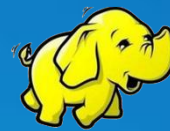


初识Hadoop

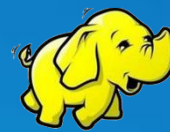


Hadoop离线计算— 杜黎明





- 1.Hadoop发展历史，生态圈，组成及其基本架构了解。
- 2.Linux系统安装。
- 3.Hadoop集群环境搭建。
- 4.MapReduce demo运行，测试搭建的环境。



1

前言

2

数据存储和分析

3

Hadoop生态圈

4

Hadoop分布式文件存储系统HDFS

5

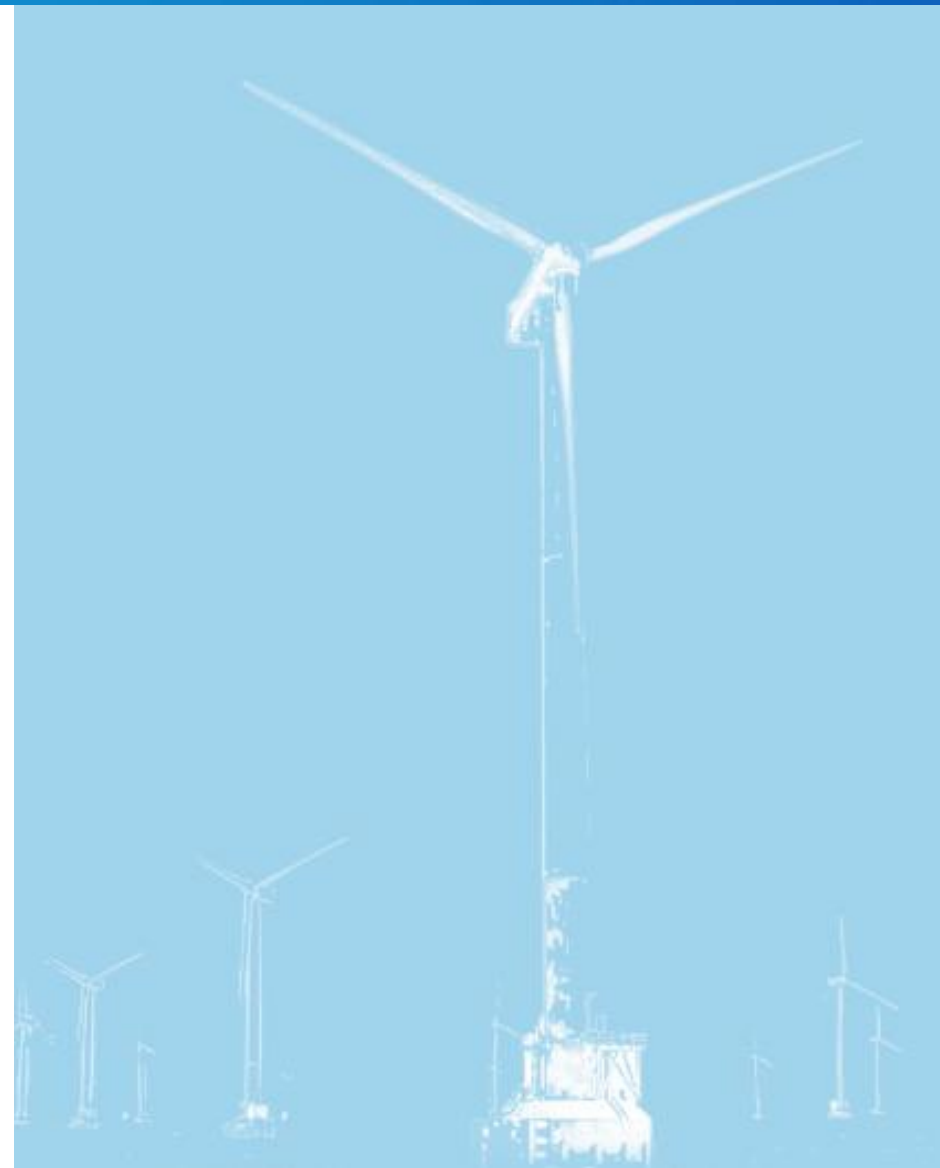
Hadoop资源管理器YARN

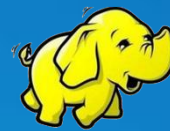
6

Hadoop分布式计算框架MapReduce

7

Hadoop环境搭建

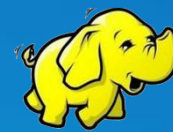




在古时候，人们用牛来拉重物。当一头牛拉不动一根圆木时，人们从来没有考虑过要培育更强壮的牛。同理，我们也不该想方设法的打造超级计算机，而应该千方百计综合利用更多计算机来解决问题。

——格蕾丝·霍博(Grace Hopper)

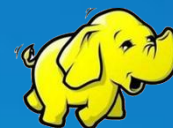




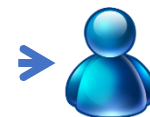
我们生活在这个数据爆炸的时代，很难估算全球电子设备中存储的数据总共有多少。但是据IDC估计2006年“数字全球”项目(digital universe)的数据总量为0.18 ZB，并且预测到2011年这个数字将达到1.8 ZB。

- ◆ 纽约证交所每天产生的交易数据多达1TB。
- ◆ 脸谱网存储的照片约100亿张，存储容量为1PB。
- ◆ 互联网档案馆的存储数据约为2PB，并以每月20TB的速度持续增长。
- ◆ 瑞士日内瓦附近的大型强子对撞机每年产生约15PB的数据。

