

大数据技术原理与应用

2. 大数据处理架构Hadoop

陈建文

电子信息与通信学院

chenjw@hust.edu.cn

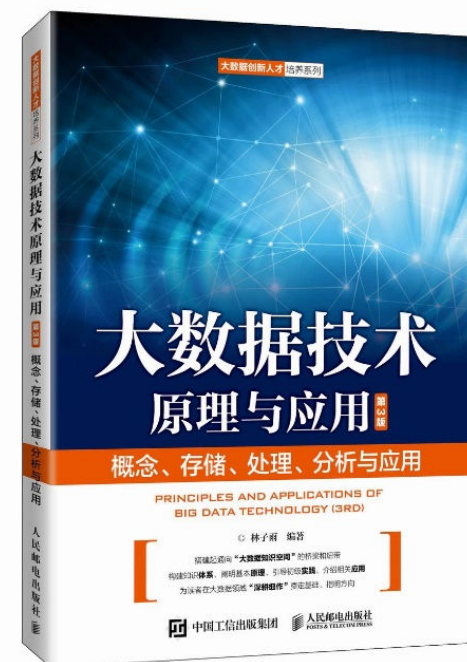
2. 大数据处理架构Hadoop

2.1 Hadoop 概述

2.2 Hadoop 项目结构

2.3 Hadoop 安装与使用

2.4 Hadoop 集群部署与使用



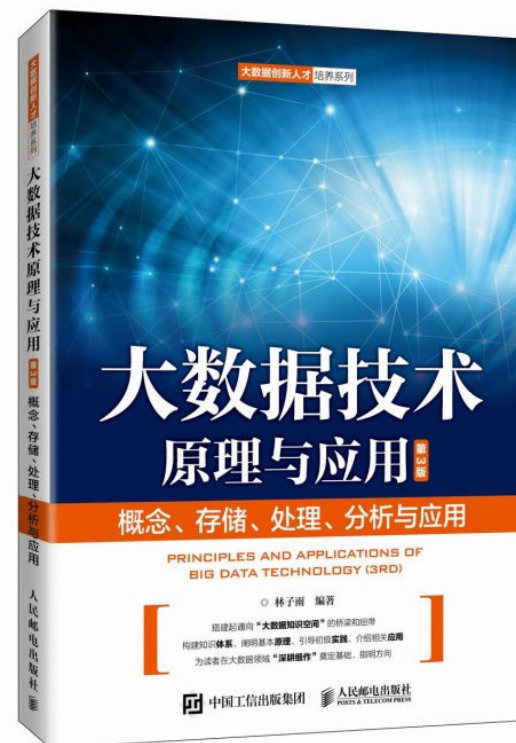
2.1 Hadoop 概述

2.1.1 Hadoop简介

2.1.2 Hadoop特性

2.1.3 Hadoop应用

2.1.4 Hadoop版本演变



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**。



Hadoop的标志

- 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并且也是**Lucene**项目的一部分；
- 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**），也就是**HDFS**的前身；
- 2004**年，谷歌公司又发表了另一篇具有深远影响的论文，阐述了**MapReduce**分布式编程思想；
- 2005**年，**Nutch**开源实现了谷歌的**MapReduce**。

- 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**更是把**1TB**数据排序时间缩短到**62**秒；**Hadoop**从此名声大震，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。

2.1.2 Hadoop 特性

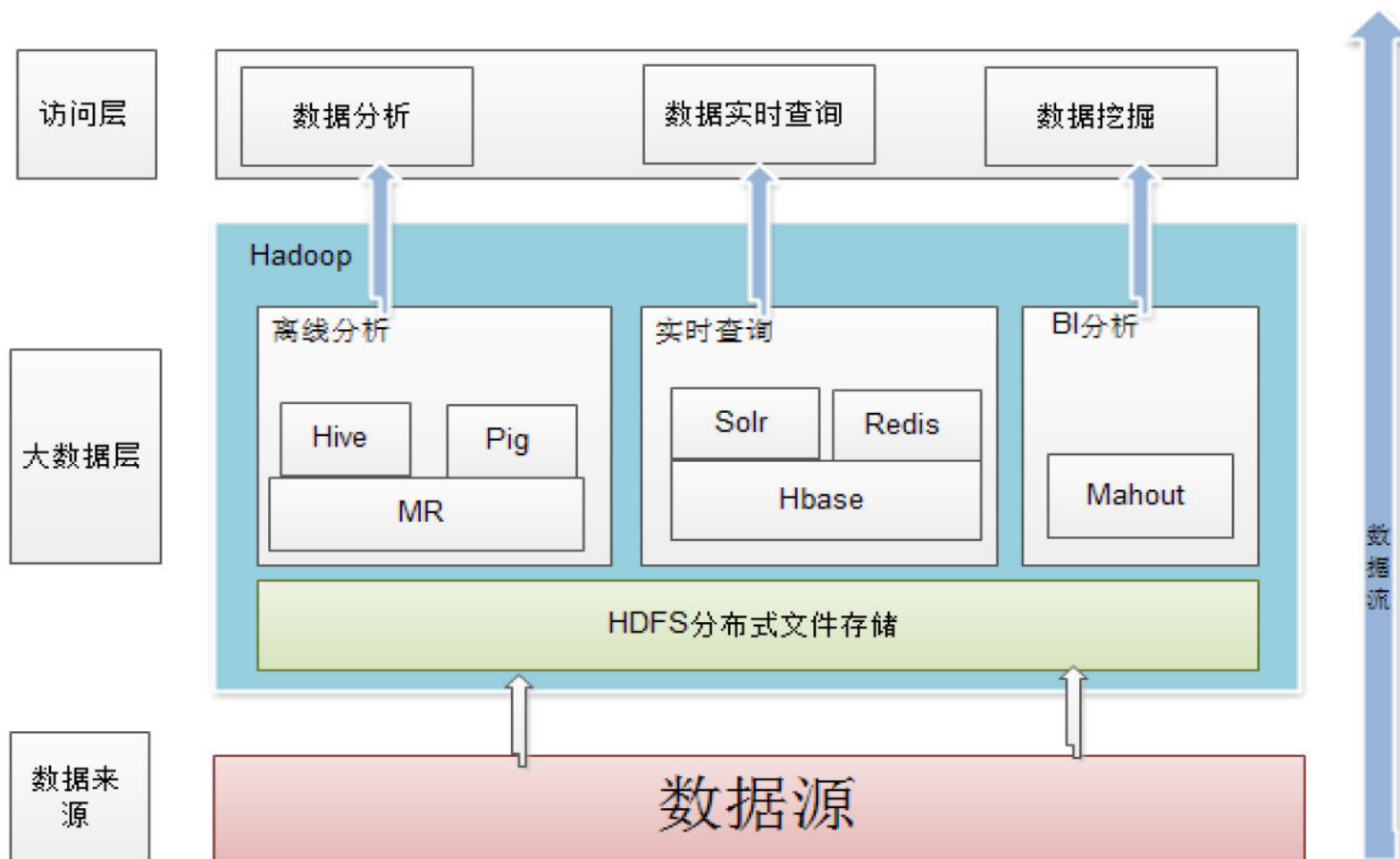
Hadoop是一个能够对大量数据进行分布式处理的软件框架，并且是以一种可靠、高效、可伸缩的方式进行处理，它具有以下几个方面的特性：

