**－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－**

**集群**

一组通过高速网络互联的计算组。将很多服务器集中起来，提供同一种服务，在客户端看起来就像只有一个服务器。可以在付出较低成本的情况下获得在性能、可靠性、灵活性方面的相对较高的收益。

分类：高可用集群 负载均衡集群 高性能计算集群

高可用：在服务出现故障时，集群系统可以自动将服务从故障节点切换到另一个备用节点，从而提供不间断性服务，保证了业务的持续运行。常用的高可用开源软件包括keepalived，hertbeat。

keepalived的工作原理：

1、先介绍VRRP

keepalived高可用对之间通过VRRP通信的。VRRP，全称Virtual Router Redundancy Protocal，虚拟路由冗余协议，VRRP的出现是为了解决静态路由的单点故障。VRRP用IP多播的方式实现高可用对之间的通信，工作时主节点发包，备节点收包，当备节点接收不到主节点发的数据时，启动接管程序，接管主节点的资源。被节点可以有多个，通过优先级竞选。

2、再介绍Keepalived工作原理

   keepalived高可用对之间通过VRRP进行通信的，VRRP是通过竞选机制来确定主备的，主的优先级高于备，因此工作时主会优先获得所有的资源，备节点处于等待状态，当主挂了的时候，备节点就会接管主节点的资源，然后顶替主节点对外提供服务。

   在keepalived服务对之间，只有作为主的服务器会一直发送VRRP广播包，告诉备它还活着，此时备不会抢占主，当主不可用时，即备监听不到主发送的广播包时，就会启动相关服务接管资源，保证业务的连续性。

负载均衡：通常有两台或两台以上的服务器组成，分为前段负载调度，和后端节点两个部分。调度器获取客户端的请求，并通过自身定义的负载分担策略，将请求平均分配到后端各个服务节点，每个节点都承担一定的访问请求压力。

lvs负载均衡的模式

三种部署方式：VS/NAT（nat 模式） VS/DR(路由模式) VS/TUN（隧道模式）。

一、NAT模式（VS-NAT）

原理：就是把客户端发来的数据包的IP头的目的地址，在负载均衡器上换成其中一台RS的IP地址并发至此RS来处理,RS处理完后把数据交给负载均衡器,负载均衡器再把数据包原IP地址改为自己的IP将目的地址改为客户端IP地址即可期间,无论是进来的流量,还是出去的流量,都必须经过负载均衡器

优点：集群中的物理服务器可以使用任何支持TCP/IP操作系统，只有负载均衡器需要一个合法的IP地址

缺点：扩展性有限。当服务器节点（普通PC服务器）增长过多时,负载均衡器将成为整个系统的瓶颈因为所有的请求包和应答包的流向都经过负载均衡器。当服务器节点过多时大量的数据包都交汇在负载均衡器那，速度就会变慢！

二、IP隧道模式（VS-TUN）

原理：首先要知道，互联网上的大多Internet服务的请求包很短小，而应答包通常很大那么隧道模式就是，把客户端发来的数据包，封装一个新的IP头标记(仅目的IP)发给RS。RS收到后,先把数据包的头解开,还原数据包,处理后,直接返回给客户端,不需要再经过负载均衡器。注意,由于RS需要对负载均衡器发过来的数据包进行还原,所以说必须支持IPTUNNEL协议，所以,在RS的内核中,必须编译支持IPTUNNEL这个选项。

优点：负载均衡器只负责将请求包分发给后端节点服务器，而RS将应答包直接发给用户

所以，减少了负载均衡器的大量数据流动，负载均衡器不再是系统的瓶颈，就能处理很巨大的请求量这种方式，一台负载均衡器能够为很多RS进行分发。而且跑在公网上就能进行不同地域的分发。

缺点：隧道模式的RS节点需要合法IP，这种方式需要所有的服务器支持”IP Tunneling”

