LVS + Keepalived + Mycat

分布式数据库高可用实施方案

文档信息

文档编写人	HanSenJ	编写日期	2016-08-01
文档评审人		评审日期	

版本修订历史记录

版本号	作者	参与者	起止日期	备注
V1.0				

内容修订历史记录

章节	修改内容	修订人	修订日期	修订原因

目录

高可用体系介绍

什么是高可用

高可用性 H.A.（High Availability）指的是通过尽量缩短因日常维护操作（计划）和突发的系统崩溃（非计划）所导致的停机时间，以提高系统和应用的可用性。它与被认为是不间断操作的容错技术有所不同。HA 系统是目前企业防止核心计算机系统因故障停机的最有效手段。

高可用集群的衡量标准

高可用性群集是通过系统的可靠性(reliability)和可维护性(maintainability)来度量的。工程上，通常用平均无故障时间(MTTF)来度量系统的可靠性,用平均维修时间（MTTR）来度量系统的可维护性。于是可用性被定义为：HA=MTTF/(MTTF+MTTR)*100%

具体 HA 衡量标准:

99% 一年宕机时间不超过 4 天

99.9% 一年宕机时间不超过 10 小时

99.99% 一年宕机时间不超过 1 小时

99.999% 一年宕机时间不超过 6 分钟

高可用集群软件，常用组合

heartbeat v2+haresource 一般常用于 CentOS 5.X

heartbeat v3+pacemaker 一般常用于 CentOS 6.X

corosync+pacemaker 现行常用的组合

cman + rgmanager 红帽集群套件中的组件

keepalived+lvs 常用于 lvs 的高可用

、

Keepalived 简介

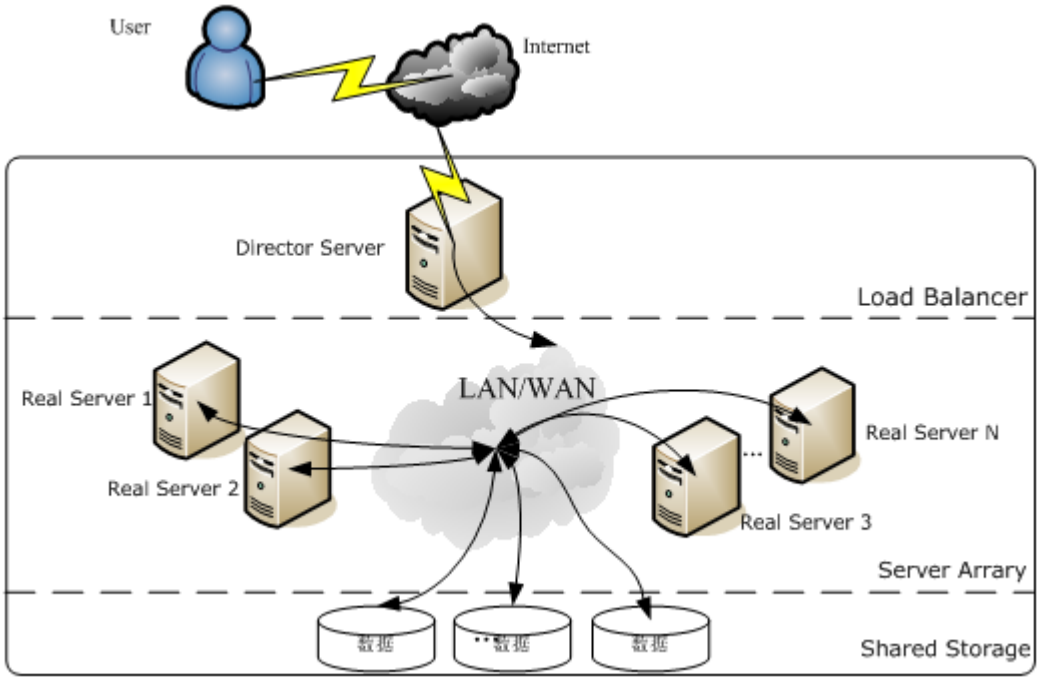
keepalived 是以 VRRP 协议为实现基础的，VRRP 全称 Virtual Router Redundancy Protocol，即虚拟路由冗余协议。

