# 第二章Hadoop环境设置

## 实验一：搭建Hadoop单机模式环境（可选）

|  |
| --- |
| 【实验名称】 搭建Hadoop单机模式环境 |
| 【实验目的】 |
| 掌握Hadoop单机模式环境的搭建   1. 安装Java环境 2. 搭建Hadoop单机模式   注意：前置条件，虚拟机、ubuntu，并且环境变量设置已经完成，未安装的学生请参考教材虚拟机，ubuntu安装步骤。 |
| 【实验原理】 |
| 安装Java环境和搭建Hadoop单机模式环境 |
| 【实验环境】 |
| ubuntu系统 |
| 【实验步骤】 |
| 1.安装前的准备  1.1关闭防火墙（ubuntu系统是默认关闭的）  查看防火墙状态：  $ sudo ufw status  状态：激活  若是激活状态则关闭防火墙：  $ sudo ufw disable  防火墙在系统启动时自动禁用  查看防火墙状态：  $ sudo ufw status  状态：不活动  1.2设置NTP时间服务器  查看时间服务是否运行：  $ sudo dpkg -l | grep ntp  若NTP尚未安装就运行安装命令：  $ sudo apt-get install ntp  1.3创建用户  1.3.1添加用户名为hadoop的用户:  $ sudo adduser hadoop  切换到新帐户：  $ su - hadoop  1.3.2 SSH设置和密钥生成：  安装SSH服务端openssh-server：  $ sudo apt-get install openssh-server  重启ssh服务  $ sudo /etc/init.d/ssh restart  SSH密钥生成：  $ ssh-keygen -t rsa -P ''  查看密钥：  $ ls ~/.ssh  将公钥追加到~/.ssh/authorized\_keys文件中：  $ ssh-copy-id -i ~/.ssh/id\_rsa.pub 主机名  2.下载安装JDK  2.1下载JDK1.8  打开网页下载  http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html    再通过WinSCP工具（WinSCP主要功能是在本地与远程计算机间安全的复制文件，需先安装）上传至Ubuntu系统桌面目录下。如果虚拟机已经安装VMware tools，可以直接将安装包文件从Windows系统拖拽至Ubuntu桌面  2.2解压源码包：  进入终端切换至桌面目录，并将其解压到~目录下  $ cd ~  $ tar -zxvf jdk-8u181-linux-x64.tar.gz -C ~  2.3配置JDK环境变量  $ vi ~/.bashrc  然后在末尾添加  export JAVA\_HOME=~/jdk1.8.0\_171  export JRE\_HOME=${JAVA\_HOME}/jre  export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib  export PATH=${JAVA\_HOME}/bin:$PATH  2.4刷新配置：使配置立即生效  $ source ~/.bashrc  2.5检验是否安装成功  $ java –version  出现以下版本号信息为成功安装  java version "1.8.0\_171"  Java(TM) SE Runtime Environment (build 1.8.0\_171-b11)  Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)  3.下载安装Hadoop  下载方法一（推荐使用）：  进入Apache官网http://archive.apache.org/dist/hadoop/common/选择对应版本的hadoop二进制（binary）压缩包，下载到windows系统目录下，再通过WinSCP工具将安装包上传至Ubuntu系统。  下载方法二：在Ubuntu终端用wget命令直接下载至Ubuntu系统中（命令在同一行）。  $wget http://archive.apache.org/dist/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz  解压安装包：解压安装包至~目录下，并创建软链接  $ tar zxvf hadoop-2.7.3.tar.gz -C ~  $ ln -s ~/hadoop-2.7.3 ~/hadoop  执行命令$ ./bin/hadoop version 查看hadoop版本信息：  $ cd ~/hadoop/etc/hadoop  $ vi hadoop-env.sh  找到export JAVA\_HOME，把#去掉，并修改为：  # The java implementation to use.  export JAVA\_HOME=/home/hadoop/jdk1.8.0\_171  如果出现以下信息，说明安装成功：  Hadoop 2.7.3  Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be5982de4719c1c8af91ccff  Compiled by root on 2016-08-18T01:41Z  Compiled with protoc 2.5.0  From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4  This command was run using ~/hadoop/share/hadoop/common/hadoop-common-2.7.3.jar |
