

背景:

随着你的网站业务量的增长你网站的服务器压力越来越大? 需要负载均衡方案! 商业的硬件如 F5 又太贵, 你们又是创业型互联公司如何有效节约成本, 节省不必要的浪费? 同时实现商业硬件一样的高性能高可用的功能? 有什么好的负载均衡可伸张可扩展的方案吗? 答案是肯定的! 有! 我们利用 LVS+Keepalived 基于完整开源软件的架构可以为你提供一个负载均衡及高可用的服务器。

一、LVS+Keepalived 介绍

LVS

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。

目前 LVS 主要有三种请求转发方式和八种调度算法。

LVS 的三种请求转发方式:

Virtual Server via NAT(VS-NAT): 用地址翻译实现虚拟服务器。地址转换器有能被外界访问到的合法 IP 地址, 它修改来自专有网络的流出包的地址。外界看起来包是来自地址转换器本身, 当外界包送到转换器时, 它能判断出应该将包送到内部网的哪个节点。

优点是节省 IP 地址, 能对内部进行伪装

缺点是效率低, 因为返回给请求方的流量经过转换器

Virtual Server via IP Tunneling(VS-TUN): 用 IP 隧道技术实现虚拟服务器。这种方式是在集群的节点不在同一个网段时可用的转发机制, 是将 IP 包封装在其他网络流量中的方法。为了安全的考虑, 应该使用隧道技术中的 VPN, 也可使用租用专线。集群所能提供的服务是基于 TCP/IP 的 Web 服务、Mail 服务、News 服务、DNS 服务、Proxy 服务器等等。

Virtual Server via Direct Routing(VS-DR): 用直接路由技术实现虚拟服务器。当参与集群的计算机和作为控制管理的计算机在同一个网段时可以用此法, 控制管理的计算机接收到请求包时直接送到参与集群的节点。优点是返回给客户的流量不经过控制主机, 速度快开销少。

LVS 的包转发策略有三种:

NAT (Network Address Translation)模式。 LB 收到用户请求包后, LB 将请求包中虚拟服务器的 IP 地址转换为某个选定 RS 的 IP 地址, 转发给 RS; RS 将应答包发给 LB, LB 将应答包中 RS 的 IP 转为虚拟服务器的 IP 地址, 回送给用户。

IP 隧道 (IP Tunneling)模式。 LB 收到用户请求包后, 根据 IP 隧道协议封装该包, 然后传给某个选定的 RS; RS 解出请求信息, 直接将应答内容传给用户。此时要求 RS 和 LB 都要支持 IP 隧道协议。

DR(Direct Routing)模式。 LB 收到请求包后, 将请求包中目标 MAC 地址转换为某个选定 RS 的 MAC 地址后将包转发出去, RS 收到请求包后, 可直接将应答内容传给用户。此时要求 LB 和所有 RS 都必须在一个物理段内, 且 LB 与 RS 群共享一个虚拟 IP。

LVS 的八种调度算法:

1. 轮叫调度 (Round Robin)

调度器通过“轮叫”调度算法将外部请求按顺序轮流分配到集群中的真实服务器上，它均等地对待每一台服务器，而不管服务器上实际的连接数和系统负载。

2. 加权轮叫 (Weighted Round Robin)

调度器通过“加权轮叫”调度算法根据真实服务器的不同处理能力来调度访问请求。这样可以保证处理能力强的服务器能处理更多的访问流量。调度器可以自动问询真实服务器的负载情况，并动态地调整其权值。

3. 最少链接 (Least Connections)

调度器通过“最少连接”调度算法动态地将网络请求调度到已建立的链接数最少的服务器上。如果集群系统的真实服务器具有相近的系统性能，采用“最小连接”调度算法可以较好地均衡负载。

4. 加权最少链接 (Weighted Least Connections)

在集群系统中的服务器性能差异较大的情况下，调度器采用“加权最少链接”调度算法优化负载均衡性能，具有较高权值的服务器将承受较大比例的活动连接负载。调度器可以自动问询真实服务器的负载情况，并动态地调整其权值。

5. 基于局部性的最少链接 (Locality-Based Least Connections)