虚拟路由冗余协议，可以认为是实现路由器高可用的协议，即将 N 台提供相同功能的路由器组成一个路由器组，这个组里面有一个 master 和多个 backup，master 上面有一个对外提供服务的 vip（该路由器所在局域网内其他机器的默认路由为该 vip），master 会发组播，当 backup 收不到 vrrp 包时就认为 master 宕掉了，这时就需要根据 VRRP 的优先级来选举一个 backup 当 master。这样的话就可以保证路由器的高可用了。

keepalived 主要有三个模块，分别是 core、check 和 vrrp。core 模块为 keepalived 的核心，负责主进程的启动、维护以及全局配置文件的加载和解析。check 负责健康检查，包括常见的各种检查方式。vrrp 模块是来实现 VRRP 协议的。

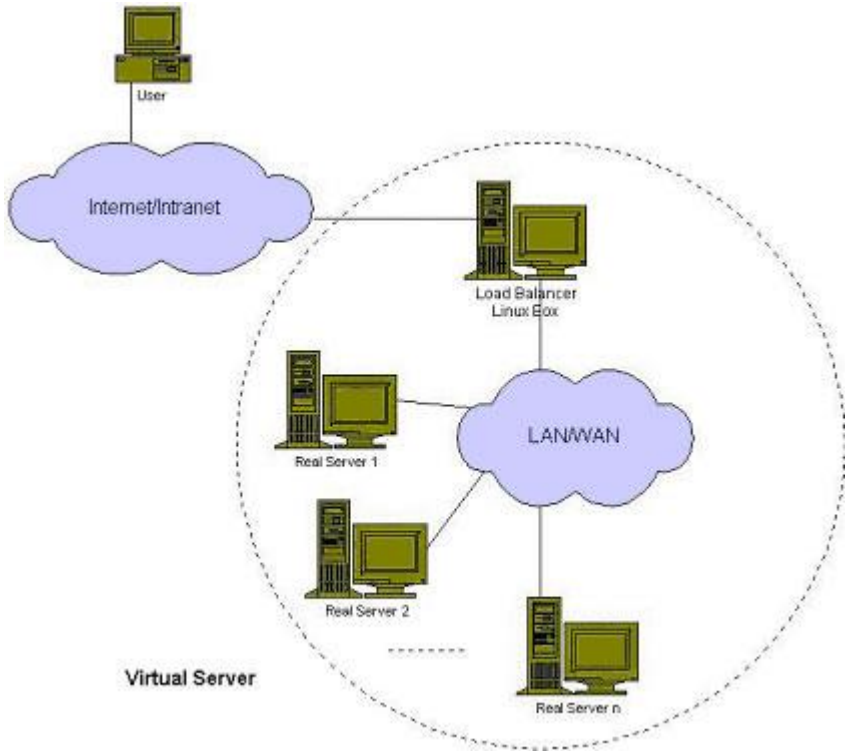
LVS(Linux Virtual Server)是一个高可用性虚拟的服务器集群系统。本项目在 1998 年 5 月由章文嵩博士成立，是中国国内最早出现的自由软件项目之一。

LVS 主要用于多服务器的负载均衡，作用于网络层。LVS 构建的服务器集群系统中，前端的负载均衡层被称为 Director Server；后端提供服务的服务器组层被称为 Real Server。通过下图可以大致了解 LVS 的基础架构。



LVS 简介

LVS 有三种工作模式，分别是 DR（Direct Routing 直接路由）、TUN(Tunneling IP 隧道）、NAT（Network Address Translation 网络地址转换）。其中 TUN 模式能够支持更多的 Real Server，但需要所有服务器支持 IP 隧道协议；DR 也可以支持相当的 Real Server，但需要保证 Director Server 虚拟网卡与物理网卡在同一网段；NAT 扩展性有限，无法支持更多的 Real Server，因为所有的请求包和应答包都需要 Director Server 进行解析再生，影响效率。 同时，LVS 负载均衡有 10 中调度算法，分别是 rr、wrr、lc、wlc、lblc、lblcr、dh、sh、sed、nq。

