## Hadoop学习

集群里面。一台机器就是一个节点，单机部署就是单节点，三台机器搭建的集群就是三节点

CDH安装相关目录结构和路径

1. 相关目录 /var/log/cloudera-scm-installer : 安装日志目录。  
   /var/log/\* : 相关日志文件（相关服务的及CM的）。  
   /usr/share/cmf/ : 程序安装目录。  
   /usr/lib64/cmf/ : Agent程序代码。  
   /var/lib/cloudera-scm-server-db/data : 内嵌数据库目录。  
   /usr/bin/postgres : 内嵌数据库程序。  
   /etc/cloudera-scm-agent/ : agent的配置目录。  
   /etc/cloudera-scm-server/ : server的配置目录。  
   /opt/cloudera/parcels/ : Hadoop相关服务安装目录。  
   /opt/cloudera/parcel-repo/ : 下载的服务软件包数据，数据格式为parcels。  
   /opt/cloudera/parcel-cache/ : 下载的服务软件包缓存数据。  
   /etc/hadoop/\* : 客户端配置文件目录。

hdfs-site.xml配置参数详情

|  |  |  |  |
| --- | --- | --- | --- |
| name | value | | Description |
| dfs.default.chunk.view.size | | 32768 | namenode的http访问页面中针对每个文件的内容显示大小，通常无需设置。 |
| dfs.datanode.du.reserved | | 1073741824 | 每块磁盘所保留的空间大小，需要设置一些，主要是给非hdfs文件使用，默认是不保留，0字节 |
| dfs.name.dir | | /opt/data1/hdfs/name, /opt/data2/hdfs/name, /nfs/data/hdfs/name | NN所使用的元数据保存，一般建议在nfs上保留一份，作为1.0的HA方案使用，也可以在一台服务器的多块硬盘上使用 |
| dfs.web.ugi | | nobody,nobody | NN,JT等所使用的web tracker页面服务器所使用的用户和组 |
| dfs.permissions | | true | false | dfs权限是否打开，我一般设置false，通过开发工具培训别人界面操作避免误操作，设置为true有时候会遇到数据因为权限访问不了。 |
| dfs.permissions.supergroup | | supergroup | 设置hdfs超级权限的组，默认是supergroup，启动hadoop所使用的用户通常是superuser。 |
| dfs.data.dir | | /opt/data1/hdfs/data, /opt/data2/hdfs/data, /opt/data3/hdfs/data, … | 真正的datanode数据保存路径，可以写多块硬盘，逗号分隔 |
| dfs.datanode.data.dir.perm | | 755 | datanode所使用的本地文件夹的路径权限，默认755 |
| dfs.replication | | 3 | hdfs数据块的复制份数，默认3，理论上份数越多跑数速度越快，但是需要的存储空间也更多。有钱人可以调5或者6 |
| dfs.replication.max | | 512 | 有时dn临时故障恢复后会导致数据超过默认备份数。复制份数的最多数，通常没什么用，不用写配置文件里。 |
| dfs.replication.min | | 1 | 最小份数，作用同上。 |
| dfs.block.size | | 134217728 | 每个文件块的大小，我们用128M，默认是64M。这个计算需要128\*1024^2，我碰上过有人直接写128000000，十分浪漫。 |
| dfs.df.interval | | 60000 | 磁盘用量统计自动刷新时间，单位是毫秒。 |
| dfs.client.block.write.retries | | 3 | 数据块写入的最多重试次数，在此次数之前不会捕获失败。 |
| dfs.heartbeat.interval | | 3 | DN的心跳检测时间间隔。秒 |
| dfs.namenode.handler.count | | 10 | NN启动后展开的线程数。 |
| dfs.balance.bandwidthPerSec | | 1048576 | 做balance时所使用的每秒最大带宽，使用字节作为单位，而不是bit |
| dfs.hosts | | /opt/hadoop/conf/hosts.allow | 一个主机名列表文件，这里的主机是允许连接NN的，必须写绝对路径，文件内容为空则认为全都可以。 |
