

大数据管理

Big Data Management



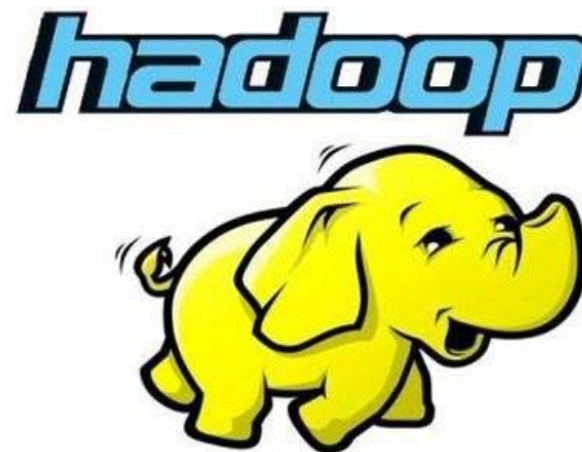
张海腾

htzhang@ecust.edu.cn

第2章 大数据处理架构Hadoop



- 2.1 概述
- 2.2 Hadoop生态系统
- 2.3 Hadoop的安装和使用
- 2.4本章小结



2.1 概述~2.1.1 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HDFS（Hadoop Distributed File System）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、雅虎、微软、思科、淘宝等，都支持Hadoop

2.1.2 Hadoop的发展简史

- Hadoop最初是由Apache Lucene项目的创始人Doug Cutting开发的文本搜索库。Hadoop源自始于2002年的**Apache Nutch项目**——一个开源的网络搜索引擎（也是Lucene项目的一部分）
- 2003年，谷歌公司发布了**分布式文件系统GFS**方面的论文
- 在2004年，Nutch项目也模仿GFS开发了自己的分布式文件系统NDFS（Nutch Distributed File System）（即**HDFS**的前身）
- 2004年，谷歌公司又发表了另一篇的论文，阐述了**MapReduce分布式编程思想**（具有深远的影响）
- 2005年，Nutch开源**实现了谷歌的MapReduce**

2.1.2 Hadoop的发展简史

□到了2006年2月，Nutch中的NDFS和MapReduce开始**独立出来**，成为Lucene项目的一个子项目，称为**Hadoop**，同时，Doug Cutting加盟雅虎



□ 2008年1月，Hadoop正式成为Apache顶级项目，Hadoop也逐渐开始被雅虎之外的其他公司使用

□ 2008年4月，Hadoop打破世界纪录，成为**最快排序**1TB数据的系统，它采用一个由910个节点构成的**集群**进行运算，排序时间只用了209秒

□在2009年5月，Hadoop把1TB数据排序时间缩短到62秒。Hadoop从此迅速发展成为**大数据时代最具影响力的开源分布式开发平台**，并**成为事实上的大数据处理标准**

2.1.3 Hadoop的特性

Hadoop是一个能够对大量数据进行分布式处理的软件框架，并且是以一种可靠、高效、可伸缩的方式进行处理的，它具有以下几个方面的特性：

- 高可靠性
- 高效性
- 高可扩展性
- 高容错性
- 成本低
- 运行在Linux平台上
- 支持多种编程语言

2.1.3 Hadoop的特性

- (1) **高可靠性**：采用**冗余数据存储方式**，即使一个副本发生故障，其他副本也可以保证正常对外提供服务；
- (2) **高效性**：作为并行分布式计算平台，Hadoop采用**分布式存储和分布式处理两大核心技术**，能够高效地处理PB级数据；
- (3) **高可扩展性**：Hadoop的设计目标是可以高效稳定地**运行在廉价的计算机集群上**，可以扩展到数以千计的计算机节点上；
- (4) **高容错性**：**采用冗余数据存储方式**，自动保存数据的多个副本，并且能够自动将失败的任务进行重新分配；

2.1.3 Hadoop的特性

- (5) 成本低：Hadoop采用廉价的计算机集群，成本比较低，普通用户也可以用自己的PC搭建Hadoop运行环境；
- (6) 运行在Linux平台上：Hadoop是基于Java语言开发的，可以较好地运行在Linux平台上；
- (7) 支持多语言编程：Hadoop上的应用程序也可以使用其他语言编写。

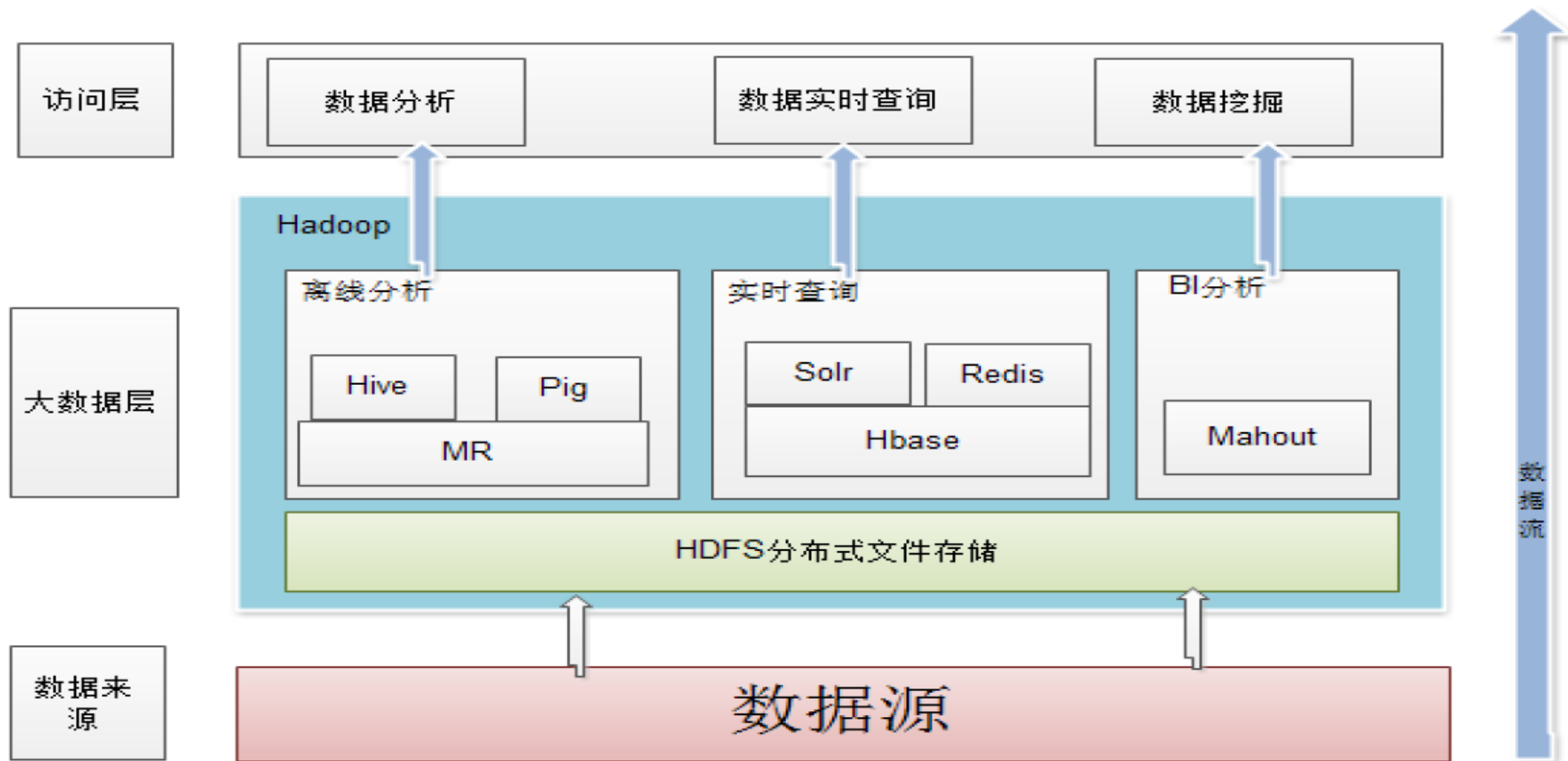
2.1.4 Hadoop的应用现状

□Hadoop凭借其突出的优势，**已经在各个领域得到了广泛的应用**，而互联网领域是其应用的主阵地：

- 2007年，**雅虎**在Sunnyvale总部建立了M45——一个包含了4000个处理器和1.5PB容量的Hadoop集群系统
- **Facebook**作为全球知名的社交网站，Hadoop是非常理想的选择，Facebook主要将Hadoop平台用于日志处理、推荐系统和数据仓库等方面
- **国内采用Hadoop的公司**主要有百度、淘宝、网易、华为、中国移动等，其中，淘宝的Hadoop集群比较大