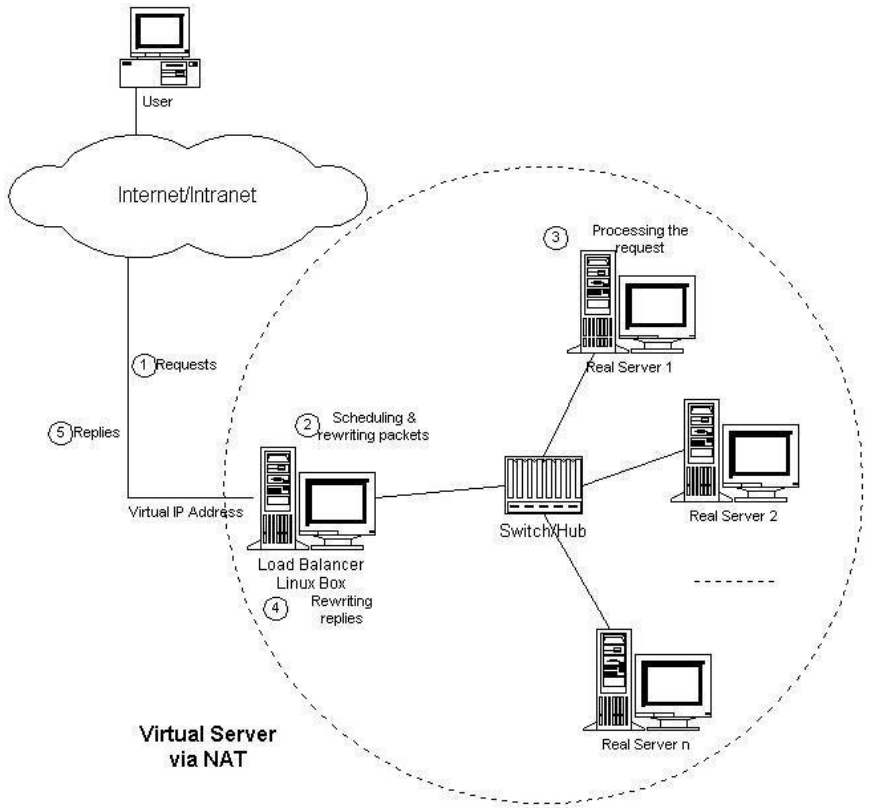


Load Balancer 是负载调度器，由它将网络请求无缝隙调度到真实服务器，至于此集群使用的是哪一种 IP 负载均衡技术(LVS 有三种负载均衡技术，分别是 VS/NAT、VS/TUN 和 VS/DR)。LVS 与其他基于应用层或基于 IP 层的负载均衡应用拥有类似的一点：一台及其以上的负载调度器和数台甚至成百上千台真实服务器。用 LVS 来搭建负载均衡集群，理论上来说，只需要在负载调度器上安装 LVS 核心软件 ipvs 和 ipvs 的功能实现软件 ipvsadm，而真实服务器无需额外安装软件。

当前，大部分 Linux 发行版本已经集成了 ipvs，因此我们只需要安装它的实现软件 ipvsadm 即可。在安装这个软件的时候，我们不需要考虑我们将使用哪一种 IP 负载均衡技术，直接安装即可。

以下介绍三种调度方式的配置方法。

VS/NAT 体系结构图



NAT 本身是一种将私有地址转换为合法 IP 地址的一种技术，在 VS/NAT 结构中，整个集群系统只有一个对外的合法地址，这个 IP 在 Load Balancer 上对外可见，其他的真实服务器与这个调度器组成了一个对外不可见的内网，Internet 上的用户访问集群必须通过集群提供的对外 IP 访问调度器，调度器再利用 NAT 技术，将真实服务器置于网络上。

注意事项：

1、VS/NAT 的真实服务器可以使用任何机器，任何操作系统，支持大部分的网络服务(如：httpd、ftpd、telnetd 等等)，唯一的要求就是真实服务器必须支持 TCP/IP，不过我们可以忽略这样要求。