- 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应用

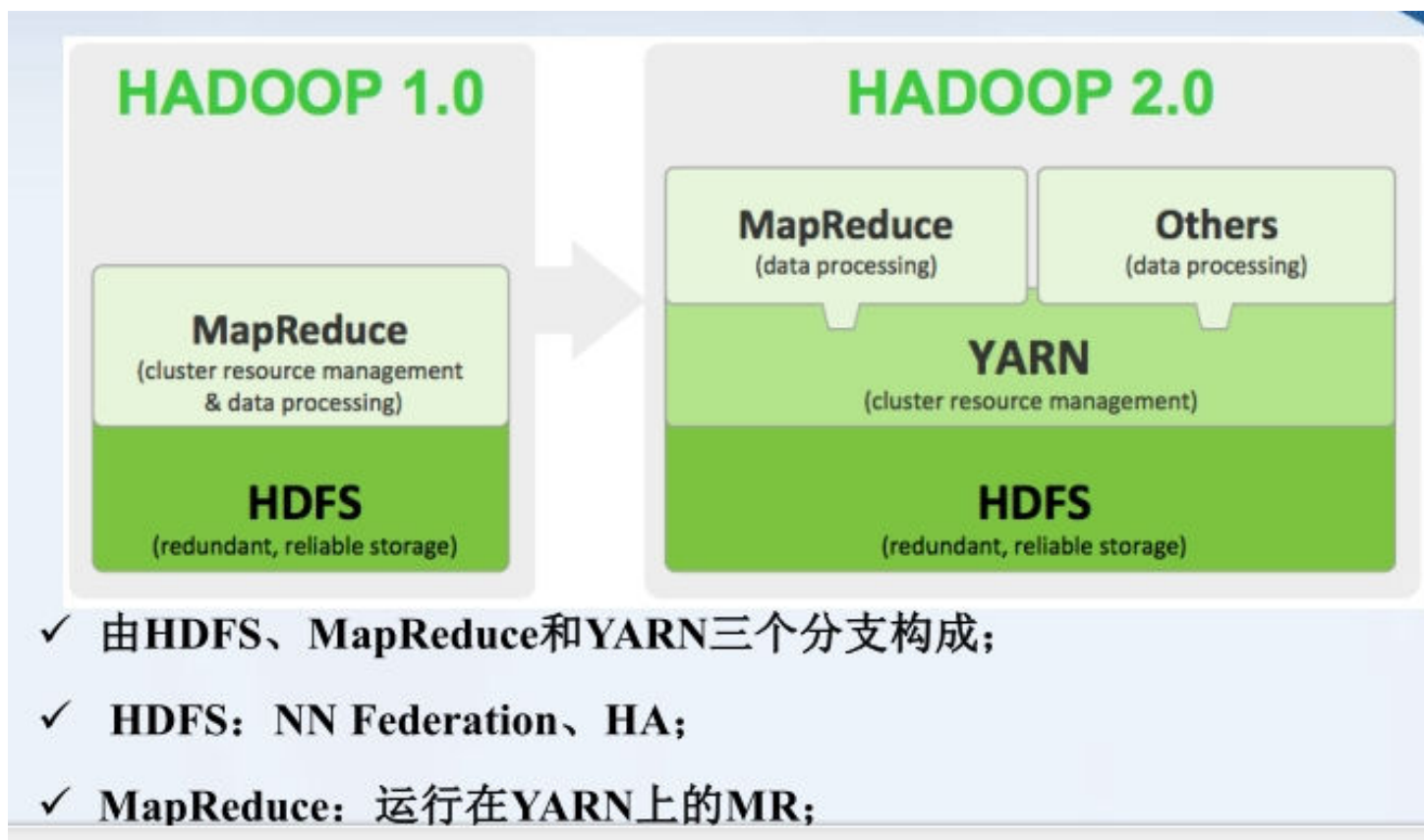
- **Hadoop**凭借其突出的优势，已经在各个领域得到了广泛的应用，而互联网领域是其应用的主阵地；
- **2007年**，雅虎在**Sunnyvale**总部建立了**M45**——一个包含了**4000**个处理器和**1.5PB**容量的**Hadoop**集群系统；
- **Facebook**作为全球知名的社交网站，**Hadoop**是非常理想的选择，**Facebook**主要将**Hadoop**平台用于日志处理、推荐系统和数据仓库等方面；
- 国内采用**Hadoop**的公司主要有百度、淘宝、网易、华为、中国移动等，其中，淘宝的**Hadoop**集群比较大。

Hadoop在企业中的应用架构



2.1.4 Apache Hadoop版本演变

- **Apache Hadoop**版本分为两代，我们将第一代**Hadoop**称为**Hadoop 1.0**，第二代**Hadoop**称为**Hadoop 2.0**；
- 第一代**Hadoop**包含三个大版本，分别是**0.20.x**，**0.21.x**和**0.22.x**，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**NameNode HA**等新的重大特性；
- 第二代**Hadoop**包含两个版本，分别是**0.23.x**和**2.x**，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**两个系统，相比于**0.23.x**，**2.x**增加了**NameNode HA**和**Wire-compatibility**两个重大特性。
- **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，也就是**Hadoop3.0**。



- Apache Hadoop
- Hortonworks
- Cloudera (CDH: Cloudera Distribution Hadoop)
- MapR
-

