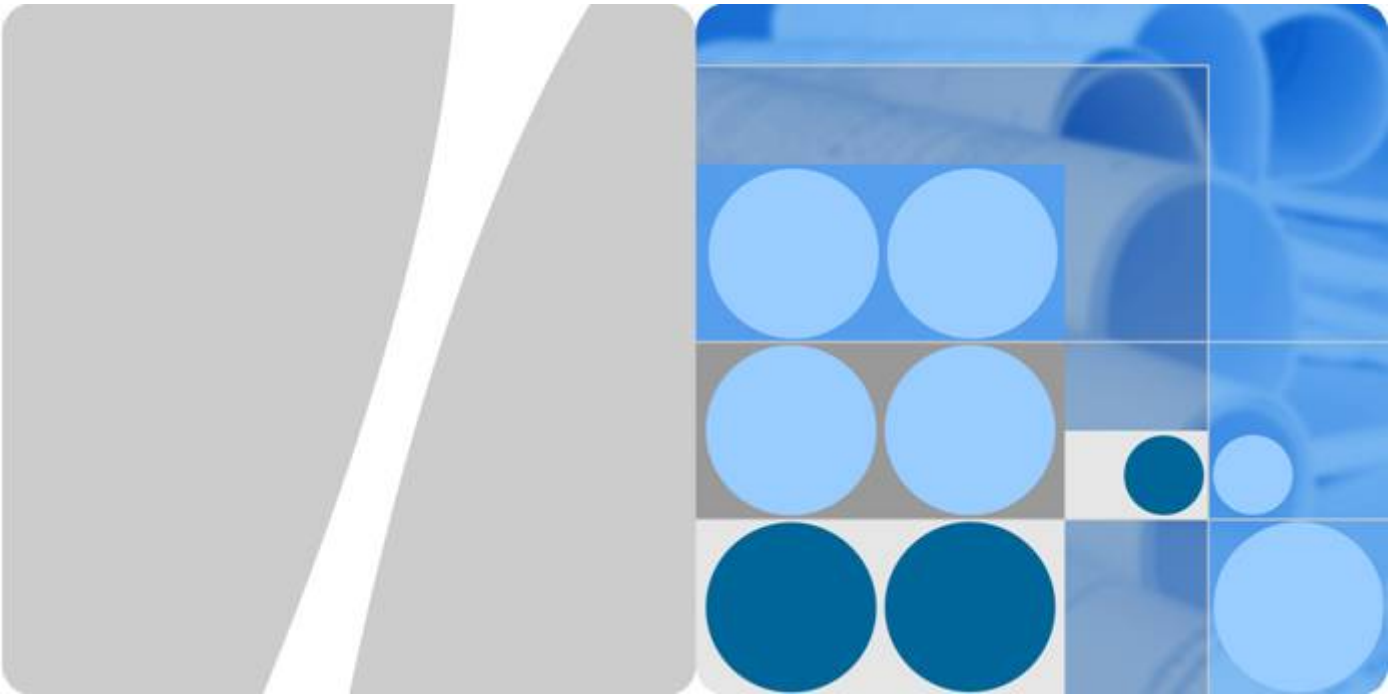


EulerOS V2.0SP2 IPVS 配置说明

文档版本　01

发布日期　2019-08-12

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： http://www.huawei.com

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

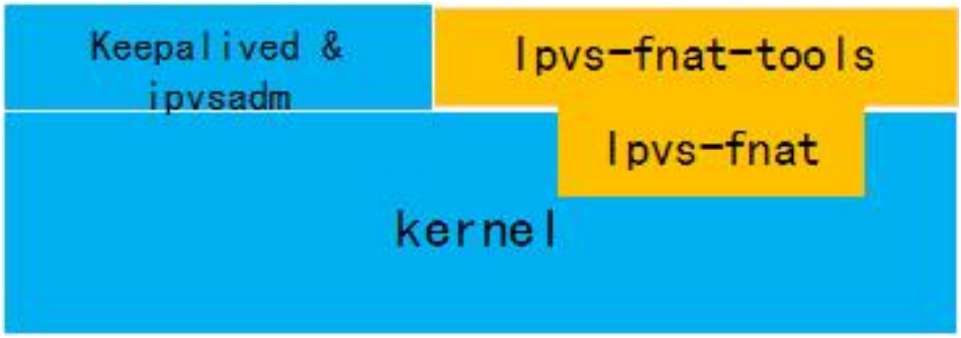
目 录

1 简介

什么是 LVS

LVS （Linux Virtual Server）是一种集群（Cluster）技术，采用IP负载均衡技术和基于内容请求分发技术。调度器具有很好的吞吐率，将请求均衡地转移到不同的服务器上执行，且调度器自动屏蔽掉服务器的故障，从而将一组服务器构成一个高性能的、高可用的虚拟服务器。整个服务器集群的结构对客户是透明的，而且无需修改客户端和服务器端的程序。

LVS 主要组成部分

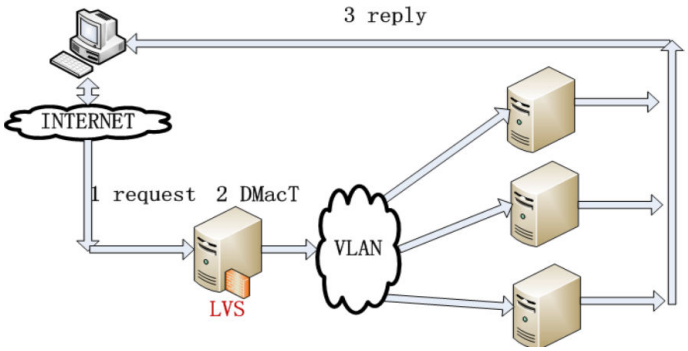


- keepalived&ipvsadm，用户态工具，用于健康检查、定义及管理lvs服务。
- ipvs-fnat-tools ， fullnat以及原生内核的ipvs模块切换工具。
- ipvs-fnat，fullnat模式实现模块。

LVS 负载均衡方式

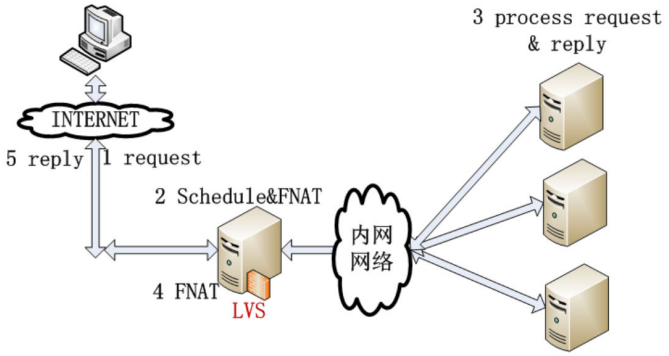
- VS/DR（Virtual Server via Direct Routing）

  VS/DR方式是通过改写请求报文中的MAC地址部分来实现的。Director和RealServer必需在物理上有一个网卡通过不间断的局域网相连。 RealServer上绑定的VIP配置在各自Non-ARP的网络设备上(如lo或tunl),Director的VIP地址对外可见，而RealServer的VIP对外是不可见的。RealServer的地址既可以是内部地址，也可以是真实地址。



