



《大数据基础》

第2章 大数据处理架构Hadoop

洪韬

E-mail: 2993400893@qq.com ▶▶



提纲

- **2.1 概述**
- **2.2 Hadoop项目结构**
- **2.3 Hadoop的安装与使用**
- **2.4 Hadoop集群的部署与使用**



2.1 概述

- 2.1.1 Hadoop简介
- 2.1.2 Hadoop发展简史
- 2.1.3 Hadoop的特性
- 2.1.4 Hadoop的应用现状



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发展简史



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



2.1.2 Hadoop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



2.1.3 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，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



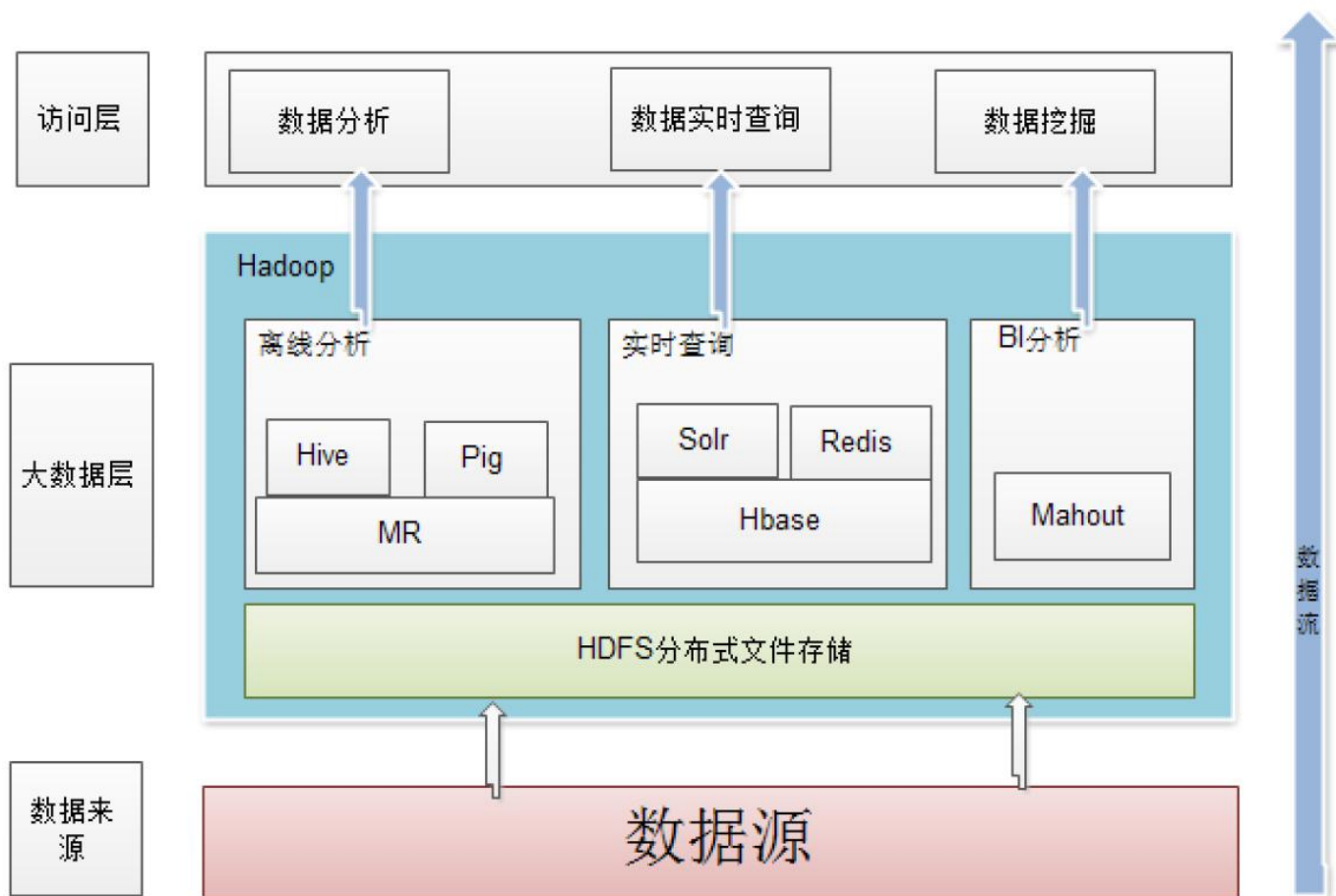
2.1.3 Hadoop的应用现状

- 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



2.1.3 Hadoop的应用现状

Hadoop在企业中的应用架构



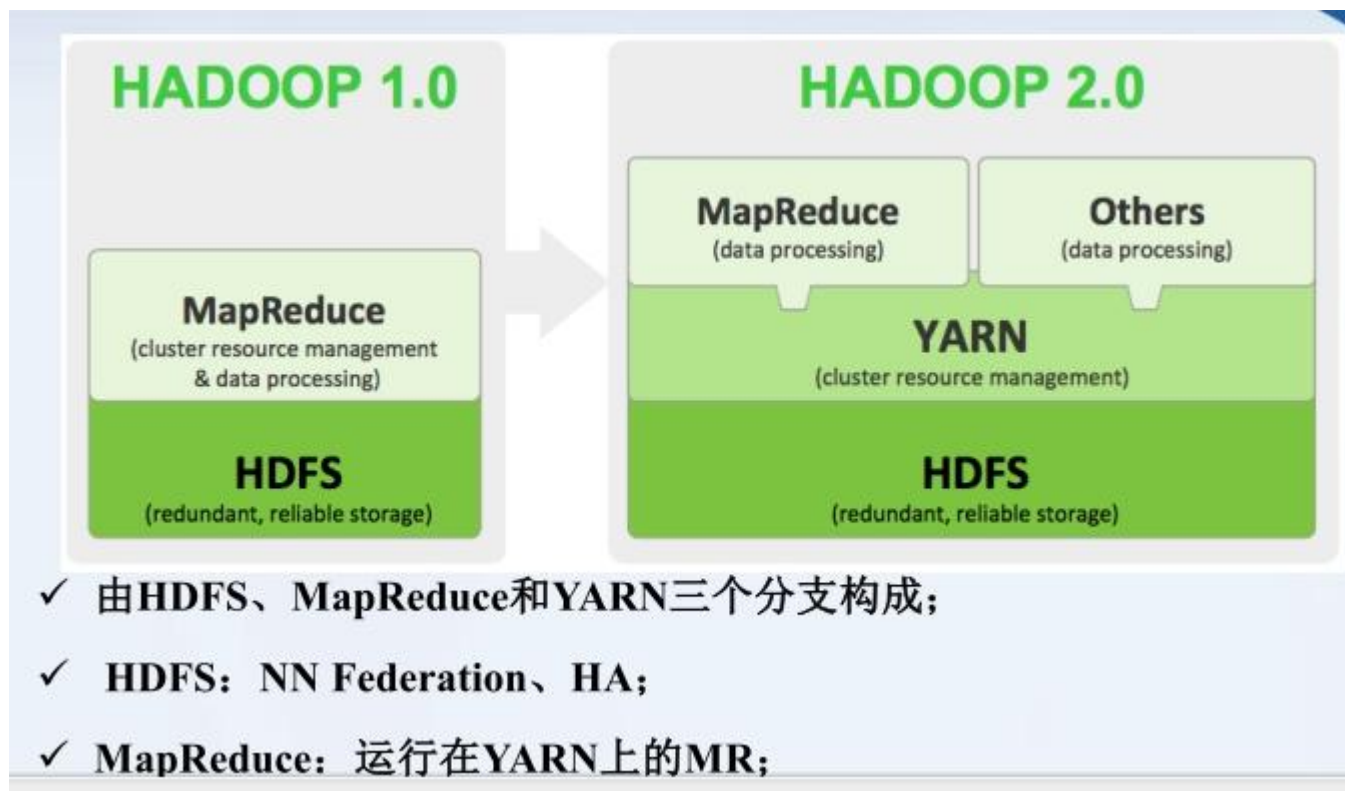


2.1.4 Apache Hadoop版本演变

- Apache Hadoop版本分为三代，分别是Hadoop1.0, Hadoop2.0, Hadoop3.0
- 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(high Availability)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。Hadoop3.0中引入了一些重要功能和优化，包括HDFS可擦除编码、多名称节点支持等。



2.1.4 Apache Hadoop版本演变





2.1.5 Hadoop各种版本

- Apache Hadoop（开源版本）
- 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
- 是否经实践检验
- 是否有强大的社区支持



2.1.5 Hadoop各种版本

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的项目结构不断丰富发展，已经形成一个丰富的Hadoop生态系统。根据Apache Hadoop官网给出的资料，目前Hadoop有四个核心模块，以及十三个与Hadoop相关联的模块。事实上，与Hadoop相关的组件不止列出的这些，还包括storm, Kafka, Impala, Azkaban等，这其中有Apache基金会下的开源项目，也有商业公司提供的组件工具。

Modules

The project includes these modules:

- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

Who Uses Hadoop?

A wide variety of companies and organizations use Hadoop for both research and production. Users are encouraged to add themselves to the Hadoop

[PoweredBy wiki page](#).

