**第十四章 使用Varnish为网站加速**

**本节所讲内容：**

**14.1 varnish概述**

**14.2 安装varnish**

**14.3 varnish 缓存原理**

**14.4 实战：缓存一个网站**

**14.5 实战：使用varnish缓存多个网站**

**14.1 varnish概述**

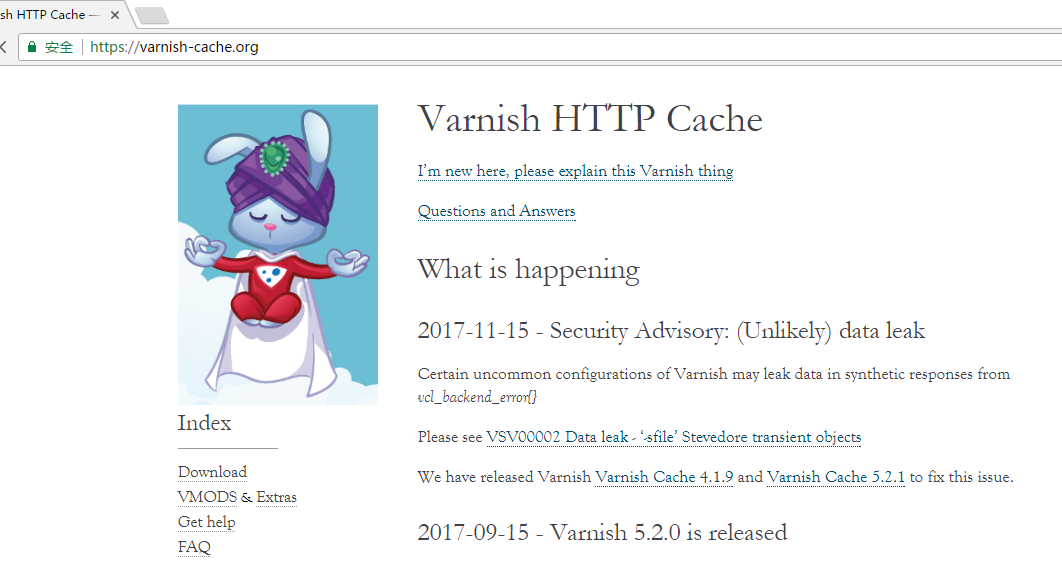
**Varnish是一款高性能的开源HTTP加速器，挪威最大的在线报纸 Verdens Gang (http://www.vg.no) 使用3台Varnish代替了原来的12台squid，性能居然比以前更好。**

**Varnish 的作者Poul-Henning Kamp是FreeBSD的内核开发者之一，他认为现在的计算机比起1975年已经复杂许多。在1975年时，储存媒介只有两种：内存与硬盘。但现在计算 机系统的内存除了主存外，还包括了cpu内的L1、L2，甚至有L3快取。硬盘上也有自己的快取装置，因此squid cache自行处理物件替换的架构不可能得知这些情况而做到最佳化，但操作系统可以得知这些情况，所以这部份的工作应该交给操作系统处理，这就是 Varnish cache设计架构。**

