

```
#####
# File name   : 02 - 搭建LVS/DR-HA高可用负载均衡
# Description : 02 of Web2.0 大型网站架构试验
# Requirement : 01 - 系统及网络 虚拟环境搭建
#
#
# Copyright(C), fkoo, 2008, All Rights Reserved.
#
# Author: Far Young Chen / fkoo (fkoo.com@gmail.com)
# URL: http://www.fkoo.net
#
#####

# 说明: 本案例仅以DR (直接路由) 方式实现 LVS
#*****
#                               V02 安装 piranha LVS/DR
#*****
#-----安装LVS运行环境-----
# 用 nohup 命令启动安装脚本在后台运行, 同时将屏幕输出到当前目录下 nohup.out 文件中;
# 经测试用 LAMP--with-apxs2--enable-cli.sh 脚本安装, piranha-gui 服务能正常启动
# 说明: piranha-gui 依赖安装了 apache 和 php (apxs2 的 apache2handler 模式) 运行环境
[root@V02 ~]# nohup /mnt/hgfs/share/LAMP--with-apxs2.sh &
#-----
# 查看 shell 进程 PID: [root@V02 ~]# ps -al |grep sh
# 杀掉所有 shell 进程的PID, 中断安装: [root@V02 ~]# kill -9 PID
# 查看安装脚本执行进度 [root@V02 ~]# more nohup.out |grep OK

# nohup .sh 执行完毕后, 由于 LVS/DR 上不用跑 mysql 服务, 关闭之
service mysql stop
chkconfig mysql off

#-----安装LVS软件-----
# 安装 ipvsadm, 为 LVS 管理软件
# 编译 piranha 的 RPM 源码包, 为 LVS 功能软件
[root@V02 ~]#
rpm -ivh /mnt/hgfs/share/ipvsadm-1.24-11.i386.rpm
rpm -i /mnt/hgfs/share/piranha-0.8.4-9.3.el5.0.1.src.rpm
cd /usr/src/redhat/SPECS
rpmbuild -bp piranha.spec
cd /usr/src/redhat/BUILD/piranha
#-----
[root@T254 piranha]# mv /usr/src/redhat/BUILD/piranha/Makefile /usr/src/redhat/BUILD/piranhaMakefile.bak
[root@T254 piranha]# cp /mnt/hgfs/share/Makefile.piranha-gui /usr/src/redhat/BUILD/piranha/Makefile
```

```

# 说明: 安装php后 libphp5.so 模块文件所在位置为 /usr/local/apache/modules/
[root@V02 piranha]# vi Makefile
#Location of libraries // 修正默认 $(LIBDIR) 为 /usr/local
DEFAULT_LIBDIR = /usr/local
# Web Installation. // 修正 libphp5.so 的 modules 路径; httpd 命令路径; 注意 ln 前面的缩进是 Tab 键, 而不能是空格
ln -sf $(LIBDIR)/apache/modules $(HADIR)/modules
ln -sf /usr/local/apache/bin/httpd $(SBIN)/piranha_gui

#-----
[root@V02 piranha]# // 开始安装 piranha
make && make install
cd ../..
#rm -rf BUILD/piranha
#rm -rf SPECS/piranha.spec
#rm -rf SOURCES/piranha*

mv /etc/sysconfig/ha/conf/httpd.conf /etc/sysconfig/ha/conf/httpd.conf.bak
cp /mnt/hgfs/share/httpd.conf.piranha-gui /etc/sysconfig/ha/conf/httpd.conf

#----按需修改默认的 piranha-gui 服务-----
# 修改 piranha-gui 的 httpd.conf, 以匹配相应的设置
[root@V02 redhat]# vi /etc/sysconfig/ha/conf/httpd.conf
# 删除所有 LoadModule 项, 仅保留 php5_module 项即可
LoadModule php5_module          modules/libphp5.so

# 修改监听端口号 3636 为自定义的 6666
# 修改 用户名 / 用户组 为上述自定义的 fkoo / fkoogroup
Listen 6666
User fkoo
Group fkoogroup

<Directory /etc/sysconfig/ha/web>
# 修改 Options 为 -Indexes 从而关闭 web 目录的目录树浏览
# 设定此目录允许访问IP段为: 192.168.1.0/255.255.255.0
AllowOverride All
Options -Indexes Includes FollowSymLinks
Order deny,allow
Deny from all
Allow from 192.168.1.0/255.255.255.0
</Directory>

<Directory /etc/sysconfig/ha/web/secure>
AllowOverride All

```

```

Order deny,allow
# 设定此目录允许访问IP段为: 192.168.1.0/255.255.255.0
Deny from all
Allow from 192.168.1.0/255.255.255.0
AuthUserFile "/etc/sysconfig/ha/conf/piranha.passwd"
AuthGroupFile /dev/null