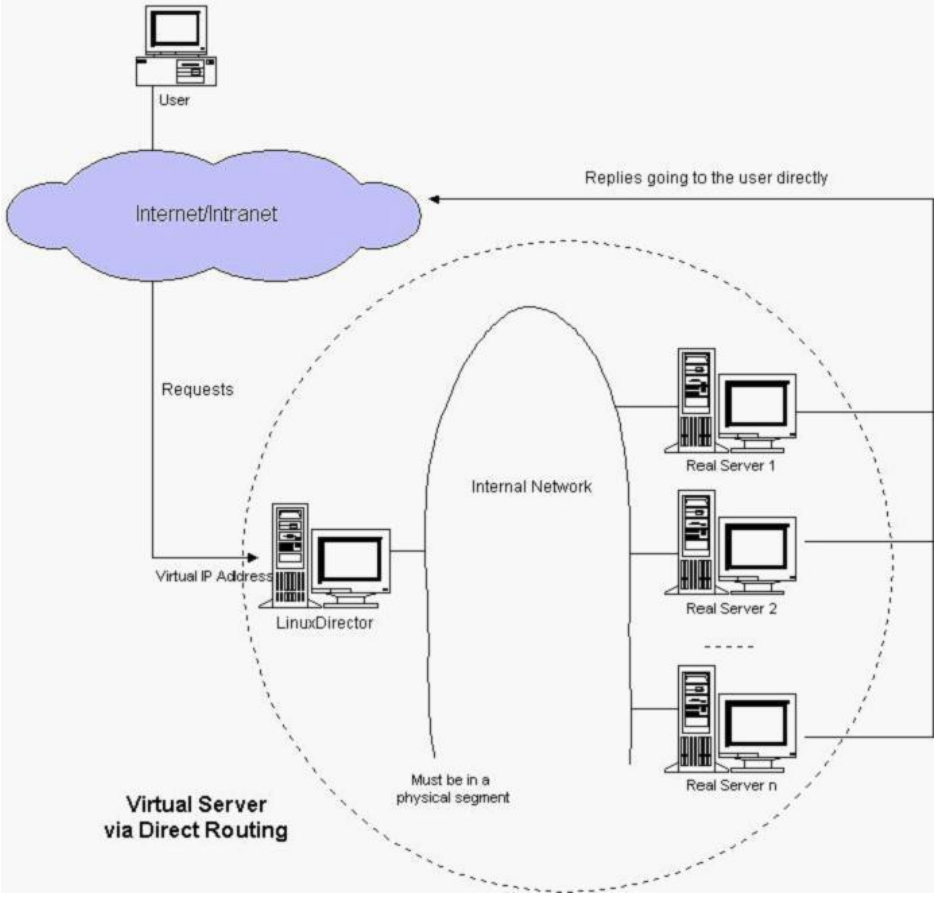
2、在 VS/NAT 模式下，必须打开 Load Balancer 的 ip_forward，关闭 ICMP 重定向。

3、所有的真实服务器网关地址必须指向到调度器的内网地址。

4、你的真实服务器上必须已部署好 http 服务器，并为缺省主页写入不同的内容以测试调度器调度结果。

5、客户机和真实服务器在两个不同的网络中，理论上也可以在一个网络中，需要做一些额外的设置。

VS/DR 体系结构



该图是官方提供的 VS/DR 体系结构图，在 VS/DR 模式下，Load Balancer 和所有的 Real Server 在物理上有一个网卡通过不分断的局域网相连，调度器和真实服务器必须绑定同一 VIP，该 VIP 在调度器上对外可见，而真实服务器上只需将 VIP 配置在 Non-ARP 网络设备上，它对外不可见，只是用于欺骗真实服务器用于处理目标地址为 VIP 的网络请求。真实服务器将请求处理后，直接返回给用户，不需要在通过调度器返回，所以在 VS/DR 模式下，真实服务器的网关地址不需要指向调度器。

注意事项：

1、基本上大部分搭载着 unices 和 Microsoft OS 的服务器都可以在 VS/DR 模式下作为真实服务器使用。

2、Load Balancer 和所有的 Real Server 在物理上必须有一个网卡通过不分断的局域网相连。

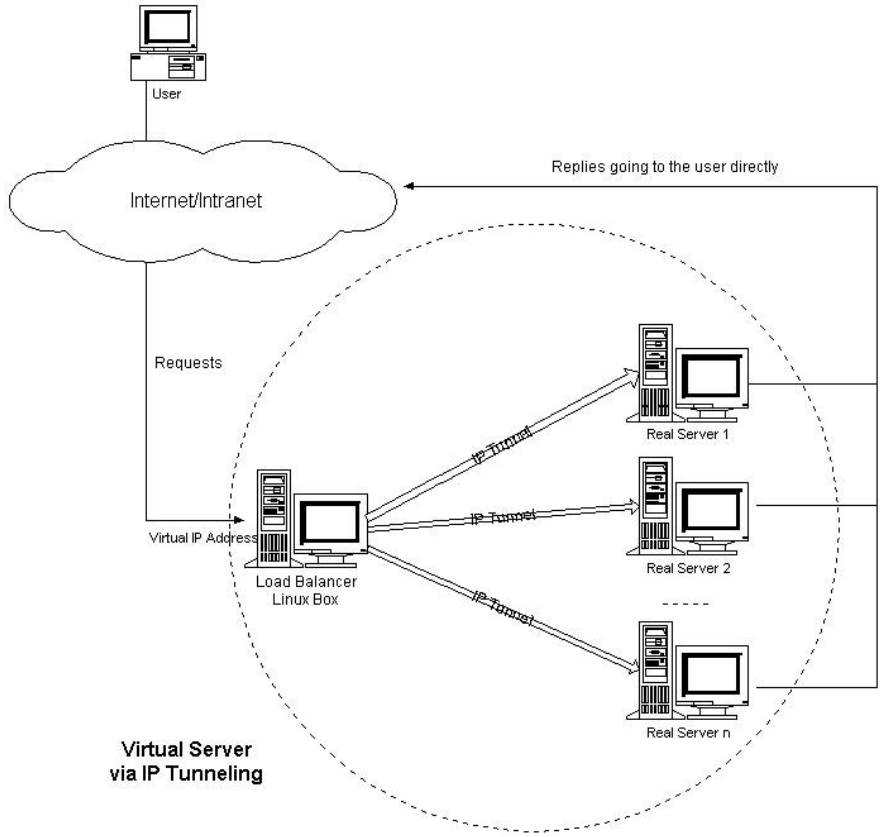
3、调度器上的 VIP 地址对外可见；真实服务器必须将 VIP 绑定到 Nor-ARP 网卡上，它对外不可见，只是用于欺骗真实服务器用于处理目标地址为 VIP 的网络请求。

