


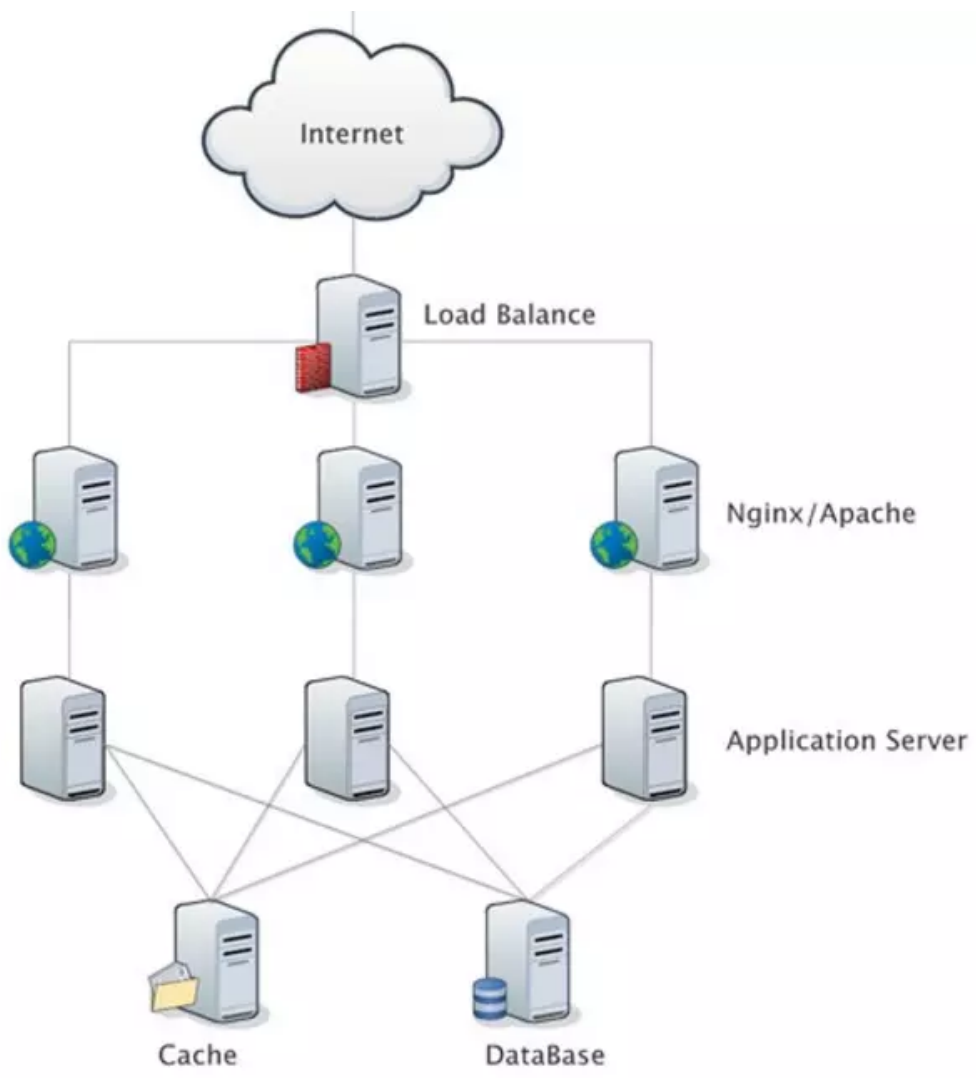
LVS、Nginx、HAProxy、keepalive 的工作原理

 jiangmo (/u/de31051e96e1) [+ 关注](#)

2017.09.22 10:40* 字数 5574 阅读 2359 评论 1 喜欢 17

(/u/de31051e96e1)

当前大多数的互联网系统都使用了服务器集群技术，集群是将相同服务部署在多台服务器上构成一个集群整体对外提供服务，这些集群可以是 Web 应用服务器集群，也可以是数据库服务器集群，还可以是分布式缓存服务器集群等等。



在实际应用中，在 Web 服务器集群之前总会有一台负载均衡服务器，负载均衡设备的任务就是作为 Web 服务器流量的入口，挑选最合适的一台 Web 服务器，将客户端的请求转发给它处理，实现客户端到真实服务端的透明转发。

