尚硅谷大数据技术之 Hadoop（入门）

(作者：尚硅谷大数据研发部)

版本：V2.0

第 1 章 大数据概论

1.1 大数据概念

大数据概念如图 2-1 所示。

图 2-1 大数据概念

1.2 大数据特点（4V）

大数据特点如图 2-2，2-3，2-4，2-5 所示

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。

图 2-2 大数据特点之大量

2、Velocity（高速）

这是大数据区分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

图 2-3 大数据特点之高速

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。

订单数据

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

网络日志

让天下没有难学的技术

图 2-4 大数据特点之多样

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。

图 2-5 大数据特点之低价值密度

1.3 大数据应用场景

大数据应用场景如图 2-6，2-7，2-8，2-9，2-10，2-11 所示

1、物流仓储：大数据分析系统助力商家精细化运营、提升销量、节约成本。

让天下没有难学的技术

图 2-6 大数据应用场景之物流仓储

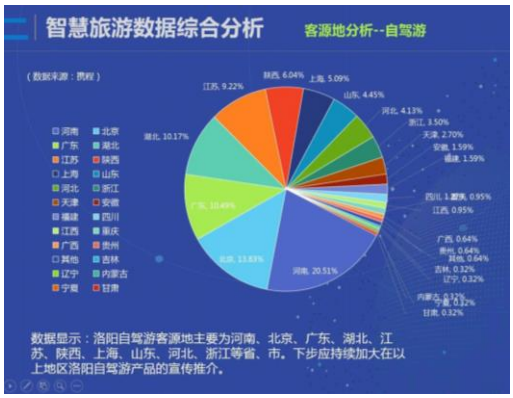
2、零售：分析用户消费习惯，为用户购买商品提供方便，从而提升商品销量。

经典案例，子尿布+啤酒。

让天下没有难学的技术

图 2-7 大数据应用场景之零售

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。



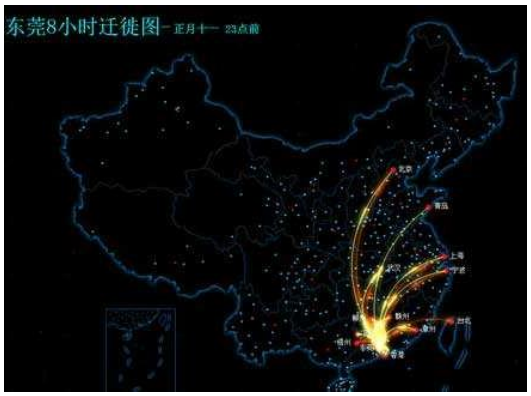


图 2-8 大数据应用场景之旅游

4、商品广告推荐：给用户推荐可能喜欢的商品

图 2-9 大数据应用场景之商品广告推荐

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。

图 2-10 大数据应用场景之保险、金融及房产

8、人工智能：

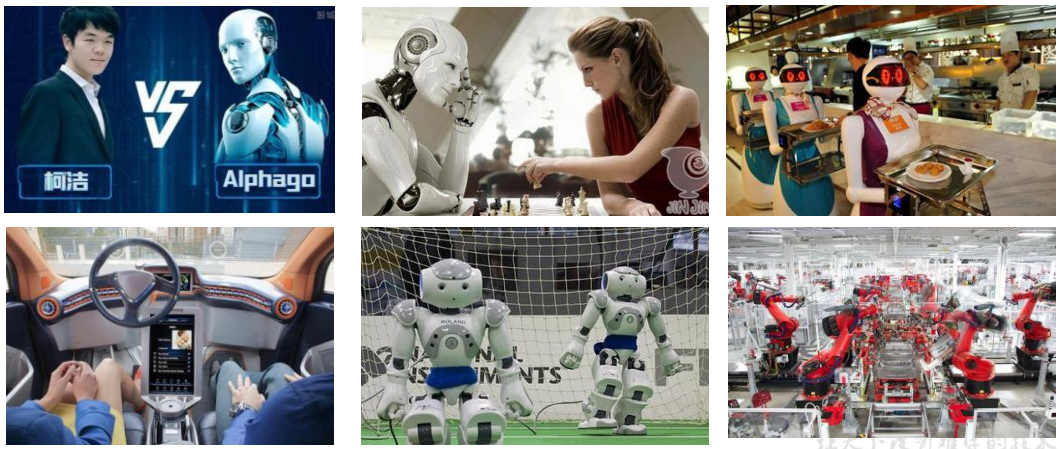


图 2-11 大数据应用场景人工智能

1.4 大数据发展前景

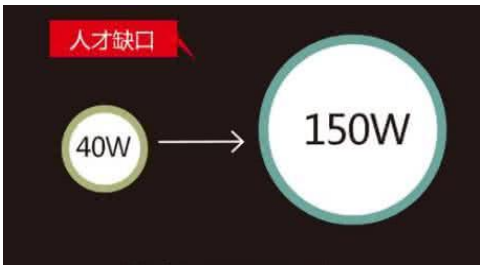
大数据发展前景如图 2-12，2-13，2-14，2-15，2-16 所示

1、党的十八大提出“**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。

2、党的十九大提出“**推动互联网、大数据、人工智能和实体经济深度融合**”。

图 2-12 大数据发展前景之国家政策

3、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/5年 = 30w/年

30w/12月=2.5w/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！

图 2-13 大数据发展前景之国际方面

4、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。

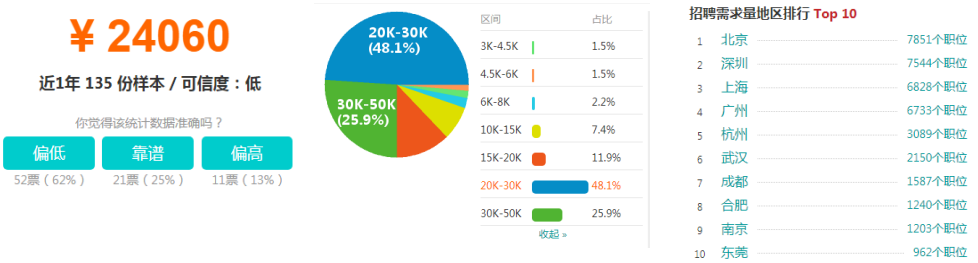
5、大数据属于高新技术，大牛少，升职竞争小；

让天下没有难学的技术

图 2-14 大数据发展前景之高校方面

大数据发展前景 尚硅谷

6、在北京大数据开发工程师的平均薪水已经到24060元（数据统计来职友集），而且目前还保持强劲的发展势头。



让天下没有难学的技术

图 2-15 大数据发展前景之平均薪资

7、智联招聘网站上的大数据工程师薪水如下

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
大数据研发工程师	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

让天下没有难学的技术

图 2-16 大数据发展前景之整体薪资

1.5 大数据部门业务流程分析

大数据部门业务流程分析如图 2-17 所示。

图 2-17 大数据部门业务流程分析

1.6 大数据部门组织结构（重点）

大数据部门组织结构，适用于大中型企业，如图 2-18 所示。

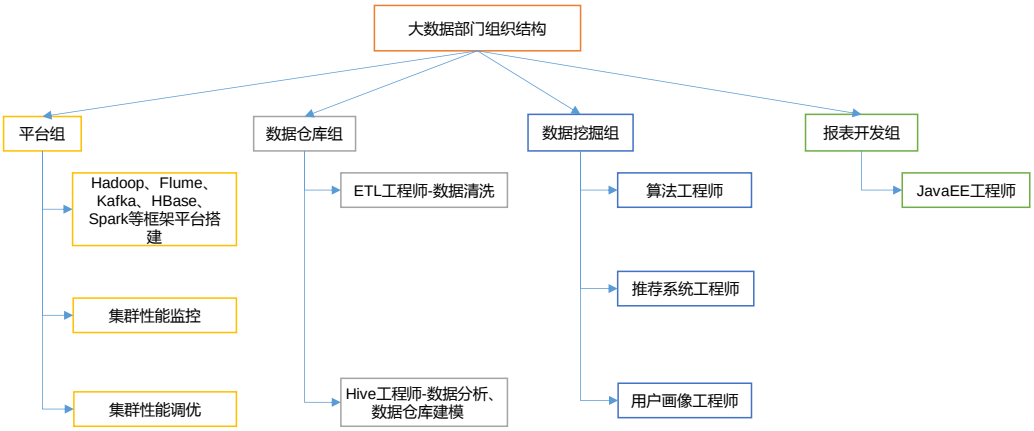


图 2-18 大数据部门组织结构

第 2 章 从 Hadoop 框架讨论大数据生态

2.1 Hadoop 是什么

2.2 Hadoop 发展历史

Hadoop发展历史

1）Lucene框架是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。

Hadoop创始人Doug Cutting

2）2001年年底Lucene成为Apache基金会的一个子项目。

3）对于海量数据的场景，Lucene面对与Google同样的困难，**存储数据困难，检索速度慢。**

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

让天下没有难学的技术

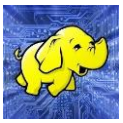
Hadoop发展历史

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。

8）2006 年 3 月份，Map-Reduce和Nutch Distributed File System (NDFS) 分别被纳入到 Hadoop 项目中，Hadoop就此正式诞生，标志着大数据时代来临。

