

## 搭建伪分布模式

伪分布模式的安装与单机模式前面过程相同，在此基础上需要对hadoop进行一些额外的相关配置才可以正常运行。

1. 参照单机环境模式准备环境，并保证能够正常运行验证程序。
2. 编辑修改 etc/hadoop/core-site.xml，增加指定使用hdfs分布式文件系统，这里我们将其指向本机的9000端口。

```
1 <property>
2   <name>fs.defaultFS</name>
3   <value>hdfs://localhost:9000</value>
4 </property>
```

3. hdfs-site.xml 增加设置分布式文件系统中文件存储数据的副本数，因为在单机上运行，只有一个数据节点，所以参数值为1。

**注意：**当然单机模式加入的参数可保留不变

```
1 <property>
2   <name>dfs.replication</name>
3   <value>1</value>
4 </property>
```

4. hdfs-site.xml增加namenode数据存放目录，如果不指定参数，默认情况下，在进行HDFS格式化时，会将namenode数据放到/tmp目录下。

```
1 <property>
2   <name>dfs.namenode.name.dir</name>
3   <value>file:///data/hdfs/name</value>
4 </property>
```

5. hdfs-site.xml增加设置分布式文件系统数据存放目录，同样，如果不指定参数，默认情况下会将数据存入到/tmp目录下。

```
1 <property>
2   <name>dfs.datanode.data.dir</name>
3   <value>file:///data/hdfs/data</value>
4 </property>
```

6. 修改mapred-site.xml文件

```

1  <configuration>
2    <property>
3      <name>mapreduce.framework.name</name>
4      <value>yarn</value>
5    </property>
6    <!-- 原2.9.x版本可以下设置 -->
7    <property>
8      <name>mapreduce.application.classpath</name>
9      <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
10    </property>
11  </configuration>

```

## 7.修改yarn-site.xml

```

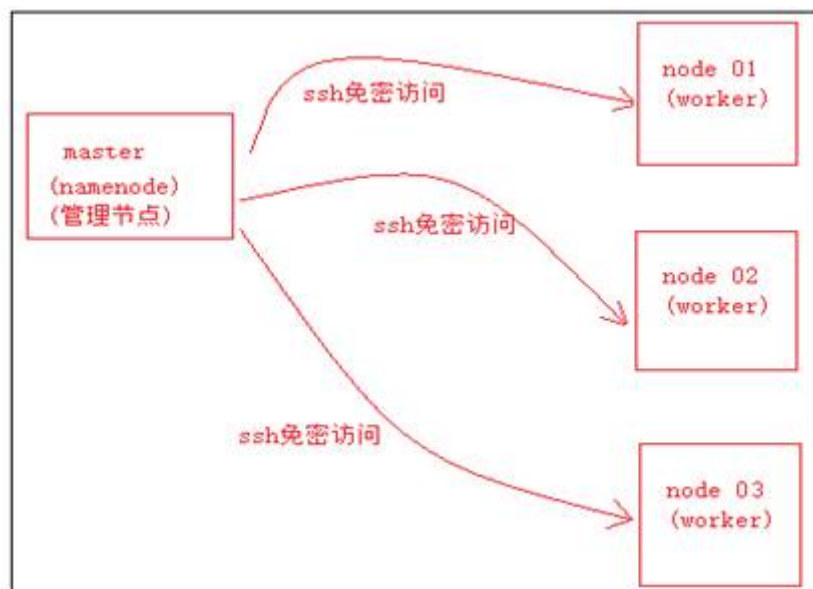
1  <configuration>
2    <property>
3      <name>yarn.nodemanager.aux-services</name>
4      <value>mapreduce_shuffle</value>
5    </property>
6
7    <!-- 原2.9.x版本可以下设置 -->
8    <property>
9      <name>yarn.nodemanager.env-whitelist</name>
10
11      <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASS
12      PATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
13    </property>
14  </configuration>