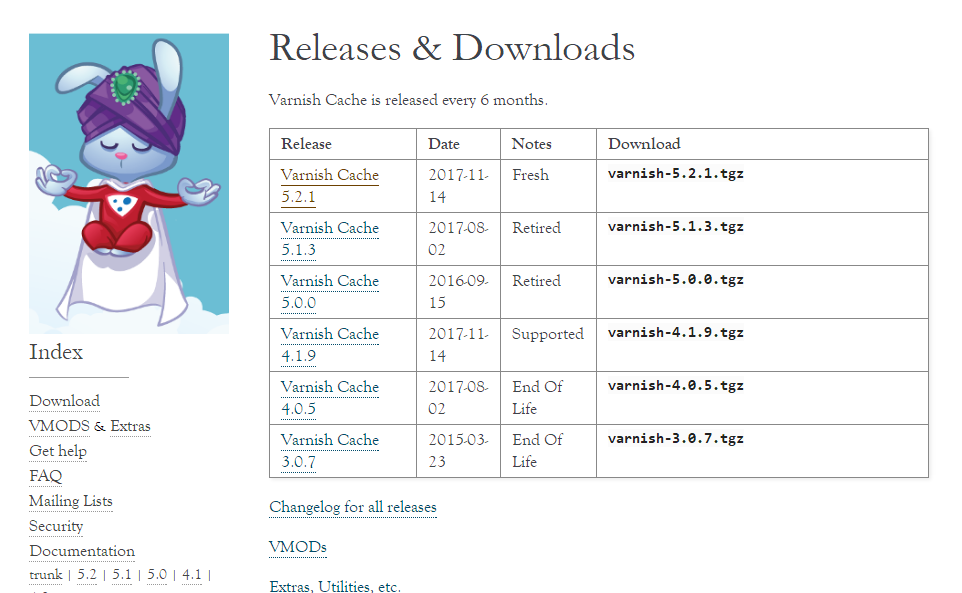
**监听端口号： 6081**

**官方网站：**

**https://www.varnish-cache.org/**



**下载：**

**<http://varnish-cache.org/releases/index.html>**[](http://varnish-cache.org/releases/index.html)

**14.1.1 varnish基础概念详解**

**1、比起squid更加轻量级，大致有以下几个特点：**

**2、可以基于内存缓存，也可以在磁盘上缓存**

**3、如果期望内存大小超过几十个G，比如图片服务器，纯粹使用内存，性能未必好，这时候可以使用磁盘进行缓存，或使用SSD X 2 做RAID1避免磁盘损坏，在实现随机访问上 ssd硬盘要比机械硬盘要好的多，如果必须要缓存在磁盘上还是建议使用ssd磁盘**

**4、可以利用虚拟内存方式，IO性能会非常好**

**5、支持设置0-60秒 精确缓存时间**

**14.1.2 支持VCL**

**其配置是通过vcl编程语言来完成的。   其配置需要先转换成C代码，所以使用vcl所写的配置，要先转换成C语言代码，因此要依赖于GCC 临时的编译vcl配置的，编译完之后才能运行起来**

**注: Varnish Configuration Language - VCL（varnish配置语言-VCL）。**

**Varnish使用区域配置语言，这种语言叫做“VCL”（varnish configuration language），在执行vcl时，varnish就把VCL转换成二进制代码。**

**VCL文件被分为多个子程序，不同的子程序在不同的时间里执行，比如一个子程序在接到请求时执行，另一个子程序在接收到后端服务器传送的文件时执行。**

**14.1.3 独特的日志存储及管理机制**

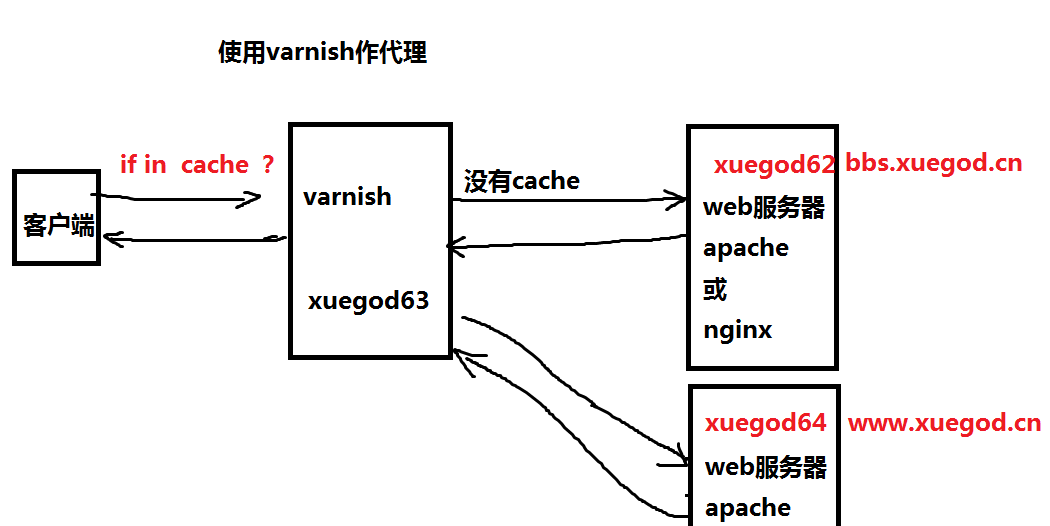
**日志既然保存在内存中，日志可以供多个应用程序所访问，所以一般查看命中率，当前请求有多少get post 方法等等，都需使用专用的工具才可以查看，比如varnishshtopvarnishlog 等命令工具用来查看日志信息**