4、在 VS/DR 模式下，无需使用 ip_forward 功能，因此为了安全考虑，关闭了该功能。

5、真实服务器不再使用调度器作为网关，因此打开调度器的 ICMP 重定向。

4、你的真实服务器上必须已部署好 http 服务器，并为缺省主页写入不同的内容以测试调度器调度结果。

VS/TUN 架构体系



这个体系结构中，调度器根据各个服务器的负载情况，动态地选择一台服务器，将请求报文封装在另一个 IP 报文中，再将封装后的 IP 报文转发给选出的服务器；服务器收到报文后，先将报文解封获得原来目标地址为 VIP 的报文，服务器发现 VIP 地址被配置在本地的 IP 隧道设备上，所以就处理这个请求，然后根据路由表将响应报文直接返回给客户。

因此，VS/TUN 的设置与 VS/DR 有部分相似之处，VS/TUN 的调度器和真实服务器也需要绑定一个 VIP，不同的是 VS/TUN 模式下，VIP 绑定的网卡与 VS/DR 不同，因为 VS/TUN 需要 ip 隧道支持。

注意事项：

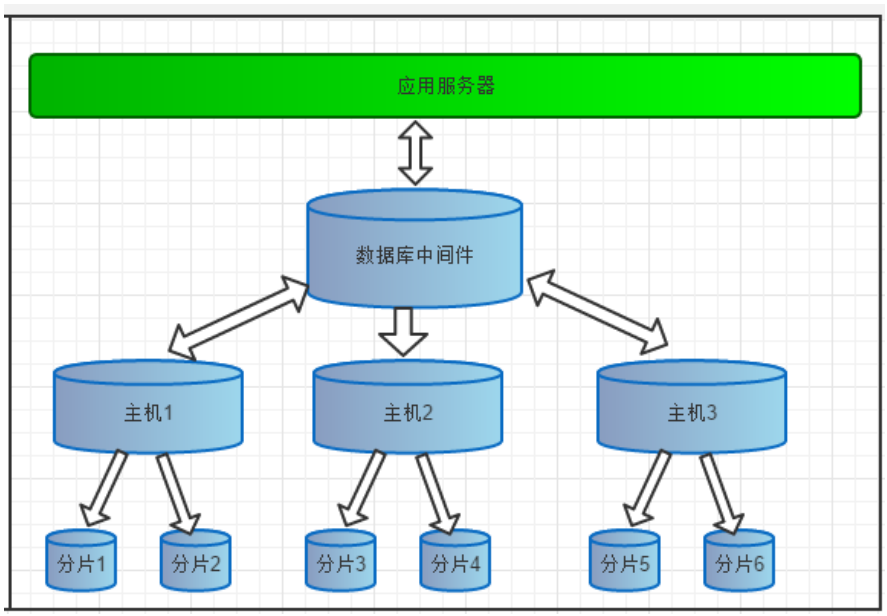
1、 VS/TUN 模式下的所有真实服务器必须使用 Linux 系统，因为迄今为止，只有 Linux 系统支持隧道技术。

2、Load Balancer 和 Real Server 的 VIP 必须绑定在 tunl0 网卡上，且关闭 Real Server 网卡 tunl0 的 arp 功能。

3、 你的真实服务器上必须已部署好 http 服务器，并为缺省主页写入不同的内容以测试调度器调度结果。

Mycat 简介

Mycat 是数据库中间件，就是介于数据库与应用之间，进行数据处理与交互的中间服务。由于对数据进行分片处理后，从原有的一个库被切换为分片数据库，所有的分片数据库集群构成了整个完整的数据库存储。