- VS/FNAT（Virtual Server via Full NAT）

  VS/FNAT方式是即对报文做DNAT把VIP VPORT修改成RIP RPORT，又做SNAT把CIP CPORT修改成LIP LPORT。把报文转发给RealServer处理。从RealServer回来的报文根据LIP回到LVS，做相反的NAT。后端的网络设备和RealServer不需要做任何配置。



说明

- LVS特性只适用于Layer 4分发，不适用Layer 7分发。

2 配置准备

2.1 工具安装

1. 当前已经安装EulerOS。
2. 系统已经设置root密码，并且配置好了IP。
3. 对于EulerOS，可以采用iso方式安装或者配置repo源方式安装。

   说明

   当前支持在物理机和kvm虚拟机上部署。在虚拟机上部署时，当前仅支持virtio-net虚拟网卡，其他类型网卡的暂不支持。

iso 方式安装

步骤1 配置/etc/yum.repos.d/local.repo如下：

```
[iso]
baseurl=file:///iso
gpgcheck=0
enable=1
```

步骤2 挂载iso

```
mount -o loop -t iso9660 /home/elb/EulerOS-V2.0SP2-x86_64-dvd.iso /iso
```

步骤3 安装

```
yum insatll keepalived
yum install ipvsadm
yum install ipvs-fnat
yum install ipvs-fnat-tools
```

----结束

repo 方式安装

步骤1 配置/etc/yum.repos.d/local.repo如下：

```
[base]
name=EulerOS-2.0SP2 base
baseurl=http://developer.huawei.com/ict/site-euleros/euleros/repo/yum/2.2/os/x86_64/
```



```
enabled=1
gpgcheck=1
gpgkey=http://developer.huawei.com/ict/site-euleros/euleros/repo/yum/2.2/os/RPM-GPG-KEY-EulerOS
```

步骤2 安装

```
yum insatll keepalived
yum install ipvsadm
yum install ipvs-fnat
yum install ipvs-fnat-tools
```

说明

如果需要安装或者升级ipvs-fnat和ipvs-fnat-tools，请确保keepalived服务为关闭状态。

----结束

2.2 切换 ipvs 内核模块

概述

本节介绍如何切换ipvs的内核模块，以便使用fullnat模式。

注意事项

- 切换ipvs内核模块。
- 切换者必须有root权限。
- 切换前必须关闭keepalived，否则会提示keepalived还在运行。
- 切换时，会断流并删除所有内核中ipvs tables。
- 升级内核重启系统后，需要重新执行一次切换动作。

验证操作

以root权限，在shell命令行环境下操作。

使用ipvs-switch命令切换ipvs模式。

说明

ipvs-switch命令不支持并发。

- ipvs-switch orig：表示ipvs模式将切换成原生模式，此模式使用内核自带的ipvs模块，不支持fullnat模式。

```
[root@localhost keepalived]# ipvs-switch orig
Clear ipvs tables
rmmod ip_vs
doing depmod
modprobe ip_vs
switch to original mode successfully
```

- ipvs-swtich fnat：表示ipvs模式将切换成fullnat模式，此模式将卸载原生内核ipvs模块，modprobe支持fullnat的ipvs模块。

```
[root@localhost keepalived]# ipvs-switch fnat
Clear ip_vs tables
rmmod ip_vs modules
doing depmod
modprobe ip_vs modules
switch to fnat mode successfully
```


3 配置指导

3.1 配置网络

概述

本节介绍如何进行网络配置，实现网络互通。

包括DR模式，FNAT模式。

注意事项

1、确保客户端ping通该lvs服务器，lvs服务器能ping通后端。

2、需重启网络生效配置。

3、关闭防火墙。

配置步骤

以root权限，在shell命令行环境下操作。

步骤1 编辑配置文件。在任意路径下，使用如下命令：

```
vi  /etc/sysconfig/network-scripts/ifcfg-xx
```

网络参数需要用户根据实际业务情况进行相应的配置，此处简单举例如下：

```
TYPE=Ethernet
BOOTPROTO="static"
NAME=eth0
UUID=2845c1a2-4dcb-47c6-920e-72387845aaa3
DEVICE=eth0
#ONBOOT=no
```



```
IPADDR=192.168.31.247      //配置ip
NETMASK=255.255.0.0        //配置掩码
GATEWAY=192.168.0.1        //配置网关
STARTMODE="auto"
```

步骤2 重启网络服务，使配置生效，执行：

```
systemctl restart network
```

----结束

3.2 配置 keepalived

概述

本节介绍如何对keepalived进行配置。

注意事项

- 用户修改完配置文件，使用systemctl restart keepalived或者systemctl reload keepalived使配置生效。
- 配置文件必须为常规文件，如果具有可执行权限，会跳过不进行加载。
- 启动keepalived前需要关闭selinux。
- 当使用quorum_up配置vip时，如果配置的virtual server过多，后端上线过程中keepalived会出现卡住的现象。在后端为“cpu：CPU_E5-2650v3@2.30GHz*2；网卡：Intel_82599_10GE；内存：256G，后端监听服务为nginx”的情况下，最多能支持35000个左右的virtual server正常上线。若需配置更多的vip，建议使用“vip_bind_dev lo”配置，该方式与quorum_up的区别：
  a. 当未配置alpha或者配置了check_off时，virtual server下不配置real server，此时quorum_up会配置vip，而vip_bind_dev不会配置vip。
  b. 当停止keepalived服务时，vip_bind_dev会自动删除有real server在线的virtual server对应的vip，而仅配置quorum_up且未配置quorum_down时不会删除vip。
- vip_bind_dev、quorum_up、vrrp都可以配置vip，为了避免冲突，建议只使用其中一种方式配置vip。

说明

keepalived服务启动后，若是向keepalived子进程发送9号信号（SIGKILL），由于keepalived主进程无法响应信号，keepalived主进程会再次拉起keepalived子进程。若此时修改keepalived配置信息且配置中删除了一些后端，或者有的后端此时健康检查还没有上线，可能会导致内核与用户态的后端信息不一致，使用ipvsadm -ln查看时可能会残存一些下线的后端信息。

配置步骤

说明

1. keepalived对配置文件的参数不做语法拼写的异常检查，配置参数语法或者拼写由用户保证正确性。
2. keepalived配置文件的修改与reload不能并发执行，否则会出现不可预知的配置残留问题。

以root权限，在shell命令行环境下操作。

步骤1 编辑keepalived配置文件：

```
vi /etc/keepalived/keepalived.conf
```

FNAT的配置示例如下：

```
! Configuration File for keepalived

#全局定义模块
global_defs {
}

local_address_group laddr_g1 { #laddr组，fullnat模式专用
     192.168.31.247    #LOCAL ip 用于和后端设备通信
}

virtual_server 192.168.31.200 80 {   #vip地址和端口
    delay_loop 6    #健康检查时间间隔
    lb_algo rr      #LVS调度算法，支持rr|wrr|lc|wlc|lblc|sh|dh|consh
    persistence_timeout 60 #LVS持续会话超时时间，单位秒
    lb_kind FNAT    #均衡转发模式，FNAT，DR
    protocol TCP    #支持的协议模式是TCP还是UDP
    laddr_group_name laddr_g1   #指定local ip对应的group名称
    alpha           #开启alpha模式，在keepalived启动时，假设所有的RS都是down，等健康检查通过后rs才
上线。
    omega           #开启omega模式，在keepalived终止时，会执行quorum_down指令所定义的脚本。
    quorum 1        #设置服务是否有效的阀值
    hysteresis 0    #延迟系数（跟quorum 配合使用，保证小于quorum）
    #高于或低于阀值时会执行以下脚本。
    quorum_up "/usr/sbin/ip addr add 192.168.31.200/32 dev lo;"
    real_server 192.168.31.248 80 {     #真实服务ip和端口
        weight 6                    #该实节点权重
        TCP_CHECK {                 #健康检查方式
            connect_port 80         #检查的端口
            connect_timeout 15      #连接超时时间
            nb_get_retry 5          #重连次数
            delay_before_retry 3    #重连间隔
        }
    }
}
```

