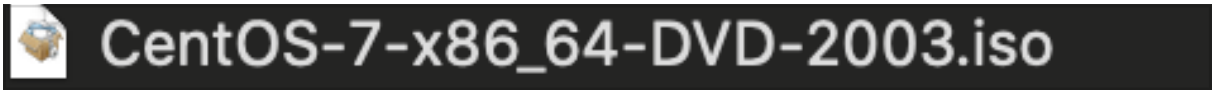


利用虚拟机搭建Hadoop完全分布式集群

大致流程

- 虚拟机操作软件选择VMware Fusion
- 虚拟机选择Centos 7，官网下载，认准x86架构



- 因为集群需要多台虚拟机，先配好一台，然后克隆
 - 配置网络，多台虚拟机ip不同
 - 配置host，与ip一一对应
 - 安装rsync，编写xsync集群部署脚本
 - 安装Java和Hadoop，并配置环境变量，官网下载，Hadoop认准Hadoop2.7.X，Java就1.8，版本选择别整事
- 集群多台机器需要启动的服务不同，我用了三台虚拟机（配置不行要我高低我整10台），两主一从，启动的服务一会说

虚拟机Nat模式网络配置

- 虚拟机的终端贼他妈不好使，可以用mac本地终端ssh一下

物理机端配置

- 首先需要关闭mac端的DHCP（重要），保证虚拟机网关与IP处于同一网段

```
修改/Library/Preferences/VMware Fusion 目录下的networking文件  
将answer VNET_3_DHCP yes 设置为no
```

- 确保虚拟机处于Nat模式下（Nat模式与物理机处于同一网段，二者同级）
- 然后进入vnet8目录下，查看nat.conf文件，nat.conf里的ip和natmask是一会我们虚拟机配置里的gateway和netmask，虚拟机ipaddr需要和网关同一网段内

```
(base) madi:vmnet8/ $ cat nat.conf
# VMware NAT configuration file
# Manual editing of this file is not recommended. Using UI is preferred.

[host]

# NAT gateway address
ip = 192.168.194.2
netmask = 255.255.255.0

# VMnet device if not specified on command line
device = vmnet8
```

- 修改完配置重启网络

```
sudo ifconfig en0 down
```

```
sudo ifconfig en0 up
```

- 查看DNS，设置-网络-高级-DNS，如有多个，写成DNS1=XXX，DNS2=XXX
- 修改完配置ping一下网络，看看物理机有没有被玩坏

虚拟机端配置

- 进入/etc/sysconfig/network-scripts目录，修改ifcfg-ens33（Centos6为ifcfg-enth0）

```
BOOTPROTO = static
ONBOOT = yes

IPADDR = 物理机dhcp文件的range段里
NETMASK = 物理机NETMASK （一般是255.255.255.0）
GATEWAY = 物理机nat.conf文件里的ip
DNS = 设置里的DNS值
```

- 修改完重启网络（

```
CentOS7: systemctl network restart
```

```
CentOS6: service network restart
```

- 如果ping不通试着重启VMware和虚拟机
- 名字修改，修改/etc/hosts，IP对应端口号

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.194.131  hadoop101
192.168.194.132  hadoop102
192.168.194.133  hadoop103
```

网络配置详细博客

[CentOS7配置网卡为静态IP，如果你还学不会那真的没有办法了！_牧马人-CSDN博客](#)
[_centos7 网卡配置及静态ip配置 原创](#)

防火墙相关

```
查看防火墙状态: firewall-cmd --state
关闭防火墙: systemctl stop firewalld.service
禁止防火墙开机启动: systemctl disable firewalld.service
查看服务列表: systemctl list-unit-files
```

xsync相关

- 是一个集群文件分发脚本
- 文件传输可用scp，但是很麻烦，集群内文件目录相同，就没必要一个一个写
- 可分发多个集群，shell里的while循环

xsync配置

- 先下载包

```
yum -y install rsync # xsync是基于rsync的，rsync是远程同步工具，xsync是脚本
```

- 配置集群里各主机的/etc/hosts，让ip和主机名一一对应，不对应也可以，在xsync脚本中把主机名改成ip地址就行，麻烦点
 - 确保主机间可以ssh的情况下，编写脚本并执行
-

```
#!/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 -P指向实际物理地址，防止软连接
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=102; host<104; host++)); do
    echo ----- hadoop$host -----
    rsync -rvl $pdir/$fname root@hadoop$host:$pdir
done
```

JAVA相关

- 下载Java，java应该是一个具体的版本，选择开发版，可以先yum search看一下

```
yum -y install java
```

- 配置环境变量

```
vi /etc/profile
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-0.el7_8.x86_64
```

```
export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/root/hadoop-2.7.2
export PATH=$PATH:$HADOOP_HOME/bin

# 具体配置看情况，别忘记激活，完事echo $JAVA_HOME看一下
source /etc/profile
```

Hadoop相关

- 下载Hadoop，官网找，可以调出Hadoop的配置文件的文档，写的比CSDN上那帮2B好多了
- 解压，放在自己的目录下，这个目录就是环境变量
- 配置环境变量 /etc/profile

