**Hadoop配置手记**

# 硬件环境

|  |  |  |
| --- | --- | --- |
| **主机名** | **IP地址** | **操作系统** |
| Cloud-master | 202.38.64.164 | Ubuntu 9.04 Server |
| Cloud-slave1 | 210.45.64.36 | Ubuntu 9.04 Server |
| Cloud-slave2 | 210.45.64.102 | Ubuntu 9.04 Server |
| Cloud-slave3 | 210.45.64.87 | Ubuntu 9.04 Server |
| Cloud-slave4 | 210.45.64.134 | Ubuntu 9.04 Server |
| Cloud-slave5 | 210.45.64.185 | Ubuntu 9.04 Server |
| Cloud-slave6 | 202.38.64.203 | Ubuntu 9.04 Server |
| Cloud-slave7 | 202.38.64.195 | Ubuntu 9.04 Server |
| Cloud-slave8 | 210.45.64.60 | Ubuntu 9.04 Server |

注：202.38.64.164作namenode和jobtracker

210.45.64.\* 作datanode和tasktrakcer

# 配置/etc/hosts

1、在cloud-master机器上配置/etc/hosts

添加：202.38.64.164 cloud-master

210.45.64.36 cloud-slave1

210.45.64.102 cloud-slave2

210.45.64.87 cloud-slave3

210.45.64.134 cloud-slave4

2、在cloud-slave机器上配置/etc/hosts

Slave1机器添加：202.38.64.164 cloud-master

210.45.64.36 cloud-slave1

Slave2机器添加：202.38.64.164 cloud-master

210.45.64.102 cloud-slave2

Slave3机器添加：202.38.64.164 cloud-master

210.45.64.87 cloud-slave3

Slave4机器添加：202.38.64.164 cloud-master

210.45.64.134 cloud-slave4

# SSH配置

在cloud-master机器上的配置：

[cloud@cloud-master]$ssh-keygen -t rsa

这个命令将为cloud-master上的用户cloud生成其密钥对，询问其保存路径时直接回车采用默认路径，当提示要为生成的密钥输入passphrase的时候，直接回车，也就是将其设定为空密码。生成的密钥对id\_rsa，id\_rsa.pub，默认存储在/home/cloud/.ssh目录下。然后将id\_rsa.pub的内容复制到每个机器(也包括本机)的/home/cloud/.ssh/authorized\_keys文件中。  
[cloud@cloud-master:.ssh]$cp id\_rsa.pub authorized\_keys

在cloud-slave机器上的配置：

[cloud@cloud-slave2:~]$mkdir .ssh

[cloud@cloud-master:.ssh]$scp authorized\_keys cloud-slave2:/home/cloud/.ssh/  
此处的scp就是通过ssh进行远程copy，此处需要输入远程主机的密码，即cloud-slave2机器上cloud帐户的密码。

[cloud@cloud-slave2:~/.ssh]$chmod 644 authorized\_keys

# Hadoop环境变量配置

修改/home/cloud/hadoop-0.20.1/conf/中的hadoop-env.sh文件

设置export JAVA\_HOME=/usr/lib/jvm/java-6-sun

修改/etc/profile文件

添加

export JAVA\_HOME=/usr/lib/jvm/java-6-sun

export CLASSPATH=$CLASSPATH:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

export PATH=$JAVA\_HOME/lib:$JAVA\_HOME/jre/bin:$PATH:$HOME/bin

export HADOOP\_HOME=/home/cloud/hadoop-0.20.1

export PATH=$PATH:$HADOOP\_HOME/bin

# Hadoop配置文件

修改/home/cloud/hadoop-0.20.1/conf/中的masters文件

添加 cloud-master

修改/home/cloud/hadoop-0.20.1/conf/中的slaves文件

添加 cloud-slave1

cloud-slave2

cloud-slave3

cloud-slave4

修改/home/cloud/hadoop-0.20.1/conf/中的core-site.xml文件

添加

<configuration>  
<property>  
   <name>fs.default.name</name>  
   <value>hdfs://cloud-master:9000</value>  
</property>  
</configuration>

修改/home/cloud/hadoop-0.20.1/conf/中的hdfs-site.xml文件

添加

<configuration>

<property>

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

<value>/home/cloud/hadoop-0.20.1/namestore</value>

</property>

<property>

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

<value>/home/cloud/hadoop-0.20.1/datastore</value>

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

<description>Default block replication.

