www.linuxidc.com

高可用性 Linux 集群实现

摘要
随着网络的普及，对网络的需求日益曾强，而服务商提供的服务的质量也需要提高，以前单服务器的服务已经很难满足人们的需要。搭建高可用性的服务集群迫在眉睫，很多大公司（IBM，SUN 等等）都提供了一系列的商业方案，而我们采用的是开源阵营里面的 LVS 和 HA 来实现，在经济上是绝对有可取之处，技术上好不逊色。

术语
HA：High Availability，高可用性。
Linux Director:整个集群的入口机器，在最终用户和真实服务器之间转发数据额作用，以下简称为 LD。
End User：最终的用户（客户端），以下简称客户。
Real Server：真正提供服务（httpd，ftp，等等）的服务器，以下简称为 RS。
Virtual IP address：VIP，虚拟 IP，最终用户访问使用的 IP。
Real IP address：RIP，真实 IP，服务器真实的 IP 地址。
Heartbeat：心跳检测，实现 HA 的一个软件。

第一部分：LVS
LVS 简介
Linux Virtual Server Project(LVS)是一个由章文嵩先生提起的开源项目。主要是通过 linux 内核在第四层实现数据交换，在真实的服务器群中实现简单的负载均衡。LVS 只能使用在 Linux 中，但是真实的服务器可以是任意的支持 TCP 或 UDP 的操作系统。

LVS 工作机制
四层交换
何谓四层交换？当 LD 接收到 TCP 或者 UDP 数据时，通过一定的算法，将数据转发到真实的服务器中。在我们这次实验中，使用的是 wlc（加权最少连接数）算法。

数据转发机制：
在 LVS 中，有以下三种数据转发机制：
Network Address Translation(NAT):NAT 机制。NAT 这中方法在网络中运用很多，通过 IP 伪装就可以实现。在客户发起一个请求后，LD 接收到，然后转发到 RS，RS 再通过 LD 回复给客户，LD 在这里就要起一个网关的作用。

Direct Routing：DR 机制。客户请求不做任何修改直接转发到 RS。使用 DR 时，RS 必须要接收 VIP 的数据包，可以通过虚假接口或包过滤来重定向数据包。当 RS 接收到请求后，直接转发数据给客户，不再通过 LD，在一定程度上，可以减轻 LD 的负担，我们在实验中也采用的 DR 机制。

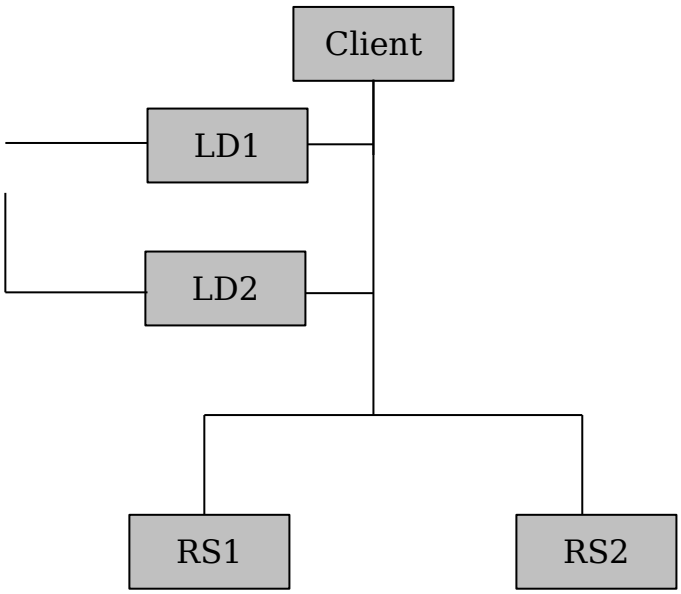
IP-IP Encapsulation(Tunnelling):隧道机制。类似于 DR，只是数据的封装形式不一样，隧道采用的是 ip 包，而 DR 采用的是以太网的帧。这也决定了，如果不同的网络，就必须采用隧道机制了。