| dfs.hosts.exclude | | /opt/hadoop/conf/hosts.deny | 基本原理同上，只不过这里放的是禁止访问NN的主机名称列表。这在从集群中摘除DN会比较有用。 |
| dfs.max.objects | | 0 | dfs最大并发对象数，HDFS中的文件，目录块都会被认为是一个对象。0表示不限制 |
| dfs.replication.interval | | 3 | NN计算复制块的内部间隔时间，通常不需写入配置文件。默认就好 |
| dfs.support.append | | true | false | 新的hadoop支持了文件的APPEND操作，这个就是控制是否允许文件APPEND的，但是默认是false，理由是追加还有bug。 |
| dfs.datanode.failed.volumes.tolerated | | 0 | 能够导致DN挂掉的坏硬盘最大数，默认0就是只要有1个硬盘坏了，DN就会shutdown。 |
| dfs.secondary.http.address | | 0.0.0.0:50090 | SNN的tracker页面监听地址和端口 |
| dfs.datanode.address | | 0.0.0.0:50010 | DN的服务监听端口，端口为0的话会随机监听端口，通过心跳通知NN |
| dfs.datanode.http.address | | 0.0.0.0:50075 | DN的tracker页面监听地址和端口 |
| dfs.datanode.ipc.address | | 0.0.0.0:50020 | DN的IPC监听端口，写0的话监听在随机端口通过心跳传输给NN |
| dfs.datanode.handler.count | | 3 | DN启动的服务线程数 |
| dfs.http.address | | 0.0.0.0:50070 | NN的tracker页面监听地址和端口 |
| dfs.https.enable | | true | false | NN的tracker是否监听在HTTPS协议，默认false |
| dfs.datanode.https.address | | 0.0.0.0:50475 | DN的HTTPS的tracker页面监听地址和端口 |
| dfs.https.address | | 0.0.0.0:50470 | NN的HTTPS的tracker页面监听地址和端口 |
| dfs.datanode.max.xcievers | | 2048 | 相当于linux下的打开文件最大数量，文档中无此参数，当出现DataXceiver报错的时候，需要调大。默认256 |
| name | | value | Description |
| dfs.default.chunk.view.size | | 32768 | namenode的http访问页面中针对每个文件的内容显示大小，通常无需设置。 |
| dfs.datanode.du.reserved | | 1073741824 | 每块磁盘所保留的空间大小，需要设置一些，主要是给非hdfs文件使用，默认是不保留，0字节 |
| dfs.name.dir | | /opt/data1/hdfs/name, /opt/data2/hdfs/name, /nfs/data/hdfs/name | NN所使用的元数据保存，一般建议在nfs上保留一份，作为1.0的HA方案使用，也可以在一台服务器的多块硬盘上使用 |
| dfs.web.ugi | | nobody,nobody | NN,JT等所使用的web tracker页面服务器所使用的用户和组 |
| dfs.permissions | | true | false | dfs权限是否打开，我一般设置false，通过开发工具培训别人界面操作避免误操作，设置为true有时候会遇到数据因为权限访问不了。 |
| dfs.permissions.supergroup | | supergroup | 设置hdfs超级权限的组，默认是supergroup，启动hadoop所使用的用户通常是superuser。 |
| dfs.data.dir | | /opt/data1/hdfs/data, /opt/data2/hdfs/data, /opt/data3/hdfs/data, … | 真正的datanode数据保存路径，可以写多块硬盘，逗号分隔 |
| dfs.datanode.data.dir.perm | | 755 | datanode所使用的本地文件夹的路径权限，默认755 |
| dfs.replication | | 3 | hdfs数据块的复制份数，默认3，理论上份数越多跑数速度越快，但是需要的存储空间也更多。有钱人可以调5或者6 |
| dfs.replication.max | | 512 | 有时dn临时故障恢复后会导致数据超过默认备份数。复制份数的最多数，通常没什么用，不用写配置文件里。 |
| dfs.replication.min | | 1 | 最小份数，作用同上。 |
