**15Architecture\_06完全分布式+节点管理+NFS网关**

**一 完全分布式-安装部署**

**1.1 安装与部署**

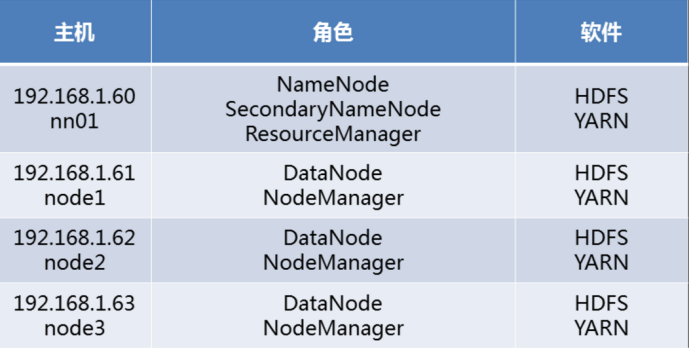
hadoop三大核心组件

分布式文件系统-HDFS已经部署完毕

分布式计算框架-mapreduce(程序员开发时搭建)

集群资源管理-yarn

系统规划



**1.2 mapred部署**

**1.2.1 分布式计算框架配置文件mapred-site.xml**

备份并改名: FROM: mapred-site.xml.template

To: mapred-site.xml

nn01 ~]# cd /usr/local/hadoop/etc/hadoop/

nn01 hadoop]# cp mapred-site.xml.template mapred-site.xml

**1.2.2 资源管理类**

mapreduce.framework.name

**1.2.3 分布式计算框架mapred-site.xml**

**只支持local和yarn两种:单机使用local,集群使用yarn**

nn01 hadoop]# vim mapred-site.xml

19 <configuration>

**20 <property>**

**21 <name>mapreduce.framework.name</name>**

**22 <value>yarn</value>**

**23 </property>**

24 </configuration>

**1.3 yarn部署**

1.3.1 资源管理配置文件yarn-site.xml

1.3.2 resourcemanager地址:yarn.resoucremanager.hostname

1.3.3 nodemanager使用的计算框架:yarn.nodemanager.aux-services

1.3.4 mapreduceshuffle计算框架名称:mapreduce\_shuffle(由开发提供)

1.3.5 资源管理配置文件yarn-site.xml

nn01 ~]# cd /usr/local/hadoop/etc/hadoop/

nn01 hadoop]# cp yarn-site.xml{,.bak}

nn01 hadoop]# vim yarn-site.xml

15 <configuration>

16

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

1**8 <property>**

**19 <name>yarn.resourcemanager.hostname</name>**

**20 <value>nn01</value> #resourcemanager安装位置**

**21 </property>**

**22 <property>**

**23 <name>yarn.nodemanager.aux-services</name>**

**24 <value>mapreduce\_shuffle</value> #框架名找开发**

**25 </property>**

26 </configuration>

**1.4 web访问hadoop**

**1.4.1 同步配置文件到61\62\63上**

nn01 ~]# cd /usr/local/hadoop/etc/hadoop/

nn01 hadoop]# scp yarn-site.xml mapred-site.xml

root@192.168.1.61\62\63:/usr/local/hadoop/etc/hadoop/

**1.4.2 启动服务**

nn01 ~]# /usr/local/hadoop/sbin/start-yarn.sh

# nn01 ~]# /usr/local/hadoop/sbin/stop-yarn.sh #停止服务

**1.4.3 验证服务**

nn01 hadoop]# ./bin/yarn node -list

Total Nodes:3

Node-Id Node-State Node-Http-Address Number-of-Running-Containers

node61:36254 RUNNING node61:8042 0

node62:46305 RUNNING node62:8042 0

node63:44037 RUNNING node63:8042 0

nn01 ~]# jps

30531 SecondaryNameNode

18339 Jps

30340 NameNode

**17993 ResourceManager**

**1.4.1 web访问以下页面**

[http://192.168.1.60:500](http://192.168.1.60:5700)70 #主机nn01上namenode页面

<http://192.168.1.60:50090> #主机nn01上secondarynamenode页面

<http://192.168.1.60:8088> #主机nn01上resourcemanger页面

<http://192.168.1.61:50075> #datenode页面(61\62\63)

