

尚硅谷大数据技术之 Hadoop（入门）

（作者：大海哥）

官网：www.atguigu.com

版本：V1.3

一 大数据概论

1.1 大数据概念



尚硅谷
www.atguigu.com

一、大数据概念

大数据（big data）：指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据集合，是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产。

主要解决，海量数据的存储和海量数据的分析计算问题。

按顺序给出数据存储单位：bit、Byte、KB、MB、GB、TB、PB、EB、ZB、YB、BB、NB、DB。

1Byte = 8bit 1K = 1024Byte 1MB = 1024K
1G = 1024M 1T = 1024G 1P = 1024T

才存 100T



1.2 大数据的特点

尚硅谷
www.atguigu.com

二、大数据特点

1、Volume（大量）：

截至目前，人类生产的所有印刷材料的数据量是**200PB**，而历史上全人类总共说过的话的数据量大约是**5EB**。当前，典型个人计算机硬盘的容量为**TB**量级，而一些大企业的数据量已经接近**EB**量级。



尚硅谷
www.atguigu.com

二、大数据特点

2、Velocity（高速）：

这是大数据区别于传统数据挖掘的最显著特征。根据IDC的“数字宇宙”的报告，预计到2020年，全球数据使用量将达到**35.2ZB**。在如此海量的数据面前，处理数据的效率就是企业的生命。

天猫双十一：2017年3分01秒，天猫交易额超过100亿



二、大数据特点

3、Variety (多样):

这种类型的多样性也让数据被分为结构化数据和非结构化数据。相对于以往便于存储的以数据库/文本为主的结构化数据，非结构化数据越来越多，包括网络日志、音频、视频、图片、地理位置信息等，这些多类型的数据对数据的处理能力提出了更高要求。



id	用户	日期	购买商品	购买数量
1001	canglaoshi	20170710-9:10:10	面膜	2
1002	xiaozelaoshi	20170710-9:11:20	化妆品	3
1003	boduolaoshi	20170710-9:22:50	内衣	4
1004	sslaoshi	20170710-10:12:20	海狗人参丸	100



二、大数据特点

4、Value (低价值密度):

价值密度的高低与数据总量的大小成反比。比如，在一天监控视频中，我们只关心宋宋老师晚上在床上健身那一分钟，如何快速对有价值数据“提纯”成为目前大数据背景下待解决的难题。



1.3 大数据能干啥？



三、大数据能干啥

1、O2O：百度大数据+平台通过先进的线上线下打通技术和客流分析能力，助力商家精细化运营，提升销量。



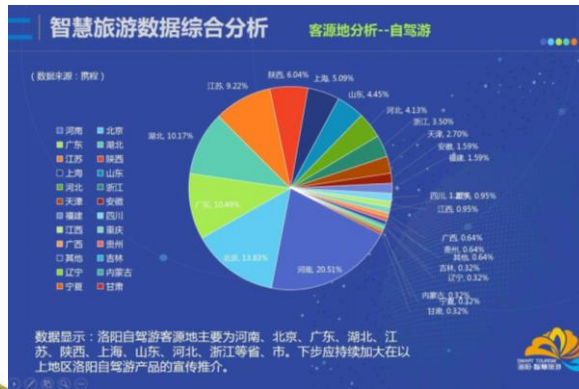
三、大数据能干啥

2、零售：探索用户价值，提供个性化服务解决方案；贯穿网络与实体零售，携手创造极致体验。经典案例，子尿布+啤酒。



三、大数据能干啥

3、旅游：深度结合大数据能力与旅游行业需求，共建旅游产业智慧管理、智慧服务和智慧营销的未来。



三、大数据能干啥

4、商品广告推荐：给用户推荐访问过的商品广告类型

我选了一种药，又推荐了8种，太棒了，么么哒！

商品已成功加入购物车！

【3万人好评 买2送1】罗博士 海狗人参丸100粒 男性保健品含淫羊藿非速效延时持...
数量: 1

[查看商品详情](#) [去购物车结算](#)

购买了该商品的用户还购买了

<p>【3万人好评 京东配送】罗博士 海狗人参丸100粒 秘鲁进口</p> <p>¥98.00 商品</p> <p>加入购物车</p>	<p>罗博士 洋参滋羊鞭软胶囊90粒 男性保健品 非延迟药持久</p> <p>¥108.00 商品</p> <p>加入购物车</p>	<p>罗博士 维生素C咀嚼片vc100片</p> <p>¥32.00 商品</p> <p>加入购物车</p>	<p>罗博士 深海牡蛎片60片 男性保健品</p> <p>¥158.00 商品</p> <p>加入购物车</p>
<p>罗博士 b12维生素100片 复合维生素b12b6b12多种VB</p> <p>¥38.00 商品</p> <p>加入购物车</p>	<p>罗博士 成人益生菌粉 复合益生元膳食纤维 2g*60包</p> <p>¥42.00</p> <p>加入购物车</p>	<p>罗博士 高浓度玛卡60片 玛卡精片 玛卡片</p> <p>¥168.00</p> <p>加入购物车</p>	<p>罗博士 富硒江草软胶囊100粒 男士女士备孕强免疫力</p> <p>¥61.80 商品</p> <p>加入购物车</p>

您可能还需要

1 2 3 4

三、大数据能干啥

5、保险：海量数据挖掘及风险预测，助力保险行业精准营销，提升精细化定价能力。

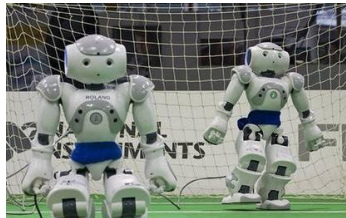
6、金融：多维度体现用户特征，帮助金融机构推荐优质客户，防范欺诈风险。

7、房产：大数据全面助力房地产行业，打造精准投策与营销，选出更合适的地，建造更合适的楼，卖给更合适的人。



三、大数据能干啥

8、人工智能：



1.4 大数据发展前景

U 尚硅谷
www.atguigu.com

四、大数据发展前景

1、党的十八届五中全会提出“实施国家大数据战略”，国务院印发《促进大数据发展行动纲要》，大数据技术和应用处于创新突破期，国内市场需求处于爆发期，我国大数据产业面临重要的发展机遇。

U 尚硅谷
www.atguigu.com

四、大数据发展前景

2、国际数据公司IDC预测，到2020年，企业基于大数据计算分析平台的支出将突破5000亿美元。目前，我国大数据人才只有46万，未来3到5年人才缺口达150万之多。

人才缺口计算
 $150w - 40w = 110w$
 $110W / 5年 = 22w/年$
 $22w / 12月 = 1.83w/月$

自古不变的真理：先入行者吃肉，后入行者喝汤，最后到的买单！

四、大数据发展前景

3、2017年北京大学、中国人民大学、北京邮电大学等25所高校成功申请开设大数据课程。



4、大数据属于高新技术，大牛少，升职竞争小；

四、大数据发展前景

5、在北京大数据开发工程师的平均薪水已经到17800元（数据统计来职友集），而且目前还保持强劲的发展势头。



四、大数据发展前景

6、智联招聘网站上的大数据工程师薪水如下

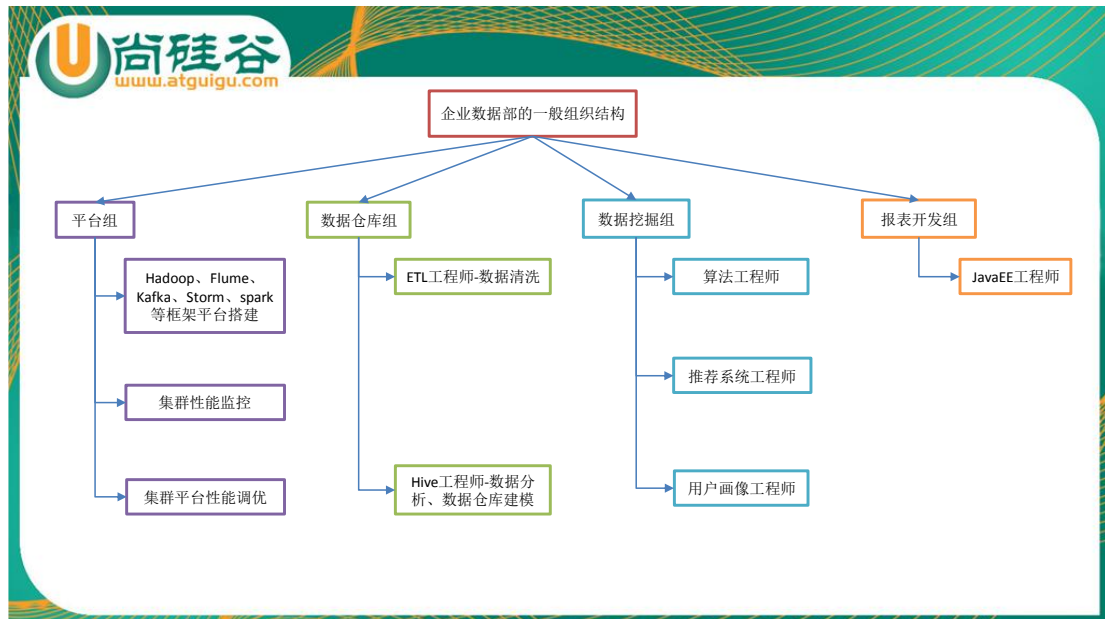
职位名称	经验要求	公司名称	薪资范围	城市	发布日期
软件工程师 (Java/大数据方向)		中金数据系统有限公司	15001-20000	北京	03-11
全国各地大区大数据高级客户经理	97%	广州市冠升网络科技有限公司	20000-24999	北京	03-11
大数据算法工程师	79%	北京三好互动教育科技有限公司	15001-20000	北京	03-11
大数据开发工程师	66%	北京腾信软创信息技术有限公司	15000-20000	北京	03-11
高级ERP开发工程师 (大数据)		万科链家 (北京) 装饰有限公司	15000-30000	北京	03-11
大数据分析师工程师		北京百通无限网络技术有限公司	15001-20000	北京	03-11
高级咨询顾问-财务、大数据、金融、信息化方向		远光软件股份有限公司北京分公司	20000-40000	北京	03-11
大数据系统/算法工程师/数据工程师		北京知趣科技有限公司	15001-20000	北京	03-11
大数据分析师工程师		中金云金融 (北京) 大数据科技股份有限公司	12000-18000	北京-朝阳区	03-11
大数据平台架构师 (java)		中金云金融 (北京) 大数据科技股份有限公司	20000-28000	北京	03-11
大数据工程师-2237		完美世界 (北京) 软件有限公司 BEST	15000-25000	北京-朝阳区	03-11

1.5 企业数据部的业务流程分析



1.6 企业数据部的一般组织结构

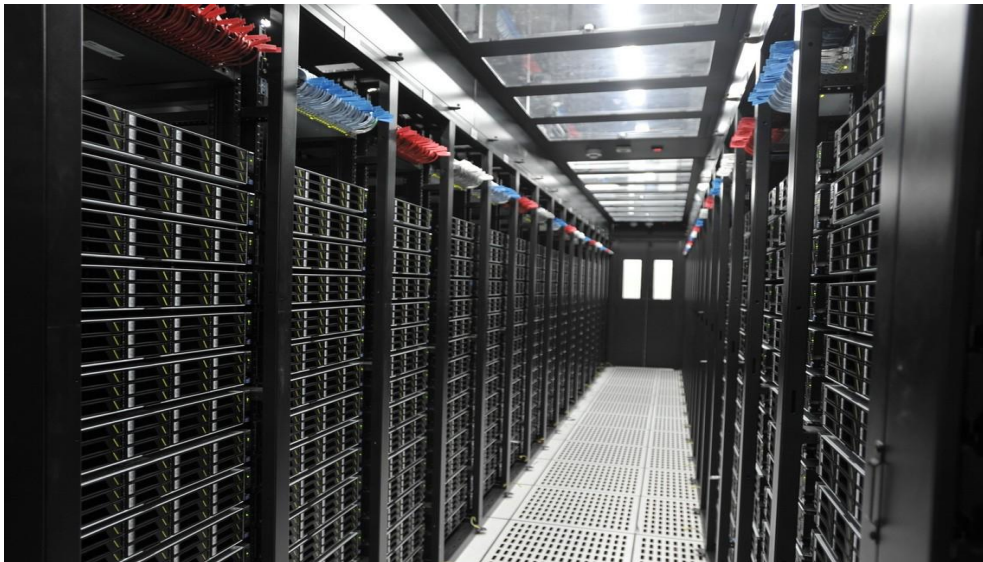
企业数据部的一般组织结构，适用于大中型企业。



二 从 Hadoop 框架讨论大数据生态

2.1 Hadoop 是什么

- 1) Hadoop 是一个由 Apache 基金会所开发的分布式系统基础架构
- 2) 主要解决，海量数据的存储和海量数据的分析计算问题。
- 3) 广义上来说，HADOOP 通常是指一个更广泛的概念——HADOOP 生态圈



2.2 Hadoop 发展历史

- 1) Lucene--Doug Cutting 开创的开源软件，用 java 书写代码，实现与 Google 类似的全文搜索功能，它提供了全文检索引擎的架构，包括完整的查询引擎和索引引擎