说明

- 对于quorum_up参数，如果没有开启alpha模式，每次启动keepalived服务或者加载keepalived新配置项时，都会触发执行quorum_up里的命令。
- 由于quorum_up和quorum_down可能会出现并发的场景，导致命令执行结果不符合预期，所以不建议配置quorum_down。
- keepalived的一个virtual_server下，配置real_server时不允许出现两个ip和port均相同的情况，否则会导致无法预知的后果。

DR的配置示例如下（部分参数说明可参考FNAT的配置示例）：

```
! Configuration File for keepalived
global_defs {
    router_id LVS_DEVEL_168 #节点标识，唯一即可
}

vrrp_instance LVS_DR_HW {#主备模式下需配置该结构
    state MASTER      #节点的初始状态，但启动后还是通过竞选由优先级来确定
    interface eth1    #虚拟IP绑定的网卡
    virtual_router_id 200 #设置VRID，相同的VRID为一个组，决定多播的MAC地址
    priority 168 #设置本节点的优先级，启动后优先级高的为master
    advert_int 1 #检查间隔，默认为1秒
    virtual_ipaddress { #VIP，它随着state的变化而增加/删除，当state为master的时候就在该节点添加
        192.168.100.2 #当为backup时删除。正常情况下由优先级来决定的，此处可以设置VIP
    }
}
virtual_server 192.168.100.2 5002{
    delay_loop 6          #服务轮询的时间间隔
    lb_algo sh            #LVS调度算法，支持rr|wrr|lc|wlc|lblc|sh|dh
    lb_kind DR            #均衡转发模式，支持FNAT，DR
    protocol TCP          #支持的协议模式是TCP还是UDP
    real_server 192.168.100.10 5002 {
       weight 1
       TCP_CHECK {
            connect_port 5002
```

```
            connect_timeout 15
            nb_get_retry 5
            delay_before_retry 3
        }
    }
    real_server 192.168.100.11 5002 {
        weight 2
        TCP_CHECK {
            connect_port 5002
            connect_timeout 15
            nb_get_retry 5
            delay_before_retry 3
        }
    }
}
```

步骤2 启动keepalived，使用命令：

```
systemctl start keepalived
```

步骤3 用ipvsadm -Ln 查看配置有没有下发成功：

```
ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Established(Sec.) Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConnTCP
     192.168.31.200:80    rr                           persistent 60
  -> 192.168.31.248:80            FullNat 6      0          0
```

如果不成功，进行故障定位参见**3.5 故障定位**。

步骤4 设置开机启动：

```
systemctl enable keepalived.service
```

----结束

3.3 设置/etc/sysctl.conf

概述

本节介绍如何设置/etc/sysctl.conf。

注意事项

IPVS DR和IPVS FNAT需要做的配置有差异。

配置步骤

以root权限，在shell命令行环境下操作。

- 在RealSever上配置（DR方式需要做此配置）：

步骤1 配置vip到lo口：

```
ip addr add <vip>/32 dev lo
```

步骤2 修改内核参数：

```
vi /etc/sysctl.conf
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.all.rp_filter = 0
net.ipv4.ip_forward = 1
```


步骤3 使参数生效：

```
sysctl -p
```

----结束

- 在LVS上需要把ip_forward 打开。 DR和FNAT方式都需要做此配置。

步骤1 修改内核参数：

```
vi /etc/sysctl.conf
net.ipv4.ip_forward = 1
```

步骤2 使参数生效：

```
sysctl -p
```

----结束

3.4 验证配置成功

概述

本节介绍如何验证配置是否成功。

注意事项

如果配置不成功，请参见下章故障定位。

验证操作

以root权限，在shell命令行环境下操作。

步骤1 在Real Server上安装httpd 或者nginx。

说明

安装httpd 或者nginx不在此说明，请用户查阅相关文档。

步骤2 在LVS 启动 keepalived：

```
systemctl start keepalived
```

步骤3 在LVS上，用ipvsadm查询下发表项是否生效：

```
ipvsadm -Ln
Prot LocalAddress:Port Scheduler Established(Sec.) Flags
-> RemoteAddress:Port          Forward Weight ActiveConn InActConnTCP
   192.168.31.200:80    rr                            persistent 60
-> 192.168.31.248:80           FullNat 6      0          0
```

表项有效的标志是有vip表项->RealServer

步骤4 在客户端用浏览器或者curl访问vip的地址：

```
curl 192.168.31.200:80
```

客户端如果是PC机，可以用IE或者Chrome，地址栏输入 http://192.168.31.200:80进行验证。

----结束

3.5 故障定位

概述

本节介绍常见IPVS故障。

注意事项

无。

常见故障

1. 使用ping命令检查 lvs物理口地址，如果无法ping通，请检查网络配置。
2. 使用ping命令检查VIP，如果无法ping 通，检查vip 是否以VIP/32的方式配置在LVS网卡上。
3. 从用户能ping通VIP，但是客户端浏览器无法访问IPVS配置的业务端口，检查该报文是否在LVS上或者RealServer上 被iptables拦截，可以用iptables -F清除所有规则。
4. 使用ipvsadm -Ln看ipvs表项有没有正确下发。
5. 安装tcpdump软件包，在LVS上抓ipvs的报文，比如tcpdump | grep <cip>。查看报文是否被捕捉到，并转发给RealServer。
6. 在RealServer上抓报文，查看报文中是否包含来自LVS的报文，比如 tcpdump | grep <cip> 或tcpdump | grep <lip>。
7. 用ipvsadm -Ln 查询ipvs 的流是否建立：
   a. 正常情况， 表项中有->指向Real Server的端口。
   ```
   Prot LocalAddress:Port Scheduler Established(Sec.) Flags
   -> RemoteAddress:Port           Forward Weight ActiveConn InActConnTCP
     192.168.31.200:80    rr                          persistent 60
   -> 192.168.31.248:80            FullNat 6      0          0
   ```
   b. 不正常情况， 表项中没有->指向Real Server的端口，可能的故障是到Real Server 网络不通，或者RealServer 相关服务或者端口没有开启。
   ```
   Prot LocalAddress:Port Scheduler Established(Sec.) Flags
   -> RemoteAddress:Port           Forward Weight ActiveConn InActConnTCP
     192.168.31.200:80    rr                          persistent 60
   ```
8. 在DR模式，如果RealServer能正常收发IPVS报文，但是业务还是不通，检查**3.3 设置/etc/sysctl.conf**的RealServer系统参数是否配置生效：
   a. cat /proc/sys/net/ipv4/conf/all/arp_ignore 应该等于1
   b. cat /proc/sys/net/ipv4/conf/all/arp_announce 应该等于2
   c. cat /proc/sys/net/ipv4/conf/all/rp_filter 应该等于0
   d. cat /proc/sys/net/ipv4/ip_forward 应该等于1
9. 在DR模式下，需要RealServer ping通客户端确保RealServer 的报文能直接发给客户端。
10. 在NAT模式，确保RealServer回来的报文通过配置默认网关或者策略路由的方式回流到LVS，使报文能在LVS上做SNAT。
11. 在配置vip realserver时，如果一个VIP的端口有多个real server。当其中一个real server连接状态不正常时，会触发keepalive执行quorum动作，如果quorum动作是删除VIP，那会导致该VIP整体不可用。