我们有一个好消息和一个坏消息。好消息是有海量数据！坏消息是我们正在为存储和分析这些数据而奋斗不息。



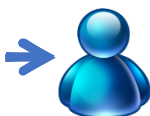
1990年，硬盘1370M，传输速度4.4 MB/s。5分钟读取完成。

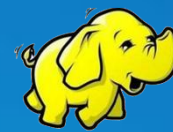


2010年，硬盘1TB，传输速度100MB/s。2.5小时读取完成。



试想，如果有100块硬盘，每块硬盘存储1%的数据，并行读取。
那么我们需要多少时间读完所有数据呢？ **2分钟**





1. 硬件故障问题如何解决？

◆ RAID

◆ HDFS

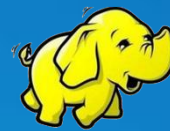
⑩ HDFS建立的思想是：一次写入、多次读取模式是最高效的。

HDFS是一个高度容错性的系统，适合部署在廉价的机器上。

HDFS能提供高吞吐量的数据访问，非常适合大规模数据集上的应用

2. 数据打散分布到多台机器的多块硬盘上，如何保证数据的正确性和一致性？

◆ Mapreduce



2002年10月，Doug Cutting和Mike Cafarella创建了开源网页爬虫项目Nutch。

2003年10月，Google发表Google File System论文。

2004年7月，Doug Cutting和Mike Cafarella在Nutch中实现了类似GFS的功能，即后来HDFS的前身。

2004年10月，Google发表了MapReduce论文。

2005年2月，Mike Cafarella在Nutch中实现了MapReduce的最初版本。

2005年12月，开源搜索项目Nutch移植到新框架，使用MapReduce和NDIFS(Nutch Distributed File System)来运行，在20个节点稳定运行。

2006年1月，Doug Cutting加入雅虎，Yahoo!提供一个专门的团队和资源将Hadoop发展成一个可在网络上运行的系统。

2006年2月，Apache Hadoop项目正式启动以支持MapReduce和HDFS的独立发展。

2006年2月，Yahoo!的网格计算团队采用Hadoop。

2006年3月，Yahoo!建设了第一个Hadoop集群用于开发。

2006年4月，第一个Apache Hadoop发布。

2006年4月，在188个节点上（每个节点10GB）运行排序测试集需要47.9个小时。

2006年5月，在500个节点上运行排序测试集需要42个小时（硬件配置比4月的更好）。

2006年11月，Google发表了Bigtable论文，这最终激发了HBase的创建。

2006年12月，排序测试集在20个节点上运行1.8个小时，100个节点上运行3.3小时，500个节点上运行5.2小时，900个节点上运行7.8个小时。

2007年10月，第一个Hadoop用户组会议召开，社区贡献开始急剧上升。

2008年1月，Hadoop成为Apache顶级项目。

2008年2月，Yahoo!运行了世界上最大的Hadoop应用，宣布其搜索引擎产品部署在一个拥有1万个内核的Hadoop集群上。

2008年4月，在900个节点上运行1TB排序测试集仅需209秒，成为世界最快。

2008年6月，Hadoop的第一个SQL框架——Hive成为了Hadoop的子项目。

2008年7月，Hadoop打破1TB数据排序基准测试记录。Yahoo!的一个Hadoop集群用209秒完成1TB数据的排序，比上一年的纪录保持者保持的297秒快了将近90秒。

2008年8月，第一个Hadoop商业化公司Cloudera成立。

2008年11月，Apache Pig的最初版本发布。

2009年3月，Cloudera推出世界上首个Hadoop发行版——CDH（Cloudera's Distribution including Apache Hadoop）平台，完全由开放源码软件组成。

2009年5月，Yahoo的团队使用Hadoop对1 TB的数据进行排序只花了62秒时间。

2009年6月，Cloudera的工程师Tom White编写的《Hadoop权威指南》初版出版，后被誉为Hadoop圣经。

2009年8月，Hadoop创始人Doug Cutting加入Cloudera担任首席架构师。

2010年9月，Hive(Facebook) 脱离Hadoop，成为Apache顶级项目。

2010年9月，Pig脱离Hadoop，成为Apache顶级项目。

2010年-2011年，扩大的Hadoop社区忙于建立大量的新组件（Crunch，Sqoop，Flume，Oozie等）来扩展Hadoop的使用场景和可用性。

2011年1月，ZooKeeper 脱离Hadoop，成为Apache顶级项目。

2011年7月，Yahoo!和硅谷风险投资公司 Benchmark Capital创建了Hortonworks 公司，旨在让Hadoop更加可靠，并让企业用户更容易安装、管理和使用Hadoop。

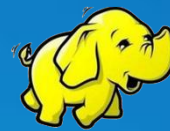
2012年3月，企业必须的重要功能HDFS NameNode HA被加入Hadoop主版本。