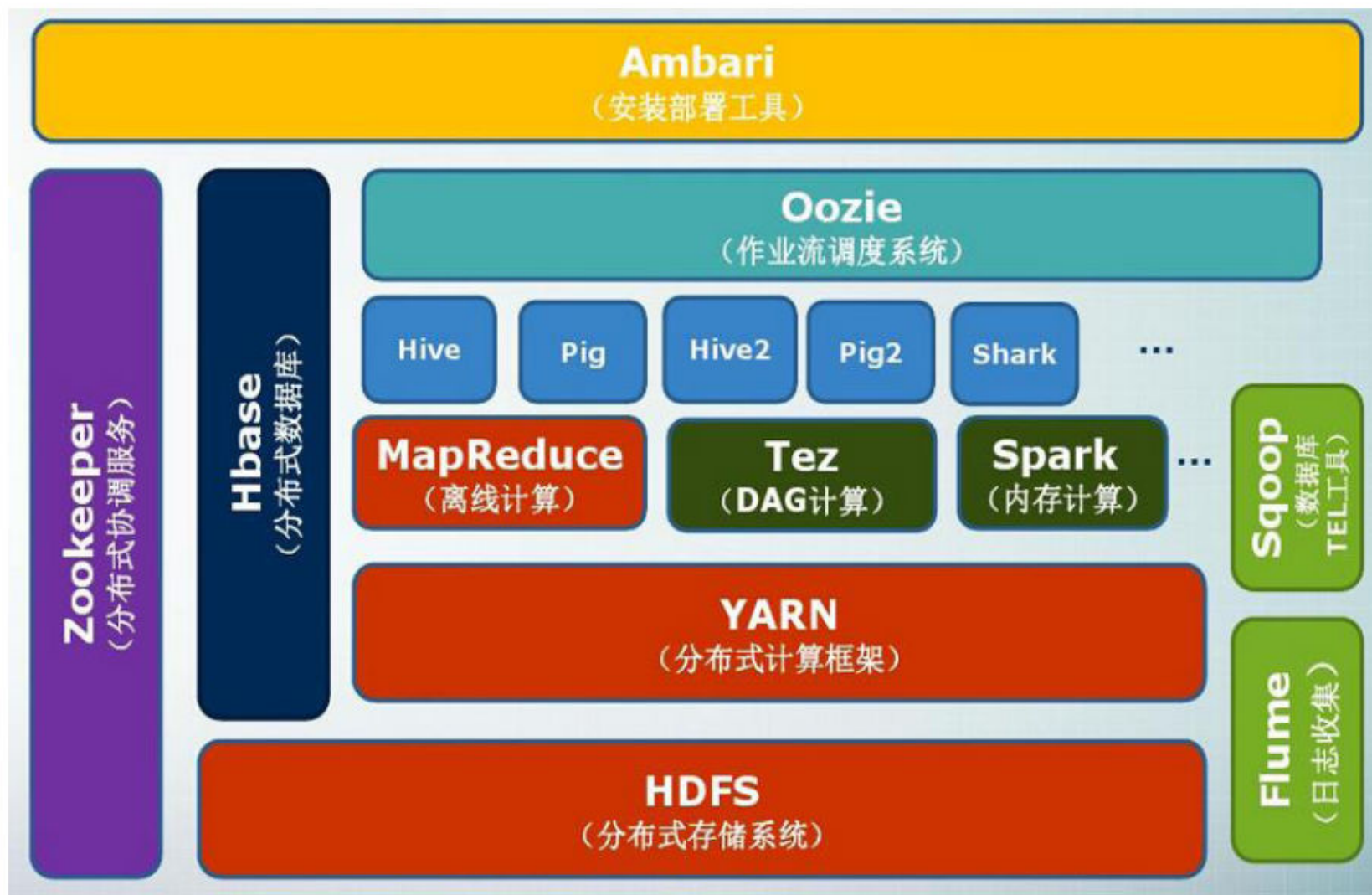
选择 **Hadoop** 版本的考虑因素:

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

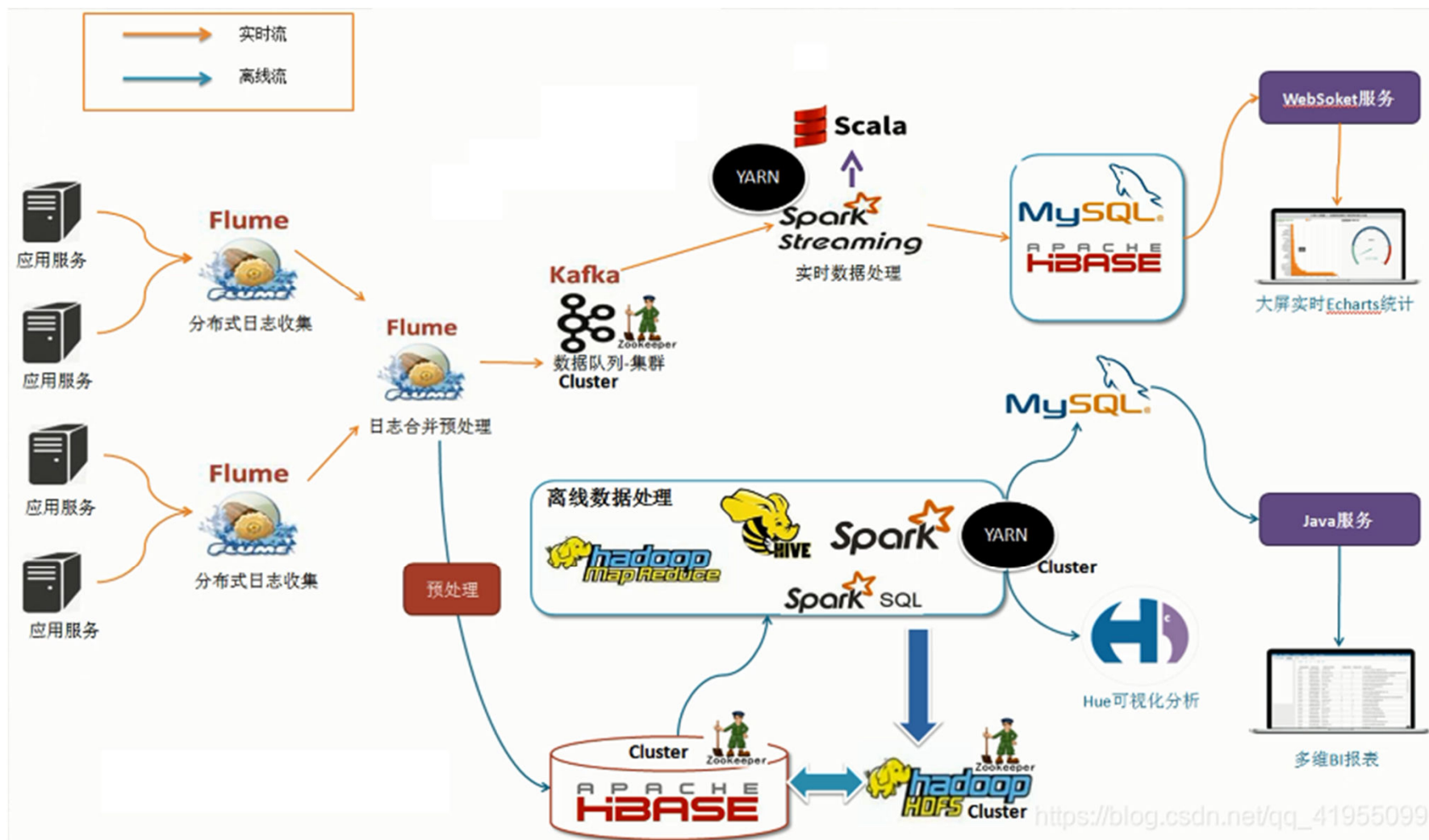
厂商名称	开放性	易用性 (★)	平台功能	性能 (★)	本地支持	总体评价 (★)
apache	完全开源、Hadoop就是托管在apache社区里面	安装: 2 使用: 2 维护: 2	Apache是标准的Hadoop平台，所有厂商都是在apache的平台上面进行改进	2	没有	2
cloudera	与Apache功能同步，部分代码开源	安装: 5 使用: 5 维护: 5	有自主研发的产品如：impala、navigator等	4.5	2014年刚进入中国，上海	4.5
hortonworks	与apache功能同步，也是完全开源	安装: 4.5 使用: 5 维护: 5	是apache hadoop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平台上面优化很多、从而形成自己的产品	5	没有	3.5
星环	核心组件与apache同步、底层的优化比较多、完全封闭的一个平台	安装: 5 使用: 4 维护: 4	有自主的Hadoop产品如 Inceptor、Hyperbase	4	本地厂商	4

