**大数据离线阶段**

**02**

# 课程计划

目录

[一、 课程计划 2](#_Toc77620111)

[二、 Apache Hadoop 4](#_Toc77620112)

[1． Hadoop介绍 4](#_Toc77620113)

[2． Hadoop发展简史 6](#_Toc77620114)

[3． Hadoop特性优点 6](#_Toc77620115)

[4． Hadoop国内外应用 7](#_Toc77620116)

[三、 Hadoop集群搭建 8](#_Toc77620117)

[1． 发行版本 8](#_Toc77620118)

[2． 集群简介 10](#_Toc77620119)

[3． 服务器基础环境准备 11](#_Toc77620120)

[4． JDK环境安装 12](#_Toc77620121)

[5． Hadoop重新编译 13](#_Toc77620122)

[6． Hadoop安装包目录结构 13](#_Toc77620123)

[7． Hadoop配置文件修改 14](#_Toc77620124)

[7.1． hadoop-env.sh 14](#_Toc77620125)

[7.2． core-site.xml 14](#_Toc77620126)

[7.3． hdfs-site.xml 15](#_Toc77620127)

[7.4． mapred-site.xml 16](#_Toc77620128)

[7.5． yarn-site.xml 17](#_Toc77620129)

[7.6． workers 18](#_Toc77620130)

[8． scp同步安装包 19](#_Toc77620131)

[9． Hadoop环境变量 19](#_Toc77620132)

[四、 Hadoop集群启动、初体验 20](#_Toc77620133)

[1． 启动方式 20](#_Toc77620134)

[1.1． 单节点逐个启动 20](#_Toc77620135)

[1.2． 脚本一键启动 21](#_Toc77620136)

[2． 集群web-ui 21](#_Toc77620137)

[3． Hadoop初体验 22](#_Toc77620138)

[3.1． HDFS使用 22](#_Toc77620139)

[3.2． 运行mapreduce程序 22](#_Toc77620140)

[五、 MapReduce jobHistory 23](#_Toc77620141)

[1． 修改mapred-site.xml 23](#_Toc77620142)

[2． 分发配置到其他机器 24](#_Toc77620143)

[3． 启动jobHistoryServer服务进程 24](#_Toc77620144)

[4． 页面访问jobhistoryserver 24](#_Toc77620145)

[六、 HDFS的垃圾桶机制 25](#_Toc77620146)

[1． 垃圾桶机制解析 25](#_Toc77620147)

[2． 垃圾桶机制配置 25](#_Toc77620148)

[3． 垃圾桶机制验证 25](#_Toc77620149)

# Apache Hadoop

## Hadoop介绍

Hadoop是Apache旗下的一个用java语言实现开源软件框架，是一个开发和运行处理大规模数据的软件平台。允许使用简单的编程模型在大量计算机集群上对大型数据集进行分布式处理。

