



DASHUANG DU.

study and practice, years of it.

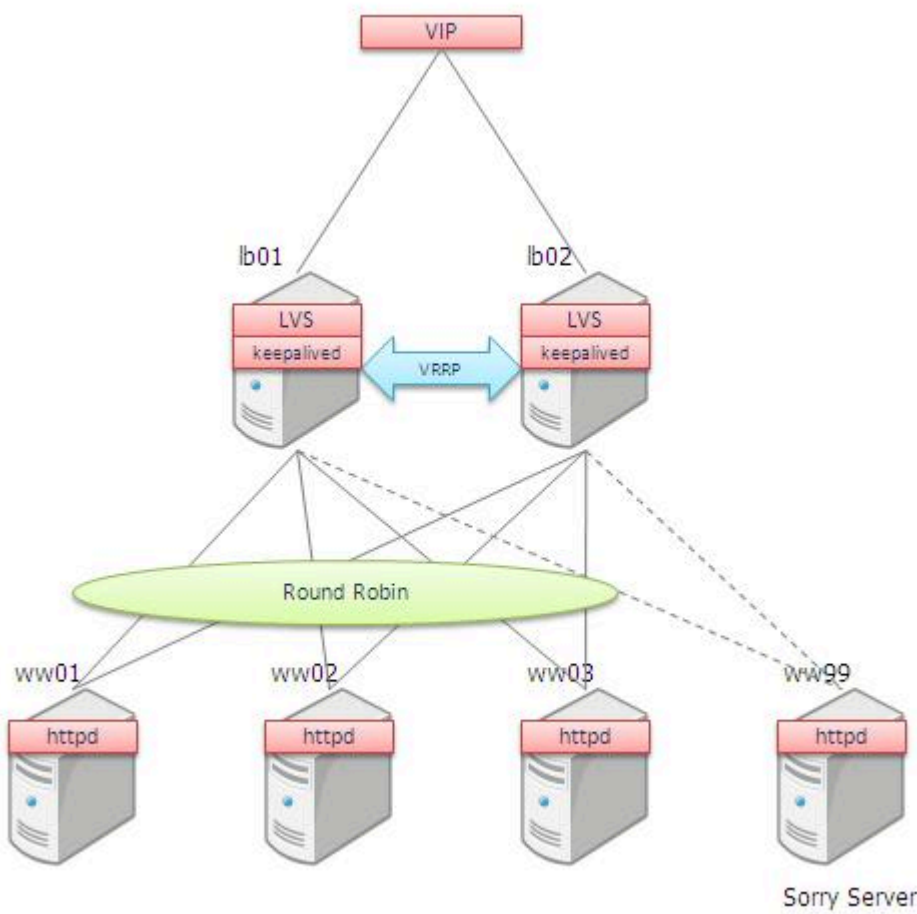
基于 LVS + Keepalived 搭建负载均衡

2018, MAY 28

LVS Keepalived 集群的组成

LVS (linux virtual server, linux 虚拟服务器)，由以下三部分组成：

- 负载均衡层 (load balance)：** 位于整个集群的最前端，由一台或多台负载调度器 (Director Server) 组成。其上安装了 LVS 的核心模块 IPVS，负责把用户请求分发给服务器群组层的应用服务器 (Real Server)。同时还有监控模块 (Ldirectord)，用于监测各个 Real Server 服务的健康情况，当 real server 不可用时， 将其从路由表中剔除，待主机恢复后重新加入。
- 服务器群组 (Server Array)：** 由一组实际运行应用服务的主机组成。每个 Real Server 之间通过高速 LAN 相连。
- 数据共享存储 (Shared Storage)：** 为所有 Real Server 提供共享存储空间和内容一致性的存储区域，一般由磁盘阵列设备组成。



IPVS 的简单介绍

IPVS： 安装于 Director Server 上，并在 Director Server 上虚拟出一个 VIP (Virtual IP)。用户的访问请求通过 VIP 到达负载调度器，然后由负载调度器从 Real Server 列表中选取一个服务节点响应用户的请求。

IPVS 转发请求的 3 种方式：

- VS/NAT (Virtual Server via Network Address Translation)：** 当用户请求到达调度器时，调度器将请求报文的目标地址和端口地址改写 成选定的 Real Server 的相应地址和端口，并将请求报文发送给选定的 Real Server。当 Real Server 返回数据时， 还需要再次将报文的源地址和端口更改为 VIP 和相应的端口后，再发送给用户。 因为请求和响应报文都需要经过 Director Server 重写， 所以当高并发时， 调度器的处理能力将会成为瓶颈。
- VS/TUN (Virtual Server via IP Tunneling)：** 也就是 IP 隧道技术实现虚拟服务器。调度器采用 IP 隧道技术将用户请求转发到某个 Real Server，而这个 Real Server 将直接响应用户的请求，不再经过前端调度器，此外，对 Real Server 的地域位置没有要求，可以和 Director Server 位于同一个网段，也可以是独立的一个网络。由于在 TUN 方式中，调度器将只处理用户的报文请求， 集群系统的吞吐量大大提高。
- VS/DR (Virtual Server via Direct Routing)：** 也就是用直接路由技术实现虚拟服务器。VS/DR 通过改写请求报文的 MAC 地址，将请求发送到 Real Server，而 Real Server 将响应直接返回给客户，免去了 VS/TUN 中的 IP 隧道开销。这种方式是三种负载调度机制中性能最高、最好的，但是必须要求 Director Server 与 Real Server 都有一块网卡连在同一物理网段上。