(IP Encapsulation)协议，服务器可能只局限在部分Linux系统上

三、直接路由模式（VS-DR）

原理：负载均衡器和RS都使用同一个IP对外服务但只有DR对ARP请求进行响应所有RS对本身这个IP的ARP请求保持静默也就是说,网关会把对这个服务IP的请求全部定向给DR

而DR收到数据包后根据调度算法,找出对应的RS,把目的MAC地址改为RS的MAC（因为IP一致）并将请求分发给这台RS这时RS收到这个数据包,处理完成之后，由于IP一致，可以直接将数据返给客户则等于直接从客户端收到这个数据包无异,处理后直接返回给客户端由于负载均衡器要对二层包头进行改换,所以负载均衡器和RS之间必须在一个广播域也可以简单的理解为在同一台交换机上

优点：和TUN（隧道模式）一样，负载均衡器也只是分发请求，应答包通过单独的路由方法返回给客户端与VS-TUN相比，VS-DR这种实现方式不需要隧道结构，因此可以使用大多数操作系统做为物理服务器。

缺点：（不能说缺点，只能说是不足）要求负载均衡器的网卡必须与物理网卡在一个物理段上。

**LVS、Nginx、HAproxy有什么区别？工作中你怎么选择？**

LVS： 是基于四层的转发

HAproxy： 是基于四层和七层的转发，是专业的代理服务器

Nginx： 是WEB服务器，缓存服务器，又是反向代理服务器，可以做七层的转发

区别： LVS由于是基于四层的转发所以只能做端口的转发

而基于URL的、基于目录的这种转发LVS就做不了

工作选择：

HAproxy和Nginx由于可以做七层的转发，所以URL和目录的转发都可以做

在很大并发量的时候我们就要选择LVS，像中小型公司的话并发量没那么大

选择HAproxy或者Nginx足已，由于HAproxy由是专业的代理服务器

配置简单，所以中小型企业推荐使用HAproxy

**lvs/nginx/haproxy优缺点**

nginx分析

优点：

工作在7层，可以针对http做分流策～

正则表达式比HAProxy强大

安装、配置、测试间大，通过日志可以解决多数问题

并发量可以达到几万次

Nginx还可以作为Web服务器使用

缺点：

仅支持http、https、mail协议，应用面小

监控检查仅通过端口，无法使用url检查

LVS分析

优点

负载能力强，工作在4层，对内存、CPU消耗低

配置性低，没有大多可配置性，减少人为错误

应用面广，几乎可以为所有应用提供负载均衡

缺点

不支持正则表达式，不能实现动静分离

如果网站架构庞大，LVS-DR配置比较繁琐

HAProxy分析

优点

支持session、cookie功能

可以通过url进行健康检查

效率、负载均衡速度，高于nginx、低于LVS

HAProxy支持TCP，可以对Mysql进行负载均衡

调度算法丰富

缺点

正则弱于nginx

日志依赖于syslogd，不支持apache日志

Nginx的优点是：

1、工作在网络的7层之上，可以针对http应用做一些分流的策略，比如针对域名、目录结构它的正则规则比HAProxy更为强大和灵活，这也是它目前广泛流行的主要原因之一

Nginx单凭这点可利用的场合就远多于LVS了。

2、Nginx对网络稳定性的依赖非常小，理论上能ping通就就能进行负载功能，这个也是它的优势之一相反LVS对网络稳定性依赖比较大，这点本人深有体会；

3、Nginx安装和配置比较简单，测试起来比较方便，它基本能把错误用日志打印出来LVS的配置、测试就要花比较长的时间了，LVS对网络依赖比较大。

4、可以承担高负载压力且稳定，在硬件不差的情况下一般能支撑几万次的并发量，负载度比LVS相对小些。

5、Nginx可以通过端口检测到服务器内部的故障，比如根据服务器处理网页返回的状态码、超时等等，并且会把返回错误的请求重新提交到另一个节点，不过其中缺点就是不支持url来检测。比如用户正在上传一个文件，而处理该上传的节点刚好在上传过程中出现故障，Nginx会把上传切到另一台服务器重新处理，而LVS就直接断掉了如果是上传一个很大的文件或者很重要的文件的话，用户可能会因此而不满。