Related projects

Other Hadoop-related projects at Apache include:

- **Ambari™**: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig and Sqoop. Ambari also provides a dashboard for viewing cluster health such as heatmaps and ability to view MapReduce, Pig and Hive applications visually alongwith features to diagnose their performance characteristics in a user-friendly manner.
- **Avro™**: A data serialization system.
- **Cassandra™**: A scalable multi-master database with no single points of failure.
- **Chukwa™**: A data collection system for managing large distributed systems.
- **HBase™**: A scalable, distributed database that supports structured data storage for large tables.
- **Hive™**: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- **Mahout™**: A Scalable machine learning and data mining library.
- **Ozone™**: A scalable, redundant, and distributed object store for Hadoop.
- **Pig™**: A high-level data-flow language and execution framework for parallel computation.
- **Spark™**: A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a

<https://hadoop.apache.org/>



2.2 Hadoop项目结构



Apache Pig



Apache Oozie



Apache Impala



2.2 Hadoop项目结构

组件	功能
HadoopCommon	The common utilities that support the other Hadoop modules.
HDFS	Hadoop Distributed File System. A distributed file system that provides high-throughput access to application data.
YARN	A framework for job scheduling and cluster resource management.
MapReduce	A YARN-based system for parallel processing of large data sets.(Hadoop2.0之后)
Tez	A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases.
Spark	A fast and general compute engine for Hadoop data.
Storm	Apache Storm is a free and open source distributed realtime computation system.
Kafka	Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.
Flume	Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.
Zookeeper	ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.