- LVS、Nginx、HAProxy 是目前使用最广泛的三种软件负载均衡软件。

一般对负载均衡的使用是随着网站规模的提升根据不同的阶段来使用不同的技术。具体的应用需求还得具体分析，如果是中小型的 Web 应用，比如日 PV 小于1000万，用 Nginx 就完全可以了；如果机器不少，可以用 DNS 轮询，LVS 所耗费的机器还是比较多的；大型网站或重要的服务，且服务器比较多时，可以考虑用 LVS。

(/apps/
utm_sc
banner

目前关于网站架构一般比较合理流行的架构方案：

- Web 前端采用 Nginx/HAProxy+Keepalived 作负载均衡器；
- 后端采用 MySQL 数据库一主多从和读写分离，采用 LVS+Keepalived 的架构。
(MySQL自带主从设置，如何理解用LVS？)

LVS

LVS 是 Linux Virtual Server 的简称，也就是 Linux 虚拟服务器。现在 LVS 已经是 Linux 标准内核的一部分，从 Linux2.4 内核以后，已经完全内置了 LVS 的各个功能模块，无需给内核打任何补丁，可以直接使用 LVS 提供的各种功能。

LVS 的体系结构

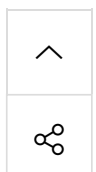
(https:/
click.ye
slot=30
389e-4
40b69e

LVS 架设的服务器集群系统有三个部分组成：

- 最前端的负载均衡层，用 Load Balancer 表示
- 中间的服务器集群层，用 Server Array 表示
- 最底端的数据共享存储层，用 Shared Storage 表示

LVS 负载均衡机制

LVS 不像 HAProxy 等七层软负载面向的是 HTTP 包，所以七层负载可以做的 URL 解析等工作，LVS 无法完成。



LVS 是四层负载均衡，也就是说建立在 OSI 模型的第四层——传输层之上，传输层上有我们熟悉的 TCP/UDP，LVS 支持 TCP/UDP 的负载均衡。因为 LVS 是四层负载均衡，因此它相对于其它高层负载均衡的解决办法，比如 DNS 域名轮流解析、应用层负载的调度、客户端的调度等，它的效率是非常高的。

所谓四层负载均衡，也就是主要通过报文中的目标地址和端口。七层负载均衡，也称为“内容交换”，也就是主要通过报文中的真正有意义的应用层内容。

LVS 的转发主要通过修改 IP 地址（NAT 模式，分为源地址修改 SNAT 和目标地址修改 DNAT）、修改目标 MAC（DR 模式）来实现。

LVS负载均衡模式

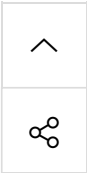
NAT 模式：网络地址转换

NAT（Network Address Translation）是一种外网和内网地址映射的技术。
NAT 模式下，网络数据报的进出都要经过 LVS 的处理。LVS 需要作为 RS（真实服务器）的网关。

当包到达 LVS 时，LVS 做目标地址转换（DNAT），将目标 IP 改为 RS 的 IP。RS 接收到包以后，仿佛是客户端直接发给它的一样。RS 处理完，返回响应时，源 IP 是 RS IP，目标 IP 是客户端的 IP。这时 RS 的包通过网关（LVS）中转，LVS 会做源地址转换（SNAT），将包的源地址改为 VIP，这样，这个包对客户端看起来就仿佛是 LVS 直接返回给它的。

(/apps/
utm_sc
banner

(https:/
click.y
slot=30
389e-4
40b69e



(/apps/
utm_sc
banner

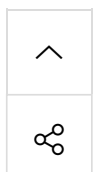
DR 模式：直接路由

DR 模式下需要 LVS 和 RS 集群绑定同一个 VIP（RS 通过将 VIP 绑定在 loopback 实现），但与 NAT 的不同点在于：请求由 LVS 接受，由真实提供服务的服务器（RealServer，RS）直接返回给用户，返回的时候不经过 LVS。

详细来看，一个请求过来时，LVS 只需要将网络帧的 MAC 地址修改为某一台 RS 的 MAC，该包就会被转发到相应的 RS 处理，注意此时的源 IP 和目标 IP 都没变，LVS 只是做了一下移花接木。RS 收到 LVS 转发来的包时，链路层发现 MAC 是自己的，到上面的网络层，发现 IP 也是自己的，于是这个包被合法地接受，RS 感知不到前面有 LVS 的存在。而当 RS 返回响应时，只要直接向源 IP（即用户的 IP）返回即可，不再经过 LVS。

(https://
click.y
slot=30
389e-4
40b69e

DR 负载均衡模式数据分发过程中不修改 IP 地址，只修改 mac 地址，由于实际处理请求的真实物理 IP 地址和数据请求目的 IP 地址一致，所以不需要通过负载均衡服务器进行地址转换，可将响应数据包直接返回给用户浏览器，避免负载均衡服务器网卡带宽成为