IPVS 的负载调度算法：

上面我们谈到，负载调度器是根据各个服务器的负载情况，动态地选择一台 Real Server 响应用户请求，那么动态选择是如何实现呢，其实也就是我们这里要说的负载调度算法， 根据不同的网络服务需求和服务器配置，IPVS 实现了如下八种负载调度算法， 这里我们详细讲述最常用的四种调度算法， 剩余的四种种调度算法请参考其它资料。

- 轮询调度 (Round Robin)

“轮询”调度也叫1:1调度，调度器通过“轮询”调度算法将外部用户请求按顺序1:1的分配到集群中的每个 Real Server 上， 这种算法平等地对待每一台 Real Server， 而不管服务器上实际的负载状况和连接状态。

- 加权轮询调度 (Weighted Round Robin)

“加权轮询”调度算法是根据 Real Server 的不同处理能力来调度访问请求。可以对每台 Real Server 设置不同的调度权值，对于性能相对较好的 Real Server 可以设置较高的权值，而对于处理能力较弱的 Real Server，可以设置较低的权值，这样保证了处理能力强的服务器处理更多的访问流量，充分合理的利用了服务器资源。同时，调度器还可以自动查询 Real Server 的负载情况，并动态地调整其权值。

- 最少链接调度（Least Connections）

“最少连接”调度算法动态地将网络请求调度到已建立的链接数最少的服务器上。如果集群系统的真实服务器具有相近的系统性能，采用“最小连接”调度算法可以较好地均衡负载。

- 加权最少链接调度（Weighted Least Connections）

“加权最少链接调度”，每个服务节点可以用相应的权值表示其处理能力，而系统管理员可以动态的设置相应的权值，缺省权值为1，加权最小连接调度在分配新连接请求时尽可能使服务节点的已建立连接数和其权值成正比。

- 其它四种调度算法分别为：
 - 基于局部性的最少链接（Locality-Based Least Connections）
 - 带复制的基于局部性最少链接（Locality-Based Least Connections with Replication）
 - 目标地址散列（Destination Hashing）
 - 源地址散列（Source Hashing）

keepalived 的原理的简单介绍：

VRRP（Virtual Router Redundancy Protocol，虚拟路由器冗余协议）： 在现实的网络环境中，主机之间的通信都是通过配置静态路由（默认网关）完成的，而主机之间的路由器一旦出现故障，通信就会失败，因此，在这种通信模式中，路由器就成了一个单点瓶颈，为了解决这个问题，就引入了 VRRP 协议。

VRRP 可以将两台或多台物理路由器设备虚拟成一个虚拟路由器，每个虚拟路由器都有一个唯一标识，称为 VRID，一个 VRID 与一组 IP 地址构成了一个虚拟路由器。这个虚拟路由器通过虚拟IP（一个或多个）对外提供服务。而在虚拟路由器内部，同一时间只有一台物理路由器对外提供服务，这台物理路由器被称为主路由器（处于 MASTER 角色）。而其他物理路由器不拥有对外的虚拟 IP，也不提供对外网络功能，仅仅接收 MASTER 的 VRRP 状态通告信息，这些路由器被统称为备份路由器（处于 BACKUP 角色）。当主路由器失效时，处于 BACKUP 角色的备份路由器将重新进行选举，产生一个新的主路由器进入 MASTER 角色继续提供对外服务，整个切换过程对用户来说完全透明。

Keepalived 作为一个高性能集群软件，它还能实现对集群中服务器运行状态的监控及故障隔离。下面继续介绍下 Keepalived 对服务器运行状态监控和检测的工作原理。

Keepalived 工作在 TCP/IP 参考模型的第三、第四和第五层，也就是网络层、传输层和应用层。根据 TCP/IP 参考模型各层所能实现的功能，Keepalived运行机制如下：