12. 不能频繁reload keepalived服务：reload间隔时间建议大于配置中（nb_get_retry +1） * （connect_timeout + delay_before_retry）的最大值再加上读取配置的时间，否则会出现部分后端不通时无法下线的现象。nb_get_retry 、connect_timeout、delay_before_retry参数详见**3.2 配置keepalived**中的示例代码。
13. 如果不进行内核升级，直接将2.2.6之前的版本的ipvs-toa rpm包升级到EulerOS_2.2.6及以后版本的ipvs-toa rpm包，会存在无法通过modprobe加载toa模块的问题，需要通过insmod /lib/modules/EulerOS/toa/toa.ko方式加载。
14. 当业务大压力场景下，不要进行多进程频繁查询接口cat /proc/net/ip_vs，否则有概率性造成系统Oops复位。建议单进程查询cat /proc/net/ip_vs并查询间隔不小于10s。
15. 配置了checker_merge的conf文件中，alpha的设置必须要保持一致，否则会出现部分vip对应后端上下线失败的现象。
16. keepalived配置文件的修改与reload不能并发执行，否则会出现不可预知的配置残留问题。
17. keepalived升级场景下，kill keepalived进程期间，不允许修改（含增删）配置文件，否则会出现配置残留；待keepalived服务重新拉起后，才能进行文件修改操作。

    a. kill -9 杀死keepalived进程，增加1个配置文件，start keepalived服务成功且新增配置生效；

    b. kill -9 杀死keepalived进程，删除1个配置文件，start keepalived服务成功且删除的svc残留，stop keealived服务后，ipvsadm-ln仍可查询到配置文件对应的svc。
18. 使用ipvsadm -ln与cat /proc/net/ip_vs查询命令不能与添加、删除virtual_server的操作并发，否则可能出现查询结果重复或丢失。

4 升级指导

4.1 ipvs-fnat 升级指导

1. 停止keepalived服务：
```
systemctl stop keepalived
```
2. 确认keepalived服务停止，停止执行ipvsadm命令以及调用该命令的脚本，停止引用ip_vs内核ko模块，在确认ip_vs的内核模块不会再被引用后，升级ipvs-fnat的rpm包：
```
rpm -Uvh ipvs-fnat-1.0.1-85.x86_64.rpm --force
```
3. 确保rpm包升级完毕后，启动keepalived服务：
```
systemctl start keepalived
```

说明

ipvs-fnat模块在升级阶段会断流，请注意使用场景，如不允许断流请先将流量切换到其他LVS节点再进行升级。

4.2 keepalived 升级指导

1. 移除并备份keepalived配置文件：
```
mkdir /home/user/keepalived-backup
mv /etc/keepalived/* /home/user/keepalived-backup/
```
2. 强制关闭keepalived checker子进程，停止keepalived服务，然后升级keepalived的rpm包：
```
kill -9 `cat /var/run/checkers.pid`
systemctl stop keepalived
rpm -Uvh --force keepalived-1.2.13-22.h49.x86_64.rpm --nodeps
```
3. 恢复keepalived配置文件，启动keepalived服务：
```
mv /home/user/keepalived-backup/* /etc/keepalived/
systemctl start keepalived
```

说明

使用kill -9 checker子进程之前，要移除/etc/keepalived/文件夹下的配置文件，否则关闭keepalived服务后，内核的配置会被清除，导致断流。

5 自研新特性

5.1 不断流特性（draining）

概述

为了实现后端服务器的平滑升级，需要在lvs中实现不断流特性即在后端服务器从lvs服务中移除后，在指定时间内，只要后端服务器的网络正常，已经建立连接的数据依旧能够送到后端服务器，等到当前的连接全部结束后即可升级后端服务器。

注意事项

从带有此特性的版本开始，keepalived的部分常用特性参数配置修改，无需restart keeplived，只需要reload keepalived即可。

配置步骤

以root权限，在shell命令行环境下操作。

步骤1 编辑配置文件。在任意路径下，使用如下命令：

```
vi  /etc/keepalived/keepalived.conf
```

步骤2 参数名称：tcp_draining_timeout

tcp_draining_timeout 表示关闭通道的超时时间，单位为秒。 0表示永不超时。

说明

tcp_draining_timeout 取值范围0-3600。

----结束

参数使用情况举例：

情况一：表示使能tcp_draining，并且相关流都自然老化，没有时间限制。

```
virtual_server 192.168.31.240 80{
    delay_loop 6
    lb_algo rr
    persistence_timeout 60
    lb_kind FNAT
    protocol TCP
    laddr_group_name laddr_g1
    alpha
    omega
    quorum 1
    hysteresis 0
    quorum_up "/usr/sbin/ip addr add 192.168.31.240/32 dev lo;"
    tcp_draining_timeout 0
    real_server 192.168.31.246 80{
        weight 1
        TCP_CHECK {
            connect_port 80
            connect_timeout 15
            nb_get_retry 5
            delay_before_retry 3
        }
    }
}
```

说明

tcp_draining_timeout后无参数请参考情况一。

情况二：使能tcp_draining，并且相关流在5秒后，强制老化。

```
virtual_server 192.168.31.240 80{
    delay_loop 6
    lb_algo rr
    persistence_timeout 60
    lb_kind FNAT
    protocol TCP
    laddr_group_name laddr_g1
    alpha
    omega
    quorum 1
    hysteresis 0
    quorum_up "/usr/sbin/ip addr add 192.168.31.240/32 dev lo;"
    tcp_draining_timeout 5
    real_server 192.168.31.246 80{
        weight 1
        TCP_CHECK {
            connect_port 80
            connect_timeout 15
            nb_get_retry 5
            delay_before_retry 3
        }
    }
}
```


5.2 一致性 hash 算法

概述

lvs自带的load balance算法中的源地址hash算法在任意一台后端服务器down时，会导致所有后端服务器的连接几乎全部异常，同时流量的均衡性做的不够好。现网集群模式下部署需依赖前端（DNS或者ECMP等价路由）的sh算法才能使用。因此，需要一种新的sh算法保证当后端服务器异常时的影响降到最低，同时还能保证同一客户端访问集群中任意一台lvs服务器的时候，确保分发到同一后端服务器，这种新的sh算法叫做一致性hash算法。