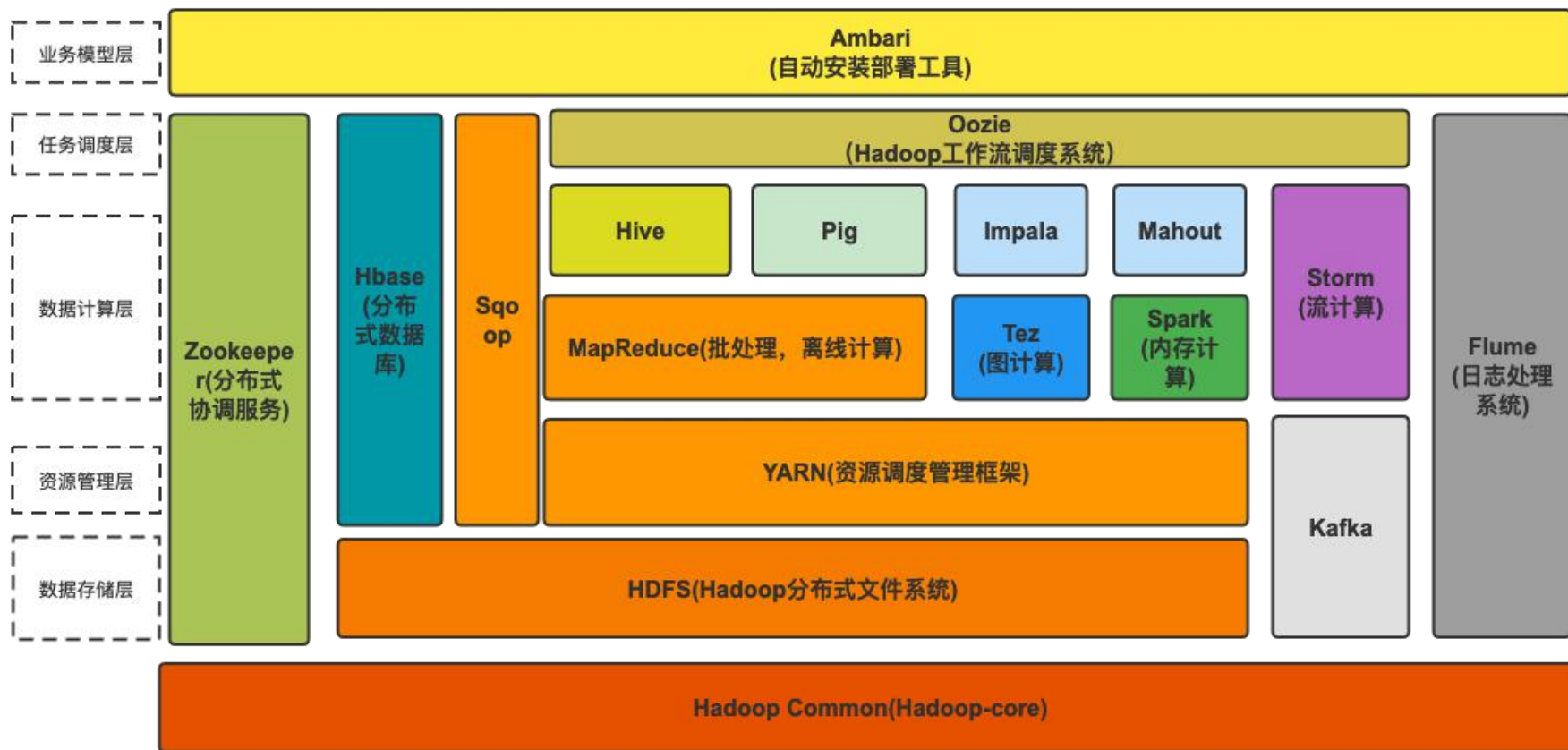


2.2 Hadoop项目结构

组件（续表）	功能（续表）
HBase	Apache HBase is an open-source, distributed, versioned, non-relational database which provide random, realtime read/write access to your Big Data.
Sqoop	Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases. Retired in 2021-06.
Hive	A data warehouse infrastructure that provides data summarization and ad hoc querying.
Pig	Apache Pig is a platform for analyzing large data sets that consists of a high-level language,Pig Latin, for expressing data analysis programs, coupled with infrastructure for evaluating these programs.
Impala	Impala raises the bar for SQL query performance on Apache Hadoop while retaining a familiar user experience. With Impala, you can query data, whether stored in HDFS or Apache HBase – including SELECT, JOIN, and aggregate functions – in real time.
Mahout	Apache Mahout is a distributed linear algebra framework and mathematically expressive Scala DSL designed to let mathematicians, statisticians, and data scientists quickly implement their own algorithms.
Oozie	Oozie is a workflow scheduler system to manage Apache Hadoop jobs.
Ambari	The Apache Ambari project is aimed at making Hadoop management simpler by developing software for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an intuitive, easy-to-use Hadoop management web UI backed by its RESTful APIs.