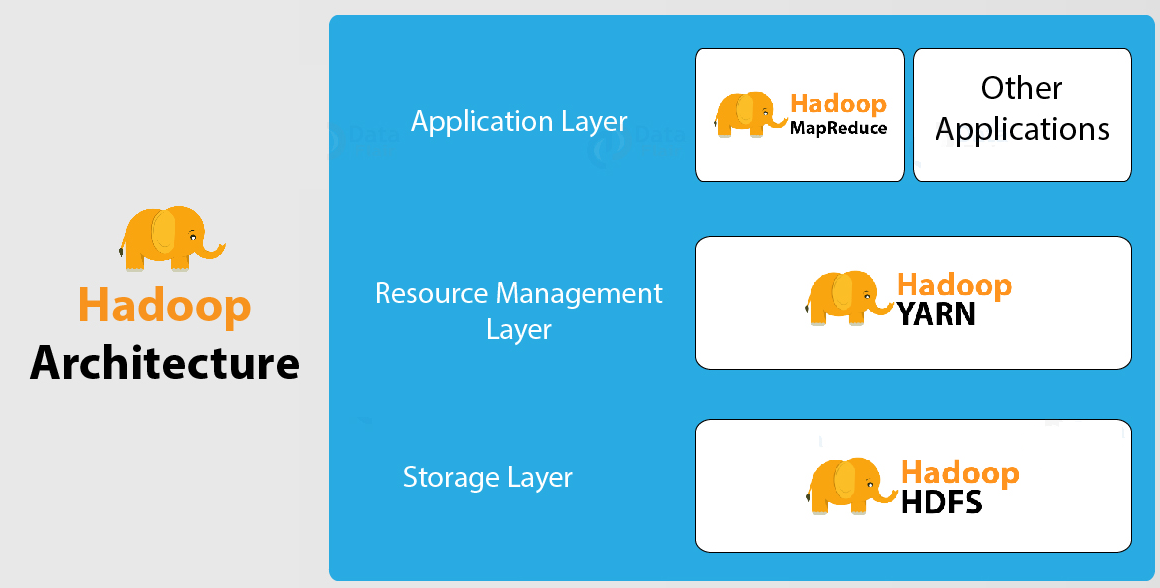


狭义上说，Hadoop指Apache这款开源框架，它的核心组件有：

HDFS（分布式文件系统）：解决海量数据存储

YARN（作业调度和集群资源管理的框架）：解决资源任务调度

MAPREDUCE（分布式运算编程框架）：解决海量数据计算



广义上来说，Hadoop通常是指一个更广泛的概念——Hadoop生态圈。



当下的Hadoop已经成长为一个庞大的体系，随着生态系统的成长，新出现的项目越来越多，其中不乏一些非Apache主管的项目，这些项目对HADOOP是很好的补充或者更高层的抽象。

## Hadoop发展简史

Hadoop是Apache Lucene创始人 Doug Cutting 创建的。最早起源于Nutch，它是Lucene的子项目。Nutch的设计目标是构建一个大型的全网搜索引擎，包括网页抓取、索引、查询等功能，但随着抓取网页数量的增加，遇到了严重的可扩展性问题：如何解决数十亿网页的存储和索引问题。

2003年Google发表了一篇论文为该问题提供了可行的解决方案。论文中描述的是谷歌的产品架构，该架构称为：谷歌分布式文件系统（GFS）,可以解决他们在网页爬取和索引过程中产生的超大文件的存储需求。

2004年 Google发表论文向全世界介绍了谷歌版的MapReduce系统。

同时期，Nutch的开发人员完成了相应的开源实现HDFS和MAPREDUCE，并从Nutch中剥离成为独立项目HADOOP，到2008年1月，HADOOP成为Apache顶级项目，迎来了它的快速发展期。

2006年Google发表了论文是关于BigTable的，这促使了后来的Hbase的发展。

因此，Hadoop及其生态圈的发展离不开Google的贡献。

## Hadoop特性优点

扩容能力（Scalable）：Hadoop是在可用的计算机集群间分配数据并完成计算任务的，这些集群可用方便的扩展到数以千计的节点中。

成本低（Economical）：Hadoop通过普通廉价的机器组成服务器集群来分发以及处理数据，以至于成本很低。

高效率（Efficient）：通过并发数据，Hadoop可以在节点之间动态并行的移动数据，使得速度非常快。

可靠性（Rellable）：能自动维护数据的多份复制，并且在任务失败后能自动地重新部署（redeploy）计算任务。所以Hadoop的按位存储和处理数据的能力值得人们信赖。

