目录

[一、LVS简介 1](#_Toc457572990)

一、LVS简介  
LVS是章文嵩博士发起和领导的优秀的集群解决方案，  
许多商业的集群产品，比如RedHat的Piranha，TurboLinux公司的Turbo Cluster等，都是基于LVS的核心代码的。  
在现实的应用中，LVS得到了大量的部署，请参考 http://www.linuxvirtualserver.org/deployment.html  
关于Linux LVS的工作原理和更详细的信息，请参考 http://www.linuxvirtualserver.org。  
  
  
二、配置实例  
本例由两台物理服务器实现;  
使用的服务包括有三种:  
 telnet;  
 WWW;  
 图片下载;  
  
  
1. 网络拓扑结构  
如图1所示:  
                 VS1: 192.168.175.44  
                  |                                |  
 RS1: 192.168.175.45            RS2: 192.168.175.46  
  
  
三、服务器硬件配置  
首先说明，虽然本文的测试环境中用的是2台相同配置的服务器，但LVS并不要求集群中的服务器规格划一，  
相反，可以根据服务器的不同配置和负载情况，调整负载分配策略，充分利用集群环境中的每一台服务器。  
  
  
这2台服务器中，rs1即是提供服务的服务器，也作为虚拟服务器（即负载平衡服务器），  
负责将用户的访问请求转发到集群内部的rs1,rs2，然后由rs1,rs2分别处理。  
  
  
client为客户端测试机器，可以为任意操作系统。  
  
  
2台服务器的操作系统和网络配置分别为：  
vs1: RedHat 4.1.2-54, Kernel 2.6.18-348.el5  
vs1: eth0 192.168.0.1  
vs1: eth0:101 192.168.175.44  
  
  
rs1: Red Hat 4.1.2-54,  Kernel  2.6.18-348.el5  
rs1: eth0 192.168.175.45  
rs1: dummy0 192.168.0.101  
  
  
rs2: Red Hat 4.1.2-54, Kernel  2.6.18-348.el5  
rs2: eth0 192.168.175.46  
rs2: dummy0 192.168.0.101  
  
  
client: Windows xp  
client: eth0 10.2.9.99  
  
  
其中，192.168.175.44是允许用户访问的IP。  
  
四、LVS集群配置  
1. 在配置之前，先检查内核是否打了LVS补丁:  
如果你使用的是RedHat自带的内核或者从RedHat下载的内核版本，已经预先打好了LVS的补丁。  
这可以通过查看  
/usr/src/linux/net/目录下有没有几个ipvs开头的文件来判断：  
如果有，则说明已经打过补丁。  
  
查看内核中是否已经加载ip\_vs模块  
[root ~]# lsmod | grep ip\_vs  
  
  
查询内核版本(现在的版本一般都会满足)  
[root ~]# uname -r  
2.6.18-348.el5  
  
  
创建链接,将内核模块做软连接至/usr/src/目录下, 并改名为/usr/src/linux/  
这样做的好处是，在编译keepalived的时候可以直接找到内核文件，而不用指定内核路径，  
如果不做此项操 作，则在编译时需指定内核路径  
[root ~]# ln -s /usr/src/kernels/2.6.18-348.el5-x86\_64/ /usr/src/linux  
  
2. 对两台物理服务器对建立一个目录来下载文件并安装  
[root ~]# mkdir /opt/lvs  
[root ~]# cd /opt/lvs  
  
下载 ipvsadm-1.24-8.1.x86\_64.rpm 和 piranha-0.8.4-11.x86\_64.rpm  
到目录 /opt/lvs  
[root lvs]# ls  
ipvsadm-1.24-8.1.x86\_64.rpm  piranha-0.8.4-11.x86\_64.rpm  
  
安装这两个模块:  
方式一: 用rpm模块安装  
[root lvs]# rpm -ivh ipvsadm-1.24-8.1.x86\_64.rpm   
Preparing...                ########################################### [100%]  
   1:ipvsadm                ########################################### [100%]  
[root lvs]# rpm -ivh piranha-0.8.4-11.x86\_64.rpm   
error: Failed dependencies:  
        php is needed by piranha-0.8.4-11.x86\_64  
[root lvs]# rpm -ivh  --nodeps piranha-0.8.4-11.x86\_64.rpm   
Preparing...                ########################################### [100%]  
   1:piranha                ########################################### [100%]  
  
方式二: 用yum安装  
先检查下yum是否可用，如果可用，就用yum安装  
[root lvs]# yum list  
[root lvs]# yum -y install ipvsadm piranha  
Loaded plugins: fastestmirror, security  
Loading mirror speeds from cached hostfile  
...  
Total download size: 5.4 M  
Downloading Packages:  
(1/5): ipvsadm-1.24-13.el5.x86\_64.rpm                                                |  34 kB     00:00       
(2/5): php-common-5.1.6-43.el5\_10.x86\_64.rpm                                         | 155 kB     00:17       
(3/5): piranha-0.8.4-25.el5\_9.1.x86\_64.rpm                                           | 717 kB     00:01       
(4/5): php-cli-5.1.6-43.el5\_10.x86\_64.rpm                                            | 2.2 MB     00:00       
(5/5): php-5.1.6-43.el5\_10.x86\_64.rpm                                                | 2.4 MB     00:00       
-------------------------------------------------------------------------------------  
Total                                                                                46 kB/s | 5.4 MB     01:59       
Running rpm\_check\_debug  
Running Transaction Test  
Finished Transaction Test  
Transaction Test Succeeded  
Running Transaction  
  Installing     : php-common                                                         1/5   
  Installing     : php-cli                                                            2/5   
  Installing     : php                                                                3/5   
  Installing     : ipvsadm                                                            4/5   
  Installing     : piranha                                                            5/5   
  
  
Installed:  
  ipvsadm.x86\_64 0:1.24-13.el5                                                        piranha.x86\_64 0:0.8.4-25.el5\_9.1                                                           
  
  
Dependency Installed:  
  php.x86\_64 0:5.1.6-43.el5\_10    php-cli.x86\_64 0:5.1.6-43.el5\_10                    php-common.x86\_64 0:5.1.6-43.el5\_10                            
  
  
Complete!  
  