瓶颈。

IP隧道模式

隧道模式则类似于VPN的方式,使用网络分层的原理,在从客户端发来的数据包的基础上,封装一个新的IP头标记(不完整的IP头,只有目的IP部)发给RS,RS收到后,先把DR发过来的数据包的头给解开,还原其数据包原样,处理后,直接返回给客户端,而不需要再经过DR。需要注意的是,由于REALSERVER需要对DR发过来的数据包进行还原,也就是说必须支持IPTUNNEL协议。所以,在REALSERVER的内核中,必须编译支持IPTUNNEL这个选项。

因此，DR 模式具有较好的性能，也是目前大型网站使用最广泛的一种负载均衡手段。

LVS已实现了以下八种调度算法:

- LVS负载均衡算法---1.轮询调度(Round-RobinScheduling)

调度器通过"轮询"调度算法将外部请求按顺序轮流分配到集群中的真实服务器上,它均等地对待每一台服务器,而不管服务器上实际的连接数和系统负载。

- LVS负载均衡算法---2.加权轮询调度(WeightedRound-RobinScheduling)

调度器通过"加权轮询"调度算法根据真实服务器的不同处理能力来调度访问请求。这样可以保证处理能力强的服务器处理更多的访问流量。调度器可以自动问询真实服务器的负载情况,并动态地调整其权值。

- LVS负载均衡算法---3.最小连接调度(Least-ConnectionScheduling)

调度器通过"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。

- LVS负载均衡算法---4.加权最小连接调度(WeightedLeast-ConnectionScheduling)

在集群系统中的服务器性能差异较大的情况下,调度器采用"加权最少链接"调度算法优化负载均衡性能,具有较高权值的服务器将承受较大比例的活动连接负载。调度器可以自动问询真实服务器的负载情况,并动态地调整其权值

- LVS负载均衡算法---5.基于局部性的最少链接(Locality-BasedLeastConnectionsScheduling)

基于局部性的最少链接"调度算法是针对目标IP地址的负载均衡,目前主要用于Cache集群系统。该算法根据请求的目标IP地址找出该目标IP地址最近使用的服务器,若该服务器是可用的且没有超载,将请求发送到该服务器;若服务器不存在,或者该服务器超载且有服务器处于一半的工作负载,则用"最少链接"的原则选出一个可用的服务器,将请求发送到该服务器。

- LVS负载均衡算法---6.带复制的基于局部性最少链接(Locality-BasedLeastConnectionswithReplicationScheduling)

(/apps/
utm_sc
banner

(https:/
click.y
slot=30
389e-4
40b69e



带复制的基于局部性最少链接"调度算法也是针对目标IP地址的负载均衡,目前主要用于Cache集群系统。它与LBLC算法的不同之处是它要维护从一个目标IP地址到一组服务器的映射,而LBLC算法维护从一个目标IP地址到一台服务器的映射。该算法根据请求的目标IP地址找出该目标IP地址对应的服务器组,按"最小连接"原则从服务器组中选出一台服务器,若服务器没有超载,将请求发送到该服务器,若服务器超载;则按"最小连接"原则从这个集群中选出一台服务器,将该服务器加入到服务器组中,将请求发送到该服务器。同时,当该服务器组有一段时间没有被修改,将最忙的服务器从服务器组中删除,以降低复制的程度

LVS负载均衡算法---7.目标地址散列调度(DestinationHashingScheduling)

目标地址散列"调度算法根据请求的目标IP地址,作为散列键(HashKey)从静态分配的散列表找出对应的服务器,若该服务器是可用的且未超载,将请求发送到该服务器,否则返回空

LVS负载均衡算法---8.源地址散列调度(SourceHashingScheduling)

源地址散列"调度算法根据请求的源IP地址,作为散列键(HashKey)从静态分配的散列表找出对应的服务器,若该服务器是可用的且未超载,将请求发送到该服务器,否则返回空。

LVS 的优点

抗负载能力强、是工作在传输层上仅作分发之用，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。
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。
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。
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 的性能不会受到大流量的影响。
应用范围比较广，因为 LVS 工作在传输层，所以它几乎可以对所有应用做负载均衡，包括 http、数据库、在线聊天室等等。

LVS 的缺点

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。
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，相对而言，Nginx/HAProxy+Keepalived就简单多了