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】

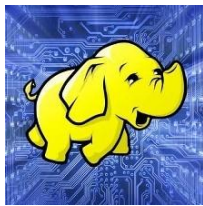
- 2) 2001 年年底成为 apache 基金会有一个子项目
- 3) 对于大量的场景, Lucene 面对与 Google 同样的困难
- 4) 学习和模仿 Google 解决这些问题的办法 : 微型版 Nutch
- 5) 可以说 Google 是 hadoop 的思想之源(Google 在大数据方面的三篇论文)

GFS --->HDFS

Map-Reduce --->MR

BigTable --->Hbase

- 6) 2003-2004 年, Google 公开了部分 GFS 和 Mapreduce 思想的细节, 以此为基础 Doug Cutting 等人用了 2 年业余时间实现了 DFS 和 Mapreduce 机制, 使 Nutch 性能飙升
- 7) 2005 年 Hadoop 作为 Lucene 的子项目 Nutch 的一部分正式引入 Apache 基金会。2006 年 3 月份, Map-Reduce 和 Nutch Distributed File System (NDFS) 分别被纳入称为 Hadoop 的项目中
- 8) 名字来源于 Doug Cutting 儿子的玩具大象



- 9) Hadoop 就此诞生并迅速发展, 标志这云计算时代来临

2.3 Hadoop 三大发行版本

Hadoop 三大发行版本: Apache、Cloudera、Hortonworks。

Apache 版本最原始(最基础)的版本, 对于入门学习最好。

Cloudera 在大型互联网企业中用的较多。

Hortonworks 文档较好。

- 1) Apache Hadoop

官网地址: <http://hadoop.apache.org/releases.html>

下载地址: <https://archive.apache.org/dist/hadoop/common/>

- 2) Cloudera Hadoop

官网地址: <https://www.cloudera.com/downloads/cdh/5-10-0.html>

下载地址: <http://archive-primary.cloudera.com/cdh5/cdh/5/>

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷(中国)官网 www.atguigu.com 下载区】

(1) 2008 年成立的 Cloudera 是最早将 Hadoop 商用的公司，为合作伙伴提供 Hadoop 的商用解决方案，主要是包括支持、咨询服务、培训。

(2) 2009 年 Hadoop 的创始人 Doug Cutting 也加盟 Cloudera 公司。Cloudera 产品主要为 CDH, Cloudera Manager, Cloudera Support

(3) CDH 是 Cloudera 的 Hadoop 发行版，完全开源，比 Apache Hadoop 在兼容性，安全性，稳定性上有所增强。

(4) Cloudera Manager 是集群的软件分发及管理监控平台，可以在几个小时内部署好一个 Hadoop 集群，并对集群的节点及服务进行实时监控。Cloudera Support 即是对 Hadoop 的技术支持。

(5) Cloudera 的标价为每年每个节点 4000 美元。Cloudera 开发并贡献了可实时处理大数据的 Impala 项目。

3) Hortonworks Hadoop

官网地址: <https://hortonworks.com/products/data-center/hdp/>

下载地址: <https://hortonworks.com/downloads/#data-platform>

(1) 2011 年成立的 Hortonworks 是雅虎与硅谷风投公司 Benchmark Capital 合资组建。

(2) 公司成立之初就吸纳了大约 25 名至 30 名专门研究 Hadoop 的雅虎工程师，上述工程师均在 2005 年开始协助雅虎开发 Hadoop，贡献了 Hadoop 80% 的代码。

(3) 雅虎工程副总裁、雅虎 Hadoop 开发团队负责人 Eric Baldeschwieler 出任 Hortonworks 的首席执行官。

(4) Hortonworks 的主打产品是 Hortonworks Data Platform (HDP)，也同样是 100% 开源的产品，HDP 除常见的项目外还包括了 Ambari，一款开源的安装和管理系统。

(5) HCatalog，一个元数据管理系统，HCatalog 现已集成到 Facebook 开源的 Hive 中。Hortonworks 的 Stinger 开创性的极大的优化了 Hive 项目。Hortonworks 为入门提供了一个非常好的，易于使用的沙盒。

(6) Hortonworks 开发了很多增强特性并提交至核心主干，这使得 Apache Hadoop 能够在包括 Window Server 和 Windows Azure 在内的 microsoft Windows 平台上本地运行。定价以集群为基础，每 10 个节点每年为 12500 美元。

2.4 Hadoop 的优势

1) 高可靠性: 因为 Hadoop 假设计算元素和存储会出现故障，因为它维护多个工作数据副本
【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】

本，在出现故障时可以对失败的节点重新分布处理。

2) 高扩展性：在集群间分配任务数据，可方便的扩展数以千计的节点。

3) 高效性：在 MapReduce 的思想下，Hadoop 是并行工作的，以加快任务处理速度。

4) 高容错性：自动保存多份副本数据，并且能够自动将失败的任务重新分配。

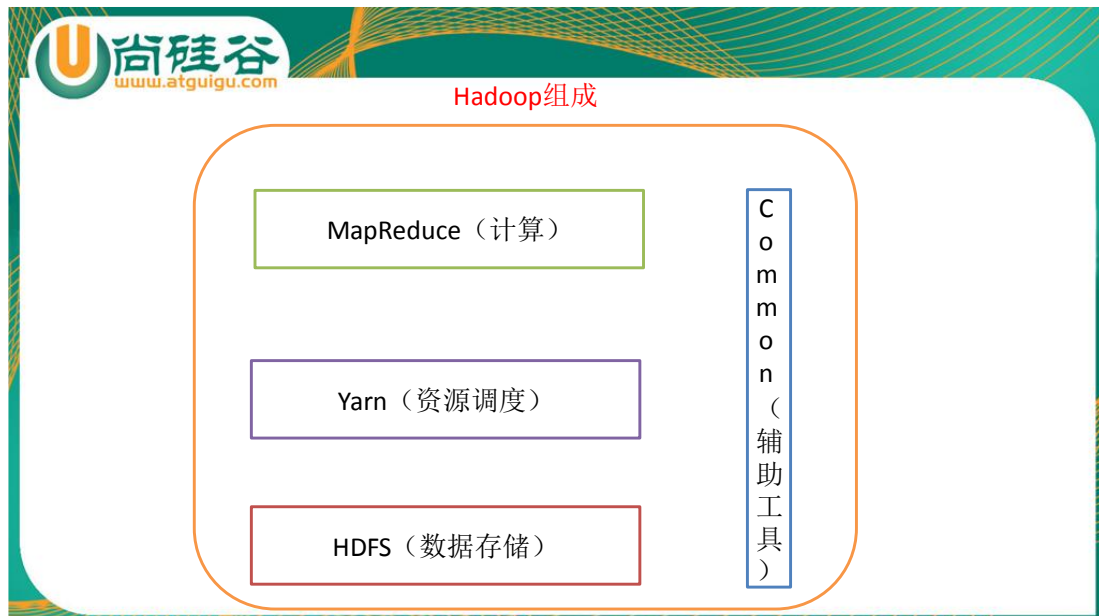
2.5 Hadoop 组成

1) Hadoop HDFS：一个高可靠、高吞吐量的分布式文件系统。

2) Hadoop MapReduce：一个分布式的离线并行计算框架。

3) Hadoop YARN：作业调度与集群资源管理的框架。

4) Hadoop Common：支持其他模块的工具模块（Configuration、RPC、序列化机制、日志操作）。



2.5.1 HDFS 架构概述

尚硅谷
www.atguigu.com

HDFS架构概述

1) NameNode (nn): 存储文件的元数据, 如文件名, 文件目录结构, 文件属性 (生成时间、副本数、文件权限), 以及每个文件的块列表和块所在的DataNode等。



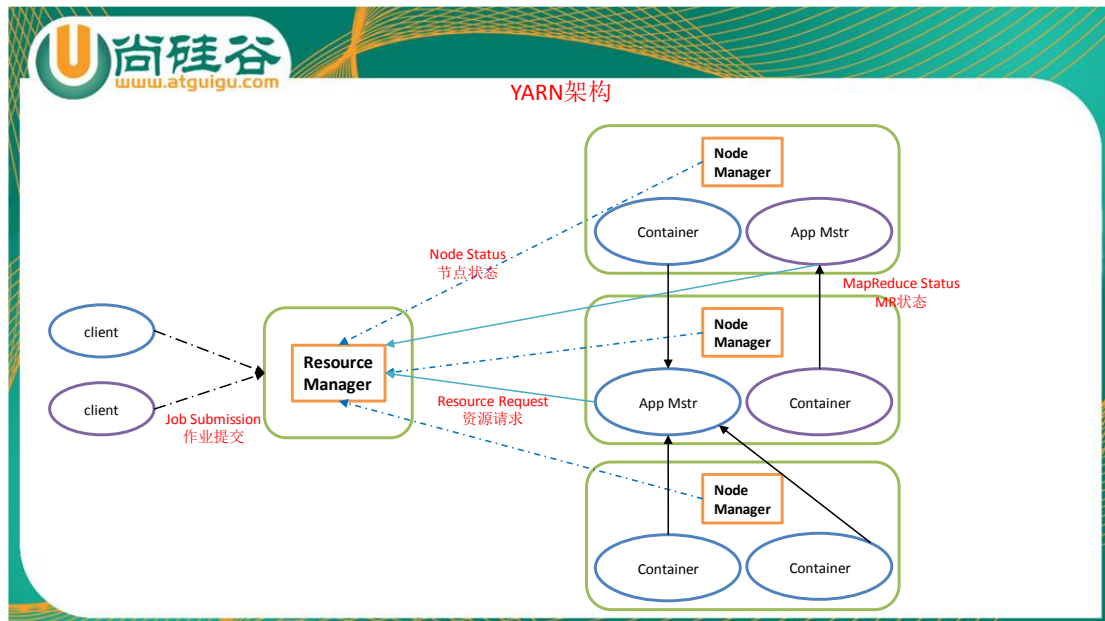
2) DataNode(dn): 在本地文件系统存储文件块数据, 以及块数据的校验和。



3) Secondary NameNode(2nn): 用来监控HDFS状态的辅助后台程序, 每隔一段时间获取HDFS元数据的快照。

2.5.2 YARN 架构概述

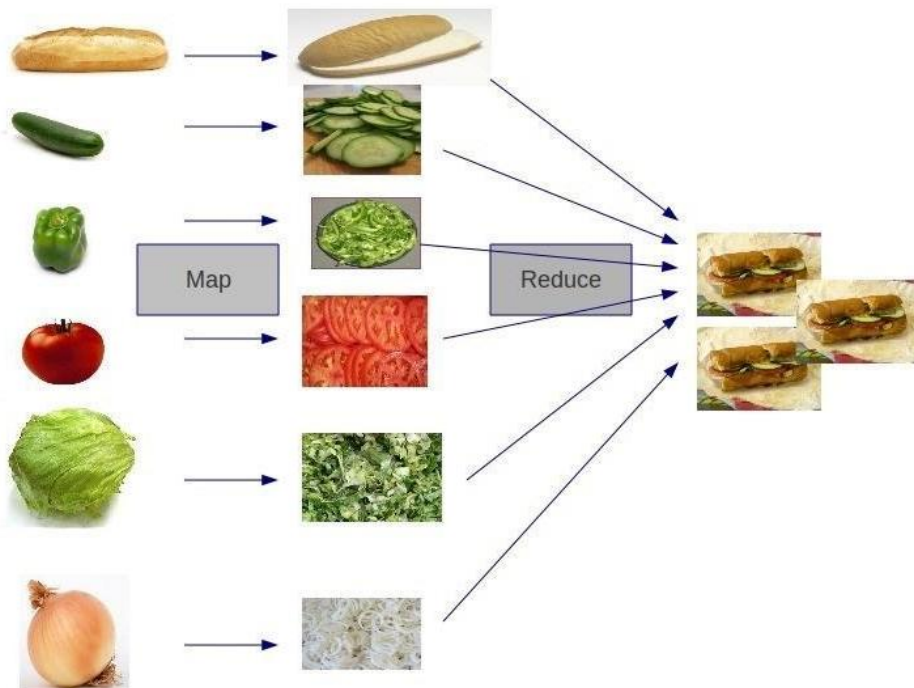
- 1) ResourceManager(rm): 处理客户端请求、启动/监控 ApplicationMaster、监控 NodeManager、资源分配与调度;
- 2) NodeManager(nm): 单个节点上的资源管理、处理来自 ResourceManager 的命令、处理来自 ApplicationMaster 的命令;
- 3) ApplicationMaster: 数据切分、为应用程序申请资源, 并分配给内部任务、任务监控与容错。
- 4) Container: 对任务运行环境的抽象, 封装了 CPU、内存等多维资源以及环境变量、启动命令等任务运行相关的信息。