## Hadoop国内外应用

不管是国内还是国外，Hadoop最受青睐的行业是互联网领域，可以说互联网公司是hadoop的主要使用力量。

国外来说，Yahoo、Facebook、IBM等公司都大量使用hadoop集群来支撑业务。比如：

Yahoo的Hadoop应用在支持广告系统、用户行为分析、支持Web搜索等。

Facebook主要使用Hadoop存储内部日志与多维数据，并以此作为报告、分析和机器学习的数据源。

国内来说，BAT领头的互联网公司是当仁不让的Hadoop使用者、维护者。比如Ali云梯（14年国内最大Hadoop集群）、百度的日志分析平台、推荐引擎系统等。



国内其他非互联网领域也有不少hadoop的应用，比如：

金融行业： 个人征信分析

证券行业： 投资模型分析

交通行业： 车辆、路况监控分析

电信行业： 用户上网行为分析

总之：hadoop并不会跟某种具体的行业或者某个具体的业务挂钩，它只是一种用来做海量数据分析处理的工具。

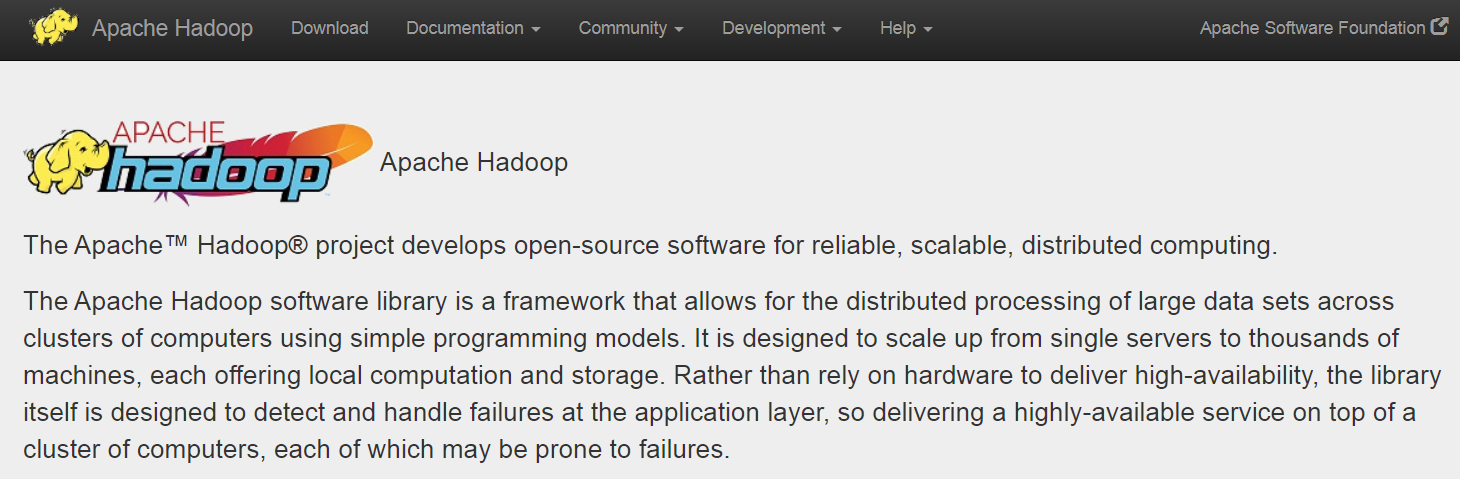
# Hadoop集群搭建

## 发行版本

Hadoop发行版本分为开源**社区版**和**商业版**。

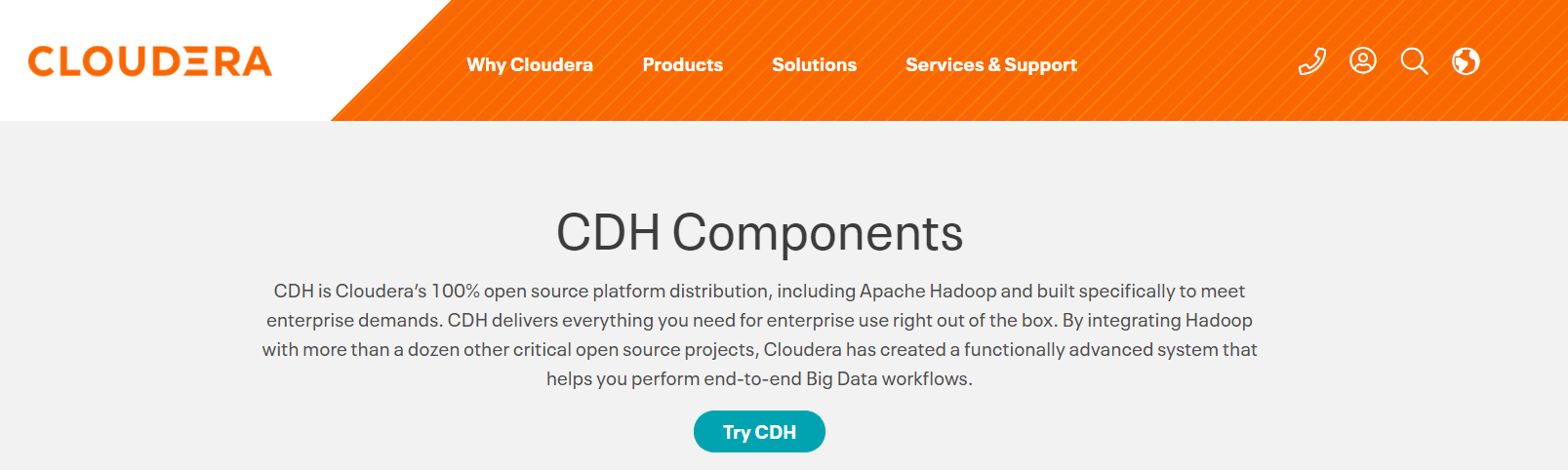
社区版是指由Apache软件基金会维护的版本，是官方维护的版本体系。

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



商业版Hadoop是指由第三方商业公司在社区版Hadoop基础上进行了一些修改、整合以及各个服务组件兼容性测试而发行的版本，比较著名的有**cloudera的CDH**、mapR、hortonWorks等。