AuthName "access to the piranha web GUI"
AuthType Basic
<Limit GET>
    # 修改 <Limit GET> 下 require user piranha 为 上述预置设定的用户名 fkoo
    require user fkoo
</Limit>
</Directory>
#-----
cp /usr/sbin/piranha-passwd /mnt/hgfs/share/piranha-passwd

# 修改 piranha-gui 的默认登录用户名 piranha 为预置设定的用户名 fkoo
# 同时须修正 htpasswd 命令的实际路径为 /usr/local/apache/bin/htpasswd
# 同时修正 piranha.passwd 所属的用户名.用户组 为预置设定的 fkoo.fkoogroup
[root@V02 redhat]# vi /usr/sbin/piranha-passwd
    /usr/local/apache/bin/htpasswd -b $DEST/piranha.passwd fkoo "$password"
    /usr/local/apache/bin/htpasswd -c -b $DEST/piranha.passwd fkoo "$password"
chown fkoo.fkoogroup $DEST/piranha.passwd
#-----
# 将默认的 piranha-gui 服务登录用户名由 piranha 改为预置设定的 fkoo , 组名改为 fkoogroup
# 说明: 例中的数值 60,14198 依据各自系统可能不同; 使用者无需照样更改, 请保持您系统数值原样
cp /etc/passwd /mnt/hgfs/share/passwd.piranha-gui
cp /etc/shadow /mnt/hgfs/share/shadow.piranha-gui
cp /etc/group /mnt/hgfs/share/group.piranha-gui

[root@V02 redhat]# vi /etc/passwd
fkoo:x:60:60::/etc/sysconfig/ha:/dev/null
[root@V02 redhat]# vi /etc/shadow
fkoo:!!:14198::::::
[root@V02 piranha]# vi /etc/group
fkoogroup:x:60:

#-----启动 piranha-gui 服务, 并设置密码-----
# 至此, 如果安装成功则可以正常启动 piranha-gui 服务
# 说明: piranha-gui 服务不依赖 apache 服务, 无需同时开启 httpd 服务
[root@V02 redhat]# service piranha-gui start
#-----
# 修改预置设定用户 fkoo 的登录密码, 需要重复确认输入

```

```
# 说明: 设定密码成功后显示 Updating password for user fkoo
[root@V02 redhat]# piranha-passwd
# 说明: 查看设定的密码, 显示为已经过加密的格式
[root@V02 redhat]# cat /etc/sysconfig/ha/conf/piranha.passwd
fkoo:qeVL8lnMXL0dI
```

```
#-----编辑LVS的配置文件-----
cp /etc/sysconfig/ha/lvs.cf /mnt/hgfs/share/lvs.cf.V02

cp /etc/sysconfig/ha/lvs.cf /mnt/hgfs/share/lvs.cf.V03
```

```
[root@V02 redhat]# vi /etc/sysconfig/ha/lvs.cf
serial_no = 1
primary = 10.0.0.2
service = lvs
backup_active = 1
backup = 10.0.0.3
heartbeat = 1
heartbeat_port = 539
keepalive = 6
deadtime = 18
network = direct
debug_level = NONE
monitor_links = 1
virtual HTTPVS {
    active = 1
    address = 10.0.0.1 eth0:1
    vip_mask = 255.255.255.255
    port = 80
    send = "GET / HTTP/1.0\r\n\r\n"
    expect = "HTTP"
    use_regex = 0
    load_monitor = none
    scheduler = wrr
    protocol = tcp
    timeout = 6
    reentry = 15
    quiesce_server = 0
    server W11 {
        address = 10.0.0.11
        active = 1
        weight = 1
```

```

}
server W12 {
    address = 10.0.0.12
    active = 1
    weight = 1
}
server W13 {
    address = 10.0.0.13
    active = 1
    weight = 1
}
# 说明：初步测试，LVS 服务器本身不能作为 Realserver；调度进程会挂起在 LVS 服务器
# 因此注释掉相关配置
# server V02 {
#     address = 10.0.0.2
#     active = 1
#     weight = 1
# }
# server V03 {
#     address = 10.0.0.3
#     active = 1
#     weight = 1
# }
}

#-----管理 LVS 服务-----
# 用 chkconfig（系统服务启动管理工具）添加 pulse, piranha-gui 服务
# 说明：piranha-gui 服务提供 LVS 的 WEB 管理界面；pulse 为 LVS 的后台运行程序
# 设置 pulse 服务为默认开机启动
# 出于安全考虑，设置 piranha-gui 服务为默认开机不启动（需要通过 WEB 设置 LVS 时可手工启动服务）
# 启动 pulse, piranha-gui 服务
[root@V02 redhat]#
chkconfig --add piranha-gui