2.5.3 MapReduce 架构概述

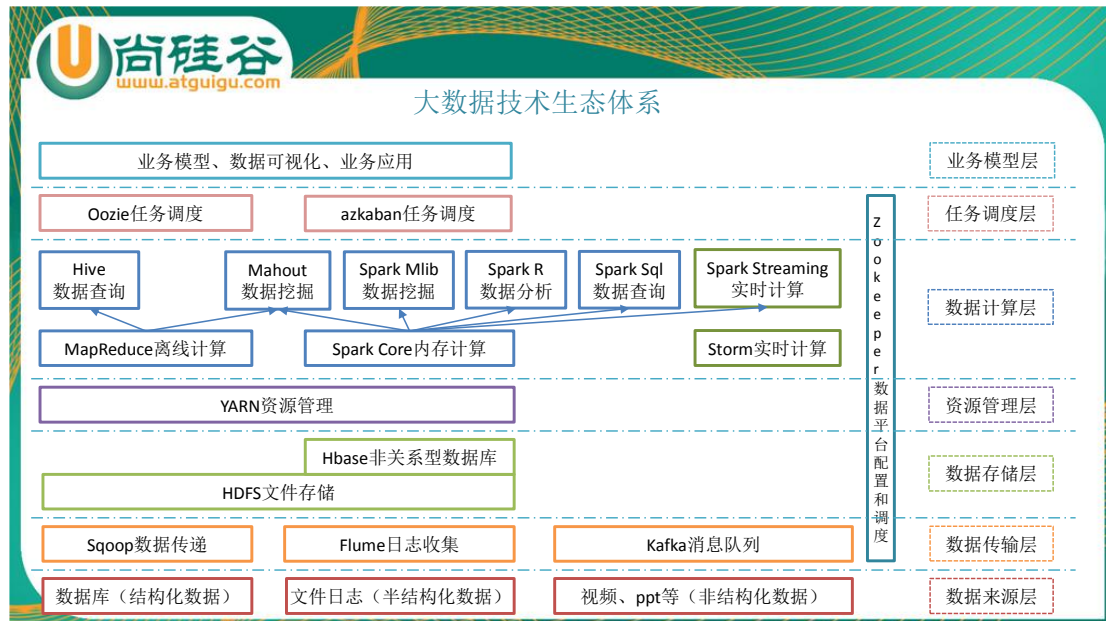
MapReduce 将计算过程分为两个阶段：Map 和 Reduce

- 1) Map 阶段并行处理输入数据
- 2) Reduce 阶段对 Map 结果进行汇总



上图简单的阐明了 map 和 reduce 的两个过程或者作用，虽然不够严谨，但是足以提供一个大概的认知，map 过程是一个蔬菜到制成食物前的准备工作，reduce 将准备好的材料合并进而制作出食物的过程。

2.6 大数据技术生态体系



图中涉及的技术名词解释如下：

1) Sqoop: sqoop 是一款开源的工具，主要用于在 Hadoop(Hive)与传统的数据库(mysql)间进行数据的传递，可以将一个关系型数据库（例如：MySQL, Oracle 等）中的数据导进到 Hadoop 的 HDFS 中，也可以将 HDFS 的数据导进到关系型数据库中。

2) Flume: Flume 是 Cloudera 提供的一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统, Flume 支持在日志系统中定制各类数据发送方, 用于收集数据; 同时, Flume 提供对数据进行简单处理, 并写到各种数据接受方（可定制）的能力。

3) Kafka: Kafka 是一种高吞吐量的分布式发布订阅消息系统，有如下特性：

（1）通过 O(1)的磁盘数据结构提供消息的持久化，这种结构对于即使数以 TB 的消息存储也能够保持长时间的稳定性能。

（2）高吞吐量：即使是非常普通的硬件 Kafka 也可以支持每秒数百万的消息

（3）支持通过 Kafka 服务器和消费机集群来分区消息。

（4）支持 Hadoop 并行数据加载。

4) Storm: Storm 为分布式实时计算提供了一组通用原语，可被用于“流处理”之中，实时处理消息并更新数据库。这是管理队列及工作者集群的另一种方式。Storm 也可被用于“连续计算”（continuous computation），对数据流做连续查询，在计算时就将结果以流的形式输出给用户。

5) Spark: Spark 是当前最流行的开源大数据内存计算框架。可以基于 Hadoop 上存储的大【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】

数据进行计算。

6) Oozie: Oozie 是一个管理 Hadoop 作业 (job) 的工作流程调度管理系统。Oozie 协调作业就是通过时间 (频率) 和有效数据触发当前的 Oozie 工作流程。

7) Hbase: HBase 是一个分布式的、面向列的开源数据库。HBase 不同于一般的关系数据库, 它是一个适合于非结构化数据存储的数据库。

8) Hive: hive 是基于 Hadoop 的一个数据仓库工具, 可以将结构化的数据文件映射为一张数据库表, 并提供简单的 sql 查询功能, 可以将 sql 语句转换为 MapReduce 任务进行运行。其优点是学习成本低, 可以通过类 SQL 语句快速实现简单的 MapReduce 统计, 不必开发专门的 MapReduce 应用, 十分适合数据仓库的统计分析。

10) R 语言: R 是用于统计分析、绘图的语言和操作环境。R 是属于 GNU 系统的一个自由、免费、源代码开放的软件, 它是一个用于统计计算和统计制图的优秀工具。

11) Mahout:

Apache Mahout 是个可扩展的机器学习和数据挖掘库, 当前 Mahout 支持主要的 4 个用例:

推荐挖掘: 搜集用户动作并以此给用户推荐可能喜欢的事物。

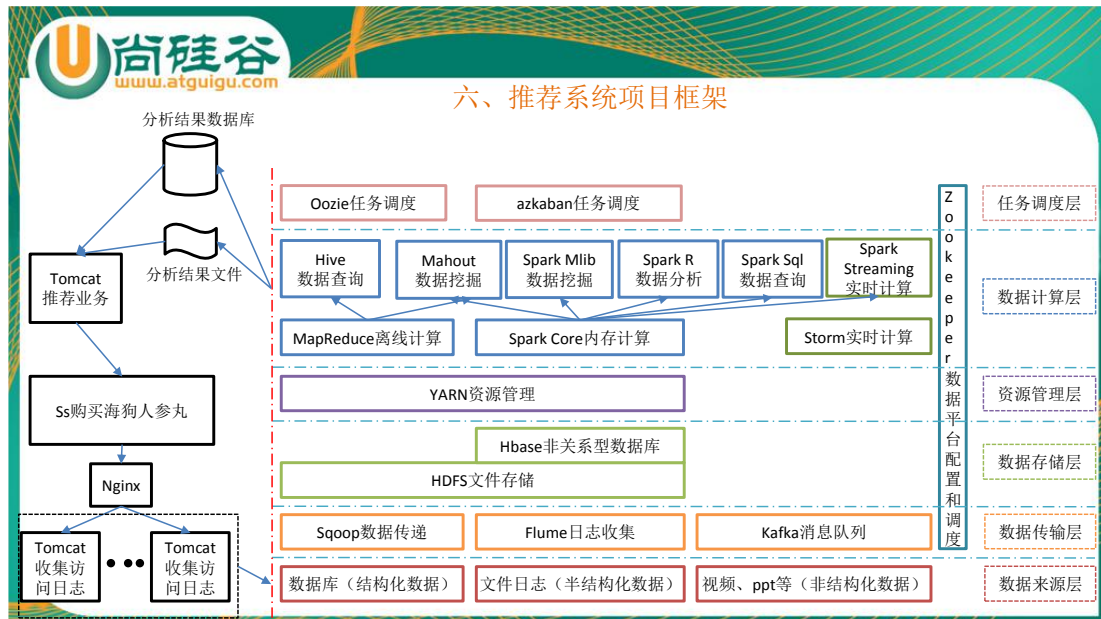
聚集: 收集文件并进行相关文件分组。

分类: 从现有的分类文档中学习, 寻找文档中的相似特征, 并为无标签的文档进行正确的归类。

频繁项集挖掘: 将一组项分组, 并识别哪些个别项会经常一起出现。

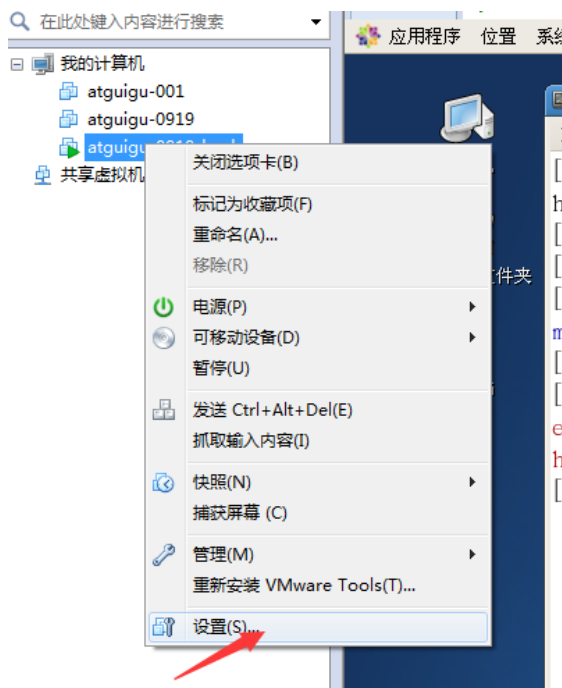
12) ZooKeeper: Zookeeper 是 Google 的 Chubby 一个开源的实现。它是一个针对大型分布式系统的可靠协调系统, 提供的功能包括: 配置维护、名字服务、分布式同步、组服务等。ZooKeeper 的目标就是封装好复杂易出错的关键服务, 将简单易用的接口和性能高效、功能稳定的系统提供给用户。

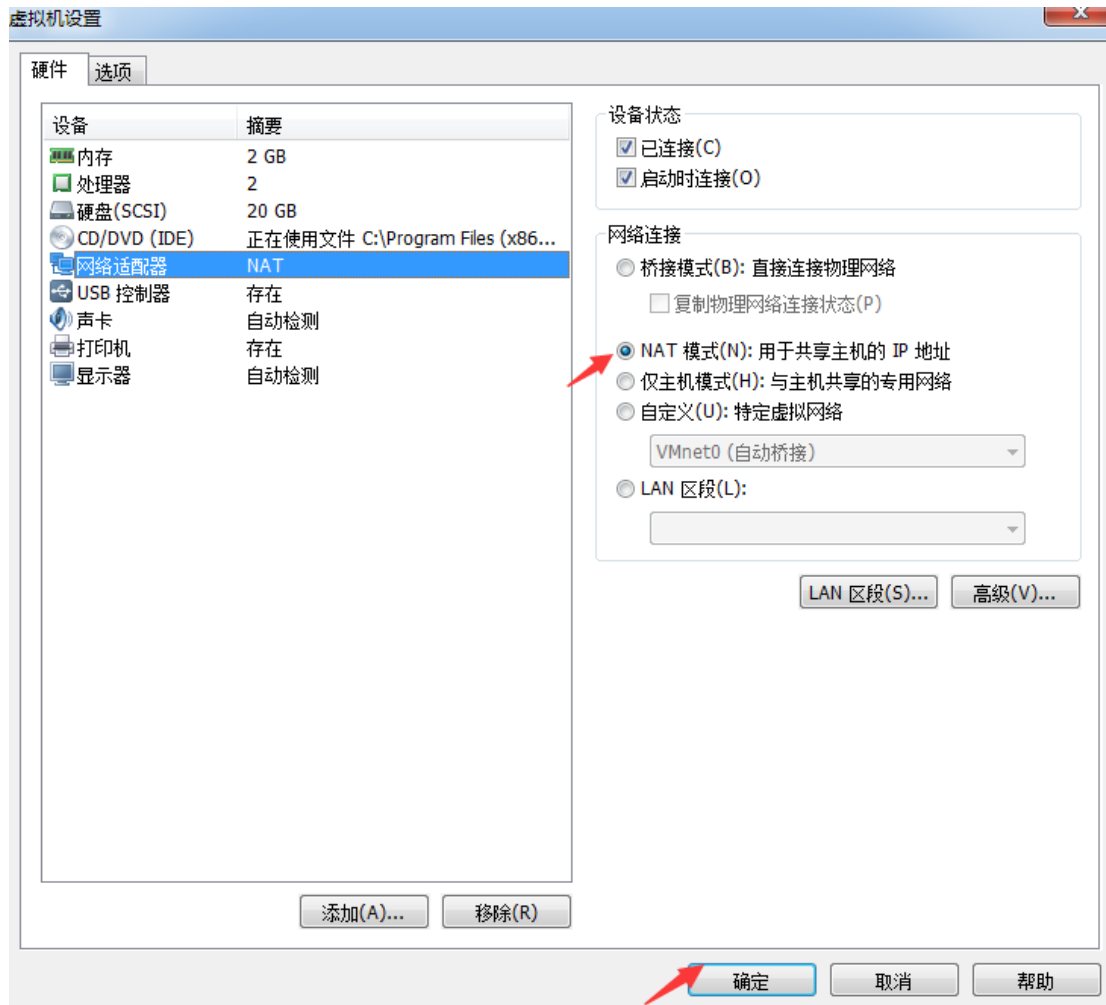
2.7 推荐系统框架图



三 Hadoop 运行环境搭建

3.1 虚拟机网络模式设置为 NAT





最后，重新启动系统。

```
[root@hadoop101 ~]# sync
```

```
[root@hadoop101 ~]# reboot
```

3.2 克隆虚拟机

1) 克隆虚拟机



克隆虚拟机向导



克隆虚拟机向导

克隆源
您想从哪个状态创建克隆?