9）名字来源于Doug Cutting儿子的玩具大象，如图2-20。



Hadoop的logo

让天下没有难学的技术

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/

（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 Hadoop80%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（4 高）

Hadoop的优势（4高）

1）高可靠性：Hadoop底层维护多个数据副本，所以即使Hadoop某个计算元素或存储出现故障，也不会导致数据的丢失。

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。

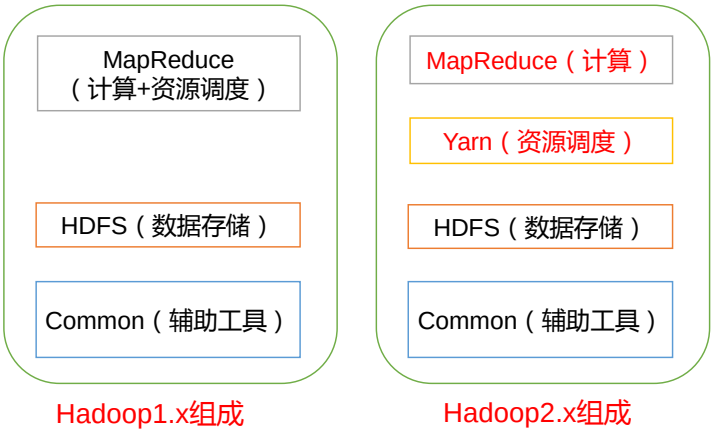
3）高效性：在MapReduce的思想下，Hadoop是并行工作的，以加快任务处理速度。

4）高容错性：能够自动将失败的任务重新分配。

让天下没有难学的技术

2.5 Hadoop 组成（面试重点）

Hadoop1.x和Hadoop2.x区别



在Hadoop1.x时代，Hadoop中的MapReduce同时处理业务逻辑运算和资源的调度，耦合性较大，在Hadoop2.x时代，增加了Yarn。Yarn只负责资源的调度，MapReduce只负责运算。