<http://192.168.1.61:8042> #nodemanager页面(61\62\63)

core-site.xml中定义的

hdfs://nn01:9000

**二 完全分布式-DHFS基本使用**

**2.1 HDFS使用**

在/usr/local/hadoop/目录下执行的HDFS命令(两个/代表不同的目录)

HDFS基本命令 对应shell命令

./bin/hadoop fs -ls / ls /

./bin/hadoop fs -mkdir /abc mkdir /abc

./bin/hadoop fs -touchz /urfile touch /rufile

上传文件

./bin/hadoop fs -put localfile /remotefile

下载文件[到当前目录下]

./bin/hadoop fs -get /reomotefile

案例:Hadoop词频统计:

在集群文件系统里创建文件夹,上传要分析的文件到目录中,分析上传文件,展示结果

查看集群文件系统的根有没有内容

nn01 hadoop]# ./bin/hadoop fs -ls /

在集群文件系统下创建input目录

nn01 hadoop]# ./bin/hadoop fs -mkdir /input

再次查看，有刚创建的input目录

nn01 hadoop]# ./bin/hadoop fs -ls /

上传文件到集群文件系统下的input目录

nn01 hadoop]# ./bin/hadoop fs -put /etc/passwd /input

数据分析input里的passwd文件

nn01 hadoop]# ./bin/hadoop jar \

> share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar

> wordcount /input /output

nn01 hadoop]# ./bin/hadoop fs -ls /output

nn01 hadoop]# ./bin/hadoop fs -cat /output/part-r-00000

**三 节点管理-HDFS节点管理**

**3.1 增加节点**

**3.1.1 创建节点,设置IP与主机名,设置SSH免密登录**

room9pc01 ~]# base-vm node64 (node64 192.168.1.64)

nn01 ~]# ssh-copy-id 192.168.1.64

**3.1.2 在所有节点修改/etc/hosts文件,增加新节点的主机信息**

nn01 ~]# vim /etc/hosts

192.168.1.60 nn01

192.168.1.61 node61

192.168.1.62 node62

192.168.1.63 node63

192.168.1.64 node64

nn01 ~]# scp /etc/hosts [root@192.168.1.61\62\63\64:/etc/hosts](mailto:root@192.168.1.61/62/63/64:/etc/hosts)

**3.1.3 安装Java运行环境**

node64 ~]# yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel

**3.1.4 修改namenode的slaves文件增加该节点**

nn01 ~]# vim /usr/local/hadoop/etc/hadoop/slaves

node61

node62

node63

node64

nn01 ~]# scp /usr/local/hadoop/etc/hadoop/slaves

root@192.168.1.61\62\63:/usr/local/hadoop/etc/hadoop/slaves

**3.1.5 拷贝namenode的/usr/local/hadoop目录到node64主机**

nn01 ~]# scp -r /usr/local/hadoop/ [root@192.168.1.64:/usr/local/](mailto:root@192.168.1.64:/usr/local/)

**3.1.6 在node64启动datanode**

node64 ~]# cd /usr/local/hadoop/

node64 hadoop]# ./sbin/hadoop-daemon.sh start datanode

**3.1.7 验证**

node64 hadoop]# jps

816 DataNode

901 Jps

nn01 ~]# /usr/local/hadoop/bin/hdfs dfsadmin -report

Name: 192.168.1.64:50010 (node64)