chkconfig --add pulse
chkconfig pulse on
chkconfig piranha-gui on
service pulse start
service piranha-gui restart
#-----
# 说明：以后台 vi /etc/sysconfig/ha/lvs.cf 方式修改 LVS 配置后，需要手工重载 pulse 服务，使配置生效；
# 手工重启 piranha-gui 服务后，WEB 管理界面获得更新的显示
[root@V02 redhat]#
service pulse reload

```

```

service piranha-gui restart
service ipvsadm save
ipvsadm -Ln

#-----LVS 路由转发-----
# 检查 ip_forward 路由转发功能是否开启; 0 表示关闭, 1 表示开启
[root@V02 redhat]# cat /proc/sys/net/ipv4/ip_forward
1
#-----
# 若 ip_forward 为 0 关闭; 修改并激活设置为开启
[root@V02 redhat]# vi /etc/sysctl.conf
net.ipv4.ip_forward = 1

# 使 sysctl.conf 配置立即激活生效
[root@V02 redhat]# (二选一即可)
sysctl -p
echo "1"> /proc/sys/net/ipv4/ip_forward

#*****
#                               V03 安装 piranha LVS
#*****
# 基本与 V02 相同, 只需对调 lvs.cf 中的 primary 和 backup 的 IP 即可
#-----编辑LVS的配置文件-----
[root@V02 redhat]# vi /etc/sysconfig/ha/lvs.cf
primary = 10.0.0.3
backup = 10.0.0.2

# 基本与 V02 相同, 只需对调 lvs.cf 中的 primary 和 backup 的 IP 即可
#-----编辑LVS的配置文件-----
[root@V02 redhat]# vi /etc/sysconfig/ha/lvs.cf
primary = 10.0.0.3
backup = 10.0.0.2

ping -I 10.0.0.1 192.168.1.x
#*****
#                               W11 安装 LVS          Realserver 支持
#*****
nohup /mnt/hgfs/share/LAMP--without-apxs2.sh &

#-----路由转发-----
# 检查 ip_forward 路由转发功能是否开启; 0 表示关闭, 1 表示开启
[root@V02 redhat]# cat /proc/sys/net/ipv4/ip_forward
1

```

```

#-----
# 若 ip_forward 为 0 关闭; 修改并激活设置为开启
[root@V02 redhat]# vi /etc/sysctl.conf
net.ipv4.ip_forward = 0

# 使 sysctl.conf 配置立即激活生效
[root@V02 redhat]# (二选一即可)
sysctl -p
echo "0"> /proc/sys/net/ipv4/ip_forward

# 目的: Realserver 不主动向默认网关 T254 发送请求, T254 不将 VIP 的 MAC 替换为 Realserver 的 MAC
#-----为响应 VIP 的请求做哑设备的配置-----
# 永久 (重启后不丢失) 新建哑设备 dummy0, 并将 漂移 VIP 10.0.0.1 绑定其上
# 在 W11 上添加到 漂移 VIP 10.0.0.1 的永久路由 (重启后不丢失) 到哑设备 dummy0 上
[root@W11 ~]# vi /etc/sysconfig/network-scripts/ifcfg-dummy0
DEVICE=dummy0
BROADCAST=10.0.0.1
IPADDR=10.0.0.1
NETMASK=255.255.255.255
ONBOOT=yes
[root@W11 ~]# vi /etc/sysconfig/network-scripts/route-dummy0
10.0.0.1/32 via 0.0.0.0 dev dummy0
# 重启 network 服务, 使 dummy0 设备生效
[root@W11 ~]# service network restart
#-----
# 此处的命令与上面效果相同; 不同在于为即时生效命令, 重启后丢失
[root@W11 ~]#
ifconfig dummy0 10.0.0.1 broadcast 10.0.0.1 netmask 255.255.255.255 up
route add -host 10.0.0.1 dev dummy0

#-----关闭 Realserver 的被动 ARP广播响应-----
# 在 /etc/sysctl.conf 中添加
[root@W11 ~]# vi /etc/sysctl.conf
net.ipv4.conf.dummy0.arp_ignore = 1
net.ipv4.conf.dummy0.arp_announce = 2
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
# 应用修改的 sysctl.conf 配置, 使生效
[root@W11 ~]# sysctl -p
#-----
# 此处的命令与上面效果相同; 不同在于为即时生效命令, 重启后丢失
[root@W11 ~]#

```

```

echo 1 > /proc/sys/net/ipv4/conf/dummy0/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/dummy0/arp_announce
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce

#-----安装配置 arptables_jf 直接路由服务-----
# 说明: 使得 Realserver 将 LVS DR 转发来的请求, 以其IP返回给客户端
# 将 arptables_jf 设置为开机自动启动
[root@W11 ~]#
rpm -ivh /mnt/hgfs/share/arptables_jf-0.0.8-13.fc10.i386.rpm