让天下没有难学的技术

图 2-21 Hadoop1.x 与 Hadoop2.x 的区别

2.5.1 HDFS 架构概述

HDFS（Hadoop Distributed File System）的架构概述，如图 2-23 所示。

图 2-23 HDFS 架构概述

2.5.2 YARN 架构概述

YARN 架构概述，如图 2-24 所示。

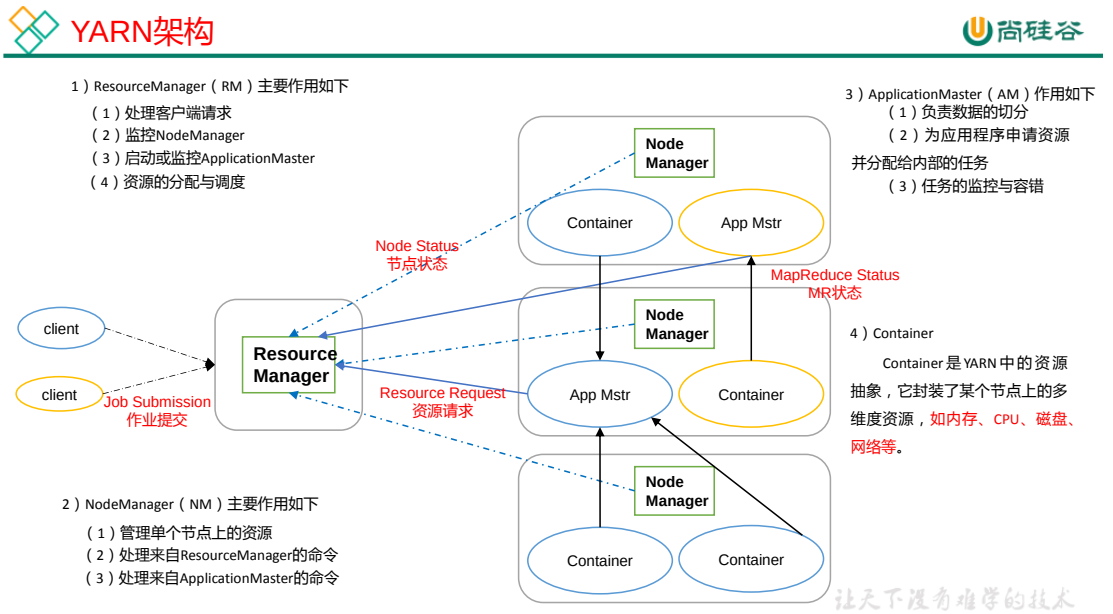


图 2-24　YARN 架构概述

2.5.3 MapReduce 架构概述

MapReduce 将计算过程分为两个阶段：Map 和 Reduce，如图 2-25 所示

1）Map 阶段并行处理输入数据

2）Reduce 阶段对 Map 结果进行汇总

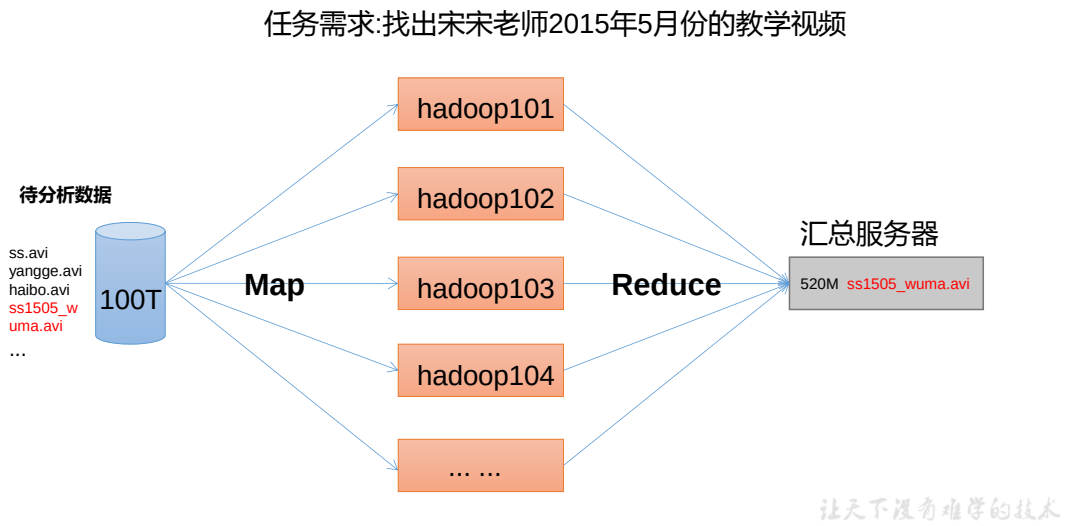


图 2-25 MapReduce 架构概述

2.6 大数据技术生态体系

大数据技术生态体系如图 2-26 所示。

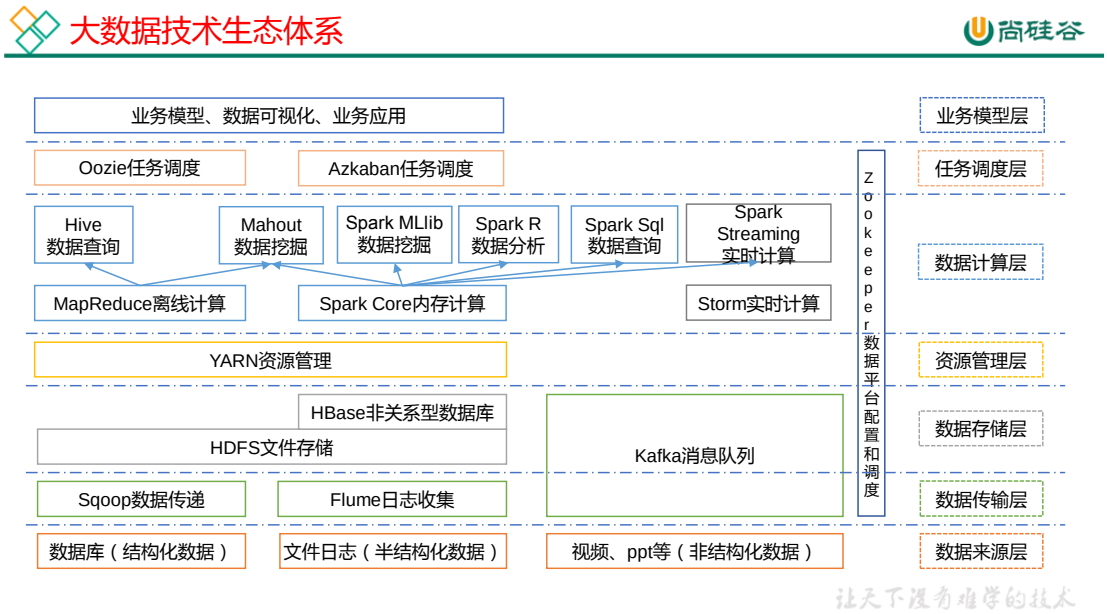


图 2-26　大数据技术生态体系

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 用于“连续计算”，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。

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。

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

推荐系统项目架构如图 2-27 所示。

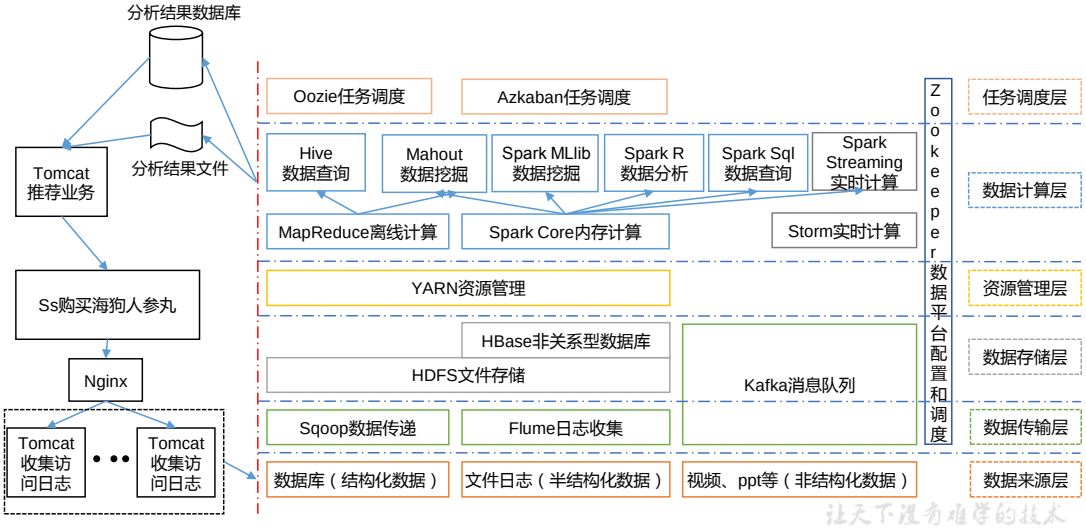


图 2-27 推荐系统项目架构

第 3 章 Hadoop 运行环境搭建（开发重点）

3.1 虚拟机环境准备

1. 克隆虚拟机
2. 修改克隆虚拟机的静态 IP
3. 修改主机名
4. 关闭防火墙
5. 创建 atguigu 用户
6. 配置 atguigu 用户具有 root 权限（详见《尚硅谷大数据技术之 Linux》）
7. 在/opt 目录下创建文件夹

（1）在/opt 目录下创建 module、software 文件夹

```
[atguigu@hadoop101 opt]$ sudo mkdir module
[atguigu@hadoop101 opt]$ sudo mkdir software
```

（2）修改 module、software 文件夹的所有者 cd

```
[atguigu@hadoop101 opt]$ sudo chown atguigu:atguigu module/software/
[atguigu@hadoop101 opt]$ ll
总用量 8
drwxr-xr-x. 2 atguigu atguigu 4096 1 月  17 14:37 module
drwxr-xr-x. 2 atguigu atguigu 4096 1 月  17 14:38 software
```

