

用**DNSPod**和**Squid**打造自己的**CDN**

奶罩学堂系列文章

Copyright © 2007 Sam Wu, wuhongsheng.com. All rights reserved.

目录

用DNSPOD和SQUID打造自己的CDN	1
目录	2
前言	3
第 1 章 CDN和智能DNS.....	4
1.什么是CDN.....	4
2.什么是智能DNS，什么是DNSPod.....	4
第 2 章 前期准备	6
第 3 章 设置域名的DNS.....	7
1.在DNSPod上面添加域名	7
A.Windows用户.....	8
B.Linux/Unix/Mac用户.....	8
2.把域名的DNS转到DNSPod	9
第 4 章 安装CENTOS LINUX.....	10
1.基本安装	10
2.初次启动	18
3.远程连接	18
4.LINUX的基本命令	19
5.VIM的基本使用.....	19
A.打开文件.....	20
B.编辑文件.....	20
C.保存文件	21
6.设置安装更新	21
第 5 章 安装SQUID的前期准备	22
1.LINUX下面程序安装的基本知识	22
A.程序安装包的分类.....	22
B.通过源代码包安装程序.....	22
C.程序安装到哪里？	22
D.如何得到configure的参数？	23
E.如何删除安装好的程序？	23
F.还有个make clean是干什么用的？	23
2.源代码编译时的优化	23
3.编译SQUID的前期准备	24
第 6 章 编译并安装SQUID	26
第 7 章 配置SQUID	28
1.设置HOSTS表	28
2.设置SQUID	28
第 8 章 测试并运行SQUID.....	33
1.前期准备	33
2.运行SQUID.....	33
3.测试SQUID	33
4.正式运行SQUID.....	34
5.让SQUID跟随系统启动	34

前言

本篇教程是顺应大家的要求而写。教程内大部分都是奶罩在为VeryCD等大型网站构建CDN时所累积的经验。在一些概念方面可能会有一些错漏，希望大家指正。

本教程面对的对象是个人站长，所以各方面会力求傻瓜化，让大家都能看懂。当然，前提是需要你要有一颗好学的心，还有坚韧的毅力。

另外一方面，本篇文章基本不会涉及到Windows的内容，整套系统将会是架构在CentOS Linux 5上。别问我为什么不是Windows，因为Windows只能拿一个字形容：烂。不喜欢Linux，或者认为Linux学习起来有困难的朋友，可以到此打住。

可能有人要问，为什么是Linux，而不是FreeBSD？因为FreeBSD不管从系统的优化配置，还是文件系统的性能，都不合适初学者。

先前说过，本篇文章针对的是没有多少技术基础的个人站长，所以文章内会有Linux的安装、基本操作等一些东西教给你，让你学得简单，时间花得有价值。

再罗嗦一下，在接下来的时间内你将要面对一个黑漆漆的，全是英文的屏幕，请做好心理准备。

注意：本篇教程将*不会*涉及到以下内容：

- 1.动态内容（比如BBS）
- 2.大文件（比如软件下载站的软件）

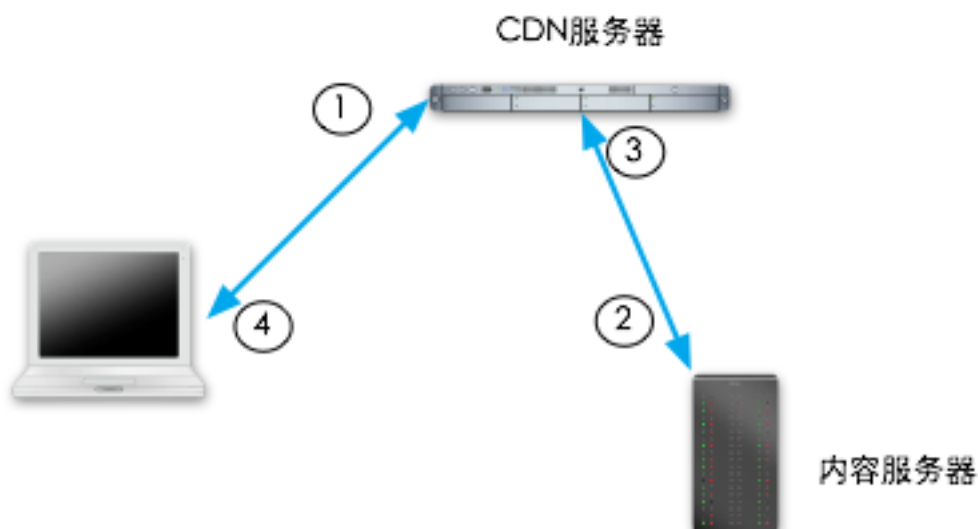
第 1 章 CDN和智能DNS

1.什么是CDN

简单来说，CDN就是能让你的网站访问起来更加快的东西。例如网通用户访问了你在网通的CDN服务器上面某个文件，CDN系统就会判断这个文件是否被请求过，如果被请求过，在没有失效的情况下，就会从系统的缓存里面读取这个文件并返回给用户。如果这个文件没有被请求过，那么CDN会自动从你的主服务器上面获取这个文件，然后缓存到CDN系统内，再给用户返回。

