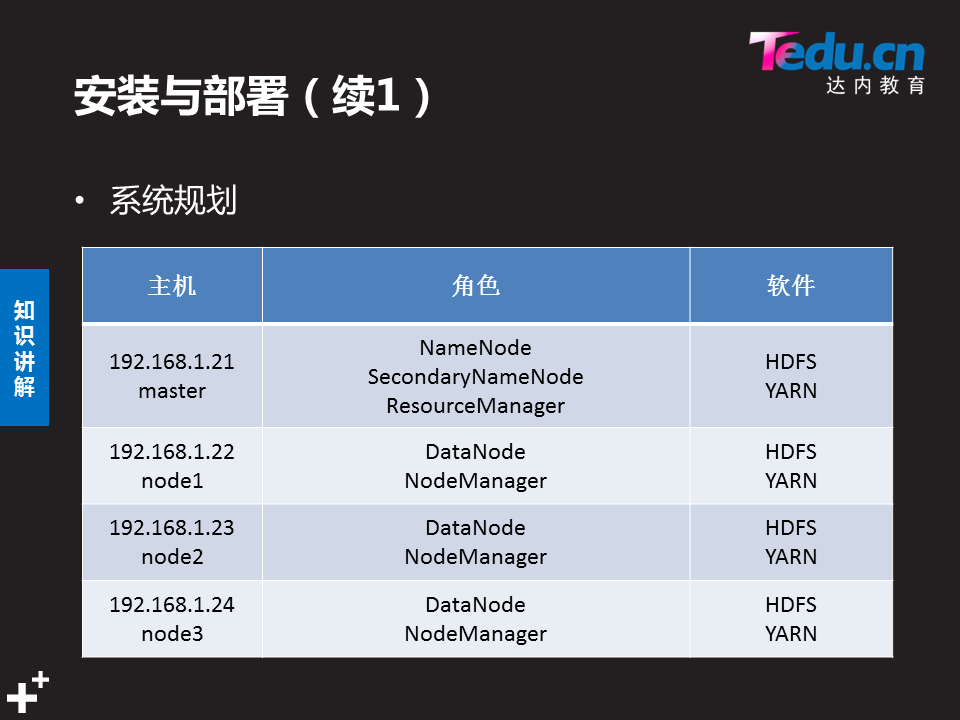
**Morning：**

1. [续] 完全分布式Hadoop

**步骤1：系统规划**



划分原则：不争抢硬件资源。

DataNode读写数据，占用I/O资源

NodeManager负责计算，占用内存、CPU

NameNode计算/记录数据映射，占用少量CPU、少量I/O、中等量内存

Secondary NameNode更新fsimage，占用少量CPU、少量I/O

ResourceManager负责切分文件，占用少量I/O、少量CPU、少量内存

**步骤2：mapred部署**

[root@nn01 hadoop]# mv mapred-site.xml.template \

> mapred-site.xml

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

#分布式计算框架设置为集群

</configuration>

**步骤3：yarn部署**

[root@nn01 hadoop]# vim yarn-site.xml

<configuration>

<property>

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

<value>nn01</value>

</property>

#指定resourcemanager的地址

<property>

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

<value>mapreduce\_shuffle</value>

</property>

#指定nodemanager的计算框架名称

</configuration>

**步骤4：同步配置文件**

[root@nn01 hadoop]# for i in node{1..3}

> do

> rsync -avz --delete /usr/local/hadoop/etc ${i}:/usr/local/hadoop/

> done

**步骤5：启动并验证服务**

[root@nn01 hadoop]# ./sbin/start-yarn.sh

[root@xxx ]# jps

#验证jps节点信息

[root@nn01 hadoop]# ./bin/yarn node -list

node2:46256 RUNNING node2:8042

node3:46267 RUNNING node3:8042

node1:35441 RUNNING node1:8042

#验证yarn节点信息

**步骤6：Web访问Hadoop**

http://192.168.1.10:50070/ #NameNode页面

http://192.168.1.10:50090/ #Secondary NameNode页面

http://192.168.1.11:50075/ #DataNode页面

http://192.168.1.10:8088/ #Resourcemanger页面

http://192.168.1.11:8042/ #Nodemanager页面

1. HDFS基本使用

**-HDFS基本命令**

|  |  |
| --- | --- |
| HDFS命令 | 备注 |
| ./bin/hadoop fs -ls / | ls |
| ./bin/hadoop fs -mkdir /abc | mkdir |
| ./bin/hadoop fs -rmdir /abc | rmdir |
| ./bin/hadoop fs -touchz /1.txt | touch |
| ./bin/hadoop fs -cat /1.txt | cat |
| ./bin/hadoop fs -rm /1.txt | rm |
| ./bin/hadoop fs -put localfile /remotefile | 上传文件 |
| ./bin/hadoop fs -get /remotefile | 下载文件 |