“基于局部性的最少链接”调度算法是针对目标 IP 地址的负载均衡，目前主要用于 Cache 集群系统。该算法根据请求的目标 IP 地址找出该目标 IP 地址最近使用的服务器，若该服务器是可用的且没有超载，将请求发送到该服务器；若服务器不存在，或者该服务器超载且有服务器处于一半的工作负载，则用“最少链接”的原则选出一个可用的服务器，将请求发送到该服务器。

6. 带复制的基于局部性最少链接 (Locality-Based Least Connections with Replication)

“带复制的基于局部性最少链接”调度算法也是针对目标 IP 地址的负载均衡，目前主要用于 Cache 集群系统。它与 LBLC 算法的不同之处是它要维护从一个目标 IP 地址到一组服务器的映射，而 LBLC 算法维护从一个目标 IP 地址到一台服务器的映射。该算法根据请求的目标 IP 地址找出该目标 IP 地址对应的服务器组，按“最小连接”原则从服务器组中选出一台服务器，若服务器没有超载，将请求发送到该服务器；若服务器超载，则按“最小连接”原则从这个集群中选出一台服务器，将该服务器加入到服务器组中，将请求发送到该服务器。同时，当该服务器组有一段时间没有被修改，将最忙的服务器从服务器组中删除，以降低复制的程度。

7. 目标地址散列 (Destination Hashing)

“目标地址散列”调度算法根据请求的目标 IP 地址，作为散列键 (Hash Key) 从静态分配的散列表找出对应的服务器，若该服务器是可用的且未超载，将请求发送到该服务器，否则返回空。

8. 源地址散列 (Source Hashing)

“源地址散列”调度算法根据请求的源 IP 地址，作为散列键 (Hash Key) 从静态分配的散列表找出对应的服务器，若该服务器是可用的且未超载，将请求发送到该服务器，否则返回

空。

了解这些算法原理能够在特定的应用场合选择最适合的调度算法，从而尽可能地保持 Real Server 的最佳利用性。当然也可以自行开发算法，不过这已超出本文范围，请参考有关算法原理的资料。

*以下根据我们的业务环境，只讨论 VS-DR 转发方式

二、集群环境

系统：CentOS5.5-64 位

IP 信息列表：

LVS-DR 模式虚拟 IP	LVS-DR-VIP	WAN:223.202.22.118
LAN:192.168.22.118		
LVS-DR 模式主转发器	LVS-DR-Master	WAN:223.202.22.114
LAN:192.168.22.114		
LVS-DR 模式从转发器	LVS-DR-BACKUP	WAN:223.202.22.115
LAN:192.168.22.115		
真实 WEB-SERVER1	WEB1-Realserver	WAN:223.202.22.116
LAN:192.168.22.116		
真实 WEB-SERVER2	WEB2-Realserver	WAN:223.202.22.117
LAN:192.168.22.117		
网段网关	GateWay	WAN:223.202.22.113

三、安装 LVS 和 Keepalived

1. 安装系统必备包

```
yum -y install gd php-gd gcc gcc-c++ autoconf libmcrypt* libjpeg gettext libjpeg-devel libpng  
libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2  
glib2-devel bzip2 bzip2-devel ncurses ncurses-devel curl curl-devel
```

2.lvs 安装

```
yum install -y ipvsadm
```

3.检查 lvs 是否安装成功

```
[root@VM01 ~]# lsmod |grep ip_vs
```

```
[root@VM01 ~]# ipvsadm
```

```
IP Virtual Server version 1.2.1 (size=4096)
```

```
Prot LocalAddress:Port Scheduler Flags
```

```
-> RemoteAddress:Port Forward Weight ActiveConn InActConn
```

```
[root@VM01 ~]# lsmod |grep ip_vs
```

```
ip_vs 122113 0
```

如出现上述输出表明 lvs 已经安装正确并且已经加载入内核

4.在 LVS-DR 模式主转发器和从转发器上安装 keepalived

```
./configure --prefix=/opt/keepalived --with-kernel-dir=/usr/src/kernels/2.6.18-194.el5-x86_64 &&  
make && make install
```

