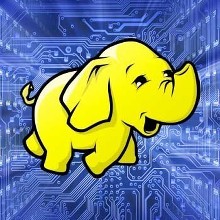
大数据技术之Hadoop（入门）

**一 大数据概论**

预科内容

**二 从Hadoop框架讨论大数据生态**

* 名字起源
* 该项目的创建者，Doug Cutting解释Hadoop的得名 ：“这个名字是我孩子给一个棕黄色的大象玩具命名的
* ​​
* 项目起源
* Hadoop由 Apache Software Foundation 公司于 2005 年秋天作为[Lucene](https://baike.baidu.com/item/Hadoop/_blank)的子项目[Nutch](https://baike.baidu.com/item/Hadoop/_blank)的一部分正式引入。它受到最先由 Google Lab 开发的 Map/Reduce 和 Google File System([GFS](https://baike.baidu.com/item/Hadoop/_blank)) 的启发
* 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

​​

* **Hadoop的优势**
* **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，在出现故障时可以对失败的节点重新分布处理。

* **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。

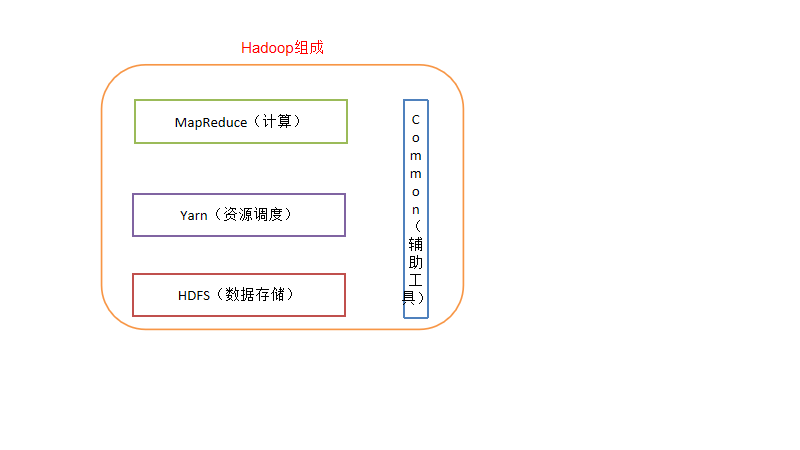
* **高效性：**

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

* **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。

* **Hadoop组成**

​​

* Hadoop HDFS：
* 一个高可靠、高吞吐量的分布式文件系统。
* Hadoop MapReduce：
* 一个分布式的离线并行计算框架。
* Hadoop YARN：
* 作业调度与集群资源管理的框架。
* Hadoop Common：
* 支持其他模块的工具模块（Configuration、RPC、序列化机制、日志操作）。
* **HDFS架构概述**
* **Namenode：存储元数据**
* **Datanode：存储数据的节点，会对数据块进行校验**
* **Secondarynamenode：监控namenode 的元数据，每隔一定的时间进行元数据的合并**
* **YARN架构概述**
* **ResourceManager(rm)：**

处理客户端请求、启动/监控ApplicationMaster、监控NodeManager、资源分配与调度

* **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

* **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

* **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

* **MapReduce架构概述**
* MapReduce将计算过程分为两个阶段：Map和Reduce
* Map阶段并行处理输入数据
* Reduce阶段对Map结果进行汇总

**三 Hadoop运行环境搭建**

**环境配置**

* **关闭防火墙**
* 关闭防火墙：systemctl stop firewalld.service
* 禁用防火墙：systemctl disable firewalld.service
* 查看防火墙：systemctl status firewalld.service
* 关闭Selinux：vi /etc/selinux /config
* 将SELINUX=enforcing改为SELINUX=disabled
* **修改IP**
* 善用Tab键
* vi /etc/sysconfig/network-scripts/ifcfg-ens33
* BOOTPROTO=static
* ONBOOT=yes
* IPADDR=192.168.X.51
* GATEWAY=192.168.X.2
* DNS1=8.8.8.8
* DNS2=8.8.4.4
* NETMASK=255.255.255.0
* vi /etc/resolv.conf
* nameserver 8.8.8.8
* nameserver 8.8.4.4

重启网卡：service network restart

* **修改主机名**
* hostnamectl set-hostname 主机名
* **IP和主机名关系映射**
* vi /etc/hosts

192.168.1.51 bigdata111

192.168.1.52 bigdata112

192.168.1.53 bigdata113

* 在windows的C:\Windows\System32\drivers\etc路径下找到hosts并添加

192.168.1.51 bigdata111

192.168.1.52 bigdata112

192.168.1.53 bigdata113

* **Xshell**

输入IP、用户名和密码

* **在opt目录下创建文件（此步可选）**
* **创建itstar用户**
* adduser itstar
* passwd itstar
* **设置itstar用户具有root权限**
* vi /etc/sudoers 92行 找到root ALL=(ALL) ALL
* 复制一行：itstar ALL=(ALL) ALL
* **安装jdk**
* 卸载现有jdk

（1）查询是否安装java软件：

rpm -qa|grep java

（2）如果安装的版本低于1.7，卸载该jdk：

rpm -e 软件包名字

* 在/opt目录下创建两个子文件

mkdir /opt/mod /opt/soft

* 解压jdk到/opt/module目录下

tar -zxvf jdk-8u144-linux-x64.tar.gz -C /opt/mod/

* 配置jdk环境变量

vi /etc/profile

export JAVA\_HOME=/opt/mod/jdk1.8.0\_144

export PATH=$PATH:$JAVA\_HOME/bin

source /etc/profile

* 测试jdk安装成功
* java -version
* java version "1.8.0\_144"

