们已经把JDK1.8的安装包jdk-8u162-linux-x64.tar.gz放在了百度云盘，[可以点击这里到百度云盘下载JDK1.8安装包](https://pan.baidu.com/s/1lLjW9cmS1tkBHkrqtpkjWw" \t "_blank)（提取码：ziyu）。请把压缩格式的文件jdk-8u162-linux-x64.tar.gz下载到本地电脑，假设保存在“/home/linziyu/Downloads/”目录下。

在Linux命令行界面中，执行如下Shell命令（注意：当前登录用户名是hadoop）：

1. cd /usr/lib
2. sudo mkdir jvm *#创建/usr/lib/jvm目录用来存放JDK文件*
3. cd ~ *#进入hadoop用户的主目录*
4. cd Downloads *#注意区分大小写字母，刚才已经通过FTP软件把JDK安装包jdk-8u162-linux-x64.tar.gz上传到该目录下*
5. sudo tar -zxvf ./jdk-8u162-linux-x64.tar.gz -C /usr/lib/jvm *#把JDK文件解压到/usr/lib/jvm目录下*

Shell 命令

上面使用了解压缩命令tar，如果对Linux命令不熟悉，可以参考[常用的Linux命令用法](http://dblab.xmu.edu.cn/blog/1624-2/" \t "_blank)。

JDK文件解压缩以后，可以执行如下命令到/usr/lib/jvm目录查看一下：

1. cd /usr/lib/jvm
2. ls

Shell 命令

可以看到，在/usr/lib/jvm目录下有个jdk1.8.0\_162目录。  
下面继续执行如下命令，设置环境变量：

1. cd ~
2. vim ~/.bashrc

Shell 命令

上面命令使用vim编辑器（[查看vim编辑器使用方法](http://dblab.xmu.edu.cn/blog/1607-2/)）打开了hadoop这个用户的环境变量配置文件，请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\_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，继续执行如下命令让.bashrc文件的配置立即生效：

1. source ~/.bashrc

Shell 命令

这时，可以使用如下命令查看是否安装成功：

1. java -version

Shell 命令

如果能够在屏幕上返回如下信息，则说明安装成功：

hadoop@ubuntu:~$ java -version

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。

**安装 Hadoop3.1.3**

Hadoop安装文件，可以到[Hadoop官网](https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.1.3/hadoop-3.1.3.tar.gz" \t "_blank)下载hadoop-3.1.3.tar.gz。  
也可以直接[点击这里从百度云盘下载软件](https://pan.baidu.com/s/1gbmPBXrJDCxwqPGkfvX5Xg" \t "_blank)（提取码：lnwl），进入百度网盘后，进入“软件”目录，找到hadoop-3.1.3.tar.gz文件，下载到本地。  
我们选择将 Hadoop 安装至 /usr/local/ 中：

1. sudo tar -zxvf ~/hadoop-3.1.3.tar.gz -C /usr/local *# 解压到/usr/local中*
2. cd /usr/local/
3. sudo mv ./hadoop-3.1.3/ ./hadoop *# 将文件夹名改为hadoop*
4. sudo chown -R hadoop ./hadoop *# 修改文件权限*

Shell 命令

Hadoop 解压后即可使用。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：

1. cd /usr/local/hadoop
2. ./bin/hadoop version

Shell 命令

**相对路径与绝对路径**

请务必注意命令中的相对路径与绝对路径，本文后续出现的 ./bin/...，./etc/... 等包含 ./ 的路径，均为相对路径，以 /usr/local/hadoop 为当前目录。例如在 /usr/local/hadoop 目录中执行 ./bin/hadoop version 等同于执行 /usr/local/hadoop/bin/hadoop version。可以将相对路径改成绝对路径来执行，但如果你是在主文件夹 ~ 中执行 ./bin/hadoop version，执行的会是 /home/hadoop/bin/hadoop version，就不是我们所想要的了。

**Hadoop单机配置(非分布式)**

Hadoop 默认模式为非分布式模式（本地模式），无需进行其他配置即可运行。非分布式即单 Java 进程，方便进行调试。

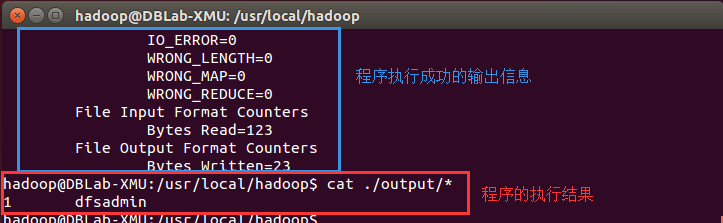
现在我们可以执行例子来感受下 Hadoop 的运行。Hadoop 附带了丰富的例子（运行 ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar 可以看到所有例子），包括 wordcount、terasort、join、grep 等。

