MHA、Lvs、HAProxy、Keepalive、Ceph、Openstack

# 集群(Cluster)     所谓集群是指一组独立的计算机系统构成的一个松耦合的多处理器系统，它们之间通过网络实现进程间的通信。应用程序可以通过网络共享内存进行消息传送，实现分布式计算机。通俗一点来说，就是让若干台计算机联合起来工作(服务)，可以是并行的，也可以是做备份。

 负载均衡(Load Balance)  
   网络的负载均衡是一种动态均衡技术，常见的实现方式是通过一些工具实时地分析数据包，掌握网络中的数据流量状况，把任务合理均衡地分配出去。负载均衡的核心就是“分摊压力”

衡量负载均衡性能的因素：会话率，并发会话数，吞吐量，数据速率。

高可用（HA）

指以减少服务中断时间为目的的服务器集群技术。它通过保护用户的业务程序对外不间断提供的服务，把因软件、硬件、人为造成的故障对业务的影响降低到最小程度。简单说就是保证服务不断运行

高可用集群的衡量标准：通常用平均无故障时间(MTTF)来度量系统的可靠性,用平均故障维修时间（MTTR）来度量系统的可维护性

MHA

作用：部署mysql高可用集群,

集群定义： 使用多台服务提供相同的服务

高可用集群定义：主备模式，被客户端访问的称作主，当主宕机时，备用

一、装包

服务端：

所有节点： yum -y install perl-\*.rpm

rpm -ivh mha4mysql-node-0.56-0.el6.noarch.rpm

管理节点：mha\_manage,mha\_node

yum -y install perl-ExtUtils-\* perl-CPAN\*（环境需要）

tar -zxvf mha4mysql-manager-0.56.tar.gz

cd mha4mysql-manager-0.56

perl Makefile.PL

make

make install

数据节点：mha\_node

1. 环境配置
2. 所有主机互相免密登录
3. 配置主从，主库和备用主库开起半同步复制，备用主库开启不自动删除本机的中继日志文件

set global relay\_log\_purge=off

3、在主机检查是否有同步数据的用户 repluser

mysql> show grants for repluser@"%" ;

4、所有主机上设置一个用于访问的用户

mysql> grant all on \*.\* to root@"%" identified by "123456";

5、拷贝命令

cp mha4mysql-manager-0.56/bin/\* /usr/local/bin/

6、创建工作目录和主配置文件

mkdir /etc/mha\_manager/

cp mha4mysql-manager-0.56/samples/conf/app1.cnf /etc/mha\_manager/

vim /etc/mha\_manager/app1.cnf

[server default]

manager\_workdir=/etc/mha\_manager

manager\_log=/etc/mha\_manager/manager.log

master\_ip\_failover\_script=/usr/local/bin/master\_ip\_failover // 自动failover时候的切换脚本（地址改成设置的vip地址）

ssh\_user=root

ssh\_port=22

repl\_user=repluser // 主从同步用户名

repl\_password=123456 // 主从同步密码

user=root // 连接数据库服务器用户名

password=123456 // 密码

[server1]

hostname=x.x.x.x

port=3306

[server2]

hostname=x.x.x.x

port=3306

candidate\_master=1 // 设置为候选 master

[server3]

hostname=x.x.x.x

port=3306

candidate\_master=1 // 设置为候选 master

[server4]

hostname=x.x.x.x

port=3306

no\_master=1 // 不竞选 master

7、验证配置

验证ssh 免密码登录 数据节点主机

masterha\_check\_ssh --conf=/etc/mha\_manager/app1.cnf

验证 数据节点的主从同步配置（要不调用故障切换脚本）

masterha\_check\_repl --conf=/etc/mha\_manager/app1.cnf

8、配置vip地址

在主库上手动部署vip 地址 192.168.4.100

ifconfig eth0:1 192.168.4.100/24

9、修改故障切换脚本 指定vip地址的部署信息

vim /usr/local/bin/master\_ip\_failover

my $vip = '192.168.4.100/24'; # Virtual IP

10、启动服务并查看服务状态

masterha\_manager --conf=/etc/mha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover

--ignore\_last\_failover:当第一次出现主库宕机后，会调用failover这个脚本，但8小时内不会再开始failover机制，一种方法是清除错误文件，可以通过find搜索\*.failover.error,然后再启动failover机制，还有一种就是通过设置这个参数。让mha不管上一次failover状态

