LVS 操作手册

BY 王平安_阿骨打 吴佳明_普空

1 集群模式-配置（单台）

a) **安装系统和工具**

```
# 下载 LVS 开源目录 kernel/和 tools/下的源码进行编译安装；
# 目录 kernel 为改进后的 LVS 内核源码；
# 目录 tools 下包括 ipvsadm/keepalived/quagga，ipvsadm/keepalived 用于管理配置 LVS，quagga 用于实现 LVS 集群；
```

b) **内核启动参数**

在 kernel 一行中，添加“**nohz=off** ”

注：如果不关闭 nohz，大压力下 CPU0 可能会消耗过高，压力不均匀；

c) **Sysctl 配置**

路径：/etc/sysctl.conf

```
# configure for lvs
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.core.netdev_max_backlog = 500000
```

d) **配置网卡参数**

路径：/etc/rc.local

关闭网卡 LRO 和 GRO

```
# ethtool -K eth0 gro off
# ethtool -K eth0 lro off
```

绑定网卡中断

```
# set_irq_affinity eth0 #脚本参见附录,该脚本是 ixgbe/igb driver 网卡
```

e) **关闭系统参数**

路径：/etc/rc.local

关闭 irqbalance

```
# service irqbalance stop
# chkconfig --level 2345 irqbalance off
```

f) **LocalAddress 配置**

路径：/etc/rc.local

Local address 绑定到内网(下联)网卡上

```
ip addr add 192.168.100.1/32 dev eth1
ip addr add 192.168.100.2/32 dev eth1
ip addr add 192.168.100.3/32 dev eth1
ip addr add 192.168.100.4/32 dev eth1
ip addr add 192.168.100.5/32 dev eth1
```

g) **Zebra.conf**

路径：/etc/quagga/zebra.conf

启动方式：/usr/sbin/zebra -d -f /etc/quagga/zebra.conf

```
hostname lvs-route-4
password 8 123456
enable password 8 123456
log file /var/log/zebra.log
service password-encryption
```

h) Ospf.conf

路径：/etc/quagga/ospf.conf

启动方式：/usr/sbin/ospfd -d -f /etc/quagga/ospf.conf

```
hostname lvs-4-ospfd
password 8 123456
enable password 8 123456
log file /var/log/ospf.log
log stdout
log syslog
service password-encryption

interface eth0 //上连网卡号
ip ospf message-digest-key 8 md5 123456
 ip ospf hello-interval 3
 ip ospf dead-interval 12

router ospf
ospf router-id 192.168.0.14   //route id 配置为 上连网卡接口 ip
log-adjacency-changes
auto-cost reference-bandwidth 1000
network 1.1.1.0/24 area 0.0.0.11  // VIP 网段
network 192.168.0.12/30 area 0.0.0.11  // 上连 IP 网段
area 0.0.0.11 authentication message-digest
area 0.0.0.11 stub no-summary
```

i) Keepalived.conf

启动：service keepalived start

更新：service keepalived reload

停止：service keepalived stop

Keepalived 的配置包含 2 个文件，以 taobao 业务为例：

说明：一个集群内的所有 LVS 配置文件基本相同，区别的地方见红色区域

i. 主配置文件 keepalived.conf

路径：/etc/keepalived/keepalived.conf

```
! Configuration File for keepalived
global_defs {
#   notification_email {
#     abc@taobao.com
```

```
#   }
#   notification_email_from abc@taobao.com
#   smtp_server 192.168.200.1
#   smtp_connect_timeout 40
}

local_address_group laddr_g1 {
      192.168.100.1
      192.168.100.2
      192.168.100.3
      192.168.100.4
      192.168.100.5
}

! include virtual server configure file
include taobao.conf
```