在此我们选择运行 grep 例子，我们将 input 文件夹中的所有文件作为输入，筛选当中符合正则表达式 dfs[a-z.]+ 的单词并统计出现的次数，最后输出结果到 output 文件夹中。

1. cd /usr/local/hadoop
2. mkdir ./input
3. cp ./etc/hadoop/\*.xml ./input *# 将配置文件作为输入文件*
4. ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar grep ./input ./output 'dfs[a-z.]+'
5. cat ./output/\* *# 查看运行结果*

Shell 命令

执行成功后如下所示，输出了作业的相关信息，输出的结果是符合正则的单词 dfsadmin 出现了1次

Hadoop单机模式运行grep的输出结果

**注意**，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，需要先将 ./output 删除。

1. rm -r ./output

Shell 命令

**Hadoop伪分布式配置**

Hadoop 可以在单节点上以伪分布式的方式运行，Hadoop 进程以分离的 Java 进程来运行，节点既作为 NameNode 也作为 DataNode，同时，读取的是 HDFS 中的文件。

Hadoop 的配置文件位于 /usr/local/hadoop/etc/hadoop/ 中，伪分布式需要修改2个配置文件 **core-site.xml** 和 **hdfs-site.xml** 。Hadoop的配置文件是 xml 格式，每个配置以声明 property 的 name 和 value 的方式来实现。

修改配置文件 **core-site.xml** (通过 gedit 编辑会比较方便: gedit ./etc/hadoop/core-site.xml)，将当中的

1. <configuration>
2. </configuration>

XML

修改为下面配置：

1. <configuration>
2. <property>
3. <name>hadoop.tmp.dir</name>
4. <value>file:/usr/local/hadoop/tmp</value>
5. <description>Abase for other temporary directories.</description>
6. </property>
7. <property>
8. <name>fs.defaultFS</name>
9. <value>hdfs://localhost:9000</value>
10. </property>
11. </configuration>

XML

同样的，修改配置文件 **hdfs-site.xml**：

1. <configuration>
2. <property>
3. <name>dfs.replication</name>
4. <value>1</value>
5. </property>
6. <property>
7. <name>dfs.namenode.name.dir</name>
8. <value>file:/usr/local/hadoop/tmp/dfs/name</value>
9. </property>
10. <property>
11. <name>dfs.datanode.data.dir</name>
12. <value>file:/usr/local/hadoop/tmp/dfs/data</value>
13. </property>
14. </configuration>

XML

**Hadoop配置文件说明**

Hadoop 的运行方式是由配置文件决定的（运行 Hadoop 时会读取配置文件），因此如果需要从伪分布式模式切换回非分布式模式，需要删除 core-site.xml 中的配置项。

此外，伪分布式虽然只需要配置 fs.defaultFS 和 dfs.replication 就可以运行（官方教程如此），不过若没有配置 hadoop.tmp.dir 参数，则默认使用的临时目录为 /tmp/hadoo-hadoop，而这个目录在重启时有可能被系统清理掉，导致必须重新执行 format 才行。所以我们进行了设置，同时也指定 dfs.namenode.name.dir 和 dfs.datanode.data.dir，否则在接下来的步骤中可能会出错。

配置完成后，执行 NameNode 的格式化:

1. cd /usr/local/hadoop
2. ./bin/hdfs namenode -format

Shell 命令

成功的话，会看到 “successfully formatted” 的提示，具体返回信息类似如下：

2020-01-08 15:31:31,560 INFO namenode.NameNode: STARTUP\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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 directory /usr/local/hadoop/tmp/dfs/name \*\*has been successfully formatted\*\*.

2020-01-08 15:31:35,700 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt\_0000000000000000000 using no compression

2020-01-08 15:31:35,770 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt\_0000000000000000000 of size 393 bytes saved in 0 seconds .

2020-01-08 15:31:35,810 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0

2020-01-08 15:31:35,816 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid = 0 when meet shutdown.