通过ipvsadm 或者 keepAlive进行配置管理

Nginx

Nginx 是一个强大的 Web 服务器软件，用于处理高并发的 HTTP 请求和作为反向代理服务做负载均衡。具有高性能、轻量级、内存消耗少，强大的负载均衡能力等优势。

Nignx 的架构设计

相对于传统基于进程或线程的模型（Apache就采用这种模型）在处理并发连接时会为每一个连接建立一个单独的进程或线程，且在网络或者输入/输出操作时阻塞。这将导致内存和 CPU 的大量消耗，因为新起一个单独的进程或线程需要准备新的运行时环境，包括堆和栈内存的分配，以及新的执行上下文，当然，这些也会导致多余的 CPU 开销。最终，会由于过多的上下文切换而导致服务器性能变差。

(/apps/
utm_sc
banner

反过来，Nginx 的架构设计是采用模块化的、基于事件驱动、异步、单线程且非阻塞。

Nginx 大量使用多路复用和事件通知，Nginx 启动以后，会在系统中以 daemon 的方式在后台运行，其中包括一个 master 进程， $n(n \geq 1)$ 个 worker 进程。所有的进程都是单线程（即只有一个主线程）的，且进程间通信主要使用共享内存的方式。

其中，master 进程用于接收来自外界的信号，并给 worker 进程发送信号，同时监控 worker 进程的工作状态。worker 进程则是外部请求真正的处理者，每个 worker 请求相互独立且平等的竞争来自客户端的请求。请求只能在一个 worker 进程中被处理，且一个 worker 进程只有一个主线程，所以同时只能处理一个请求。（原理同 Netty 很像）

(https://
click.y
slot=3C
389e-4
40b69e

Nginx 负载均衡

Nginx 负载均衡主要是对七层网络通信模型中的第七层应用层上的 http、https 进行支持。

Nginx 是以反向代理的方式进行负载均衡的。反向代理（Reverse Proxy）方式是指以代理服务器来接受 Internet 上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给 Internet 上请求连接的客户端，此时代理服务器对外就表现为一个服务器。

Nginx 实现负载均衡的分配策略有很多，Nginx 的 upstream 目前支持以下几种方式：



- 轮询（默认）：每个请求按时间顺序逐一分配到不同的后端服务器，如果后端服务器 down 掉，能自动剔除。
- weight：指定轮询几率，weight 和访问比率成正比，用于后端服务器性能不均的情况。
- ip_hash：每个请求按访问 ip 的 hash 结果分配，这样每个访客固定访问一个后端服务器，可以解决 session 的问题。
- fair（第三方）：按后端服务器的响应时间来分配请求，响应时间短的优先分配。
- url_hash（第三方）：按访问 url 的 hash 结果来分配请求，使每个 url 定向到同一个后端服务器，后端服务器为缓存时比较有效。

(/apps/
utm_sc
banner

Nginx 的优点

跨平台：Nginx 可以在大多数 Unix like OS 编译运行，而且也有 Windows 的移植版本

配置异常简单：非常容易上手。配置风格跟程序开发一样，神一般的配置

非阻塞、高并发连接：官方测试能够支撑5万并发连接，在实际生产环境中跑到2~3万并发连接数

事件驱动：通信机制采用 epoll 模型，支持更大的并发连接

Master/Worker 结构：一个 master 进程，生成一个或多个 worker 进程

内存消耗小：处理大并发的请求内存消耗非常小。在3万并发连接下，开启的10个 Nginx 进程才消耗150M 内存（15M*10=150M）

内置的健康检查功能：如果 Nginx 代理的后端的某台 Web 服务器宕机了，不会影响前端访问

节省带宽：支持 GZIP 压缩，可以添加浏览器本地缓存的 Header 头

稳定性高：用于反向代理，宕机的概率微乎其微

(https://
click.y
slot=3C
389e-4
40b69e

Nginx 的缺点

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

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

HAProxy

HAProxy 支持两种代理模式 TCP（四层）和 HTTP（七层），也是支持虚拟主机的。

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。

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。

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。



HAProxy 负载均衡策略非常多：Round-robin（轮循）、Weight-round-robin（带权轮循）、source（原地址保持）、RI（请求URL）、rdp-cookie（根据cookie）。