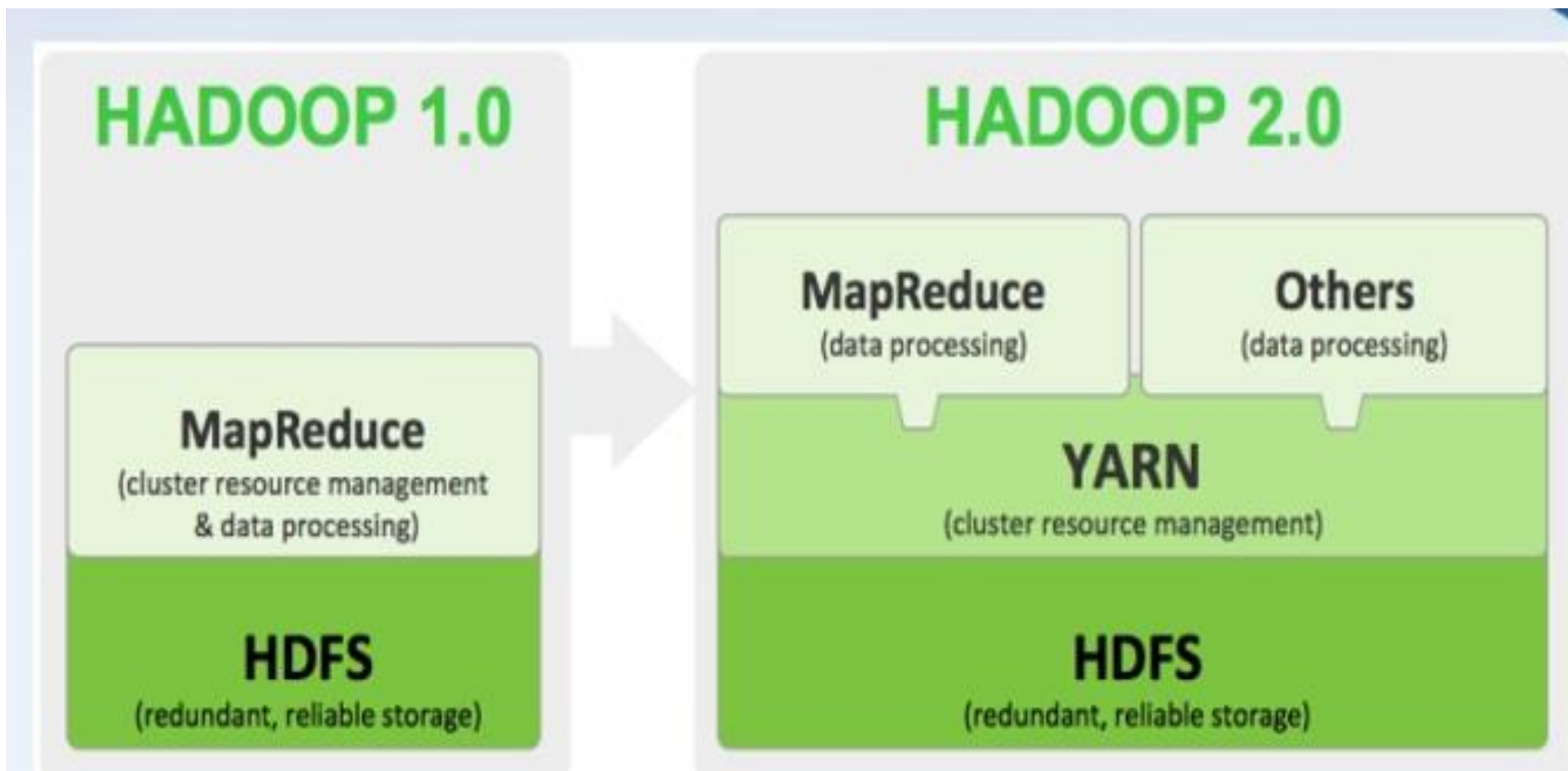
2.1.4 Hadoop的应用现状

□ Hadoop在企业中的应用架构:



2.1.5 Hadoop的版本

- Apache Hadoop版本分为三代，分别是：**Hadoop 1.0**、**Hadoop 2.0**、**Hadoop 3.0**，Hadoop 2.0是一个全新的架构。



2.1.5 Hadoop的版本

厂商名称	开放性	易用性 (★)	平台功能	性能 (★)	本地支持	总体评价 (★)
apache	完全开源、Hadoop就是托管在apache社区里面	安装: 2 使用: 2 维护: 2	Apache是标准的Hadoop平台, 所有厂商都是在apache的平台上进行改进	2	没有	2
cloudera	与Apache功能同步, 部分代码开源	安装: 5 使用: 5 维护: 5	有自主研发的产品如: impala、navigator等	4.5	2014年刚进入中国, 上海	4.5
hortonworks	与apache功能同步, 也是完全开源	安装: 4.5 使用: 5 维护: 5	是apache hadoop平台的最大贡献者, 如Tez	4.5	没有	4.5
MapR	在apache的hadoop版本上面修改很多	安装: 4.5 使用: 5 维护: 5	在apache平台上优化很多、从而形成自己的产品	5	没有	3.5

2.1.5 Hadoop的版本

□选择 Hadoop版本的考虑因素：

- 是否开源（即是否免费）
- 是否有稳定版
- 是否经实践检验
- 是否有强大的社区支持

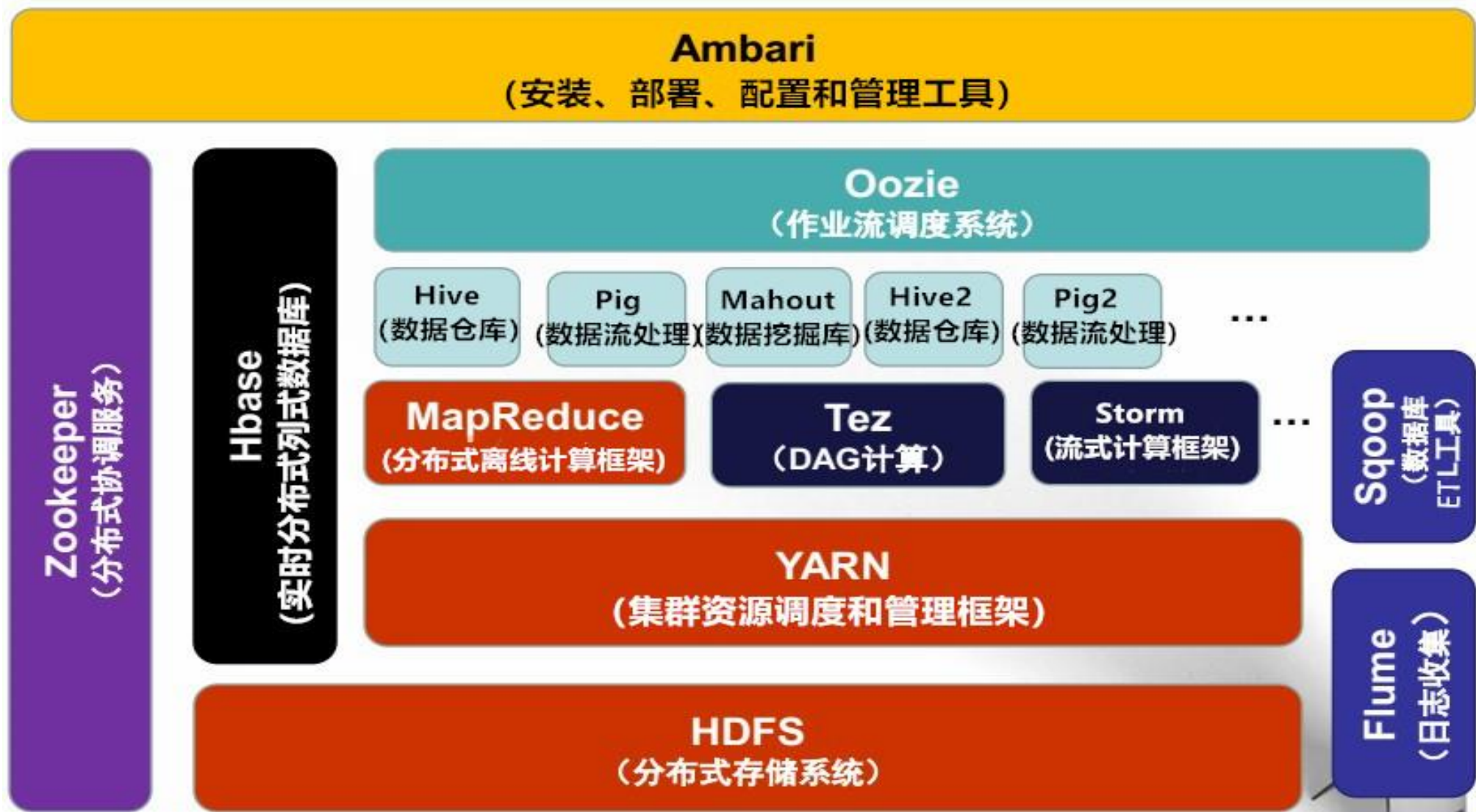
2.2 Hadoop生态系统

□从狭义上来说：Hadoop就是单独指代Hadoop这个软件，包括：

- **HDFS**：分布式文件系统
- **MapReduce**：分布式计算框架
- **Yarn**：分布式集群资源调度和管理框架
- **Common模块**：支持其他模块的工具模块（配置、RPC、序列化机制、日志操作等）

□从广义上来说：Hadoop指代大数据的一个生态圈，包括很多其他软件

2.2 Hadoop生态系统



2.2 Hadoop生态系统

组件	功能
HDFS	分布式文件系统，提供了高可靠性、高扩展性和高吞吐率的数据存储服务
MapReduce	分布式并行编程模型，具有易于编程、高容错性和高扩展性等优点
YARN	资源调度和管理框架，负责集群资源的统一管理和调度，使多种计算框架可以运行在一个集群中
Tez	运行在YARN之上的下一代Hadoop查询处理框架
Hive	Hadoop上的数据仓库工具，定义了类SQL的查询语言Hive QL，适合数据仓库的统计分析

2.2 Hadoop生态系统

组件	功能
HBase	Hadoop上的 非关系型的实时分布式列式数据库 ，是Google BigTable的开源实现
Pig	一个基于Hadoop的 大规模数据分析平台 ，提供类似SQL的查询语言Pig Latin
Sqoop	SQL-to-Hadoop的缩写，用于 在Hadoop与传统数据库之间进行数据传递
Oozie	Hadoop上的 作业流调度和管理系统
Zookeeper	是Google Chubby的开源实现，高效、可靠的 协同工作系统 ，提供分布式协调一致性服务

2.2 Hadoop生态系统

组件	功能
Storm	流计算框架
Flume	一个高可用的、高可靠的、分布式的海量日志采集、聚合和传输的系统
Ambari	基于Web的快速部署工具，支持Apache Hadoop集群的安装、部署、配置和管理
Kafka	一种高吞吐量的分布式发布订阅消息系统，可以处理消费者规模的网站中的所有动作流数据
Spark	类似于Hadoop MapReduce的通用并行处理框架

2.3 Hadoop的安装和使用

□在Windows中使用VirtualBox安装Ubuntu

一、材料和工具

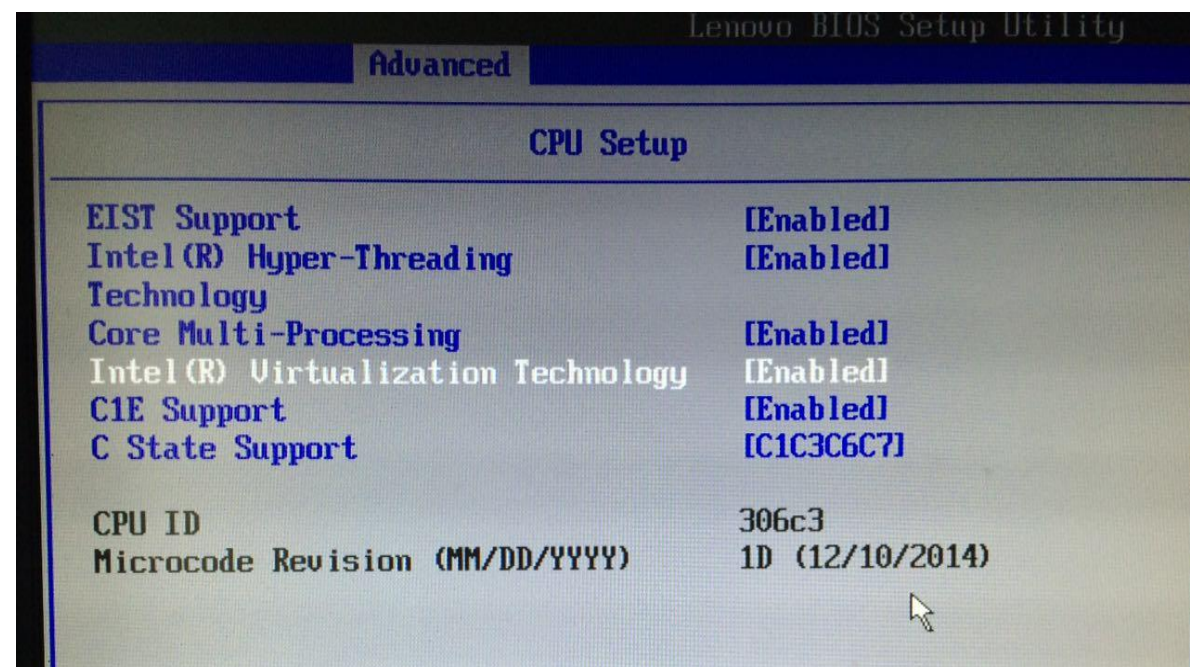
<https://dblab.xmu.edu.cn/blog/337/>

- 1、安装好的VirtualBox虚拟机软件
- 2、Ubuntu LTS 16.04 ISO映像文件

二、步骤

1、确认系统版本

如果选择的系统是64位Ubuntu系统，在安装虚拟机前，进入BIOS开启CPU的虚拟化。



2.3 Hadoop的安装和使用

补充概念：

□ VirtualBox：

VirtualBox 是由德国 Innotek 公司开发，由Sun Microsystems公司出品的软件，使用Qt编写，在 Sun 被 Oracle 收购后正式更名成 Oracle VM VirtualBox。