注意事项

EulerOS2.2版本keepalived配置中新增consh配置项表示一致性hash算法。安装方法请参考**2.2 切换ipvs内核模块**。

配置步骤

以root权限，在shell命令行环境下操作。

步骤1 编辑配置文件。在任意路径下，使用如下命令：

```
vi  /etc/keepalived/keepalived.conf
```

步骤2 参数使用情况举例：

```
virtual_server 192.168.31.240 80{
    delay_loop 6
    lb_algo consh
    persistence_timeout 60
    lb_kind FNAT
    protocol TCP
    laddr_group_name laddr_g1
    alpha
    omega
    quorum 1
    hysteresis 0
    quorum_up "/usr/sbin/ip addr add 192.168.31.240/32 dev lo;"
    tcp_draining_timeout 0
    real_server 192.168.31.246 80{
        weight 1
        TCP_CHECK {
            connect_port 80
            connect_timeout 15
            nb_get_retry 5
            delay_before_retry 3
        }
    }
}
```

----结束

5.3 lvs 流量控制

概述

当前elb组网环境需要一种支持租户cps、并发数控制的功能，当租户的连接大于申请连接时，对租户的访问连接进行限制，才能有效的对租户进行隔离。当某个客户受到攻击时可以通过限制并发连接和新建连接，第一时间降低影响，提供安全防护。

注意事项

- 当前支持的粒度为vip+端口。
- 只在fullnat模式下支持该特性。

配置步骤

以root权限，在shell命令行环境下操作。

步骤1 编辑配置文件。在任意路径下，使用如下命令：

```
vi /etc/keepalived/keepalived.conf
```

步骤2 参数设置。

- 新建连接控制
  - flow_control_cps，cps过载控制。0：关闭；＞0：开启。
  - flow_control_interval，流控时间。0：关闭；＞0：开启。只有在此参数开启的情况下，flow_control_cps参数才有效。
- 并发连接控制
  - flow_control_pc，并发连接数限制。0：关闭；＞0 : 开启。

说明

1. 以上参数必须设置一个数字，不能为空。
2. 对于UDP连接，只有在连接超时之后才认为连接断开，超时时间可以使用 ipvsadm --set命令设置。
3. 新建连接控制、并发连接控制偶尔会有一定误差，导致实际允许通过的连接数略大于所设置的阈值，但误差范围不超过cpu核数。

参数使用情况举例：

```
virtual_server 192.168.31.240 80{
    delay_loop 6
    lb_algo rr
    persistence_timeout 60
    lb_kind FNAT
    protocol TCP
    laddr_group_name laddr_g1
    alpha
    omega
    quorum 1
    hysteresis 0
    quorum_up "/usr/sbin/ip addr add 192.168.31.240/32 dev lo;"
    tcp_draining_timeout 60
    flow_control_cps 5
    flow_control_interval 5
    flow_control_pc 1000000
    real_server 192.168.31.246 80{
        weight 1
```

```
        TCP_CHECK {
            connect_port 80
            connect_timeout 15
            nb_get_retry 5
            delay_before_retry 3
        }
    }
}
```

上述流控参数配置说明：

- cps：5秒内最大支持5*5条连接，超过该连接数不能建链，下个5秒周期恢复。
- pc : 同一时刻最多允许存在1000000条连接，当连接数超过该数值时不能新建连接。

----结束

5.4 客户端黑白名单

概述

黑白名单可以实现允许或拒绝特定的客户端访问该lvs服务器。黑白名单以virtual server为粒度单位进行配置。如果只配置了黑名单，那么被列入黑名单中的客户端不能访问该virtual server，其他客户端可以正常访问；如果只配置了白名单，那么只有被列入白名单中的客户端才能访问该virtual server，其他客户端都不允许访问。

注意事项

- 只在fullnat模式下支持该特性。
- 黑白名单以一个virtual server为粒度，各个virtual server之间的名单是互相独立的。
- 如果同一个virtual server中同时配置了黑白名单，此时只有白名单生效，黑名单是无效的。
- 如下命令仅用于调测，大规格（超过1k）添加黑白名单后，不允许执行该命令，否则可能有rlock风险。

  ```
  cat /proc/net/ip_vs_*_client
  ```

配置步骤

以root权限，在shell命令行环境下操作。

步骤1 执行echo 1 > /proc/sys/net/ipv4/vs/client_list_switch使能该功能。1表示使能功能，0表示关闭功能，默认为关闭。

步骤2 编辑配置文件。在任意路径下，使用如下命令：

```
vi  /etc/keepalived/keepalived.conf
```

步骤3 白名单的关键字为client_ipaddr_allow，黑名单的关键字为client_ipaddr_exclude，配置方法如下：

```
关键字 {
     192.168.11.22
     192.168.0.1/24
     192.168.201.18/8
 }
```

参数使用情况举例：

```
virtual_server 192.168.31.240 80 {
    delay_loop 6
```



```
    lb_algo rr
    persistence_timeout 60
    lb_kind FNAT
    protocol TCP
    laddr_group_name laddr_g1
    alpha
    omega
    quorum 1
    hysteresis 0
    client_ipaddr_allow {
        192.168.11.22
        192.168.0.1/24
        193.168.201.18/8
    }
    real_server 192.168.31.246 80 {
        weight 1
        TCP_CHECK {
            connect_port 80
            connect_timeout 15
            nb_get_retry 5
            delay_before_retry 3
        }
    }
}
```

上述名单配置说明：

- 如果要配置黑名单，修改关键字client_ipaddr_allow为client_ipaddr_exclude即可。
- 名单中的IP有两种写法：
  - 192.168.11.22不带掩码长度的绝对IP；
  - 192.168. 0.1/24带有掩码长度的IP，这里的24就是掩码长度。IP和掩码长度之间用"/"隔开。192.168. 0.1/24表示192. 168. 0.0-192. 168. 0.255之间的IP。

    说明

    - 掩码长度的有效范围为1-32。
    - IP地址和掩码长度之间不能有空格。
    - 名单中的每行必须要有IP地址，且是合法的IP地址。
    - 当前不支持IPv6地址。
    - 每个virtual server中的黑名单或者白名单中，最多配置1000项。
    - 关键字和后面的左括号{之间需要有空格隔开，IP项从左括号开始到最近的右括号结束。

----结束

5.5 健康检查

5.5.1 支持健康检查关闭

概述

当前健康检查采用rst报文作为结束报文，这种非友好结束链接的方式可能导致客户后端低性能设备产生大量日志挂死，同时给过滤报文造成困难。

关闭健康检查特性可以降低keepalived的cpu使用率，提升keepalived能容纳的listener和vip数量。

注意事项

- 只针对Layer 4客户场景使用。
- 只在FULLNAT模式下使用。

配置接口

使用checker_off作为开关，以listener为单位进行配置。

以root权限，在shell命令行环境下操作。

步骤1 编辑配置文件。在任意路径下，使用如下命令：

```
vi  /etc/keepalived/keepalived.conf
```

步骤2 参数使用情况举例：

参数名称：checker_off

```
virtual_server 192.168.31.200 80 {   #vip地址和端口
    delay_loop 6    #健康检查时间间隔
    lb_algo rr      #LVS调度算法，支持rr|wrr|lc|wlc|lblc|sh|dh|consh
    persistence_timeout 60 #LVS持续会话超时时间，单位秒
    lb_kind FNAT    #均衡转发模式，FNAT，DR
    protocol TCP    #支持的协议模式是TCP还是UDP
    laddr_group_name laddr_g1   #指定local ip对应的group名称
    alpha           #开启alpha模式，在keepalived启动时，假设所有的RS都是down，等健康检查通过后rs才
上线。
    omega           #开启omega模式，在keepalived终止时，会执行quorum_down指令所定义的脚本。
    checker_off     #关闭健康检查
    quorum 1        #设置服务是否有效的阀值
    hysteresis 0    #延迟系数（跟quorum 配合使用，保证小于quorum）

    #高于或低于阀值时会执行以下脚本。
    quorum_up "/usr/sbin/ip addr add 192.168.31.200/32 dev lo;"
    real_server 192.168.31.248 80 {     #真实服务ip和端口
        weight 6                    #该实节点权重
        TCP_CHECK {                 #健康检查方式
            connect_port 80         #检查的端口
            connect_timeout 15      #连接超时时间
            nb_get_retry 5          #重连次数
            delay_before_retry 3    #重连间隔
        }
    }
}
```