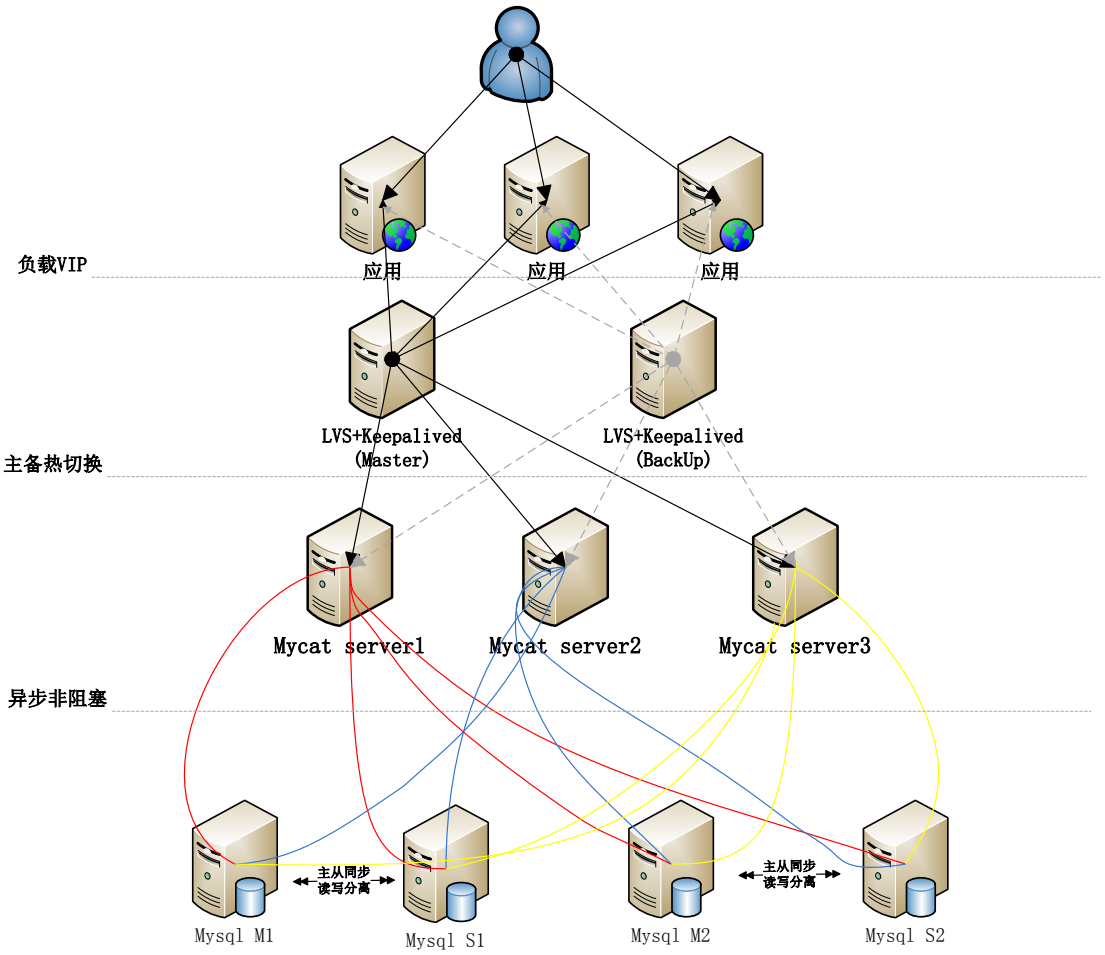


如上图所示，数据被分到多个分片数据库后，应用如果需要读取数据，就需要处理多个数据源的数据。如果没有数据库中间件，那么应用将直接面对分片集群，数据源切换、数据聚合都需要应用直接处理，原本该专注于业务的应用将花大量的工作来处理分片后的问题，最重要的是每个应用处理将是完全的重复制造轮子。

所以有了数据库中间件，应用只需要集中业务处理，大量的通用的数据聚合、事务，数据源切换都由中间件来处理，中间件的性能与处理能力将直接决定应用的读写性能，所以一款好的数据库中间件至关重要。