2012年8月，另外一个重要的企业适用功能YARN成为Hadoop子项目。

2012年10月，第一个Hadoop原生MPP查询引擎Impala加入到了Hadoop生态圈。

2014年2月，Spark逐渐代替MapReduce成为Hadoop的缺省执行引擎，并成为Apache基金会顶级项目。

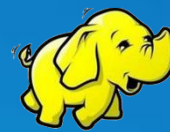




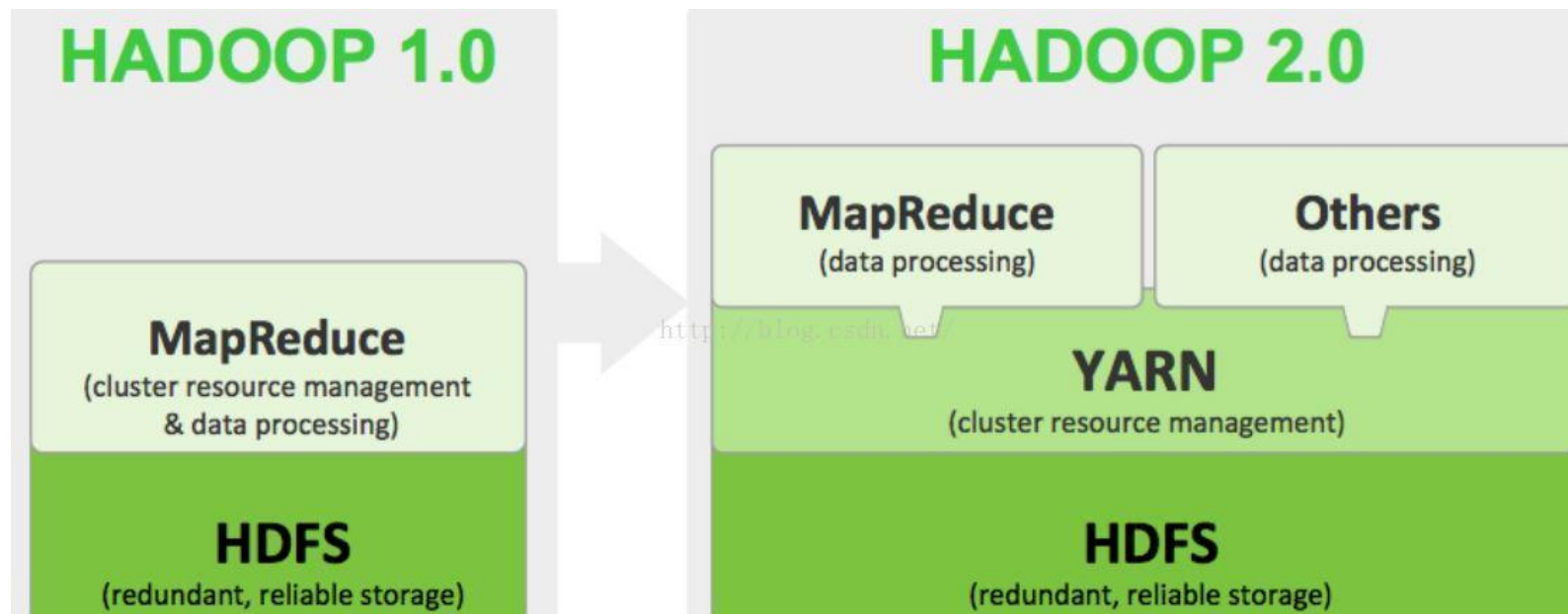
- ◆ 源代码开源。
- ◆ 社区活跃、参与者众多。
- ◆ 涉及分布式存储和计算的方方面面。
- ◆ 已得到了企业界的认证。

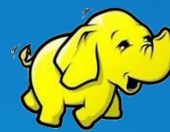


Hadoop1.0与Hadoop2.0



Hadoop离线计算

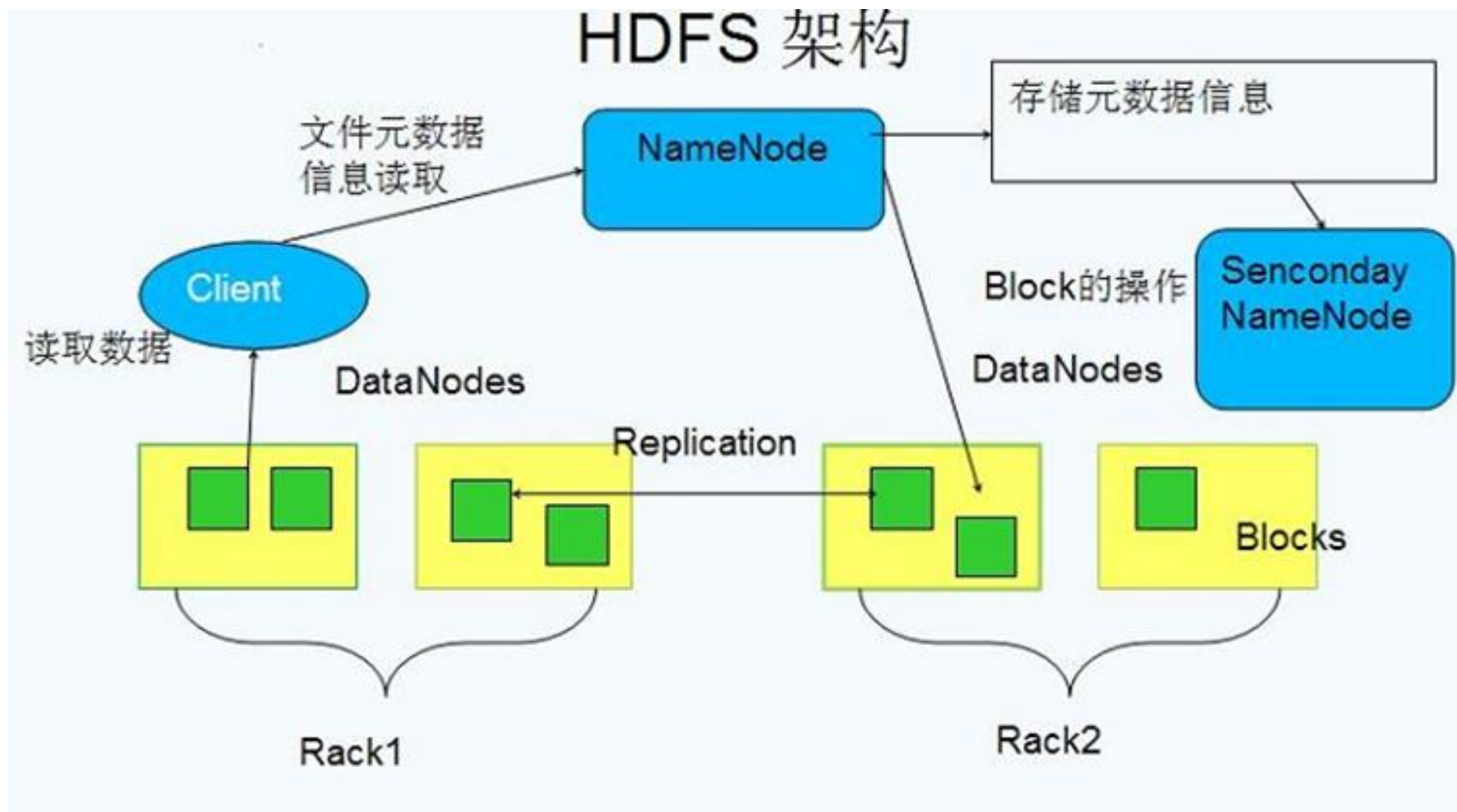
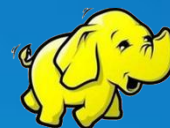