**3.1.8 设置同步带宽,并同步数据**

# nn01 ~]# /usr/local/hadoop/bin/hdfs dfsadmin #查看带宽设置选项

# [-setBalancerBandwidth <bandwidth in bytes per second>]

nn01 ~]# /usr/local/hadoop/bin/hdfs dfsadmin \

-setBalancerBandwidth 50000000 #单位为byte

nn01 ~]# /usr/local/hadoop/sbin/start-balancer.sh #同步数据

**3.1.9 查看集群状态**

nn01 ~]# /usr/local/hadoop/bin/hdfs dfsadmin -report

**3.2 修复节点**

HDFS修复节点

修复节点比较简单,与增加节点基本一致

注意:新节点的IP和主机名,要与损坏节点主机的IP和主机名一致

启动服务(namenode上执行此命令):

/usr/local/hadoop/sbin/hadoop-daemon.sh start datanode

数据恢复是自动的

上线以后会自动恢复数据,如果数据量非常巨大,可以需要一定的时间

**3.3 删除节点**

HDFS删除节点

**3.3.1 配置namenode的hdfs-site.xml**

增加dfs.hosts.exclude配置(排除主机列表)

nn01 ~]# cd /usr/local/hadoop/

nn01 hadoop]# vim etc/hadoop/hdfs-site.xml #修改后不需要重启

<configuration>

<property>

<name>dfs.namenode.http-address</name>

<value>nn01:50070</value>

</property>

<property>

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

<value>nn01:50090</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

**<property> #添加此4行**

**<name>dfs.hosts.exclude</name> #排除主机列表及路径**

**<value>/usr/local/hadoop/etc/hadoop/exclude</value>**

**</property>**

</configuration>

**3.3.2 增加exclude配置文件,写入要删除的节点主机名**

nn01 hadoop]# vim etc/hadoop/exclude

node64

**3.3.3 更新数据**

nn01 hadoop]# ./bin/hdfs dfsadmin -refreshNodes

Refresh nodes successful

**3.3.4 HDFS删除节点状态-查看状态**

nn01 hadoop]# ./bin/hdfs dfsadmin -report

Live datanodes (4):

Name: 192.168.1.64:50010 (node64)

Hostname: node64

Decommission Status : Decommissioned

**normal: 正常状态**

**decommissioned in program: 数据正在迁移**

**decommissioned: 数据迁移完成**

**注意: 只有状态变成decommissioned才能down机下线**

node64 ~]# poweroff

**四 节点管理-yarn节点管理**

**4.1 yarn的相关操作**

由于hadoop在2.X引入了yarn框架,对于计算节点的操作已经变得非常简单

增加节点:

nn01 ~]# cd /usr/local/hadoop/

nn01 hadoop]# sbin/yarn-daemon.sh start nodemanager

删除节点:

nn01 hadoop]# sbin/yarn-daemon.sh stop nodemanager

查看节点:(resourcemanager上执行)

nn01 hadoop]# ./bin/yarn node -list

**五 NFS网关-NFS配置**

**5.1 NFS网关**

**5.1.1 用途**

用户可以通过操作系统兼容的本地NFSv3客户端来浏览HDFS文件系统

用户可以从HDFS文件系统下载文档到本地文件系统

用户可以通过挂载点直接流化数据,支持文件附加,但是不支持随机写

NFS网关支持NFSv3和允许HDFS作为客户端文件系统的一部分被挂载

**5.1.2 特性**

HDFS超级用户是与nomenode进程本身具有相同标识的用户,超级用户可以执行任何操作,因为权限检查永远不会认为超级用户失败

**5.1.3 注意事项**

在非安全模式下,运行网关进程的用户是代理用户

在安全模式下,kerberos keytab中的用户是代理用户

**5.2 配置用户**

**5.2.1 配置代理用户**

在namenode和NFSGW上添加代理用户

代理用户的UID,GID,用户名必须完全相同

如果因特殊原因客户端的用户和NFS网关的用户UID不能保持一直,需要我们配置nfs.map的静态映射关系,配置如下

nfs.map

uid 10 100 # Map the remote UID 10 the local UID 100

gid 11 101 # Map the remote GID 11 to the local GID 101

**5.2.2 配置core-site.xml**

核心配置core-site.xml

hadoop.proxyuser.{代理用户}.groups

hadoop.proxyuser.{代理用户}.hosts

这里的{代理用户}是主机上真实运行的nfs3的用户

在非安全模式下,运行nfs网关的用户为代理用户

groups为挂载点用户所使用的组

hosts为挂载点主机地址

**步骤:(不使用node64主机)**

**创建虚拟主机**

room9pc01 ~]# base-vm nfsgw65 (nfsgw65 192.168.1.65)