ii. **业务配置文件 “业务名.conf”**

路径：/etc/keepalived/taobao.conf

```
virtual_server_group taobao {
    1.1.1.1 80  //vip1
    1.1.1.2 80  //vip2
}

!for taobao.com
virtual_server group taobao {
    delay_loop 7
    lb_algo rr
    lb_kind FNAT
    protocol TCP
    syn_proxy
    laddr_group_name laddr_g1
     alpha        //启动 alpha 模式，以便自动绑定 vip
     omega       // 启动 omega 模式，以便自动解除 vip
     quorum 1
     hysteresis 0
     quorum_up " ip addr add 1.1.1.1/32 dev lo; ip addr add 1.1.1.2/32 dev lo;"
     quorum_down " ip addr del 1.1.1.1/32 dev lo; ip addr del 1.1.1.2/32 dev lo;"

    /* healthcheck for L4 */
    real_server 192.168.1.1 80 {
        weight 100
       inhibit_on_failure
        TCP_CHECK {
```

```
            connect_timeout 5
        }
    }

    /* healthcheck for L7 */
    real_server 192.168.1.2 80 {
        weight 100
        inhibit_on_failure
        HTTP_GET {
          url {
             path /index.html
             status_code 200
          }
        connect_timeout 3
        nb_get_retry 2
        delay_before_retry 5
       }
    }
}
```

j) 环境检查

i. 重要性高

在 LVS 刚部署完毕，或者运维操作完毕时，都必须检查以下配置；

- 命令 ip addr list，查看后端 VIP 是否绑定正确，查看 local address 是否绑定正确
- 命令 ipvsadm –ln，查看流量是否过来，各 RS 上流量是否均匀，流量大小是否符合预期；
- 命令 ps aux | grep keepalived，查看 keepalived 进程个数是否正确
- 命令 tcpdump –i any –nnn | grep OSPF， 查看 ospf 心跳是否正常
- 命令 route –n，查看 ospf 生成的路由是否正常
- 命令 tail –n 1000 /var/log/message，查看 keepalived 启动日志是否异常

ii. 重要性低

除了检查“3.1 重要性高”的点，还需要检查以下信息：

- 执行 cat /proc/interrupts | grep ethx，其中 ethx 为万兆网卡，查看网卡中断是否被正确地绑定在 N 个核上；
- 在 client 上 curl vip，在 lvs 上 curl rs_ip，查看能否 curl 通；

2 主备模式-配置（单台）

a) 安装系统和工具

```
# 下载 LVS 开源目录 kernel/和 tools/下的源码进行编译安装；
# 目录 kernel 为改进后的 LVS 内核源码；
# 目录 tools 下包括 ipvsadm/keepalived/quagga，ipvsadm/keepalived 用于管理配置 LVS，quagga 用于实现 LVS 集群；
```

b) 内核启动参数

在 kernel 一行中，添加“**nohz=off** ”

注：如果不关闭 nohz，大压力下 CPU0 可能会消耗过高，压力不均匀；

c) **Sysctl 配置**

路径：/etc/sysctl.conf

```
# configure for lvs
net.core.netdev_max_backlog = 500000
```

d) **配置网卡参数**

路径：/etc/rc.local

```
关闭网卡 LRO 和 GRO
# ethtool -K eth0 gro off
# ethtool -K eth0 lro off
绑定网卡中断
# set_irq_affinity eth0 #脚本参见附录,该脚本是 ixgbe/igb driver 网卡
```

e) **关闭系统参数**

路径：/etc/rc.local

```
关闭 irqbalance
# service irqbalance stop
# chkconfig --level 2345 irqbalance off
```

f) **LocalAddress 配置**

路径：/etc/rc.local

Local address 绑定到内网(下联)网卡上

```
ip addr add 192.168.100.1/32 dev eth1
ip addr add 192.168.100.2/32 dev eth1
ip addr add 192.168.100.3/32 dev eth1
ip addr add 192.168.100.4/32 dev eth1
ip addr add 192.168.100.5/32 dev eth1
```

g) **Keepalived.conf**

启动：service keepalived start

更新：service keepalived reload

停止：service keepalived stop

Keepalived 的配置包含 2 个文件，以 taobao 业务为例：

说明：一个集群内的所有 LVS 配置文件基本相同，区别的地方见红色区域

i. **主配置文件 keepalived.conf**

路径：/etc/keepalived/keepalived.conf

```
! Configuration File for keepalived
global_defs {
#   notification_email {
#     abc@taobao.com
#   }
#   notification_email_from abc@taobao.com
#   smtp_server 192.168.200.1
#   smtp_connect_timeout 40
}
```

```
local_address_group laddr_g1 {
      192.168.100.1
      192.168.100.2
      192.168.100.3
      192.168.100.4
      192.168.100.5
}

! include virtual server configure file
include taobao.conf
```