说明

checker_off与alpha，或者checker_merge同时打开的时候，只有checker_off生效。

----结束

5.5.2 支持健康检查合并

概述

对多个VS中ip重复的后端的健康检查进行合并，所有配置了该项的VS中相同ip的后端只会做一次健康检查。存在ip重复项的后端个数最多不能超过1024个。（系统中arp表默认为1024，当存在ip重复项的后端个数太多时，arp表无法刷新，会引起网络异常，可以通过修改arp表项阈值解决）。

注意事项

- 只支持TCP_CHECK健康检查方式。
- 只在FULLNAT模式下使用。

配置步骤

说明

所有配置了checker_merge的conf文件中，alpha的设置必须要保持一致，否则会出现部分vip对应后端上下线失败的现象。

使用checker_merge作为开关，以listener为单位进行配置。

以root权限，在shell命令行环境下操作。

步骤1 编辑配置文件。在任意路径下，使用如下命令：

```
vi  /etc/keepalived/keepalived.conf
```

步骤2 参数使用情况举例：

参数名称：checker_merge

```
virtual_server 192.168.31.200 80 {   #vip地址和端口
    delay_loop 6    #健康检查时间间隔
    lb_algo rr      #LVS调度算法，支持rr|wrr|lc|wlc|lblc|sh|dh|consh
    persistence_timeout 60 #LVS持续会话超时时间，单位秒
    lb_kind FNAT    #均衡转发模式，FNAT，DR
    protocol TCP    #支持的协议模式是TCP还是UDP
    laddr_group_name laddr_g1   #指定local ip对应的group名称
    alpha           #开启alpha模式，在keepalived启动时，假设所有的RS都是down，等健康检查通过后rs才上线。
    omega           #开启omega模式，在keepalived终止时，会执行quorum_down指令所定义的脚本。
    checker_merge   #后端健康检查合并
    quorum 1        #设置服务是否有效的阀值
    hysteresis 0    #延迟系数（跟quorum 配合使用，保证小于quorum）

    #高于或低于阀值时会执行以下脚本。
    quorum_up "/usr/sbin/ip addr add 192.168.31.200/32 dev lo;"
    real_server 192.168.31.248 80 {     #真实服务ip和端口
        weight 6                    #该实节点权重
        TCP_CHECK {                 #健康检查方式
            connect_port 80         #检查的端口
            connect_timeout 15      #连接超时时间
            nb_get_retry 5          #重连次数
            delay_before_retry 3    #重连间隔
        }
    }
}
```

说明

checker_merge与checker_off同时打开的时候，checker_merge不生效。

----结束

5.5.3 支持 UDP 健康检查

概述

原有keepalived的UDP健康检查方式是通过调用python脚本处理，python脚本创建socket后会持续等待返回结果，当有大量UDP健康检查事件后会导致系统的cpu升高。基于这个问题，EulerOS新增加一种UDP的健康检查方式，用以解决上述脚本调用导致的系统性能不足的问题。

注意事项

1. 当前系统中有icmp报文限速，会导致unreachable报文回复较慢，从而出现后端不能及时下线的情况，此时需要关闭icmp报文限速。
   - EulerOS需要在rs上配置内核参数，在/etc/sysctl.conf中新增net.ipv4.icmp_ratelimit=0。
   - Windows需要自行配置解除icmp报文速率限制。
2. 支持的最大后端数与本地可用的udp端口数相同。
3. MISC_CHECK健康检查性能差，会极大消耗keepalived检查子进程的CPU，且MISC_CHECK可靠性差，所以强烈建议使用UDP_CHECK进行健康检查。

配置步骤

以root权限，在shell命令行环境下操作。

步骤1 编辑配置文件。在任意路径下，使用如下命令：

```
vi  /etc/keepalived/keepalived.conf
```

参数使用情况举例：

```
virtual_server 192.168.31.200 53 {   #vip地址和端口
    delay_loop 6    #健康检查时间间隔
    lb_algo rr      #LVS调度算法，支持rr|wrr|lc|wlc|lblc|sh|dh|consh
    persistence_timeout 60 #LVS持续会话超时时间，单位秒
    lb_kind FNAT    #均衡转发模式，FNAT，DR
    protocol UDP    #支持的协议模式是TCP还是UDP
    laddr_group_name laddr_g1   #指定local ip对应的group名称
    alpha           #开启alpha模式，在keepalived启动时，假设所有的RS都是down，等健康检查通过后rs才上线。
    omega           #开启omega模式，在keepalived终止时，会执行quorum_down指令所定义的脚本。
    quorum 1        #设置服务是否有效的阀值
    hysteresis 0    #延迟系数（跟quorum 配合使用，保证小于quorum）

    #高于或低于阀值时会执行以下脚本。
    quorum_up "/usr/sbin/ip addr add 192.168.31.200/32 dev lo;"
    real_server 192.168.31.248 53 {     #真实服务ip和端口
        weight 6                    #该实节点权重
        UDP_CHECK {                 #健康检查方式
            connect_port 53         #检查的端口
            connect_timeout 15      #连接超时时间
            nb_get_retry 5          #重连次数
            delay_before_retry 3    #重连间隔
            ping_check_off          #关闭ping检查，不配置该项默认开启ping检查
        }
    }
}
```

说明

关闭ping检查会影响“real_server”某些场景下的正常下线。

步骤2 开启ping检查时，需要修改内核参数，新增“**net.ipv4.ping_group_range=0 0**”：

```
vi /etc/sysctl.conf
net.ipv4.ping_group_range=0 0
```