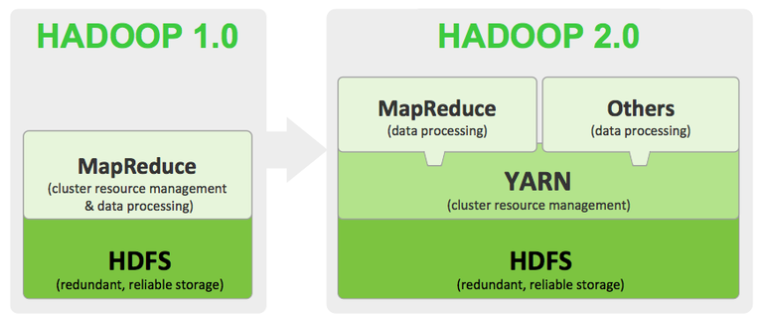
<https://www.cloudera.com/products/open-source/apache-hadoop/key-cdh-components.html>

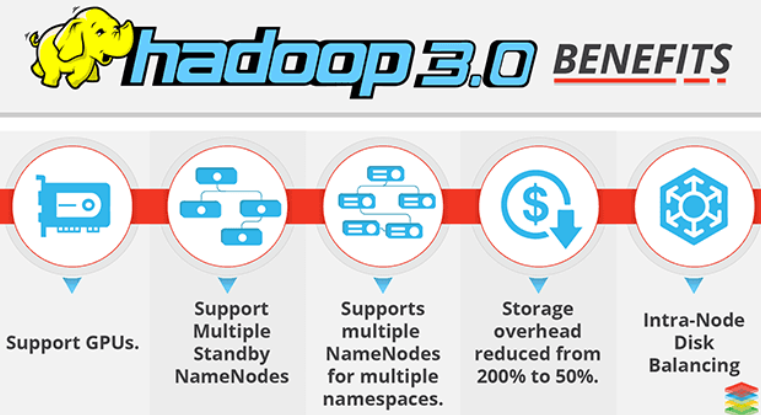


Hadoop的版本很特殊，是由多条分支并行的发展着。大的来看分为3个大的系列版本：1.x、2.x、3.x。

Hadoop1.0由一个分布式文件系统HDFS和一个离线计算框架MapReduce组成。架构落后，已经淘汰。

Hadoop 2.0则包含一个分布式文件系统HDFS，一个资源管理系统YARN和一个离线计算框架MapReduce。相比于Hadoop1.0，Hadoop 2.0功能更加强大，且具有更好的扩展性、性能，并支持多种计算框架。

  
 Hadoop 3.0相比之前的Hadoop 2.0有一系列的功能增强。目前已经趋于稳定，可能生态圈的某些组件还没有升级、整合完善。



我们课程中使用的是：Apache Hadoop 3.3.0。

## 集群简介

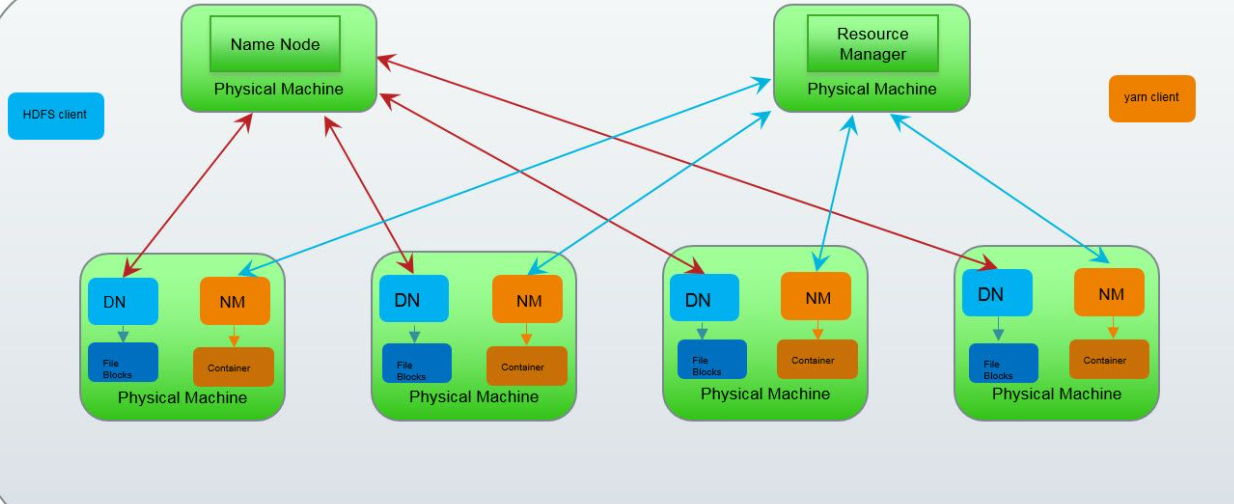
HADOOP集群具体来说包含两个集群：HDFS集群和YARN集群，两者逻辑上分离，但物理上常在一起。