--remove\_dead\_master\_conf app1.cnf 文件里删除宕机的主库的信息

LVS：Linux虚拟服务器

1、工作模式：

（1）tun隧道模式：很少使用

（2）nat模式：相对较少

（3）dr模式：路由模式，应用最多

（4）full nat：在很大规模环境下使用

2、调度算法

（1）轮询rr

（2）加权轮询wrr

（3）最少连接lc

（4）加权最少连接wlc

（5）基于局部的最少连接lblc

（6）带复制的基于局部的最少连接lblcr

（7）源地址散列sh

（8）目标地址散列dh

（9）期望的最少的延迟sed

（10）最少队列调度nq

LVS－NAT模式

客户端请求到达vip，vip将报文经过DNAT （服务器发布）转发给RealServer，RS进行处理，然后RS将网关指向DIP，这样，再经过目标地址转换，将处理的请求发送给客户端。调度器处理大量的数据，因此负压较大。

调度器（配置不通网段的地址并打开路由转发）

1、装包

yum install -y ipvsadm

2、创建虚拟服务器

ipvsadm -A -t vip地址:80 -s rr

3、向虚拟服务器中加入real server（-m表示NAT模式，-w设置权重，-t是TCP）

ipvsadm -a -t vip地址:80 -r RS地址 -m

4、查看

ipvsadm -Ln

5、验证：访问http://vip地址/bbs

在两台服务器上制作不同的主页，以便访问时可以区分算法

6、相关命令

从虚拟服务器中删除RIP

ipvsadm -d -t vip地址:80 -r RS地址

删除虚拟服务器

ipvsadm -D -t vip地址:80

修改调度算法为WRR

ipvsadm -E -t vip地址:80 -s wrr

每隔1秒钟运行ipvsadm -Ln

watch -n1 ipvsadm -Ln

LVS－DR模式 (直接路由，客户端通过调度器直接访问RS)

有两种情况A，vip dip rip不在同一网络内，dip rip在同一个局域网内，vip为i外网IP。在Director中通过广播维护有RS的mac：ip地址对应表，当请求报文到来之后，更改了目标vip的mac地址，在物理网络内由于是通过mac地址通信，因此可以正确找到对应的RS服务器。在对应的RS服务器内，配置有vip，这样，通过vip处理，得到响应数据包源地址vip，目标地址cip。这个时候，只能通过RS的网卡进行通信（虽然在RS中有vip，但是这个只能修改响应报文的源IP，不能与外部主机进行通信），这样，RS就不能接入与vip相同的路由器，因为他们不在同一个网段。这个时候就需要给RS提供一个单独的路由器，然后与cip进行通信

1、在调度器的物理网卡上添加vip

cd /etc/sysconfig/network-scripts/

cp ifcfg-eth0{,:0}

vim ifcfg-eth0:0

TYPE=Ethernet

BOOTPROTO=none

NAME=eth0:0

DEVICE=eth0:0

ONBOOT=yes

IPADDR=192.168.4.100

PREFIX=24

ifup eth0:0

1. 在real server的lo上配置vip

在lo上而不是在物理网口上绑定vip原因：real server发现报文的目标地址是VIP，然后发现在本地的网络设备lo上也绑着这个VIP，于是认为这活是他该干的没错，然后服务器处理这个报文，然后根据路由表将响应报文直接返回给客户