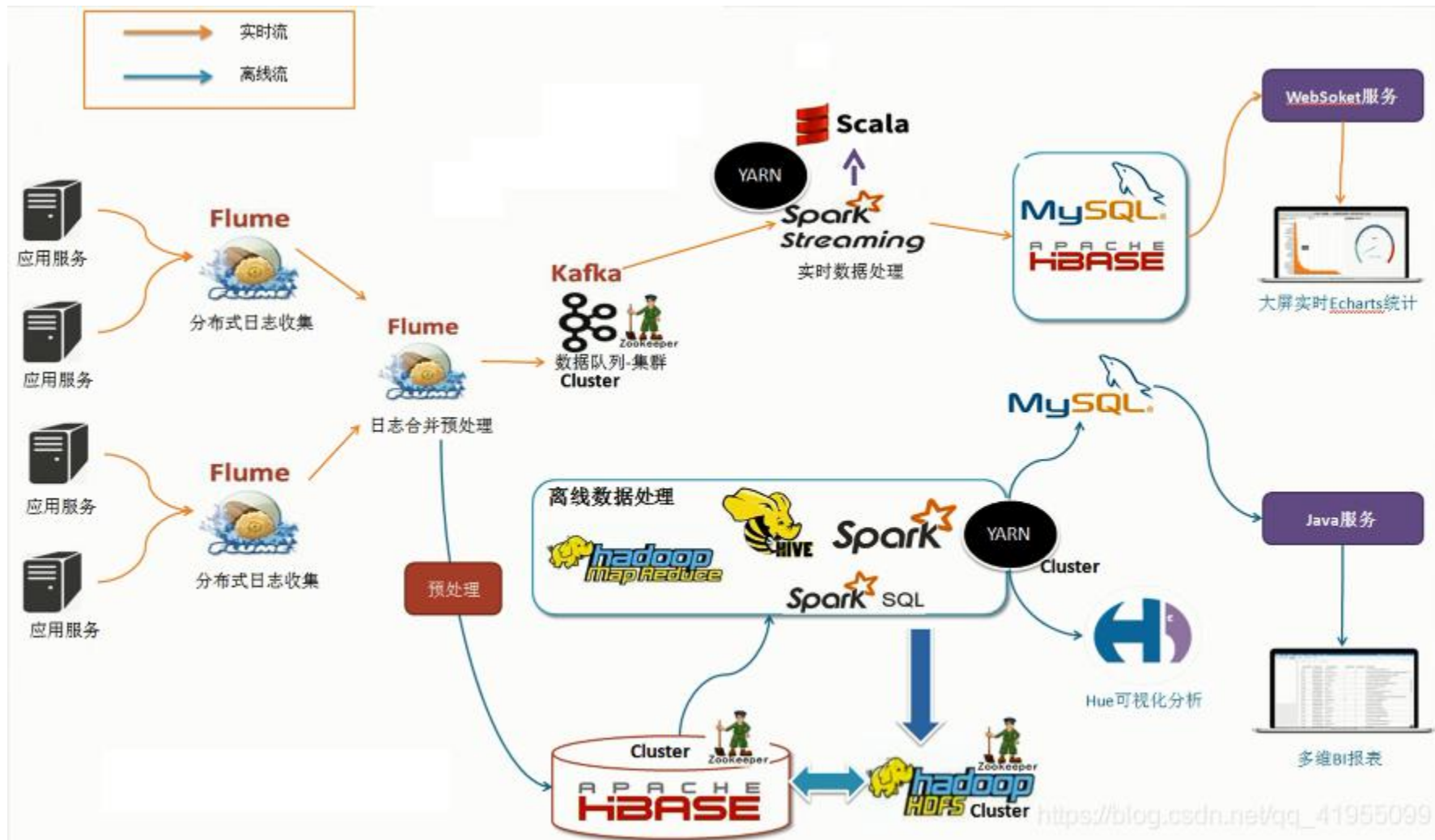


2.2 Hadoop项目结构





2.2 Hadoop项目结构





2.3 Hadoop的安装与使用

- 2.3.1 Hadoop安装之前的预备知识
- 2.3.2 安装Linux虚拟机
- 2.3.3 详解Hadoop的安装与使用



2.3.1 Hadoop安装之前的预备知识

(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



2.3.1 Hadoop安装之前的预备知识

(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



2.3.1 Hadoop安装之前的预备知识

(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



2.3.1 Hadoop安装之前的预备知识

(四) Hadoop安装方式

•Hadoop包括三种安装模式:

- 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（或18.04）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>



2.3.2 安装Linux虚拟机

(二)安装前的准备

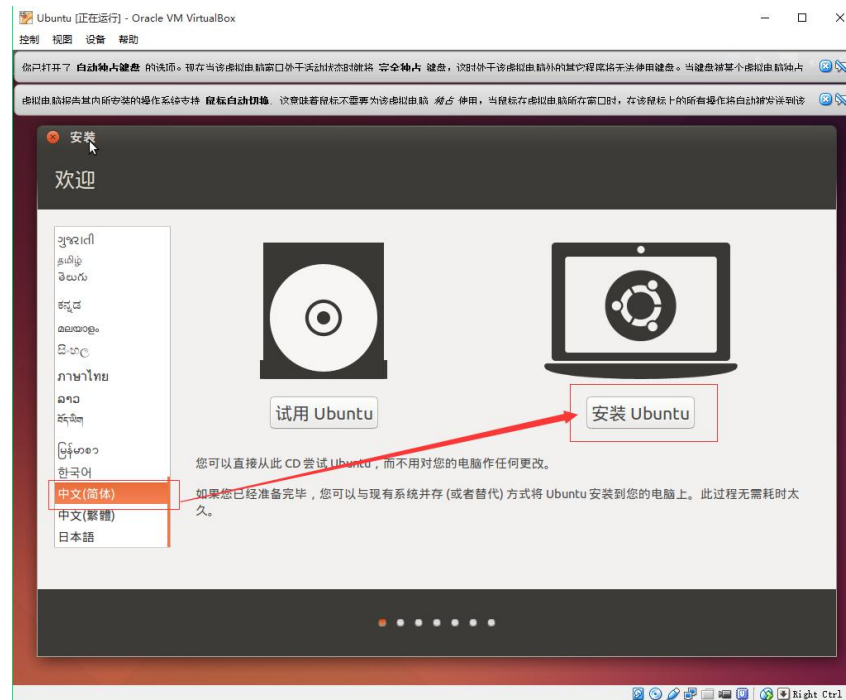
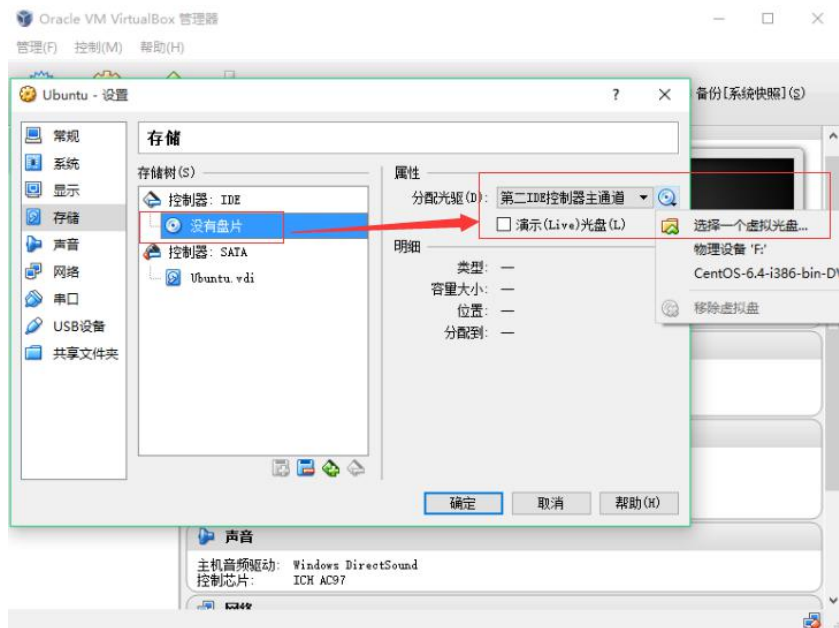
- 1.打开VirtualBox, 点击“创建”按钮, 创建一个虚拟机
- 2.给虚拟机命名, 选择操作系统, 版本
- 3.选择内存大小, 这里设置的1024M
- 4.创建虚拟硬盘
- 5.选择虚拟硬盘文件类型VDI
- 6.虚拟硬盘选择动态分配
- 7.选择文件存储的位置和容量大小
- 8.点击创建