|  |

## 实验二：搭建Hadoop伪分布式模式环境

|  |
| --- |
| 【实验名称】 搭建Hadoop伪分布式模式环境 |
| 【实验目的】 |
| 掌握Hadoop伪分布式环境搭建 |
| 【实验原理】 |
|  |
| 【实验环境】 |
| 1. 已经部署好的Hadoop单机模式环境的个人计算机上   或 （2）在教学实验系统EDU搭建  注：若是在教学实验系统EDU搭建伪分布式，需要同学自己把hadoop-2.7.3.tar.gz上传至/home/账号 |
| 【实验步骤】  1.在已经部署好的Hadoop单机模式环境的个人计算机上  配置Hadoop环境变量：  使用vi命令打开~/.bashrc文件：  $ vi ~/.bashrc  在打开文件的末尾添加以下两行代码，保存并退出。  export HADOOP\_HOME=~/hadoop  export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin  为了可以在任意目录下使用hadoop命令，我们需要在~/.bashrc目录下配置hadoop的环境变量： |
| 使配置立即生效  $ source ~/.bashrc  验证：切换到任意目录，比如~目录，用hadoop version命令验证，输出版本信息为正确配置。  $ cd ~  $ hadoop version   1. 设置Hadoop配置文件   配置hadoop-env.sh（注：若在EDU教学实验环境中配置JAVA\_HOME可以用命令：whereis Java 来查找JAVA\_HOME路径，然后配置到hadoop-env.sh）  进入目录在${HADOOP\_HOME}/etc/hadoop:  $ cd ~/hadoop/etc/hadoop  打开hadoop-env.sh:  $ vi hadoop-env.sh  将export JAVA\_HOME=${JAVA\_HOME} 改为  export JAVA\_HOME=~/jdk1.8.0\_171  配置core-site.xml  <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>hadoop.tmp.dir</name>  <value>/home/hadoop/hadoop/tmp</value>  </property>  <property>  <name>fs.defaultFS</name>  <value>hdfs://localhost</value>  </property>  </configuration>  配置hdfs-site.xml  <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  </configuration>  配置mapred-site.xml  $ cp mapred-site.xml.template mapred-site.xml  <?xml version="1.0"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  </configuration>  配置yarn-site.xml  <?xml version="1.0"?>  <configuration>  <property>  <name>yarn.resourcemanager.hostname</name>  <value>localhost</value>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  </configuration>   1. 格式化HDFS文件系统：   格式化的过程是创建初始目录和文件系统结构的过程。格式化只需进行一次，下次启动不再需要格式化。执行以下命令：  $ hdfs namenode -format   1. 启动Hadoop守护进程验证   分别启动HDFS和YARN进程：  启动HDFS：  $start-dfs.sh  用$ jps命令验证，正确启动会出现以下三个进程：  $ jps  NameNode  DataNode  SecondaryNameNode  启动YARN：  $start-yarn.sh  用$ jps命令验证，正确启动将多出以下两个进程：  $ jps  ResourceManager  NodeManager  注意：启动命令在{$HADOOP\_HOME}/sbin目录下。   1. 浏览器验证：   1） 浏览器访问Hadoop文件系统  浏览器访问Hadoop文件系统默认端口号为50070，打开Ubuntu自带的火狐浏览器，输入以下网址获得Hadoop Web UI的服务，可以查看 NameNode 和 Datanode 信息，还可以在线查看 HDFS 中的文件。网址为：http://localhost:50070    注意：如想在宿主机（物理机）的浏览器访问则需要将localhost换成虚拟机的ip，用ifconfig命令查看虚拟机ip ，如：<http://192.168.30.128:50070>  2） 浏览器查看集群所有应用程序  访问集群中的所有应用程序的默认端口号为8088。使用以下URL访问该服务。网址为：http://localhost:8088     1. 停止所有进程   分别关闭HDFS和YARN  $stop-dfs.sh  $stop-yarn.sh  或者使用以下命令停止所有进程：  $stop-all.sh  用命令jps查看：关闭了Hadoop所有进程  $ jps  jps  至此Hadoop伪分布式搭建完成。 |