调度算法：

Linux公社(LinuxIDC.com) 是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

在 LVS 中，所有的算法都是以模块的方式安装在内核的。最长用的算法有：
rr：循环算法，在 RS 之间一个接一个的分配任务
wrr：加权循环，通过权值来标明 RS 的性能，权值越高，性能越高
lc：最少连接数
wlc：加权最少连接数
当然，还有其它的一些算法，可以根据实际情况选择，我们在实验中采用的是 wlc 算法。
关于更多的 LVS 的内容，在后面的参考资源里面有更多的文章。

以上是整个集群相关的一些基础知识，下面是对整个集群的实现过程。集群的网络连接如下图：



LD 和 RS 的 IP 设置如下：

```
LD1 primary: RIP eth0  172.26.46.110
          VIP eth0:0  172.26.46.119
     RIP eth1 192.168.0.1 LD1 与 LD2 间心跳检测传输数据
LD2 backup: RIP eth0 172.26.46.111
     VIP eth0:0 172.26.46.119
     RIP eth1  192.168.0.2  功能同 LD1 的 eth1
RS1: RIP eth0 172.26.46.80
     VIP lo:0 172.26.46.119
RS2: RIP eth0 172.26.46.81
     VIP lo:0 172.26.46.119
```

LVS 的安装
在一些 Linux 的发型版中，在内核中已经集成了 lvs 的模块，大家可以查阅相关资料获取自己使用的发型版的信息。在实验过程中，LD1 安装的是 RedHat9，LD2 安装的是 Debian sarge 3.1，内核都是采用的是 2.4.20。RS 安装的都是 Fedora Core 3，内核是 2.6.9。下面我们开始安装：
1.下载并解压 kernel 和 LVS，

```
wget http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.20.tar.bz2
wget http://www.linux-vs.org/software/kernel-2.4/ipvs-1.0.10.tar.gz
tar jxvf linux-2.4.20.tar.bz2
tar zxvf ipvs-1.0.tar.gz
```

2.给解压出来的新内核增加补丁，

```
patch -p1 < $DOWNLOAD_PATH/ipvs-1.0.10/contrib/patches/hidden-2.4.20pre10-1.diff  (arp for LVS-DR /LVS-Tun)
```

3.配置kernel

1.make mrproper

2.make menuconfig

在make menuconfig的时候，请选种以下相关选项：

```
Networking options  --->
     IP: Netfilter Configuration  --->
     IP: Virtual Server Configuration  --->
```

网络的配置信息如下：

```
<*> Packet socket
 [ ] Packet socket: mmapped IO
 [*] Kernel/User netlink socket
 [*] Routing messages
 <*> Netlink device emulation
 [*] Network packet filtering (replaces ipchains)
 [*] Network packet filtering debugging
 [*] Socket Filtering
 <*> Unix domain sockets
 [*] TCP/IP networking
 [ ]   IP: multicasting
 [*]   IP: advanced router
 [*]     IP: policy routing
 [*]       IP: use netfilter MARK value as routing key
 [*]       IP: fast network address translation
 [*]     IP: equal cost multipath
 [*]     IP: use TOS value as routing key
 [*]     IP: verbose route monitoring
 [*]     IP: large routing tables
 [*]   IP: kernel level autoconfiguration
 [ ]     IP: BOOTP support
 [ ]     IP: RARP support
 <M>   IP: tunneling
 < >   IP: GRE tunnels over IP
 [ ]   IP: multicast routing
 [ ]   IP: ARP daemon support (EXPERIMENTAL)
 [ ]   IP: TCP Explicit Congestion Notification support
 [ ]   IP: TCP syncookie support (disabled per default)
   IP: Netfilter Configuration  --->
   IP: Virtual Server Configuration  --->
 < >   The IPv6 protocol (EXPERIMENTAL)
 < >   Kernel httpd acceleration (EXPERIMENTAL)
 [ ] Asynchronous Transfer Mode (ATM) (EXPERIMENTAL)
```