HDFS集群负责海量数据的存储，集群中的角色主要有：

NameNode、DataNode、SecondaryNameNode

YARN集群负责海量数据运算时的资源调度，集群中的角色主要有：

ResourceManager、NodeManager



那mapreduce是什么呢？它其实是一个分布式运算编程框架，是应用程序开发包，由用户按照编程规范进行程序开发，后打包运行在HDFS集群上，并且受到YARN集群的资源调度管理。

Hadoop部署方式分三种，Standalone mode（独立模式）、Pseudo-Distributed mode（伪分布式模式）、Cluster mode（群集模式），其中前两种都是在单机部署。

独立模式又称为单机模式，仅1个机器运行1个java进程，主要用于调试。

伪分布模式也是在1个机器上运行HDFS的NameNode和DataNode、YARN的 ResourceManger和NodeManager，但分别启动单独的java进程，主要用于调试。

集群模式主要用于生产环境部署。会使用N台主机组成一个Hadoop集群。这种部署模式下，主节点和从节点会分开部署在不同的机器上。

我们以3节点为例进行搭建，角色分配如下：

node1 NameNode DataNode ResourceManager

node2 DataNode NodeManager SecondaryNameNode

node3 DataNode NodeManager

## 服务器基础环境准备

|  |
| --- |
| 1.0 配置好各虚拟机的网络（采用NAT联网模式）    1.1修改各个虚拟机主机名  vi /etc/hostname    node1.itcast.cn    1.2修改主机名和IP的映射关系  vi /etc/hosts    192.168.227.151 node1.itcast.cn node1  192.168.227.152 node2.itcast.cn node2  192.168.227.153 node3.itcast.cn node3    1.3关闭防火墙  #查看防火墙状态  systemctl status firewalld.service  #关闭防火墙  systemctl stop firewalld.service  #关闭防火墙开机启动  systemctl disable firewalld.service  1.4.配置ssh免登陆(配置node1-->node1,node2,node3)  #node1生成ssh免登陆密钥  ssh-keygen -t rsa （一直回车）  执行完这个命令后，会生成两个文件id\_rsa（私钥）、id\_rsa.pub（公钥）  将公钥拷贝到要免密登陆的目标机器上  ssh-copy-id node1  ssh-copy-id node2  ssh-copy-id node3    1.5 同步集群时间  yum install ntpdate  ntpdate cn.pool.ntp.org |

## JDK环境安装

|  |
| --- |
| 2.1上传jdk  jdk-8u65-linux-x64.tar.gz    2.2解压jdk  tar -zxvf jdk-8u65-linux-x64.tar.gz -C /export/server    2.3将java添加到环境变量中  vim /etc/profile  #在文件最后添加  export JAVA\_HOME=/export/server/jdk1.8.0\_65  export PATH=$PATH:$JAVA\_HOME/bin  export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  #刷新配置  source /etc/profile |

## Hadoop重新编译

Hadoop官方一般都给出了对应版本安装包，一般情况下是不需要自己进行编译的，但是由于官方编译好的hadoop的安装包没有提供带C程序访问的接口，所以在使用本地库（**本地库可以用来做压缩，以及支持C程序等等**）的时候就会出问题，因此生产环境中，一般会重新编译。

此外，作为开源软件，针对源码进行修改某些属性，之后也需要重编译。

可以使用课程提供编译好的安装包。

## Hadoop安装包目录结构

解压hadoop-3.3.0-Centos7-64-with-snappy.tar.gz，目录结构如下：

**bin**：Hadoop最基本的管理脚本和使用脚本的目录，这些脚本是sbin目录下管理脚本的基础实现，用户可以直接使用这些脚本管理和使用Hadoop。

**etc**：Hadoop配置文件所在的目录，包括core-site,xml、hdfs-site.xml、mapred-site.xml等从Hadoop1.0继承而来的配置文件和yarn-site.xml等Hadoop2.0新增的配置文件。

**include**：对外提供的编程库头文件（具体动态库和静态库在lib目录中），这些头文件均是用C++定义的，通常用于C++程序访问HDFS或者编写MapReduce程序。

**lib**：该目录包含了Hadoop对外提供的编程动态库和静态库，与include目录中的头文件结合使用。

**libexec**：各个服务对用的shell配置文件所在的目录，可用于配置日志输出、启动参数（比如JVM参数）等基本信息。

**sbin**：Hadoop管理脚本所在的目录，主要包含HDFS和YARN中各类服务的启动/关闭脚本。

**share**：Hadoop各个模块编译后的jar包所在的目录，官方自带示例。

## Hadoop配置文件修改

Hadoop安装主要就是配置文件的修改，一般在主节点进行修改，完毕后scp下发给其他各个从节点机器。

### hadoop-env.sh

文件中设置的是Hadoop运行时需要的环境变量。JAVA\_HOME是必须设置的，即使我们当前的系统中设置了JAVA\_HOME，它也是不认识的，因为Hadoop即使是在本机上执行，它也是把当前的执行环境当成远程服务器。

|  |
| --- |
| export JAVA\_HOME=/export/server/jdk1.8.0\_241  #文件最后添加  export HDFS\_NAMENODE\_USER=root  export HDFS\_DATANODE\_USER=root  export HDFS\_SECONDARYNAMENODE\_USER=root  export YARN\_RESOURCEMANAGER\_USER=root  export YARN\_NODEMANAGER\_USER=root |