```
export HADOOP_HOME=/root/hadoop-2.7.2
export PATH=$PATH:$HADOOP_HOME/bin
# 路径根据自己需要调整
# 公司级的服务器有严格的目录标准，我图省事放root下了
```

集群配置

- 需要配置四个东西HDFS，YARN，Mapreduce，Core（核心文件），Slaves文件
- core-site.xml，其中namenode地址、tmp文件夹位置按需选择
 - tmp文件夹的作用是保存mapreduce的中间操作生成的临时文件
 - 当mapreduce完成时，临时文件被删除
 - 劝你选个好找的

```
# 指定NameNode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>/root/hadoop-2.7.2/data/tmp</value>
</property>
```

- HDFS

- HDFS有两个文件需要修改，hadoop-env.sh和hdfs-site.xml
- 为什么叫hadoop-env我也不知道，可能因为HDFS是Hadoop的灵魂吧，我猜的

```
# hadoop-env.sh
export JAVA_HOME=$JAVA_HOME
# 所有涉及env的都是修改JAVA_HOME
# 注意，文件的注释里有了，解开就行
```

```
# hdfs-site.xml
<!-- 指定Hadoop辅助名称节点主机配置 -->
# 3代表上传文件的备份数，默认为3
<property>
    <name>dfs.replication</name>
    <value>3</value>
</property>

<!-- 指定Hadoop备份从主机地址 -->
# 我的是101主机，102备份主机，103从机
<property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>hadoop102:50090</value>
</property>

<property>
    <name>dfs.datanode.http-address</name>
    <value>hadoop103:50010</value>
</property>
```

- YARN

```
# yarn-site.xml

<!-- Site specific YARN configuration properties -->
<!-- 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>

<!-- 指定YARN的SecondaryNameNode的地址 -->
<property>
    <name>yarn.secondarynamenode.hostname</name>
    <value>hadoop102</value>
</property>

<!-- 指定YARN的DataNode的地址 -->
<property>
    <name>yarn.datanode.hostname</name>
    <value>hadoop103</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 -->
<property>
    <name>yarn.log-aggregation.retain-seconds</name>
```

```
<value>604800</value>
</property>
```

- MapReduce

```
# mapred-site.xml
<!-- 指定MR运行在Yarn上 -->
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
# 这里是没有这个文件的，只有一个mapred-site-temp.xml，多此一举，复制一下，或者直接把模版删了就行
```

至此，单一虚拟机就全部搞完了，可以玩命克隆了，你想整几个节点都行

克隆的虚拟机配置修改

- IP，去/etc/sysconfig/network-scripts目录，修改ifcfg-ens33，把IPADDR改了，不同节点不一样，最好跟名字对应，改完重启网络

```
systemctl restart network
```

- 如果没有在克隆之前完成Java和Hadoop的配置也没关系，利用集群分发脚本

```
rsync -rvl /root /root
# 第一个路径是你配好环境的root目录，第二个是其他节点的目录，按需选择
# 具体看xsync脚本，里面的IP和host都可控
```

- Slaves文件

- 在hadoop家目录下/etc/hadoop/slaves
- 加入你所有的节点名字，注意不要有空格空行
- 为什么不能空，我也不知道，我看的博客说了，别惹事还是加上

- 以上任何文件如果是克隆前修改的都可以通过xsync发到其他节点，过程需要一个一个输密

码，实际可以改成公私钥的形式，免除密码

集群启动

振奋人心的时刻

- 我的集群
 - Hadoop101（主节点）：启动了ResourceManager, NodeManager, NameNode
 - Hadoop102（备份主）：启动了SecondaryNameNode, NodeManager
 - Hadoop103（从节点）：启动了DataNode, NodeManager
- 启动文件都在HADOOP_HOME的sbin下

```
# 第一次启动需要格式化NameNode
hdfs namenode -format

# 主节点启动YARN和NameNode
# YARN包括了RM和NM
# NameNode和ResourceManger如果不是同一台机器，不能在NameNode上启动 YARN，应该在
ResourceManager所在的机器上启动YARN
```

集群测试

- 浏览器访问集群管理UI界面
 - HDFS: hadoop101:50070
 - YARN: hadoop101:8088
 - 具体情况看集群启动服务的情况
- 利用命令在HDFS上创建文件夹，上传文件夹

```
hadoop fs -mkdir /Test
hadoop fs -put /root/test.txt /Test
```

- 执行mapreduce demo

```
# 先上传一个wordcount.txt文件
# 然后执行mapreduce脚本
```

在/root/hadoop-2.7.2/share/hadoop/mapreduce下

```
hadoop jar hadoop-mapreduce-examples-2.7.2.jar wordcount /Test/wordcount.txt /  
Test/Output
```

第一个路径是测试文件在HDFS的路径，第二个为输出结果路径

第二个路径必须是不存在的，否则报错