**设置/etc/hosts文件并传输给所有主机**

nfsgw65 ~]# vim /etc/hosts

192.168.1.60 nn01

192.168.1.61 node61

192.168.1.62 node62

192.168.1.63 node63

192.168.1.64 node64

192.168.1.65 nfsgw65

nfsgw65 ~]# scp :/etc/hosts [root@192.168.1.61\62\63\64:/etc/hosts](mailto:root@192.168.1.61/62/63/64:/etc/hosts)

**安装依赖软件**

nfsgw65 ~]# yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel

**配置代理用户**

nn01 ~]# groupadd -g 800 nfsuser

nn01 ~]# useradd -g 800 -u 800 -r -d /var/hadoop nfsuser

nfsgw65 ~]# groupadd -g 800 nfsuser

nfsgw65 ~]# useradd -g 800 -u 800 -r -d /var/hadoop nfsuser

**停止所有集群服务**

nn01 ~]# cd /usr/local/hadoop/

nn01 hadoop]# ./sbin/stop-all.sh

**同步配置文件到所有主机**

slaves文件去掉node64,exclude文件清空,并传给61\62\63\64

**修改core-site.xml文件比传给所有主机**

nn01 hadoop]# vim etc/hadoop/core-site.xml

19 <configuration>

20 <property>

21 <name>fs.defaultFS</name>

22 <value>**hdfs://nn01:9000**</value>

23 </property>

24 <property>

25 <name>hadoop.tmp.dir</name>

26 <value>/var/hadoop</value>

27 </property>

**28 <property>**

**29 <name>hadoop.proxyuser.nfsuser.groups</name>**

**30 <value>\*</value>**

**31 </property>**

**32 <property>**

**33 <name>hadoop.proxyuser.nfsuser.hosts</name>**

**34 <value>\*</value>**

**35 </property>**

36 </configuration>

nn01 hadoop]# scp etc/hadoop/core-site.xml

root@192.168.1.61\62\63:/usr/local/hadoop/etc/hadoop/core-site.xml

**启动hdfs**

nn01 hadoop]# ./sbin/start-dfs.sh

**验证**

nn01 hadoop]# ./bin/hdfs dfsadmin -report

Live datanodes (3):

Name: 192.168.1.63:50010 (node63)

Name: 192.168.1.61:50010 (node61)

Name: 192.168.1.62:50010 (node62)

**六 NFS网关-NFSGW配置**

**6.1 配置步骤**

**6.1.1 启动一个新的系统,卸载rpcbing hfs-utils**(原本就没有安装)

nfsgw65 ~]# rpm -qa | grep -i rpcbind

nfsgw65 ~]# rpm -qa | grep -i nfs-utils

**6.1.2 配置/etc/hosts,添加所有namenode和datanode的主机名与IP对应关系**(已完成)

**6.1.3 安装Java运行环境(Java-1.8.0-openjdk-devel)**(已完成)

**6.1.4 同步namenode的/usr/local/hadoop到本机**

nn01 ~]# scp -r /usr/local/hadoop/ [root@192.168.1.65:/usr/local/](mailto:root@192.168.1.65:/usr/local/)

**6.1.5 配置hdfs-site.xml**

配置文件hdfs-site.xml

**nfs.exports.allowed.hosts**

默认情况下,export可以被任何客户端挂载.为了更好的控制访问,可以设置属性.值和字符串对应集群名和访问策略,通过空格来分隔.机器名的格式可以是单一的主机\Java的正则表达式或者ipv4地址