### core-site.xml

hadoop的核心配置文件，有默认的配置项core-default.xml。

core-default.xml与core-site.xml的功能是一样的，如果在core-site.xml里没有配置的属性，则会自动会获取core-default.xml里的相同属性的值。

|  |
| --- |
| <property>  <name>fs.defaultFS</name>  <value>hdfs://node1:8020</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>/export/data/hadoop-3.3.0</value>  </property>  <!-- 设置HDFS web UI用户身份 -->  <property>  <name>hadoop.http.staticuser.user</name>  <value>root</value>  </property>  <!-- 整合hive -->  <property>  <name>hadoop.proxyuser.root.hosts</name>  <value>\*</value>  </property>  <property>  <name>hadoop.proxyuser.root.groups</name>  <value>\*</value>  </property> |

### hdfs-site.xml

HDFS的核心配置文件，有默认的配置项hdfs-default.xml。

hdfs-default.xml与hdfs-site.xml的功能是一样的，如果在hdfs-site.xml里没有配置的属性，则会自动会获取hdfs-default.xml里的相同属性的值。

|  |
| --- |
| <!-- 指定secondarynamenode运行位置 -->  <property>  <name>dfs.namenode.secondary.http-address</name>  <value>node2:50090</value>  </property> |

### mapred-site.xml

MapReduce的核心配置文件，有默认的配置项mapred-default.xml。

mapred-default.xml与mapred-site.xml的功能是一样的，如果在mapred-site.xml里没有配置的属性，则会自动会获取mapred-default.xml里的相同属性的值。

|  |
| --- |
| <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  <property>  <name>yarn.app.mapreduce.am.env</name>  <value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>  </property>  <property>  <name>mapreduce.map.env</name>  <value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>  </property>  <property>  <name>mapreduce.reduce.env</name>  <value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>  </property> |

### yarn-site.xml

YARN的核心配置文件，有默认的配置项yarn-default.xml。

yarn-default.xml与yarn-site.xml的功能是一样的，如果在yarn-site.xml里没有配置的属性，则会自动会获取yarn-default.xml里的相同属性的值。

|  |
| --- |
| <!-- 指定YARN的主角色（ResourceManager）的地址 -->  <property>  <name>yarn.resourcemanager.hostname</name>  <value>node1</value>  </property>    <!-- NodeManager上运行的附属服务。需配置成mapreduce\_shuffle，才可运行MapReduce程序默认值："" -->  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <!-- 是否将对容器实施物理内存限制 -->  <property>  <name>yarn.nodemanager.pmem-check-enabled</name>  <value>false</value>  </property>  <!-- 是否将对容器实施虚拟内存限制。 -->  <property>  <name>yarn.nodemanager.vmem-check-enabled</name>  <value>false</value>  </property>  <!-- 开启日志聚集 -->  <property>  <name>yarn.log-aggregation-enable</name>  <value>true</value>  </property>  <!-- 设置yarn历史服务器地址 -->  <property>  <name>yarn.log.server.url</name>  <value>http://node1:19888/jobhistory/logs</value>  </property>  <!-- 保存的时间7天 -->  <property>  <name>yarn.log-aggregation.retain-seconds</name>  <value>604800</value>  </property> |

### workers

workers文件里面记录的是集群主机名。主要作用是配合一键启动脚本如start-dfs.sh、stop-yarn.sh用来进行集群启动。这时候workers文件里面的主机标记的就是从节点角色所在的机器。

|  |
| --- |
| vi workers  node1.itcast.cn  node2.itcast.cn  node3.itcast.cn |

## scp同步安装包

|  |
| --- |
| cd /export/server  scp -r hadoop-3.3.0 root@node2:$PWD  scp -r hadoop-3.3.0 root@node3:$PWD |

在node1上进行了配置文件的修改，使用scp命令将修改好之后的安装包同步给集群中的其他节点。

## Hadoop环境变量

3台机器都需要配置环境变量文件。

|  |
| --- |
| vim /etc/proflie  export HADOOP\_HOME=/export/server/hadoop-3.3.0  export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin  source /etc/profile |