cp ifcfg-lo{,:0}

vim ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.4.100

NETMASK=255.255.255.255 //子网掩码必须是4个255，否则其他主机启动时，都是地址被占用

NETWORK=192.168.4.100

BROADCAST=192.168.4.100

ONBOOT=yes

NAME=lo:0

1. 在real server上修改内核参数，使得客户端发送到VIP的请求只有调度服务器响应，real server(web服务器)不要要响应

arp\_ignore参数的作用是控制系统在收到外部的arp（地址解析协议）请求时，是否要返回arp响应。

1：只响应目的IP地址为接收网卡上的本地地址的arp请求。

arp\_announce的作用是控制系统在对外发送arp请求时，如何选择arp请求数据包的源IP地址。

2：忽略IP数据包的源IP地址，选择该发送网卡上最合适的本地地址作为arp请求的源IP地址。

sysctl -a | grep arp\_ig

echo "net.ipv4.conf.all.arp\_ignore = 1" >> /etc/sysctl.conf

echo "net.ipv4.conf.lo.arp\_ignore = 1" >> /etc/sysctl.conf

sysctl -a | grep arp\_ann

echo "net.ipv4.conf.all.arp\_announce = 2" >> /etc/sysctl.conf

echo "net.ipv4.conf.lo.arp\_announce = 2" >> /etc/sysctl.conf

sysctl -p

4、创建规则，默认LVS采用DR模式，也可以明确使用-g

ipvsadm -A -t 192.168.4.100:80 -s lc

ipvsadm -a -t 192.168.4.100:80 -r 192.168.4.2

ipvsadm -a -t 192.168.4.100:80 -r 192.168.4.3 -g

5、配置服务

touch /etc/sysconfig/ipvsadm

systemctl start ipvsadm

systemctl enable ipvsadm

HAProxy：

1、HAProxy是一个使用C语言编写的自由及开放源代码软件，其提供[高可用性](https://baike.baidu.com/item/%E9%AB%98%E5%8F%AF%E7%94%A8%E6%80%A7" \t "/root/文档\\x/_blank)、[负载均衡](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1" \t "/root/文档\\x/_blank)，以及基于[TCP](https://baike.baidu.com/item/TCP" \t "/root/文档\\x/_blank)和[HTTP](https://baike.baidu.com/item/HTTP" \t "/root/文档\\x/_blank)的应用程序[代理](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86" \t "/root/文档\\x/_blank)。

HAProxy特别适用于那些负载特大的web站点，这些站点通常又需要会话保持或七层处理。HAProxy运行在当前的硬件上，完全可以支持数以万计的[并发连接](https://baike.baidu.com/item/%E5%B9%B6%E5%8F%91%E8%BF%9E%E6%8E%A5" \t "/root/文档\\x/_blank)。并且它的运行模式使得它可以很简单安全的整合进您当前的架构中， 同时可以保护你的web服务器不被暴露到网络上。

1、它的工作原理是Proxy代理。客户端把请求发送到HAProxy后，HAProxy替用户发送请求到web服务器，web服务器响应HAProxy，把页面发给HAProxy。HAProxy再把页面发给客户端。

2、拓扑与LVS－DR完全一样。

Haproxy调度器

1. 装包

yum install -y haproxy

2、修改配置文件

vim /etc/haproxy/haproxy.cfg

把main frontend which proxys to the backends后面的配置全部删除

listen stats

bind 0.0.0.0:1080 # 监控页面的端口号

stats refresh 30s # 页面自动刷新时间是30秒

stats uri /monitor # 页面网址

stats realm HaManager

stats auth admin:admin # 用户名：密码

listen myweb 0.0.0.0:80

cookie SERVERID rewrite

balance roundrobin # 调度算是rr

server web1 192.168.4.2:80 cookie a1i1 check inter 2000 rise 2 fall 5 #每2000ms检查一次服务器，2次成功是好的，5次失败表示故障

server web2 192.168.4.3:80 cookie a1i2 check inter 2000 rise 2 fall 5

5、启动服务

systemctl start haproxy

日志

1. 构成 # man 5 rsyslog.conf

Facility.priority 相当于是 服务.优先级

Facility可以是这些关键字：auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, security ，syslog, user, uucp 以及local0到local7

Local0到local7是预留出来的接口，供第三方应用调用。

Priority可以使用的关键字：debug, info,notice, warning, warn ，err, error，crit, alert,emerg, panic