克隆自

☒ 虚拟机中的当前状态(C)
从当前状态创建链接克隆将创建一个新快照。

☐ 现有快照(仅限关闭的虚拟机)(S):

此虚拟机没有可克隆的快照。

< 上一步(B) **下一步(N) >** 取消

克隆虚拟机向导

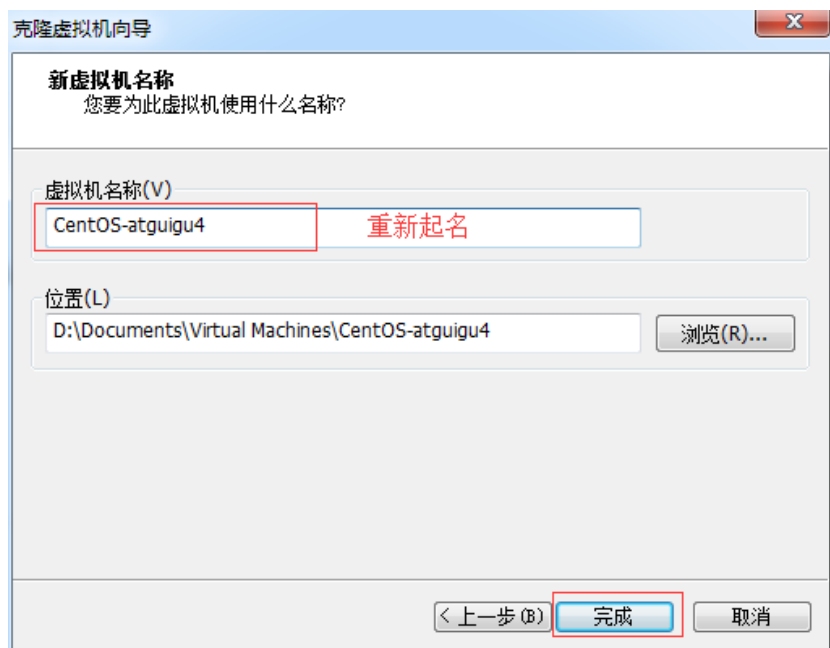
克隆类型
您希望如何克隆此虚拟机?

克隆方法

☐ 创建链接克隆(L)
链接克隆是对原始虚拟机的引用，所需的存储磁盘空间较少。但是，必须能够访问原始虚拟机才能运行。

☒ **创建完整克隆(F)**
完整克隆是原始虚拟机当前状态的完整副本。此副本虚拟机完全独立，但需要较多的存储磁盘空间。

< 上一步(B) **下一步(N) >** 取消





2) 启动虚拟机

3.3 修改为静态 ip

1) 在终端命令窗口中输入

```
[root@hadoop101 /]#vim /etc/udev/rules.d/70-persistent-net.rules
```

进入如下页面，删除 eth0 该行；将 eth1 修改为 eth0，同时复制物理 ip 地址

```
# PCI device 0x1022:0x2000 (vmxnet)                                删除该行
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:0c:08:34", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

# PCI device 0x1022:0x2000 (vmxnet)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:34:c4:3f", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
~
~
~
```

复制该地址

修改为eth0

2) 修改 IP 地址

```
[root@hadoop101 /]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

需要修改的内容有 5 项：

IPADDR=192.168.1.101

GATEWAY=192.168.1.2

ONBOOT=yes

BOOTPROTO=static

DNS1=192.168.1.2

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】

(1) 修改前

```
DEVICE=eth0
TYPE=Ethernet
UUID=109fb0a1-4949-4e2b-80dd-3d72ad223e33
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none          修改为static
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
HWADDR=01:0c:29:0c:08:34  地址是从刚才复制的地址中粘贴过来
IPADDR=192.168.10.102     修改为想要的IP地址
GATEWAY=192.168.10.2
LAST_CONNECT=1490351033
NETMASK=255.255.255.0
USERCTL=no
PEERDNS=yes
```

(2) 修改后

```
DEVICE=eth0
HWADDR=00:0c:29:e0:09:89
TYPE=Ethernet
UUID=50a63ff6-0bb5-42cd-8c9e-1386ace2608f
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static

IPADDR=192.168.1.101
GATEWAY=192.168.1.2
DNS1=8.8.8.8
```

: wq 保存退出

3) 执行

```
[root@hadoop101 /]# service network restart
```

关闭环回接口 :

[确定]

弹出环回接口 :

[确定]

弹出界面 eth0 : 错误 : 激活连接失败 : The connection is not for this device.

[失败]

4) 如果报错, reboot, 重启虚拟机。

```
[root@hadoop101 /]# reboot
```

3.4 修改主机名

1) 修改 linux 的 hosts 文件

(1) 进入 Linux 系统查看本机的主机名。通过 hostname 命令查看。


```
[root@hadoop100 /]# hostname
```

```
hadoop100
```

(2) 如果感觉此主机名不合适, 我们可以进行修改。通过编辑/etc/sysconfig/network 文件。

```
[root@hadoop100 ~]# vi /etc/sysconfig/network
```

修改文件中主机名称

```
NETWORKING=yes
```

```
NETWORKING_IPV6=no
```

```
HOSTNAME= hadoop101
```

注意: 主机名称不要有 “_” 下划线

(3) 打开此文件后, 可以看到主机名。修改此主机名为我们想要修改的主机名 hadoop101。

(4) 保存退出。

(5) 打开/etc/hosts

```
[root@hadoop100 ~]# vim /etc/hosts
```

添加如下内容

```
192.168.1.100 hadoop100
```

```
192.168.1.101 hadoop101
```

```
192.168.1.102 hadoop102
```

```
192.168.1.103 hadoop103
```

```
192.168.1.104 hadoop104
```

```
192.168.1.105 hadoop105
```

```
192.168.1.106 hadoop106
```


```
192.168.1.107 hadoop107
```

```
192.168.1.108 hadoop108
```

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷(中国)官网 www.atguigu.com 下载区】

192.168.1.109 hadoop109

192.168.1.110 hadoop110

(6) 并重启 ，重启后，查看主机名，已经修改成功

2) 修改 window7 的 hosts 文件

(1) 进入 C:\Windows\System32\drivers\etc 路径

(2) 打开 hosts 文件并添加如下内容

192.168.1.100 hadoop100

192.168.1.101 hadoop101

192.168.1.102 hadoop102

192.168.1.103 hadoop103

192.168.1.104 hadoop104

192.168.1.105 hadoop105

192.168.1.106 hadoop106

192.168.1.107 hadoop107

192.168.1.108 hadoop108

192.168.1.109 hadoop109

192.168.1.110 hadoop110

3.5 关闭防火墙

1) 查看防火墙开机启动状态

```
[root@hadoop101 ~]# chkconfig iptables --list
```

2) 关闭防火墙

```
[root@hadoop101 ~]# chkconfig iptables off
```

3.6 在 opt 目录下创建文件

1) 创建 atguigu 用户

在 root 用户里面执行如下操作

```
[root@hadoop101 opt]# adduser atguigu
```

```
[root@hadoop101 opt]# passwd atguigu
```

更改用户 test 的密码。

新的 密码:

无效的密码: 它没有包含足够的不同字符

无效的密码: 是回文

重新输入新的 密码:

passwd: 所有的身份验证令牌已经成功更新。

2) 设置 atguigu 用户具有 root 权限

修改 /etc/sudoers 文件, 找到下面一行, 在 root 下面添加一行, 如下所示:

```
[root@hadoop101 atguigu]# vi /etc/sudoers
```

```
## Allow root to run any commands anywhere
```

```
root    ALL=(ALL)    ALL
```

```
atguigu  ALL=(ALL)    ALL
```

修改完毕, 现在可以用 atguigu 帐号登录, 然后用命令 `su -`, 即可获得 root 权限进行操作。

3) 在 /opt 目录下创建文件夹

(1) 在 root 用户下创建 module、software 文件夹

```
[root@hadoop101 opt]# mkdir module
```

```
[root@hadoop101 opt]# mkdir software
```

(2) 修改 module、software 文件夹的所有者

```
[root@hadoop101 opt]# chown atguigu:atguigu module
```

```
[root@hadoop101 opt]# chown atguigu:atguigu software
```

```
[root@hadoop101 opt]# ls -al
```

总用量 16

```
drwxr-xr-x. 6 root    root 4096 4 月 24 09:07 .
```

```
dr-xr-xr-x. 23 root    root 4096 4 月 24 08:52 ..
```

```
drwxr-xr-x. 4 atguigu atguigu 4096 4 月 23 16:26 module
```

```
drwxr-xr-x. 2 atguigu atguigu 4096 4 月 23 16:25 software
```

3.7 安装 jdk

1) 卸载现有 jdk

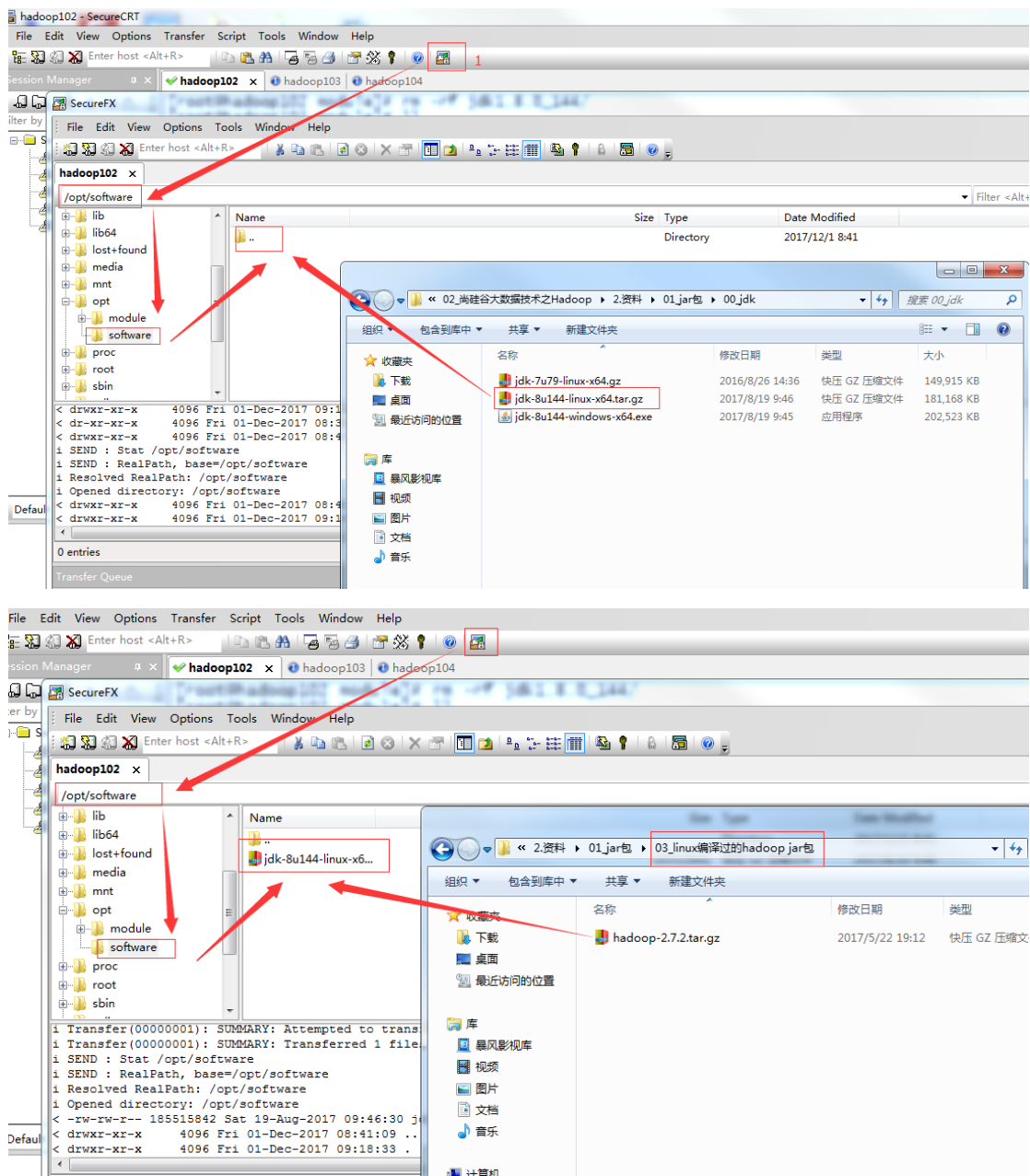
(1) 查询是否安装 java 软件:

```
[root@hadoop101 opt]# rpm -qa|grep java
```

(2) 如果安装的版本低于 1.7, 卸载该 jdk:

```
[root@hadoop101 opt]# rpm -e 软件包
```

2) 用 SecureCRT 工具将 jdk、Hadoop-2.7.2.tar.gz 导入到 opt 目录下面的 software 文件夹下面



3) 在 linux 系统下的 opt 目录中查看软件包是否导入成功。

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】


```
[root@hadoop101 opt]# cd software/
```

```
[root@hadoop101 software]# ls
```

```
hadoop-2.7.2.tar.gz  jdk-8u144-linux-x64.tar.gz
```

4) 解压 jdk 到/opt/module 目录下

```
[root@hadoop101 software]# tar -zxvf jdk-8u144-linux-x64.tar.gz -C /opt/module/
```

5) 配置 jdk 环境变量

(1) 先获取 jdk 路径:

```
[root@hadoop101 jdk1.8.0_144]# pwd
```

```
/opt/module/jdk1.8.0_144
```

(2) 打开/etc/profile 文件:

```
[root@hadoop101 jdk1.8.0_144]# vi /etc/profile
```

在 profile 文件末尾添加 jdk 路径:

```
##JAVA_HOME
```

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

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

(3) 保存后退出:

```
:wq
```

(4) 让修改后的文件生效:

```
[root@hadoop101 jdk1.8.0_144]# source /etc/profile
```

(5) 重启 (如果 java -version 可以用就不用重启):

```
[root@hadoop101 jdk1.8.0_144]# sync
```

```
[root@hadoop101 jdk1.8.0_144]# reboot
```

6) 测试 jdk 安装成功

```
[root@hadoop101 jdk1.8.0_144]# java -version
```

```
java version "1.8.0_144"
```

3.8 安装 Hadoop

1) 进入到 Hadoop 安装包路径下:

```
[root@hadoop101 ~]# cd /opt/software/
```

2) 解压安装文件到/opt/module 下面

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】

```
[root@hadoop101 software]# tar -zxvf hadoop-2.7.2.tar.gz -C /opt/module/
```

3) 查看是否解压成功

```
[root@hadoop101 software]# ls /opt/module/
```

```
hadoop-2.7.2
```

4) 在/opt/module/hadoop-2.7.2/etc/hadoop 路径下配置 hadoop-env.sh

(1) Linux 系统中获取 jdk 的安装路径:

```
[root@hadoop101 jdk1.8.0_144]# echo $JAVA_HOME  
  
/opt/module/jdk1.8.0_144
```

(2) 修改 hadoop-env.sh 文件中 JAVA_HOME 路径:

```
[root@hadoop101 hadoop]# vi hadoop-env.sh  
  
修改 JAVA_HOME 如下  
  
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

5) 将 hadoop 添加到环境变量

(1) 获取 hadoop 安装路径:

```
[root@hadoop101 hadoop-2.7.2]# pwd  
  
/opt/module/hadoop-2.7.2
```

(2) 打开/etc/profile 文件:

```
[root@hadoop101 hadoop-2.7.2]# vi /etc/profile
```

在 profile 文件末尾添加 jdk 路径: (shift+g)

```
##HADOOP_HOME
```

```
export HADOOP_HOME=/opt/module/hadoop-2.7.2
```

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

```
export PATH=$PATH:$HADOOP_HOME/sbin
```

(3) 保存后退出:

```
:wq
```

(4) 让修改后的文件生效:

```
[root@hadoop101 hadoop-2.7.2]# source /etc/profile
```

(5) 重启(如果 hadoop 命令不能用再重启):

```
[root@hadoop101 hadoop-2.7.2]# sync
```

```
[root@hadoop101 hadoop-2.7.2]# reboot
```

6) 修改/opt 目录下的所有文件所有者为 atguigu

```
[root@hadoop101 opt]# chown atguigu:atguigu -R /opt/
```

7) 切换到 atguigu 用户

```
[root@hadoop101 opt]# su atguigu
```

四 Hadoop 运行模式

1) 官方网址

(1) 官方网站:

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

(2) 各个版本归档库地址

<https://archive.apache.org/dist/hadoop/common/hadoop-2.7.2/>

(3) hadoop2.7.2 版本详情介绍

<http://hadoop.apache.org/docs/r2.7.2/>

2) Hadoop 运行模式

(1) 本地模式 (默认模式):

不需要启用单独进程, 直接可以运行, 测试和开发时使用。

(2) 伪分布式模式:

等同于完全分布式, 只有一个节点。

(3) 完全分布式模式:

多个节点一起运行。

4.1 本地运行 Hadoop 案例

4.1.1 官方 grep 案例

1) 创建在 hadoop-2.7.2 文件下面创建一个 input 文件夹

```
[atguigu@hadoop101 hadoop-2.7.2]$ mkdir input
```

2) 将 hadoop 的 xml 配置文件复制到 input

```
[atguigu@hadoop101 hadoop-2.7.2]$ cp etc/hadoop/*.xml input
```

3) 执行 share 目录下的 mapreduce 程序

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hadoop jar
```

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷 (中国) 官网 www.atguigu.com 下载区】

```
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar grep input output 'dfs[a-z.]+'
```

4) 查看输出结果

```
[atguigu@hadoop101 hadoop-2.7.2]$ cat output/*
```

4.1.2 官方 wordcount 案例

1) 创建在 hadoop-2.7.2 文件下面创建一个 wcinput 文件夹

```
[atguigu@hadoop101 hadoop-2.7.2]$ mkdir wcinput
```

2) 在 wcinput 文件下创建一个 wc.input 文件

```
[atguigu@hadoop101 hadoop-2.7.2]$ cd wcinput
```

```
[atguigu@hadoop101 wcinput]$ touch wc.input
```

3) 编辑 wc.input 文件

```
[atguigu@hadoop101 wcinput]$ vim wc.input
```

在文件中输入如下内容

```
hadoop yarn
```

```
hadoop mapreduce
```

```
atguigu
```

```
atguigu
```

```
保存退出: : wq
```

4) 回到 hadoop 目录/opt/module/hadoop-2.7.2

5) 执行程序:

```
[atguigu@hadoop101 hadoop-2.7.2]$ hadoop jar
```

```
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount wcinput wcoutput
```

6) 查看结果:

```
[atguigu@hadoop101 hadoop-2.7.2]$ cat wcoutput/part-r-00000
```

```
atguigu 2
```

```
hadoop 2
```

```
mapreduce 1
```

```
yarn 1
```


4.2 伪分布式运行 Hadoop 案例

4.2.1 启动 HDFS 并运行 MapReduce 程序

1) 分析:

- (1) 准备 1 台客户机
- (2) 安装 jdk
- (3) 配置环境变量
- (4) 安装 hadoop
- (5) 配置环境变量
- (6) 配置集群
- (7) 启动、测试集群增、删、查
- (8) 执行 wordcount 案例

2) 执行步骤

需要配置 hadoop 文件如下

(1) 配置集群

(a) 配置: `hadoop-env.sh`

Linux 系统中获取 jdk 的安装路径:

```
[root@hadoop101 ~]# echo $JAVA_HOME  
  
/opt/module/jdk1.8.0_144
```

修改 `JAVA_HOME` 路径:

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

(b) 配置: `core-site.xml`

```
<!-- 指定 HDFS 中 NameNode 的地址 -->  
<property>  
  <name>fs.defaultFS</name>  
  <value>hdfs://hadoop101:9000</value>  
</property>  
  
<!-- 指定 hadoop 运行时产生文件的存储目录 -->  
<property>  
  <name>hadoop.tmp.dir</name>  
  <value>/opt/module/hadoop-2.7.2/data/tmp</value>  
</property>
```

(c) 配置: hdfs-site.xml

```
<!-- 指定 HDFS 副本的数量 -->
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
```

(2) 启动集群

(a) 格式化 namenode (第一次启动时格式化, 以后就不要总格式化)

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs namenode -format
```

(b) 启动 namenode

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/hadoop-daemon.sh start namenode
```

(c) 启动 datanode

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/hadoop-daemon.sh start datanode
```

(3) 查看集群

(a) 查看是否启动成功

```
[atguigu@hadoop101 hadoop-2.7.2]$ jps
```

```
13586 NameNode
```

```
13668 DataNode
```

```
13786 Jps
```

(b) 查看产生的 log 日志

当前目录: /opt/module/hadoop-2.7.2/logs

```
[atguigu@hadoop101 logs]$ ls
```

```
hadoop-atguigu-datanode-hadoop.atguigu.com.log
```

```
hadoop-atguigu-datanode-hadoop.atguigu.com.out
```

```
hadoop-atguigu-namenode-hadoop.atguigu.com.log
```

```
hadoop-atguigu-namenode-hadoop.atguigu.com.out
```

```
SecurityAuth-root.audit
```

```
[atguigu@hadoop101 logs]# cat hadoop-atguigu-datanode-hadoop101.log
```

(c) web 端查看 HDFS 文件系统

<http://192.168.1.101:50070/dfshealth.html#tab-overview>

注意: 如果不能查看, 看如下帖子处理

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷(中国)官网 www.atguigu.com 下载区】

<http://www.cnblogs.com/zls1ch/p/6604189.html>

(4) 操作集群

(a) 在 hdfs 文件系统上**创建**一个 input 文件夹

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -mkdir -p /user/atguigu/input
```

(b) 将测试文件内容**上传**到文件系统上

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -put wcinput/wc.input  
/user/atguigu/input/
```

(c) **查看**上传的文件是否正确

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -ls /user/atguigu/input/  
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -cat /user/atguigu/  
input/wc.input
```

(d) 运行 mapreduce 程序

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hadoop jar  
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount /user/atguigu/input/  
/user/atguigu/output
```

(e) 查看输出结果

命令行查看:

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -cat /user/atguigu/output/*
```

浏览器查看

Browse Directory

/user/atguigu/output							Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	atguigu	supergroup	0 B	2017/12/1 上午11:05:18	1	128 MB	_SUCCESS
-rw-r--r--	atguigu	supergroup	38 B	2017/12/1 上午11:05:18	1	128 MB	part-r-00000

(f) 将测试文件内容**下载**到本地

```
[atguigu@hadoop101 hadoop-2.7.2]$ hadoop fs -get /user/atguigu/  
output/part-r-00000 ./wcoutput/
```

(g) **删除**输出结果

```
[atguigu@hadoop101 hadoop-2.7.2]$ hdfs dfs -rmr /user/atguigu/output
```

4.2.2 YARN 上运行 MapReduce 程序

1) 分析:

- (1) 准备 1 台客户机
- (2) 安装 jdk
- (3) 配置环境变量
- (4) 安装 hadoop
- (5) 配置环境变量
- (6) 配置集群 yarn 上运行
- (7) 启动、测试集群增、删、查
- (8) 在 yarn 上执行 wordcount 案例

2) 执行步骤

(1) 配置集群

(a) 配置 yarn-env.sh

配置一下 JAVA_HOME

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

(b) 配置 yarn-site.xml

```
<!-- reducer 获取数据的方式 -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>

<!-- 指定 YARN 的 ResourceManager 的地址 -->
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>hadoop101</value>
</property>
```

(c) 配置: mapred-env.sh

配置一下 JAVA_HOME

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

(d) 配置: (对 mapred-site.xml.template 重新命名为) mapred-site.xml

```
[atguigu@hadoop101 hadoop]$ mv mapred-site.xml.template mapred-site.xml
```

```
[atguigu@hadoop101 hadoop]$ vi mapred-site.xml
```

```
<!-- 指定 mr 运行在 yarn 上 -->
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
```

(2) 启动集群

(a) 启动 resourcemanager

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start resourcemanager
```

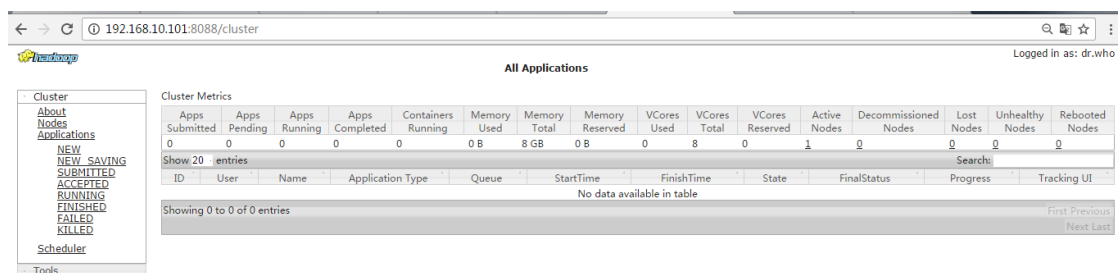
(b) 启动 nodemanager

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start nodemanager
```

(3) 集群操作

(a) yarn 的浏览器页面查看

<http://192.168.1.101:8088/cluster>



The screenshot shows the Hadoop YARN web interface. On the left is a navigation menu with links like Cluster, About, Nodes, Applications, NEW, SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, Scheduler, and Tools. The main area is titled 'All Applications' and shows 'Cluster Metrics' with various statistics. Below this is a table of applications, currently showing 'Showing 0 to 0 of 0 entries'.

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
Showing 0 to 0 of 0 entries										

(b) 删除文件系统上的 output 文件

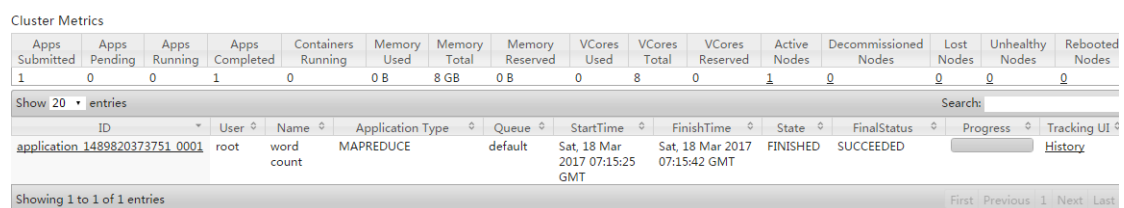
```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -rm -R /user/atguigu/output
```

(c) 执行 mapreduce 程序

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hadoop jar
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount /user/atguigu/input
/user/atguigu/output
```