LVS 的配置信息如下：

```
<M> virtual server support (EXPERIMENTAL)
 [*]   IP virtual server debugging (NEW)
 (12)   IPVS connection table size (the Nth power of 2) (NEW)
 --- IPVS scheduler
 <M>   round-robin scheduling (NEW)
 <M>   weighted round-robin scheduling (NEW)
 <M>   least-connection scheduling scheduling (NEW)
 <M>   weighted least-connection scheduling (NEW)
 <M>   locality-based least-connection scheduling (NEW)
 <M>   locality-based least-connection with replication
scheduling (NEW)
 <M>   destination hashing scheduling (NEW)
 <M>   source hashing scheduling (NEW)
 --- IPVS application helper
 <M>   FTP protocol helper (NEW)
```

Ip 过滤的配置如下：

```
<M> Connection tracking (required for masq/NAT)
 <M>   FTP protocol support
 <M> Userspace queueing via NETLINK (EXPERIMENTAL)
 <M> IP tables support (required for filtering/masq/NAT)
 <M>   limit match support
 <M>   MAC address match support
 <M>   netfilter MARK match support
 <M>   Multiple port match support
 <M>   TOS match support
 <M>   Connection state match support
 <M>   Unclean match support (EXPERIMENTAL)
 <M>   Owner match support (EXPERIMENTAL)
 <M>   Packet filtering
 <M>     REJECT target support
 <M>     MIRROR target support (EXPERIMENTAL)
 <M>   Full NAT
 <M>     MASQUERADE target support
 <M>     REDIRECT target support
 <M>   Packet mangling
 <M>     TOS target support
 <M>     MARK target support
 <M>   LOG target support
 < > ipchains (2.2-style) support
 < > ipfwadm (2.0-style) support
```

说明：以上配置信息只是节选了 LVS 必须需要的部分，其它模块请根据自己的实际情况选择。

3.make dep
4.make clean
5.make bzImage
6.make modules
7.make modules install

4.更新 GRUB

cp /usr/src/linux2.4.20/arch/i386/boot/bzImage /boot/vmlinuz-2.4.20-LVS

cp /usr/src/linux2.4.20/System.map /boot/System.map （覆盖原来的 System.map 文件）

```
  vi /boot/grub/grub.conf
    title kernel-2.4.20-LVS
        root(hd0,0)
        kernel /boot/vmlinuz-2.4.20-LVS ro root=LABEL=/
        hdb=ide-scsi
        initrd /boot/initrd-2.4.20.img
```

5.Reboot，使用有 LVS 的新内核登陆。

说明：如果你需要再多台机器上安装 LVS，可以直接把编译好了的 2.4.20 内核的文件打包，操作如下：

```
  tar czf linux2.4.20-lvs.tgz /usr/src/linux2.4.20
```

然后就接着

```
    make modules
    make modules install
```

更新 GRUB，操作如前。

6.登陆后，安装 ipvs，进入 ipvs 解压目录，执行

```
    make all
    make install
```

7.安装结束，输入 ipvsadm，如果出现下面信息，则安装成功。

```
  IP Virtual Server version 1.0.9 (size=4096)
  Prot LocalAddress:Port Scheduler Flags
    ->RemoteAddress:Port          Forward Weight
    ActiveConn InActConn
```