3.2 安装 JDK

1. 卸载现有 JDK

（1）查询是否安装 Java 软件：

```
[atguigu@hadoop101 opt]$ rpm -qa | grep java
```

（2）如果安装的版本低于 1.7，卸载该 JDK：

```
[atguigu@hadoop101 opt]$ sudo rpm -e 软件包
```

（3）查看 JDK 安装路径：

```
[atguigu@hadoop101 ~]$ which java
```

2. 用 SecureCRT 工具将 JDK 导入到 opt 目录下面的 software 文件夹下面，如图 2-28 所示

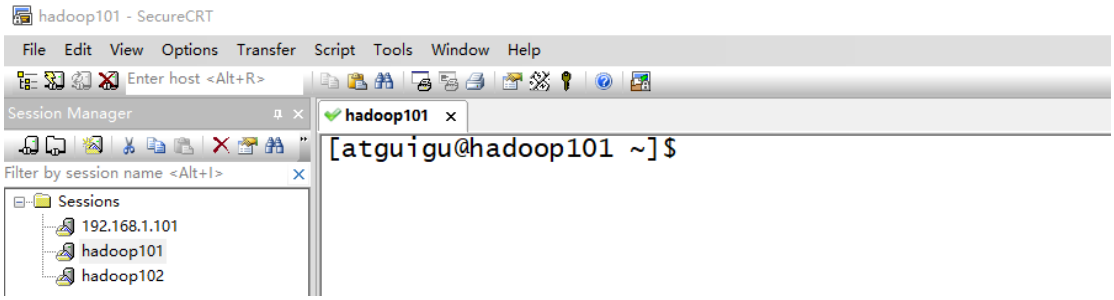


图 2-28 导入 JDK

“alt+p”进入 sftp 模式，如图 2-29 所示

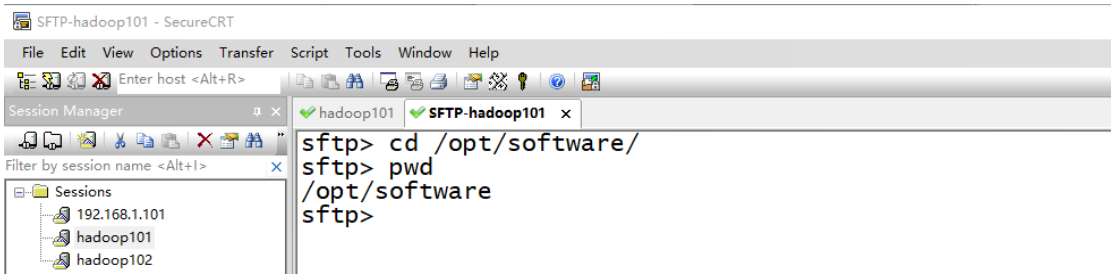


图 2-29 进入 sftp 模式

选择 jdk1.8 拖入，如图 2-30，2-31 所示

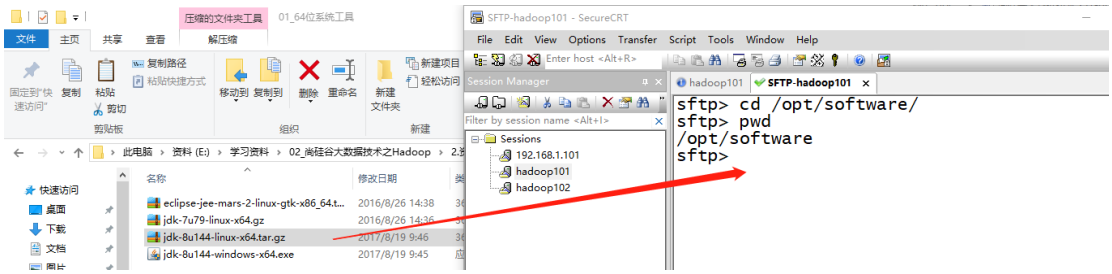


图 2-30 拖入 jdk1.8

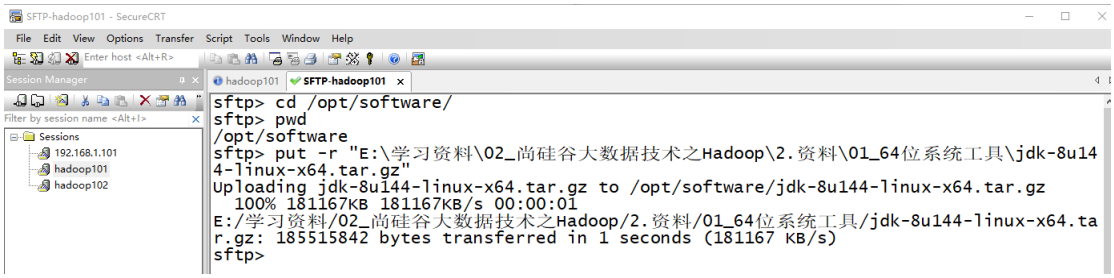


图 2-31 拖入 jdk1.8 完成

3. 在 Linux 系统下的 opt 目录中查看软件包是否导入成功

```
[atguigu@hadoop101 opt]$ cd software/
[atguigu@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

```
[atguigu@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

```
[atguigu@hadoop101 jdk1.8.0_144]$ pwd
/opt/module/jdk1.8.0_144
```

（2）打开/etc/profile 文件

```
[atguigu@hadoop101 software]$ sudo 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

```
[atguigu@hadoop101 jdk1.8.0_144]$ source /etc/profile
```

6. 测试 JDK 是否安装成功

```
[atguigu@hadoop101 jdk1.8.0_144]# java -version
java version "1.8.0_144"
```

注意：重启（如果 java -version 可以用就不用重启）

```
[atguigu@hadoop101 jdk1.8.0_144]$ sync
[atguigu@hadoop101 jdk1.8.0_144]$ sudo reboot
```

3.3 安装 Hadoop

0. Hadoop 下载地址：

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

1. 用 SecureCRT 工具将 hadoop-2.7.2.tar.gz 导入到 opt 目录下面的 software 文

件夹下面

切换到 sftp 连接页面，选择 Linux 下编译的 hadoop jar 包拖入，如图 2-32 所示

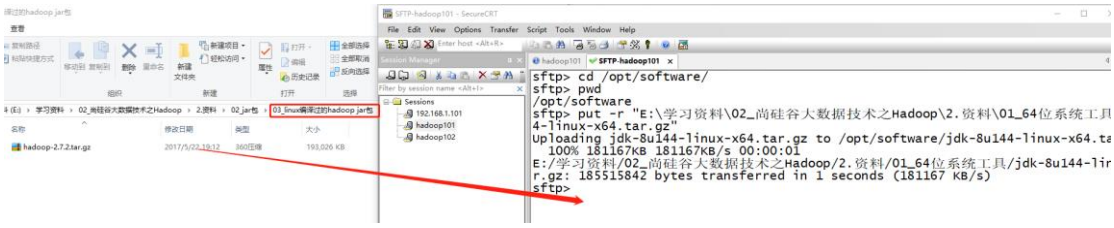


图 2-32 拖入 hadoop 的 tar 包

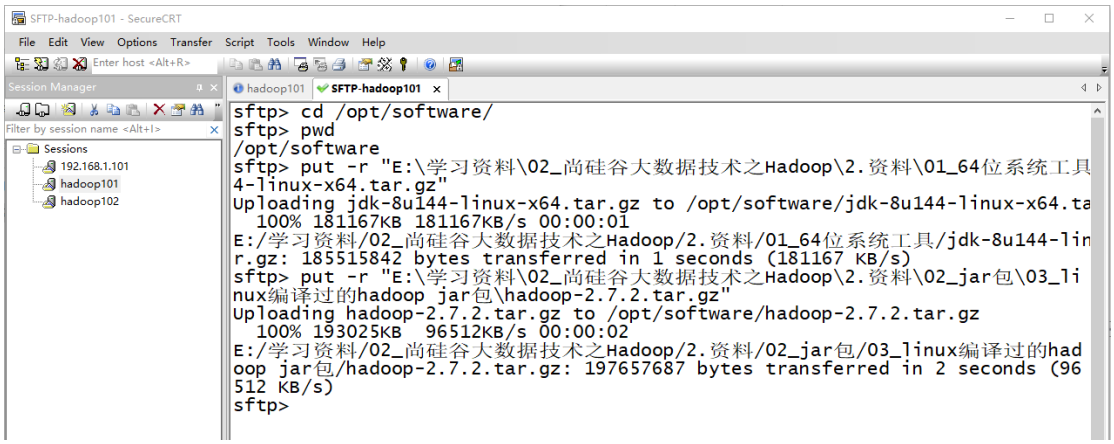


图 2-33 拖入 Hadoop 的 tar 包成功

2. 进入到 Hadoop 安装包路径下

```
[atguigu@hadoop101 ~]$ cd /opt/software/
```

3. 解压安装文件到/opt/module 下面

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

4. 查看是否解压成功

```
[atguigu@hadoop101 software]$ ls /opt/module/
hadoop-2.7.2
```

5. 将 Hadoop 添加到环境变量

（1）获取 Hadoop 安装路径

```
[atguigu@hadoop101 hadoop-2.7.2]$ pwd
/opt/module/hadoop-2.7.2
```

（2）打开/etc/profile 文件

```
[atguigu@hadoop101 hadoop-2.7.2]$ sudo vi /etc/profile
```

在 profile 文件末尾添加 JDK 路径：（shitf+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

```
[atguigu@ hadoop101 hadoop-2.7.2]$ source /etc/profile
```

6. 测试是否安装成功

```
[atguigu@hadoop101 hadoop-2.7.2]$ hadoop version
Hadoop 2.7.2
```

7. 重启(如果 Hadoop 命令不能用再重启)

```
[atguigu@ hadoop101 hadoop-2.7.2]$ sync
[atguigu@ hadoop101 hadoop-2.7.2]$ sudo reboot
```

3.4 Hadoop 目录结构

1、查看 Hadoop 目录结构

```
[atguigu@hadoop101 hadoop-2.7.2]$ ll
总用量 52
drwxr-xr-x. 2 atguigu atguigu  4096 5月  22 2017 bin
drwxr-xr-x. 3 atguigu atguigu  4096 5月  22 2017 etc
drwxr-xr-x. 2 atguigu atguigu  4096 5月  22 2017 include
drwxr-xr-x. 3 atguigu atguigu  4096 5月  22 2017 lib
drwxr-xr-x. 2 atguigu atguigu  4096 5月  22 2017 libexec
-rw-r--r--. 1 atguigu atguigu 15429 5月  22 2017 LICENSE.txt
-rw-r--r--. 1 atguigu atguigu   101 5月  22 2017 NOTICE.txt
-rw-r--r--. 1 atguigu atguigu  1366 5月  22 2017 README.txt
drwxr-xr-x. 2 atguigu atguigu  4096 5月  22 2017 sbin
drwxr-xr-x. 4 atguigu atguigu  4096 5月  22 2017 share
```

2、重要目录

（1）bin 目录：存放对 Hadoop 相关服务（HDFS,YARN）进行操作的脚本

（2）etc 目录：Hadoop 的配置文件目录，存放 Hadoop 的配置文件

（3）lib 目录：存放 Hadoop 的本地库（对数据进行压缩解压缩功能）

（4）sbin 目录：存放启动或停止 Hadoop 相关服务的脚本

（5）share 目录：存放 Hadoop 的依赖 jar 包、文档、和官方案例

第 4 章 Hadoop 运行模式

Hadoop 运行模式包括：本地模式、伪分布式模式以及完全分布式模式。

Hadoop 官方网站：http://hadoop.apache.org/

4.1 本地运行模式

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
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]$ vi 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 伪分布式运行模式

4.2.1 启动 HDFS 并运行 MapReduce 程序

1. 分析

（1）配置集群

（2）启动、测试集群增、删、查

（3）执行 WordCount 案例

2. 执行步骤

（1）配置集群

（a）配置：hadoop-env.sh　n

Linux 系统中获取 JDK 的安装路径：