(d) 查看运行结果

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -cat /user/atguigu/output/*
```



The screenshot shows the Hadoop YARN web interface with the 'All Applications' table displaying one application. The application is in the 'FINISHED' state with a 'SUCCEEDED' final status. The progress bar is at 100%.

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	0	1	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1489820373751_0001	root	word count	MAPREDUCE	default	Sat, 18 Mar 2017 07:15:25 GMT	Sat, 18 Mar 2017 07:15:42 GMT	FINISHED	SUCCEEDED	100%	History

4.2.3 配置临时文件存储路径

1) 停止进程

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh stop nodemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh stop resourcemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/hadoop-daemon.sh stop datanode
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/hadoop-daemon.sh stop namenode
```

2) 修改 `hadoop.tmp.dir`

[core-site.xml]

```
<!-- 指定 hadoop 运行时产生文件的存储目录 -->
<property>
  <name>hadoop.tmp.dir</name>
  <value>/opt/module/hadoop-2.7.2/data/tmp</value>
</property>
```

3) 将 `/opt/module/hadoop-2.7.2` 路径中的 `logs` 文件夹删除掉

```
[atguigu@hadoop101 hadoop-2.7.2]$ rm -rf logs/
```

4) 进入到 `tmp` 目录将 `tmp` 目录中 `hadoop-atguigu` 目录删除掉

```
[atguigu@hadoop101 tmp]$ rm -rf hadoop-atguigu/
```

5) 格式化 NameNode

```
[atguigu@hadoop101 hadoop-2.7.2]$ hadoop namenode -format
```

6) 启动所有进程

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/hadoop-daemon.sh start namenode
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/hadoop-daemon.sh start datanode
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start resourcemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start nodemanager
```

7) 查看 `/opt/module/hadoop-2.7.2/data/tmp` 这个目录下的内容。

4.2.4 配置历史服务器

1) 配置 `mapred-site.xml`

```
[atguigu@hadoop101 hadoop]$ vi mapred-site.xml
```

```
<property>
  <name>mapreduce.jobhistory.address</name>
  <value>hadoop101:10020</value>
```

```
</property>
<property>
  <name>mapreduce.jobhistory.webapp.address</name>
  <value>hadoop101:19888</value>
</property>
```

2) 查看启动历史服务器文件目录:

```
[atguigu@hadoop101 hadoop-2.7.2]$ ls sbin/ | grep mr
mr-jobhistory-daemon.sh
```

3) 启动历史服务器

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/mr-jobhistory-daemon.sh start historyserver
```

4) 查看历史服务器是否启动

```
[atguigu@hadoop101 hadoop-2.7.2]$ jps
```

5) 查看 jobhistory

```
http://192.168.1.101:19888/jobhistory
```

4.2.5 配置日志的聚集

日志聚集概念: 应用运行完成以后, 将日志信息上传到 HDFS 系统上。

开启日志聚集功能步骤:

(1) 配置 yarn-site.xml

```
[atguigu@hadoop101 hadoop]$ vi yarn-site.xml
```

```
<!-- 日志聚集功能使能 -->
<property>
  <name>yarn.log-aggregation-enable</name>
  <value>true</value>
</property>
<!-- 日志保留时间设置 7 天 -->
<property>
  <name>yarn.log-aggregation.retain-seconds</name>
  <value>604800</value>
</property>
```

(2) 关闭 nodemanager 、resourcemanager 和 historymanager

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh stop resourcemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh stop nodemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/mr-jobhistory-daemon.sh stop historyserver
```

(3) 启动 nodemanager 、resourcemanager 和 historymanager

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷(中国)官网 www.atguigu.com 下载区】

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start resourcemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start nodemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/mr-jobhistory-daemon.sh start historyserver
```

(4) 删除 hdfs 上已经存在的 hdfs 文件

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -rm -R /user/atguigu/output
```

(5) 执行 wordcount 程序

```
[atguigu@hadoop101 hadoop-2.7.2]$ hadoop jar
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar
wordcount
/user/atguigu/input /user/atguigu/output
```

(6) 查看日志

<http://192.168.1.101:19888/jobhistory>

192.168.101:19888/jobhistory

Logged in as: dr.wh

JobHistory

Retired Jobs

Show 20 entries

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed
2017.03.18 17:54:46 CST	2017.03.18 17:54:51 CST	2017.03.18 17:55:01 CST	job_1489830500161_0001	word count	root	default	SUCCEEDED	1	1	1	1
2017.03.18 17:20:31 CST	2017.03.18 17:20:39 CST	2017.03.18 17:20:50 CST	job_1489827711073_0001	word count	root	default	SUCCEEDED	1	1	1	1
2017.03.18 16:21:38 CST	2017.03.18 16:21:42 CST	2017.03.18 16:21:52 CST	job_1489820373751_0003	word count	root	default	SUCCEEDED	1	1	1	1
2017.03.18 15:29:57 CST	2017.03.18 15:30:02 CST	2017.03.18 15:30:12 CST	job_1489820373751_0002	word count	root	default	SUCCEEDED	1	1	1	1
2017.03.18 15:15:25 CST	2017.03.18 15:15:31 CST	2017.03.18 15:15:42 CST	job_1489820373751_0001	word count	root	default	SUCCEEDED	1	1	1	1

Logged in as: dr.wh

MapReduce Job job_1489830500161_0001

Job Overview

Job Name:	word count
User Name:	root
Queue:	default
State:	SUCCEEDED
Uberized:	false
Submitted:	Sat Mar 18 17:54:46 CST 2017
Started:	Sat Mar 18 17:54:51 CST 2017
Finished:	Sat Mar 18 17:55:01 CST 2017
Elapsed:	9sec
Diagnostics:	
Average Map Time:	2sec
Average Shuffle Time:	2sec
Average Merge Time:	0sec
Average Reduce Time:	0sec

ApplicationMaster	Attempt Number	Start Time	Node	Logs
	1	Sat Mar 18 17:54:49 CST 2017	hadoop.atguigu.com:8042	logs

Task Type	Total	Complete
Map	1	1
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	1
Reduces	0	0	1

192.168.10.101:19888/jobhistory/logs/hadoop.atguigu.com:43668/container_1489830500161_0001_01_000001/job_1489830500161_0001/root

🔍 🌐 ⭐



```
Log Type: stderr
Log Length: 222
log4j:WARN No appenders could be found for logger (org.apache.hadoop.ipc.Server).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

Log Type: stdout
Log Length: 312
Java HotSpot(TM) Server VM warning: You have loaded library /opt/module/hadoop-2.5.0/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c /libfile', or link it with '-z noexecstack'.

Log Type: syslog
Log Length: 34561
Showing 4096 bytes of 34561 total. Click here for the full log.
2017-03-18 17:55:02,058 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Copied to done location: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/
2017-03-18 17:55:02,060 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Copied hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/root/. staging/job_1489830
2017-03-18 17:55:02,082 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Copied to done location: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/
2017-03-18 17:55:02,088 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Moved tmp to done: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/done_i
2017-03-18 17:55:02,090 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Moved tmp to done: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/done_i
2017-03-18 17:55:02,090 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Stopped JobHistoryEventHandler. super.stop()
2017-03-18 17:55:02,092 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Setting job diagnostics to
2017-03-18 17:55:02,093 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: History url is http://hadoop.atguigu.com:19888/jobhistory/job/job_1489830500161_0001
2017-03-18 17:55:02,106 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Waiting for application to be successfully unregistered.
2017-03-18 17:55:03,112 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Final Stats: PendingHeds:0 ScheduledHeds:0 AssignedHeds:0 AssignedHeds:1 CompleteHeds:0
2017-03-18 17:55:03,113 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.WMApMaster: Deleting staging directory hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/root/. staging/job_148983050016
2017-03-18 17:55:03,119 INFO [Thread-64] org.apache.hadoop.ipc.Server: Stopping server on 56227
2017-03-18 17:55:03,120 INFO [IPC Server listener on 56227] org.apache.hadoop.ipc.Server: Stopping IPC Server listener on 56227
2017-03-18 17:55:03,122 INFO [TaskHeartbeatHandler PingChecker] org.apache.hadoop.mapreduce.v2.app.TaskHeartbeatHandler: TaskHeartbeatHandler thread interrupted
```

4.2.6 配置文件说明

Hadoop 配置文件分两类：默认配置文件和自定义配置文件，只有用户想修改某一默认配置值时，才需要修改自定义配置文件，更改相应属性值。

- (1) 默认配置文件：存放在 hadoop 相应的 jar 包中

[core-default.xml]

hadoop-common-2.7.2.jar/ core-default.xml

[hdfs-default.xml]

hadoop-hdfs-2.7.2.jar/ hdfs-default.xml

[yarn-default.xml]

hadoop-yarn-common-2.7.2.jar/ yarn-default.xml

[core-default.xml]

hadoop-mapreduce-client-core-2.7.2.jar/ core-default.xml

- (2) 自定义配置文件：存放在 \$HADOOP_HOME/etc/hadoop

core-site.xml

hdfs-site.xml

yarn-site.xml

mapred-site.xml

4.3 完全分布式部署 Hadoop

分析：

- 1) 准备 3 台客户机（关闭防火墙、静态 ip、主机名称）

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】

- 2) 安装 jdk
- 3) 配置环境变量
- 4) 安装 hadoop
- 5) 配置环境变量
- 6) 安装 ssh
- 7) 配置集群
- 8) 启动测试集群

4.3.1 虚拟机准备

详见 3.2-3.3 章。

4.3.2 主机名设置

详见 3.4 章。

4.3.3 scp

- 1) scp 可以实现服务器与服务器之间的数据拷贝。

- 2) 案例实操

(1) 将 hadoop101 中 /opt/module 和 /opt/software 文件拷贝到 hadoop102、hadoop103 和 hadoop104 上。

```
[root@hadoop101 /]# scp -r /opt/module/ root@hadoop102:/opt
[root@hadoop101 /]# scp -r /opt/software/ root@hadoop102:/opt
[root@hadoop101 /]# scp -r /opt/module/ root@hadoop103:/opt
[root@hadoop101 /]# scp -r /opt/software/ root@hadoop103:/opt
[root@hadoop101 /]# scp -r /opt/module/ root@hadoop104:/opt
[root@hadoop101 /]# scp -r /opt/software/ root@hadoop105:/opt
```

- (2) 将 hadoop101 服务器上的 /etc/profile 文件拷贝到 hadoop102 上。

```
[root@hadoop102 opt]# scp root@hadoop101:/etc/profile /etc/profile
```

- (3) 实现两台远程机器之间的文件传输 (hadoop103 主机文件拷贝到 hadoop104 主机上)

```
[atguigu@hadoop102 test]$ scp atguigu@hadoop103:/opt/test/haha
atguigu@hadoop104:/opt/test/
```


4.3.4 SSH 无密码登录

1) 配置 ssh

(1) 基本语法

ssh 另一台电脑的 ip 地址

(2) ssh 连接时出现 Host key verification failed 的解决方法

```
[root@hadoop102 opt]# ssh 192.168.1.103
```

The authenticity of host '192.168.1.103 (192.168.1.103)' can't be established.

RSA key fingerprint is cf:1e:de:d7:d0:4c:2d:98:60:b4:fd:ae:b1:2d:ad:06.

Are you sure you want to continue connecting (yes/no)?

Host key verification failed.

(3) 解决方案如下：直接输入 yes

2) 无密钥配置

(1) 进入到我的 home 目录

```
[atguigu@hadoop102 opt]$ cd ~/.ssh
```

(2) 生成公钥和私钥：

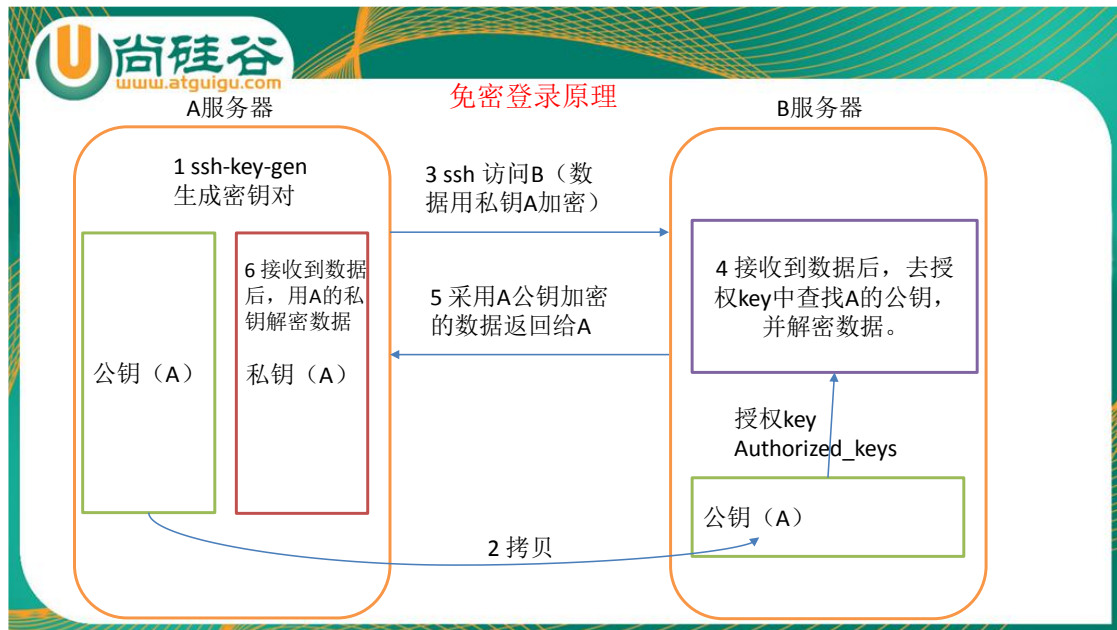
```
[atguigu@hadoop102 .ssh]$ ssh-keygen -t rsa
```

然后敲（三个回车），就会生成两个文件 id_rsa（私钥）、id_rsa.pub（公钥）

(3) 将公钥拷贝到要免密登录的目标机器上