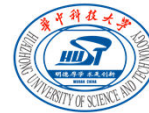
2.2 Hadoop项目结构

Hadoop的项目结构形成一个丰富的Hadoop生态系统



组件	功能
HDFS	分布式文件系统
MapReduce	分布式并行编程模型
YARN	资源管理和调度器
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
HBase	Hadoop 上的非关系型的分布式数据库
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	用于在 Hadoop 与传统数据库之间进行数据传递
Oozie	Hadoop 上的工作流管理系统
Zookeeper	提供分布式协调一致性服务
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	Hadoop 快速部署工具，支持 Apache Hadoop 集群的供应、管理和监控
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 的通用并行框架





2.3 Hadoop安装与使用

2.3.1 Hadoop安装预备知识

2.3.2 安装Linux虚拟机

2.3.3 安装双操作系统

2.3.4 详解Hadoop安装与使用

详细安装教程请参考厦门大学数据实验室建设的中国高校大数据课程公共服务平台上的技术文章：《大数据技术原理与应用（第3版） 第二章 大数据处理架构Hadoop 学习指南》，给出了每步安装命令和效果截图。

访问地址：<http://dbllab.xmu.edu.cn/blog/2544-2/>

Hadoop安装一览表

Windows系统:

- (1) Hadoop单机版
- (2) Linux虚拟机 (Hadoop单机、伪分布和分布版)
- (3) Linux双机系统 (Hadoop单机、伪分布和分布版)

Mac系统:

- (1) Hadoop单机版
- (2) Hadoop伪分布版
- (3) Hadoop分布版

Linux系统:

- (1) Hadoop单机版
- (2) Hadoop伪分布版
- (3) Hadoop分布版

2.3.1 Hadoop安装预备知识

(一) Linux的选择

(1) 选择哪个Linux发行版？

■在Linux系统各个发行版中，CentOS系统和Ubuntu系统在服务端和桌面端使用占比最高，网络上资料最是齐全，所以建议使用CentOS 或Ubuntu；

■在学习Hadoop方面，虽然两个系统没有多大区别，但是推荐使用Ubuntu操作系统。

(2) 选择32位还是64位？

■如果电脑比较老或者内存小于2G，那么建议选择32位系统版本的Linux；

■如果内存大于4G，那么建议选择64位系统版本的Linux。

（二）系统安装方式：选择虚拟机安装还是双系统安装

- 建议电脑比较新或者配置内存**4G**以上的电脑可以选择虚拟机安装；
- 电脑较旧或配置内存小于等于**4G**的电脑强烈建议选择双系统安装，否则，在配置较低的计算机上运行**Linux**虚拟机，系统运行速度会非常慢；
- 鉴于目前教师和学生的计算机硬件配置一般不高，建议在实践教学中采用双系统安装，确保系统运行速度。

（三）关于Linux的一些基础知识

■ Shell

是指“提供使用者使用界面”的软件（命令解析器），类似于DOS下的command和后来的cmd.exe。它接收用户命令，然后调用相应的应用程序；

■ 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；

■ 输入密码

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，读者不要误以为键盘没有响应；

■ 输入法中英文切换

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；

.....