## 实验三：搭建Hadoop完全分布式模式环境（可选）

|  |
| --- |
| 【实验名称】 搭建Hadoop完全分布式模式环境 |
| 【实验目的】 |
| 掌握搭建Hadoop完全分布式模式环境 |
| 【实验原理】 |
|  |
| 【实验环境】 |
| 已经部署好的Hadoop伪分布式环境 |
| 【实验步骤】 |
| 准备工作：  1. 在伪分布模式下，确保已关闭Hadoop所有进程。  2. 为了完全分布式能正确格式化HDFS系统，删除${HADOOP\_HOME}/tmp及${HADOOP\_HOME}/logs目录下的所有内容。  1）删除${HADOOP\_HOME}/tmp目录下所有内容：  $ rm -rf ~/hadoop/tmp/\*  2) 删除${HADOOP\_HOME}/log目录下所有内容：  $ rm -rf ~/hadoop/log/\*  配置完全分布式模式：   1. 修改主机名称   1）将/etc/ hostname文件修改成Master：  $sudo vi /etc/hostname  Master  注意：reboot重启后才生效  2）将/etc/hosts文件修改成如下内容：  $ sudo vi /etc/hosts  127.0.0.1 localhost  192.168.30.128 Master  192.168.30.129 Slave1  192.168.30.130 Slave2  注意：这里先指定了Slave1/Slave2机器及对应的IP，这些机器在后面步骤会由Master机器克隆出来。  3）重启Ubuntu生效：  $ reboot  重启后主机名变成了Master，用命令$ ping Master，就可以ping通Master主机了。  hadoop@Master:~$ ping Master  PING Master (192.168.30.128) 56(84) bytes of data.  64 bytes from Master (192.168.30.128): icmp\_seq=1 ttl=64 time=0.014 ms  64 bytes from Master (192.168.30.128): icmp\_seq=2 ttl=64 time=0.085 ms   1. 删除原来的SSH配置   因为主机名称已改变，新的主机名还没有加到公私密钥中，用$ ssh Master不能免密登录，需要删除原来的SSH，待重新生成新的SSH。  $ rm -rf ~/.ssh/\*   1. 克隆客户机   因为要搭建由3台机器组成的完全分布式集群，现在只有1台Master机器，所以需要用Master机器克隆出另外2台机器。  克隆方法：  1） 关闭ubuntu客户机，否则进行无法克隆操作  2） 左侧栏点击要克隆的ubuntu客户机，右键🡪管理🡪克隆    3） 连续点击下一步至克隆类型，选择创建完整克隆，点击下一步    4） 修改虚拟机名称和位置：关于位置选择，最好每台客户机单独建一个文件夹：    5） 点击完成，等待克隆完成，此过程大概需要3-5分钟。  按照如上方法，一共克隆出2台机器。  6） 修改2台克隆机的主机名称  打开第一台克隆机，将其主机名称命名为Slave1，然后reboot重启生效。  $ sudo vi /etc/hostname  Slave1  $ reboot  打开第二台克隆机，将它的主机名称命名为Slave2，然后reboot重启生效。  $ sudo vi /etc/hostname  Slave2  $ reboot   1. SSH配置   在完全分布式模式下，集群内所有机器需要免密登录到集群内所有机器，即 Master、Slave1、Slave2三台机器需要免密登录到其他任意两台机器。免密登录的方法和伪分布的免密登录方法一样，在Master、Slave1、Slave2机器生成公/私密钥对，然后将公钥发送给集群内的所有机器，以Master免密登录其他集群内的所有机器为例，其他两台机器重复以下操作。  1）在Master主机生成密钥对  $ ssh-keygen -t rsa -P ''  2） 将Master公钥id\_rsa.pub追加到Master的~/.ssh/authorized\_keys文件中，并复制到另外2台Slave机器上  $ ssh-copy-id -i ~/.ssh/id-rsa.pub Master  $ ssh-copy-id -i ~/.ssh/id-rsa.pub Slave1  $ ssh-copy-id -i ~/.ssh/id-rsa.pub Slave2  3） 验证免密登录：在Master主机验证免密码登录集群所有机器  $ ssh Master  $ ssh Slave1  $ ssh Slave2   1. 