The actual number of replications can be specified when the file is created.

The default is used if replication is not specified in create time.

</description>

</property>

</configuration>

修改/home/cloud/hadoop-0.20.1/conf/中的mapred-site.xml文件

添加

<configuration>

<property>

<name>mapred.job.tracker</name>

<value> hdfs://cloud-master:9001</value>

</property>

</configuration>

# 部署Hadoop

将cloud-master机器上的配置文件部署到其他机器上，保证目录结构一致

$scp –r /home/cloud/hadoop-0.20.1 cloud-slave1:/home/cloud/

$scp –r /home/cloud/hadoop-0.20.1 cloud-slave2:/home/cloud/

$scp –r /home/cloud/hadoop-0.20.1 cloud-slave3:/home/cloud/

$scp –r /home/cloud/hadoop-0.20.1 cloud-slave4:/home/cloud/

# 启动Hadoop

Format a new distributed filesystem:

$ bin/hadoop namenode -format

Start the HDFS with the following command, run on the designated NameNode:

$ bin/start-dfs.sh

Start Map-Reduce with the following command, run on the designated JobTracker:

$ bin/start-mapred.sh

通过web界面查看hdfs运行状态：

http://namenode ip:50070/dfshealth.jsp

通过web界面查看map-reduce信息：

http://jobtracker ip:50030/jobtracker.jsp

直接命令行查看：

$bin/hadoop dfsadmin -report

# 停止Hadoop

Stop HDFS with the following command, run on the designated NameNode:

$ bin/stop-dfs.sh

Stop Map/Reduce with the following command, run on the designated the designated JobTracker:  
$ bin/stop-mapred.sh

# 停止Datanode

比如我原来有10个节点的集群. 现在我想停掉2个,但数据不能丢失,只能让它们的数据转移到其它8台上.这道过程就是decommission. 我们不能直接把那2台停掉, 要在停掉之前把数据转移出去.  
  
首先在namenode机器上建立一个excludes文件.它是一个文本, 里面每行就是想要停掉的主机名.这里excludes 放在 $HADOOP\_HOME下   
例如  
CS02  
CS03  
CS04  
CS05  
CS06  
CS07  
CS08  
CS09  
CS10   
这样就表明要把9台机器给停了.

然后在conf/hadoop-site.xml添加这样的属性:  
  <property>                                                                                                                          
        <name>dfs.hosts.exclude</name>                                                                                                
        <value>excludes</ value>                                                                                                       
  </property>     
  
做完这步, 用bin/hadoop dfsadmin -refreshNodes命令更新结点以及hadoop配置   
  
然后你可以通过 bin/hadoop dfsadmin -report 就可以查看到类似这样的信息:  
Name: 192.168.0.86:50010  
State          : Decommission in progress  
Total raw bytes: 30452150272 (28.36 GB)  
Remaining raw bytes: 17727606637(16.51 GB)  
Used raw bytes: 6480990208 (6.04 GB)  
% used: 21.28%  
Last contact: Thu Nov 27 02:25:40 CST 2008  
  
Name: 192.168.0.89:50010  
State          : Decommissioned  
Total raw bytes: 0 (0 KB)  
Remaining raw bytes: 0(0 KB)  
Used raw bytes: 4703211520 (4.38 GB)  
% used: ∞%  
Last contact: Thu Jan 01 08:00:00 CST 1970   
  
可以看到86在停止中, 89已经停止了.

