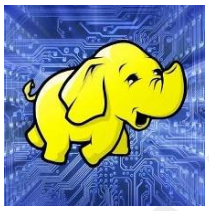
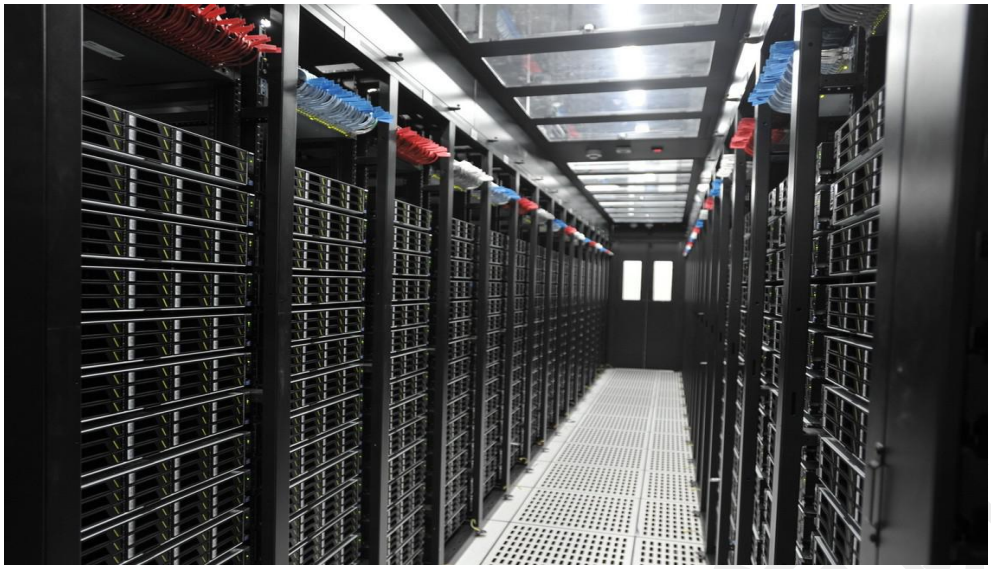
大数据技术之 Hadoop

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 的子项目 Nutch 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) 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。

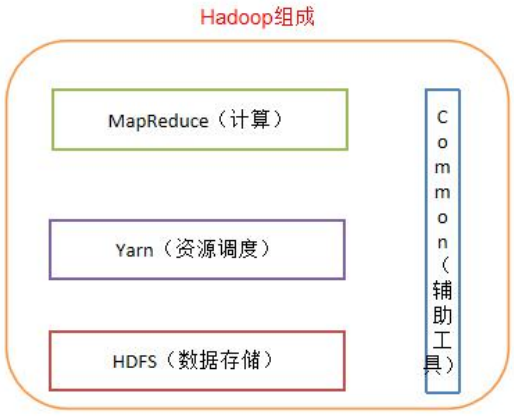
- **高效性：**

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**
- **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：servie 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

- **连接 Secure CRT & 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/module /opt/software

■ 解压 jdk 到/opt/module 目录下

tar –zxvf jdk–8u144–linux–x64.tar.gz –C /opt/module/

■ 配置 jdk 环境变量

vi /etc/profile

export JAVA_HOME=/opt/module/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-key gen -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	ResourceManager NodeManager	NodeManager	NodeManager

2）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

- 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

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