- 在网络层，运行着四个重要的协议：互连网协议 IP、互连网控制报文协议 ICMP、地址转换协议 ARP 以及反向地址转换协议 RARP。Keepalived 在网络层采 用的最常见的工作方式是通过ICMP协议向服务器集群中的每个节点发送一个 ICMP 的数据包（类似于 ping 实现的功能），如果某个节点没有返回响应数据包，那么 就认为此节点发生了故障，Keepalived 将报告此节点失效，并从服务器集群中剔除故障节点。
- 在传输层，提供了两个主要的协议：传输控制协议 TCP 和用户数据协议 UDP。传输控制协议 TCP 可以提供可靠的数据传输服务，IP 地址和端口，代表一个 TCP 连接的一个连接端。要获得 TCP 服务,须在发送机的一个端口上和接收机的一个端口上建立连接，而 Keepalived 在传输层就是利用 TCP 协议的端口连接和扫描技术来 判断集群节点是否正常的。比如，对于常见的 Web 服务默认的 80 端口、SSH 服务默认的 22 端口等，Keepalived 一旦在传输层探测到这些端口没有响应数据返回，就认为这些端口发生异常，然后强制将此端口对应的节点从服务器集群组中移除。
- 在应用层，可以运行 FTP、TELNET、SMTP、DNS 等各种不同类型的高层协议，Keepalived 的运行方式也更加全面化和复杂化，用户可以通过自定义 Keepalived 的工作方式，例如用户可以通过编写程序来运行 Keepalived，而 Keepalived 将根据用户的设定检测各种程序或服务是否允许正常，如果 Keepalived 的检测结果 与用户设定不一致时，Keepalived 将把对应的服务从服务器中移除。

安装

环境：

hostname	ip	remark
lvs-master	192.168.12.100	lvs主机
lvs-slave	192.168.12.101	lvs备用机
web-server1	192.168.12.102	realserver1
web-server2	192.168.12.103	realserver2

本次安装教程采用的是DR转发方式，上述机器均为ubuntu16.04

VIP: 192.168.12.200

安装 ipvsadm, keepalived

```
sudo apt-get install ipvsadm keepalived
```

查看路由转发设置

```
sudo ipvsadm

***** vagrant@vagrant:~ [05:38:21] *****
% sudo ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
```

keppalived 配置

修改keepalived的配置文件，没有就新建一个

```
sudo vim /etc/keepalived/keepalived.conf
```

```
# 全局配置
global_defs {
    router_id LVS_DEVEL
    notification_email {
        dudashuang1222@gmail.com
    }
    notification_email_from Alexandre.Cassen@firewall.loc
    smtp_server 192.168.12.30
    smtp_connect_timeout 30
    vrrp_skip_check_adv_addr
    vrrp_strict
}

# VRRP 实例配置
vrrp_instance VI_1 {
    state MASTER #如果使用主/备,另外一台机器需要设置为BACKUP
    interface eth1 #检测网络端口
    virtual_router_id 100 #主备的虚拟机路由ID必须一致
    priority 100 #主备的优先级,主优先级要大于备
    advert_int 1 #VRRP Multicast广播周期秒数
    authentication {
        auth_type PASS # VRRP认证方式
        auth_pass 123456 # VRRP口令字
    }
    virtual_ipaddress {
        192.168.12.200 # 如果有多个VIP,继续换行填写
    }
}

# 虚拟服务器配置
virtual_server 192.168.12.200 80 {
    delay_loop 1 # 每隔1秒查询realserver状态
    lb_algo wrr # lvs 算法
    lb_kind DR # Direct Route
    protocol TCP # 用TCP协议检查realserver状态
    persistence_timeout 50 # 会话保持时间,这段时间内,同一ip发起的请求将被转发到同一个realserver

    # 第一台realserver物理环境
    real_server 192.168.12.102 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
            connect_port 80
        }
    }

    # 第二台realserver物理环境
    real_server 192.168.12.103 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
            connect_port 80
        }
    }
}
```

interface eth1 在这里为外置网卡的端口，可以执行ip a查看192.168.12.100在哪个端口，若为eth0，则填eth0

lvs-slave的配置和主机一样，仅需要将

```
state MASTER --> BACKUP
priority 100 --> 90
```

启动 keepalived

```
# 清除路由设置
sudo ipvsadm -C
```



```
# 重启 keepalived
sudo service keepalived restart
```

此时查看路由配置

```
***** vagrant@vagrant:~ [05:38:39] *****
% sudo ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  localhost:http wrr persistent 50
  -> localhost:http                Route   1      0          0
  -> localhost:http                Route   1      0          0
```

realserver 设置

配置/etc/rc.local，添加下列配置，然后重启机器

```
ifconfig lo:0 192.168.12.200 broadcast 192.168.12.200 netmask 255.255.255.255 up
route add -host 192.168.12.200 dev lo:0
echo "0" > /proc/sys/net/ipv4/ip_forward
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
```

更新 nginx 首页

```
sudo vim /var/www/html/index.nginx-debian.html

# realserver2 换成192.168.12.103
<h2>ip:192.168.12.102</h2>
```

测试

访问<http://192.168.12.200>



假设lvs-master宕机

```
sudo service keepalived stop
```

假设 realserver1 宕机

sudo service nginx stop

参考原文：
 <https://www.jianshu.com/p/88fea188c203>



LOAD BALANCE

TOP

CONTACT ME



Copyright © 2022 dudashuang.com 版权所有 津ICP备17007417号