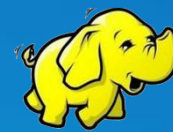


一句话(官方): 分布式存储系统HDFS (Hadoop Distributed File System) 。
其实就是一个文件系统, 类似于linux的文件系统。有目录, 目录下可以存储文件。但它又是一个分布式的文件系统。

基本原理:

- 将文件切分成等大的数据块, 分别存储到多台机器上。
- 每个数据块存在多个备份。
- 将数据切分、容错、负载均衡等功能透明化。
- 可将HDFS看成是一个巨大、具有容错性的磁盘。



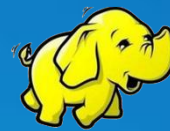


优点

- ◆ 处理超大文件。
- ◆ 流式的访问数据。
- ◆ 运行于廉价的商用机器集群上。

缺点

- ◆ 不适合存储大量小文件。
- ◆ 不适合低延迟数据访问。
- ◆ 不支持多用户写入及任意修改文件。



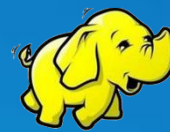
用户命令:

hadoop fs -ls /
hadoop fs -mkdir /dataAnalysis/duliming/test/t1
hadoop fs -put
hadoop fs -get
hadoop fs -copyFromLocal
hadoop fs -copyToLocal
hadoop fs -chmod
hadoop fs -rmr
hadoop fs -cat
hadoop fs -cp

管理命令:

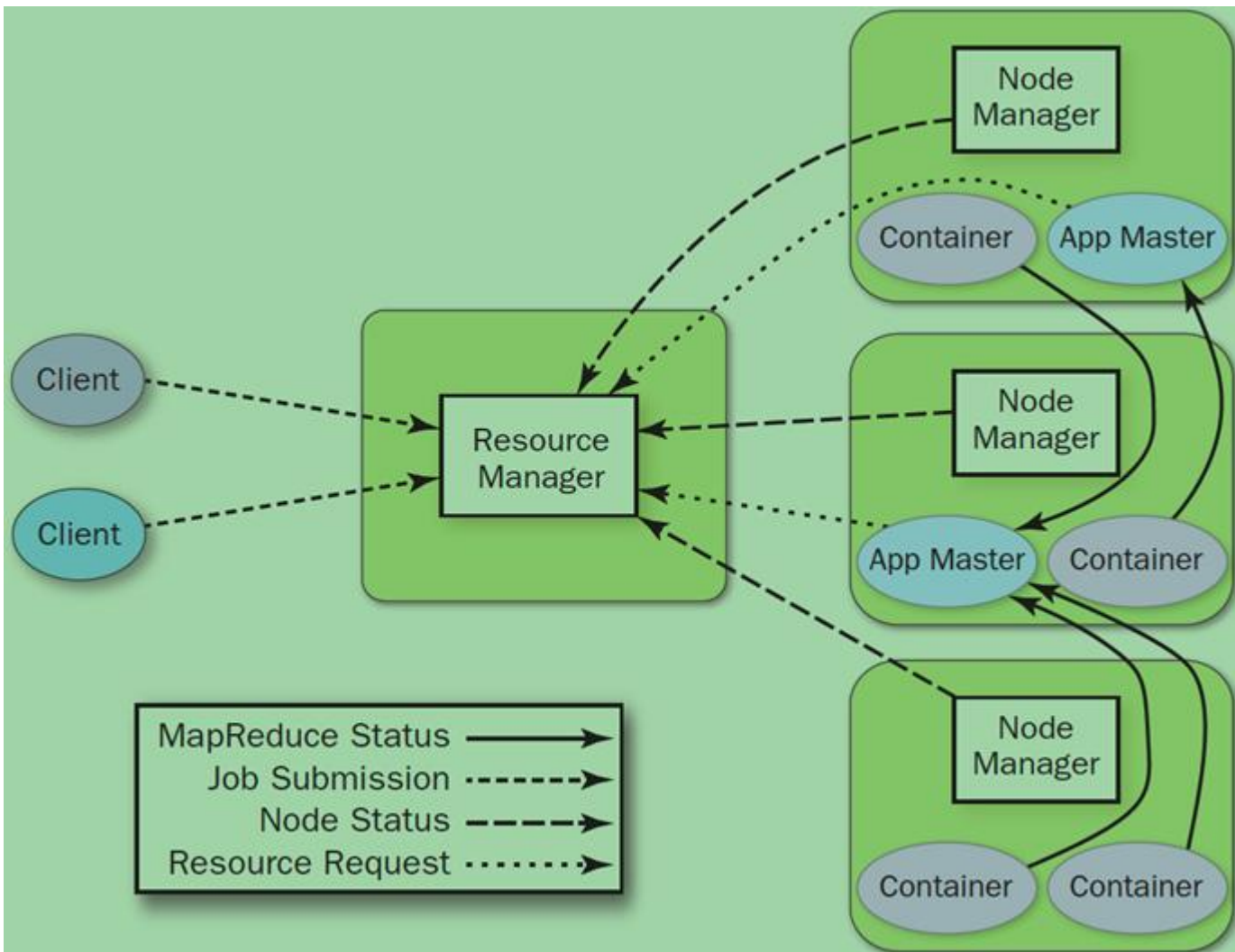
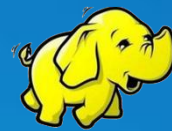
hadoop dfsadmin -report
hadoop dfsadmin -safemode enter | leave | get | wait
hadoop fsck
hadoop balancer
hadoop namenode [-format] | [-upgrade] | [-rollback]