步骤3 使参数生效：

```
sysctl -p
```

----结束

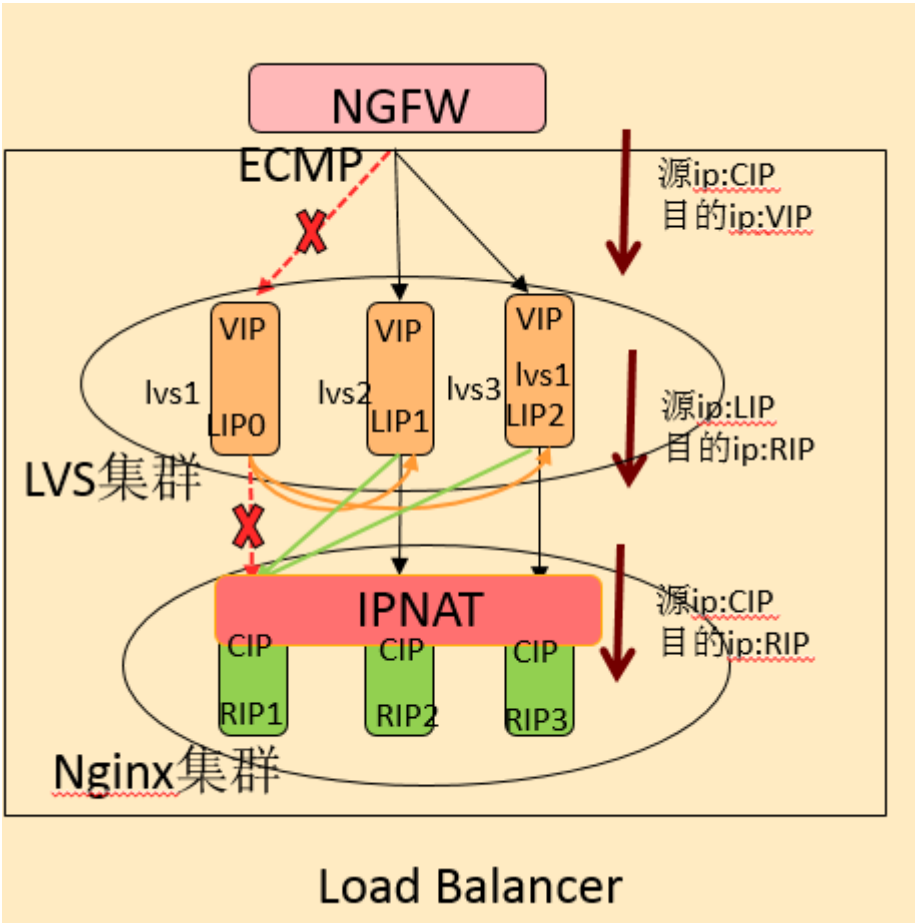
5.6 流表同步

概述

LVS作为公有云ELB最前端的组件，其冗余能力、稳定性、平滑升级能力都非常重要，当前LVS无法满足单点故障和升级不影响业务的需求。集群部署中任意一台LVS发生故障、节点升级流量都会中断，最严重的情况可能导致整个集群的流量异常（交换机不带一致性hash）。主备部署虽然能够通过keepalived切换主备，但是已有的流量依旧会中断。流表同步通过LVS节点间同步流表，达到节点升级或者故障客户无感知。

集群部署架构

LVS集群中通过组播将自己的流表信息同步到集群中的其他lvs节点上，保证任意台lvs都有一份全局的流表。当lvs集群中一台lvs需要升级或者故障下线时，防火墙或者vrouter将发给lvs1的数据转发给了其他lvs，由于lvs2和Lvs3也有lvs1的流表信息，所以也能正常转发。同时由于fullnat模式下每台LVS的出口ip（LIP）不一致，为了保证转发到nginx的不会断，需在后端nginx增加IPNAT模块，用于保证和nginx建立连接的ip保持不变。



说明

- 该特性不支持虚拟机场景，物理机场景下用户必须关闭LVS节点网卡GRO、LRO功能。
- 该特性不支持FTP协议，只支持TCP协议的应用层业务。
- 该特性支持fullnat、FNATinVXLAN与NATinVXLAN模式。

约束限制

注意

- ipvs-switch切换不能与ipvsadm启动同步进程、ipvsadm -l --daemon操作并发，否则会有造成系统oops的风险。
- 后端节点插入ipnat模块后不允许再插入ip_vs模块，或者执行ipvsadm相关命令，否则会导致业务异常或系统异常。

1. 同步广播报文中有syncid字段，该字段可以用于同网络部署多个LVS的时候，可以分割广播组。当前仅支持LVS集群配置单个syncid，所有LVS集群节点syncid必须相同。
2. 内核态LVS组播地址224.0.0.81是属于IANA预留的地址，该地址IANA未指定，但是不能保证以后被指定。因此流表同步的组网中不能有其他业务使用该广播地址。
3. 流表同步开启后，集群的新建连接能力和同步能力只能支持到“单节点”的能力。

环境准备

- 流表同步前需要增大系统socket buffer的值，尤其环境流量较大的情况下，否则同步会出错ip_vs_send_async error。配置举例：

```
sysctl -w net.core.wmem_max=412992000
sysctl -w net.core.rmem_max=412992000
```

- 网卡mtu：lvs节点间同步网卡的mtu值需要保持一致，否则mtu值大的节点往mtu值小的节点进行流表同步会失败。

配置步骤

步骤1 后端节点插入ipnat模块。

ipnat模块用于lvs后端节点，通过解析报文tcpoption字段中的client ip和port，与realserver建立流表，这样realserver节点看到的客户端ip就是client ip，当lvs节点故障或者升级时，数据从其他lvs节点访问realserver，虽然local ip发生变化，但client ip和port与realserver建立的流表没有变化，所以依然能够命中，同时client ip和port没有发生变化，所以realserver认为是一条连接，不会断开。

步骤2 使能关闭同步功能。

```
ipvsadm --start-daemon backup  --mcast-interface ethx --syncid xx
ipvsadm --start-daemon master  --mcast-interface ethx --syncid xx
ipvsadm --stop-daemon backup
ipvsadm --stop-daemon master
```

假设集群中有4台lvs，通过上面配置，lvs节点都向224.0.0.81的8848端口发送组播报文，同时也从224.0.0.81的8848端口接收同步的UDP组播报文。

说明

- lvs节点和后端必须在同一个vlan中。
- 使用ipvs-switch fnat/orig切换模式前，需要先使用ipvsadm --stop-daemon关闭同步功能，否则由于依赖关系会导致部分驱动无法删除。
- syncid取值（0-6特殊用途），ipvsadm启动同步进程时，syncid取值范围为[7-255]。

步骤3 调整同步间隔、组播的端口数

1. 当前流表同步间隔为“3 50”。

```
[root@localhost /]# cat /proc/sys/net/ipv4/vs/sync_threshold
3       50
```

该参数表示一条流中每50个报文的第3个触发同步。该参数需要客户根据网上实际情况修改，无损升级场景建议修改为1 2，每个报文都同步，降低流表无法同步的概率。

LVS设置同步间隔命令：

```
echo "1 2" > /proc/sys/net/ipv4/vs/sync_threshold
```

2. 组播端口数默认为“ 1”。

客户可以通过查看网卡的丢包数或者netstats -s查看UDP的丢包数，来确认是否需要增加该参数，增加socket数据，提升数据吞吐能力。

LVS设置端口数命令：

```
echo "8" > /proc/sys/net/ipv4/vs/sync_port_num
```

说明

- 该参数修改后需要先调用**步骤2**中的--stop-daemon，然后重新--start-daemon才能生效。
- 组播端口数最大值为64（即使该参数配置成100，启动同步进程时实际port数为64）。
- 流表同步间隔参数前者必须小于后者。

----结束

5.7 源地址透传

概述

当前TOA方案部署存在以下问题：

1. 客户来说操作复杂，每台服务器逐一部署，影响用户体验。
2. TOA模块支持的操作系统有限，例如无法支持windows系统，无法满足某些客户需求。
3. 客户设置的iptables安全规则可能导致源地址被过滤。
4. 不支持UDP。