| dfs.block.size | | 134217728 | 每个文件块的大小，我们用128M，默认是64M。这个计算需要128\*1024^2，我碰上过有人直接写128000000，十分浪漫。 |
| dfs.df.interval | | 60000 | 磁盘用量统计自动刷新时间，单位是毫秒。 |
| dfs.client.block.write.retries | | 3 | 数据块写入的最多重试次数，在此次数之前不会捕获失败。 |
| dfs.heartbeat.interval | | 3 | DN的心跳检测时间间隔。秒 |
| dfs.namenode.handler.count | | 10 | NN启动后展开的线程数。 |
| dfs.balance.bandwidthPerSec | | 1048576 | 做balance时所使用的每秒最大带宽，使用字节作为单位，而不是bit |
| dfs.hosts | | /opt/hadoop/conf/hosts.allow | 一个主机名列表文件，这里的主机是允许连接NN的，必须写绝对路径，文件内容为空则认为全都可以。 |
| dfs.hosts.exclude | | /opt/hadoop/conf/hosts.deny | 基本原理同上，只不过这里放的是禁止访问NN的主机名称列表。这在从集群中摘除DN会比较有用。 |
| dfs.max.objects | | 0 | dfs最大并发对象数，HDFS中的文件，目录块都会被认为是一个对象。0表示不限制 |
| dfs.replication.interval | | 3 | NN计算复制块的内部间隔时间，通常不需写入配置文件。默认就好 |
| dfs.support.append | | true | false | 新的hadoop支持了文件的APPEND操作，这个就是控制是否允许文件APPEND的，但是默认是false，理由是追加还有bug。 |
| dfs.datanode.failed.volumes.tolerated | | 0 | 能够导致DN挂掉的坏硬盘最大数，默认0就是只要有1个硬盘坏了，DN就会shutdown。 |
| dfs.secondary.http.address | | 0.0.0.0:50090 | SNN的tracker页面监听地址和端口 |
| dfs.datanode.address | | 0.0.0.0:50010 | DN的服务监听端口，端口为0的话会随机监听端口，通过心跳通知NN |
| dfs.datanode.http.address | | 0.0.0.0:50075 | DN的tracker页面监听地址和端口 |
| dfs.datanode.ipc.address | | 0.0.0.0:50020 | DN的IPC监听端口，写0的话监听在随机端口通过心跳传输给NN |
| dfs.datanode.handler.count | | 3 | DN启动的服务线程数 |
| dfs.http.address | | 0.0.0.0:50070 | NN的tracker页面监听地址和端口 |
| dfs.https.enable | | true | false | NN的tracker是否监听在HTTPS协议，默认false |
| dfs.datanode.https.address | | 0.0.0.0:50475 | DN的HTTPS的tracker页面监听地址和端口 |
| dfs.https.address | | 0.0.0.0:50470 | NN的HTTPS的tracker页面监听地址和端口 |
| dfs.datanode.max.xcievers | | 2048 | 相当于linux下的打开文件最大数量，文档中无此参数，当出现DataXceiver报错的时候，需要调大。默认256 |
| name | | value | Description |
| dfs.default.chunk.view.size | | 32768 | namenode的http访问页面中针对每个文件的内容显示大小，通常无需设置。 |
| dfs.datanode.du.reserved | | 1073741824 | 每块磁盘所保留的空间大小，需要设置一些，主要是给非hdfs文件使用，默认是不保留，0字节 |
| dfs.name.dir | | /opt/data1/hdfs/name, /opt/data2/hdfs/name, /nfs/data/hdfs/name | NN所使用的元数据保存，一般建议在nfs上保留一份，作为1.0的HA方案使用，也可以在一台服务器的多块硬盘上使用 |
| dfs.web.ugi | | nobody,nobody | NN,JT等所使用的web tracker页面服务器所使用的用户和组 |
| dfs.permissions | | true | false | dfs权限是否打开，我一般设置false，通过开发工具培训别人界面操作避免误操作，设置为true有时候会遇到数据因为权限访问不了。 |