□ Ubuntu

Ubuntu是一个以桌面应用为主的Linux操作系统，其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词，意思是“人性”“我的存在是因为大家的存在”，是非洲传统的一种价值观。

2.3 Hadoop的安装和使用

2、安装前的准备

- ① 新建一个虚拟机
- ② 给虚拟机命名（名字叫“Ubuntu”），选择操作系统，版本
- ③ 设置虚拟机内存大小：如果计算机内存是8GB，可以设置虚拟机内存为3GB
- ④ 创建虚拟硬盘
- ⑤ 选择虚拟硬盘文件类型VDI
- ⑥ 虚拟硬盘选择动态分配
- ⑦ 选择文件存储的位置和容量大小。建议划分30GB以上

2.3 Hadoop的安装和使用

3、安装Ubuntu

- 1) 选择下载的Ubuntu ISO映像文件
- 2) 进入存储设置界面后，点击没有盘片，再点击光盘按钮，选择一个虚拟光驱，找到已经下载到本地的Ubuntu系统安装镜像文件ubuntukylin-16.04-desktop-amd64.iso，单击“确定”按钮
- 3) 在VirtualBox管理器界面，选择刚创建的虚拟机Ubuntu，单击“启动”按钮
- 4) 在安装过程中，选择“新建分区表”，创建“交换分区”（1GB）和“根分区”
- 5) 设置用户名为“dblab”，密码为“dblab”
- 6) 安装完成，重启，选择“安装增强功能”

2.3 Hadoop的安装和使用

补充概念：

□根分区：

linux根分区是系统分区的意思，系统内所有的东西都存放在根分区中，也被称为root分区；

Linux是一个树形文件系统，根分区就是它的root节点，任何的目录文件都会挂在根节点以下；

并且linux只有一个根，不管对硬盘分多少个区，都要将这些分区挂载到根目录底下才可以使用。

2.3 Hadoop的安装和使用

□swap分区

swap分区是交换分区，是一定磁盘空间（分区或文件），用于将部分内存中的数据换下来，以腾出内存空间用于其他需求。在一个系统中，物理内存快使用完时，操作系统会使用交换分区。

当系统内存紧张时，操作系统根据一定的算法规则，将一部分最近没使用的内存页面保存到交换分区，从而为需要内存的程序留出足够的内存空间；在swap中的内存页面被访问时，系统会将其重新载入到物理内存中去运行。

2.3 Hadoop的安装和使用

□ **virtualbox**虚拟机的增强功能可以实现如下功能：

- 主机与虚拟机之间的文件共享（主要是为了这个）。
- 主机与虚拟机之间的剪切板共享（比如说在主机上复制 <http://wubangtu.com>，然后在虚拟机的浏览器中粘贴）。
- 虚拟机的direct3D支持，就是为虚拟机分配点显存，这样虚拟机窗口就可以随便放大或缩小了（当然前提是你勾选了“自动调节显示尺寸”）。

2.3 Hadoop的安装和使用

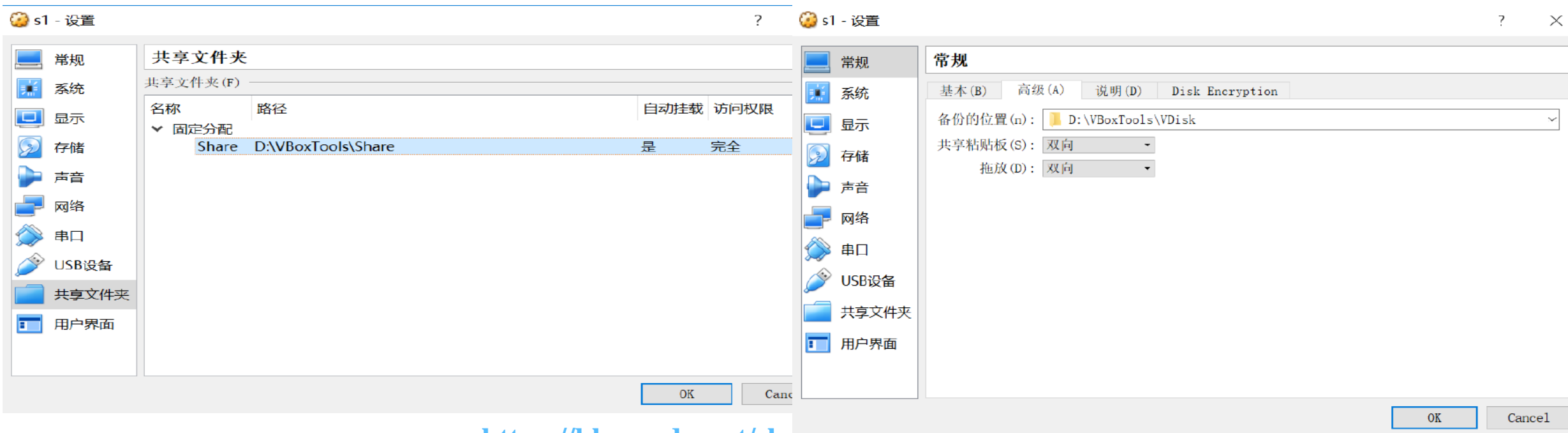
7) 设置分辨率，进行网络配置

8) 设置：共享粘贴版和拖放为：双向

9) 设置共享文件夹，主机和虚拟之间可以共享文件

➤ 没有查看“sf_share”的内容所需权限。

➤ `sudo usermod -a -G vboxsf dlab` //把dlab用户添加到用户组vboxsf中



2.3 Hadoop的安装和使用

4、检查安装配置是否成功

1) 检查网络是否安装配置成功 ifconfig windows下ping命令

```
tl@sl:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:86:1f:ef
        inet addr:192.168.3.129  Bcast:192.168.3.255  Mask:255.255.255.0
        inet6 addr: fe80::b0b7:a28:3e88:4918/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:1437 errors:0 dropped:0 overruns:0 frame:0
        TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:93070 (93.0 KB)  TX bytes:8003 (8.0 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:36 errors:0 dropped:0 overruns:0 frame:0
        TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:2743 (2.7 KB)  TX bytes:2743 (2.7 KB)
```

C:\> 管理员: 命令提示符

C:\>ping 192.168.3.129

正在 Ping 192.168.3.129 具有 32 字节的数据:
来自 192.168.3.129 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.3.129 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.3.129 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.3.129 的回复: 字节=32 时间<1ms TTL=64

192.168.3.129 的 Ping 统计信息:
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
最短 = 0ms, 最长 = 0ms, 平均 = 0ms