参考: http://hadoop.apache.org/docs/r1.0.4/cn/hdfs_shell.html

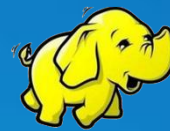


- ◆ Hadoop2.0提出的资源管理层。
- ◆ 用于集群资源管理和应用调度。
- ◆ 使得多种计算框架可以运行在一个集群中。
- ◆ mapreduce仅仅是Yarn中的一种应用模式。其他的应用模式也可以像mapreduce一样，使用既定的协议从Yarn中申请CPU、内存或其他资源，并在集群中运行。

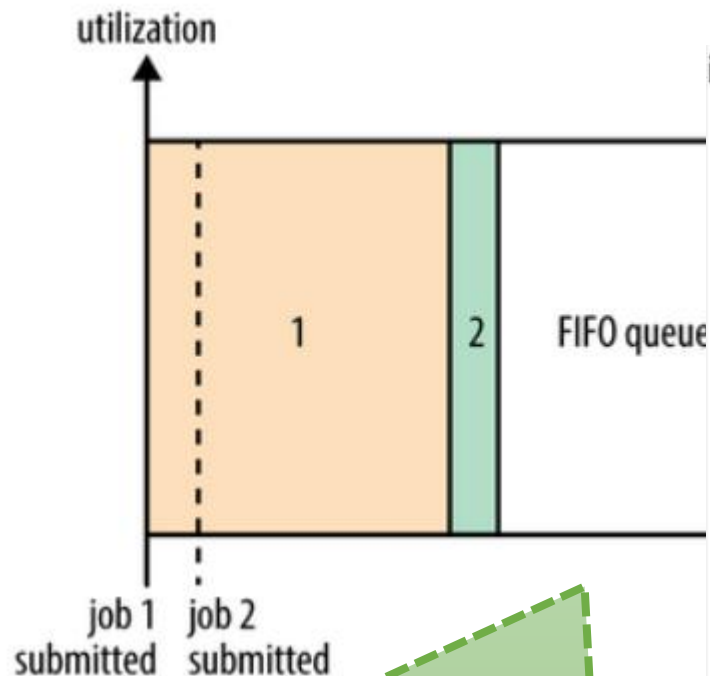




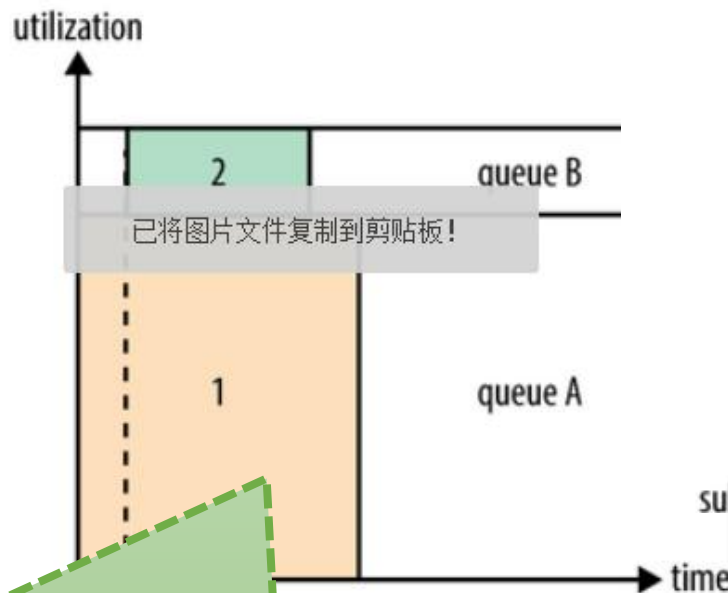
- ◆ **资源管理器(Resource Manager, RM)**:每个集群中都有一个RM的守护进程，专门负责集群中可用资源的分配和管理。
- ◆ **节点管理器(Node Manager, NM)**:每个节点都有一个NM的守护进程，负责节点的本地资源管理。在RM中，NM代表本地节点。
- ◆ **Application Mater(AM)**:每个应用都有一个AM的守护进程，它封装了应用程序所有的逻辑结构和依赖库信息。**AM**负责与RM进行资源协商，并协同NM工作以完成应用的功能。
- ◆ **容器(Container)**:这是分配给具体应用的资源的抽象形式。**AM**是一个启动和管理应用整个生命周期的特殊容器。
- ◆ **客户端(Client)**:这是集群中能向RM提交应用的实例，并且提供了执行应用所需的AM类型。