看到这里，聪明的朋友就会问：如果网通的CDN服务器，访问我在电信的主服务器速度很慢，那么是不是用户第一次请求这个文件的时候会很慢？嘿嘿，恭喜你，的确是这样。并且，在文件没有完全缓存到CDN上面的时候，所有访问这个文件的用户都会出现错误，比如网页打开不完整，文件下载到一半等。但只要你的网站访问量足够大，这个影响对用户来说是微不足道的。

然后还有朋友会问，为什么商业的CDN不存在这个问题？因为，商业的CDN有自己租用的线路，自己走路由。这就是商业CDN卖得如此之贵的原因。当然，如何走这个路由，搭建这么一个大型的CDN系统，不在本文的讨论范围以内。

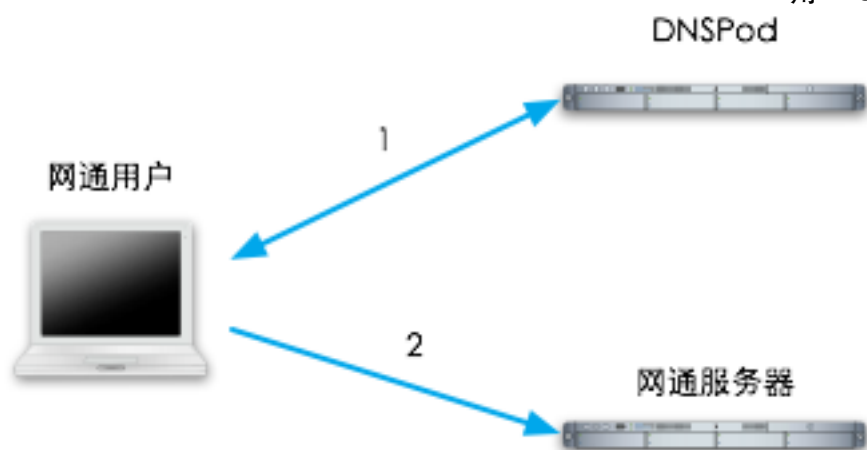


2.什么是智能DNS，什么是DNSPod

智能DNS的原理很简单：在用户解析一个域名的时候，判断一下用户的IP，然后跟DNS服务器内部的IP表匹配一下，看看用户是电信还是网通用户，然后给用户返回对应的IP地址。

DNSPod是奶罩同学旗下的网站，提供免费智能DNS服务。你只需要把你域名的DNS从原来的注册商改为DNSPod，马上就可以享用这个服务了。

注意了：DNSPod只提供域名的智能解析服务，不提供域名也不提供服务器，千万不要以为把你的域名转到DNSPod就能成为双线了。



第 2 章 前期准备

要继续后面的步骤，你必须准备好以下东西：

- ◆ 域名一个
- ◆ 服务器两台，电信网通各一台，一般的P4即可，内存2G以上
- ◆ 空白DVD R+光盘一张（或者CD-R 6张）
- ◆ DVD刻录机一个（或者CD RW刻录机一个）

然后，你还需要去下载CentOS的ISO镜像

随便在下面的镜像地址挑一个，暂时没有国内的，建议挂一个迅雷慢慢下

<http://ftp.ncnu.edu.tw/Linux/CentOS/5.0/isos/i386/>
<http://ftp.cse.yzu.edu.tw/pub/CentOS/5.0/isos/i386/>
<http://ftp.cs.pu.edu.tw/Linux/CentOS/5.0/isos/i386/>
<http://ftp.tcc.edu.tw/Linux/CentOS/5.0/isos/i386/>
<http://mirror.mirr4u.com/centos/5.0/isos/i386/>
<http://mirror.tini4u.net/centos/5.0/isos/i386/>
<http://mirror.averse.net/centos/5.0/isos/i386/>

打开上面的网址后，你可以发现类似下面的文件名

CentOS-5.0-i386-bin-1of6.iso

...

CentOS-5.0-i386-bin-6of6.iso

这些是CD镜像，一共有6个，下载刻盘比较浪费资源，所以我建议下载下面这个DVD镜像

CentOS-5.0-i386-bin-DVD.iso

废话不多说，下载回来后刻盘即可。

之后，为资源做一个安排，所以下面将会做一个假设：

假设域名为naizhao.com

假设两台服务器，电信的服务器为A，网通的服务器为B

假设A服务器为内容服务器，B服务器为CDN服务器，B服务器所有需要的内容均从A服务器取得

假设A服务器的IP是1.1.1.1，B服务器的IP是2.2.2.2

OK，到这里后我们所需要的东西基本准备好了。ISO可能还没有下载好，不急，慢慢挂着，我们先来把域名的DNS设置好。

第 3 章 设置域名的DNS

重头戏开始前，我们先来热身。

智能DNS可以说是CDN的基础。相信平时大家做网站都会有下面的情况：首页上放两个链接，电信用户请访问主力站点A，网通用户请访问镜像站点B。然后做两个域名，www.naizhao.com指向A，cnc.naizhao.com指向B。用户看了半天觉得头晕，随便点一个进去，发现访问速度真可以的，估计整个网页打开，中国都已经步入共产主义社会了。