ii. **业务配置文件 “业务名.conf”**

路径：/etc/keepalived/taobao.conf

```
virtual_server_group taobao {
    1.1.1.1 80  //vip1
    1.1.1.2 80  //vip2
}

vrrp_instance VI_1 {
    state MASTER/BACKUP
    interface eth0
    virtual_router_id 200
    priority 150/90
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        1.1.1.1
        1.1.1.2
    }
}

!for taobao.com
virtual_server group taobao {
    delay_loop 7
    lb_algo rr
    lb_kind FNAT
    protocol TCP
    syn_proxy
    laddr_group_name laddr_g1

    /* healthcheck for L4 */
```

```
    real_server 192.168.1.1 80 {
        weight 100
       inhibit_on_failure
        TCP_CHECK {
            connect_timeout 5
        }
    }

    /* healthcheck for L7 */
    real_server 192.168.1.2 80 {
        weight 100
        inhibit_on_failure
        HTTP_GET {
          url {
              path /index.html
              status_code 200
          }
        connect_timeout 3
        nb_get_retry 2
        delay_before_retry 5
       }
    }
}
```

3 RS 配置

a) **安装系统**

```
# RealServer 请采用阿里内核：https://github.com/alibaba/ali_kernel
# 该内核包含了 toa 网络模块，用于 RS 上的应用程序获得真实的 Client IP，而不是 LVS 上的 Local Address；
# toa 实现了 client ip 对于 RS 的应用层透明，但对内核层是不透明的；
```

b) **加载 TOA 模块，命令：# modprobe toa**

```
# vim /etc/rc.local
添加 modproble toa
```

4 日常操作（以集群模式为例）

4.1 添加/删除 realserver

如果添加，请确保 realserver 的监听的 port 是打开的（可以 telnet 连接该端口）。

1) 第 1 步，配置 realserver，具体参见附录 5.1；

更新内核版本，加载相应的 TOA 模块：

```
# modprobe toa.ko
# vim /etc/rc.local
添加 modproble toa
```

2) 第 2 步，修改 keepalived 的配置，注意**所有 LVS** 上都得修改；

例如 realserver 的 IP 为 10.251.X.X，业务名 taobao

```
# vim /etc/keepalived/taobao.conf
virtual_server group taobao {
    ……
    real_server 10.251.X.X 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 4
        }
    }
}
```

3) 第 3 步，发送 HUP 信号给 keepalived，使配置修改生效；

```
# service keepalived reload
```

4) 第 4 步，检查 realserver 是否操作成功

在 LVS 上，分别运行 ipvsadm -ln 观察该 realserver 的健康检查是否成功，并在 LVS 查看 session 分配是否均匀。

4.2 添加/删除 vip

假设新添 vip 为 1.1.1.3，业务名称 taobao；

1) 修改 keepalived 配置文件，添加如下内容；

第一步，创建业务配置文件；

```
#vim /etc/keepalived/taobao.conf

virtual_server_group taobao {
    1.1.1.3 80  //vip1
}

virtual_server group taobao {
    delay_loop 6
    lb_algo rr
    lb_kind FNAT
```

```
    protocol TCP
    syn_proxy
    laddr_group_name laddr_g1
    alpha      //启动 alpha 模式，以便自动绑定 vip
    quorum 1
    hysteresis 0
    quorum_up " ip addr add 1.1.1.3/32 dev lo;"
    quorum_down " ip addr del 1.1.1.3/32 dev lo;"
    ……
}
```

第二步，修改 keepalived 配置文件；

```
#vim /etc/keepalived/keepalived.conf
……
! include virtual server configure file
include www.conf
include taobao.conf
```

2) 发送 HUP 信号给 keepalived，使配置修改生效；

```
# service keepalived reload
```

3) 检查 vip 配置是否生效；

```
# ipvsadm -ln //查看 vip 是否已经配置到 lvs 中
# ip addr list //查看 lo 上 vip 是否绑定成功
# 模拟用户访问 vip，结果是否正确
```