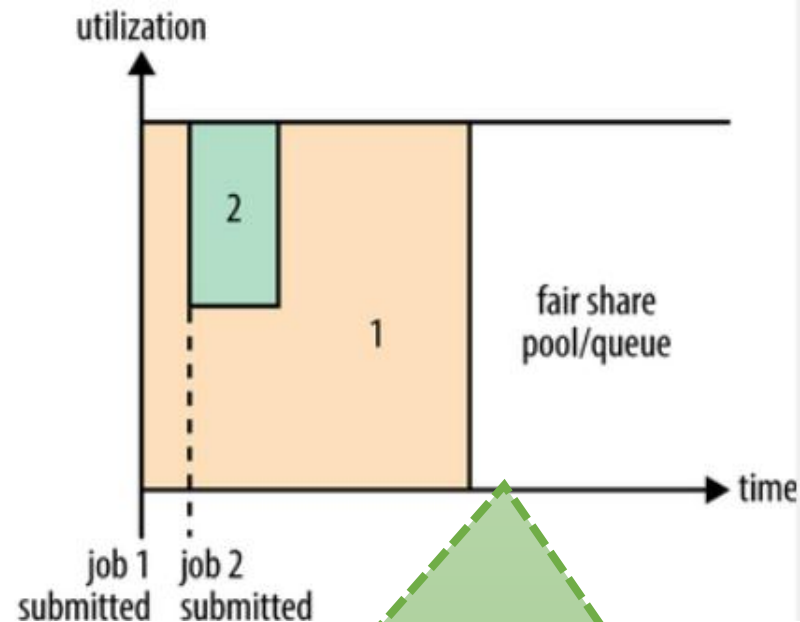
i. FIFO Scheduler



ii. Capacity Scheduler



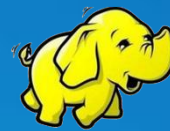
iii. Fair Scheduler



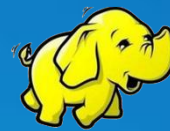
FIFO Scheduler把应用按先出队列，在进行资源分配资源，待最头上的应用大任务的执行时间会

而对于**Capacity**调度器，为小任务专门设置一个队列，大任务的执行时间会

在**Fair**调度器中，我们不需要预先占用一定的系统资源，**Fair**调度器会为所有运行的job动态的调整系统资源。如下图所示，当第一个大job提交时，只有这一个job在运行，此时它获得了所有集群资源；当第二个小任务提交后，**Fair**调度器会分配一半资源给这个小任务，让这两个任务公平的共享集群资源。



- ◆ `yarn application -list`
- ◆ `yarn application -kill appid`
- ◆ `yarn application -status appid`
- ◆ `yarn jar [jar file path] [main class] [arguments]`
- ◆ `yarn node -list`
- ◆ `yarn node -status`
- ◆ 参考: <http://hadoop.apache.org/docs/r2.7.3/hadoop-yarn/hadoop-yarn-site/YarnCommands.html>

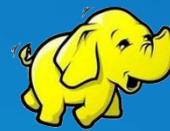


源自于Google的Mapreduce论文

- ◆ 发表于2014年2月。
- ◆ Hadoop Mpareduce是Google Mapreduce的克隆。

Mapreduce的特点

- ◆ 良好的扩展性。
- ◆ 高容错性。
- ◆ 适合PB级及海量数据的离线处理。



为什么需要MapReduce?

并行计算技术和并行程序设计的复杂性

依赖于不同类型的计算问题、数据特征、计算要求、和系统构架，并行计算技术较为复杂，程序设计需要考虑数据划分，计算任务和算法划分，数据访问和通信同步控制，软件开发难度大，难以找到统一和易于使用的计算框架和编程模型与工具。