(四) Hadoop安装方式

■单机模式:

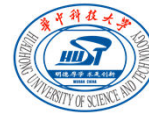
只在一台机器上运行，存储是采用本地文件系统，没有采用分布式文件系统**HDFS**;

■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;

■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位于不同机器上。



2.3.2 安装Linux虚拟机

(一) 材料和工具

(1) 下载VirtualBox虚拟机软件

<https://download.virtualbox.org/virtualbox/6.1.4/VirtualBox-6.1.4-136177-Win.exe>

(2) 下载Ubuntu LTS 16.04 (18.04, 20.04) ISO映像文件

Ubuntu LTS 16.04 下载:

<https://www.ubuntu.org.cn/download/ubuntu-kylin>

Ubuntu LTS 18.04 下载:

<https://ubuntu.com/download/desktop>

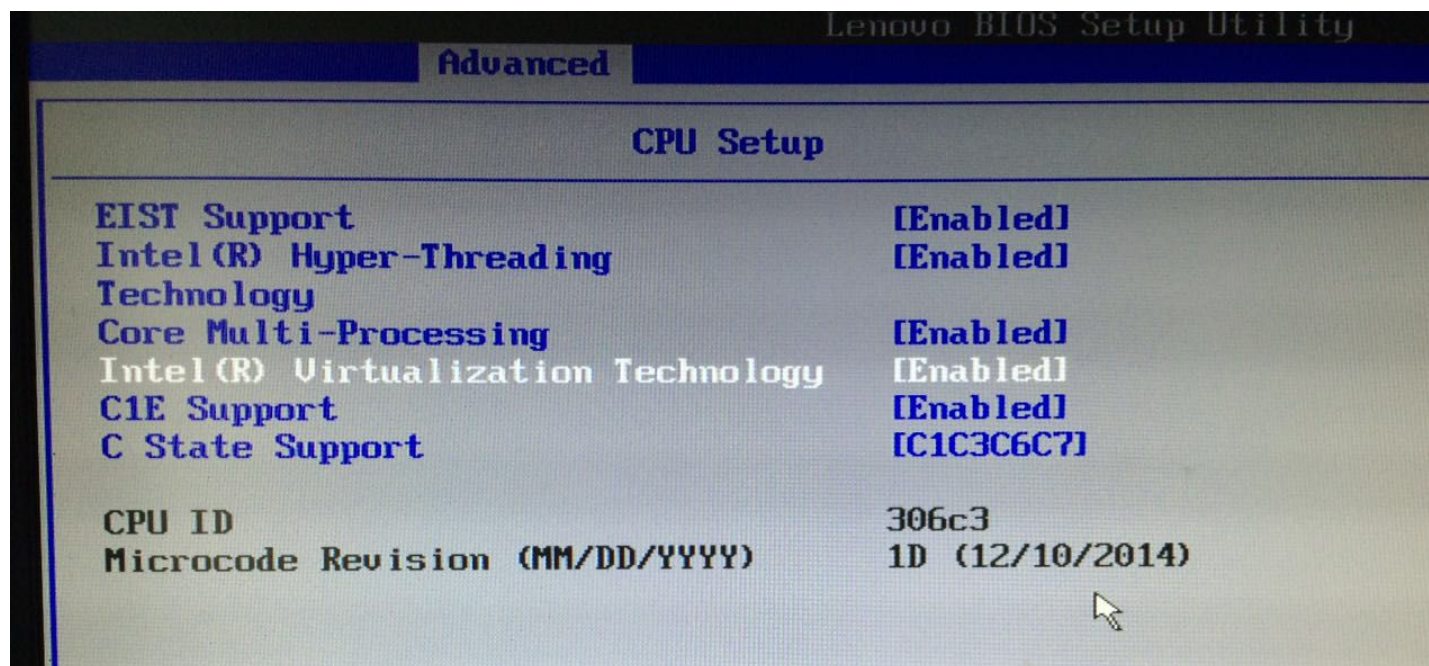
Ubuntu LTS 20.04 下载:

<https://ubuntu.com/download/desktop>

(二) 安装步骤

第一步：确认系统版本

如果选择的系统是**64位Ubuntu**系统，那么在安装虚拟机前，我们还要进入**BIOS**开启**CPU**的虚拟化。

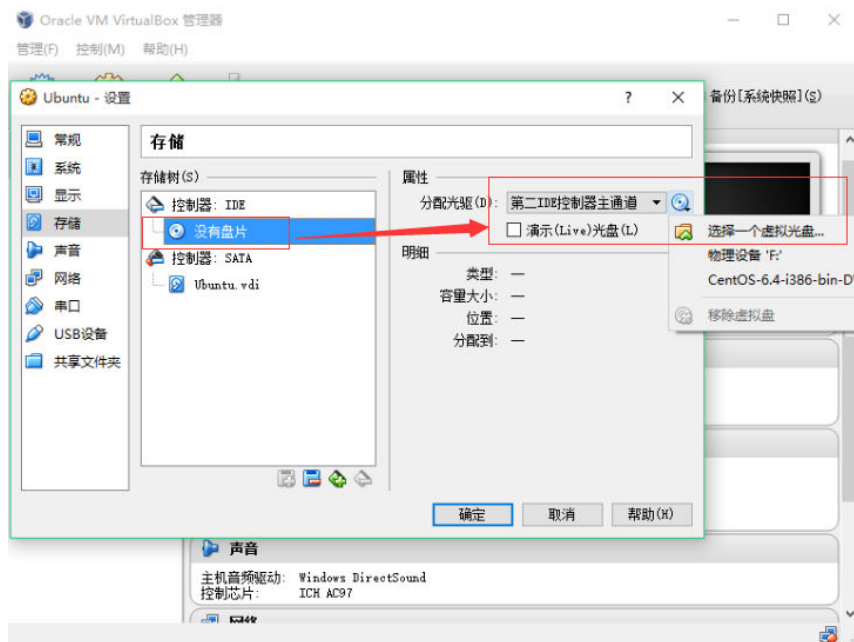


第二步：安装前的准备

- (1) 打开**VirtualBox**，点击“创建”按钮，创建一个虚拟机；
- (2) 给虚拟机命名，选择操作系统，版本；
- (3) 选择内存大小，这里设置的**1024M**；
- (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；
- (8) 点击创建。



第三步：安装Ubuntu



2.3.3 安装双操作系统

■ 第一步：制作安装U盘

具体可参考百度经验文章：

<http://jingyan.baidu.com/article/59703552e0a6e18fc007409f.html>

■ 第二步：双系统安装

具体可参考百度经验文章：

<http://jingyan.baidu.com/article/dca1fa6fa3b905f1a44052bd.html>

安装后**Window**和**Ubuntu 16.04(18.04, 20.04)**都可以用，默认**windows**优先启动；
可以在电脑启动时，选择进入**Ubuntu**系统而不是**Windows**系统

2.3.4 Hadoop安装与使用

Hadoop基本安装配置主要包括以下几个步骤:

- 第一步: 创建**Hadoop**用户
- 第二步: **SSH**登录权限设置
- 第三步: 安装**Java**环境
- 第四步: 单机安装配置
- 第五步: 伪分布式安装配置