什么是keepalived

- keepalived是保证集群高可用的一个服务软件，其功能类似于heartbeat
(<https://link.jianshu.com?t=https://github.com/chenzhiwei/linux/tree/master/heartbeat>)，用来防止单点故障。
- 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 (<https://link.jianshu.com?t=http://en.wikipedia.org/wiki/VRRP>)。
- keepalived是可以工作在第三层、第四层、第五层的检测服务器状态的软件，
- 如果有一台web服务器死机，或工作出现故障，keepalived将检测到，并将其从系统中剔除
当web服务器工作正常后keepalived自动将web服务器加入到服务器集群中

Keepalived的工作原理

- 三层、四层、五层工作在TCP/IP协议栈的IP层、TCP层、应用层。原理如下：
- 三层：keepalived使用三层方式工作是，keepalived会定期向服务器集群中的服务器发送一个ICMP的数据包，也就是ping程序，如果发现某台服务器的IP地址没有激活，keepalived便报告这台服务器失效，并将它从集群中删除，这种情况的典型例子是某台服务器被非法关机。三层的方式是以服务器的IP地址是否有效作为服务器工作正常与否的标准。
- 四层：主要是以TCP端口的状态来决定服务器工作正常与否。如web服务器的端口一般是80，如果keepalived检测到80端口没有启动，则keepalived将这台服务器从集群中剔除。
- 五层：应用层，比三层和四层要复杂一点，keepalived将根据用户的设定检查服务器程序运行是否正常，如果与用户设定的不相符，则keepalived将把服务器从服务器集群中剔除。
- 基于VRRP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 (<https://link.jianshu.com?t=http://tools.ietf.org/html/rfc5798#section-5.1>)来选举一个backup当master (https://link.jianshu.com?t=http://en.wikipedia.org/wiki/Virtual_Router_Redundancy_Protocol#Elections_of_master_routers)。这样的话就可以保证路由器的高可用了。

(/apps/
utm_sc
banner

(https:/
click.y
slot=30
389e-4
40b69e



- 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。

keepalived的作用

高可用-可持续的服务器质量

负载均衡-横向扩展

实现对失效服务器的隔离-通过健康监测，保证服务的可用性

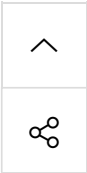
实现：vrrp协议实现。（冗余网关路由协议）

keepalived体系结构

(https://
click.ye
slot=30
389e-4
40b69e

- 1、 watchdog 负责监控checkers和vrrp进程的状况。
- 2、 Checkers 负责真实服务器的健康监测，是keepalived最主要的功能，换一句话说，可以没有vrrp statck，但是健康检查healthchecking一定要有。
- 3、 Vrrp statck 负责负载均衡器之间失败切换failover。如果只用一个负载均衡器，则vrrp不是必须的。
- 4、 Ipv6 warpper是用来发送设定的规则封装到内核ipvs代码。
- 5、 Netlink reflector 用来设定vrrp的vip地址等。

Keepalived功能十分强大，但是配置工作十分简单，keepalived各种功能的实现是通过设定配置文件keepalived.conf来完成的。



Ref :

<http://www.talkwithtrend.com/Article/216127> (<https://link.jianshu.com?>

t=<http://www.talkwithtrend.com/Article/216127>)

<http://outofmemory.cn/wiki/keepalived-configuration> (<https://link.jianshu.com?>

t=<http://outofmemory.cn/wiki/keepalived-configuration>) **【keepalived】**

```
(/apps/  
utm_sc  
banner
```

武训三十年

赞赏支持

 负载均衡 (/nb/16237416)

举报文章 © 著作权归作者所有



jiangmo (/u/de31051e96e1)

写了 396654 字，被 228 人关注，获得了 720 个喜欢

(/u/de31051e96e1)

+ 关注

原创文章原则上使用「知识共享署名-非商业性使用4.0国际许可协议」进行许可。你可以分享、修改文章内...

喜欢 | 17



更多分享

(<https://click.yourchoice.com/click?slot=30389e-40b69e>)



下载简书 App ▶

随时随地发现和创作内容



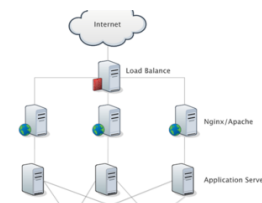
(/apps/redirect?utm_source=note-bottom-click)

被以下专题收入，发现更多相似内容



云服务 ([/c/ad840ed2feac?utm_source=desktop&utm_medium=notes-ded-collection](https://c/ad840ed2feac?utm_source=desktop&utm_medium=notes-ded-collection))


(/p/9a6eaded483c?