LVS/DR 的配置

这是 LVS 的核心工作部分，配置步骤如下：

1. 在 LD 上面的配置脚本如下：

```
#!/bin/bash
#file:conf_ld
#This script is written by XiaoKang.Leng
#2005-sep-04
#Used for the LVS director config

PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

#config the eth0,if you config eth0,comment this
#ifconfig eth0 172.26.46.110 netmask 255.255.0.0 broadcast
172.26.46.254 up

#Config the eth1,prepare for the heartbeat
ifconfig eth1 192.168.0.1 netmask 255.255.255.0 broadcast
192.168.0.254 up

#Use the netstat -rn command to check the route table,if the route
table have
#not the 172.26.46.0 net,add it,
route add -net 172.26.46.0 netmask 255.255.255.0 dev eth0

#Config the VIP for the eth0:0,the netmask must be 255.255.255.255 or
0xffffffff
ifconfig eth0:0 172.26.46.119 netmask 255.255.255.255 broadcast
172.26.46.119 up

#Add a host route for the eth:0
```

```
route add -host 172.26.46.119 dev eth0:0

#Stop the ip_forward for the secure reason,if you need ip_forward
,you can
#ENABLE it,1 for ENABLE ,0 for DISABLE
echo 0  > /proc/sys/net/ipv4/ip_forward

#Because in the LVS/DR, direcotr is not a gw for realserver ,so we
use icmp
#to redirects on, 1 for on ,0 for off
echo "1" > /proc/sys/net/ipv4/conf/all/send_redirects
echo "1" > /proc/sys/net/ipv4/conf/default/send_redirects
echo "1" > /proc/sys/net/ipv4/conf/eth0/send_redirects

#Check the icmp redirects and the ip_forward is right
cat /proc/sys/net/ipv4/conf/all/send_redirects
cat /proc/sys/net/ipv4/conf/default/send_redirects
cat /proc/sys/net/ipv4/conf/eth0/send_redirects
cat /proc/sys/net/ipv4/ip_forward
```

2.Rs 中配置脚本如下:

```
#!/bin/sh
#file:conf_rs.sh
#Written By XiaoKang.Leng
#Auto config the interface for the RS

PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH
ifconfig lo:0 172.26.46.119 netmask 255.255.255.255 broadcast
172.26.46.119 up
route add -host 172.26.46.119 dev lo:0

#All the config below is solution the ARP_Problem,
#In kernel 2.4 ,we use “hidden”
#In kernel 2.6 or later , use arp_ignore and arp_announce
#In our case ,FC3 RS is kernel 2.6

#Hidden lo:0 for the kernel 2.4 or later
#echo "1" > /proc/sys/net/ipv4/conf/all/hidden
#echo "1" > /proc/sys/net/ipv4/conf/lo/hidden
#Hidden lo:0 for the kernel 2.6 or later
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
```

3.一个标准化的 LVS 启动脚本

```
#!/bin/bash
#file:lvs.sh
#Written by XiaoKang.Leng
#Used to running the lvs

PATH=/bin:/usr/bin:/sbin:/usr/sbin
export PATH
IPVSADM=/sbin/ipvsadm

case "$1" in
```

```
start)
        if [ -x $IPVSADM ]
        then
        ifconfig eth0:0 172.26.46.119 netmask 255.255.255.255
broadcast 172.26.46.119 up
        $IPVSADM -C
        $IPVSADM -A -t 172.26.46.119:80 -s wlc
        $IPVSADM -a -t 172.26.46.119:80 -r 172.26.46.80 -w 1
        $IPVSADM -a -t 172.26.46.119:80 -r 172.26.46.81 -w 1


        fi
        ;;
stop)
if [ -x $IPVSADM ]
then
$IPVSADM -C
fi
;;
*)
echo "Usage:lvs{start|stop}"
exit1
esac
exit 0
```

说明，我们在测试过程中，主要是提供 httpd 服务。Ipvsadm 的编写方法，以及相关的参数，可以使用 ipvsadm –help 或者 man ipvsadm 查看更详细的介绍。

4.启动 lvs

可以直接采用上面的 lvs 启动脚本，启动完成后，输入 ipvsadm，可以看到以下信息:

```
IP Virtual Server version 1.0.9 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  ->RemoteAddress:Port           Forward Weight
  ActiveConn InActConn
  TCP 172.26.46.119:http wlc
  ->172.26.46.80:http Route 1    0    0
  ->172.26.46.81:http Route 1    0    0
```