```
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop103
```

```
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop104
```



3) .ssh 文件夹下的文件功能解释

- (1) ~/.ssh/known_hosts : 记录 ssh 访问过计算机的公钥(public key)
- (2) id_rsa : 生成的私钥
- (3) id_rsa.pub : 生成的公钥
- (4) authorized_keys : 存放授权过得无秘登录服务器公钥

4.3.5 rsync

rsync 远程同步工具，主要用于备份和镜像。具有速度快、避免复制相同内容和支持符号链接的优点。

rsync 和 scp 区别：用 rsync 做文件的复制要比 scp 的速度快，rsync 只对差异文件做更新。scp 是把所有文件都复制过去。

- (1) 查看 rsync 使用说明

`man rsync | more`

- (2) 基本语法

`rsync -rvl $pdir/$fname $user@hadoop$host:$pdir`

命令 命令参数 要拷贝的文件路径/名称 目的用户@主机:目的路径

选项

`-r` 递归

`-v` 显示复制过程

-l 拷贝符号连接

(3) 案例实操

把本机/opt/tmp 目录同步到 hadoop103 服务器的 root 用户下的/opt/tmp 目录

```
[atguigu@hadoop102 opt]$ rsync -rvl /opt/tmp root@hadoop103:/opt/
```

4.3.6 编写集群分发脚本 xsync

1) 需求分析：循环复制文件到所有节点的相同目录下。

(1) 原始拷贝：

```
rsync -rvl /opt/module root@hadoop103:/opt/
```

(2) 期望脚本：

xsync 要同步的文件名称

(3) 在/usr/local/bin 这个目录下存放的脚本，可以在系统任何地方直接执行。

2) 案例实操：

(1) 在/usr/local/bin 目录下创建 xsync 文件，文件内容如下：

```
[root@hadoop102 bin]# touch xsync
```

```
[root@hadoop102 bin]# vi xsync
```

```
#!/bin/bash
#1 获取输入参数个数，如果没有参数，直接退出
pcount=$#
if((pcount==0)); then
echo no args;
exit;
fi

#2 获取文件名称
p1=$1
fname=`basename $p1`
echo fname=$fname

#3 获取上级目录到绝对路径
pdir=`cd -P $(dirname $p1); pwd`
echo pdir=$pdir

#4 获取当前用户名称
user=`whoami`
```

```
#5 循环
for((host=103; host<105; host++)); do
    #echo $pdir/$fname $user@hadoop$host:$pdir
    echo ----- hadoop$host -----
    rsync -rvl $pdir/$fname $user@hadoop$host:$pdir
done
```

(2) 修改脚本 xsync 具有执行权限

```
[root@hadoop102 bin]# chmod 777 xsync
```

```
[root@hadoop102 bin]# chown atguigu:atguigu -R xsync
```

(3) 调用脚本形式: xsync 文件名称

```
[atguigu@hadoop102 opt]$ xsync tmp/
```

4.3.7 编写集群操作脚本 xcall

1) 需求分析: 在所有主机上同时执行相同的命令

xcall + 命令

2) 具体实现

(1) 在/usr/local/bin 目录下创建 xcall 文件, 文件内容如下:

```
[root@hadoop102 bin]# touch xcall
```

```
[root@hadoop102 bin]# vi xcall
```

```
#!/bin/bash
pcount=$#
if((pcount==0));then
    echo no args;
    exit;
fi

echo -----localhost-----
$@
for((host=101; host<=108; host++)); do
    echo -----hadoop$host-----
    ssh hadoop$host $@
done
```

(2) 修改脚本 xcall 具有执行权限

```
[root@hadoop102 bin]# chmod 777 xcall
```

```
[root@hadoop102 bin]# chown atguigu:atguigu xcall
```

(3) 调用脚本形式: xcall 操作命令

```
[root@hadoop102 ~]# xcall rm -rf /opt/tmp/
```

4.3.8 配置集群

1) 集群部署规划

	hadoop102	hadoop103	hadoop104
HDFS	NameNode		SecondaryNameNode
	DataNode	DataNode	DataNode
YARN	NodeManager	ResourceManager NodeManager	NodeManager

2) 配置文件

(1) core-site.xml

```
[atguigu@hadoop102 hadoop]$ vi core-site.xml
```

```
<!-- 指定 HDFS 中 NameNode 的地址 -->
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://hadoop102:9000</value>
</property>

<!-- 指定 hadoop 运行时产生文件的存储目录 -->
<property>
  <name>hadoop.tmp.dir</name>
  <value>/opt/module/hadoop-2.7.2/data/tmp</value>
</property>
```

(2) Hdfs

```
hadoop-env.sh
```

```
[atguigu@hadoop102 hadoop]$ vi hadoop-env.sh
```

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

```
hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>

  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>hadoop104:50090</value>
  </property>
```

```
</configuration>
```

```
slaves
```

```
[atguigu@hadoop102 hadoop]$ vi slaves
```

```
hadoop102
```

```
hadoop103
```

```
hadoop104
```

(3) yarn

```
yarn-env.sh
```

```
[atguigu@hadoop102 hadoop]$ vi yarn-env.sh
```

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

```
yarn-site.xml
```

```
[atguigu@hadoop102 hadoop]$ vi yarn-site.xml
```

```
<configuration>
```

```
<!-- reducer 获取数据的方式 -->
```

```
<property>
```

```
<name>yarn.nodemanager.aux-services</name>
```

```
<value>mapreduce_shuffle</value>
```

```
</property>
```

```
<!-- 指定 YARN 的 ResourceManager 的地址 -->
```

```
<property>
```

```
<name>yarn.resourcemanager.hostname</name>
```

```
<value>hadoop103</value>
```

```
</property>
```

```
</configuration>
```

(4) mapreduce

```
mapred-env.sh
```

```
[atguigu@hadoop102 hadoop]$ vi mapred-env.sh
```

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

```
mapred-site.xml
```

```
[atguigu@hadoop102 hadoop]$ vi mapred-site.xml
```

```
<configuration>
```

```
<!-- 指定 mr 运行在 yarn 上 -->
```

```
<property>
```

```
<name>mapreduce.framework.name</name>
```



```
<value>yarn</value>
</property>
</configuration>
```

3) 在集群上分发以上所有文件

```
[atguigu@hadoop102 hadoop]$ pwd
/opt/module/hadoop-2.7.2/etc/hadoop

[atguigu@hadoop102 hadoop]$ xsync /opt/module/hadoop-2.7.2/etc/hadoop/core-site.xml
[atguigu@hadoop102 hadoop]$ xsync /opt/module/hadoop-2.7.2/etc/hadoop/yarn-site.xml
[atguigu@hadoop102 hadoop]$ xsync /opt/module/hadoop-2.7.2/etc/hadoop/slaves
```

4) 查看文件分发情况

```
[atguigu@hadoop102 hadoop]$ xcall cat /opt/module/hadoop-2.7.2/etc/hadoop/slaves
```

4.3.9 集群启动及测试

1) 启动集群

(0) 如果集群是第一次启动, 需要格式化 namenode

```
[atguigu@hadoop102 hadoop-2.7.2]$ bin/hdfs namenode -format
```

(1) 启动 HDFS:

```
[atguigu@hadoop102 hadoop-2.7.2]$ sbin/start-dfs.sh
```

```
[atguigu@hadoop102 hadoop-2.7.2]$ jps
```

```
4166 NameNode
```

```
4482 Jps
```

```
4263 DataNode
```

```
[atguigu@hadoop103 hadoop-2.7.2]$ jps
```

```
3218 DataNode
```

```
3288 Jps
```

```
[atguigu@hadoop104 hadoop-2.7.2]$ jps
```

```
3221 DataNode
```

```
3283 SecondaryNameNode
```

```
3364 Jps
```

(2) 启动 yarn

```
[atguigu@hadoop102 hadoop-2.7.2]$ sbin/start-yarn.sh
```

注意: Namenode 和 ResourceManger 如果不是同一台机器, 不能在 NameNode 上启动 yarn, 应该在 ResouceManager 所在的机器上启动 yarn。

2) 集群基本测试

(1) 上传文件到集群

上传小文件

```
[atguigu@hadoop102 hadoop-2.7.2]$ bin/hdfs dfs -mkdir -p /user/atguigu/tmp/conf
```

```
[atguigu@hadoop102 hadoop-2.7.2]$ bin/hdfs dfs -put etc/hadoop/*-site.xml  
/user/atguigu/tmp/conf
```

上传大文件

```
[atguigu@hadoop102 hadoop-2.7.2]$ bin/hadoop fs -put  
/opt/software/hadoop-2.7.2.tar.gz /user/atguigu/input
```

(2) 上传文件后查看文件存放在什么位置

文件存储路径

```
[atguigu@hadoop102 subdir0]$ pwd  
  
/opt/module/hadoop-2.7.2/data/tmp/dfs/data/current/BP-938951106-192.168.10.107-149  
5462844069/current/finalized/subdir0/subdir0
```

查看文件内容

```
[atguigu@hadoop102 subdir0]$ cat blk_1073741825
```

hadoop

atguigu

atguigu

(3) 拼接

```
-rw-rw-r--. 1 atguigu atguigu 134217728 5 月 23 16:01 blk_1073741836  
-rw-rw-r--. 1 atguigu atguigu 1048583 5 月 23 16:01 blk_1073741836_1012.meta  
-rw-rw-r--. 1 atguigu atguigu 63439959 5 月 23 16:01 blk_1073741837  
-rw-rw-r--. 1 atguigu atguigu 495635 5 月 23 16:01 blk_1073741837_1013.meta  
  
[atguigu@hadoop102 subdir0]$ cat blk_1073741836>>tmp.file
```

```
[atguigu@hadoop102 subdir0]$ cat blk_1073741837>>tmp.file
```

```
[atguigu@hadoop102 subdir0]$ tar -zxvf tmp.file
```

(4) 下载

```
[atguigu@hadoop102 hadoop-2.7.2]$ bin/hadoop fs -get  
/user/atguigu/input/hadoop-2.7.2.tar.gz
```

3) 性能测试集群

写海量数据

读海量数据

4.3.10 Hadoop 启动停止方式

1) 各个服务组件逐一启动

(1) 分别启动 hdfs 组件

```
hadoop-daemon.sh start|stop namenode|datanode|secondarynamenode
```

(2) 启动 yarn

```
yarn-daemon.sh start|stop resourcemanager|nodemanager
```

2) 各个模块分开启动 (配置 ssh 是前提) 常用

(1) 整体启动/停止 hdfs

```
start-dfs.sh
```

```
stop-dfs.sh
```

(2) 整体启动/停止 yarn

```
start-yarn.sh
```

```
stop-yarn.sh
```

3) 全部启动 (不建议使用)

```
start-all.sh
```

```
stop-all.sh
```

4.3.11 集群时间同步

时间同步的方式: 找一个机器, 作为时间服务器, 所有的机器与这台集群时间进行定时的同步, 比如, 每隔十分钟, 同步一次时间。

配置时间同步实操:

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷 (中国) 官网 www.atguigu.com 下载区】

1) 时间服务器配置 (必须 root 用户)

(1) 检查 ntp 是否安装

```
[root@hadoop102 桌面]# rpm -qa|grep ntp  
ntp-4.2.6p5-10.el6.centos.x86_64  
fontpackages-filesystem-1.41-1.1.el6.noarch  
ntpdate-4.2.6p5-10.el6.centos.x86_64
```

(2) 修改 ntp 配置文件

```
[root@hadoop102 桌面]# vi /etc/ntp.conf
```

修改内容如下

a) 修改 1

```
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap 为  
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

b) 修改 2

```
server 0.centos.pool.ntp.org iburst  
server 1.centos.pool.ntp.org iburst  
server 2.centos.pool.ntp.org iburst  
server 3.centos.pool.ntp.org iburst 为  
#server 0.centos.pool.ntp.org iburst  
#server 1.centos.pool.ntp.org iburst  
#server 2.centos.pool.ntp.org iburst  
#server 3.centos.pool.ntp.org iburst
```

c) 添加 3

```
server 127.127.1.0  
fudge 127.127.1.0 stratum 10
```

(3) 修改/etc/sysconfig/ntpd 文件

```
[root@hadoop102 桌面]# vim /etc/sysconfig/ntpd
```