**14.1.4 支持使用varnish状态引擎**

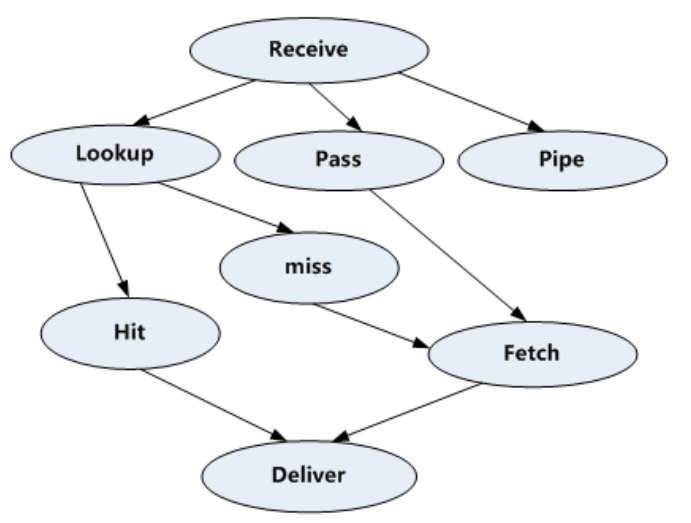
**通过巧妙的状态引擎的设计完成不同的引擎对用户的请求和缓存代理机制进行处理，用配置文件为状态引擎提供状态法则，完成缓存处理、完成代理处理等等**

**14.2 varnish 缓存原理**

**14.2.1 varnish缓存数据机制：**

****

**14.2.2 VCL处理流程图**

****

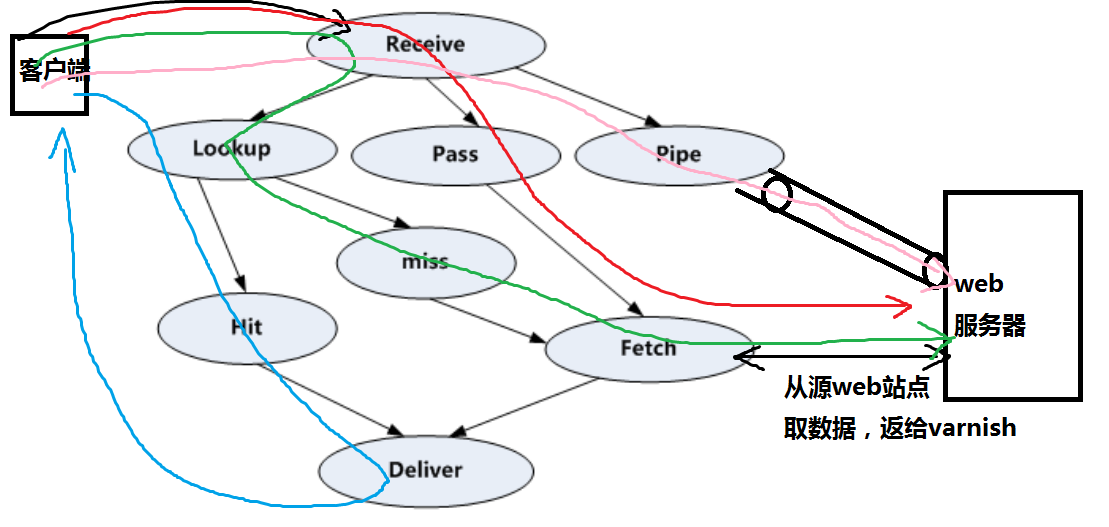
**处理过程大致分为如下几个步骤：**

1. **Receive状态，也就是请求处理的入口状态，根据VCL规则判断该请求应该是Pass或Pipe，或者进入Lookup（本地查询）。**
2. **Lookup状态，在缓存中查找用户请求的对象，如果缓存中没有其请求的对象，后续操作很可能会将其请求的对象进行缓存；进入此状态后，会在hash表中查找数据，若找到，则进入Hit（命令中）状态，否则进入miss（没命中）状态。**
3. **Pass状态，在此状态下，会进入后端请求，即进入fetch取状态。**
4. **Fetch状态，在Fetch取状态下，对请求，进行后端的获取，发送请求，获得数据，并进行本地的存储。**
5. **Deliver  提供状态， 将获取到的数据发送给客户端，然后完成本次请求。**

**注：**

**pass：绕过缓存，即不从缓存中查询内容或不将内容存储至缓存中；  
pipe：不对客户端进行检查或做出任何操作，而是在客户端与后端服务器之间建立专用“管道”，并直接将数据在二者之间进行传送；此时，keep-alive连接中后续传送的数据也都将通过此管道进行直接传送，并不会出现在任何日志中；**