Tips：

1）./bin/hadoop fs 敲回车可以查看命令帮助

2）http://nn01:50070 -- Utilities --Browse the file system查看目录

3）”/”所指的根，取决于core-site.xml里配置的默认文件系统路径

**-Hadoop的使用**

流程： 开发jar包 -> 在集群文件系统创建目录 -> 上传要分析的文件

-> 使用jar包分析文件 -> 查看结果

[root@nn01 hadoop]# ./bin/hadoop fs -put \*.txt /abc

#上传要分析的文件

[root@nn01 hadoop]# ./bin/hadoop jar \

> ./share/hadoop/mapreduce/hadoop-mapreduce-ex..-2.7.6.jar \

> wordcount /abc/\* /output

#使用jar包分析，结果存到集群文件系统的/output中

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

#查看分析结果，也可以通过http://nn01:50070下载结果

**Afternoon：**

1. 节点管理

**-HDFS.增加节点**

1） 新增虚拟机(newnode)：2G内存、2CPU、20G硬盘，IP 192.168.1.15

2）设置SSH免密登陆

3）所有节点修改/etc/hosts

4）安装java-1.8.0-openjdk-devel

5）拷贝nn01的/usr/local/hadoop

6）修改nn01的slaves文件增加节点

7）newnode启动DataNode

[root@newnode hadoop]# ./sbin/hadoop-daemon.sh start datanode

8）设置同步带宽，并同步数据

[root@nn01 hadoop]#./bin/hdfs dfsadmin -setBalancerBandwidth \

> 60000000

#设置为千兆带宽的一半，以免影响正常业务

[root@nn01 hadoop]# ./sbin/start-balancer.sh

#启动数据同步

1. 查看集群状态

[root@nn01 hadoop]# ./bin/hdfs dfsadmin -report

1. 验证集群

[root@nn01 ~]# /usr/local/hadoop/bin/hadoop fs -put \

> /root/CentOS-7-x86\_64-DVD-1708.iso /c.iso

[root@nn01 hadoop]# ./bin/hdfs dfsadmin -report

#上传大文件到集群，查看Used DFS

**-HDFS.修复节点**

#必须保证用于修复的新主机与原来下线节点的主机名/IP一致

1. 设置ssh免密登陆
2. 安装java-1.8.0-openjdk-devel
3. 拷贝/etc/hosts
4. 拷贝/usr/local/hadoop
5. mkdir /tmp/hadoop
6. 启动DataNode(会自动恢复数据)

**-HDFS.删除节点**

1. 配置nn01的hdfs-site.xml

[root@nn01 hadoop]# vim ./etc/hadoop/hdfs-site.xml

... ...

<property>

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

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

</property>

1. 配置exclude文件

[root@nn01 hadoop]# vim /usr/local/hadoop/etc/hadoop/exclude

newnode #填写要删除的节点

3）更新数据

[root@nn01 hadoop]# ./bin/hdfs dfsadmin -refreshNodes

1. 查看节点状态

[root@nn01 hadoop]# ./bin/hdfs dfsadmin -report

Normal：正常

Decommissioned in Program：数据正在迁移

Decommissioned：数据迁移完成

5）下线节点

[root@newnode hadoop]# ./sbin/hadoop-daemon.sh stop datanode

[root@newnode hadoop]# jps

1. 查看下线后节点数量

[root@newnode hadoop]# ./bin/hdfs dfsadmin -refreshNodes

[root@nn01 hadoop]# ./bin/hdfs dfsadmin -report

**-Yarn节点管理**

1. 增加节点

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

1. 删除节点

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

1. 查看节点

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

2、NFS配置

**-NFS常见架构**

ceph的弊端：同时挂载时，数据不会实时同步，需要重新挂载

nfs + rsync + inotify #支持文件量较小(<5G,数量<18个)

nfs + drbd + headbeat #<100G

hdfs + nfsgw #可无限扩展

**179 min**

**-NameNode配置步骤**

1. /etc/hosts添加nfsgw，并同步(包括nfsgw)