3. 创建虚拟网口:  
[root lvs]# cd /etc/sysconfig/network-scripts/  
[root network-scripts]# vim ifcfg-lo\:0  
DEVICE=lo:0  
BOOTPROTO=static  
IPADDR=192.168.175.44  
NETMASK=255.255.255.255  
ONBOOT=yes  
ARP=no  
【wq】  
[root network-scripts]# service network restart  
  
4. 修改系统默认配置:  
[root ~]# vim /etc/sysctl.conf  
【添加下列字段】  
net.ipv4.tcp\_max\_syn\_backlog = 20480  
net.ipv4.tcp\_synack\_retries = 1  
net.ipv4.tcp\_syn\_retries = 1  
net.core.rmem\_max = 16777216  
net.core.rmem\_default = 16777216  
net.core.wmem\_max = 16777216  
net.core.netdev\_max\_backlog = 262144  
net.core.somaxconn = 262144  
net.ipv4.conf.all.arp\_ignore = 1  
net.ipv4.conf.all.arp\_announce = 2  
net.ipv4.conf.lo.arp\_ignore = 1  
net.ipv4.conf.lo.arp\_announce = 2  
net.ipv4.conf.lo.rp\_filter = 0  
net.ipv4.tcp\_tw\_reuse = 1  
net.ipv4.tcp\_tw\_recycle = 1  
net.ipv4.tcp\_max\_tw\_buckets = 1200000  
net.ipv4.tcp\_max\_orphans = 262144  
net.ipv4.tcp\_rmem = 4096 87380 8388608  
net.ipv4.tcp\_wmem = 4096 65536 8388608  
net.ipv4.ip\_local\_port\_range = 1024 65000  
net.ipv4.tcp\_syncookies = 1  
net.ipv4.tcp\_window\_scaling = 0  
net.ipv4.tcp\_sack = 0  
net.ipv4.tcp\_timestamps = 0  
kernel.panic = 5  
net.ipv4.conf.all.arp\_ignore = 1  
net.ipv4.conf.all.arp\_announce = 2  
net.ipv4.conf.lo.arp\_ignore = 1  
net.ipv4.conf.lo.arp\_announce = 2  
net.ipv4.conf.lo.rp\_filter = 0  
net.ipv4.conf.eth0.rp\_filter = 2  
【wq】  
[root ~]# sysctl -p  
  
5. 修改LVS的配置:  
[root ~]# cd /etc/sysconfig/ha/  
[root ha]# vim lvs.cf  
serial\_no = 39  
primary = 192.168.175.45  
service = lvs  
backup\_active = 1  
backup = 192.168.175.46  
heartbeat = 1  
heartbeat\_port = 539  
keepalive = 3  
deadtime = 9  
network = direct  
debug\_level = NONE  
monitor\_links = 0  
syncdaemon = 0  
virtual XP {  
     active = 1  
     address = 192.168.175.44 eth0:1  
     vip\_nmask = 255.255.255.0  
     port = 80  
     send = "GET / HTTP/1.0\r\n\r\n"  
     expect = "HTTP"  
     use\_regex = 0  
     load\_monitor = none  
     scheduler = wlc  
     protocol = tcp  
     timeout = 5  
     reentry = 5  
     quiesce\_server = 0  
     server xp1 {  
         address = 192.168.175.45  
         active = 1  
         weight = 1  
     }  
     server xp2 {  
         address = 192.168.175.46  
         active = 1  
         weight = 1  
     }  
}  
【wq】  
  
6. 在两台物理机上分别启动心跳服务:  
[root ~]#  service pulse start  
Starting pulse:                                            [  OK  ]  
  
  
这时，能在主物理机192.168.175.45上看到负载均衡网卡已启动,  
而192.168.175.46上是没有这个网卡的:  
[root ~]# ifconfig  
eth0      Link encap:Ethernet    
          inet addr:192.168.175.45    
          ...  
eth0:1    Link encap:Ethernet  HWaddr D4:3D:7E:70:8E:54    
          inet addr:192.168.175.44  Bcast:192.168.175.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          Memory:df9a0000-df9c0000  
  
当主物理机的服务关闭时，在192.168.175.46上看到网卡已启动:  
[root ~]# ifconfig  
eth0      Link encap:Ethernet  HWaddr D4:3D:7E:70:8E:54    
          inet addr:192.168.175.46  
          ...  
eth0:1    Link encap:Ethernet  HWaddr D4:3D:7E:70:8E:54    
          inet addr:192.168.175.44  Bcast:192.168.175.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          Memory:df9a0000-df9c0000  
  
  
7. 查看虚拟服务器状态:  
[root ~]# ipvsadm -ln  
IP Virtual Server version 1.2.1 (size=4096)  
Prot LocalAddress:Port Scheduler Flags  
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn  
TCP  192.168.175.44:80 wlc  
  -> 192.168.175.46:80               Route   1      0          0           
  -> 192.168.175.45:80               Local   1      0          0  
  
将服务设成开机自启动:  
[root ~]# chkconfig --level 345 pulse on  
  
三、LVS测试  
查看nginx的访问日志:  
[root logs]# tail -0f logs/access.log  
  
如果两台机器的访问是交替的，就说明负载均衡生效了；