**总结： 用户通过varnish加速时，有4线路可以获得数据。**

****

**14.3 安装varnish**

**1、安装varnish， 将上传varnish软件包**

**[root@xuegod63 ~]# rpm -ivh varnish-libs-4.0.5-1.el7.x86\_64.rpm**

**[root@xuegod63 ~]# rpm -ivh jemalloc-3.6.0-1.el7.x86\_64.rpm**

**[root@xuegod63 ~]# rpm -ivh varnish-3.0.6-1.el6.x86\_64.rpm**

**14.3.1 vcl置文件：**

**[root@xuegod63 ~]# ls /etc/varnish/default.vcl**

**varnish主配置文件：**

**[root@xuegod63 ~]# vim /etc/varnish/varnish.params**

**启varnish服务：**

**[root@xuegod63 ~]# systemctl start varnish**

**14.4 实战：缓存一个网站**

**实战：配置xuegod63成为 varnish服务器加速xuegod64 web服务器**

**通过CDN加速WEB网站**

**网宿，蓝讯**

**智能DNS**

**14.4.1 实战：配置xuegod63成为 varnish服务器**

**[root@xuegod63 ~]# vim /etc/varnish/default.vcl**

**###配置一个后端服务器**

**改：**

**16 backend default {**

**17 .host = "127.0.0.1";**

**18 .port = "80";**

**19 }**

**为：**

**backend web1 {**

**.host = "192.168.1.64";**

**.port = "80";**

**}**

**###查看缓存命中情况**

**在：**

**90 # sub vcl\_deliver {**

**91 # }**

**追加：**

**sub vcl\_deliver {**

**if (obj.hits > 0) {**

**set resp.http.X-Cache = "HIT from " + req.http.host;**

**set resp.http.X-Cache-Hits = obj.hits;**

**} else {**

**set resp.http.X-Cache = "MISS from " + req.http.host;**

**}**

**return (deliver);**

**}**

**保存退出。**

**14.4.2 配置varnish服务端口**

**[root@xuegod63 ~]# vim /etc/varnish/varnish.params**

**改：**

**66 VARNISH\_LISTEN\_PORT=6081**

**为：**

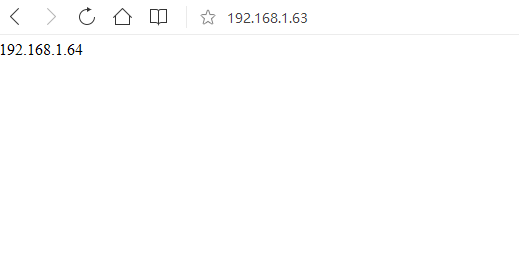
**VARNISH\_LISTEN\_PORT=80**

**测试源站点：**

**在浏览器中访问：http://192.168.1.64/**

**测试加速：**

**在浏览器中访问：http://192.168.1.63/**



**能够正常访问，说明代理设置成功。**

**14.4.5 测试缓存命中：**

**扩展：curl命令**

**[curl](http://www.aiezu.com/tag/curl" \o "查看 curl 中的全部文章" \t "_blank)是通过url语法在命令行下上传或下载文件的工具软件，它支持http,https,ftp,ftps,telnet等多种协议，常被用来抓取网页和监控Web服务器状态。**