详细安装配置过程请参考厦门大学数据库实验室出品教程, 《**Hadoop**安装教程_单机/伪分布式配置_Hadoop3.1.3/Ubuntu18.04》
<http://dblab.xmu.edu.cn/blog/2441-2/>

第一步：创建 **hadoop** 用户

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

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

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

2. 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：

```
$ sudo passwd hadoop
```

3. 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：

```
$ sudo adduser hadoop sudo
```

第二步：SSH登录权限设置

SSH是什么？

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**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**是由客户端和服务端的软件组成，服务端是一个守护进程(**daemon**)，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**（远程拷贝）、**slogin**（远程登陆）、**sftp**（安全文件传输）等其他的应用程序。

配置SSH的原因：

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，为了能够顺利登录每台机器，需要将所有机器配置为名称节点可以无密码登录它们。

Ubuntu默认已安装了**SSH**客户端，因此，这里还需要安装**SSH**服务端，请在**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
```


首先，请输入命令“**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，并将密钥加入到授权中，命令如下：

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

此时，再执行**ssh localhost**命令，无需输入密码就可以直接登录了，如图所示。

```
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
```


第三步：安装Java 环境

- Java环境可选择 Oracle 的 JDK，或是 OpenJDK
- 建议采用手工方式安装Java环境
- 具体请参考网络教程：<http://dbllab.xmu.edu.cn/blog/2441-2/>
- 到Java官网下载安装文件jdk-8u162-linux-x64.tar.gz
- 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，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  
$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
```

- 下面继续执行如下命令，设置环境变量：

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

- 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
```

- 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：

```
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)
```

第四步：单机安装配置

Hadoop 3 安装文件的下载：

Hadoop 3 可以到官网下载，需要下载 **hadoop-3.1.3.tar.gz** 这个格式的文件，这是编译好的，另一个包含 **src** 的则是 Hadoop 源代码，需要进行编译才可使用。

- 如果读者是使用虚拟机方式安装Ubuntu系统的用户，请用虚拟机中的Ubuntu自带firefox浏览器访问本指南，再点击下载地址，才能把hadoop文件下载到虚拟机ubuntu中。请不要使用Windows系统下的浏览器下载，文件会被下载到Windows系统中，虚拟机中的Ubuntu无法访问外部Windows系统的文件，造成不必要的麻烦。
- 如果读者是使用双系统方式安装Ubuntu系统的用户，请进去Ubuntu系统，在Ubuntu系统打开firefox浏览器，再点击下载。

■ 选择将 Hadoop 安装至 `/usr/local/` 中：

```
$ sudo tar -zxf ~/下载/hadoop-3.1.3.tar.gz -C /usr/local # 解压到/usr/local中
$ cd /usr/local/
$ sudo mv ./hadoop-3.1.3/ ./hadoop # 将文件夹名改为hadoop
$ sudo chown -R hadoop:hadoop ./hadoop # 修改文件权限
```

■ Hadoop 解压后即可使用。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
```

■ 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。

第五步：伪分布式安装配置

- **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。

伪分布式运行实验步骤:

- (1) 修改配置文件: **core-site.xml**, **hdfs-site.xml**, **mapred-site.xml**
- (2) 执行名称节点格式化: **hdfs namenode -format**
- (3) 启动**Hadoop**: **start-dfs.sh**
- (4) 访问**web**界面查看**Hadoop**信息: **http://localhost: 9870**
- (5) 运行**Hadoop**伪分布式实例
- (6) 关闭**Hadoop**: **stop-dfs.sh**

(1) 修改配置文件

修改配置文件 **core-site.xml**

```
<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>
```

- **hadoop.tmp.dir**表示存放临时数据的目录，即包括**NameNode**的数据，也包括**DataNode**的数据。该路径任意指定，只要实际存在该文件夹即可；
- **name**为**fs.defaultFS**的值，表示**hdfs**路径的逻辑名称。

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></configuration>
```

- **dfs.replication**表示副本的数量，伪分布式要设置为1
- **dfs.namenode.name.dir**表示本地磁盘目录，是存储**fsimage**文件的地方；
- **dfs.datanode.data.dir**表示本地磁盘目录，**HDFS**数据存放**block**的地方。

(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

```
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
*****
*****/
```

(3) 启动Hadoop

执行下面命令启动Hadoop:

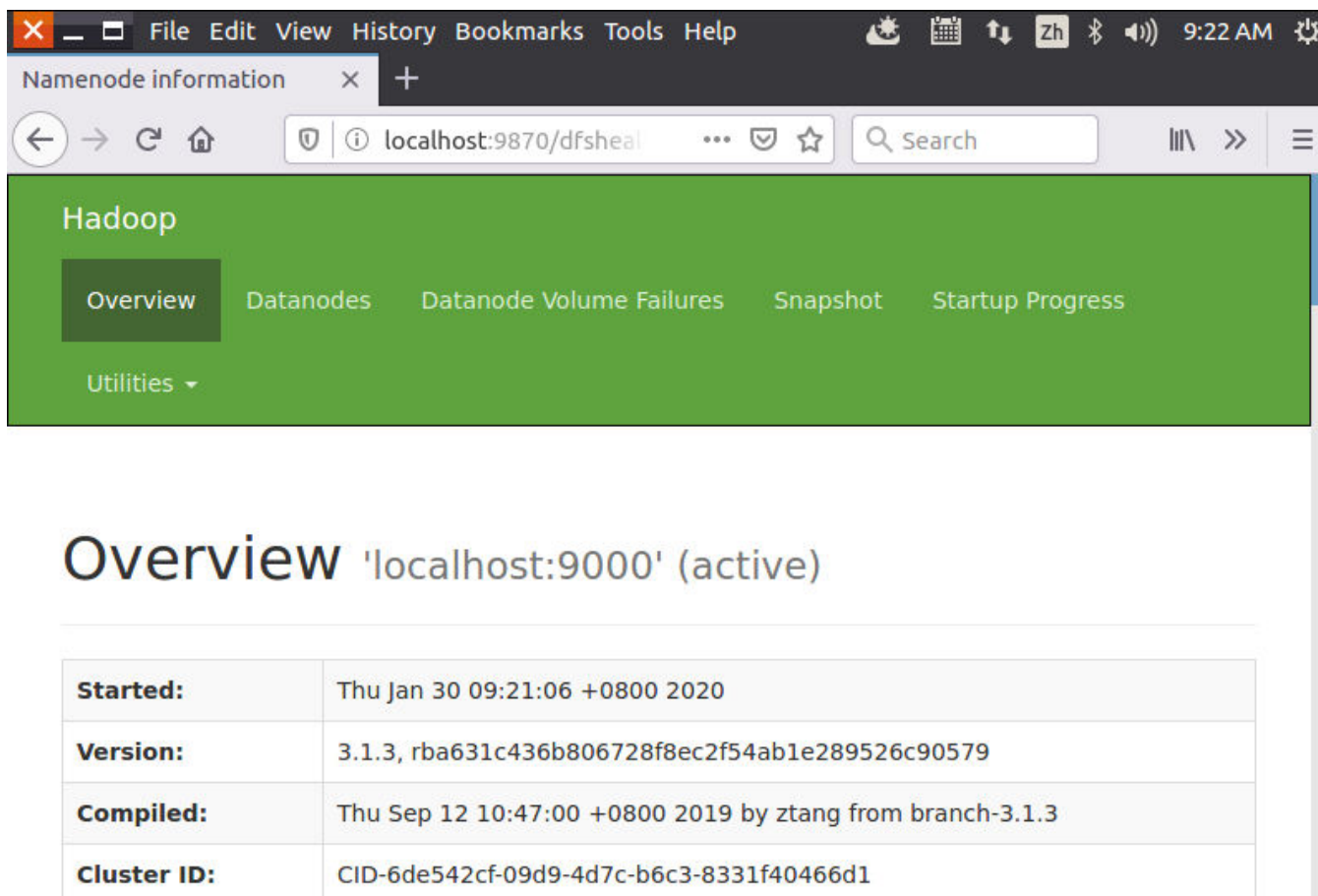
```
$ cd /usr/local/hadoop  
$ ./sbin/start-dfs.sh #start-dfs.sh是个完整的可执行文件，中间没有空格
```

如果出现如图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
```