chkconfig arptables_jf on
#-----
# 清空所有的链
# 丢弃目的地址为 VIP 10.0.0.1 的包
# 将返回给 VIP 10.0.0.1 的数据包源地址改为本 Realserver 的 IP, 直接返回给客户端
# 保存 arptables
[root@W11 ~]#
arptables --flush
arptables -A IN -d 10.0.0.1 -j DROP
arptables -A OUT -d 10.0.0.1 -j mangle --mangle-ip-s 10.0.0.11
service arptables_jf save
#-----
# 列出当前活动的 arptables
# 显示已保存的 arptables 配置文件
[root@W11 ~]#
arptables --list
cat /etc/sysconfig/arptables

#-----定制测试页面-----
# 在 apache 的 DocumentRoot 所在目录下新建标题和页面内容为 W11 的页面
[root@W11 ~]# vi /usr/local/apache/htdocs/index.html
<html>
<head>
<title> W11 </title>
<meta http-equiv="refresh" content="10">
</head>
<body>
W11
</body>
</html>

#*****
# W12, W13 安装 LVS Realserver 支持

```



```
#####
```

```
#####
```

```
# Realserver R210 / R211 / R212 分别做以下各自的配置 #
```

```
[root@R210 share]# vi /var/www/index.html
```

```
<html>
<head>
  <title> R210 </title>
  <meta http-equiv="refresh" content="10">
</head>
<body>
```

```
R210
</body>
```

```
</html>
```

```
[root@R211 share]# 在R211上配置Realserver参数
```

```
rpm -ivh arptables_jf-0.0.8-9.i386.rpm
```

```
chkconfig --level 2345 arptables_jf on
```

```
arptables --flush
```

```
arptables -A IN -d 10.10.10.200 -j DROP
```

```
arptables -A OUT -d 10.10.10.200 -j mangle --mangle-ip-s 10.10.10.211
```

```
service arptables_jf save
```

```
arptables --list
```

```
cat /etc/sysconfig/arptables
```

```
[root@R211 share]# vi /var/www/index.html
```

```
<html>
<head>
  <title> R211 </title>
  <meta http-equiv="refresh" content="10">
</head>
<body>
```

```
R211
</body>
```

```
</html>
```

```
[root@R212 share]# 在R212上配置Realserver参数
```

```
rpm -ivh arptables_jf-0.0.8-9.i386.rpm
```

```
chkconfig --level 2345 arptables_jf on
```

```
arptables --flush
```

```
arptables -A IN -d 10.10.10.200 -j DROP
```

```
arptables -A OUT -d 10.10.10.200 -j mangle --mangle-ip-s 10.10.10.212
```

```
service arptables_jf save
```

```
arptables --list
```

```
cat /etc/sysconfig/arptables
```

```
[root@R212 share]# vi /var/www/index.html
<html>
<head>
<title> R212 </title>
<meta http-equiv="refresh" content="10">
</head>
<body>
R212
</body>
</html>
[root@Realservers share]# 表示在每个Realserver上做相同的操作
[root@Realservers share]# netstat -nr
Kernel IP routing table
Destination      Gateway          Genmask          Flags      MSS Window  irtt Iface
10.10.10.200     0.0.0.0          255.255.255.255 UH          0 0        0 lo
[root@Realservers share]# cat /etc/sysconfig/arpables
[0:0] -A IN -d 10.10.10.200 -j DROP
[0:0] -A OUT -d 10.10.10.200 -j mangle --mangle-ip-s 10.10.10.210

# 测试时, 开启的浏览器客户端数量不要为Realservers数量的倍数, 可能会不轮询更新, 而保持现有Realserver Session
```

```
#####
# 用 nohup 命令启动安装脚本在后台运行, 同时将屏幕输出到当前目录下 nohup.out 文件中;
[root@W11 ~]# nohup /mnt/hgfs/share/LAMPinstall--without-apxs2--enable-cli--without-mysql.sh &
#-----
# 查看 shell 进程 PID: [root@V02 ~]# ps -al |grep sh
# 杀掉所有 shell 进程的PID, 中断安装: [root@V02 ~]# kill -9 PID
# 查看安装脚本执行进度 [root@V02 ~]# more nohup.out |grep OK
```