| dfs.permissions.supergroup | | supergroup | 设置hdfs超级权限的组，默认是supergroup，启动hadoop所使用的用户通常是superuser。 |
| dfs.data.dir | | /opt/data1/hdfs/data, /opt/data2/hdfs/data, /opt/data3/hdfs/data, … | 真正的datanode数据保存路径，可以写多块硬盘，逗号分隔 |
| dfs.datanode.data.dir.perm | | 755 | datanode所使用的本地文件夹的路径权限，默认755 |
| dfs.replication | | 3 | hdfs数据块的复制份数，默认3，理论上份数越多跑数速度越快，但是需要的存储空间也更多。有钱人可以调5或者6 |
| dfs.replication.max | | 512 | 有时dn临时故障恢复后会导致数据超过默认备份数。复制份数的最多数，通常没什么用，不用写配置文件里。 |
| dfs.replication.min | | 1 | 最小份数，作用同上。 |
| dfs.block.size | | 134217728 | 每个文件块的大小，我们用128M，默认是64M。这个计算需要128\*1024^2，我碰上过有人直接写128000000，十分浪漫。 |
| dfs.df.interval | | 60000 | 磁盘用量统计自动刷新时间，单位是毫秒。 |
| dfs.client.block.write.retries | | 3 | 数据块写入的最多重试次数，在此次数之前不会捕获失败。 |
| dfs.heartbeat.interval | | 3 | DN的心跳检测时间间隔。秒 |
| dfs.namenode.handler.count | | 10 | NN启动后展开的线程数。 |
| dfs.balance.bandwidthPerSec | | 1048576 | 做balance时所使用的每秒最大带宽，使用字节作为单位，而不是bit |
| dfs.hosts | | /opt/hadoop/conf/hosts.allow | 一个主机名列表文件，这里的主机是允许连接NN的，必须写绝对路径，文件内容为空则认为全都可以。 |
| dfs.hosts.exclude | | /opt/hadoop/conf/hosts.deny | 基本原理同上，只不过这里放的是禁止访问NN的主机名称列表。这在从集群中摘除DN会比较有用。 |
| dfs.max.objects | | 0 | dfs最大并发对象数，HDFS中的文件，目录块都会被认为是一个对象。0表示不限制 |
| dfs.replication.interval | | 3 | NN计算复制块的内部间隔时间，通常不需写入配置文件。默认就好 |
| dfs.support.append | | true | false | 新的hadoop支持了文件的APPEND操作，这个就是控制是否允许文件APPEND的，但是默认是false，理由是追加还有bug。 |
| dfs.datanode.failed.volumes.tolerated | | 0 | 能够导致DN挂掉的坏硬盘最大数，默认0就是只要有1个硬盘坏了，DN就会shutdown。 |
| dfs.secondary.http.address | | 0.0.0.0:50090 | SNN的tracker页面监听地址和端口 |
| dfs.datanode.address | | 0.0.0.0:50010 | DN的服务监听端口，端口为0的话会随机监听端口，通过心跳通知NN |
| dfs.datanode.http.address | | 0.0.0.0:50075 | DN的tracker页面监听地址和端口 |
| dfs.datanode.ipc.address | | 0.0.0.0:50020 | DN的IPC监听端口，写0的话监听在随机端口通过心跳传输给NN |
| dfs.datanode.handler.count | | 3 | DN启动的服务线程数 |
| dfs.http.address | | 0.0.0.0:50070 | NN的tracker页面监听地址和端口 |
| dfs.https.enable | | true | false | NN的tracker是否监听在HTTPS协议，默认false |
| dfs.datanode.https.address | | 0.0.0.0:50475 | DN的HTTPS的tracker页面监听地址和端口 |
| dfs.https.address | | 0.0.0.0:50470 | NN的HTTPS的tracker页面监听地址和端口 |
| dfs.datanode.max.xcievers | | 2048 | 相当于linux下的打开文件最大数量，文档中无此参数，当出现DataXceiver报错的时候，需要调大。默认256 |