**四 Hadoop运行模式**

**伪/完全分布式部署Hadoop**

* **SSH无密码登录**
* 生成公钥和私钥：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）

* 将公钥拷贝到要免密登录的目标机器上
* ssh-copy-id 主机名1
* ssh-copy-id 主机名2
* ssh-copy-id 主机名3

注：在另外两台机器上分别执行，共执行9遍

* **.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

* **配置集群(表格版)**

1. 集群部署规划:

|  |  |  |  |
| --- | --- | --- | --- |
|  | bigdata111 | bigdata112 | bigdata113 |
| HDFS | NameNode  SecondaryNameNode  DataNode | DataNode | DataNode |
| YARN | NodeManager | ResourceManager  NodeManager | NodeManager |

1. 配置文件：

|  |  |
| --- | --- |
| **文件** | **配置** |
| core-site.xml | <!-- 指定HDFS中NameNode的地址 -->  <property>  <name>fs.defaultFS</name>  <value>hdfs://主机名1:9000</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.X.X/data/tmp</value>  </property> |
| hdfs-site.xml | <!--数据冗余数-->  <property>  <name>dfs.replication</name>  <value>3</value>  </property>  <!--secondary的地址-->  <property>  <name>dfs.namenode.secondary.http-address</name>  <value>主机名1:50090</value>  </property>  <!--关闭权限-->  <property>  <name>dfs.permissions</name>  <value>false</value>  </property> |
| 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>主机名1</value>  </property>  <!-- 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(秒) -->  <property>  <name>yarn.log-aggregation.retain-seconds</name>  <value>604800</value>  </property> |
| mapred-site.xml | <!-- 指定mr运行在yarn上-->  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  <!--历史服务器的地址-->  <property>  <name>mapreduce.jobhistory.address</name>  <value>主机名1:10020</value>  </property>  <!--历史服务器页面的地址-->  <property>  <name>mapreduce.jobhistory.webapp.address</name>  <value>主机名1:19888</value>  </property> |
| hadoop-env.sh、yarn-env.sh、mapred-env.sh（分别在这些的文件中添加下面的路径）  export JAVA\_HOME=/opt/module/jdk1.8.0\_144（注：是自己安装的路径） | |
| slaves | bigdata111、bigdata112、bigdata113（自己设置的主机名） |

* 格式化Namenode：

hdfs namenode -format

**为什么要格式化？**

NameNode主要被用来管理整个分布式文件系统的命名空间(实际上就是目录和文件)的元数据信息，同时为了保证数据的可靠性，还加入了操作日志，所以，NameNode会持久化这些数据(保存到本地的文件系统中)。对于第一次使用HDFS，在启动NameNode时，需要先执行-format命令，然后才能正常启动NameNode节点的服务。

**格式化做了哪些事情？**

在NameNode节点上，有两个最重要的路径，分别被用来存储元数据信息和操作日志，而这两个路径来自于配置文件，它们对应的属性分别是dfs.name.dir和dfs.name.edits.dir，同时，它们默认的路径均是/tmp/hadoop/dfs/name。格式化时，NameNode会清空两个目录下的所有文件，之后，会在目录dfs.name.dir下创建文件

**hadoop.tmp.dir** 这个配置，会让dfs.name.dir和dfs.name.edits.dir会让两个目录的文件生成在一个目录里

* 启动集群得命令：

Namenode的主节点：sbin/start-dfs.sh

Yarn的主节点：sbin/stop-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。

* **scp文件传输**

实现两台远程机器之间的文件传输（bigdata112主机文件拷贝到bigdata113主机上）

scp -r [文件] 用户@主机名：绝对路径

注：伪分布式是一台、完全分布是三台

* **完全分布式**

步骤：

1）克隆2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

export JAVA\_HOME=/opt/mod/jdk1.8.0\_144

export HADOOP\_HOME=/opt/mod/hadoop-2.8.4

export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME\bin:$HADOOP\_HOME\sbin

6）安装ssh

7）配置集群

8）启动测试集群

注：此配置直接使用虚拟机克隆伪分布式两台即可

* **自带官方wordcount案例**
* **随意上传一个文本文件**
* **上传命令：**hadoop fs -put 文件名 /
* **执行命令：**

hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.X.X.jar wordcount /入 /出

* **命令解析：**

hadoop jar 路径的jar包 全类名 输入路径 输出路径

* **查看结果：**

hadoop fs -cat 路径

**Hadoop启动和停止命令：**

以下命令都在$HADOOP\_HOME/sbin下，如果直接使用，记得配置环境变量

|  |  |
| --- | --- |
| 启动/停止历史服务器 | mr-jobhistory-daemon.sh start|stop historyserver |
| 启动/停止总资源管理器 | yarn-daemon.sh start|stop resourcemanager |
| 启动/停止节点管理器 | yarn-daemon.sh start|stop nodemanager |
| 启动/停止 NN 和 DN | start|stop-dfs.sh |
| 启动/停止 RN 和 NM | start|stop-yarn.sh |
| 启动/停止 NN、DN、RN、NM | start|stop-all.sh |
| 启动/停止 NN | hadoop-daemon.sh start|stop namenode |
| 启动/停止 DN | hadoop-daemon.sh start|stop datanode |