 \wedge 

utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend

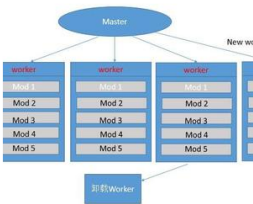
你真的掌握 LVS、Nginx 及 HAProxy 的工作原理吗 (/p/9a6eaded483c?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)

当前大多数的互联网系统都使用了服务器集群技术，集群是将相同服务部署在多台服务器上构成一个集群整体对外提供服务，这些集群可以是 Web 应用服务器集群，也可以是数据库服务器集群，还可以是分布式缓存...

 LinkedKeeper (/u/d43878459bb8?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)

(/apps/utms/banner)

(/p/8f7242cbf469?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)



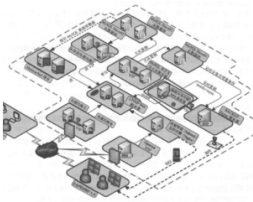
utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations
负载均衡架构 (/p/8f7242cbf469?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations)

【摘要】 面对大量用户访问、高并发请求，海量数据，可以使用高性能的服务器、大型数据库，存储设备，高性能Web服务器，采用高效率的编程语言比如(Go,Scala)等，当单机容量达到极限时，我们需要考虑业...

 静修佛缘 (/u/d6e5ad19fce?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendations

(/p/20ce413a898a?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)



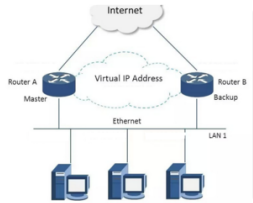
utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations
Nginx反向代理与负载均衡 (/p/20ce413a898a?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations)

《老男孩Linux运维》Nginx Documentation 集群简介 集群就是指一组（若干）相互独立的计算机，利用高速通信网络组成的一个较大的计算机服务系统，每个集群结点都是运行各自服务的独立服务器。这些服务器...

 Zhang21 (/u/f13278e94ecb?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendations

(/p/e84a3ea2e79f?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations
Nginx+keepalived双机热备（主从模式） (/p/e84a3ea2e79f?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations)


负载均衡技术对于一个网站尤其是大型网站的web服务器集群来说是至关重要的！做好负载均衡架构，可以实现故障转移和高可用环境，避免单点故障，保证网站健康持续运行。关于负载均衡介绍，可以参考：linu...

 青川刺客 (/u/d61eae619267?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendations

四层、七层负载均衡的区别 (/p/fa937b8e6712?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendations&utm_term=network&utm_term=load&utm_term=balance)

** 内容安排： ** 简介 区别 Nginx、LVS及HAProxy负载均衡软件的优缺点 一、简介 ** 所谓四层就是基于IP+端口的负载均衡；七层就是基于URL等应用层信息的负载均衡；**同理，还有基于MAC地址的二层负载...

薛晨 (/u/a2d715858c25?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


压 (/p/ac83cc8581d6?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

人都说30而立，今年28，却提前进入压力山大之时～ -----宇轩君----- 前日回京，友送，一路争辩宝宝是在身边养(自己辛苦些) 还是送回家由老人带，是给小朋友买学区房，还是随之任流听之任...

蜀锦景泰蓝 (/u/eef338669976?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

《我的团长我的团》 --一群炮灰的故事 (/p/1f83da7387fa?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

什么是炮灰？炮灰就是去送死。什么是炮灰？炮灰就是你死都不知道你会死在哪里。什么是炮灰？炮灰就是你死了以后历史书上的冰冷的数字。而这讲的就是这样一群炮灰的故事。也是一个个绝望的故事。“绝望”...

叶落织秋 (/u/3eb22b387d84?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/4af8ea6b09a3?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)




ios事件传递 (/p/4af8ea6b09a3?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

本篇主要讲解iOS事件传递的整个过程，大部分内容翻译自Apple Developer Guide，原文链接 当一个用户事件产生的时候，UIKit 会创建一个事件对象来描述这个用户事件。然后它会将该事件对象放进UIApplication...

Little_Mango (/u/c2d1b6abd479?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

鸡汤好过一事无成 (/p/b189fa13e7f8?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

奋斗是为了让自己成长， 让自己变强大， 而不是为了去和他人做比较； 奋斗的路途，没有终点。当恐惧袭来的时候，我要闭上双眼，用自己的内心做决定。在这条漫长的奋斗之路上，我希望自己日后做的每一个...

莫浮 (/u/fdc3079d23c9?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

2017-08-11 (/p/6961cb2574b5?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

什么是亲情

其实很寂寞TI (/u/25373cd7970e?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