2.3.2 安装Linux虚拟机

(三)安装Ubuntu





2.3.4 Hadoop的安装与使用

Hadoop基本安装配置主要包括以下几个步骤:

- 创建Hadoop用户
- SSH登录权限设置
- 安装Java环境
- 单机模式
- 伪分布式模式
- 分布式模式



创建Hadoop用户

如果安装 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：

```
$ 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

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
```

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
```



虚拟机网络配置

```
ht@Slave03: ~  
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...  
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...  
Processing triggers for man-db (2.9.1-1) ...  
ht@Slave03:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    inet6 fe80::8e12:5a7b:3f4d:4183 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:99:60:8e txqueuelen 1000 (Ethernet)  
    RX packets 312453 bytes 453265264 (453.2 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 73221 bytes 4488815 (4.4 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 383 bytes 33563 (33.5 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 383 bytes 33563 (33.5 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
ht@Slave03:~$
```


知乎 @一场有相逢

在VirtualBox环境下，可提供给虚拟机的Network方式主要是NAT、Bridged Adapter、Internal、Host-only Adapter四种方式；



虚拟机网络配置

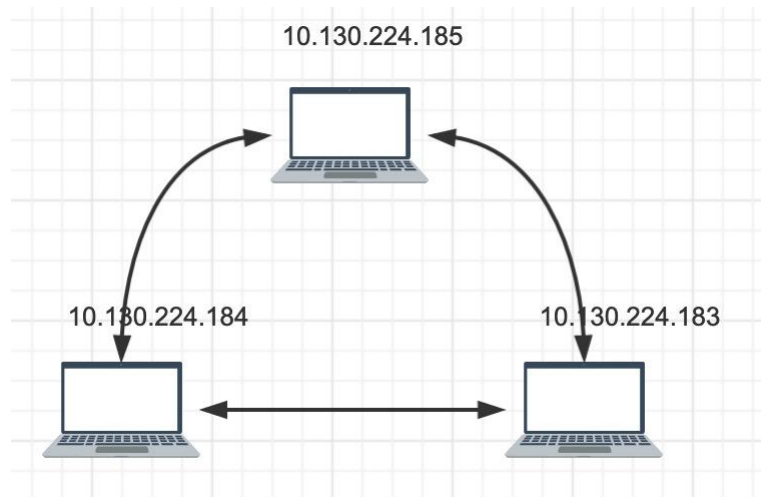
```
PING 10.130.224.185 (10.130.224.185): 56 data bytes
64 bytes from 10.130.224.185: icmp_seq=0 ttl=64 time=0.364 ms
64 bytes from 10.130.224.185: icmp_seq=1 ttl=64 time=1.213 ms
64 bytes from 10.130.224.185: icmp_seq=2 ttl=64 time=0.653 ms
64 bytes from 10.130.224.185: icmp_seq=3 ttl=64 time=0.634 ms
^C
--- 10.130.224.185 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
```



```
hadoop@master:~$ ping 10.131.124.83
PING 10.131.124.83 (10.131.124.83) 56(84) bytes of data.
64 bytes from 10.131.124.83: icmp_seq=1 ttl=64 time=0.368 ms
64 bytes from 10.131.124.83: icmp_seq=2 ttl=64 time=0.192 ms
64 bytes from 10.131.124.83: icmp_seq=3 ttl=64 time=0.511 ms
^C
--- 10.131.124.83 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.192/0.357/0.511/0.130 ms
hadoop@master:~$
```

虚拟机节点之间也是可以通过网络是联通的，为了后面Hadoop节点之间通信方便，我们修改Hosts文件信息

10.130.224.185 master
10.130.224.184 slave01
10.130.224.183 slave02





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
```




安装Java环境

- 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
```



安装Java环境

- 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)
```



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。



SSH登录权限设置

1、安装SSH，测试登录本地

```
sudo apt-get install openssh-server //安装ssh-server  
ssh localhost //验证是否可以登录本地
```

2、在master节点生成公钥:

```
cd ~/.ssh //如果提示找不到文件或路径，先执行一次ssh localhost  
rm ./id_rsa* //如果有，则删除之前的公钥  
ssh-keygen -t rsa //一路回车生成公钥
```

3、将公钥传到slave01、slave02、slave03等节点上

```
scp ~/.ssh/id_rsa.pub hadoop@slave03:/home/hadoop
```

4、slave节点将公钥加入到authorized_keys中



SSH登录权限设置

```
hadoop@master:~$ ssh slave03
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-43-generic x86_64)

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

83 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Jan  3 09:19:17 2022 from 127.0.0.1
hadoop@Slave03:~$ ls
```