Debug是最不严重的级别，panic是最严重的级别。如果日志记录优先级是info，表示比info严重的日志都需要记录。

配置haproxy日志

1、配置本机接受通过网络发来的日志

vim /etc/rsyslog.conf

# Provides UDP syslog reception

$ModLoad imudp

$UDPServerRun 514

# Provides TCP syslog reception

$ModLoad imtcp

$InputTCPServerRun 514

systemctl restart rsyslog

1. 查看日志

tail -f /var/log/messages

访问haproxy调度器，可以看到日志信息

Keepalive

搭建高可用的web集群

1. 安装服务

yum install -y keepalived

1. 修改配置文件

vim /etc/keepalived/keepalived.conf

# vrrp\_strict

vrrp\_instance VI\_1 {

state MASTER #辅助写BACKUP

interface eth0

virtual\_router\_id 51 #虚拟路由器ID号

priority 150 #优先级

advert\_int 1 # 心跳消息发送间隔

authentication { # 集群成员共享密码

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

Vip地址 # vip

}

}

Heartbeat：心跳

systemctl start keepalived

1. 查看、验证

ip address show eth0

Ceph

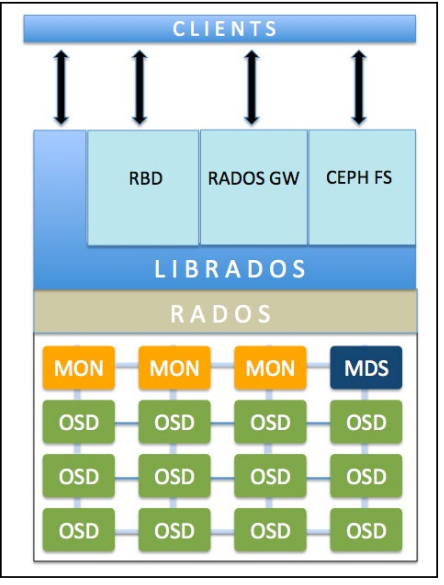
存储：

RAID：独立磁盘冗余阵列

性能、容错、空间

分布式存储：CEPH

1. 组成



1. MON：监视器。MON通过保存一系列集群状态map来监视集群的组件。MON因为保存集群状态，要防止单点故障，所以需要多台；另外，MON需要是奇数，如果出现意见分岐，采用投票机制，少数服从多数。
2. OSD：对象存储设备。真正存储数据的组件。一般来说，每块参与存储的磁盘都需要一个OSD进程。

（3）MDS：元数据服务器。只有CephFS需要它。

元数据：metadata，存储数据的数据。比如一本书内容是数据，那么书的作者、出版社、出版时间之类的信息就是元数据。

1. RADOS：可靠自主分布式对象存储。它是ceph存储的基础，保证一切都以对象形式存储。
2. RBD：RADOS块设备，提供块存储
3. CephFS：提供文件系统级别存储
4. RGW：RADOS网关，提供对象存储

存储分类：

块存储：提供硬盘，如iSCSI

文件级别存储：共享文件夹

对象存储：一切皆对象

http://storage.ctocio.com.cn/281/12110781\_2.shtml

CEPH环境准备

1. 准备6台虚拟机

主机名、IP地址

1. 在物理主机上配置名称解析

[root@room8pc16 nsd2018]# for i in {1..6}

> do

> echo -e "192.168.4.$i\tnode$i.tedu.cn\tnode$i" >> /etc/hosts

> done

3、提前将服务器的密钥保存，不需要ssh时回答yes

[root@room8pc16 nsd2018]# ssh-keyscan node{1..6} >> /root/.ssh/known\_hosts

4、实现免密登陆

[root@room8pc16 nsd2018]# for i in {1..6}

> do

> ssh-copy-id node$i

> done

5、配置yum源

[root@room8pc16 nsd2018]# mkdir /var/ftp/ceph/

[root@room8pc16 nsd2018]# vim /etc/fstab

/ISO/rhcs2.0-rhosp9-20161113-x86\_64.iso /var/ftp/ceph iso9660 defaults 0 0

[root@room8pc16 nsd2018]# mount -a

[root@room8pc16 nsd2018]# vim /tmp/server.repo

[rhel7.4]

name=rhel7.4

baseurl=ftp://192.168.4.254/rhel7.4

enabled=1

gpgcheck=0

[mon]

name=mon

baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86\_64/MON

enabled=1

gpgcheck=0

[osd]

name=osd

baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86\_64/OSD

enabled=1

gpgcheck=0

[tools]

name=tools

baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86\_64/Tools

enabled=1

gpgcheck=0

[root@room8pc16 nsd2018]# for vm in node{1..6}

> do

> scp /tmp/server.repo ${vm}:/etc/yum.repos.d/

> done

1. 配置node1节点为管理节点
2. 配置名称解析

[root@node1 ~]# for i in {1..6}; do echo -e "192.168.4.$i\tnode$i.tedu.cn\tnode$i" >> /etc/hosts; done

1. 配置免密登陆

[root@node1 ~]# ssh-keyscan node{1..6} >> /root/.ssh/known\_hosts

[root@node1 ~]# ssh-keygen -f /root/.ssh/id\_rsa -N ''