2）停止hadoop集群

[root@nn01 hadoop]# ./sbin/stop-all.sh

3）清除集群信息

[root@nn01 hadoop]# rm -rf \

> logs/\* ./etc/hadoop/slaves ./etc/hadoop/exclude

1. 配置代理用户(NameNode、nfsgw)

要求代理用户的UID、GID、用户名完全相同

[root@nn01/nfsgw ~]# groupadd -g 700 nsd1808

[root@nn01/nfsgw ~]# useradd -u 700 -g 700 -r nsd1808

1. nn01修改核心配置core-site.xml

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

... ...

<property>

<name>hadoop.proxyuser.nsd1808.groups</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.nsd1808.hosts</name>

<value>\*</value>

</property>

#代理用户授权

6）同步hadoop配置(包括nfsgw)

[root@nn01 ~]# for i in node{1..3} newnode nfsgw; do

> rsync -avz --delete /usr/local/hadoop $i:/usr/local/

> done

1. nn01启动hdfs

[root@nn01 hadoop]# ./sbin/start-dfs.sh

**-NFSGW配置步骤**

1. 卸载rpcbind、nfs-utils
2. 安装java-1.8.0-openjdk
3. 配置文件hdfs-site.xml

[root@nfsgw hadoop]# vim ./etc/hadoop/hdfs-site.xml

... ...

<property>

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

<value>\* rw</value>

</property>

#指定所有用户的权限为rw（用户名可以是主机名、IPv4地址、正则）

<property>

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

<value>/var/nfstmp</value>

</property>

#指定临时存储空间，用于临时存储无序的写操作

1. 创建临时存储空间，修改属组为代理用户

[root@nfsgw ~]# mkdir /var/nfstmp

[root@nfsgw ~]# chown 700:700 /var/nfstmp

4）设置hadoop/logs权限，使代理用户可读写日志

[root@nfsgw ~]# setfacl -m user:nsd1808:rwx \

> /usr/local/hadoop/logs

1. 先使用root用户启动portmap服务

[root@nfsgw hadoop]# ./sbin/hadoop-daemon.sh --script \

> ./bin/hdfs start portmap

1. 再使用代理用户启动nfs3

[root@nfsgw ~]# su -l nsd1808

-bash-4.2$ /usr/local/hadoop/sbin/hadoop-daemon.sh --script \

> ./bin/hdfs start nfs3

[root@nfsgw ~]# jps

注意事项：

1. nfs.dump.dir临时存储空间的所有者必须为代理用户
2. hadoop/logs 代理用户必须有rwx权限
3. 先root启动portmap、再代理用户启动nfs3
4. 卸载rpcbind，避免与portmap冲突

**-客户端挂载**

[root@cli ~]# yum -y install nfs-utils

[root@cli ~]# mount -t nfs -o vers=3,proto=tcp,noatime,nolock,sync \

> nfsgw:/ /mnt/

#NFS只能使用v3版本(vers=3)

#只能使用TCP传输协议(proto=tcp)

#不支持NLM(nolock)

#禁用access time更新，提高性能(noatime)

#最小化避免重排序写入造成不可预测的吞吐量(sync)

3、Hadoop命令整理

./bin/hadoop version / jar

#查看hadoop版本 / 执行jar包

./bin/hdfs namenode -format / dfsadmin -report

#hdfs格式化 / 验证节点

./bin/yarn node -list

#查看yarn节点信息

./bin/hadoop fs -ls /

#使用hadoop命令行

./bin/hdfs dfsadmin -setBalancerBandwidth 6000000

#增加hdfs节点-设置数据同步带宽

./bin/hdfs dfsadmin -refreshNodes

#下线hdfs节点-更新数据

./sbin/start-dfs.sh

#启动hdfs集群

./sbin/start-yarn.sh

#启动yarn集群

./sbin/hadoop-daemon.sh start datanode / stop datanode

#hdfs节点上线/下线

./sbin/yarn-daemon.sh start nodemanager / stop nodemanager

#yarn节点增加/删除

./sbin/start-balancer.sh

#增加hdfs节点-启动数据同步

./sbin/stop-all.sh

#停止hadoop集群

问题：如何把下线节点重新上线？