增加内容如下

```
SYNC_HWCLOCK=yes
```

(4) 重新启动 ntpd

```
[root@hadoop102 桌面]# service ntpd status
```

ntpd 已停

```
[root@hadoop102 桌面]# service ntpd start
```

正在启动 ntpd:

[确定]

(5) 执行:

```
[root@hadoop102 桌面]# chkconfig ntpd on
```

2) 其他机器配置 (必须 root 用户)

(1) 在其他机器配置 10 分钟与时间服务器同步一次

```
[root@hadoop103 hadoop-2.7.2]# crontab -e
```

编写脚本

```
*/10 * * * * /usr/sbin/ntpdate hadoop102
```

(2) 修改任意机器时间

```
[root@hadoop103 hadoop]# date -s "2017-9-11 11:11:11"
```

(3) 十分钟后查看机器是否与时间服务器同步

```
[root@hadoop103 hadoop]# date
```

4.3.12 配置集群常见问题

1) 防火墙没关闭、或者没有启动 yarn

```
INFO client.RMProxy: Connecting to ResourceManager at hadoop108/192.168.10.108:8032
```

2) 主机名称配置错误

3) ip 地址配置错误

4) ssh 没有配置好

5) root 用户和 atguigu 两个用户启动集群不统一

6) 配置文件修改不细心

7) 未编译源码

```
Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

```
17/05/22 15:38:58 INFO client.RMProxy: Connecting to ResourceManager at hadoop108/192.168.10.108:8032
```

8) datanode 不被 namenode 识别问题

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】

Namenode 在 format 初始化的时候会形成两个标识, blockPoolId 和 clusterId。新的 datanode 加入时, 会获取这两个标识作为自己工作目录中的标识。

一旦 namenode 重新 format 后, namenode 的身份标识已变, 而 datanode 如果依然持有原来的 id, 就不会被 namenode 识别。

解决办法, 删除 datanode 节点中的数据后, 再次重新格式化 namenode。

9) 不识别主机名称

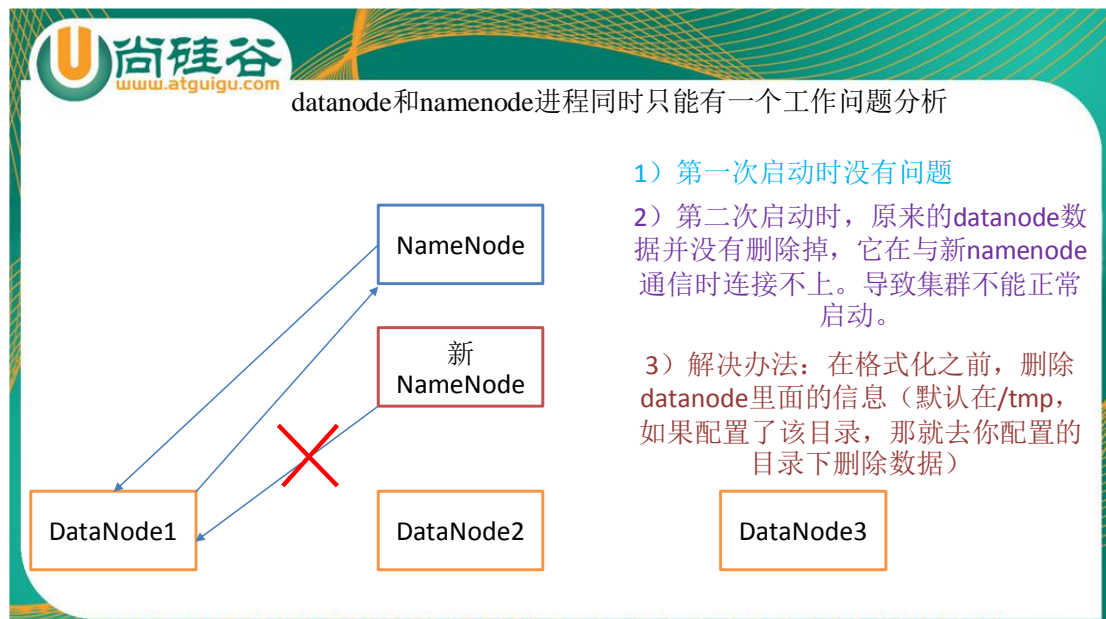
```
java.net.UnknownHostException: hadoop102: hadoop102

    at java.net.InetAddress.getLocalHost(InetAddress.java:1475)
    at
org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:146)
    at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1290)
    at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1287)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:415)
```

解决办法:

- (1) 在/etc/hosts 文件中添加 192.168.1.102 hadoop102
- (2) 主机名称不要起 hadoop hadoop000 等特殊名称

10) datanode 和 namenode 进程同时只能工作一个。



11) 执行命令不生效, 粘贴 word 中命令时, 遇到-和长-没区分开。导致命令失效

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷 (中国) 官网 www.atguigu.com 下载区】

解决办法：尽量不要粘贴 word 中代码。

12) jps 发现进程已经没有，但是重新启动集群，提示进程已经开启。原因是在 linux 的根目录下/tmp 目录中存在启动的进程临时文件，将集群相关进程删除掉，再重新启动集群。

13) jps 不生效。

原因：全局变量 `hadoop java` 没有生效，需要 `source /etc/profile` 文件。

14) 8088 端口连接不上

```
[atguigu@hadoop102 桌面]$ cat /etc/hosts
```

注释掉如下代码

```
#127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
#::1        hadoop102
```

五 Hadoop 编译源码

5.1 前期准备工作

1) CentOS 联网

配置 CentOS 能连接外网。Linux 虚拟机 ping www.baidu.com 是畅通的

注意：采用 **root** 角色编译，减少文件夹权限出现问题

2) jar 包准备(hadoop 源码、JDK7 、 maven、 ant 、 protobuf)

(1) `hadoop-2.7.2-src.tar.gz`

(2) `jdk-7u79-linux-x64.gz`

(3) `apache-ant-1.9.9-bin.tar.gz`

(4) `apache-maven-3.0.5-bin.tar.gz`

(5) `protobuf-2.5.0.tar.gz`

5.2 jar 包安装

0) 注意：所有操作必须在 **root** 用户下完成

1) JDK 解压、配置环境变量 `JAVA_HOME` 和 `PATH`，验证 `java-version`(如下都需要验证是否配置成功)

```
[root@hadoop101 software] # tar -zxf jdk-7u79-linux-x64.gz -C /opt/module/
```

```
[root@hadoop101 software]# vi /etc/profile
```

```
#JAVA_HOME
```

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 www.atguigu.com 下载区】

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
export PATH=$PATH:$JAVA_HOME/bin
```

```
[root@hadoop101 software]#source /etc/profile
```

验证命令: `java -version`

2) Maven 解压、配置 MAVEN_HOME 和 PATH。

```
[root@hadoop101 software]# tar -zxvf apache-maven-3.0.5-bin.tar.gz -C /opt/module/
```

```
[root@hadoop101 apache-maven-3.0.5]# vi /etc/profile
```

```
#MAVEN_HOME
export MAVEN_HOME=/opt/module/apache-maven-3.0.5
export PATH=$PATH:$MAVEN_HOME/bin
```

```
[root@hadoop101 software]#source /etc/profile
```

验证命令: `mvn -version`

3) ant 解压、配置 ANT_HOME 和 PATH。

```
[root@hadoop101 software]# tar -zxvf apache-ant-1.9.9-bin.tar.gz -C /opt/module/
```

```
[root@hadoop101 apache-ant-1.9.9]# vi /etc/profile
```

```
#ANT_HOME
export ANT_HOME=/opt/module/apache-ant-1.9.9
export PATH=$PATH:$ANT_HOME/bin
```

```
[root@hadoop101 software]#source /etc/profile
```

验证命令: `ant -version`

4) 安装 glibc-headers 和 g++ 命令如下:

```
[root@hadoop101 apache-ant-1.9.9]# yum install glibc-headers
```

```
[root@hadoop101 apache-ant-1.9.9]# yum install gcc-c++
```

5) 安装 make 和 cmake

```
[root@hadoop101 apache-ant-1.9.9]# yum install make
```

```
[root@hadoop101 apache-ant-1.9.9]# yum install cmake
```

6) 解压 protobuf , 进入到解压后 **protobuf** 主目录, /opt/module/protobuf-2.5.0

然后相继执行命令:

```
[root@hadoop101 software]# tar -zxvf protobuf-2.5.0.tar.gz -C /opt/module/
```

```
[root@hadoop101 opt]# cd /opt/module/protobuf-2.5.0/
```

```
[root@hadoop101 protobuf-2.5.0]# ./configure
```

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷(中国)官网 www.atguigu.com 下载区】

```
[root@hadoop101 protobuf-2.5.0]# make
[root@hadoop101 protobuf-2.5.0]# make check
[root@hadoop101 protobuf-2.5.0]# make install
[root@hadoop101 protobuf-2.5.0]# ldconfig
```

```
[root@hadoop101 hadoop-dist]# vi /etc/profile
```

```
#LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/opt/module/protobuf-2.5.0
export PATH=$PATH:$LD_LIBRARY_PATH
```

```
[root@hadoop101 software]#source /etc/profile
```

验证命令: **protoc --version**

7) 安装 openssl 库

```
[root@hadoop101 software]#yum install openssl-devel
```

8) 安装 ncurses-devel 库:

```
[root@hadoop101 software]#yum install ncurses-devel
```

到此, 编译工具安装基本完成。

5.3 编译源码

1) 解压源码到/opt/tools 目录

```
[root@hadoop101 software]# tar -zxvf hadoop-2.7.2-src.tar.gz -C /opt/
```

2) 进入到 hadoop 源码主目录

```
[root@hadoop101 hadoop-2.7.2-src]# pwd
/opt/hadoop-2.7.2-src
```

3) 通过 maven 执行编译命令

```
[root@hadoop101 hadoop-2.7.2-src]#mvn package -Pdist,native -DskipTests -Dtar
```

等待时间 30 分钟左右, 最终成功是全部 SUCCESS。

```
[ INFO] Apache Hadoop Common ..... SUCCESS [3:35.094s]
[ INFO] Apache Hadoop NFS ..... SUCCESS [5.004s]
[ INFO] Apache Hadoop KMS ..... SUCCESS [54.027s]
[ INFO] Apache Hadoop Common Project ..... SUCCESS [0.022s]
[ INFO] Apache Hadoop HDFS ..... SUCCESS [3:58.444s]
[ INFO] Apache Hadoop HttpFS ..... SUCCESS [1:02.562s]
[ INFO] Apache Hadoop HDFS BookKeeper Journal ..... SUCCESS [33.138s]
[ INFO] Apache Hadoop HDFS-NFS ..... SUCCESS [3.993s]
[ INFO] Apache Hadoop HDFS Project ..... SUCCESS [0.022s]
[ INFO] hadoop-yarn ..... SUCCESS [0.037s]
[ INFO] hadoop-yarn-api ..... SUCCESS [1:26.119s]
[ INFO] hadoop-yarn-common ..... SUCCESS [1:20.025s]
[ INFO] hadoop-yarn-server ..... SUCCESS [0.168s]
[ INFO] hadoop-yarn-server-common ..... SUCCESS [9.107s]
[ INFO] hadoop-yarn-server-nodemanager ..... SUCCESS [19.867s]
[ INFO] hadoop-yarn-server-web-proxy ..... SUCCESS [3.397s]
[ INFO] hadoop-yarn-server-applicationhistoryservice ..... SUCCESS [7.432s]
[ INFO] hadoop-yarn-server-resourcemanager ..... SUCCESS [17.078s]
[ INFO] hadoop-yarn-server-tests ..... SUCCESS [3.998s]
[ INFO] hadoop-yarn-client ..... SUCCESS [5.962s]
[ INFO] hadoop-yarn-server-sharedcachemanager ..... SUCCESS [2.803s]
[ INFO] hadoop-yarn-applications ..... SUCCESS [0.024s]
[ INFO] hadoop-yarn-applications-distributedshell ..... SUCCESS [1.841s]
[ INFO] hadoop-yarn-applications-unmanaged-am-launcher .... SUCCESS [1.876s]
```

4) 成功的 64 位 hadoop 包在/opt/hadoop-2.7.2-src/hadoop-dist/target 下。

```
[root@hadoop101 target]# pwd
```

```
/opt/hadoop-2.7.2-src/hadoop-dist/target
```

5.4 常见的问题及解决方案

1) MAVEN install 时候 JVM 内存溢出

处理方式: 在环境配置文件和 maven 的执行文件均可调整 MAVEN_OPT 的 heap 大小。

(详情查阅 MAVEN 编译 JVM 调优问题, 如:

<http://outofmemory.cn/code-snippet/12652/maven-outofmemoryerror-method>)

2) 编译期间 maven 报错。可能网络阻塞问题导致依赖库下载不完整导致, 多次执行命令 (一次通过比较难):

```
[root@hadoop101 hadoop-2.7.2-src]#mvn package -Pdist,native -DskipTests -Dtar
```

3) 报 ant、protobuf 等错误, 插件下载不完整或者插件版本问题, 最开始链接有较多特殊情况, 同时推荐

2.7.0 版本的问题汇总帖子 <http://www.tuicool.com/articles/IBn63qf>