案例

通过 LVS +Keepalived+mycat 构建一套基于负载的高可用分布式数据库集群环境。



本章只讲述 LVS+Keepalived 的安装部署方案， 应用服务、Mycat 服务与 Mysql 集群环境不做介绍。

Keepalived 安装部署

一、软件版本

1、Keepalived软件版本：keepalived-1.1.20.tar.gz

二、环境配置

1、主Keepalived服务器IP地址 192.168.100.1

2、备Keepalived服务器IP地址 192.168.100.2

3、Keepalived虚拟IP地址 192.168.100.100

4、Mycat服务器IP地址　192.168.100.3

Mycat服务器IP地址　192.168.100.4

Mycat服务器IP地址　192.168.100.5

三、软件下载地址

http://www.keepalived.org/software/keepalived-1.1.20.tar.gz

四、安装流程

1、解压软件包：

[root@localhost~]$ sudo tar -xvf keepalived-1.2.23.tar.gz

[root@localhost~]$ cd keepalived-1.2.23/

[root@localhost~]]$ sudo ln -s /usr/src/kernels/2.6.9-78.EL-i686/usr/src//linux

[root@localhost~]$sudo ./configure 此处--prefix=PREFIX（指定安装路径）

--with-kernel-version=VER (指定内核版本）

若出现错误，则用yum 安装相应依赖包。

2、编译以及编译安装

[root@localhost~] sudo make && make install

3、修改配置文件路径

[root@localhost~]$ sudo cp /usr/local/etc/rc.d/init.d/keepalived　/etc/rc.d/init.d/

[root@localhost~]$ sudo cp /usr/local/etc/sysconfig/keepalived /etc/sysconfig/

[root@localhost~]$ sudo mkdir /etc/keepalived

[root@localhost~]$ sudo cp /usr/local/etc/keepalived/keepalived.conf /etc/keepalived/

[root@localhost~]$ sudo cp /usr/local/sbin/keepalived /usr/sbin/

4、设置为服务，开机启动

[root@localhost keepalived-1.1.20]# vi /etc/rc.local

加入service keepalived start

5、配置keepalived

MASTER配置Keepalived.conf：主服务器地址192.168.100.1

必须在/etc/keepalived/目录下编辑keepalived.conf

[root@localhost~]$ sudo vim /etc/keepalived.conf

global_defs { #全局配置标识，表面下面的区域{}是全局配置

notification_email {#邮箱地址，如果keepalived在发生诸如切换操作时会的发邮件到

配置上的邮箱，　　examp@qq.com
}
notification_email_from examp@qq.com #表示发送通知邮件时邮件源地址是谁
smtp_server 127.0.0.1 #表示发送email时使用的smtp服务器地址
smtp_connect_timeout 30 #连接smtp超时时间
router_id daas
}vrrp_instance VI_1 {#各服务器上实例配置域，这里按本服务器的具体情况填值state MASTER #本实例启动状态，MASTER／SLAVE，不管填MASTER／SLAVE，最终还是要看本机器的权重
interface eth0　#节点固有IP（非VIP）的网卡，用来发VRRP包。
virtual_router_id 51 #取值在0-255之间，用来区分多个instance的VRRP组播。
priority 100 #设置本节点的优先级，优先级高的为master，不能超过255
advert_int 1 #这里设置VRID，如果两台机器是同一个备份组，设置一样
authentication { #组播信息发送间隔，同一个备份组两个机器设置必须一样，默认是1S
auth_type PASS #验证域，同组的机器auth_type（验证类型）和auth_pass（验证密码）必须一样
auth_pass 1111
}
virtual_ipaddress {# VIP，为master机器设置的虚拟地址，和实例绑定的网卡（interface）设置到一个网段
192.168.100.100
}
}

virtual_server 192.168.100.100 8066 {
delay_loop 60 #延迟轮询时间（单位秒）
lb_algo rr #后端调试算法（load balancing algorithm）
lb_kind DR # LVS调度类型NAT/DR/TUN
persistence_timeout 50 #
protocol TCP
real_server 192.168.100.3 8066 { #真正提供Mycat服务的服务器地址
weight 3 #权重
TCP_CHECK {
connect_timeout 3 #超时连接时间
nb_get_retry 3 #健康检查时间秒，根据业务情况考量允许故障切换时间值。
delay_before_retry 3
connect_port 8066 #服务端口，此处要为MyCat服务提供端口保持一致
}
}
real_server 192.168.100.4 8066{//Mycat服务器二
weight 3
TCP_CHECK {
connect_timeout 3

```
            nb_get_retry 3
            delay_before_retry 3
            connect_port 8066
        }
    }
    real_server 192.168.100.5 8066 {//Mycat 服务器三
        weight 3
        TCP_CHECK {
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
            connect_port 8066
        }
    }
}
```

BACKUP配置Keepalived.conf：主服务器地址192.168.100.2

必须在/etc/keepalived/目录下编辑keepalived.conf

[root@localhost~]$ scp user@192.168.100.1 /etc/keepalived/ home/etc/keepalived/

将MASTER文件拷贝到BACKUP服务器上

[root@localhost~]$ sudo vim /etc/keepalived.conf

将vrrp_instance VI_1 MASTER修改为BACKUP　priority 100修改为99

5、启动服务

[root@localhost~]$ sudo service keepalived start

Starting keepalived: [OK]