```
[atguigu@ 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
```

注意：jps 是 JDK 中的命令，不是 Linux 命令。不安装 JDK 不能使用 jps

（b）web 端查看 HDFS 文件系统

http://hadoop101:50070/dfshealth.html#tab-overview

注意：如果不能查看，看如下帖子处理

http://www.cnblogs.com/zlslch/p/6604189.html

（c）查看产生的 Log 日志

说明：在企业中遇到 Bug 时，经常根据日志提示信息去分析问题、解决 Bug。

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
```

（d）思考：为什么不能一直格式化 NameNode，格式化 NameNode，要注意什么？

```
[atguigu@hadoop101 hadoop-2.7.2]$ cd data/tmp/dfs/name/current/
[atguigu@hadoop101 current]$ cat VERSION
clusterID=CID-f0330a58-36fa-4a2a-a65f-2688269b5837

[atguigu@hadoop101 hadoop-2.7.2]$ cd data/tmp/dfs/data/current/
clusterID=CID-f0330a58-36fa-4a2a-a65f-2688269b5837
```

注意：格式化 NameNode，会产生新的集群 id,导致 NameNode 和 DataNode 的集群 id 不一致，集群找不到已往数据。所以，格式 NameNode 时，一定要先删除 data 数据和 log 日志，然后再格式化 NameNode。

（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，如图 2-34 所示

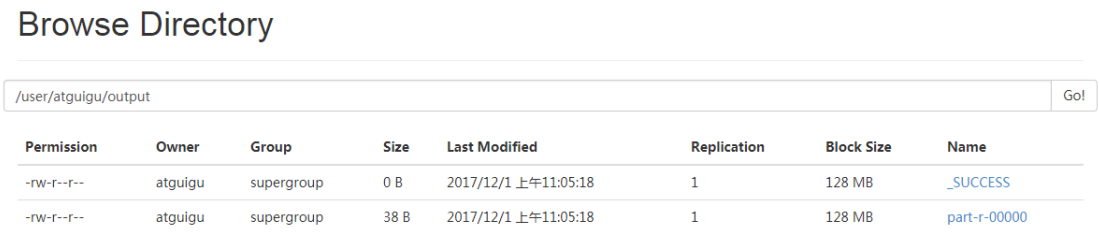


图 2-34 查看 output 文件

（f）将测试文件内容下载到本地

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

（g）删除输出结果

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

4.2.2 启动 YARN 并运行 MapReduce 程序

1. 分析

（1）配置集群在 YARN 上运行 MR

（2）启动、测试集群增、删、查

（3）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）启动前必须保证 NameNode 和 DataNode 已经启动

（b）启动 ResourceManager

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

（c）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，如图 2-35 所示

http://hadoop101:8088/cluster

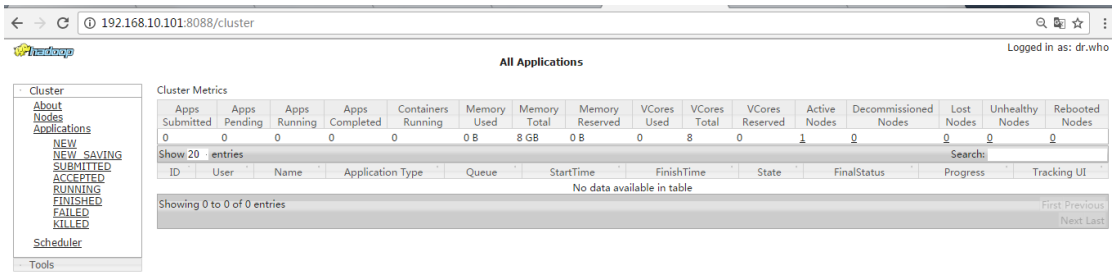


图 2-35 YARN 的浏览器页面

（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
```

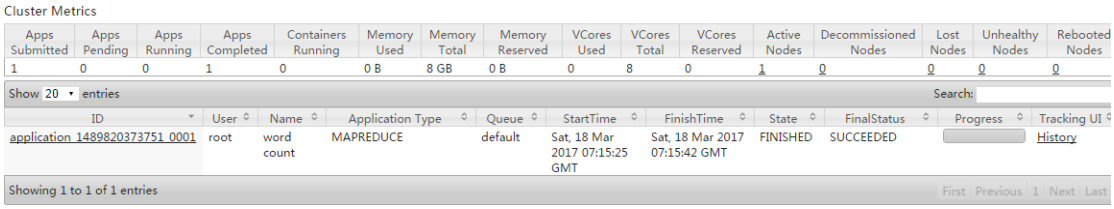
（c）执行 MapReduce 程序

```
[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，如图 2-36 所示

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

```
/user/atguigu/output/*
```



Cluster Metrics

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

Show 20 entries Search:

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		History

Showing 1 to 1 of 1 entries First Previous 1 Next Last

图 2-36 查看运行结果

4.2.3 配置历史服务器

为了查看程序的历史运行情况，需要配置一下历史服务器。具体配置步骤如下：

1. 配置 mapred-site.xml

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

在该文件里面增加如下配置。

```
<!-- 历史服务器端地址 -->
<property>
    <name>mapreduce.jobhistory.address</name>
    <value>hadoop101:10020</value>
</property>
<!-- 历史服务器 web 端地址 -->
<property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>hadoop101:19888</value>
</property>
```