5.lvs 测试

从客户端查看 http://172.26.46.119，如果 lvs 工作正常，则可以正常浏览到 172.26.46.80 和 172.26.46.81 的内容，为了确认 lvs 的正常运行，可以在 80 和 81 上设定不同的网页内容，也可以在 LD 上通过 ipvsadm 查看连接信息。

LVS 的安装工作就到此结束，两台 LD 上的 LVS 安装方式都一样，我们在实验中，Debian 和 RH9 都通过了测验。下面，开始安装 heartbeat:

heartbeat 说明:

heartbeat 是 Linux-HA 项目里面最核心的部分，主要是通过心跳检测，来达到一个高可用性的效果，我们在实验中，采用的 heartbeat 版本是 1.2.3，都采用的源代码安装方式，当然，你也可以使用 ultramonkey 分别给 RH 和 Debian 提供的安装包，效果应该相当。

第二部分　Heartbeat

Heartbeat 软件：
heartbeat 可以在 http://www.linux-ha.org 上面下载，在安装 heartbeat 前，请确认你的机器上有 libnet 库，没有，从源代码安装 heartbeat 如下：

```
./ConfigureMe bulid
make
make install
```

然后将以下几个文件拷贝到/etc/ha.d/目录：
ha.cf，haresources ，authkeys

heartbeat 的配置也不是想象中的那么困难，就上面那三个配置文件，在实验中，我们的配置信息如下：
ha.cf 文件：这是 heartbeat 通讯文件，指定了通讯方式以及时间，节点，等等，具体如下：

```
logfacility local0
keepalive 2
deadtime 10
warntime 10
initdead 20
udpport 694
auto_failback on
ucast eth1 192.168.0.1
node    rh1.cluster.net   #This must match the uname -n
node    debian.cluster.net  #This must match the uname -n
```

haresources：这是 heartbeat 的资源管理文件，所有资源必须在 resource.d 目录中包含，我们的 haresources 配置如下：

```
rh1.cluster.net  172.26.46.119/24/eth0
```

authkeys：设定两台 LD 之间的加密机制，有 sha1，crc 和 md5,我们在实验中用的是 sha1，配置如下：

```
auth 2
2 sha1 helloworld
```

关于 heartbeat 的配置，在配置文件里面都有非常详尽的说明，你也可以到 linux-ha.org 上查看相关的 heartbeat 配置文档。

启动 heartbeat
heartbeat 在 init.d 里面有标准的启动文件，可以随系统自动启动，当然，你也可以通过

```
/etc/init.d/heartbeat start
```

手动启动之。

Heartbeat 单元测试：

使用 tcpdump 命令查看两台 LD 之间是否有数据交流

```
tcpdump -n -i any port 694
```

可以通过显示的数据包来确定 heartbeat 之间的工作是否正常。

www.linuxidc.com

第三部分：LVS 与 Heartbeat 集成测试

在两台机器上启动 lvs 和 heartbeat。

1.LD 工作测试
从客户端浏览 http://172.26.46.119，如果正常浏览，则说明 LD1 主调度器正常，然后宕掉 LD1，在此从客户端浏览，如果在间隔时间内（10 秒）能正常浏览，则说明 LD 之间的工作正常。
2.LD+LVS 测试
将两台 RS 的首页设定成不同的页面，通过客户端浏览，会看到不同的页面，当然，由于缓冲的原因，你可能要多刷新几次。再做一次 LD 宕机测试，看 LD2 能否正确接管 LD1 的工作

第四部分：提高
由于时间等原因，实验就作到此了，以后的想法是，在现有的基础上，增加 mon 等资源管理程序，用来检测 RS 的宕机，让整个集群更趋于完美，让她更有效，更稳定，更安全的工作。

xk.leng@gmail.com
2005 年 9 月 12 日，初稿

Linux公社(LinuxIDC.com) 是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。