配置完全分布式文件   在伪分布式模式下已经配置了hadoop-env.sh，现在只需要配置4个xml即可，再将配置分发给其他机器。注意：以下配置均在Master机器操作。  进入Master机器的hadoop配置文件目录${HADOOP\_HOME}/etc/hadoop/  $ cd ~/hadoop/etc/hadoop/  1） 配置core-site.xml  修改core-site.xml文件内容为：  <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>hadoop.tmp.dir</name>  <value>/home/hadoop/hadoop/tmp</value>  </property>  <property>  <name>fs.defaultFS</name>  <value>hdfs://Master</value>  </property>  </configuration>  注意：fs.defaultFS修改为Master机器的IP地址或者主机名称  2）配置hdfs-site.xml  修改hdfs-site.xml文件内容为：  <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>dfs.replication</name>  <value>2</value>  </property>  </configuration>  3） 配置mapred-site.xml  mapred-site.xml与伪分布式的配置一样，不需改变。  4） 配置yarn-site.xml  修改yarn-site.xml文件内容为：  <?xml version="1.0"?>  <configuration>  <property>  <name>yarn.resourcemanager.hostname</name>  <value>Master</value>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  </configuration>  5） 分发配置：将Master的配置目录文件分发至其他机器  $ cd ~/hadoop/etc/  $ scp -r hadoop hadoop@Slave1:~/hadoop/etc/  $ scp -r hadoop hadoop@Slave2:~/hadoop/etc/   1. 指定集群Slaves配置   此过程是与伪分布式模式最根本的区别。配置Slavers文件就是制定哪些机器是Slave机器，Hadoop将变成完全分布式模式。配置方法如下：  在Master主机操作，进入配置目录${HADOOP\_HOME}/etc/hadoop/，修改slavers文件  $ cd ~/hadoop/etc/hadoop  $ vi slaves  将slaves文件的内容修改为：  Slave1  Slave2  注意：1） 此配置只需要在Master机器进行，其他从机不需要配置此文件；  2） 要将slaves文件原来的localhost删除掉，再添加Slave主机名称， Slave主机名称也可以改为Slave IP地址。   1. 格式化HDFS文件系统   $ hdfs namenode -format   1. 启动Hadoop守护进程   分别启动HDFS和YARN  $ start-dfs.sh  $ start-yarn.sh  或者用以下命令启动HDFS和YARN, 注意：此命令已经过时，不推荐使用。  $ start-all.sh   1. 验证Hadoop进程   1) $ jps命令验证：  用$ jps命令分别在每一台机器验证  $ jps  执行jps命令后，Master主机包含以下三个进程为启动成功  SecondaryNameNode  NameNode  ResourceManager  分别在三台Slave主机执行jps命令，包含以下两个进程为启动成功  NodeManager  DataNode  注：如果某个机器少了某个进程，应该到相应机器去找对应的log查看原因，log存放路径为${HADOOP\_HOME}/logs目录下。例如：Slave2机器少了DataNode进程，则应该进入到Slave2机器的${HADOOP\_HOME}/logs目录下，查看datanode相关的log，找到启动失败的根源。  $ cd ~/hadoop/logs  $ more hadoop-hadoop-datanode-Slave2.log  2） 浏览器（WebUI）验证  查看hadoop的文件系统，可以查看 NameNode 和 Datanode 信息，还可以在线查看 HDFS 中的文件：在Master机器浏览器输入网址: <http://Master:50070> ，出现以下界面为成功。 注：用虚拟机内部的浏览器。宿主机(物理机)访问则需要将localhost换成虚拟机的ip，用ifconfig命令查看虚拟机ip ，如：<http://192.168.30.128:50070>    点击Datanodes，显示有两台datanode信息。  230008702545156602  浏览器查看集群所有应用程序  访问集群中的所有应用程序的默认端口号为8088。输入网址：http://Master:8088  可以看到Active Nodes为2，说明集群有2个节点正在运行。  822054064279325494 |
|  |