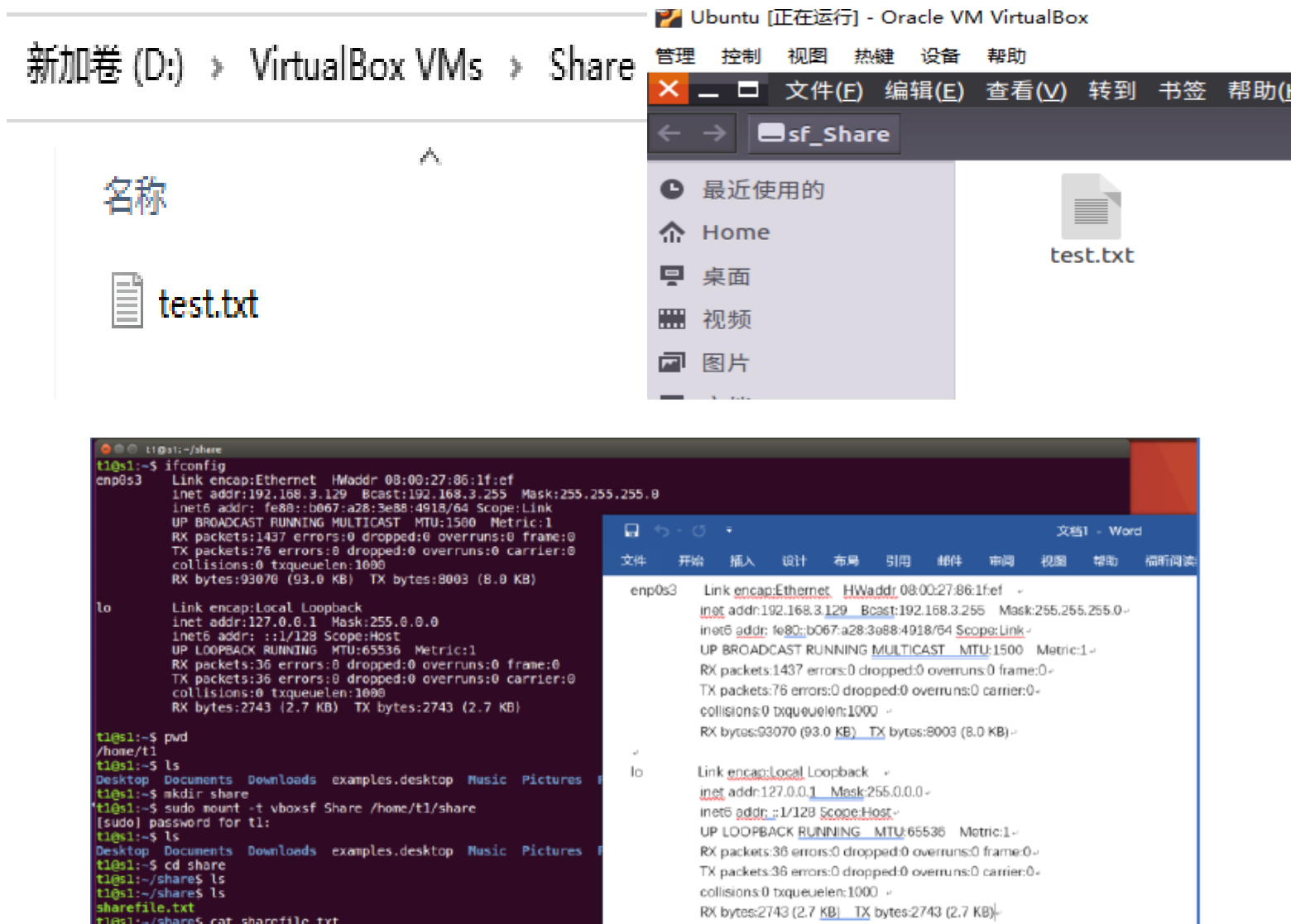
C:\>

2.3 Hadoop的安装和使用

4、检查安装配置是否成功

2) 在windows 拷贝一个 test .txt 文件到共享目录 测试共享文件是否成功

3) 测试复制粘贴功能：
鼠标选中虚拟机终端窗口的部分内容，执行拷贝功能，然后粘贴到 windows word



2.3 Hadoop的安装和使用

□生成Linux虚拟机镜像文件

- 导出生成镜像文件

关闭Ubuntu虚拟机，在VirtualBox管理器的“管理”菜单中选择“导出虚拟电脑”

- 导入Linux虚拟机镜像文件

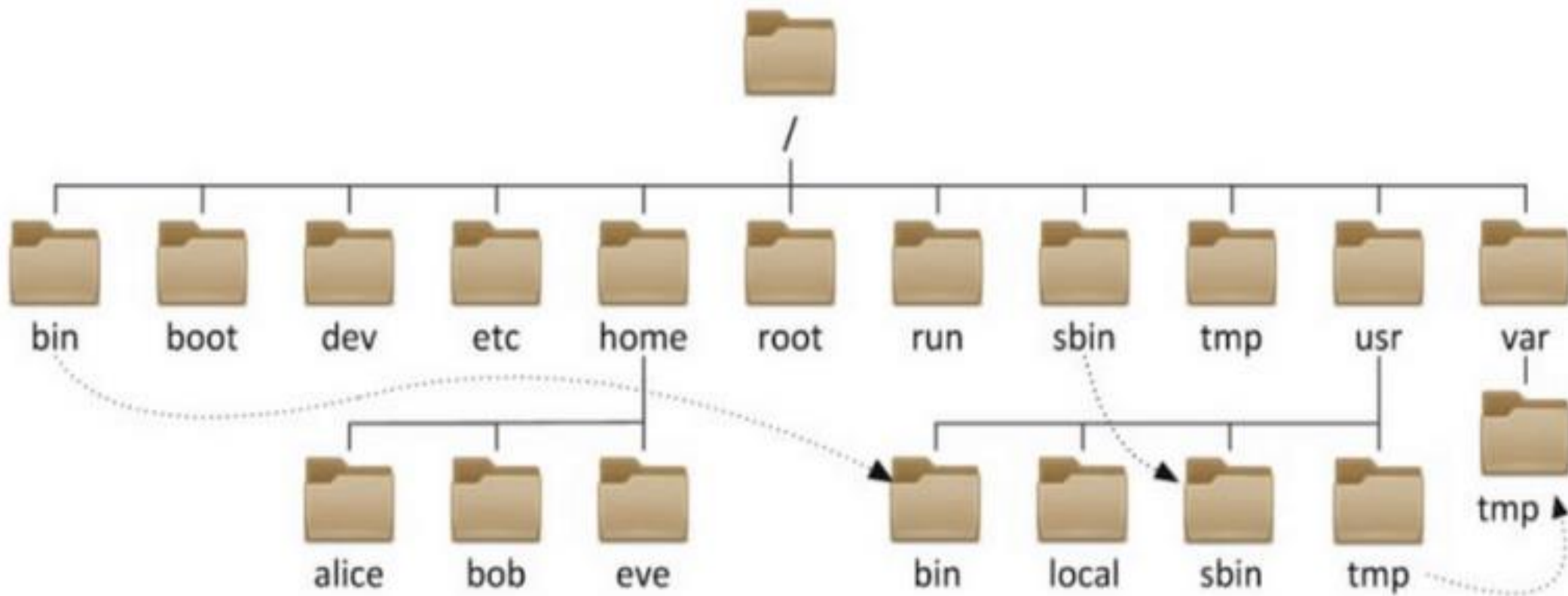
在VirtualBox管理器的“管理”菜单中选择“导入虚拟电脑”



2.3 Hadoop的安装和使用

补充概念：

□Ubuntu的目录结构：



2.3 Hadoop的安装和使用

/

这根目录。对一个电脑来说，有且只有一个根目录。所有的东西都是从这里开始。举个例子：当你在终端里输入“/home”，你其实是在告诉电脑，先从/(根目录)开始，再进入到home目录。

/root

这是系统管理员(root user)的目录。它能对系统做任何事情，甚至包括删除你的文件。因此，请小心使用root帐号。

/bin

这里存放了标准的(或者缺省的)linux的工具，比如像“ls”、“vi”还有“more”等。通常来说，这个目录已经包含在“path”系统变量里面了。当你在终端里输入ls，系统就会去/bin目录下面查找是不是有ls这个程序

2.3 Hadoop的安装和使用

/etc

这里主要存放了系统配置方面的文件。举个例子：你安装了samba这个套件，当你想要修改samba配置文件的时候，你会发现它们(配置文件)就在/etc/samba目录下。

/dev

这里主要存放与设备(包括外设)有关的文件(unix和linux系统均把设备当成文件)。想连线打印机吗?系统就是从这个目录开始工作的。另外还有一些包括磁盘驱动、USB驱动等都放在这个目录。

/home

这里主要存放你的个人数据。具体每个用户的设置文件，用户的桌面文件夹，还有用户的数据都放在这里。每个用户都有自己的用户目录，位置为：
/home/用户名。当然，root用户除外。

2.3 Hadoop的安装和使用

/tmp

这是临时目录。对于某些程序来说，有些文件被用了一次两次之后，就不会再被用到，像这样的文件就放在这里。有些linux系统会定期自动对这个目录进行清理，因此，千万不要把重要的数据放在这里。

/usr

usr是Unix Software Resource的缩写，也就是Unix操作系统软件资源所放置的目录，而不是用户的数据；所有系统默认的软件都会放置到/usr，系统安装完时，这个目录会占用最多的硬盘容量。/usr目录包含了许多子目录：/usr/bin目录用于存放程序；/usr/share用于存放一些共享的数据，比如音乐文件或者图标等等；/usr/lib目录用于存放那些不能直接运行的，但却是许多程序运行所必需的一些函数库文件。

2.3 Hadoop的安装和使用

/usr/local

这个目录一般是用来存放用户自编译安装软件的存放目录；一般是通过源码包安装的软件，如果没有特别指定安装目录的话，一般是安装在这个目录中。

/opt

这里主要存放那些可选的程序。你想尝试最新的firefox测试版吗？那就装到/opt目录下吧，这样，当你尝试完，想删掉firefox的时候，你就可以直接删除它，而不影响系统其他任何设置。安装到/opt目录下的程序，它所有的数据、库文件等等都是放在同个目录下面。

2.3 Hadoop的安装和使用

/var

这个目录的内容是经常变动的，/var下有/var/log 这是用来存放系统日志的目录。/var/www目录是定义Apache服务器站点存放目录；/var/lib用来存放一些库文件，比如MySQL的，以及MySQL数据库的的存放地；

/media

有些linux的发行版使用这个目录来挂载那些usb接口的移动硬盘(包括U盘)、CD/DVD驱动器等等。

2.3 Hadoop的安装和使用

□熟悉常用的Linux命令

- **终端**-----主要任务是接收用户输入的命令和字符，然后提交给shell，shell负责将命令翻译，在系统执行完之后将结果返回给终端。
- **Shell**----是包裹在操作系统外层的一道程序，就好像是操作系统的壳。Shell接收用户或者其他应用程序的命令，然后将这些命令转化成内核能理解的语言并传给内核，内核执行命令完成后将结果返回给用户或者应用程序。
- **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。

2.3 Hadoop的安装和使用

□熟悉常用的Linux命令

1.cd命令：切换目录

- (1) 切换到目录/usr/local `cd /usr/local`
- (2) 切换到当前目录的上一级目录 `cd ..`
- (3) 切换到当前登录Linux系统的用户的自己的主文件夹 `cd ~`
- (4) 查看“当前工作目录”的完整路径 `pwd`

2. Ls或ll命令：查看文件和目录

查看目录/usr下的所有文件和目录

`cd /usr` `ls` `ll`

说明： `drwxr-xr-x` user1 group1 filename 表示filename是个目录，user1拥有读写执行的权限，和user1所在同一个group1组里的用户拥有只读和执行权限，剩下其他用户拥有只读和执行权限。

2.3 Hadoop的安装和使用

□熟悉常用的Linux命令

3. mkdir命令：新建目录

(1) 进入/tmp目录，创建一个名为a的目录，并查看/tmp目录下已经存在哪些目录

```
cd /tmp
```

```
mkdir a
```

(2) 进入/tmp目录，创建目录a1/a2/a3/a4

```
cd /tmp
```

```
mkdir -p a1/a2/a3/a4
```

2.3 Hadoop的安装和使用

□熟悉常用的Linux命令

4. rmdir命令：删除空的目录

(1) 将上面创建的目录a（在"/tmp"目录下面）删除。

```
cd /tmp
```

```
rmdir a
```

(2) 删除上面创建的目录a1/a2/a3/a4，然后查看/tmp目录下面存在哪些目录。

```
cd /tmp
```

```
rmdir --p a1/a2/a3/a4
```

```
ls -al
```

2.3 Hadoop的安装和使用

□熟悉常用的Linux命令

5. cp命令：复制文件或目录

(1) 将当前用户的主文件夹下的文件.bashrc复制到目录/usr下，并重命名为bashrc1

```
sudo cp ~/.bashrc /usr/bashrc1
```

(2) 在目录/tmp下新建目录test，再把这个目录复制到/usr目录下

```
cd /tmp
```

```
mkdir test
```

```
sudo cp -r /tmp/test /usr
```