2020-01-08 15:31:35,816 INFO namenode.NameNode: SHUTDOWN\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SHUTDOWN\_MSG: Shutting down NameNode at hadoop/127.0.1.1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

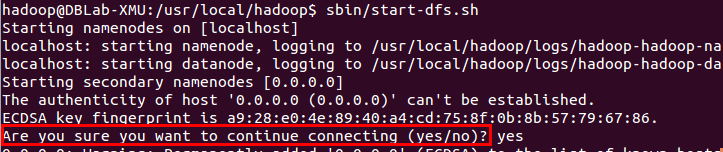
如果在这一步时提示 **Error: JAVA\_HOME is not set and could not be found.** 的错误，则说明之前设置 JAVA\_HOME 环境变量那边就没设置好，请按教程先设置好 JAVA\_HOME 变量，否则后面的过程都是进行不下去的。如果已经按照前面教程在.bashrc文件中设置了JAVA\_HOME，还是出现 **Error: JAVA\_HOME is not set and could not be found.** 的错误，那么，请到hadoop的安装目录修改配置文件“/usr/local/hadoop/etc/hadoop/hadoop-env.sh”，在里面找到“export JAVA\_HOME=${JAVA\_HOME}”这行，然后，把它修改成JAVA安装路径的具体地址，比如，“export JAVA\_HOME=/usr/lib/jvm/default-java”，然后，再次启动Hadoop。

接着开启 NameNode 和 DataNode 守护进程。

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

Shell 命令

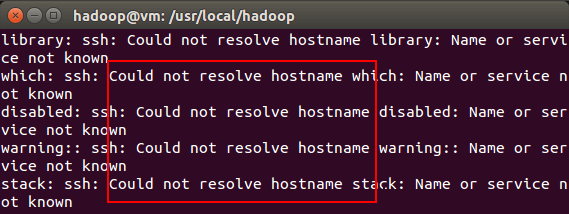
若出现如下SSH提示，输入yes即可。

启动Hadoop时的SSH提示

启动时可能会出现如下 WARN 提示：WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform… using builtin-java classes where applicable WARN 提示可以忽略，并不会影响正常使用。

**启动 Hadoop 时提示 Could not resolve hostname**

如果启动 Hadoop 时遇到输出非常多“ssh: Could not resolve hostname xxx”的异常情况，如下图所示：

启动Hadoop时的异常提示

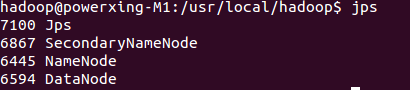
这个并不是 ssh 的问题，可通过设置 Hadoop 环境变量来解决。首先按键盘的 **ctrl + c** 中断启动，然后在 ~/.bashrc 中，增加如下两行内容（设置过程与 JAVA\_HOME 变量一样，其中 HADOOP\_HOME 为 Hadoop 的安装目录）：

1. export HADOOP\_HOME=/usr/local/hadoop
2. export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

Shell

保存后，务必执行 source ~/.bashrc 使变量设置生效，然后再次执行 ./sbin/start-dfs.sh 启动 Hadoop。

启动完成后，可以通过命令 jps 来判断是否成功启动，若成功启动则会列出如下进程: “NameNode”、”DataNode” 和 “SecondaryNameNode”（如果 SecondaryNameNode 没有启动，请运行 sbin/stop-dfs.sh 关闭进程，然后再次尝试启动尝试）。如果没有 NameNode 或 DataNode ，那就是配置不成功，请仔细检查之前步骤，或通过查看启动日志排查原因。

通过jps查看启动的Hadoop进程

**Hadoop无法正常启动的解决方法**

一般可以查看启动日志来排查原因，注意几点：

* 启动时会提示形如 “DBLab-XMU: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-DBLab-XMU.out”，其中 DBLab-XMU 对应你的机器名，但其实启动日志信息是记录在 /usr/local/hadoop/logs/hadoop-hadoop-namenode-DBLab-XMU.log 中，所以应该查看这个后缀为 **.log** 的文件；
* 每一次的启动日志都是追加在日志文件之后，所以得拉到最后面看，对比下记录的时间就知道了。
* 一般出错的提示在最后面，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，看能否找到一些相关的解决方法。

此外，**若是 DataNode 没有启动**，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）：

1. *# 针对 DataNode 没法启动的解决方法*
2. cd /usr/local/hadoop
3. ./sbin/stop-dfs.sh *# 关闭*
4. rm -r ./tmp *# 删除 tmp 文件，注意这会删除 HDFS 中原有的所有数据*
5. ./bin/hdfs namenode -format *# 重新格式化 NameNode*
6. ./sbin/start-dfs.sh *# 重启*