四、配置 LVS 和 keepalived

1.LVS-DR, 配置 LVS 脚本实现负载均衡

```
lvs_dr.sh  
#!/bin/bash  
#  
# Director Server      Startup script for the Director Server  
#  
# description: LVS of DirectorServer  
#  
# Source function library  
/etc/init.d/functions  
  
vip1=223.202.22.118  
rip1=223.202.22.116  
rip2=223.202.22.117  
  
logger $0 called with $1  
  
case "$1" in  
  
start)  
  
    echo "1" > /proc/sys/net/ipv4/ip_forward  
    /sbin/ipvsadm --set 30 5 60  
    /sbin/ifconfig eth0:0 $vip1 broadcast $vip1 netmask 255.255.255.255 broadcast $vip1 up  
    /sbin/route add -host $vip1 dev eth0:0  
    /sbin/ipvsadm -A -t $vip1:80 -s wrr -p 5  
    /sbin/ipvsadm -a -t $vip1:80 -r $rip1:80 -g -w 1  
    /sbin/ipvsadm -a -t $vip1:80 -r $rip2:80 -g -w 1  
    touch /var/lock/subsys/ipvsadm >/dev/null 2>&1  
    echo "ipvsadm Start OK"  
  
;;  
  
stop)
```

```

    /sbin/ipvsadm -C
    /sbin/ipvsadm -Z
    /sbin/ifconfig eth0:0 down
    /sbin/route del $vip1
    /bin/rm -rf /var/lock/subsys/ipvsadm >/dev/null 2>&1
    echo "ipvsadm Stoped"

;;

status)

    if [ ! -e /var/lock/subsys/ipvsadm ];then
        echo "ipvsadm stoped"
        exit 1
    else
        /sbin/ipvsadm -L -n
        echo "ipvsadm OK"
    fi

;;

*)

    echo "Usage: $0 {start|stop|status}"
    exit 1

esac

exit 0

```

2.LVS-REALSERVER,配置真实服务器脚本

```

real_server.sh
#!/bin/bash
#description : start realserver
VIP=223.202.17.118
/etc/init.d/functions

```

```

case "$1" in
start)
echo " start LVS of REALServer"
/sbin/ifconfig lo:0 $VIP broadcast $VIP netmask 255.255.255.255 up
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
;;
stop)
/sbin/ifconfig lo:0 down
echo "close LVS Directorserver"
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 "Usage: $0 {start|stop}"
exit 1
esac

```

3.配置 keepalived 实现 HA

keepalived 的所有配置都在一个配置文件里面设置，支持的配置项也比较多，分为三类：

- a.全局配置（global configuration）
- b.vrrpd 配置
- c.lvs 配置

***keepalived 只需更改 keepalived.conf 即可，master 和 backup 只相差 priority**

master 上 keepalived.conf 配置：

```

#writed by yang
#guration File for keepalived

#####
####global define####
#####

global_defs {
    router_id LVSMaster    ###定义全局路由 ID，本机标示
    notification_email    ###指定 keepalived 发生事件时候，需要发送到的 email 对象，可
    以多个，每行一个

```

```

    {
        dongjie.yang@youlu-inc.com
    }
notification_email_from dongjie.yang@youlu-inc.com
smtp_server 127.0.0.1
smtp_connect_timeout 30
}

#####
####定义内外网两个 group 分别检查内外网网卡####
#####

vrrp_sync_group VGM {
    group {
        WAN          ###定义 vvrp 外网实例组名为 WAN
        LAN          ###定义 vvrp 外网实例组名为 LAN
    }
    ###notify_master /opt/keepalived/shell/to_master.sh      ###定义切换到 master 时候的脚
    本，可加入数据库报警短信
    ###notify_backup /opt/keepalived/shell/to_backup.sh      ###定义切换到 backup 时候的
    脚本，可加入数据库报警短信
    smtp alter      ###master 和 backup 互相切换后发送邮件通知
}

#####
# vvrp_instance define #
#####

vrrp_instance WAN {
    state MASTER          ###状态为 MASTER
    interface eth0        ###网卡接口为 eth0
    lvs_sync_daemon_inteface eth0  ###负载均衡器之间的监控接口
    virtual_router_id 50  ###虚拟路由标识 50（需与 backup 保持一致）
    priority 220          ###优先级 220，要大于 backup，如 backup 为 120
    advert_int 5          ###同步检查时间间隔 5 秒
    ###nopreempt          ###不抢占
    authentication {
        auth_type PASS    ###配置 master 与 backup 间认证密码（需与 backup 保持一
        致）
        auth_pass 1111
    }
    virtual_ipaddress {
        223.202.22.118    ###虚拟 ip 地址
    }
}