```

## 8.配置ssh免密登录

在使用分布式环境时，当Hadoop启动以后，NameNode节点会通过ssh方式来启动和停止各个DataNode上的各种守护进程的。默认情况下，ssh远程进行连接时需要使用密码来进行验证，如果结点较多，一个一个输入密码是不现实的方式，所以需要配置ssh免密码访问。

虽然伪分布式环境只有一台机器，但其工作原理与完全分布式环境基本相同，也需要配置ssh免密登录。



a.) 检查/etc/hosts中localhost对应IP为127.0.0.1，这是系统默认配置

```

[hadoop@Master ~]$ 
[hadoop@Master ~]$ cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[hadoop@Master ~]$
  
```

b.) 对用户hadoop生成一对公私钥，在介绍putty使用时，生成的私钥带有密码，不满足要求，故这里需要重新生成一份。

1 | [hadoop@Master .ssh]\$ ssh-keygen

```

[hadoop@Master .ssh]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
/home/hadoop/.ssh/id_rsa already exists.  因为之前有生成过一次私钥，这里选覆盖
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Fh4/YIo/d83HKwjlJoccyklaP5F78/0ccub35W6+dY hadoop@Master
The key's randomart image is:
+----[RSA 2048]----+
  =
  . ++=+
  ...BS*oo ++
  .+.* B+o.o X
  o .+.o+o=+
  o .. ..+=E
  +--+
+----[SHA256]-----+
[hadoop@Master .ssh]$
  
```

输入密码处直接回车，表示空密码

c.) 将生成的公钥传到相关机器对应用户名目录下authorized\_keys文件中，公钥可以手工拷文件，也可以使用openssh自带的分发工具ssh-copy-id来完成，在这里因为只有一台机器，用户也是同一个hadoop帐号，可直接将公钥追加到hadoop帐号的authorized\_keys文件中即可。

```

1 [hadoop@Master ~] cd ~/.ssh
2 [hadoop@Master .ssh] cat id_rsa.pub >> authorized_keys #或者
3 [hadoop@Master .ssh] ssh-copy-id hadoop@localhost
4 [hadoop@Master .ssh] chmod 600 authorized_keys

```

#### d.) 验证ssh免密码登录

```

[hadoop@Master .ssh]$
[hadoop@Master .ssh]$ cat id_rsa.pub >> authorized_keys
[hadoop@Master .ssh]$ chmod 600 authorized_keys
[hadoop@Master .ssh]$
[hadoop@Master .ssh]$ ssh hadoop@localhost
Last login: Thu Apr 4 14:14:09 2019 from localhost
[hadoop@Master ~]$

```

9.格式化hdfs分布式文件系统，格式化时，输出信息较多，需要注意是否有错误信息，要保证格式化是成功完成的。

```

1 [hadoop@Master ~] cd /opt/hadoop
2 [hadoop@Master hadoop] bin/hdfs namenode -format

```

格式化后的目录结构如下：

```

1 [hadoop@Master hadoop]$ find /data/hdfs/name
2 /data/hdfs/name
3 /data/hdfs/name/current
4 /data/hdfs/name/current/VERSION
5 /data/hdfs/name/current/seen_txid
6 /data/hdfs/name/current/fsimage_00000000000000000000.md5
7 /data/hdfs/name/current/fsimage_00000000000000000000
8 [hadoop@Master hadoop]$
9

```

#### 10.修改start-dfs.sh中的JAVA\_HOME变量

```

1 [hadoop@Master ~] cd /opt/hadoop
2 [hadoop@Master hadoop] vi etc/hadoop/hadoop-env.sh

```

将其中的JAVA\_HOME设置为正确路径，结果如下图：

```

23
24 # The java implementation to use.
25 export JAVA_HOME=${JAVA_HOME}
26 export JAVA_HOME=/usr/lib/jvm/java
27