海量数据处理需要有效的并行处理技术

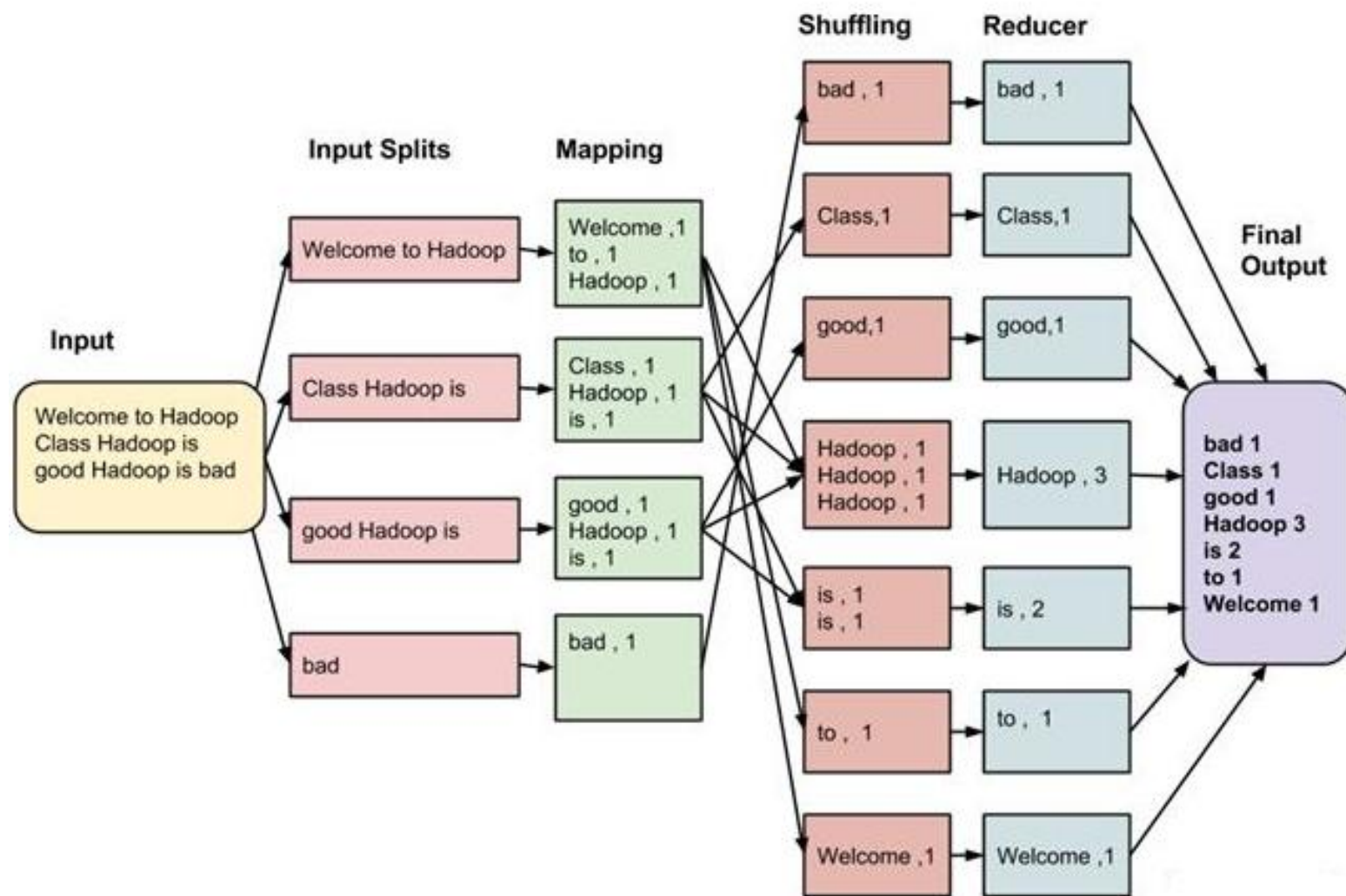
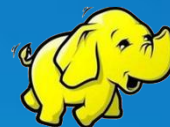
海量数据处理时，依靠MPI等并行处理技术难以凑效

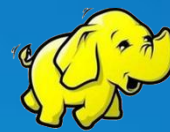
MapReduce是面向海量数据处理比较成功的技术

MapReduce是目前业界和学界公认的最为有效和最易于使用的海量数据并行处理技术。

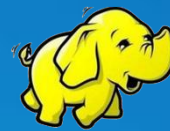
Google, Yahoo, IBM, Amazon, 百度等国内外公司普遍使用

Google: 超过7千个程序基于MapReduce开发!