Desktop Downloads Music Public Videos

知乎 @一场有相逢



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



单机安装配置

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。



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



伪分布式安装配置

实验步骤:

- ❑ 修改配置文件: core-site.xml, hdfs-site.xml, mapred-site.xml
- ❑ 初始化文件系统hadoop namenode -format
- ❑ 启动所有进程start-all.sh
- ❑ 访问web界面, 查看Hadoop信息
- ❑ 运行实例



伪分布式安装配置

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



伪分布式安装配置

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>
</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



分布式安装配置

1、在master节点安装Hadoop。在Ubuntu系统虚拟机中通过Hadoop官网中下载Hadoop3.3.1的binary版本Apache Hadoop。再将Hadoop解压，流程如下：

```
sudo tar -zxvf~/download/hadoop-3.3.1.tar.gz -C /usr/local //将Hadoop解压到/usr/local
cd /usr/local
sudo mv ./hadoop-3.3.1 ./hadoop //修改文件名为hadoop
sudo chown -R hadoop ./hadoop //修改文件权限
```

安装Hadoop后，直接执行hadoop命令是无效的，和jdk一样，需要设置环境变量，才可以直接使用hadoop 以及后面的hadoop启动命令。

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_HOME}/lib/native
export PATH=${JAVA_HOME}/bin:${HADOOP_HOME}/bin:${HADOOP_HOME}/sbin:$PATH
```

<https://zhuanlan.zhihu.com/p/450264578>



分布式安装配置

2、hadoop配置文件设置。我们要利用virtualbox虚拟机部署完全分布式Hadoop环境，那么master节点的设置，从节点数量的设置以及备份的设置都在配置文件中体现。这里我们主要修改：core-site.xml、hdfs-site.xml、yarn-site.xml、mapred-site.xml文件，这些文件都在/usr/local/hadoop/etc/hadoop目录下。

除此之外，还需要**修改当前目录下的workers文件**，workers是用来设置datanode节点。workers文件默认为localhost，即默认情况下，namenode和datanode都被部署在同一个节点上，而我们要做完全分布式Hadoop环境，所以，需要将workers中的内容修改为slave节点名称。

```
slave01  
slave02
```

到此，Hadoop在master节点上的安装和配置已经完成，接下来我们需要将相同的配置在slave节点上重新配置；



分布式安装配置

3、slave节点中Hadoop配置同步

将master节点中/usr/local/hadoop中的内容打包压缩，然后发送到各个slave节点中；

```
cd /usr/local
sudo rm -r ./hadoop/tmp          //删除Hadoop路径下的临时文件
sudo rm -r ./hadoop/logs         //删除Hadoop路径下的日志文件
tar -zcf ~/hadoop.master.tar.gz ./hadoop //将hadoop目录下的内容打包复制到~
scp ~/hadoop.master.tar.gz slave03:/home/hadoop //将打包内容复制到slav
```

在slave节点上解压hadoop.master.tar.gz文件，并给予授权；

```
sudo rm -r /usr/local/hadoop //删除之前版本
sudo tar -zxf ~/hadoop.master.tar.gz -C /usr/local //解压文件到 /usr/
sudo chown -R hadoop /usr/local/hadoop //授权
```

对集群中的其他slave节点做同样的操作即可；



启动和停止

在master节点上依次执行start-dfs.sh、start-yarn.sh等命令，可以分别在master节点和slave节点上看到对应调起的线程，用jps命令查看如下图所示；也可以直接执行start-all.sh命令；

```
hadoop@master:/usr/local/hadoop$ start-dfs.sh
Starting namenodes on [master]
Starting datanodes
slave01: WARNING: /usr/local/hadoop/logs does not exist. Creating
.
Starting secondary namenodes [master]
2022-01-15 11:22:08,972 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@master:/usr/local/hadoop$ jps
14282 Jps
13948 NameNode
14174 SecondaryNameNode
hadoop@master:/usr/local/hadoop$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@master:/usr/local/hadoop$ jps
14649 Jps
13948 NameNode
14366 ResourceManager
14174 SecondaryNameNode
hadoop@master:/usr/local/hadoop$
```

