LVS+Keepalived 负载均衡

LVS 简介及工作原理

LVS 是 Linux Virtual Server 的简写，意即 Linux 虚拟服务器，是一个虚拟的服务器集群系统。本项目在 1998 年 5 月由章文嵩博士成立，是中国国内最早出现的自由软件项目之一。

LVS 简单工作原理：用户请求 LVS VIP，LVS 根据转发方式和算法，将请求转发给后端服务器，后端服务器接受到请求，返回给用户。对于用户来说，看不到 WEB 后端具体的应用。

LVS 转发方式有三种，分别是 NAT、DR、TUN 模式，常用算法：RR、LC、WRR、WLC 模式等（RR 为轮询模式，LC 为最少连接模式）

LVS NAT 原理：用户请求 LVS 到达 director,director 将请求的报文的目标地址改成后端的 realserver 地址，同时将报文的目标端口也改成后端选定的 realserver 相应端口，最后将报文发送到 realserver，realserver 将数据返给 director，director 再把数据发送给用户。（两次请求都经过 director，所以访问大的话，director 会成为瓶颈）

LVS DR 原理：用户请求 LVS 到达 director,director 将请求的报文的目标 MAC 地址改成后端的 realserver MAC 地址，目标 IP 为 VIP（不变），
源 IP 为用户 IP 地址（保持不变），然后 Director 将报文发送到 realserver，realserver 检测到目标为自己本地 IP，如果在同一个网段，然后将请求直接返给用户。如果用户跟 realserver 不在一个网段，则通过网关返回用户。（此种转发效率最高）

LVS TUN 原理：跟 LVS DR 类似，也是改变封装 MAC 地址，多了一层隧道加密。实施环境复杂，比 LVS DR 模式效率略低。

- LVS 环境安装配置

下载 LVS 所需软件 ipvsadm-1.2.4.tar.gz 软件，编译安装：

```
wget                                                    -c
http://www.linuxvirtualserver.org/software/kernel-2.6/ipvsadm-1.24.tar.gz
ln -s /usr/src/kernels/2.6.* /usr/src/linux  //IPVS 模块编译进内核里，需要做软连接
tar xzvf ipvsadm-1.24.tar.gz &&cd ipvsadm-1.24  && make && make install
```

LVS 安装完毕之后，需要进行配置，配置的步骤有两步，第一步为定义端口服务，第二步为

添加 realserver 后端服务。

```
ipvsadm -A -t 192.168.33.188:80 -s rr
ipvsadm -a -t 192.168.33.188:80 -r 192.168.33.12 -m -w 2
ipvsadm -a -t 192.168.33.188:80 -r 192.168.33.13 -m -w 2
```

参数说明：

-A 增加一台虚拟服务器地址。

-t 虚拟服务器提供的是 tcp 服务。

-s 使用的调度算法。

-a 在虚拟服务器中增加一台后端真实服务器。

-r 指定真实服务器地址。

-m 设置当前转发方式为 NAT 模式；-g 为直接路由模式；-i 模式为隧道模式。

-w 后端真实服务器的权重。

查看 LVS 转发列表命令为：ipvsadm -Ln

```
[root@node2 ~]#
[root@node2 ~]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.149.129:80 rr
  -> 192.168.149.131:80           Masq    2      0          0
  -> 192.168.149.130:80           Masq    2      0          0
[root@node2 ~]#
```

我们会发现，如果这台 LVS 发生突发情况，down 机了，那后端所有的应用程序都访问不了。如何避免这种问题呢，这里需要用到故障切换，也就是如果有一台备用的 LVS 就好了，主 down 了，自动切换到从，怎么实现这个需求，接下来讲解的 keepalived 软件就是专门用来做故障检测及切换的。

Keepalived 基于三层检测（IP 层，TCP 层，及应用层），主要用于检测 web 服务器的状态，如果有一台 web 服务器死机，或工作出现故障，Keepalived 检测到并将有故障的 web 服务器从系统中剔除；

当 web 服务器工作正常后 Keepalived 自动将 web 服务器加入到服务器群中，这些工作全部自动完成，不需要人工干涉，需要人工做的只是修复故障的 web 服务器。

需要注意一点，如果使用了 keepalived.conf 配置，就不需要再执行 ipvs -A 命令去添加均衡的 realserver 命令了，所有的配置都会在 keepalived.conf 里面，一个配置文件搞定所有，即只需要安装 ipvs 模块。

➢ Keepalived 安装配置

官方下载 keepalived 相应稳定版本：

cd /usr/src ;wget -c http://www.keepalived.org/software/keepalived-1.1.15.tar.gz

tar -xzvf keepalived-1.1.15.tar.gz &&cd keepalived-1.1.15 && ./configure && make && make install

安装完毕，配置 keepalived 服务为系统服务。

```
DIR=/usr/local/
cp $DIR/etc/rc.d/init.d/keepalived /etc/rc.d/init.d/ && cp $DIR/etc/sysconfig/keepalived /etc/sysconfig/ && mkdir -p /etc/keepalived && cp $DIR/sbin/keepalived /usr/sbin/
```