```

```

    }
}
vrrp_instance LAN {
    state MASTER          ###状态为 MASTER
    interface eth1        ###网卡接口为 eth1
    lvs_sync_daemon_inteface eth1  ###负载均衡器之间的监控接口
    virtual_router_id 60   ###虚拟路由标识 60（需与 backup 保持一致）
    priority 200           ###优先级 200，要大于 backup，如 backup 为 100
    advert_int 5           ###同步检查时间间隔 5 秒
    authentication {
        auth_type PASS     ###配置 master 与 backup 间认证密码（需与 backup 保持一
致）
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.22.118     ###虚拟 ip 地址
    }
}

```

```

#####
# LVS 配置  virtual machine setting #
#####
# setting port 80 forward
virtual_server 223.202.22.118 80 {
    delay_loop 6          ###健康检查时间间隔 6 秒
    lb_algo wrr           ###负载均衡调度算法
    lb_kind DR            ###负载均衡转发规则
    nat_mask 255.255.255.240
    persistence_timeout 20 ###会话保持时间 20 秒，如 ipvsadm 已经配置这里可以不写
    protocol TCP          ###设置转发协议为 tcp

    real_server 223.202.22.116 80 {
        weight 100        ###权重 100(数值越大，权重越高)
        TCP_CHECK {
            connect_timeout 3 ###tcp 检查超时时间 3 秒
            nb_get_retry 3    ###重连次数
            delay_before_retry 3 ###重连间隔
            connect_port 80   ###检测端口 80
        }
    }
}

real_server 223.202.22.117 80 {
    weight 100            ###权重 100(数值越大，权重越高)
    TCP_CHECK {

```

```

        connect_timeout 3 ###tcp 检查超时时间 3 秒
        nb_get_retry 3
        delay_before_retry 3
        connect_port 80      ###检测端口 80
    }
}
}

```

backup 上 keepalived.conf 配置:

```

#writed by yang
#guration File for keepalived
#global define

global_defs {
    router_id LVSBACKUP      ###定义全局路由 ID, 本机标示
    notification_email      ###指定 keepalived 发生事件时候, 需要发送到的 email 对象, 可以多个, 每行一个
    {
        dongjie.yang@youlu-inc.com
    }
    notification_email_from dongjie.yang@youlu-inc.com
    smtp_server 127.0.0.1
    stmp_connect_timeout 30
}

#####定义内外网两个 group 分别检查内外网网卡#####
vrrp_sync_group VGM {
    group {
        WAN          ###定义 vvrp 外网实例组名为 WAN
        LAN          ###定义 vvrp 外网实例组名为 LAN
    }
    ###notify_master /opt/keepalived/shell/to_master.sh      ###定义切换到 master 时候的脚本, 可加入数据库报警短信
    ###notify_backup /opt/keepalived/shell/to_backup.sh      ###定义切换到 backup 时候的脚本, 可加入数据库报警短信
    smtp_alter      ###master 和 backup 互相切换后发送邮件通知
}

#####

```

```

# vrrp_instance define #
#####
vrrp_instance WAN {
    state BACKUP          ###状态为 MASTER
    interface eth0        ###网卡接口为 eth0
    lvs_sync_daemon_inteface eth0  ###负载均衡器之间的监控接口
    virtual_router_id 50    ###虚拟路由标识 50（需与 backup 保持一致）
    priority 120           ###优先级 200，要大于 backup，如 backup 为 100
    advert_int 5          ###同步检查时间间隔 5 秒
    authentication {
        auth_type PASS    ###配置 master 与 backup 间认证密码（需与 backup 保持一
致）
        auth_pass 1111
    }
    virtual_ipaddress {
        223.202.22.118    ###虚拟 ip 地址
    }
}

vrrp_instance LAN {
    state BACKUP          ###状态为 MASTER
    interface eth1        ###网卡接口为 eth0
    lvs_sync_daemon_inteface eth1  ###负载均衡器之间的监控接口
    virtual_router_id 60    ###虚拟路由标识 60（需与 backup 保持一致）
    priority 100          ###优先级要大于 backup
    advert_int 5          ###同步检查时间间隔 5 秒
    authentication {
        auth_type PASS    ###配置 master 与 backup 间认证密码（需与 backup 保持一
致）
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.22.118    ###虚拟 ip 地址
    }
}

#####
# virtual machine setting #
#####
# setting port 80 forward
virtual_server 223.202.22.118 80 {
    delay_loop 6          ###健康检查时间间隔 6 秒
    lb_algo wrr           ###负载均衡调度算法
    lb_kind DR            ###负载均衡转发规则
    persistence_timeout 20 ###会话保持时间 20 秒，如 ipvsadm 已经配置这里可以不写
    protocol TCP          ###设置转发协议为 tcp
}