如果hadoop启动完成，输入命令：**jps**，出现**NameNode**、**DataNode**、**secondaryNameNode**就表示Hadoop启动成功！

(4) 使用Web界面查看HDFS信息 (http://localhost:9870)



The screenshot shows a web browser window with the Hadoop NameNode information page. The browser's address bar shows the URL `localhost:9870/dfshealth.html`. The page has a green header with the word "Hadoop" and a navigation bar with tabs: "Overview", "Datanodes", "Datanode Volume Failures", "Snapshot", and "Startup Progress". The "Overview" tab is selected. Below the navigation bar, the page title is "Overview 'localhost:9000' (active)". A table displays the following information:

Started:	Thu Jan 30 09:21:06 +0800 2020
Version:	3.1.3, rba631c436b806728f8ec2f54ab1e289526c90579
Compiled:	Thu Sep 12 10:47:00 +0800 2019 by ztang from branch-3.1.3
Cluster ID:	CID-6de542cf-09d9-4d7c-b6c3-8331f40466d1

(5) 运行Hadoop伪分布式实例

要使用**HDFS**，首先需要在**HDFS**中创建用户目录（本教程全部统一采用**hadoop**用户名登录**Linux**系统），命令如下：

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

接着需要把本地文件系统的

“**/usr/local/hadoop/etc/hadoop**”目录中的所有**xml**文件作为输入文件，复制到分布式文件系统**HDFS**中的“**/user/hadoop/input**”目录中，命令如下：