接口配置

源地址透传的配置文件举例如下：

```
LVS_VXLAN_PARAM {
        id 400 #lvs的序号
        local_vtepaddr 10.41.11.24 #本地vxlan外层封装地址
}

virtual_server 192.168.31.200 80 {   #vip地址和端口
```

```
    delay_loop 6    #健康检查时间间隔
    lb_algo rr      #LVS调度算法，支持rr|wrr|lc|wlc|lblc|sh|dh|consh
    persistence_timeout 60 #LVS持续会话超时时间，单位秒
    lb_kind FNAT #转发模式，源地址透传模式下仅支持FNAT
    protocol TCP    #支持的协议模式是TCP还是UDP
    laddr_group_name laddr_g1   #指定local ip对应的group名称
    alpha           #开启alpha模式，在keepalived启动时，假设所有的RS都是down，等健康检查通过后rs才
上线。
    omega           #开启omega模式，在keepalived终止时，会执行quorum_down指令所定义的脚本。
    checker_off     #关闭健康检查
    quorum 1        #设置服务是否有效的阀值
    hysteresis 0    #延迟系数（跟quorum 配合使用，保证小于quorum）

    #高于或低于阀值时会执行以下脚本。
    quorum_up "/usr/sbin/ip addr add 192.168.31.200/32 dev lo;"
    real_server 192.168.31.248 80 {     #真实服务ip和端口
        weight 6                    #该实节点权重
        mac 52:54:00:04:xx:xx       #rs的mac地址
        vni 100                     #lvs后端L2GW的vni号
        vtepaddr 10.41.11.12 4789   #L2GW的vtep ip和端口
        dest_lb_kind NATinVXLAN     #member转发模式，支持NATinVXLAN、FNATinVXLAN
        TCP_CHECK {                 #健康检查方式
            connect_port 80         #检查的端口
            connect_timeout 15      #连接超时时间
            nb_get_retry 5          #重连次数
            delay_before_retry 3    #重连间隔
        }
    }
}
```

1. lvs模式新增NATinVXLAN、FNATinVXLAN两种模式，在原来的nat和fnat模式转发的报文中增加vxlan隧道。由于需要CNA以及L2GW等组网配合，当前只支持公有云ELB场景。
2. NATinVXLAN模式支持kvm虚拟化，不支持xen虚拟化模式，xen虚拟机可沿用原来fnat模式。
3. FNATinVXLAN用于ip target场景。
4. 支持根据不同的member设置不同的转发模式，member中的**dest_lb_kind**参数如果没有配置，则使用**lb_kind**设置的转发模式（**源地址透传场景下lb_kind只支持配置为FNAT模式**）。
5. keepalived健康检查子进程因参数值为空等错误配置导致进程异常退出时，NATinVXLAN模式下可能会出现重复的rs后端配置。
6. NATinVXLAN、FNATinVXLAN模式下不支持ipvsadm设置、删除操作，只支持ipvsadm查询。
7. NATinVXLAN、FNATinVXLAN模式下，相同rip但vni不同的后端不支持chekcer_merge功能。

说明

- member下的**vtepaddr**参数必须配置port。
- 内层报文默认按照1500的mtu来判断报文是否分片，用户可通过/proc/sys/net/ipv4/vs/dev_mtu接口修改。

5.8 增量加载

概述

在当前的配置修改方案中，keepalived如果需要修改配置，则需要修改配置文件，然后reload进程。在配置大量后端的情况下，这种操作需要大量的时间来重新上线，效率低

下，缺乏竞争力。新的方案中，在保持原来的reload的方式不变的情况下，提供了通过jsonrpc的方式来增量进行配置。当前支持删除、添加和更新virtual_server三个jsonrpc接口。

使用方法

1. 首先需要安装lvscli包。
```
yum install lvscli
yum install lvscli-devel
```
2. 通过lvscli命令行工具来调用接口（不保证商用）。
   - 删除virtual_server用法如下：
     ```
     lvscli keep del {{address}} {{port}} [protocol]
     ```
     其中address和port分别是要删除的virtual server的地址和端口。protocol是可选字段，可为TCP或UDP，默认是TCP
   - 新增virtual_server的用法如下：
     ```
     lvscli keep add {{conf_file}}
     ```
     conf_file是新增的增量配置文件的路径。
   - 更新virtual_server的用法如下：
     ```
     lvscli keep set {{conf_file}}
     ```
     conf_file与新增接口相同时配置文件的路径。
3. 通过C接口来调用rpc接口。
   - 新增keepalived config配置
     ```
     int lvs_add_service_config(char *svc_config);
     ```
   - 修改keepalived config配置
     ```
     int lvs_update_service_config(char *svc_config);
     ```
   - 删除keepalived config配置
     ```
     int lvs_del_service_config(struct lvs_service *svc);
     ```

   示例代码：
```
#include <lvs/lvs.h>
#include <lvs/lvs_common.h>
struct lvs_service svc; //调用删除接口需要传入的结构体
svc.af=AF_INET; //设置地址family为ipv4
inet_pton(svc.af,"192.168.1.1",&svc.addr); //设置要删除的virtual_server的IP地址
svc.port=htons(1984); //设置要删除virtual_server的端口
svc.protocol=IPPROTO_TCP; //设置协议为TCP
ret = lvs_del_service_config(&svc);//调用接口

char * conf = "virtual_server 192.168.200.100 443 { \
    ... \
    real_server 192.168.201.100 443 { \
       ... \
    }\
    ...\
}"; //要新增的virtual server配置
ret = lvs_add_service_config(conf); //调用新增vs接口
ret = lvs_update_service_config(conf); //调用更新vs接口
```


说明

1. 本配置文件不能嵌套include; 根配置文件只有local_address_group对象可以使用。
2. 配置文件中的virtual server不支持配置group。
3. 当前仅支持ipv4的地址。
4. 每次调用接口时，仅允许更新单个virtual server的配置文件，当存在多个virtual server需要操作时，必须分多个配置文件多次进行接口调用。
5. 当根配置文件的local_address_group更新时，必须重新reload或者restart keepalived 服务方可生效。
6. 通过add、update、delete操作的vs配置，用户必须保证其对应的conf文件必须同步在磁盘进行更新，否则将引起keepalived在reload或者restart后出现配置不一致的情况。
7. 不支持smtp方式的健康检查。
8. 单个配置的最大字符串长度为 960KB。
9. 如果用户新增的配置配置了quorum_up参数，可能会导致接口性能下降, 因为接口调用时可能需要执行用户配置的命令，属用户配置行为导致接口执行时间变长。
10. cpu占用率 100% 时，热加载耗时可能达到秒级。

6 命令参考

1. ipvsadm
   - 查看lvs的配置的负载均衡的连接配置
     ```
     ipvsadm -ln
     ```
   - 查看激活的连接(注意，由于用户态限制，lnc中倒计时使用cpu的tsc寄存器，溢出周期为100年以上)
     ```
     ipvsadm -lnc
     ```
   - 清空配置
     ```
     ipvsadm -C
     ```
2. iptables
   - iptables规则备份
     ```
     iptables-save > iptables.dump
     ```
   - 恢复iptables规则
     ```
     iptables-restore iptables.dump
     ```
3. rpm
   - 安装rpm包
     ```
     rpm -ivh xxx.rpm
     ```
   - 查询是否安装了rpm包
     ```
     rpm -qa | grep xxx
     ```
   - 卸载rpm包
     ```
     rpm -e xxx
     ```