```

```

real_server 223.202.22.116 80 {
    weight 100          ###权重 100(数值越大，权重越高)
    TCP_CHECK {
        connect_timeout 3 ###tcp 检查超时时间 3 秒
        nb_get_retry 3
        delay_before_retry 3
        connect_port 80    ###检测端口 80
    }
}

real_server 223.202.22.117 80 {
    weight 100          ###权重 100(数值越大，权重越高)
    TCP_CHECK {
        connect_timeout 3 ###tcp 检查超时时间 3 秒
        nb_get_retry 3
        delay_before_retry 3
        connect_port 80    ###检测端口 80
    }
}
}

```

*当 master 出现问题后，backup 会竞争成 master，当 master 恢复后又会竞争成 master，这样对频繁访问的业务来说不能容忍，所以可以设置成不抢占，就是当 master 失效恢复后，已经成为 master 的原来的 backup 还是 master 设置这个选项时候两台主机状态都要设置成 BACKUP，此时只通过 priority 进行抢占，在此我们只讨论抢占的情况

五、实例测试

1.启动

```

启动 realserver 上的 apache          /opt/apache/bin/apachectl -k start
启动 realserver 上的 real_server.sh   /root/real_server.sh start
启动 lvs 主从转发器上的 lvs_dr.sh    /root/lvs_dr.sh start
启动 lvs 主转发器上的 keepalived     /opt/keepalived/sbin/keepalived -f
/opt/keepalived/etc/keepalived/keepalived.conf.master
启动 lvs 从转发器上的 keepalived     /opt/keepalived/sbin/keepalived -f
/opt/keepalived/etc/keepalived/keepalived.conf.backup

```

2.观察 lvs 主从转发器系统日志

```
tail -f /var/log/message
```

观察 lvs 主

```
Nov 23 15:34:04 VM01 Keepalived: Starting Keepalived v1.1.15 (11/23,2011)
Nov 23 15:34:04 VM01 Keepalived_healthcheckers: Using MII-BMSR NIC polling thread...
Nov 23 15:34:04 VM01 Keepalived_healthcheckers: Registering Kernel netlink reflector
Nov 23 15:34:04 VM01 Keepalived_healthcheckers: Registering Kernel netlink command
channel
Nov 23 15:34:04 VM01 Keepalived_healthcheckers: Opening file
'/opt/keepalived/etc/keepalived/keepalived.conf.master'.
Nov 23 15:34:04 VM01 Keepalived_healthcheckers: Configuration is using : 13681 Bytes
Nov 23 15:34:04 VM01 Keepalived: Starting Healthcheck child process, pid=5649
Nov 23 15:34:04 VM01 Keepalived_healthcheckers: Activating healthchecker for service
[223.202.22.116:80]
Nov 23 15:34:04 VM01 Keepalived_healthcheckers: Activating healthchecker for service
[223.202.22.117:80]
Nov 23 15:34:04 VM01 Keepalived: Starting VRRP child process, pid=5650
Nov 23 15:34:04 VM01 Keepalived_vrrp: Using MII-BMSR NIC polling thread...
Nov 23 15:34:04 VM01 Keepalived_vrrp: Registering Kernel netlink reflector
Nov 23 15:34:04 VM01 Keepalived_vrrp: Registering Kernel netlink command channel
Nov 23 15:34:04 VM01 Keepalived_vrrp: Registering gratuitous ARP shared channel
Nov 23 15:34:04 VM01 Keepalived_vrrp: Opening file
'/opt/keepalived/etc/keepalived/keepalived.conf.master'.
Nov 23 15:34:04 VM01 Keepalived_vrrp: Configuration is using : 69247 Bytes
Nov 23 15:34:09 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Transition to MASTER STATE
Nov 23 15:34:09 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Transition to MASTER STATE
Nov 23 15:34:14 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Entering MASTER STATE
Nov 23 15:34:14 VM01 Keepalived_vrrp: VRRP_Group(VGM) Syncing instances to MASTER
state
Nov 23 15:34:14 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Entering MASTER STATE
Nov 23 15:34:14 VM01 Keepalived_vrrp: Netlink: error: File exists, type=(20), seq=1322033646,
pid=0
```