```

#### 11.启动hdfs分布式文件系统

```
1 [hadoop@Master hadoop] sbin/start-dfs.sh
```

如果在启动过程中，有任何问题，可以查看/opt/hadoop/logs目录下的日志文件，从中找出错误原因并进行修正。

检查JAVA进程：

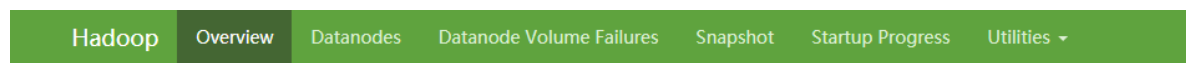
```
1 [hadoop@Master hadoop]$ jps
2 12549 NameNode
3 13029 Jps
4 12698 DataNode
5 12891 SecondaryNameNode
6 [hadoop@Master hadoop]$
7
```

## 12. 启动资源管理器

```
1 [hadoop@Master hadoop] sbin/start-yarn.sh
```

## 13. 访问hadoop分布式文件系统和资源管理器

使用浏览器打开<http://192.168.74.11:9870>即可看到hdfs的运行状态，因为CentOS在这次的环境中使用的是minimal方式安装，没有安装图形化的浏览器，所以只能用Windows系统下的浏览器进行访问。



### Overview 'localhost:9000' (active)

Started:	Tue Jun 02 17:50:38 +0800 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 23:56:00 +0800 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-71201c5f-a372-43f7-8d18-939ab36a2d6a
Block Pool ID:	BP-843796250-192.168.183.1-1591091348688

### Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 70.11 MB of 227.5 MB Heap Memory. Max Heap Memory is 839.5 MB.

Non Heap Memory used 46.9 MB of 47.94 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

资源管理器：

使用浏览器打开<http://192.168.74.11:8088>

 image-20200602180337027

## 14.测试创建用户目录并查看结果

```
1 [hadoop@Master hadoop]$ bin/hdfs dfs -mkdir /user
2 [hadoop@Master hadoop]$ bin/hdfs dfs -ls /
3 [hadoop@Master hadoop]$ bin/hdfs dfs -mkdir /user/hwadee
4 [hadoop@Master hadoop]$ bin/hdfs dfs -find /
```

或者

```
1 [hadoop@Master hadoop]$ bin/hadoop fs -mkdir /user
2 [hadoop@Master hadoop]$ bin/hadoop fs -ls /
3 [hadoop@Master hadoop]$ bin/hadoop fs -mkdir /user/hwadee
4 [hadoop@Master hadoop]$ bin/hadoop fs -find /
5
```

即hdfs dfs命令可以使用hadoop fs代替

## 15.编辑一个文件，用于测试文件中单词数量，内容不限

```
1 | vi hwadee-word.txt
```

## 16.将此文件上传到分布式文件系统中

```
1 [hadoop@Master hadoop]$ bin/hdfs dfs -put hwadee-word.txt /user/hwadee
2 [hadoop@Master hadoop]$ bin/hdfs dfs -ls /user/hwadee
3 [hadoop@Master hadoop]$ bin/hdfs dfs -cat /user/hwadee/hwadee-word.txt
```

上传测试文件到 HDFS

## 17.运行自带example程序，统计其中的单词数

```
1 [hadoop@Master hadoop]$ bin/hadoop jar \
2 share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount \
3 /user/hwadee/hwadee-word.txt /user/hwadee/output
```

```
[hadoop@Master hadoop]$
[hadoop@Master hadoop]$ bin/hdfs dfs -put hwadee-word.txt /user/hwadee
[hadoop@Master hadoop]$ bin/hdfs dfs -ls /user/hwadee
Found 1 items
-rw-r--r-- 1 hadoop supergroup 60 2019-04-04 16:40 /user/hwadee/hwadee-word.txt
[hadoop@Master hadoop]$ bin/hdfs dfs -cat /user/hwadee/hwadee-word.txt
hello hwadee
hwadee info
test
demo
abc
abc
11 22 33
11
```