2.3 Hadoop的安装和使用

□熟悉常用的Linux命令

6. mv命令：移动文件和目录，或重命名

(1) 将/usr目录下的文件bashrc1移动到/usr/test目录下

```
sudo mv /usr/bashrc1 /usr/test
```

(2) 将/usr目录下的test目录重命名为test2

```
sudo mv /usr/test /usr/test2
```

7. rm命令：移除文件或目录

(1) 将/usr/test2目录下的bashrc1文件删除

```
sudo rm /usr/test2/bashrc1
```

(2) 将/usr目录下的test2目录删除

```
sudo rm -r /usr/test2
```

2.3 Hadoop的安装和使用

□熟悉常用的Linux命令

8. cat命令：查看文件内容

查看当前用户主文件夹下的.bashrc文件内容

```
cat ~/.bashrc
```

9. touch命令：修改文件时间或创建新文件

(1) 在/tmp目录下创建一个空文件hello，并查看文件时间

```
cd /tmp
```

```
touch hello
```

```
ls -l hello
```

(2) 修改hello文件，将文件时间调整为5天前

```
touch -d "5 days ago" hello
```

2.3 Hadoop的安装和使用

□熟悉常用的Linux命令

10. chown命令：修改文件所有者权限

将hello文件所有者改为root账号，并查看属性

```
sudo chown root /tmp/hello
```

11. tar命令：压缩命令

(1) 在根目录"/"下新建文件夹test，然后在根目录"/"下打包成test.tar.gz

```
sudo mkdir /test
```

```
sudo tar -zcv -f /test.tar.gz test
```

(2) 把上面的test.tar.gz压缩包，解压缩到/tmp目录

```
sudo tar -zxv -f /test.tar.gz -C /tmp
```

```
ls -l /tmp/test
```

z:代表的是压缩 c:代表的是打包 x:代表的是解压 v:代表的是过程 f:代表的是指定文件名

2.3 Hadoop的安装和使用

□Hadoop的三种安装模式

- ① **单机模式**：存储采用本机文件系统，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（NameNode）和数据节点（DataNode）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（NameNode）和数据节点（DataNode）位于不同机器上

2.3 Hadoop的安装和使用

□安装Hadoop前准备工作

➤ 下载安装文件

① Hadoop官网：

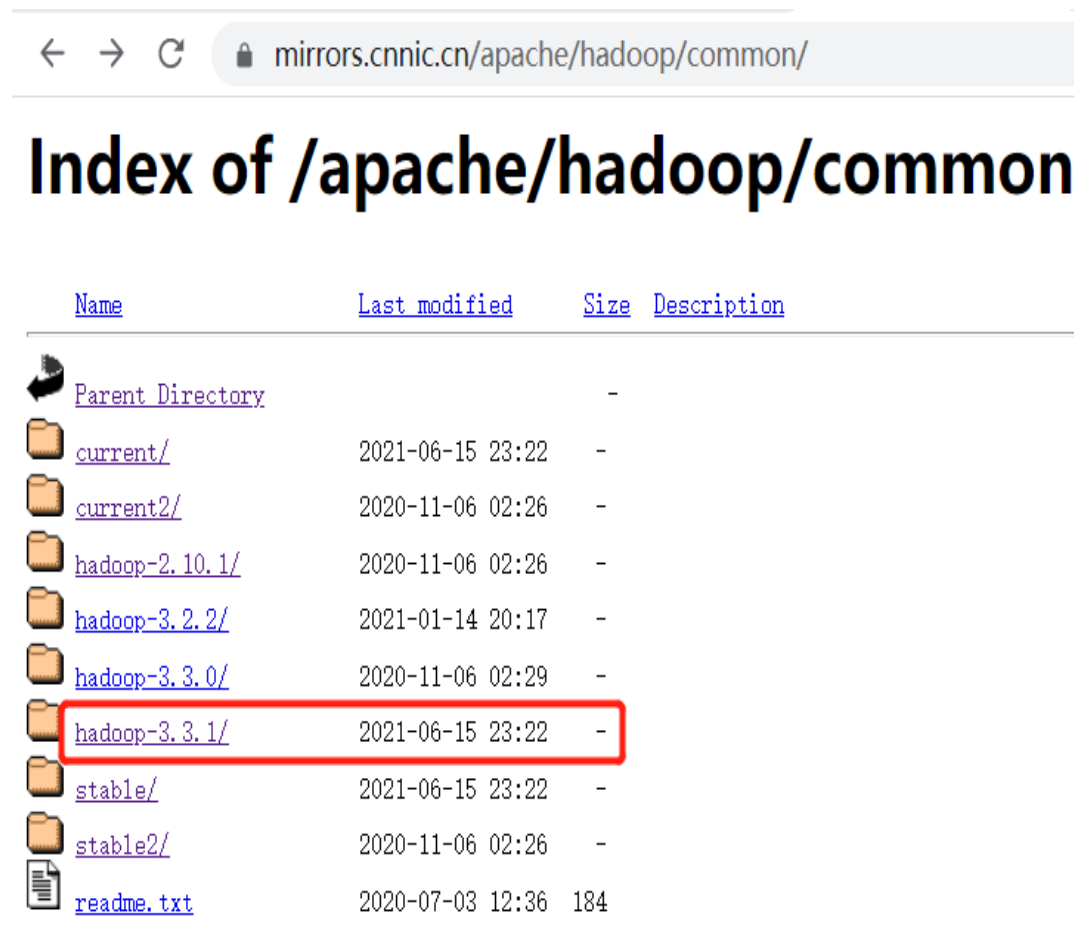
<http://mirrors.cnnic.cn/apache/hadoop/common/>











② Hadoop版本： 3.3.1

③ 在Ubuntu Linux系统中下载文件

`hadoop-3.3.1.tar.gz`

<https://dbllab.xmu.edu.cn/blog/2441/>



Index of /apache/hadoop/common			
Name	Last modified	Size	Description
 Parent Directory		-	
 current/	2021-06-15 23:22	-	
 current2/	2020-11-06 02:26	-	
 hadoop-2.10.1/	2020-11-06 02:26	-	
 hadoop-3.2.2/	2021-01-14 20:17	-	
 hadoop-3.3.0/	2020-11-06 02:29	-	
 hadoop-3.3.1/	2021-06-15 23:22	-	
 stable/	2021-06-15 23:22	-	
 stable2/	2020-11-06 02:26	-	
 readme.txt	2020-07-03 12:36	184	

2.3.1 创建hadoop用户

① 创建普通用户“hadoop”，并设置密码为“hadoop”

```
sudo useradd -m hadoop -s /bin/bash
```

这条命令创建了hadoop用户，并使用/bin/bash作为shell

```
sudo passwd hadoop
```

② 为hadoop用户增加管理员权限，以方便部署

```
sudo adduser hadoop sudo #将用户hadoop添加到sudo用户组中
```

③ 注销“dblab”用户，以“hadoop”用户登录到Ubuntu Linux

2.3.2更新APT和安装Vim编辑器

- ① 说明：APT是一款非常优秀的软件管理工具，Linux系统采用APT来安装和管理各种软件，Linux虚拟机安装成功后，应及时更新APT软件

```
sudo apt-get update
```

- ② 在Linux中安装vim编辑器

```
sudo apt-get install vim
```

vim的常用模式：

正常模式： 主要用来浏览文本内容。一开始打开vim都是正常模式。在任何模式下按下Esc键就可以返回正常模式

插入编辑模式： 则用来向文本中添加内容的。在正常模式下，输入i键即可进入插入编辑模式

退出vim： 如果有利用vim修改任何的文本，一定要记得保存。Esc键退回到正常模式中，然后输入:wq即可保存文本并退出vim

2.3.3 安装SSH并进行配置

□SSH 为 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是由客户端和服务端的软件组成。

- ① **服务端软件**：是一个守护进程，在后台运行并响应来自客户端的连接请求
- ② **客户端软件**：客户端包含ssh程序以及像scp（远程拷贝）、slogin（远程登陆）、sftp（安全文件传输）等其他的应用程序

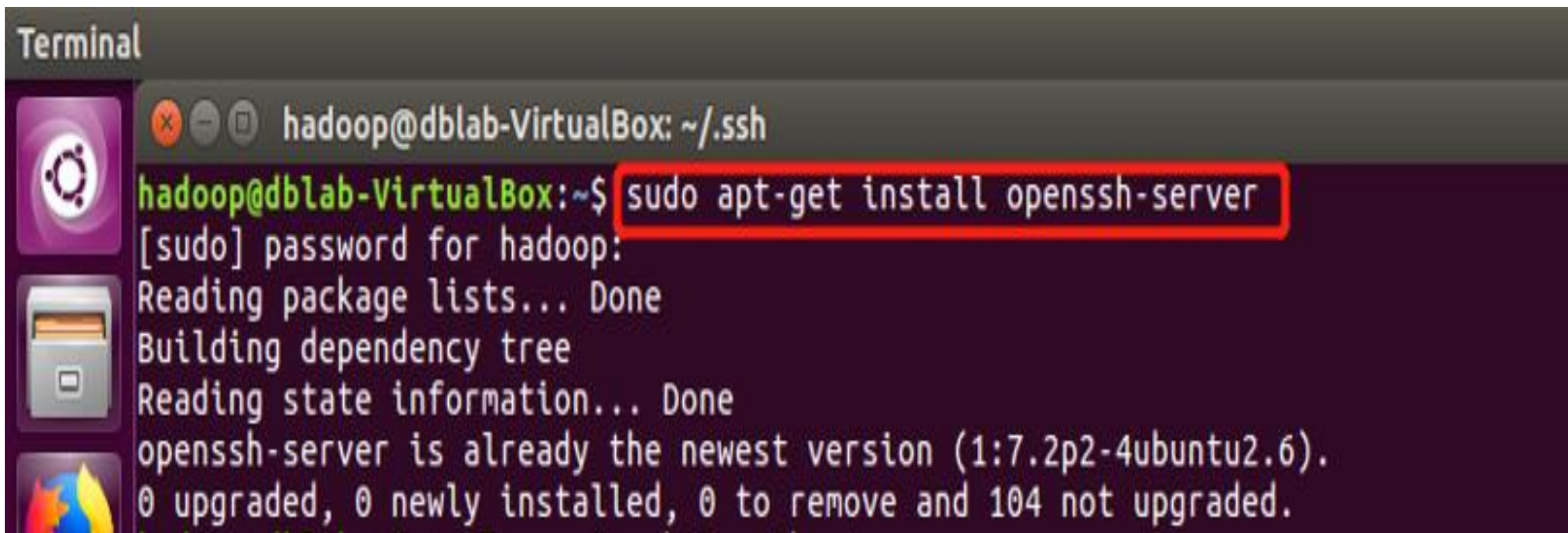
2.3.3 安装SSH并进行配置

- 安装Hadoop之前为什么要配置SSH
 - ① Hadoop名称节点（NameNode）需要启动集群中所有机器的Hadoop守护进程，需要通过SSH登录来实现
 - ② Hadoop没有提供SSH输入密码登录的形式
 - ③ 需要将所有机器配置为“名称节点（NameNode）可以无密码登录”