6、Nginx不仅仅是一款优秀的负载均衡器/反向代理软件，它同时也是功能强大的Web应用服务器LNMP也是近几年非常流行的web架构，在高流量的环境中稳定性也很好。

7、Nginx现在作为Web反向加速缓存越来越成熟了，速度比传统的Squid服务器更快，可考虑用其作为反向代理加速器

8、Nginx可作为中层反向代理使用，这一层面Nginx基本上无对手，唯一可以对比Nginx的就只有lighttpd了不过lighttpd目前还没有做到Nginx完全的功能，配置也不那么清晰易读，社区资料也远远没Nginx活跃

9、Nginx也可作为静态网页和图片服务器，这方面的性能也无对手。还有Nginx社区非常活跃，第三方模块也很多

Nginx的缺点是：

1、Nginx仅能支持http、https和Email协议，这样就在适用范围上面小些，这个是它的缺点

2、对后端服务器的健康检查，只支持通过端口来检测，不支持通过url来检测 不支持Session的直接保持，但能通过ip\_hash来解决

LVS：使用Linux内核集群实现一个高性能、高可用的负载均衡服务器它具有很好的可伸缩性（Scalability)、可靠性（Reliability)和可管理性（Manageability)

LVS的优点是：

1、抗负载能力强、是工作在网络4层之上仅作分发之用，没有流量的产生这个特点也决定了它在负载均衡软件里的性能最强的，对内存和cpu资源消耗比较低

2、配置性比较低，这是一个缺点也是一个优点，因为没有可太多配置的东西所以并不需要太多接触，大大减少了人为出错的几率

3、工作稳定，因为其本身抗负载能力很强，自身有完整的双机热备方案如LVS+Keepalived，不过我们在项目实施中用得最多的还是LVS/DR+Keepalived

4、无流量，LVS只分发请求，而流量并不从它本身出去，这点保证了均衡器IO的性能不会收到大流量的影响。

5、应用范围较广，因为LVS工作在4层，所以它几乎可对所有应用做负载均衡，包括http、数据库、在线聊天室等

LVS的缺点是：

1、软件本身不支持正则表达式处理，不能做动静分离而现在许多网站在这方面都有较强的需求，这个是Nginx/HAProxy+Keepalived的优势所在

2、如果是网站应用比较庞大的话，LVS/DR+Keepalived实施起来就比较复杂了特别后面有Windows Server的机器的话，如果实施及配置还有维护过程就比较复杂了相对而言，Nginx/HAProxy+Keepalived就简单多了。

HAProxy的特点是：

1、HAProxy也是支持虚拟主机的。

2、HAProxy的优点能够补充Nginx的一些缺点，比如支持Session的保持，Cookie的引导同时支持通过获取指定的url来检测后端服务器的状态

3、HAProxy跟LVS类似，本身就只是一款负载均衡软件单纯从效率上来讲HAProxy会比Nginx有更出色的负载均衡速度，在并发处理上也是优于Nginx的

4、HAProxy支持TCP协议的负载均衡转发，可以对MySQL读进行负载均衡对后端的MySQL节点进行检测和负载均衡，大家可以用LVS+Keepalived对MySQL主从做负载均衡

5、HAProxy负载均衡策略非常多，HAProxy的负载均衡算法现在具体有如下8种：

①roundrobin，表示简单的轮询，这个不多说，这个是负载均衡基本都具备的；

② static-rr，表示根据权重，建议关注；

③leastconn，表示最少连接者先处理，建议关注；

④ source，表示根据请求源IP，这个跟Nginx的IP\_hash机制类似我们用其作为解决session问题的一种方法，建议关注；

⑤ri，表示根据请求的URI；

⑥rl\_param，表示根据请求的URl参数’balance url\_param’ requires an URL parameter name；

⑦hdr(name)，表示根据HTTP请求头来锁定每一次HTTP请求；

⑧rdp-cookie(name)，表示根据据cookie(name)来锁定并哈希每一次TCP请求。