检查进程状态

[root@localhost~]$ ps -aux|grep keepalived

```
root  5965  0.0  0.0  39964   844 ?   Ss  15:55  0:00 keepalived -D
root  5966  0.0  0.0  42196  2104 ?   S   15:55  0:00 keepalived -D
root  5967  0.0  0.0  42068  1340 ?   S   15:55  0:00 keepalived –D
```

LVS 安装部署

一、软件版本

1、 软件安装

检查Load Balancer服务器是否已支持ipvs。

[root@localhost~]$ sudo modprobe -l|grep ipvs

若有类似以下输出，则表示服务器已支持ipvs：

kernel/net/netfilter/ipvs/ip_vs.ko
kernel/net/netfilter/ipvs/ip_vs_rr.ko
kernel/net/netfilter/ipvs/ip_vs_wrr.ko
kernel/net/netfilter/ipvs/ip_vs_lc.ko
kernel/net/netfilter/ipvs/ip_vs_wlc.ko
kernel/net/netfilter/ipvs/ip_vs_lblc.ko
kernel/net/netfilter/ipvs/ip_vs_lblcr.ko
kernel/net/netfilter/ipvs/ip_vs_dh.ko
kernel/net/netfilter/ipvs/ip_vs_sh.ko
kernel/net/netfilter/ipvs/ip_vs_sed.ko
kernel/net/netfilter/ipvs/ip_vs_nq.ko
kernel/net/netfilter/ipvs/ip_vs_ftp.ko
kernel/net/netfilter/ipvs/ip_vs_pe_sip.ko

若已经支持ipvs ，则将

若服务器不支持ipvs，则需要手动下载ipvs并编译安装。

通过 yum 安装

[root@localhost~]$ sudo yum -y install ipvsadm*

2、 Realserver 脚本配置

分别对3台mycat server 进行realserver脚本的设置。（192.168.100.3、192.168.100.4、192.168.100.5）

[root@localhost~]$ sudo vim /etc/rc.local

/sbin/ifconfig lo:0 192.168.100.100 broadcast 192.168.100.100 netmask 255.255.255.255 up

/sbin/route add -host 192.168.100.100 dev lo:0

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

sysctl –p

可以将该段脚本加入启动脚本中，也可以单独设置 realserver 执行脚本，此处为了方便加入启动项中。

3、 启动服务、检查lvs 运行状态

[root@localhost~]$sudo service ipvsadm start

[root@localhost~]$sudo ipvsadm -ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 10.249.5.100:8066 rr persistent 50

-> 192.168.100.3:8066 Route 3 0 0

-> 192.168.100.4:8066 Route 3 0 0

-> 192.168.100.5:8066 Route 3 0 0

统计自该条转发规则生效以来的包：

[root@localhost~]$sudo ipvsadm -l –stats

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Conns InPkts OutPkts InBytes OutBytes

-> RemoteAddress:Port

TCP	192.168.100.100:8066	0	30	0	0	0
	192.168.100.3:8066	0	10	0	0	0
	192.168.100.4:8066	0	10	0	0	0
	192.168.100.5:8066	0	10	0	0	0

```
1. 1. Conns    (connections scheduled)  已经转发过的连接数
2. 2. InPkts   (incoming packets)       入包个数
3. 3. OutPkts  (outgoing packets)       出包个数
4. 4. InBytes  (incoming bytes)         入流量（字节）
5. 5. OutBytes (outgoing bytes)         出流量（字节）
```

检查速率信息：

[root@localhost~]$sudo ipvsadm -l –rate

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port CPS InPPS OutPPS InBPS OutBPS

-> RemoteAddress:Port

TCP	192.168.100.100:8066	0	350	0	0	0
	192.168.100.3:8066	0	120	0	0	0
	192.168.100.4:8066	0	130	0	0	0
	192.168.100.5:8066	0	100	0	0	0

```
1. 1. CPS      (current connection rate)   每秒连接数
2. 2. InPPS    (current in packet rate)    每秒的入包个数
3. 3. OutPPS   (current out packet rate)   每秒的出包个数
4. 4. InBPS    (current in byte rate)      每秒入流量（字节）
5. 5. OutBPS   (current out byte rate)      每秒入流量（字节）
```

高可用测试验证

验证暂不列入，请根据自身情况进行验证测试。