2.3.3 安装SSH并进行配置

- Ubuntu默认已安装了SSH客户端，需要安装SSH服务端

`sudo apt-get install openssh-server`



The image shows a terminal window titled "Terminal" with a dark background. The window has a sidebar on the left with icons for a terminal, a file manager, and a web browser. The terminal content shows the user "hadoop" at the prompt "hadoop@dblab-VirtualBox: ~/.ssh" typing the command "sudo apt-get install openssh-server". The command is highlighted with a red rectangle. The output shows the password prompt, package list reading, dependency tree building, and state information reading. It then states that "openssh-server" is already the newest version (1:7.2p2-4ubuntu2.6) and that 0 packages are upgraded, 0 are newly installed, 0 are to be removed, and 104 are not upgraded.

```
Terminal
hadoop@dblab-VirtualBox: ~/.ssh
hadoop@dblab-VirtualBox:~$ sudo apt-get install openssh-server
[sudo] password for hadoop:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.2p2-4ubuntu2.6).
0 upgraded, 0 newly installed, 0 to remove and 104 not upgraded.
```

2.3.3 安装SSH并进行配置

- 安装后，可以使用以下命令登录本机

`ssh localhost`

A terminal window with a dark background. The prompt is 'hadoop@dblab-VirtualBox:~\$'. The command 'ssh localhost' is entered and highlighted with a red rectangular box. Below the command, the terminal displays the following text: 'The authenticity of host 'localhost (127.0.0.1)' can't be established.', 'ECDSA key fingerprint is SHA256:1NYsKlYf0NiynwpYVJKvD3ofi2jzqfg0dVhzBhs3Fpo.', and 'Are you sure you want to continue connecting (yes/no)?'. The word 'yes' is entered at the end of the last line.

```
hadoop@dblab-VirtualBox:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:1NYsKlYf0NiynwpYVJKvD3ofi2jzqfg0dVhzBhs3Fpo.
Are you sure you want to continue connecting (yes/no)? yes
```

需要输入密码

2.3.3 安装SSH并进行配置

- 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。

①首先输入exit退出刚才的SSH，回到原来的终端窗口

②利用命令ssh-keygen生成密钥

exit # 退出刚才的 ssh localhost

cd ~/.ssh/ # 若没有该目录，请先执行一次ssh localhost

ssh-keygen -t rsa # 会有提示，都按回车就可以

- ~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/"。

2.3.3 安装SSH并进行配置

```
hadoop@dblab-VirtualBox:~$ cd ~/.ssh/
hadoop@dblab-VirtualBox:~/.ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:06e03VOx+mHn0Y8CsTWRyIAUSviICdxISDFLxXvKxtE hadoop@dblab-VirtualBox
The key's randomart image is:
+---[RSA 2048]---+
|+==O..+...  .|
|++++ . .  o o|
|o+.o+      . . .|
|o .o.E      . o .|
|  o +      S  + . o|
|   =       .o  o|
|  .        + .. oo .|
|           . * .+ . *|
|           o . .+o.=|
+-----[SHA256]-----+
```

2.3.3 安装SSH并进行配置

③ 将生成的密钥id_rsa.pub加入授权中（添加到授权文件authorized_keys中）

`cat ./id_rsa.pub >> ./authorized_keys` # 加入授权

④ 再执行`ssh localhost`命令，无须输入密码就可以直接登录了。

```
hadoop@dblab-VirtualBox:~/.ssh$ ls
id_rsa  id_rsa.pub  known_hosts
hadoop@dblab-VirtualBox:~/.ssh$ cat ./id_rsa.pub >> ./authorized_keys
hadoop@dblab-VirtualBox:~/.ssh$ ssh localhost
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-43-generic x86_64)

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

105 packages can be updated.
0 updates are security updates.

hadoop@dblab-VirtualBox:~$
```

2.3.4 安装Java环境

□ Hadoop3.1.3需要JDK版本在1.8及以上。按照下面步骤安装JDK1.8。

① 把压缩格式的文件jdk-8u162-linux-x64.tar.gz下载到本地电脑，保存在“/home/Hadoop/Downloads/”目录下。(解决权限不足问题的方法就是将自己登录的用户，添加到vboxsf组中)

```
sudo usermod -a -G vboxsf hadoop
```

2.3.4 安装Java环境

- ② 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）：

```
cd /usr/lib
```

```
sudo mkdir jvm #创建/usr/lib/jvm目录用来存放JDK文件
```

```
cd ~ #进入hadoop用户的主目录
```

```
cd Downloads #注意区分大小写字母
```

- ③ 把JDK文件解压到/usr/lib/jvm目录下

```
sudo tar -zxvf ./jdk-8u162-linux-x64.tar.gz -C /usr/lib/jvm
```

```
cd /usr/lib/jvm
```

```
ls
```


2.3.4 安装Java环境

- ④ 继续执行如下命令，设置环境变量：

```
cd ~
```

```
vim ~/.bashrc
```

- 上面命令使用vim编辑器打开hadoop用户的环境变量配置文件，在文件的开头位置，添加如下几行内容，保存.bashrc文件退出vim编辑器。

```
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
```

- 继续执行如下命令让.bashrc文件的配置立即生效：

```
source ~/.bashrc
```

2.3.4 安装Java环境

⑤ 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：

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)

2.3.5 安装单机Hadoop

(1) 将 Hadoop 安装包hadoop-3.1.3.tar.gz解压至 /usr/local/ 中:

```
sudo tar -zxf ~/下载/hadoop-3.1.3.tar.gz -C /usr/local
```

```
cd /usr/local/
```

```
sudo mv ./hadoop-3.1.3/ ./hadoop          # 将文件夹名改为hadoop
```

```
hadoop@dblab-VirtualBox:~$ sudo tar -zxf ~/下载/hadoop-3.3.1.tar.gz -C /usr/local
[sudo] hadoop 的密码:
hadoop@dblab-VirtualBox:~$ cd /usr/local
hadoop@dblab-VirtualBox:/usr/local$ ls
bin  etc  games  hadoop-3.3.1  hbase  include  lib  libexec  lua  man  redis  sbin  share  src
hadoop@dblab-VirtualBox:/usr/local$ sudo mv ./hadoop-3.3.1/ ./hadoop
[sudo] hadoop 的密码:
hadoop@dblab-VirtualBox:/usr/local$ ls
bin  etc  games  hadoop  hbase  include  lib  libexec  lua  man  redis  sbin  share  src
hadoop@dblab-VirtualBox:/usr/local$
```

2.3.5 安装单机Hadoop

(2) 使用chown命令把用户hadoop改为Hadoop的安装目录/usr/local/Hadoop的拥有者

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

```
hadoop@dblab-VirtualBox:/usr/local$ sudo chown -R hadoop ./hadoop
hadoop@dblab-VirtualBox:/usr/local$ cd hadoop
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ls
bin  etc  include  lib  libexec  LICENSE.txt  NOTICE.txt  README.txt  sbin  share
```

2.3.5 安装单机Hadoop

(3) 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
```

说明： ./ 的路径为相对路径，以 /usr/local/hadoop 为当前目录

```
hadoop@dblab-VirtualBox:/usr/local$ ./hadoop/bin/hadoop version  
Hadoop 3.3.1  
Source code repository https://github.com/apache/hadoop.git -r a3b9c37a397ad4188041dd80621bdeefc46885f2  
Compiled by ubuntu on 2021-06-15T05:13Z  
Compiled with protoc 3.7.1  
From source with checksum 88a4ddb2299aca054416d6b7f81ca55  
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.1.jar  
hadoop@dblab-VirtualBox:/usr/local$
```

2.3.5 安装单机Hadoop

- Hadoop的默认模式为非分布式模式（即本地模式），无须进行其他配置即可运行。
- 查看安装好的Hadoop的目录内容：

```
hadoop@dblab-VirtualBox:/usr/local$ cd hadoop
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ls -l
总用量 112
drwxr-xr-x 2 hadoop dblab 4096 6月 15 2021 bin
drwxr-xr-x 3 hadoop dblab 4096 6月 15 2021 etc
drwxr-xr-x 2 hadoop dblab 4096 6月 15 2021 include
drwxr-xr-x 3 hadoop dblab 4096 6月 15 2021 lib
drwxr-xr-x 4 hadoop dblab 4096 6月 15 2021 libexec
-rw-rw-r-- 1 hadoop dblab 23450 6月 15 2021 LICENSE-binary
drwxr-xr-x 2 hadoop dblab 4096 6月 15 2021 licenses-binary
-rw-rw-r-- 1 hadoop dblab 15217 6月 15 2021 LICENSE.txt
-rw-rw-r-- 1 hadoop dblab 29473 6月 15 2021 NOTICE-binary
-rw-rw-r-- 1 hadoop dblab 1541 5月 22 2021 NOTICE.txt
-rw-rw-r-- 1 hadoop dblab 175 5月 22 2021 README.txt
drwxr-xr-x 3 hadoop dblab 4096 6月 15 2021 sbin
drwxr-xr-x 4 hadoop dblab 4096 6月 15 2021 share
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.3.5 安装单机Hadoop

(4) Hadoop附帶了丰富的例子，运行如下命令可以查看所有例子。

`./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar`

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar
An example program must be given as the first argument.
Valid program names are:
aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.
aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input files.
bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
dbcount: An example job that count the pageview counts from a database.
distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
grep: A map/reduce program that counts the matches of a regex in the input.
join: A job that effects a join over sorted, equally partitioned datasets
multifilewc: A job that counts words from several files.
pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
randomwriter: A map/reduce program that writes 10GB of random data per node.
secondarysort: An example defining a secondary sort to the reduce.
sort: A map/reduce program that sorts the data written by the random writer.
sudoku: A sudoku solver.
teragen: Generate data for the terasort
terasort: Run the terasort
teravalidate: Checking results of terasort
wordcount: A map/reduce program that counts the words in the input files.
wordmean: A map/reduce program that counts the average length of the words in the input files.
wordmedian: A map/reduce program that counts the median length of the words in the input files.
wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in the input files.
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.3.5 安装单机Hadoop