Shell 命令

成功启动后，可以访问 Web 界面 [http://localhost:9870](http://localhost:9870/) 查看 NameNode 和 Datanode 信息，还可以在线查看 HDFS 中的文件。

**运行Hadoop伪分布式实例**

上面的单机模式，grep 例子读取的是本地数据，伪分布式读取的则是 HDFS 上的数据。要使用 HDFS，首先需要在 HDFS 中创建用户目录：

1. ./bin/hdfs dfs -mkdir -p /user/hadoop

Shell 命令

**注意**

教材《大数据技术原理与应用》的命令是以”./bin/hadoop dfs”开头的Shell命令方式，实际上有三种shell命令方式。  
1. hadoop fs  
2. hadoop dfs  
3. hdfs dfs

hadoop fs适用于任何不同的文件系统，比如本地文件系统和HDFS文件系统  
hadoop dfs只能适用于HDFS文件系统  
hdfs dfs跟hadoop dfs的命令作用一样，也只能适用于HDFS文件系统

接着将 ./etc/hadoop 中的 xml 文件作为输入文件复制到分布式文件系统中，即将 /usr/local/hadoop/etc/hadoop 复制到分布式文件系统中的 /user/hadoop/input 中。我们使用的是 hadoop 用户，并且已创建相应的用户目录 /user/hadoop ，因此在命令中就可以使用相对路径如 input，其对应的绝对路径就是 /user/hadoop/input:

1. ./bin/hdfs dfs -mkdir input
2. ./bin/hdfs dfs -put ./etc/hadoop/\*.xml input

Shell 命令

复制完成后，可以通过如下命令查看文件列表：

1. ./bin/hdfs dfs -ls input

Shell 命令

伪分布式运行 MapReduce 作业的方式跟单机模式相同，区别在于伪分布式读取的是HDFS中的文件（可以将单机步骤中创建的本地 input 文件夹，输出结果 output 文件夹都删掉来验证这一点）。

1. ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar grep input output 'dfs[a-z.]+'

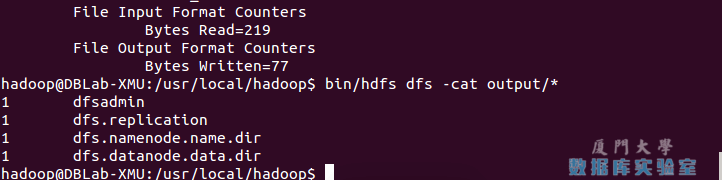
Shell 命令

查看运行结果的命令（查看的是位于 HDFS 中的输出结果）：

1. ./bin/hdfs dfs -cat output/\*

Shell 命令

结果如下，注意到刚才我们已经更改了配置文件，所以运行结果不同。

Hadoop伪分布式运行grep结果

我们也可以将运行结果取回到本地：

1. rm -r ./output *# 先删除本地的 output 文件夹（如果存在）*
2. ./bin/hdfs dfs -get output ./output *# 将 HDFS 上的 output 文件夹拷贝到本机*
3. cat ./output/\*

Shell 命令

Hadoop 运行程序时，输出目录不能存在，否则会提示错误 “org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://localhost:9000/user/hadoop/output already exists” ，因此若要再次执行，需要执行如下命令删除 output 文件夹:

1. ./bin/hdfs dfs -rm -r output *# 删除 output 文件夹*

Shell 命令

**运行程序时，输出目录不能存在**

运行 Hadoop 程序时，为了防止覆盖结果，程序指定的输出目录（如 output）不能存在，否则会提示错误，因此运行前需要先删除输出目录。在实际开发应用程序时，可考虑在程序中加上如下代码，能在每次运行时自动删除输出目录，避免繁琐的命令行操作：

1. Configuration conf = new Configuration();
2. Job job = new Job(conf);
4. */\* 删除输出目录 \*/*
5. Path outputPath = new Path(args[1]);
6. outputPath.getFileSystem(conf).delete(outputPath, true);

Java

若要关闭 Hadoop，则运行

1. ./sbin/stop-dfs.sh

Shell 命令

**注意**

下次启动 hadoop 时，无需进行 NameNode 的初始化，只需要运行 ./sbin/start-dfs.sh 就可以！