文件内容

17.使用命令查看结果：

```
1 [hadoop@Master hadoop]$ bin/hdfs dfs -ls /user/hwadee/output
2
```

```
[hadoop@Master hadoop]$
[hadoop@Master hadoop]$ bin/hdfs dfs -ls /user/hwadee/output
Found 2 items
-rw-r--r-- 1 hadoop supergroup          0 2019-04-04 16:53 /user/hwadee/output/_SUCCESS
-rw-r--r-- 1 hadoop supergroup        59 2019-04-04 16:53 /user/hwadee/output/part-r-00000
```

说明运行成功

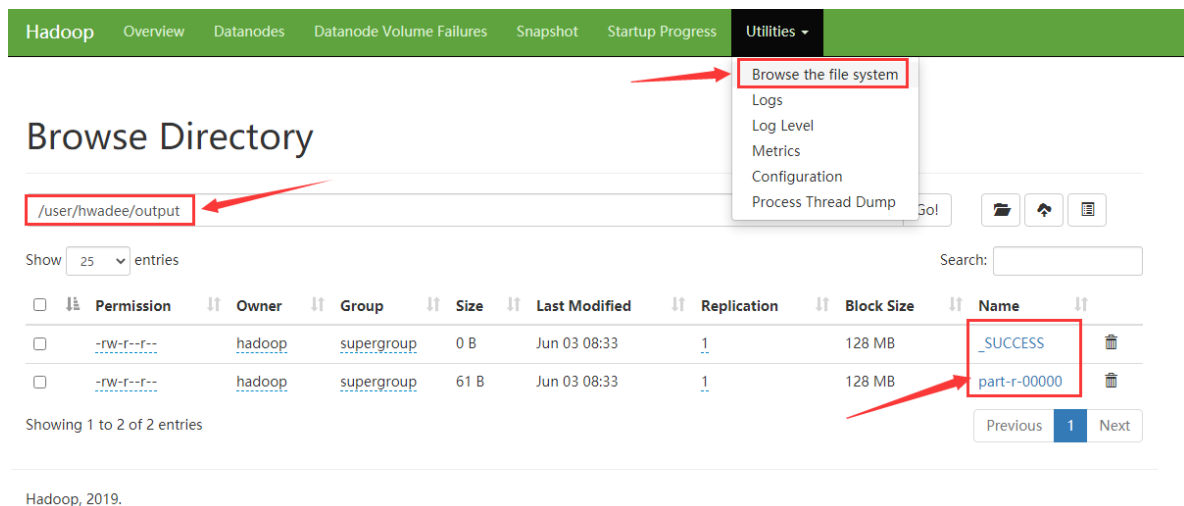
计算结果文件

```
[hadoop@Master hadoop]$ bin/hdfs dfs -cat /user/hwadee/output/part-r-00000
11      2
22      1
33      1
abc     2
demo    1
hello   1
hwadee  2
info    1
test    1
```

单词

每个单词对应数量

18.使用浏览器查看结果:



19.完善浏览器访问(可选):

在上图中, 如果在浏览器里点击文件出现文件详细信息, 如果此时想要下载文件, 会出现不成功的情况。



不能下载的原因是Download指向的链接会跳转为使用Linux的主机名或localhost，如果我们的Windows机器不能解析此主机名称，当然就不能下载，同时这里下载时会使用到另一个tcp端口9864，所以要检查Linux系统中的防火墙是否打开此端口，允许其它机器连入(建议测试时直接关闭防火墙服务)。

解决方法（总体思想就是将使用本机可对外访问的IP来运行服务）：

/etc/hosts中增加当前主机名设置

```
1 | sudo echo "192.168.74.11 master" >> /etc/hosts
```

重启dfs文件系统

```
1 | $sbin/stop-dfs.sh
2 | $sbin/start-dfs.sh
```

编辑文件C:\windows\system32\drivers\etc\hosts

增加一行

```
1 | 192.168.74.11    master
```

请将IP和Linux机器名称替换为你自己设置，然后就可以在浏览器中下载文件了。

## pi 计算显示内存分配问题，调整以下参数

yarn-site.xml

```
1 | <property>
2 |   <name>yarn.scheduler.minimum-allocation-mb</name>
3 |   <value>2048</value>
4 | </property>
5 |
```