- **aggregatewordcount**: 一个基于聚合的map/reduce程序，用于统计输入文件中的单词。
- **bbp**: 一个使用Bailey-Borwein-Plouffe计算pi的精确值的map/reduce程序。
- **grep**: 一个map/reduce程序，用于统计输入中正则表达式的匹配情况。
- **join**: 一种在已排序、分区相等的数据集上实现联接的作业
- **multifilewc**: 对多个文件中的单词进行计数的作业。
- **pi**: 一个map/reduce程序，使用“拟蒙特卡罗方法”估计pi。
- **sort**: 一个map/reduce程序，用于对随机写入器写入的数据进行排序。
- **sudoku**: 一个数独解算器。
- **wordcount**: 一个对输入文件中的单词进行计数的map/reduce程序。
- **wordmean**: 一个计算输入文件中单词平均长度的map/reduce程序。
- **randomtextwriter**: 一个map/reduce程序，每个节点写入10GB的随机文本数据。

2.3.5 安装单机Hadoop

- 运行自带的程序grep：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。输出的结果是符合正则的单词 `dfsadmin` 出现了1次

`cd /usr/local/hadoop`

`mkdir ./input`

`cp ./etc/hadoop/*.xml ./input`

`# 将配置文件作为输入文件`

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ mkdir input
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ls
bin  etc  include  input  lib  libexec  LICENSE-binary  licenses-binary  LICENSE.txt
hadoop@dblab-VirtualBox:/usr/local/hadoop$ sudo cp ./etc/hadoop/*.xml ./input
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ls -l ./input
总用量 56
-rw-r--r-- 1 root root 9213 3月  2 12:40 capacity-scheduler.xml
-rw-r--r-- 1 root root  774 3月  2 12:40 core-site.xml
-rw-r--r-- 1 root root 11765 3月  2 12:40 hadoop-policy.xml
-rw-r--r-- 1 root root  683 3月  2 12:40 hdfs-rbf-site.xml
-rw-r--r-- 1 root root  775 3月  2 12:40 hdfs-site.xml
-rw-r--r-- 1 root root  620 3月  2 12:40 httpfs-site.xml
-rw-r--r-- 1 root root 3518 3月  2 12:40 kms-acls.xml
-rw-r--r-- 1 root root  682 3月  2 12:40 kms-site.xml
-rw-r--r-- 1 root root  758 3月  2 12:40 mapred-site.xml
-rw-r--r-- 1 root root  690 3月  2 12:40 yarn-site.xml
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.3.5 安装单机Hadoop

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

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar grep ./input ./output 'dfs[a-z.]+'
```

```
2022-03-02 12:44:50,507 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2022-03-02 12:44:50,942 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2022-03-02 12:44:50,942 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2022-03-02 12:44:51,437 INFO input.FileInputFormat: Total input files to process : 10
2022-03-02 12:44:51,535 INFO mapreduce.JobSubmitter: number of splits:10
2022-03-02 12:44:52,215 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1785671686_0001
2022-03-02 12:44:52,216 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-03-02 12:44:52,568 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2022-03-02 12:44:52,569 INFO mapreduce.Job: Running job: job_local1785671686_0001
2022-03-02 12:44:52,575 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2022-03-02 12:44:52,587 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
```

`cat ./output/*` # 查看运行结果

注意：下一次运行时，应该先删除输出目录，否则会出错：`rm -r output`

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ cat ./output/*
1      dfsadmin
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.3.5 安装单机Hadoop

- 在单机模式下运行自带的程序wordcount:
 - ①在/usr/local/hadoop目录下创建一个文件夹myinput2，用vim编辑器创建2个文件file1.txt， file2.txt，内容分别为：

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ mkdir myinput2
```

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ cat ./myinput2/file1.txt
delight myself volume value velocity variety
value volume velocity variety
hadoop mapreduce
hadoop@dblab-VirtualBox:/usr/local/hadoop$ cat ./myinput2/file2.txt
surprise angry volume value velocity variety
value volume velocity variety
hadoop mapreduce
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.3.5 安装单机Hadoop

②运行wordcount程序，把文件夹myinput2中的所有文件作为输入，结果输出到output2中：

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar wordcount  
./myinput2 ./output2
```

```
19/02/26 22:37:14 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id  
19/02/26 22:37:14 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=  
19/02/26 22:37:15 INFO input.FileInputFormat: Total input paths to process : 2  
19/02/26 22:37:15 INFO mapreduce.JobSubmitter: number of splits:2  
19/02/26 22:37:15 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1241773735_0001  
19/02/26 22:37:15 INFO mapreduce.Job: The url to track the job: http://localhost:8080/  
19/02/26 22:37:15 INFO mapreduce.Job: Running job: job_local1241773735_0001
```

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ls ./output2
```

```
part-r-000000 _SUCCESS
```

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ cat ./output2/*
```

```
angry 1  
delight 1  
hadoop 2  
mapreduce 2  
myself 1  
surprise 1  
value 4  
variety 4  
velocity 4  
volume 4
```

_SUCCESS文件：表示MapReduce运行成功

part-r-000000文件：存放结果，是默认生成的结果文件

2.3.6 Hadoop伪分布式模式配置

- 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。

2.3.6 Hadoop伪分布式模式配置

- 修改配置文件core-site.xml

该配置文件用于确定Hadoop的核心信息，包括临时目录、访问地址(HDFS URL)等

```
<configuration>
```

```
<property>
```

```
<name>hadoop.tmp.dir</name>
```

```
<value>file:/usr/local/hadoop/tmp</value>
```

```
<description>A base for other temporary directories.</description>
```

```
</property>
```

```
<property>
```

```
<name>fs.defaultFS</name>
```

```
<value>hdfs://localhost:9000</value>
```

```
</property>
```

```
</configuration>
```

用于保存临时文件。如果不配置该参数，则默认的临时目录(/tmp)在Hadoop重启时有可能被系统清理掉，导致问题出现。

用来定义Hadoop文件系统的类型，指定HDFS的访问地址(URL)，其中，9000是端口号。

2.3.6 Hadoop伪分布式模式配置

- 修改配置文件hdfs-site.xml

```
<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>
```

```
<property>
```

```
<name>dfs.namenode.acls.enabled</name>
```

```
<value>true</value>
```

```
</property>
```

```
</configuration>
```

该配置文件用于确定文件的备份个数、数据文件夹的路径（名称节点和数据节点的存放位置）、文件的读取权限等

用于指定副本的数量，这里为1

用于设定名称节点（**NameNode**）的元数据的保存目录

用于设定数据节点（**DataNode**）的数据的保存目录

Hadoop HDFS 默认没有使用 **ACL(Access Control List)** 权限控制机制。设置为**true**表示开启 **HDFS** 的权限控制机制

2.3.6 Hadoop伪分布式模式配置

- 配置完成后，执行 NameNode 的格式化命令，成功的话，会看到 "successfully formatted" 的提示

`cd /usr/local/hadoop`

`./bin/hdfs namenode -format`

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs namenode -format
WARNING: /usr/local/hadoop/logs does not exist. Creating.
2022-03-02 13:17:57,668 INFO namenode.NameNode: STARTUP_MSG:
2022-03-02 13:18:00,852 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1856939348-127.0.1.1-1646198280830
2022-03-02 13:18:00,940 INFO common.Storage: Storage directory /usr/local/hadoop/tmp/dfs/name has been successfully formatted.
2022-03-02 13:18:01,051 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt_000000000000000000 using no compression
2022-03-02 13:18:01,276 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt_000000000000000000 of size 398 bytes saved in 0 seconds .
2022-03-02 13:18:01,358 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2022-03-02 13:18:01,441 INFO namenode.FSNamesystem: Stopping services started for active state
2022-03-02 13:18:01,441 INFO namenode.FSNamesystem: Stopping services started for standby state
2022-03-02 13:18:01,450 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2022-03-02 13:18:01,450 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at dblab-VirtualBox/127.0.1.1
*****/
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./sbin/start-dfs.sh
```


2.3.6 Hadoop伪分布式模式配置

- 接着开启 NameNode 和 DataNode 守护进程。
`./sbin/start-dfs.sh` #start-dfs.sh是个完整的可执行文件，中间没有空格
- 启动完成后，通过命令 `jps` 判断是否成功启动，成功启动则会列出进程：
"NameNode"、"DataNode" 和 "SecondaryNameNode"

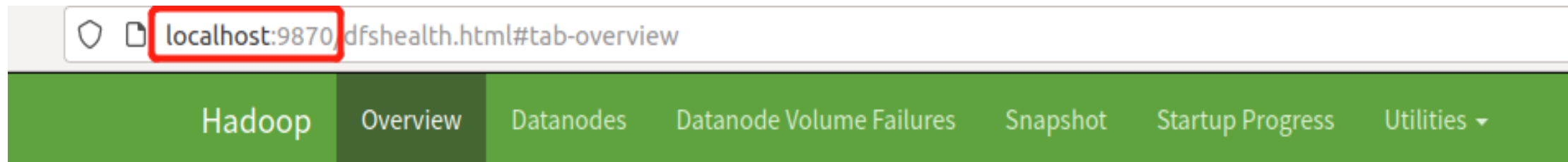
`jps`

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [dblab-VirtualBox]
hadoop@dblab-VirtualBox:/usr/local/hadoop$ jps
5826 DataNode
6053 SecondaryNameNode
5674 NameNode
6236 Jps
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.3.6 Hadoop伪分布式模式配置

- 若是 NameNode或者DataNode 没有启动，解决方法如下：
`cd /usr/local/hadoop`
`./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 NameNode`
`./sbin/start-dfs.sh # 重启`
- 成功启动后，可以访问 Web 界面 `http://localhost:9870` 查看 NameNode 和 Datanode 信息，还可以在线查看 HDFS 中的文件。

2.3.6 Hadoop伪分布式模式配置



Overview 'localhost:9000' (✓active)

Started:	Wed Mar 02 13:21:51 +0800 2022
Version:	3.3.1, ra3b9c37a397ad4188041dd80621bdeefc46885f2
Compiled:	Tue Jun 15 13:13:00 +0800 2021 by ubuntu from (HEAD detached at release-3.3.1-RC3)
Cluster ID:	CID-9e5549b3-1a15-4314-b36f-330733536097
Block Pool ID:	BP-1856939348-127.0.1.1-1646198280830

2.3.6 Hadoop伪分布式模式配置

- 启动Hadoop后，使用命令hdfs dfs查看HDFS常用命令的用法：

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs
Usage: hadoop fs [generic options]
    [-appendToFile <localsrc> ... <dst>]
    [-cat [-ignoreCrc] <src> ...]
    [-checksum <src> ...]
    [-chgrp [-R] GROUP PATH...]
    [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
    [-chown [-R] [OWNER][:[GROUP]] PATH...]
    [-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] <localsrc> ... <dst>]
    [-copyToLocal [-f] [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
    [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] <path> ...]
```

- 查看某命令的用法和帮助：hdfs dfs -usage ls 和 hdfs dfs -help ls

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -usage ls
Usage: hadoop fs [generic options] -ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] [<path> ...]
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -help ls
-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] [<path> ...] :
  List the contents that match the specified file pattern. If path is not
  specified, the contents of /user/<currentUser> will be listed. For a directory a
  list of its direct children is returned (unless -d option is specified).

  Directory entries are of the form:
      permissions - userId groupId sizeOfDirectory(in bytes)
      modificationDate(yyyy-MM-dd HH:mm) directoryName
```