```
#-----EOF-----
```

# 下一篇: 02 - 搭建LVS-HA高可用负载均衡 / 02 of Web2.0 大型网站架构试验

```
[root@W11 ~]# nohup /mnt/hgfs/share/LAMPinstall-no.sh &
[root@W11 ~]# ls /usr/local/php-fcgi/bin/
pear peardev pecl php php-cgi php-config phpize
[root@W11 ~]# tail -25 nohup.out
Installing PHP SAPI module:      cgi
```

```
[root@W13 ~]# nohup /mnt/hgfs/share/LAMPinstall-ac.sh &
```

```
[root@W13 ~]# ls /usr/local/php-fcgi/bin/
php-config  phpize
[root@W13 ~]# tail -25 nohup.out
Installing PHP SAPI module:      apache2handler
-----
[root@V02 ~]# nohup /mnt/hgfs/share/LAMPinstall-a.sh &
[root@V02 ~]# ls /usr/local/php-fcgi/bin/
pear  peardev  pecl  php  php-config  phpize
[root@V02 ~]# tail -35 nohup.out
Installing PHP SAPI module:      apache2handler
-----
[root@V03 ~]# nohup /mnt/hgfs/share/LAMPinstall-c.sh &
[root@V03 ~]# ls /usr/local/php-fcgi/bin/
php-cgi  php-config  phpize
[root@V03 ~]# tail -18 nohup.out
Installing PHP SAPI module:      cgi
-----
```

```
# 为支持 php 页面编辑 Apache 配置文件, 添加 AddType 和 index.php 语句
# vi /usr/local/apache/conf/httpd.conf
vi /etc/httpd/httpd.conf
```

```
cp /etc/httpd/httpd.conf /mnt/hgfs/share/httpd.conf.fcgid.php
```

```
mv /etc/httpd/httpd.conf /etc/httpd/httpd.conf.apache.php.bak
cp /mnt/hgfs/share/httpd.conf.apache.php /etc/httpd/httpd.conf
service httpd restart
```

```
<IfModule mime_module>
    AddType application/x-httpd-php .php
    AddType application/x-httpd-php-source .phps
</IfModule>
```

```
<IfModule dir_module>
    DirectoryIndex index.php index.html index.htm
</IfModule>
```

```
# 安装 CGI/FastCGI - mod_fcgid 模块支持
cp /mnt/hgfs/share/mod_fcgid.2.2.gz /tmp/
cd /tmp
tar zxvf mod_fcgid.2.2.gz
```

```

cd mod_fcgid.2.2
make top_dir=/usr/local/apache
make install top_dir=/usr/local/apache
cd ..
rm -rf mod_fcgid.2.2*

LoadModule fastcgi_module modules/mod_fastcgi.so
<IfModule mod_fastcgi.c>
    AddHandler fastcgi-script .php .fcgi .fcg .fpl
</IfModule>

LoadModule fcgid_module modules/mod_fcgid.so

<IfModule mod_fcgid.c>
    AddHandler fcgid-script .php .py .pl .fcgi
    SocketPath /tmp/fcgid.sock
    IdleTimeout 600
    ProcessLifeTime 3600
    MaxProcessCount 8
    DefaultMinClassProcessCount 3
    DefaultMaxClassProcessCount 3
    IPCConnectTimeout 20
    IPCCommTimeout 48
</IfModule>

<Directory "/usr/local/httpd/htdocs">
    # remarked this!
    # Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    # new insert for fcgid
    FCGIWrapper /usr/local/php-fcgi/bin/php-cgi .php
    FCGIWrapper /usr/local/php-fcgi/bin/php-cgi .php5
    Options ExecCGI SymLinksIfOwnerMatch
    Allow from all

</Directory>

# 安装 CGI/FastCGI - mod_fastcgi 模块支持
cp /mnt/hgfs/share/mod_fastcgi-2.4.6.tar.tar /tmp/
cd /tmp
tar xvf mod_fastcgi-2.4.6.tar.tar
cd mod_fastcgi-2.4.6

```

```
cp Makefile.AP2 Makefile
make top_dir=/usr/local/apache
make install top_dir=/usr/local/apache
cd ..
rm -rf mod_fastcgi-2.4.6*
```