*表明 keepalived 已经启动成为内外网两个实例的 master 并且激活对后端 realserver 的健康检查

观察 lvs 从

```
Nov 23 15:34:07 VM01 Keepalived: Starting Healthcheck child process, pid=5373
Nov 23 15:34:07 VM01 Keepalived_healthcheckers: Using MII-BMSR NIC polling thread...
Nov 23 15:34:07 VM01 Keepalived_healthcheckers: Registering Kernel netlink reflector
Nov 23 15:34:07 VM01 Keepalived_healthcheckers: Registering Kernel netlink command
channel
Nov 23 15:34:07 VM01 Keepalived_vrrp: Using MII-BMSR NIC polling thread...
Nov 23 15:34:07 VM01 Keepalived_vrrp: Registering Kernel netlink reflector
Nov 23 15:34:07 VM01 Keepalived_vrrp: Registering Kernel netlink command channel
Nov 23 15:34:07 VM01 Keepalived_vrrp: Registering gratuitous ARP shared channel
```

```

Nov 23 15:34:07 VM01 Keepalived_healthcheckers: Opening file
'/opt/keepalived/etc/keepalived/keepalived.conf.backup'.
Nov 23 15:34:07 VM01 Keepalived_vrrp: Opening file
'/opt/keepalived/etc/keepalived/keepalived.conf.backup'.
Nov 23 15:34:07 VM01 Keepalived_vrrp: Configuration is using : 69213 Bytes
Nov 23 15:34:07 VM01 Keepalived_healthcheckers: Configuration is using : 13647 Bytes
Nov 23 15:34:07 VM01 Keepalived: Starting VRRP child process, pid=5374
Nov 23 15:34:07 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Entering BACKUP STATE
Nov 23 15:34:07 VM01 Keepalived_healthcheckers: Activating healthchecker for service
[223.202.22.116:80]
Nov 23 15:34:07 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Entering BACKUP STATE
Nov 23 15:34:07 VM01 Keepalived_healthcheckers: Activating healthchecker for service
[223.202.22.117:80]
*表明 keepalived 已经启动成为内外网两个实例的 backup 并且激活对后端 realserver 的健康
检查

```

3.观察 lvs 转发是否正常

用 ie 打开 <http://223.202.22.118>，出现 223.202.22.116 It works!

*为方便查看转发是否成功，我们分别在真实服务器的根目录用自己的 IP 编辑 index.html

在 lvs-master 上输入命令：watch ipvsadm -L

```
Every 2.0s: ipvsadm -L
```

```

IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  223.202.22.118:http wrr persistent 20
  -> 223.202.22.117:http          Route   100    0         0
  -> 223.202.22.116:http          Route   100    2         0

```

此时表明 223.202.22.116 这个真实服务器上已经有两个激活的链接

不断刷新页面后，会出现下边这种情况，表明根据调度算法真实服务器已经转发到 223.202.22.117，IE 出现 223.202.22.117 It works!

```
Every 2.0s: ipvsadm -L
```

```

IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  223.202.22.118:http wrr persistent 20
  -> 223.202.22.117:http          Route   100    2         0

```

```
-> 223.202.22.116:http          Route    100    0        0
```

4.观察后端真实服务器 1 上 apache 服务停止后的状态

LVS-master 的 message 上会出现以下情况

```
Nov 23 15:49:05 VM01 Keepalived_healthcheckers: TCP connection to [223.202.22.116:80] failed !!!
```

```
Nov 23 15:49:05 VM01 Keepalived_healthcheckers: Removing service [223.202.22.116:80] from VS [223.202.22.118:80]
```

```
Nov 23 15:49:05 VM01 Keepalived_healthcheckers: SMTP connection ERROR to [127.0.0.1:25].
```

ipvsadm 会出现以下情况

```
IP Virtual Server version 1.2.1 (size=4096)
```

```
Prot LocalAddress:Port Scheduler Flags
```

```
-> RemoteAddress:Port          Forward Weight ActiveConn InActConn
```

```
TCP  223.202.22.118:http wrr persistent 20
```

```
-> 223.202.22.117:http          Route    100    0        0
```