2.3.6 Hadoop伪分布式模式配置

□ 运行Hadoop伪分布式实例

创建多级目录，要使用选项-p，否则出错

- 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：

```
cd /usr/local/hadoop
```

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

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -mkdir -p /user/hadoop
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -ls .
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -ls /user/hadoop
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

刚创建好的用户目录/user/hadoop下面没有任何文件

2.3.6 Hadoop伪分布式模式配置

- 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。

`./bin/hdfs dfs -mkdir input`

`./bin/hdfs dfs -put ./etc/hadoop/*.xml input`

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -mkdir myinput
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -ls -R .
drwxr-xr-x   - hadoop supergroup          0 2022-03-02 15:51 bigfile
-rw-r--r--   1 hadoop supergroup 209715200 2022-03-02 15:51 bigfile/bigfile1
drwxr-xr-x   - hadoop supergroup          0 2022-03-02 16:38 myinput
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.3.6 Hadoop伪分布式模式配置

- 复制完成后，可以通过如下命令查看文件列表：

`./bin/hdfs dfs -ls input`

```
hadoop@dblab-VirtualBox: /usr/local/hadoop$ ./bin/hdfs dfs -put ./etc/hadoop/*.xml ./myinput
hadoop@dblab-VirtualBox: /usr/local/hadoop$ ./bin/hdfs dfs -ls ./myinput
Found 10 items
-rw-r--r--  1 hadoop supergroup      9213 2022-03-02 16:48 myinput/capacity-scheduler.xml
-rw-r--r--  1 hadoop supergroup     1076 2022-03-02 16:48 myinput/core-site.xml
-rw-r--r--  1 hadoop supergroup    11765 2022-03-02 16:48 myinput/hadoop-policy.xml
-rw-r--r--  1 hadoop supergroup      683 2022-03-02 16:48 myinput/hdfs-rbf-site.xml
-rw-r--r--  1 hadoop supergroup     1241 2022-03-02 16:48 myinput/hdfs-site.xml
-rw-r--r--  1 hadoop supergroup      620 2022-03-02 16:48 myinput/httpfs-site.xml
-rw-r--r--  1 hadoop supergroup     3518 2022-03-02 16:48 myinput/kms-acls.xml
-rw-r--r--  1 hadoop supergroup      682 2022-03-02 16:48 myinput/kms-site.xml
-rw-r--r--  1 hadoop supergroup      878 2022-03-02 16:48 myinput/mapred-site.xml
-rw-r--r--  1 hadoop supergroup      823 2022-03-02 16:48 myinput/yarn-site.xml
hadoop@dblab-VirtualBox: /usr/local/hadoop$
```


2.3.6 Hadoop伪分布式模式配置

- 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。

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

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar grep myinput myoutput 'dfs[a-z.]+'
2022-03-02 17:39:06,235 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2022-03-02 17:39:07,446 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1646213932929_0001
2022-03-02 17:39:08,982 INFO input.FileInputFormat: Total input files to process : 10
2022-03-02 17:39:09,194 INFO mapreduce.JobSubmitter: number of splits:10
2022-03-02 17:39:09,973 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1646213932929_0001
2022-03-02 17:39:09,973 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-03-02 17:39:10,490 INFO conf.Configuration: resource-types.xml not found
2022-03-02 17:39:10,490 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-03-02 17:39:11,081 INFO impl.YarnClientImpl: Submitted application application_1646213932929_0001
2022-03-02 17:39:11,263 INFO mapreduce.Job: The url to track the job: http://dblab-VirtualBox:8088/proxy/application_1646213932929_0001/
2022-03-02 17:39:11,264 INFO mapreduce.Job: Running job: job_1646213932929_0001
2022-03-02 17:39:29,838 INFO mapreduce.Job: Job job_1646213932929_0001 running in uber mode : false
2022-03-02 17:39:29,840 INFO mapreduce.Job: map 0% reduce 0%
2022-03-02 17:41:07,339 INFO mapreduce.Job: map 20% reduce 0%
2022-03-02 17:41:08,347 INFO mapreduce.Job: map 60% reduce 0%
2022-03-02 17:42:49,220 INFO mapreduce.Job: map 100% reduce 20%
2022-03-02 17:42:55,261 INFO mapreduce.Job: map 100% reduce 67%
2022-03-02 17:43:10,334 INFO mapreduce.Job: map 100% reduce 100%
2022-03-02 17:43:14,361 INFO mapreduce.Job: Job job_1646213932929_0001 completed successfully
```


2.3.6 Hadoop伪分布式模式配置

- 查看运行结果的命令（查看的是位于 HDFS 中的输出结果）：

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

```
WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=261
File Output Format Counters
  Bytes Written=105
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -ls
Found 3 items
drwxr-xr-x - hadoop supergroup          0 2022-03-02 15:51 bigfile
drwxr-xr-x - hadoop supergroup          0 2022-03-02 16:48 mvinout
drwxr-xr-x - hadoop supergroup          0 2022-03-02 17:44 myoutput
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -cat myoutput/*
1      dfsadmin
1      dfs.replication
1      dfs.namenode.name.dir
1      dfs.namenode.acls.enabled
1      dfs.datanode.data.dir
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.3.6 Hadoop伪分布式模式配置

- 也可以将运行结果取回到本地：：

`rm -r ./myoutput` # 先删除本地的 myoutput 文件夹（如果存在）

`./bin/hdfs dfs -get myoutput ./output` # 将 HDFS 上的 myoutput 文件夹
拷贝到本机

`cat ./output/*`

- 说明：若要再次执行，需要执行如下命令删除 myoutput 文件夹：

`./bin/hdfs dfs -rm -r output`

- 若要关闭 Hadoop，则运行：

`./sbin/stop-dfs.sh`

若要删除的目录中包含子目录，应该使用参数 `-r`

```
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -rm -r myoutput
Deleted myoutput
hadoop@dblab-VirtualBox:/usr/local/hadoop$ ./bin/hdfs dfs -ls myoutput
ls: `myoutput': No such file or directory
hadoop@dblab-VirtualBox:/usr/local/hadoop$
```

2.4 本章小结

- **Hadoop被视为事实上的大数据处理标准**，本章介绍了Hadoop的发展历程，并阐述了Hadoop的高可靠性、高效性、高可扩展性、高容错性、成本低、运行在Linux平台上、支持多种编程语言等特性
- **Hadoop目前已经在各个领域得到了广泛的应用**，雅虎、Facebook、百度、淘宝、网易等公司都建立了自己的Hadoop集群
- **经过多年发展，Hadoop项目已经变得非常成熟和完善**，包括YARN、Zookeeper、HDFS、MapReduce、HBase、Hive、Mahout、Pig等子项目，其中，HDFS和MapReduce是Hadoop的两大核心组件

2.4 本章小结

□本章最后介绍了如何在Linux系统下完成Hadoop的安装和配置，这个部分是后续章节实践环节的基础

□Hadoop基本的安装配置主要包括以下6个步骤：

- (1) 创建hadoop用户（用户名可自己命名）
- (2) 更新apt和安装vim编辑器
- (3) 安装SSH和配置SSH免密登录
- (4) 安装Java环境
- (5) 安装单机Hadoop
- (6) Hadoop伪分布式配置

小练习

1、Hadoop源自于谷歌的哪三篇论文？

GFS (Google File System)、MapReduce、BigTable

2、Yarn是Hadoop第 (**2**) 版中的集群资源调度和管理框架，英文全称是 (**Yet Another Resource Negotiator**) 。

3、安装Ubuntu虚拟机时，至少要有几个分区？分别是什么？

2个分区，分别是交换分区 (swap) 和根分区 (/)

4、Hadoop安装成功后，已经包含了 (**HDFS**) 和 (**MapReduce**) ，不需要额外安装，而 (**Hbase**) 等其他组件，则需要另外下载安装。

5、Linux根目录下的目录usr的含义是 (**Unix Software Resources或Unix System Resources**) ，这是系统存放程序的目录；当安装Linux发行版官方提供的软件包时，大多安装在这里。

小练习

6、配置Hadoop时，Java的路径JAVA_HOME是在哪一个配置文件中设置的？

JAVA_HOME在用户目录下的配置文件/home/hadoop/.bashrc中设置

7、hdfs dfs 中的命令-get 和-put命令操作对象是（ **C** ）。

A. 文件 B. 目录 C. 两者都是 D. 两者都不是

8、下列哪一项通常是集群的最主要的瓶颈？（ **C** ）

A. CPU B. 网络 C. 磁盘I/O D. 内存

9、下面与HDFS类似的框架是（ **D** ）。

A. NTFS B. FAT32 C. EXT3 D. GFS

小练习

10、下列关于MapReduce说法不正确的是（ **C** ）。

A. MapReduce是一种计算框架

B. MapReduce来源于Google的学术论文

C. MapReduce程序只能用Java语言编写

D. MapReduce隐藏了并行计算的细节，方便使用

11、假设已经配置好环境变量，启动Hadoop和关闭Hadoop的命令分别是（ **D** ）。

A. start-dfs.sh, stop-dfs.sh

B. start-hdfs.sh, stop-hdfs.sh

C. start-hdfs.sh, stop-dfs.sh

D. start-dfs.sh, stop-hdfs.sh

12、Hadoop2.X采用（ **C** ）技术构建源代码？

A. ant

B. ivy

C. maven

D. makefile

小练习

13、说明Hadoop本地模式、伪分布式模式和分布式模式的特点

单机（本地）模式的特点：不会存在守护进程，所有东西都运行在一个JVM上，也没有分布式文件系统DFS，使用的是本地文件系统。单机模式适用于开发过程中运行MapReduce程序，是使用最少的一种模式。

伪分布式模式的特点：适用于开发和测试环境，所有守护进程都在同一台机器上运行。

分布式模式的特点：用于生产环境，使用N台主机组成一个Hadoop集群，Hadoop守护进程运行在每一台主机之上。集群里有NameNode运行的主机，有DataNode运行的主机，TaskTracker运行的主机。分布式环境下，主节点和从节点会分开。

小练习

14、Hadoop集群中，“jps”命令的用处：

检查NameNode,DataNode, SecondaryNameNode,TaskTracker,JobTracker等是否正常工作。

15、Hadoop集群需求什么样的网络？

Hadoop核心使用Secure Shell(SSH)来驱动从节点上的服务器进程，并在主节点和从节点之间使用password-less SSH连接。