在 MASTER 上/etc/keepalived/目录创建 keepalived.conf 配置文件，并写入如下内容：

```
! Configuration File for keepalived

global_defs {
   notification_email {
      wgkgood@163.com
   }
   notification_email_from wgkgood@163.com
   smtp_server 127.0.0.1
   smtp_connect_timeout 30
   router_id LVS_DEVEL
}

# VIP1
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    lvs_sync_daemon_inteface eth0
    virtual_router_id 51
    priority 100
    advert_int 5
    nopreempt
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
```

```
        192.168.33.188
    }
}

virtual_server 192.168.33.188 80 {
    delay_loop 6
    lb_algo wrr
    lb_kind DR
#    persistence_timeout 60
    protocol TCP

    real_server 192.168.33.12 80 {
        weight 100
        TCP_CHECK {
        connect_timeout 10
        nb_get_retry 3
        delay_before_retry 3
        connect_port 80
        }
    }
    real_server 192.168.33.13 80 {
        weight 100
        TCP_CHECK {
        connect_timeout 10
        nb_get_retry 3
        delay_before_retry 3
        connect_port 80

        }
    }
}
```

如上配置文件,红色标记的地方需要注意,state 状态主服务器设置 MASTER,从设置为 BACKUP,优先级备机设置比 MASTER 小，例如设置 90，使用 TCP 端口检测。

在 LVS BACKUP 服务器写入如下配置，需要注意的是客户端的配置要修改优先级及状态：

```
! Configuration Filefor keepalived
global_defs {
    notification_email {
      wgkgood@163.com
    }
    notification_email_from wgkgood@163.com
    smtp_server  127.0.0.1
    smtp_connect_timeout  30
    router_id LVS_DEVEL
```

```
}
# VIP1
vrrp_instance VI_1 {
    state BACKUP
    interface  eth0
    lvs_sync_daemon_inteface eth0
    virtual_router_id  51
    priority  90
    advert_int  5
    authentication {
        auth_type PASS
        auth_pass  1111
    }
    virtual_ipaddress {
        192.168.33.11
    }
}
#REAL_SERVER_1
virtual_server  192.168.33.11  80  {
    delay_loop  6
    lb_algo wlc
    lb_kind DR
    persistence_timeout  60
    protocol TCP
    real_server  192.168.33.130  80  {
        weight  100
        TCP_CHECK {
        connect_timeout  10
        nb_get_retry  3
        delay_before_retry  3
        connect_port  80
        }
}
#REAL_SERVER_2
    real_server  192.168.33.131  80  {
        weight  100
        TCP_CHECK {
        connect_timeout  10
        nb_get_retry  3
        delay_before_retry  3
        connect_port  80
        }
    }
}
```

如上设置，LVS 主备配置完毕，接下来需要在 realserver 配置 LVS VIP，为什么要在 realserver 绑定 VIP 呢？

客户端访问 director 的 VIP，director 接收请求，将通过相应的算法将请求转发给相应的 realserver。在转发的过程中，会修改请求包的目的 mac 地址，目的 ip 地址不变。

Realserver 接收请求，并直接响应客户端。这时便出现一个问题，director 此时与 realserver 位于同一个网络中，当 director 直接将请求转发给 realserver 时，realserver 检测到该请求包的目的 ip 是 vip 而并非自己，便会丢弃，而不会响应。为了解决这个问题，所以需要在所有 Realserver 上都配上 VIP。

为什么一定要配置在 lo 接口上呢？

在 realserver 上的 lo 口配置 VIP,这样限制了 VIP 不会在物理交换机上产生 MAC 地址表，从而避免 IP 冲突。

客户端启动 Realserver.sh 脚本内容：

```
#!/bin/sh
#LVS Client Server
VIP=192.168.33.188
case  $1  in
start)
    ifconfig lo:0 $VIP netmask 255.255.255.255 broadcast $VIP
    /sbin/route add -host $VIP dev lo:0
    echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
    echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
    echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
    echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
    sysctl -p >/dev/null 2>&1
    echo "RealServer Start OK"
    exit 0
;;
stop)
    ifconfig lo:0 down
    route del $VIP >/dev/null 2>&1
    echo "0" >/proc/sys/net/ipv4/conf/lo/arp_ignore
    echo "0" >/proc/sys/net/ipv4/conf/lo/arp_announce
    echo "0" >/proc/sys/net/ipv4/conf/all/arp_ignore
    echo "0" >/proc/sys/net/ipv4/conf/all/arp_announce
    echo "RealServer Stoped OK"
    exit 1
;;
*)
```

```
    echo "Usage: $0 {start|stop}"
;;
esac
```

LVS 网站故障排查经验：

如果发现主网站无法访问，首先第一步 ping 网站域名是否能 ping 通，如果域名无法访问，试着使用 IP 能不能访问，如果 IP 能访问，首先排查到域名解析问题。

如果 IP 也无法访问，登录 LVS 服务器，使用命令 ipvsadm -Ln 查看当前连接状态和查看/var/log/messages 日志信息，可以在 LVS 上访问 realserver ip，进行排查。

如果 LVS 服务正常，后端 realserver 服务异常，然后查看 nginx 日志信息，是否有大量恶意访问，临时重启看是否能访问。

如果有恶意 ip 访问，找出恶意 ip，经确认可以关闭后，使用 iptables 防火墙临时关闭即可。

LVS+keepalived+Nginx+Apache+PHP+MySQL+Memcached+Redis