2. 启动历史服务器

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

3. 查看历史服务器是否启动

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

4. 查看 JobHistory

http://hadoop101:19888/jobhistory

4.2.4 配置日志的聚集

日志聚集概念：应用运行完成以后，将程序运行日志信息上传到 HDFS 系统上。

日志聚集功能好处：可以方便的查看到程序运行详情，方便开发调试。

注意：开启日志聚集功能，需要重新启动 NodeManager 、ResourceManager 和 HistoryManager。

开启日志聚集功能具体步骤如下：

1. 配置 yarn-site.xml

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

在该文件里面增加如下配置。

```
<!-- 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

```
[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 上已经存在的输出文件

```
[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，如图 2-37，2-38，2-39 所示

http://hadoop101:19888/jobhistory

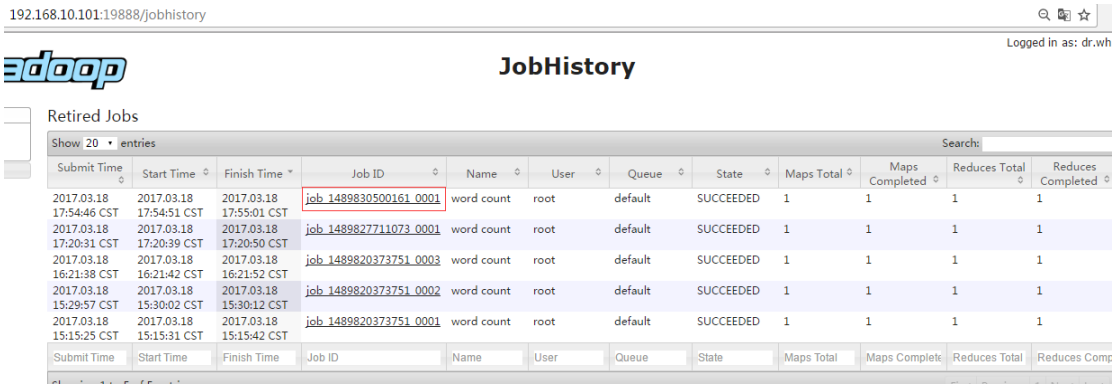


图 2-37 Job History

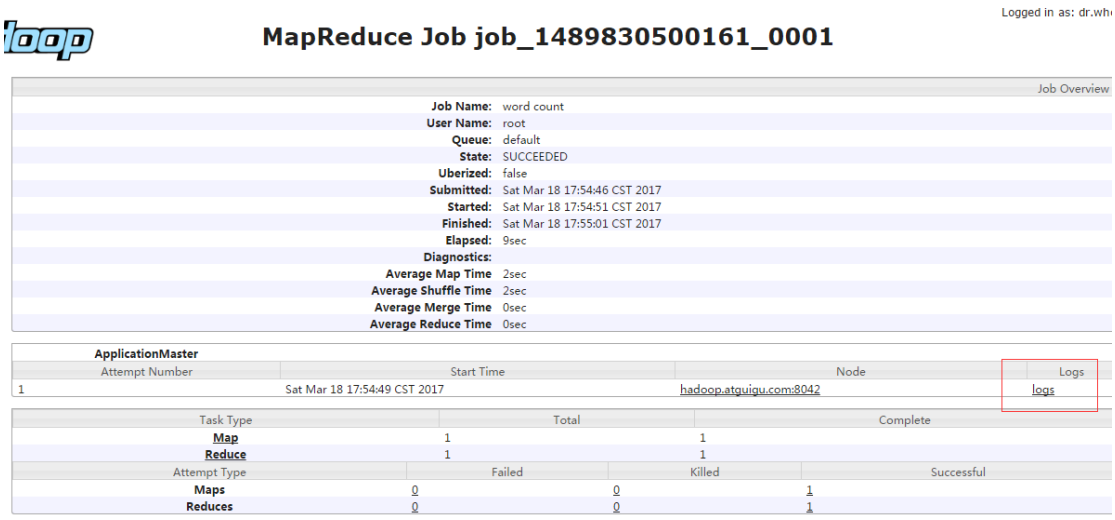


图 2-38 job 运行情况

图 2-39 查看日志

4.2.5 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：

表 2-1

要获取的默认文件	文件存放在 Hadoop 的 jar 包中的位置
[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
[mapred-default.xml]	hadoop-mapreduce-client-core-2.7.2.jar/ mapred-default.xml

（2）自定义配置文件：

core-site.xml、**hdfs-site.xml**、**yarn-site.xml**、**mapred-site.xml** 四个配置文件存放在 $HADOOP_HOME/etc/hadoop 这个路径上，用户可以根据项目需求重新进行修改配置。

4.3 完全分布式运行模式（开发重点）

分析：

1）准备 3 台客户机（关闭防火墙、静态 ip、主机名称）

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）单点启动

8）配置 ssh

9）群起并测试集群

4.3.1 虚拟机准备

详见 3.1 章。

4.3.2 编写集群分发脚本 xsync

1. scp（secure copy）安全拷贝

（1）scp 定义：

scp 可以实现服务器与服务器之间的数据拷贝。（from server1 to server2）

（2）基本语法

scp	-r	$pdir/$fname	$user@hadoop$host:$pdir/$fname
命令	递归	要拷贝的文件路径/名称	目的用户@主机:目的路径/名称

（3）案例实操

（a）在 hadoop101 上，将 hadoop101 中/opt/module 目录下的软件拷贝到 hadoop102 上。

```
[atguigu@hadoop101 /]$ scp -r /opt/module root@hadoop102:/opt/module
```

（b）在 hadoop103 上，将 hadoop101 服务器上的/opt/module 目录下的软件拷贝到 hadoop103 上。

```
[atguigu@hadoop103 opt]$sudo scp -r atguigu@hadoop101:/opt/module root@hadoop103:/opt/module
```

（c）在 hadoop103 上操作将 hadoop101 中/opt/module 目录下的软件拷贝到 hadoop104 上。

```
[atguigu@hadoop103 opt]$ scp -r atguigu@hadoop101:/opt/module root@hadoop104:/opt/module
```

注意：拷贝过来的/opt/module 目录，别忘了在 hadoop102、hadoop103、hadoop104 上修改所有文件的，所有者和所有者组。sudo chown atguigu:atguigu -R /opt/module

（d）将 hadoop101 中/etc/profile 文件拷贝到 hadoop102 的/etc/profile 上。

```
[atguigu@hadoop101 ~]$ sudo scp /etc/profile root@hadoop102:/etc/profile
```

（e）将 hadoop101 中/etc/profile 文件拷贝到 hadoop103 的/etc/profile 上。

```
[atguigu@hadoop101 ~]$ sudo scp /etc/profile root@hadoop103:/etc/profile
```

（f）将 hadoop101 中/etc/profile 文件拷贝到 hadoop104 的/etc/profile 上。

```
[atguigu@hadoop101 ~]$ sudo scp /etc/profile root@hadoop104:/etc/profile
```

注意：拷贝过来的配置文件别忘了 source 一下/etc/profile，。

2. rsync 远程同步工具

rsync 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）基本语法

rsync	-rvl	$pdir/$fname	$user@hadoop$host:$pdir/$fname
命令	选项参数	要拷贝的文件路径/名称	目的用户@主机:目的路径/名称

选项参数说明

表 2-2

选项	功能
-r	递归
-v	显示复制过程
-l	拷贝符号连接

（2）案例实操

（a）把 hadoop101 机器上的/opt/software 目录同步到 hadoop102 服务器的 root 用户下的/opt/目录

```
[atguigu@hadoop101 opt]$ rsync -rvl /opt/software/ root@hadoop102:/opt/software
```

3. xsync 集群分发脚本

（1）需求：循环复制文件到所有节点的相同目录下

（2）需求分析：

（a）rsync 命令原始拷贝：

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

（b）期望脚本：

xsync 要同步的文件名称

（c）说明：在/home/atguigu/bin 这个目录下存放的脚本，atguigu 用户可以在系统任何地方直接执行。

（3）脚本实现

（a）在/home/atguigu 目录下创建 bin 目录，并在 bin 目录下 xsync 创建文件，文件内容如下：

```
[atguigu@hadoop102 ~]$ mkdir bin
[atguigu@hadoop102 ~]$ cd bin/
[atguigu@hadoop102 bin]$ touch xsync
[atguigu@hadoop102 bin]$ vi xsync
```

在该文件中编写如下代码

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

（b）修改脚本 xsync 具有执行权限

```
[atguigu@hadoop102 bin]$ chmod 777 xsync
```

（c）调用脚本形式：xsync 文件名称

```
[atguigu@hadoop102 bin]$ xsync /home/atguigu/bin
```

注意：如果将 xsync 放到/home/atguigu/bin 目录下仍然不能实现全局使用，可以将 xsync 移动到/usr/local/bin 目录下。

4.3.3 集群配置

1. 集群部署规划

表 2-3

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

2. 配置集群

（1）核心配置文件

配置 core-site.xml

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

在该文件中编写如下配置

```
<!-- 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 配置文件

配置 hadoop-env.sh

```
[atguigu@hadoop102 hadoop]$ vi hadoop-env.sh
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

配置 hdfs-site.xml

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

在该文件中编写如下配置

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

<!-- 指定 Hadoop 辅助名称节点主机配置 -->
<property>
      <name>dfs.namenode.secondary.http-address</name>
      <value>hadoop104:50090</value>
</property>
```

（3）YARN 配置文件

配置 yarn-env.sh

```
[atguigu@hadoop102 hadoop]$ vi yarn-env.sh
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

配置 yarn-site.xml

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

在该文件中增加如下配置

```
<!-- 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>
```

（4）MapReduce 配置文件

配置 mapred-env.sh

```
[atguigu@hadoop102 hadoop]$ vi mapred-env.sh
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