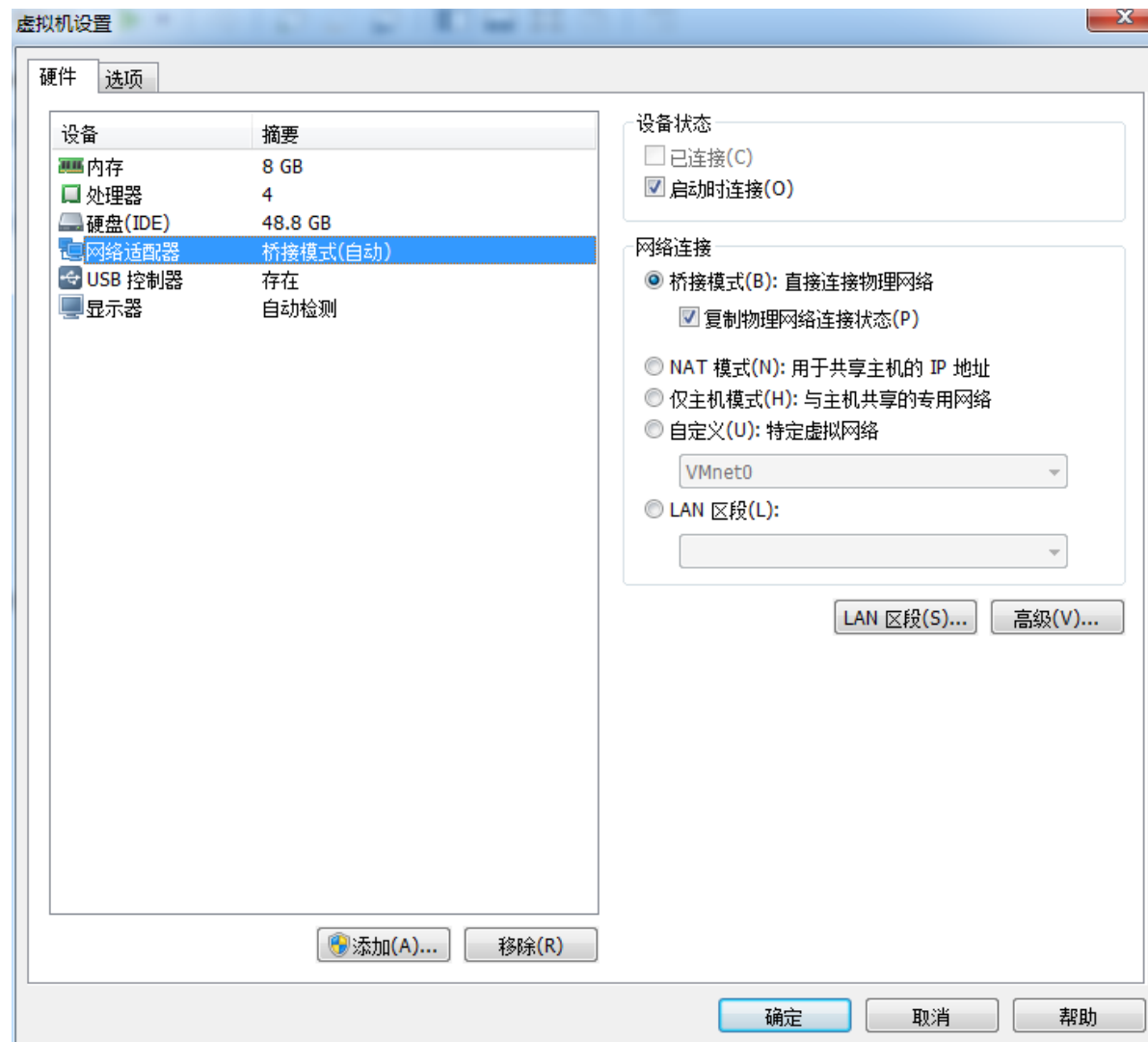


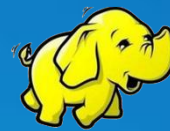


- 建立虚拟机。
- 设置IP地址。
- 安装JDK。
- 编辑Host文件。
- 关闭防火墙。
- 部署免密SSH。
- 下载Hadoop2.x并解压。
- 修改配置文件。
- 分发hadoop到各个节点。
- 启动集群。



- 注意网络连接选择为桥接模式。
便于SSH客户端连接，而无需在虚拟机控制台上操作。





vi /etc/sysconfig/network-scripts/ifcfg-eth0 [编辑

网卡的配置文件]

BOOTPROTO=none

ONBOOT=yes

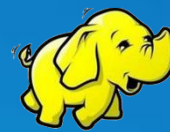
IPADDR=192.168.4.10

NETMASK=255.255.255.0

GATEWAY=192.168.4.1

DNS1=192.168.88.1

```
DEVICE=eth0
TYPE=Ethernet
UUID=fc613194-205e-41aa-a45e-247a4c8cea60
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
HWADDR=C4:54:44:12:94:48
IPADDR=192.168.88.101
PREFIX=24
GATEWAY=192.168.88.1
DNS1=192.168.88.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
LAST_CONNECT=1487280243
```



下载JDK1.7

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jdk-7u80-oth-JPR>

```
mkdir /usr/local/java
```

通过tar -vxf命令将jdk文件解压到java文件夹。

配置环境变量

在"/etc/profile"文件的尾部添加以下内容：

```
# set java environment
```

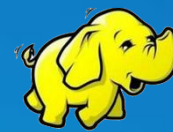
```
export JAVA_HOME=/usr/local/java/jdk1.7.0_75
```

```
export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

```
export PATH=$PATH:$JAVA_HOME/bin
```

输入java -version 判断是否安装成功

每个节点都需要安装JDK



vi /etc/sysconfig/network

NETWORKING=yes #启动网络

NETWORKING_IPV6=no

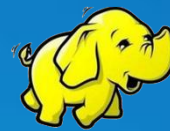
HOSTNAME=master #主机名

检测

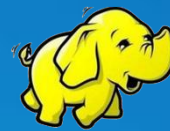
hostname

```
[root@idh101 tools]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.88.101 idh101
192.168.88.102 idh102
192.168.88.103 idh103
192.168.88.104 idh104
192.168.88.105 idh105

[root@idh101 tools]# hostname
idh101
[root@idh101 tools]#
```

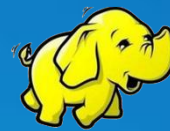


- 输入命令，`ssh-keygen -t rsa`，生成key，都不输入密码，一直回车，`/root`就会生成`.ssh`文件夹，每台服务器都要设置，
- 合并公钥到`authorized_keys`文件，在Master服务器，进入`/root/.ssh`目录，通过SSH命令合并，
`cat id_rsa.pub >> authorized_keys`
`ssh root@192.168.88.230 cat ~/.ssh/id_rsa.pub >> authorized_keys`
`ssh root@192.168.88.231 cat ~/.ssh/id_rsa.pub >> authorized_keys`
- 把Master服务器的`authorized_keys`、`known_hosts`复制到Slave服务器的`/root/.ssh`目录
- 完成，`ssh root@192.168.88.230`、`ssh root@192.168.88.231`就不需要输入密码了

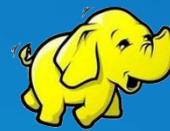


下载并解压Hadoop包。

<http://hadoop.apache.org/releases.html#25+August%2C+2016%3A+Release+2.7.3+available>



- 1.Hadoop由来，要解决的问题？
- 2.分布式文件存储系统HDFS。
- 3.资源管理器YARN。
- 4.分布式计算框架MapReduce。
- 5.Hadoop分布式环境搭建，及其简单使用。



1. 《Hadoop权威指南》第3章。
2. 《Hadoop实战-陆嘉恒》第9章。
3. 推荐董西成的Hadoop课程，观看第2讲。

THANKS

致敬每一个看似微渺的小梦想！

致敬每一位正在路上的追梦人！