```
hadoop@slave01:/usr/local/hadoop/etc$ cd ..
hadoop@slave01:/usr/local/hadoop$ ls
bin  include  LICENSE-binary  logs  README.txt
dfs  lib      licenses-binary  NOTICE-binary  sbin
etc  libexec  LICENSE.txt     NOTICE.txt    share
hadoop@slave01:/usr/local/hadoop$ rm -rf logs
hadoop@slave01:/usr/local/hadoop$ ls
bin  include  LICENSE-binary  NOTICE-binary  sbin
dfs  lib      licenses-binary  NOTICE.txt    share
etc  libexec  LICENSE.txt     README.txt     tmp
hadoop@slave01:/usr/local/hadoop$ cd dfs
hadoop@slave01:/usr/local/hadoop/dfs$ ls
data
hadoop@slave01:/usr/local/hadoop/dfs$ rm -rf data/
hadoop@slave01:/usr/local/hadoop/dfs$ ls
hadoop@slave01:/usr/local/hadoop/dfs$ cd ..
hadoop@slave01:/usr/local/hadoop$ jps
5463 DataNode
5550 Jps
hadoop@slave01:/usr/local/hadoop$ jps
5463 DataNode
5719 Jps
5627 NodeManager
hadoop@slave01:/usr/local/hadoop$
```

知乎 @一场有相逢

在master节点对应执行stop-yarn.sh、stop-dfs.sh命令即可，也可直接执行stop-all.sh命令。



2.4 Hadoop集群的部署与使用

- 2.4.1 集群节点类型
- 2.4.2 集群规模
- 2.4.3 集群硬件配置
- 2.4.4 集群网络拓扑
- 2.4.5 集群的建立与安装
- 2.4.6 集群基准测试
- 2.4.7 在云计算环境中使用Hadoop



2.4.1 Hadoop集群中有哪些节点类型

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



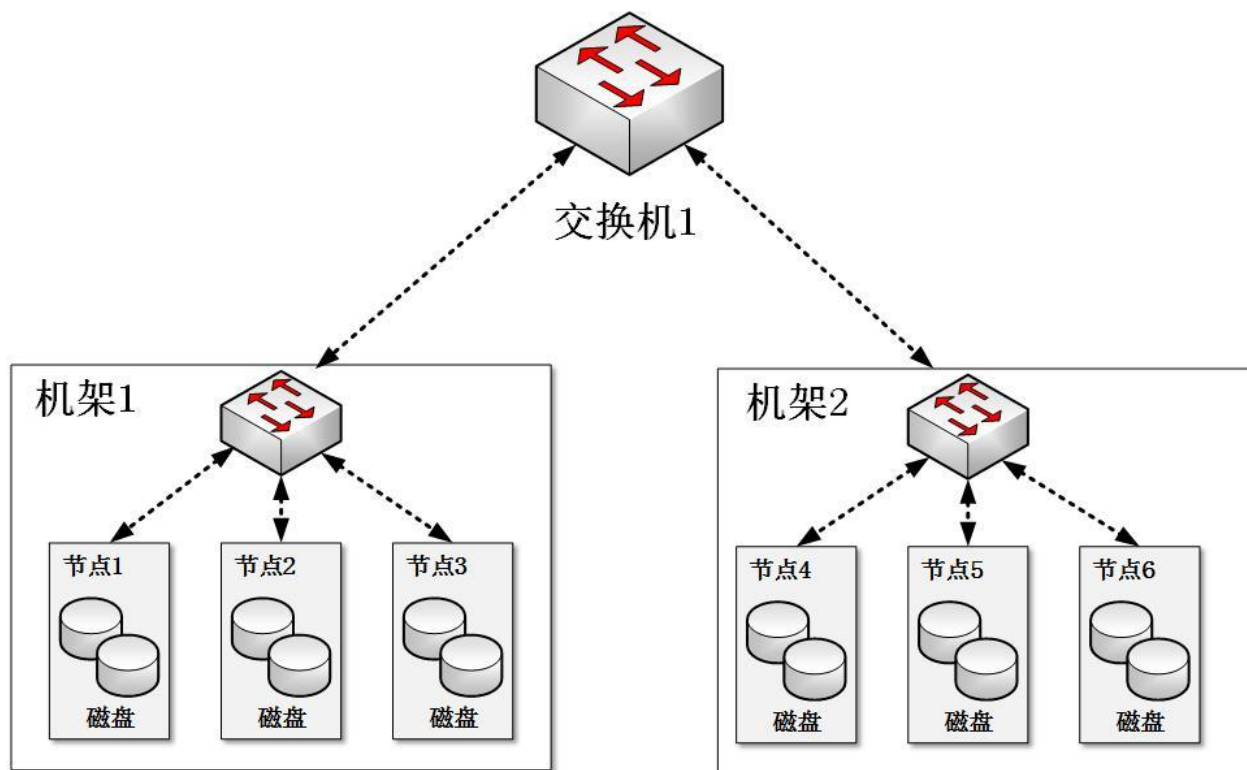
2.4.3 集群规模要多大

- 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 集群的建立与安装

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 在云计算环境中使用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

The background of the slide is a solid blue color. In the upper portion, there are two faint, light-blue silhouettes of groups of people holding hands in a circle. In the lower-left corner, there is a silhouette of a person's head and shoulders. In the lower-right corner, there is a silhouette of a person's arm and hand, with the index finger pointing upwards.

Thank You!