核心配置：

<configuration>

<property>

<name>dfs.replication</name>

<value>2</value>

<description>设置副本数</description>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///root/BigData/hadoop-3.0.0/hdfs/name</value>

<description>设置存放NameNode的文件路径</description>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///root/BigData/hadoop-3.0.0/hdfs/data</value>

<description>设置存放DataNode的文件路径</description>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

<description>权限还是不要的好</description>

</property>

<property>

<name>dfs.secondary.http.address</name>

<value>master:50070</value>

<description>SNN路径</description>

</property>

# linux|批量创建文件、文件夹或删除——通配符

注意：下面示例中{}中，要写..，而不是…。

一、 批量创建文件

touch file:平时我们都是这样创建一个文件。

如果我们想创建的文件，它的名字都类似:file0.txt,file1.txt … … file9.txt，那我们可不可以用一个命令直接快速创建多个文件？

touch file{0..9}.txt:这条命令便可以实现上面的要求。

二、 批量删除文件

rm -rf file:删除一个文件。

如果我们想把上面批量创建的那些文件全部删除 该如何做呢？

rm -rf file{0..9}

三、 批量创建文件夹

mkdir dir:创建一个文件夹。

如果我们想快速创建名字类似的文件夹该如何做呢? 同理，

mkdir dir{0..9}:这条命令便可以实现上面的要求。

mkdir -p /disk{1..10}/dfs/dn

四、 批量删除文件夹

rmdir dir: 只可以删除一个空文件夹。

rm -rf dir:可以删除一个空、非空文件夹。

如果批量删除上面的生成的文件夹。同理，

rmdir dir{0..9}或者rm -rf dir{0..9}。

重要概念

任何路径的任何地方：通配符可以用于路径的任何部分。

无论何处使用路径：由于通配符替换是由系统完成的，而不是命令。因此可以在使用路径的任何地方使用它们。

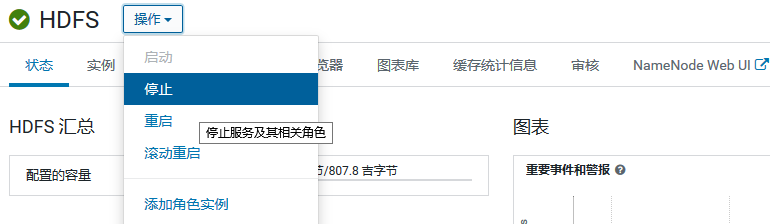
————————————————

版权声明：本文为CSDN博主「瑾兰」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/qq_28296925/article/details/81163373>

# [CDH 更换 HDFS 数据目录](https://www.cnblogs.com/jhxxb/p/11596071.html)

先停止 HDFS 角色。

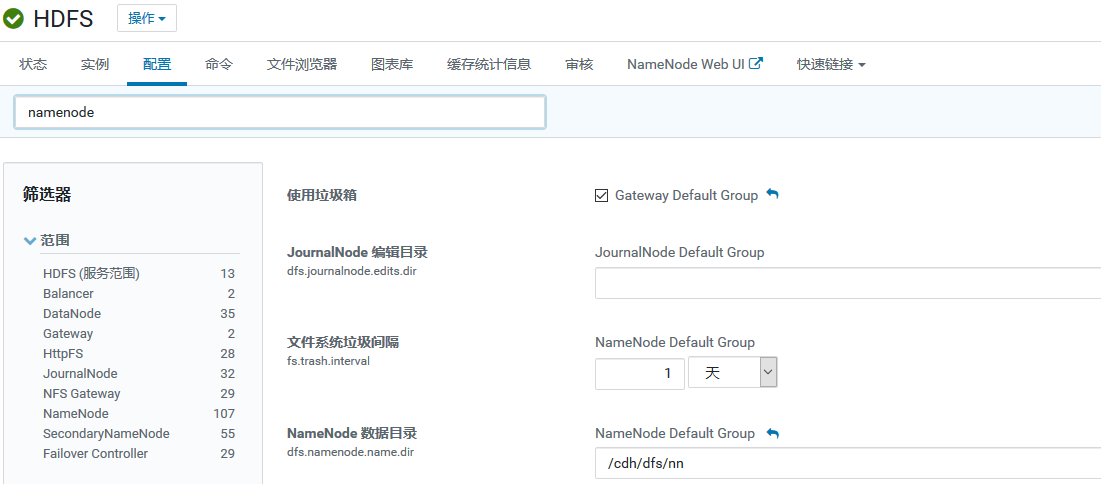


数据文件位置默认在 /dfs/ 中，这里配置 NameNode、SecondaryNameNode、DataNode 数据目录。

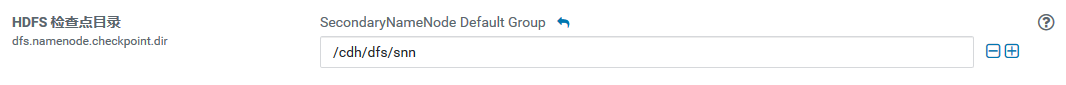
先在所有 HDFS 的主机上把数据拷贝过去，然后再修改 CDH 上的配置。

mkdir /cdh/dfs

cp -af /dfs/\* /cdh/dfs/







再启动 HDFS 角色。

# 执行 fsck 检查是否复制正确

hdfs fsck /

数据库可以远程连接其它数据库：

华为云MySQL连接：mysql -h 119.3.61.59 -P 3306 -u sqoopuser -pZxm10@@@