**参数：**

**-I只取http响应头的信息，不取网页内容**

**例：**

**[root@xuegod64 src]# curl -I 192.168.1.64 #在xuegod63上查看xuegod64开启动的web服务器类型**

**HTTP/1.1 200 OK**

**Date: Tue, 23 Jan 2018 07:55:44 GMT**

**Server: Apache/2.4.6 (CentOS)**

**Last-Modified: Tue, 23 Jan 2018 07:54:01 GMT**

**ETag: "d-5636cd999c5e4"**

**Accept-Ranges: bytes**

**Content-Length: 13**

**Content-Type: text/html; charset=UTF-8**

**14.4.6 实例2：测试缓存命中**

**[root@xuegod63 ~]# curl -I 192.168.1.63**

**HTTP/1.1 200 OK**

**Date: Wed, 07 Feb 2018 14:00:49 GMT**

**Server: Apache/2.4.6 (CentOS)**

**Last-Modified: Wed, 07 Feb 2018 13:55:00 GMT**

**ETag: "d-5649fa431c0cb"**

**Content-Length: 13**

**Content-Type: text/html; charset=UTF-8**

**X-Varnish: 9**

**Age: 0**

**Via: 1.1 varnish-v4**

**X-Cache: MISS from 192.168.1.63 //表示没有命中**

**Connection: keep-alive**

**[root@xuegod63 ~]# curl -I 192.168.1.63**

**HTTP/1.1 200 OK**

**Date: Tue, 23 Jan 2018 07:54:50 GMT**

**Server: Apache/2.4.6 (CentOS)**

**Last-Modified: Tue, 23 Jan 2018 07:54:01 GMT**

**ETag: "d-5636cd999c5e4"**

**Content-Length: 13**

**Content-Type: text/html; charset=UTF-8**

**X-Varnish: 11 6**

**Age: 125**

**Via: 1.1 varnish-v4**

**X-Cache: HIT from 192.168.1.63 //表示命中**

**X-Cache-Hits: 11**

**Connection: keep-alive**

**14.5 实战：使用varnish缓存多个网站**

**14.5.1 实战：使用varnish加速多个不同域名站点的web服务器**

**配置xuegod64和xuegod62为web服务器**

**xuegod64之前已经配置成web服务器**

**14.5.2 配置xuegod62为web2服务器**

**[root@xuegod62 ~]# yum install httpd -y**

**[root@xuegod62 ~]# echo 192.168.1.62 > /var/www/html/index.html**

**[root@xuegod62 ~]# vim /etc/hosts**

**127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4**

**::1 localhost localhost.localdomain localhost6 localhost6.localdomain6**

**192.168.1.62 xuegod62.cn**

**[root@xuegod62 ~]# systemctl start httpd**

**14.5.3 配置xuegod63 上的varnish服务器：**

**[root@xuegod63 ~]# vim /etc/varnish/default.vcl**

**改：**

**backend web1 {**

**.host = "192.168.1.64";**

**.port = "80";**

**}**

**为：**

**backend web1 {**

**.host = "192.168.1.64";**

**.port = "80";**

**}**

**backend web2 {**

**.host = "192.168.1.62";**

**.port = "80";**

**}**

**#当访问www.xuegod.cn域名时从web1上取数据，访问bbs.xuegod.cn域名时到web2取数据**

**[root@xuegod63 ~]# vim /etc/varnish/default.vcl #在sub vcl\_deliver 处定义以下内容：**

**sub vcl\_recv {**

**if (req.http.host ~ "(?i)^(www.)?xuegod.cn$") {**

**set req.http.host = "www.xuegod.cn";**

**set req.backend\_hint = web1;**

**} elsif (req.http.host ~ "(?i)^bbs.xuegod.cn$") {**

**set req.backend\_hint = web2;**

**return(hash);**

**}**

**}**

**14.5.4 重新加载varnish配置文件：**

**[root@xuegod63 ~]# systemctl reload varnish**

**测试：网页访问 www.xuegod.cn  和 bbs.xuegod.cn则会显示不同的内容。**

**在xuegod62上测试：**

**[root@xuegod62 ~]# vim /etc/hosts #添加hosts文件**

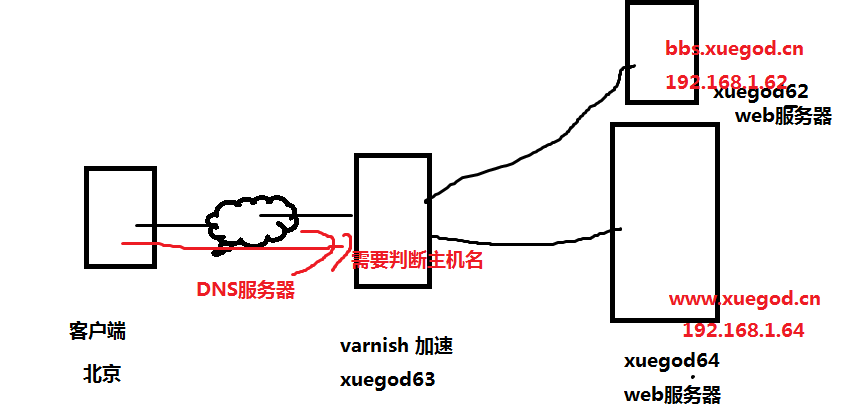
**127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4**

**::1 localhost localhost.localdomain localhost6 localhost6.localdomain6**

**192.168.1.62 xuegod62.cn**

**192.168.1.63 www.xuegod.cn**

**192.168.1.63 bbs.xuegod.cn**

****

**[root@xuegod62 ~]# yum install elinks -y**

**[root@xuegod62 ~]# elinks www.xuegod.cn --dump**

**192.168.1.64**

**[root@xuegod62 ~]# elinks bbs.xuegod.cn --dump**

**192.168.1.62**

**总结：**

**14.1 varnish概述**

**14.2 安装varnish**

**14.3 varnish 缓存原理**

**14.4 实战：缓存一个网站**

**14.5 实战：使用varnish缓存多个网站**