# 增加Datanode

如何在集群中增加机器，而不需要重启集群   
操作如下：   
首先，把新节点的IP或主机名 加入主节点（master）的 conf/slaves 文件。   
然后登录新的从节点，执行以下命令：   
$ cd /home/cloud/hadoop-0.20.0

$ bin/hadoop-daemon.sh start datanode

$ bin/hadoop-daemon.sh start tasktracker

然后就可以在namenode机器上运行balancer，执行负载均衡   
$bin/hadoop balancer

# HDFS-over-ftp配置

运行于HDFS之上的ftp服务器，它允许用户使用任何FTP客户端连接到HDFS。使用运行于HDFS之上的FTP服务器需要配置hdfs-over-ftp.conf和user.conf文件，它允许使用SSL安全连接并支持所有的HDFS权限。

配置如下：

1、下载hdfs-over-ftp-0.20.0.tar.gz软件包并解压得到hdfs-over-ftp

2、修改hdfs-over-ftp.conf文件，设定需要开放的ftp server端口、ssl编码的端口以及hdfs的位置，我的配置内容如下：

#uncomment this to run ftp server

port = 21

data-ports = 20

#uncomment this to run ssl ftp server

ssl-port = 990

ssl-data-ports = 989

# hdfs uri

hdfs-uri = hdfs://cloud-master:9000

# max number of login

max-logins = 1000

# max number of anonymous login

max-anon-logins = 1000

# have to be a user which runs HDFS

# this allows you to start ftp server as a root to use 21 port

# and use hdfs as a superuser

superuser = cloud

3、修改hdfs-over-ftp.sh文件中的JAVA\_HOME路径，比如我的配置：

JAVA\_HOME=/usr/lib/jvm/java-6-sun

4、修改log4j.conf文件中的log4j.appender.R.File路径，比如我的配置：

log4j.appender.R.File=/home/cloud/hadoop-0.20.0/logs/ftp/ftpd.log

5、修改register-user.sh文件中的JAVA\_HOME路径，比如我的配置：

JAVA\_HOME=/usr/lib/jvm/java-6-sun

6、修改users.conf文件，添加合法的ftp使用者。每一个使用者在users.conf文件中的语法格式如下：

ftpserver.user.[username].userpassword=[md5 hash]

ftpserver.user.[username].homedirectory=/

ftpserver.user.[username].enableflag=true

ftpserver.user.[username].writepermission=true

ftpserver.user.[username].maxloginnumber=0

ftpserver.user.[username].maxloginperip=0

ftpserver.user.[username].idletime=0

ftpserver.user.[username].uploadrate=0

ftpserver.user.[username].downloadrate=0

ftpserver.user.[username].groups=[username],users

使用：

$ ./register-user.sh username password >> users.conf添加用户，然后按照上述语法格式修改该用户的信息。

7、启动或者停止ftp服务器

使用命令：

$ sudo ./hdfs-over-ftp.sh start

$ sudo ./hdfs-over-ftp.sh stop

8、在linux系统下使用curlftpfs挂载ftp到本地目录

命令：

$mkdir /home/cloud/temp（创建本地目录）

$sudo apt-get install curlftpfs（安装curlftpfs）

$cd hdfs-over-ftp

$ sudo ./hdfs-over-ftp.sh start

$sudo curlftpfs –o allow\_other <ftp://user:pass@localhost:21> /home/cloud/temp（挂载ftp本地目录，localhost可以换成机器名）

如果不再使用ftp服务了，则将已经挂载的本地目录卸载即可，命令如下：

$ sudo ./hdfs-over-ftp.sh stop

$sudo umount /home/cloud/temp

9、启动任意一个ftp client连接运行于HDFS之上的ftp服务器，进行上传、下载等操作。（**目前的实验过程中都是在linux系统下使用ftp client连接运行于HDFS之上的ftp服务器**）