4.3 添加/删除 local address

注：local address 和内网接口 ip 绝对不能重合；

以添加/删除 192.168.100.4 为例，其内网网卡为 eth1，需要配置 2 个地方：

1. 修改/etc/rc.local

   ```
   添加 ip addr add 192.168.100.4/32 dev eth1
   ```

2. 修改/etc/keepalived/keepalived.conf

   ```
   #vim /etc/keepalived/keepalived.conf
   local_address_group laddr_g1 {
         ……
         192.168.100.4
   }
   ```

3. 发送 HUP 信号给 keepalived，使配置修改生效；

   ```
   # service keepalived reload
   ```

4. 检查 local address 配置是否生效；

   ```
   # ip addr list //查看网卡上是否已经绑定 ip
   # ipvsadm -G //查看 vip 上是否已经绑定 ip
   ```

5 set_irq_affinity 脚本(源自 intel 82599 driver)

```
# setting up irq affinity according to /proc/interrupts
# 2008-11-25 Robert Olsson
# 2009-02-19 updated by Jesse Brandeburg
#
# > Dave Miller:
# (To get consistent naming in /proc/interrups)
# I would suggest that people use something like:
#   char buf[IFNAMSIZ+6];
#
#   sprintf(buf, "%s-%s-%d",
#           netdev->name,
#       (RX_INTERRUPT ? "rx" : "tx"),
#       queue->index);
#
#  Assuming a device with two RX and TX queues.
#  This script will assign:
#
#   eth0-rx-0  CPU0
#   eth0-rx-1  CPU1
#   eth0-tx-0  CPU0
#   eth0-tx-1  CPU1
#


set_affinity()
{
    if [ $VEC -ge 32 ]
    then
        MASK_FILL=""
        MASK_ZERO="00000000"
        let "IDX = $VEC / 32"
        for ((i=1; i<=$IDX;i++))
        do
            MASK_FILL="${MASK_FILL},${MASK_ZERO}"
        done

        let "VEC -= 32 * $IDX"
        MASK_TMP=$((1<<$VEC))
        MASK=`printf "%X%s" $MASK_TMP $MASK_FILL`
    else
        MASK_TMP=$((1<<$VEC))
        MASK=`printf "%X" $MASK_TMP`
```

```
    fi

    printf "%s mask=%s for /proc/irq/%d/smp_affinity\n" $DEV $MASK $IRQ
    printf "%s" $MASK > /proc/irq/$IRQ/smp_affinity
}

if [ "$1" = "" ] ; then
    echo "Description:"
    echo "    This script attempts to bind each queue of a multi-queue NIC"
    echo "    to the same numbered core, ie tx0|rx0 --> cpu0, tx1|rx1 --> cpu1"
    echo "usage:"
    echo "    $0 eth0 [eth1 eth2 eth3]"
fi



# check for irqbalance running
IRQBALANCE_ON=`ps ax | grep -v grep | grep -q irqbalance; echo $?`
if [ "$IRQBALANCE_ON" == "0" ] ; then
    echo " WARNING: irqbalance is running and will"
    echo "          likely override this script's affinitization."
    echo "          Please stop the irqbalance service and/or execute"
    echo "          'killall irqbalance'"
fi

#
# Set up the desired devices.
#

for DEV in $*
do
  for DIR in rx tx TxRx
  do
     MAX=`grep $DEV-$DIR /proc/interrupts | wc -l`
     if [ "$MAX" == "0" ] ; then
       MAX=`egrep -i "$DEV:.*$DIR" /proc/interrupts | wc -l`
     fi
     if [ "$MAX" == "0" ] ; then
       echo no $DIR vectors found on $DEV
       continue
     fi
     for VEC in `seq 0 1 $MAX`
     do
        IRQ=`cat /proc/interrupts | grep -i $DEV-$DIR-$VEC"$"  | cut  -d:  -f1 |
sed "s/ //g"`
```

```
        if [ -n  "$IRQ" ]; then
          set_affinity
        else
           IRQ=`cat /proc/interrupts | egrep -i $DEV:v$VEC-$DIR"$"  | cut  -d:
-f1 | sed "s/ //g"`
           if [ -n  "$IRQ" ]; then
             set_affinity
           fi
        fi
     done
  done
done
```