[root@node1 ~]# for i in {1..6}; do ssh-copy-id node$i; done

[root@node1 ~]# for vm in node{1..6}

> do

> scp /etc/hosts ${vm}:/etc/

> done

7、NTP网络时间协议，基于UDP123端口。用于时间同步

时区：地球一圈360度，经度每15度角一个时区，共24个时区。以英国格林威治这个城市所在纵切面为基准。北京在东八区。

夏季节约时间：夏令时。DST

Stratum：时间服务器的层级。

时间准确度：原子钟。

8、配置node6为时间服务器

（1）配置

[root@node6 ~]# yum install -y chrony

[root@node6 ~]# vim /etc/chrony.conf

server 0.centos.pool.ntp.org iburst

#server 1.centos.pool.ntp.org iburst

#server 2.centos.pool.ntp.org iburst

#server 3.centos.pool.ntp.org iburst

allow 192.168.4.0/24

local stratum 10

1. 启动服务

[root@node6 ~]# systemctl enable chronyd

[root@node6 ~]# systemctl restart chronyd

将node1-5配置为NTP的客户端

（1）配置

[root@node1 ~]# vim /etc/chrony.conf

#server 0.rhel.pool.ntp.org iburst

#server 1.rhel.pool.ntp.org iburst

#server 2.rhel.pool.ntp.org iburst

#server 3.rhel.pool.ntp.org iburst

server 192.168.4.6 iburst

[root@node1 ~]# systemctl restart chronyd

1. 测试

[root@node1 ~]# date -s "2018-7-13 12:00:00"

[root@node1 ~]# ntpdate 192.168.4.6

[root@node1 ~]# date

（3）同步其他主机

[root@node1 ~]# for i in {2..5}

> do

> scp /etc/chrony.conf node$i:/etc/

> done

[root@node1 ~]# for vm in node{2..5}

> do

> ssh $vm systemctl restart chronyd

> done

9、为node1-3各添加3块10GB的磁盘

可以在虚拟机不关机的情况下，直接添加硬盘

安装ceph

1. 在node1上安装部署软件

[root@node1 ~]# yum install -y ceph-deploy

1. 创建ceph部署工具的工作目录

[root@node1 ~]# mkdir ceph-clu

1. 创建参与集群节点的配置文件

[root@node1 ceph-clu]# ceph-deploy new node{1..3}

[root@node1 ceph-clu]# ls

4、在3个节点上安装软件包

[root@node1 ceph-clu]# ceph-deploy install node{1..3}

1. 初始化mon服务

[root@node1 ceph-clu]# ceph-deploy mon create-initial

如果出现以下错误：

[node1][ERROR ] admin\_socket: exception getting command descriptions: [Errno 2] No such file or directory

解决方案：

[root@node1 ceph-clu]# vim ceph.conf 最下面加入行：

public\_network = 192.168.4.0/24

再执行以下命令：

[root@host1 ceoh-clu]# ceph-deploy --overwrite-conf config push node1 node2 node3

1. 把node1-3的vdb作为日志盘。Ext／xfs都是日志文件系统，一个分区分成日志区和数据区。为了更好的性能，vdb专门作为vdc和vdd的日志盘。

[root@node1 ceph-clu]# for vm in node{1..3}

> do

> ssh $vm parted /dev/vdb mklabel gpt

> done

[root@node1 ceph-clu]# for vm in node{1..3}; do ssh $vm parted /dev/vdb mkpart primary 1M 50% ; done

[root@node1 ceph-clu]# for vm in node{1..3}; do ssh $vm parted /dev/vdb mkpart primary 50% 100% ; done

[root@node1 ceph-clu]# for vm in node{1..3}; do ssh ${vm} chown ceph.ceph /dev/vdb? ; done

1. 创建OSD设备

[root@node1 ceph-clu]# for i in {1..3}

> do

> ceph-deploy disk zap node$i:vdc node$i:vdd

> done

[root@node1 ceph-clu]# for i in {1..3}

> do

> ceph-deploy osd create node$i:vdc:/dev/vdb1 node$i:vdd:/dev/vdb2

> done

1. 验证

到第7步为止，ceph已经搭建完成。查看ceph状态

[root@node1 ceph-clu]# ceph -s 如果出现health HEALTH\_OK表示正常

1. 排错

https://www.zybuluo.com/dyj2017/note/920621

CEPH应用

1. 块存储：使用最多的一种方式
2. cephFS：了解，不建议在生产环境中使用，因为还不成熟
3. 对象存储：了解，使用亚马逊的s3

使用RBD(Rados块设备)

1. 查看存储池

[root@node1 ~]# ceph osd lspools

可以查看到0号镜像池，名字为rbd

1. 创建名为demo-img的镜像大小为10GB

[root@node1 ~]# rbd create demo-img --image-feature layering --size 10G

[root@node1 ~]# rbd list

[root@node1 ~]# rbd info demo-img

3、创建第2个镜像，名为image，指定它位于rbd池中

[root@node1 ~]# rbd create rbd/image --image-feature layering --size 10G

编写UDEV规则，使得vdb1和vdb2重启后，属主属组仍然是ceph

[root@node1 ~]# vim /etc/udev/rules.d/90-cephdisk.rules

ACTION=="add", KERNEL=="vdb[12]", OWNER="ceph", GROUP="ceph"