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

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
```

(6) 关闭Hadoop

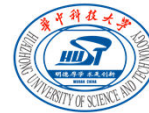
如果要关闭**Hadoop**，可以执行下面命令：

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

关于三种**Shell**命令方式的区别:

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。



2.4 Hadoop集群部署与使用

2.4.1 集群节点类型

2.4.2 集群硬件配置

2.4.3 集群规模

2.4.4 集群网络拓扑

2.4.5 集群建立与安装

2.4.6 Hadoop集群基准测试

2.4.7 在云计算环境中使用Hadoop

2.4.1 集群节点类型

- **Hadoop**框架中最核心的设计是为海量数据提供存储的**HDFS**和对数据进行计算的**MapReduce**;
- **MapReduce**的作业主要包括：（1）从磁盘或从网络读取数据，即**IO**密集工作；（2）计算数据，即**CPU**密集工作；
- **Hadoop**集群的整体性能取决于**CPU**、内存、网络以及存储之间的性能平衡。因此运营团队在选择机器配置时要针对不同的工作节点选择合适硬件类型；
- 一个基本的**Hadoop**集群中的节点主要有
 - **NameNode**: 负责协调集群中的数据存储；
 - **DataNode**: 存储被拆分的数据块；
 - **JobTracker**: 协调数据计算任务；
 - **TaskTracker**: 负责执行由**JobTracker**指派的任务；
 - **SecondaryNameNode**: 帮助**NameNode**收集文件系统运行的状态信息。

2.4.2 集群硬件配置

在集群中，大部分的机器设备是作为**Datanode**和**TaskTracker**工作的**Datanode/TaskTracker**的硬件规格可以采用以下方案：

- 4个磁盘驱动器（单盘1-2T），支持JBOD(Just a Bunch Of Disks，磁盘簇)
- 2个4核CPU,至少2-2.5GHz
- 16-24GB内存
- 千兆以太网

NameNode提供整个**HDFS**文件系统的**NameSpace**(命名空间)管理、块管理等所有服务，因此需要更多的**RAM**，与集群中的数据块数量相对应，并且需要优化**RAM**的内存通道带宽，采用双通道或三通道以上内存。硬件规格可以采用以下方案：

- 8-12个磁盘驱动器（单盘1-2T）
- 2个4核/8核CPU
- 16-72GB内存
- 千兆/万兆以太网

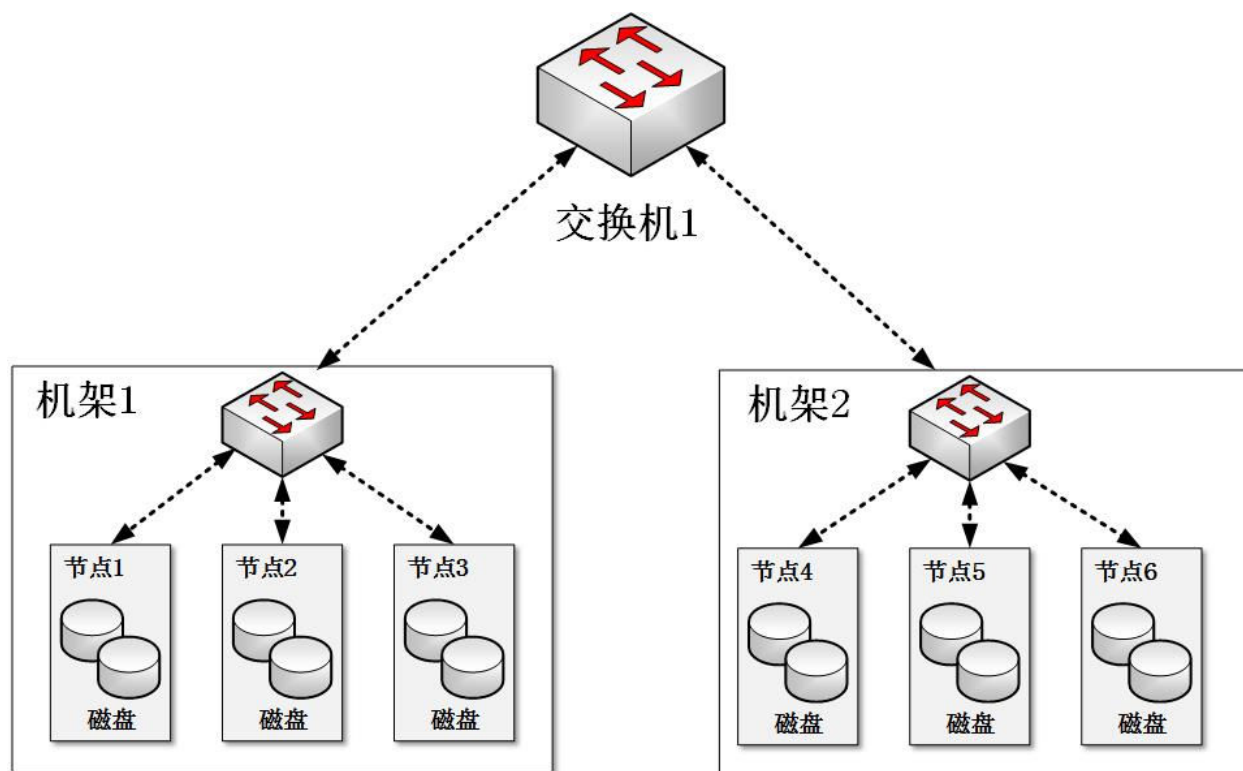
SecondaryNameNode在小型集群中可以**NameNode**共用一台机器，较大的群集可以采用与**NameNode**相同的硬件。

2.4.3 集群规模

- **Hadoop**集群规模可大可小，初始时，可以从一个较小规模的集群开始，比如包含**10**个节点，然后，规模随着存储器和计算需求的扩大而扩大；
- 如果数据每周增大**1TB**，并且有三个**HDFS**副本，然后每周需要一个额外的**3TB**作为原始数据存储。要允许一些中间文件和日志（假定**30%**）的空间，由此，可以算出每周大约需要增加一台新机器。存储两年数据的集群，大约需要**100**台机器；
- 对于一个小的集群，名称节点（**NameNode**）和**JobTracker**运行在单个节点上，通常是可以接受的。但是，随着集群和存储在**HDFS**中的文件数量的增加，名称节点需要更多的主存，这时，名称节点和**JobTracker**就需要运行在不同的节点上；
- 第二名称节点（**SecondaryNameNode**）会和名称节点可以运行在相同的机器上，但是，由于第二名称节点和名称节点几乎具有相同的主存需求，因此，二者最好运行在不同节点上。

2.4.4 集群网络拓扑

- 普通的Hadoop集群结构由一个两阶网络构成；
- 每个机架（**Rack**）有**30-40**个服务器，配置一个**1GB**的交换机，并向上传输到一个核心交换机或者路由器（**1GB**或以上）；
- 在相同的机架中的节点间的带宽的总和，要大于不同机架间的节点间的带宽总和。



2.4.5 集群建立与安装

采购好相关的硬件设备后，就可以把硬件装入机架，安装并运行**Hadoop**。

安装**Hadoop**有多种方法：

(1) 手动安装

(2) 自动化安装

- 为了缓解安装和维护每个节点上相同的软件的负担，可以使用一个自动化方法实现完全自动化安装，比如**Red Hat Linux' Kickstart**、**Debian**或者**Docker**;
- 自动化安装部署工具，会通过记录在安装过程中对于各个选项的回答来完成自动化安装过程。

2.4.6 Hadoop集群基准测试

- 如何判断一个**Hadoop**集群是否已经正确安装？可以运行基准测试；
- **Hadoop**自带有一些基准测试程序，被打包在测试程序**JAR**文件中；
- 用**TestDFSIO**基准测试，来测试**HDFS**的**IO**性能；
- 用排序测试**MapReduce**：**Hadoop**自带一个部分排序的程序，这个测试过程的整个数据集都会通过洗牌（**Shuffle**）传输至**Reducer**，可以充分测试**MapReduce**的性能。

2.4.7 云计算环境中使用Hadoop

- **Hadoop**不仅可以运行在企业内部的集群中，也可以运行在云计算环境中；
- 可以在**Amazon EC2**中运行**Hadoop**。**EC2**是一个计算服务，允许客户租用计算机（实例），来运行自己的应用。客户可以按需运行或终止实例，并且按照实际使用情况来付费；
- **Hadoop**自带有一套脚本，用于在**EC2**上面运行**Hadoop**；
- 在**EC2**上运行**Hadoop**尤其适用于一些工作流。例如，在**Amazon S3**中存储数据，在**EC2**上运行集群，在集群中运行**MapReduce**作业，读取存储在**S3**中的数据，最后，在关闭集群之前将输出写回**S3**中；如果长期使用集群，复制**S3**数据到运行在**EC2**上的**HDFS**中，则可以使得数据处理更加高效，因为，**HDFS**可以充分利用数据的位置，**S3**则做不到，因为，**S3**与**EC2**的存储不在同一个节点上。

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**Common**、**Avro**、**Zookeeper**、**HDFS**、**MapReduce**、**HBase**、**Hive**、**Chukwa**、**Pig**等子项目，其中，**HDFS**和**MapReduce**是**Hadoop**的两大核心组件；
- 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。