配置 mapred-site.xml

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

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

在该文件中增加如下配置

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

3. 在集群上分发配置好的 Hadoop 配置文件

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

4. 查看文件分发情况

```
[atguigu@hadoop103 hadoop]$ cat /opt/module/hadoop-2.7.2/etc/hadoop/core-site.xml
```

4.3.4 集群单点启动

（1）如果集群是第一次启动，需要**格式化 NameNode**

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

（2）在 hadoop102 上启动 NameNode

```
[atguigu@hadoop102 hadoop-2.7.2]$ hadoop-daemon.sh start namenode
[atguigu@hadoop102 hadoop-2.7.2]$ jps
3461 NameNode
```

（3）在 hadoop102、hadoop103 以及 hadoop104 上分别启动 DataNode

```
[atguigu@hadoop102  hadoop-2.7.2]$  hadoop-daemon.sh  start
datanode
[atguigu@hadoop102 hadoop-2.7.2]$ jps
3461 NameNode
3608 Jps
3561 DataNode
[atguigu@hadoop103  hadoop-2.7.2]$  hadoop-daemon.sh  start
datanode
[atguigu@hadoop103 hadoop-2.7.2]$ jps
3190 DataNode
3279 Jps
[atguigu@hadoop104  hadoop-2.7.2]$  hadoop-daemon.sh  start
datanode
[atguigu@hadoop104 hadoop-2.7.2]$ jps
3237 Jps
3163 DataNode
```

（4）思考：每次都一个一个节点启动，如果节点数增加到 1000 个怎么办？

早上来了开始一个一个节点启动，到晚上下班刚好完成，下班？

4.3.5 SSH 无密登录配置

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

```
[atguigu@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）免密登录原理，如图 2-40 所示

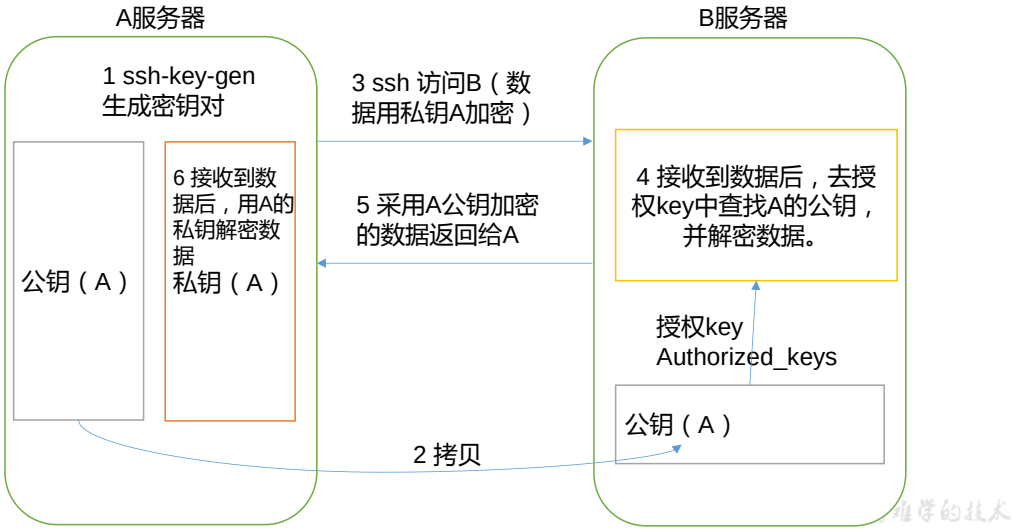


图 2-40　免密登陆原理

（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 hadoop102
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop103
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop104
```

注意：

还需要在 hadoop102 上采用 root 账号，配置一下无密登录到 hadoop102、hadoop103、hadoop104；

还需要在 hadoop103 上采用 atguigu 账号配置一下无密登录到 hadoop102、hadoop103、hadoop104 服务器上。

3. .ssh 文件夹下（~/.ssh）的文件功能解释

表 2-4

known_hosts	记录 ssh 访问过计算机的公钥(public key)
id_rsa	生成的私钥
id_rsa.pub	生成的公钥
authorized_keys	存放授权过得无密登录服务器公钥

4.3.6 群起集群

1. 配置 slaves

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

在该文件中增加如下内容：

```
hadoop102
hadoop103
hadoop104
```

注意：该文件中添加的内容结尾不允许有空格，文件中不允许有空行。

同步所有节点配置文件

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

2. 启动集群

（1）如果集群是第一次启动，需要格式化 NameNode（注意格式化之前，一定要先停止上次启动的所有 namenode 和 datanode 进程，然后再删除 data 和 log 数据）

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

（2）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
```

（3）启动 YARN

```
[atguigu@hadoop103 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。

（4）Web 端查看 SecondaryNameNode

（a）浏览器中输入：http://hadoop104:50090/status.html

（b）查看 SecondaryNameNode 信息，如图 2-41 所示。

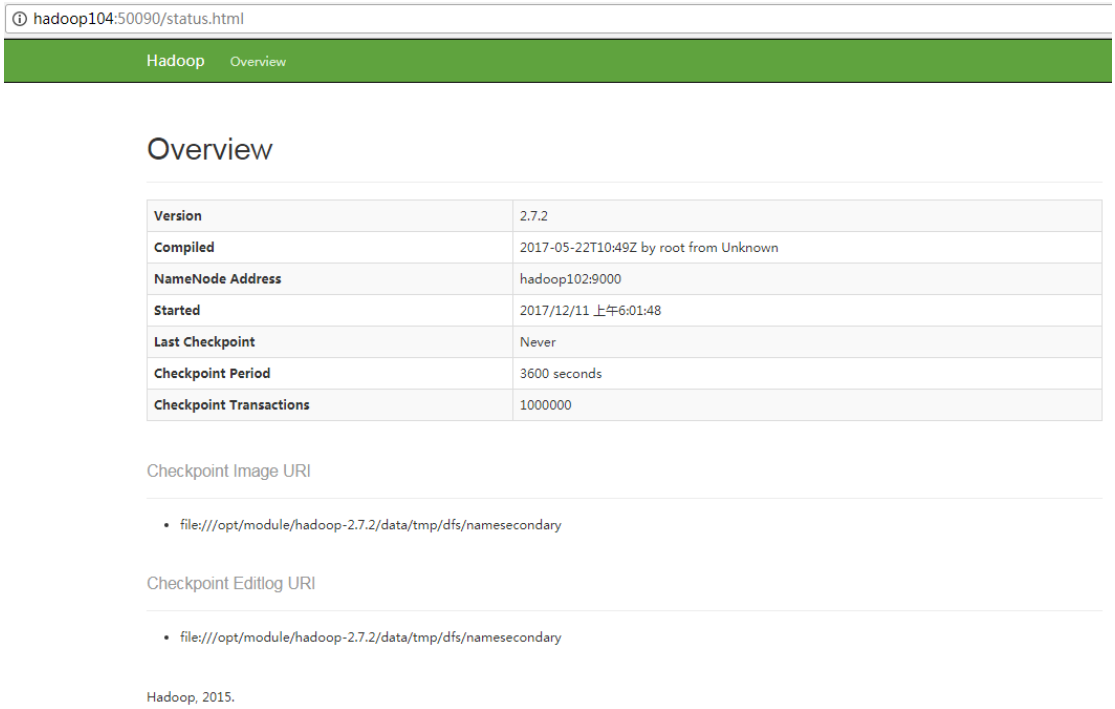


图 2-41 SecondaryNameNode 的 Web 端

3. 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]$ hdfs dfs -mkdir -p /user/atguigu/input
[atguigu@hadoop102 hadoop-2.7.2]$ hdfs dfs -put wcinput/wc.input /user/atguigu/input
```

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

（a）查看 HDFS 文件存储路径

```
[atguigu@hadoop102 subdir0]$ pwd
```

/opt/module/hadoop-2.7.2/data/tmp/dfs/data/current/BP-938951106-192.168.10.107-1495462844069/current/finalized/subdir0/subdir0

（b）查看 HDFS 在磁盘存储文件内容