此时表明 lvs 调度器已经删除了后端的真实服务器 1

5.观察后端真实服务器 1 上 apache 服务由停止变成正常后的状态

LVS-master 的 message 上会出现以下情况

```
Nov 23 15:53:35 VM01 Keepalived_healthcheckers: TCP connection to [223.202.22.116:80] success.
```

```
Nov 23 15:53:35 VM01 Keepalived_healthcheckers: Adding service [223.202.22.116:80] to VS [223.202.22.118:80]
```

```
Nov 23 15:53:35 VM01 Keepalived_healthcheckers: SMTP connection ERROR to [127.0.0.1:25].
```

ipvsadm 会出现以下情况

```
IP Virtual Server version 1.2.1 (size=4096)
```

```
Prot LocalAddress:Port Scheduler Flags
```

```
-> RemoteAddress:Port          Forward Weight ActiveConn InActConn
```

```
TCP  223.202.22.118:http wrr persistent 20
```

```
-> 223.202.22.116:http          Route    100    0        0
```

```
-> 223.202.22.117:http          Route    100    0        0
```

此时表明 lvs 调度器已经又重新添加了后端的真实服务器 1

6.观察 lvs 从转发器 keepalived 失效的情况

在 lvs 从转发器 keepalived 上输入命令：`pkill keepalived`

此时 lvs 主转发器 message 不会发生任何变化

7.重启 lvs 从转发器 keepalived，观察 lvs 主转发器 keepalived 失效的情况
在 lvs 主转发器 keepalived 上输入命令：pkill keepalived

查看 lvs 从转发器 message

```
Nov 23 16:03:59 VM01 Keepalived_vrrp: VRRP_Group(VGM) Syncing instances to MASTER state
Nov 23 16:03:59 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Transition to MASTER STATE
Nov 23 16:03:59 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Entering MASTER STATE
Nov 23 16:04:04 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Entering MASTER STATE
```

此时表明 lvs 从转发器已经接管竞争成 master

8.恢复 lvs 主转发器 keepalived，查看两台转发器的状态

lvs 主转发器 message

```
Nov 23 16:10:21 VM01 Keepalived: Starting VRRP child process, pid=6469
Nov 23 16:10:21 VM01 Keepalived_healthcheckers: Opening file
'/opt/keepalived/etc/keepalived/keepalived.conf.master'.
Nov 23 16:10:21 VM01 Keepalived_healthcheckers: Configuration is using : 13681 Bytes
Nov 23 16:10:21 VM01 Keepalived_healthcheckers: Activating healthchecker for service
[223.202.22.116:80]
Nov 23 16:10:21 VM01 Keepalived_vrrp: Opening file
'/opt/keepalived/etc/keepalived/keepalived.conf.master'.
Nov 23 16:10:21 VM01 Keepalived_healthcheckers: Activating healthchecker for service
[223.202.22.117:80]
Nov 23 16:10:21 VM01 Keepalived_vrrp: Configuration is using : 69247 Bytes
Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Transition to MASTER STATE
Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Transition to MASTER STATE
Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Received lower prio advert,
forcing new election
Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Group(VGM) Syncing instances to MASTER
state
Nov 23 16:10:29 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Entering MASTER STATE
Nov 23 16:10:29 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Entering MASTER STATE
Nov 23 16:10:29 VM01 Keepalived_vrrp: Netlink: error: File exists, type=(20), seq=1322035823,
pid=0
```

lvs 从转发器 message

```
Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Received higher prio advert
Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Entering BACKUP STATE
Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Group(VGM) Syncing instances to BACKUP
```

state

Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Entering BACKUP STATE

Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Transition to MASTER STATE

Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Group(VGM) Syncing instances to MASTER state

Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Transition to MASTER STATE

Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Received higher prio advert

Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(LAN) Entering BACKUP STATE

Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Group(VGM) Syncing instances to BACKUP state

Nov 23 16:10:24 VM01 Keepalived_vrrp: VRRP_Instance(WAN) Entering BACKUP STATE

此时表明已经变成 lvs 主转发器的原来的 lvs 从转发器在收到原来 lvs 主转发器的更高的优先级通知后，降级并且再次变成 lvs 从转发器

9.关闭两台真实服务器的外网网卡后 lvs 转发器也能正确删除转发，在此不做赘于

10.关闭 lvs 主转发器的外网网卡后 lvs 从转发器也能正确接管，在此不做赘于