# Hadoop集群启动、初体验

## 启动方式

要启动Hadoop集群，需要启动HDFS和YARN两个集群。

注意：**首次启动HDFS时，必须对其进行格式化操作**。本质上是一些清理和准备工作，因为此时的HDFS在物理上还是不存在的。

hadoop namenode -format

### 单节点逐个启动

在主节点上使用以下命令启动HDFS NameNode：

$HADOOP\_HOME/bin/hdfs --daemon start namenode

在每个从节点上使用以下命令启动HDFS DataNode：

$HADOOP\_HOME/bin/hdfs --daemon start datanode

在node2上使用以下命令启动HDFS SecondaryNameNode：

$HADOOP\_HOME/bin/hdfs --daemon start secondarynamenode

在主节点上使用以下命令启动YARN ResourceManager：

$HADOOP\_HOME/bin/yarn --daemon start resourcemanager

在每个从节点上使用以下命令启动YARN nodemanager：

$HADOOP\_HOME/bin/yarn --daemon start nodemanager

如果想要停止某个节点上某个角色，只需要把命令中的**start**改为**stop**即可。

### 脚本一键启动

如果配置了etc/hadoop/workers和ssh免密登录，则可以使用程序脚本启动所有Hadoop两个集群的相关进程，在主节点所设定的机器上执行。

hdfs：$HADOOP\_PREFIX/sbin/start-dfs.sh

yarn: $HADOOP\_PREFIX/sbin/start-yarn.sh

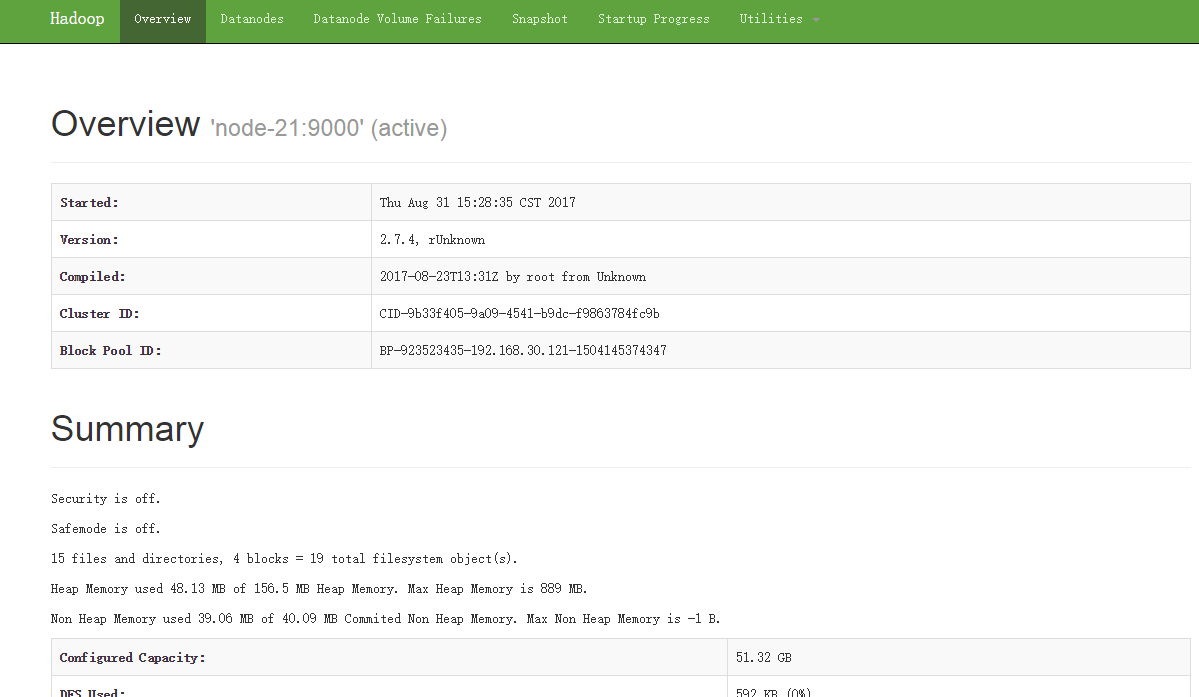
停止集群：stop-dfs.sh、stop-yarn.sh

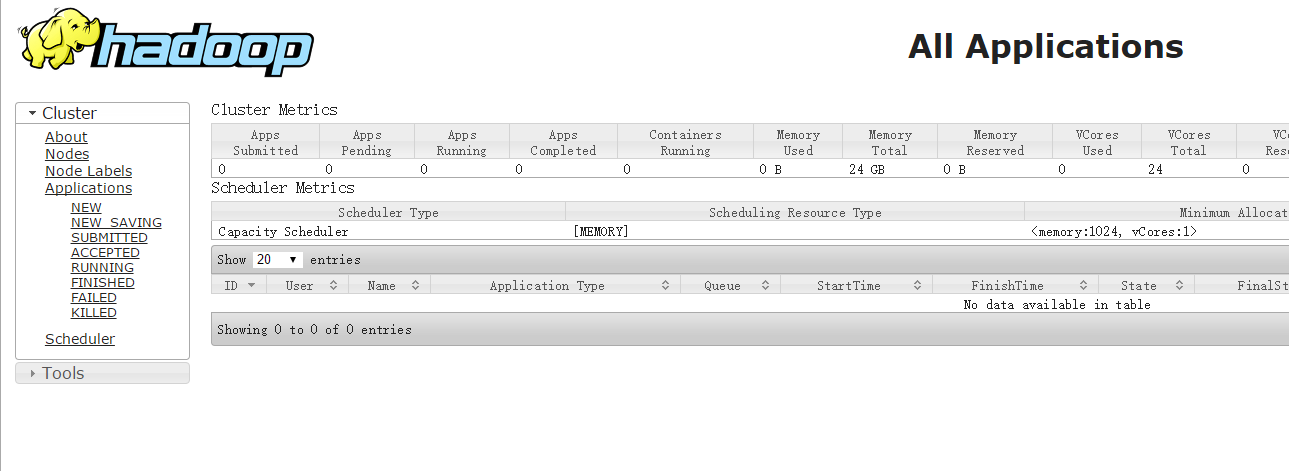
## 集群web-ui

一旦Hadoop集群启动并运行，可以通过web-ui进行集群查看，如下所述：

NameNode http://nn\_host:port/ 默认9870.

ResourceManager http://rm\_host:port/ 默认 8088.





## Hadoop初体验

### HDFS使用

从Linux本地上传一个文本文件到hdfs的/test/input目录下

hadoop fs -mkdir -p /wordcount/input

hadoop fs -put /root/somewords.txt /test/input

### 运行mapreduce程序

在Hadoop安装包的share/hadoop/mapreduce下有官方自带的mapreduce程序。我们可以使用如下的命令进行运行测试。

示例程序jar:

hadoop-mapreduce-examples-3.3.0.jar

计算圆周率:

hadoop jar hadoop-mapreduce-examples-3.3.0.jar pi 20 50

关于圆周率的估算，感兴趣的可以查询资料Monte Carlo方法来计算Pi值。

# MapReduce jobHistory

JobHistory用来记录已经finished的mapreduce运行日志，日志信息存放于HDFS目录中，默认情况下没有开启此功能，需要在mapred-site.xml中配置并手动启动。

## 修改mapred-site.xml

cd /export/servers/hadoop-3.3.0/etc/hadoop

vim mapred-site.xml

MR JobHistory Server管理的日志的存放位置

<property>

<name>mapreduce.jobhistory.address</name>

<value>node1:10020</value>

</property>

查看历史服务器已经运行完的Mapreduce作业记录的web地址，需要启动该服务才行

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>node1:19888</value>

</property>

## 分发配置到其他机器

cd /export/servers/hadoop-3.3.0/etc/hadoop

scp -r mapred-site.xml node2:$PWD

scp –r mapred-site.xml node3:$PWD

## 启动jobHistoryServer服务进程

mapred --daemon start historyserver

如果关闭的话 用下述命令

mapred --daemon stop historyserver

## 页面访问jobhistoryserver

<http://node1:19888/jobhistory>

# HDFS的垃圾桶机制

## 垃圾桶机制解析

每一个文件系统都会有垃圾桶机制，便于将删除的数据回收到垃圾桶里面去，避免某些误操作删除一些重要文件。回收到垃圾桶里里面的资料数据，都可以进行恢复。

## 垃圾桶机制配置

HDFS的垃圾回收的默认配置属性为 0，也就是说，如果你不小心误删除了某样东西，那么这个操作是不可恢复的。

修改core-site.xml ：

那么可以按照生产上的需求设置回收站的保存时间，这个时间以分钟为单位，例如1440=24h=1天。

<property>

<name>fs.trash.interval</name>

<value>1440</value>

</property>

然后重启hdfs集群

## 垃圾桶机制验证

如果启用垃圾箱配置，dfs命令删除的文件不会立即从HDFS中删除。相反，HDFS将其移动到垃圾目录（每个用户在/user/<username>/.Trash下都有自己的垃圾目录）。只要文件保留在垃圾箱中，文件可以快速恢复。

使用skipTrash选项删除文件，该选项不会将文件发送到垃圾箱。它将从HDFS中完全删除。