```
[atguigu@hadoop102 subdir0]$ cat blk_1073741825
hadoop yarn
hadoop mapreduce
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 ./
```

4.3.7 集群启动/停止方式总结

1. 各个服务组件逐一启动/停止

（1）分别启动/停止 HDFS 组件

hadoop-daemon.sh　start / stop　namenode / datanode / secondarynamenode

（2）启动/停止 YARN

yarn-daemon.sh　start / stop　resourcemanager / nodemanager

2. 各个模块分开启动/停止（配置 ssh 是前提）常用

（1）整体启动/停止 HDFS

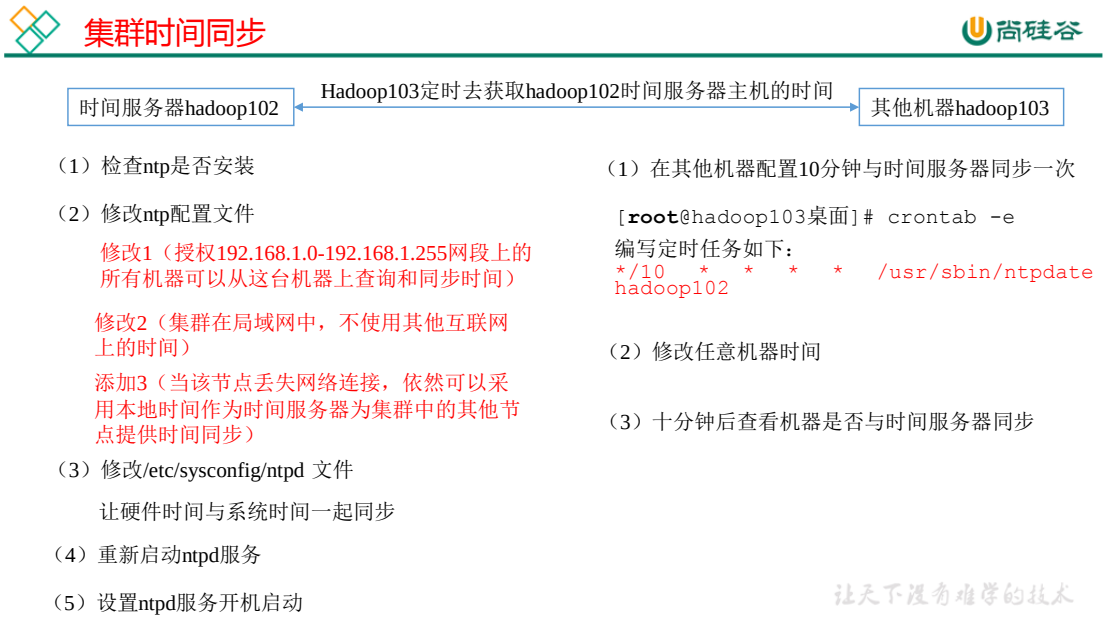
start-dfs.sh　/　stop-dfs.sh

（2）整体启动/停止 YARN

start-yarn.sh　/　stop-yarn.sh

4.3.8 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。



配置时间同步具体实操：

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（授权 192.168.1.0-192.168.1.255 网段上的所有机器可以从这台机器上查询和同步时间）

#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（集群在局域网中，不使用其他互联网上的时间）

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（当该节点丢失网络连接，依然可以采用本地时间作为时间服务器为集群中的其他节点提供时间同步）

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（让硬件时间与系统时间一起同步）

```
SYNC_HWCLOCK=yes
```

（4）重新启动 ntpd 服务

```
[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）设置 ntpd 服务开机启动

```
[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 桌面]# crontab -e
```

编写定时任务如下：

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

（2）修改任意机器时间

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

说明：测试的时候可以将 10 分钟调整为 1 分钟，节省时间。

第 5 章 Hadoop 编译源码（面试重点）

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、JDK8、maven、ant 、protobuf)

（1）hadoop-2.7.2-src.tar.gz
（2）jdk-8u144-linux-x64.tar.gz
（3）apache-ant-1.9.9-bin.tar.gz（build 工具，打包用的）
（4）apache-maven-3.0.5-bin.tar.gz
（5）protobuf-2.5.0.tar.gz（序列化的框架）

5.2 jar 包安装

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-8u144-linux-x64.tar.gz -C /opt/module/

[root@hadoop101 software]# vi /etc/profile
#JAVA_HOME:
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 conf/settings.xml

<mirrors>
   <!-- mirror
    | Specifies a repository mirror site to use instead of a given repository. The repository that
    | this mirror serves has an ID that matches the mirrorOf element of this mirror. IDs are used
    | for inheritance and direct lookup purposes, and must be unique across the set of mirrors.
    |
<mirror>
      <id>mirrorId</id>
      <mirrorOf>repositoryId</mirrorOf>
      <name>Human Readable Name for this Mirror.</name>
      <url>http://my.repository.com/repo/path</url>
     </mirror>
    -->
      <mirror>
             <id>nexus-aliyun</id>
             <mirrorOf>central</mirrorOf>
             <name>Nexus aliyun</name>

<url>http://maven.aliyun.com/nexus/content/groups/public</url>
      </mirror>
</mirrors>
```

```
[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
[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/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，如图 2-42 所示。

```
[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]
```

图 2-42 编译源码

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. 编译源码过程中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,nativeN -DskipTests -Dtar
```

（3）报 ant、protobuf 等错误，插件下载未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

第 6 章 常见错误及解决方案

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）不识别主机名称

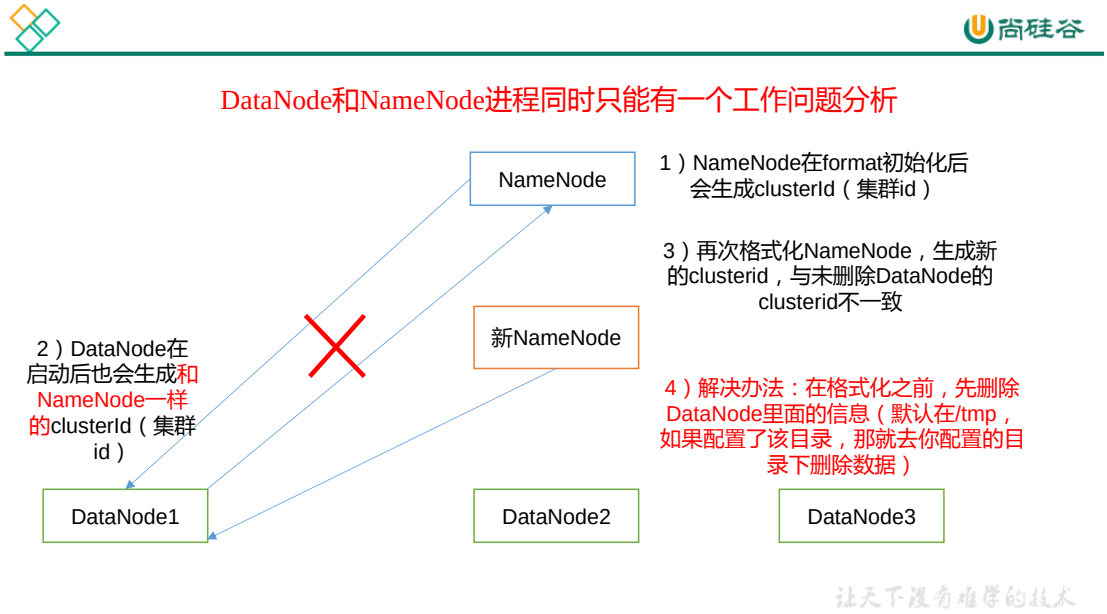
```
java.net.UnknownHostException: hadoop102: hadoop102
        at
java.net.InetAddress.getLocalHost(InetAddress.java:1475)
        at
org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(J
obSubmitter.java:146)
        at
org.apache.hadoop.mapreduce.Job$10.run(Job.java:1290)
        at
org.apache.hadoop.mapreduce.Job$10.run(Job.java:1287)
        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

9）DataNode 和 NameNode 进程同时只能工作一个。



10）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

解决办法：尽量不要粘贴 word 中代码。

11）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。

12）jps 不生效。

原因：全局变量 hadoop java 没有生效。解决办法：需要 source /etc/profile 文件。

13）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