使用rw或ro可以指定导出目录的读写或只读权限

如果访问策略没被提供,默认为只读,每个条目使用分号;分隔

**nfs.dump.dir**

用户需要更新文件转储目录参数.NFS客户端经常重新安排写操作,顺序的写操作会随机到达NFS网关.这个目录常用于临时存储无序的写操作.对于每个文件,无序的写操作会在他们积累在内存中超过一定阀值(如1M)时被转储.需要确保有足够的空间的目录

nfsgw65 ~]# vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml

19 <configuration>

20 <property>

21 <name>dfs.namenode.http-address</name>

22 <value>nn01:50070</value>

23 </property>

24 <property>

25 <name>dfs.namenode.secondary.http-address</name>

26 <value>nn01:50090</value>

27 </property>

28 <property>

29 <name>dfs.replication</name>

30 <value>2</value>

31 </property>

32 <property>

33 <name>dfs.hosts.exclude</name>

34 <value>/usr/local/hadoop/etc/hadoop/exclude</value>

35 </property>

**36 <property>**

**37 <name>nfs.exports.allowed.hosts</name>**

**38 <value>\* rw</value>**

**39 </property>**

**40 <property>**

**41 <name>nfs.dump.dir</name>**

**42 <value>/var/nfstmp</value>**

**43 </property>**

44 </configuration>

**6.1.6 启动服务**

nfsgw65 ~]# mkdir /var/nfstmp

nfsgw65 ~]# chown nfsuser:nsfuser /var/nfstmp

nfsgw65 ~]# setfacl -m u:nfsuser:rwx /usr/local/hadoop/logs/

**使用root用户启动portmap服务(portmap相较于nfs3先启动后关闭)**

nfsgw65 ~]# cd /usr/local/hadoop/

nfsgw65 hadoop]# ./sbin/hadoop-daemon.sh --script ./bin/hdfs start portmap

nfsgw65 hadoop]# jps

22977 Jps

22918 Portmap

**使用代理用户启动nfs3服务(nfs3相较于portmap后启动先关闭)**

nfsgw65 hadoop]# sudo -u nfsuser ./sbin/hadoop-daemon.sh \

--script ./bin/hdfs start nfs3

nfsgw65 hadoop]# jps

23008 Nfs3

23090 Jps

22918 Portmap

注意: 启动portmap需要使用root用户

启动nfs3需要使用core-site里面设置的代理用户

必须先启动portmap再启动nfs3,在重启之后nfs3也需要重启

停止时必须先停止nfs3,再停止portmap

**6.2 NFS启动与挂载**

目前NFS只能使用v3版本 vers=3

仅使用TCP作为传输协议 proto=tcp

不支持NLM nolock

禁用access time的时间更新 noatime

禁用acl扩展权限 noacl

强烈建议使用安装选项sync,它可以最小化避免重排序写入造成不可预测的吞吐量,未指定同步选项可能会导致上传大文件时出现不可靠行为

启动一台机器并安装nfs-utils

node64 ~]# yum -y install nfs-utils

node64 ~]# mount -t nfs -o \

vers=3,proto=tcp,noatime,nolock,sync,noacl 192.168.1.65:/ /mnt/

node64 ~]# df -h

192.168.1.65:/ 90G 4.4G 86G 5% /mnt

node64 ~]# ls /mnt/

abc input output system tmp urfile

**补充: stat 文件名 查看文件属性**

nfsgw65 hadoop]# stat LICENSE.txt

文件："LICENSE.txt"

大小：86424 块：176 IO 块：4096 普通文件

设备：fd01h/64769d Inode：55574637 硬链接：1

权限：(0644/-rw-r--r--) Uid：( 0/ root) Gid：( 0/ root)

最近访问：2019-10-18 15:17:55.177678889 +0800

最近更改：2019-10-18 15:17:55.178678886 +0800 #内容修改时间

最近改动：2019-10-18 15:17:55.178678886 +0800 #属性修改时间

创建时间：-