有了智能DNS后，上面的问题迎刃而解。用户只需要输入www.naizhao.com，系统就会自动判断用户是电信线路还是网通线路上网，然后自动返回电信或者网通的服务器IP。整个过程对于用户来说是不可见的，用户只会觉得刷的一下网页就打开了，那叫一个爽。

废话不多说，let's do it！

1. 在DNSPod上面添加域名

打开www.dnspod.com，在右上角的导航菜单选择注册，根据提示输入信息完成注册。注册成功后返回首页登录。

登录后，看到有个添加新域名的框，输入naizhao（别输www。另外本文的例子就是naizhao.com），然后右边选择.com，完成后点击增加。



添加后，系统会自动跳到新添加的域名管理页面。可以看到里面什么东西都没有。不急，我们慢慢来。

在页面下方可以看到几个输入框和下拉列表，用于新增解析记录。我们先试着添加一条电信的记录看看。按照下面的格式输入：

主机记录：www

记录类型：A

线路类型：电信

记录值：1.1.1.1

优先级：（留空）

TTL：3600（默认）

完成后，点击增加。



有几个地方给大家说明一下：

主机记录：也就是域名前面的东西，比如www，那么记录生效后就是www.naizhao.com。如果要实现naizhao.com（前面没有www），那么添加的时候留空就好，系统会自动给加上一个@，代表域名本身。

记录类型：有A、CNAME、MX三种。最常用的就是A，记录值是一个IP地址。另外是MX，也就是邮件记录，做邮件服务器的时候用到，这里不深入讨论了。

线路类型：比较重要。如果选择了电信，那么代表这条记录*只能*给电信用户解析出来，网通用户是解析不出这条记录的。如果你一个域名只添加了一个电信的记录，那么网通用户将会无法访问。同样道理，网通的线路类型也一样。另外有一点要说明：非网通、非教育网的用户将会被解析到电信记录去。也就是说，国外用户、铁通、联通等用户都会被解析到电信服务器。

还有一个通用，通用的意思是，不管你是电信还是网通还是其他用户，都可以解析出这条记录。在存在电信和网通两条记录的情况下，不建议添加通用记录。

记录值：根据记录类型，分别输入一个IP或者一个域名（不是URL转发的域名）

优先级：用在MX记录中，数字越小，优先级越高。作用是：从优先级最大的服务器开始，如果邮件服务器出问题，邮件无法寄达，将会转发到优先级小一级的服务器上去。如果你没有很多台邮件服务器，这个就不用管了。

TTL：同样是比较重要的东西。数字是按照秒来算，3600也就是一个小时。意思是，用户请求一个域名的解析，在得到IP后，3600秒之内不会向DNS服务器重新请求解析，3600秒之内访问此域名，都将直接从自己机器的缓存里面获取IP。超出3600秒后，将会重新向DNS服务器请求获取IP。

OK，我们继续，添加一条网通的记录，让添加完成后的记录像下面的样子

管理域名记录						
主机记录	记录类型	线路类型	记录值	优先级	TTL	
www	A	电信	1.1.1.1		3600	修改 删除
www	A	网通	2.2.2.2		3600	修改 删除

到此，我们就添加成功了。等个半分钟，我们来测试一下是否已经生效。

A.Windows用户

我们使用nslookup命令，先在开始菜单-运行-cmd，回车打开命令行。

然后输入nslookup www.naizhao.com ns1.dnspod.net，稍等一下，看看服务器返回的结果：

Name: www.naizhao.com

Address: 1.1.1.1

如果你是网通用户，那么会得到下面的结果

Name: www.naizhao.com

Address: 2.2.2.2

只要你按照说明的步骤来说，基本不会有问题出现。如果无法正常返回结果，请稍等两分钟。如果等了两分钟还不行，那么就检查下你添加的记录是否正确了。

B.Linux/Unix/Mac用户

直接使用系统自带的dig命令。dig命令比nslookup更为强大。

输入dig @ns1.dnspod.net www.naizhao.com，会返回一大堆结果。我们只需要找到下面的文字就证明一切正常

```
;; ANSWER SECTION:
```

```
www.naizhao.com.      3600  IN    A      1.1.1.1
```

网通用户会看到

```
;; ANSWER SECTION:
```

```
www.naizhao.com.      3600  IN    A      2.2.2.2
```

至此，DNSPod方面的工作已经完成，下一步就是把域名的DNS服务器改为DNSPod的。

2.把域名的DNS转到DNSPod

域名迁移以前，我先提示一下风险：

根据域名的DNS协议规定，所有域名的NS记录（也就是指向DNS服务器的记录）的TTL时间必须为172800秒，也就是48小时。根据我们前面所学到有关TTL的知识，也就是说，一个域名如果被请求过一次以后，只要缓存不被清空，那么NS记录就会存在48小时。48小时内，任何有关此域名的记录的请求都发送到这个DNS服务器。换个角度来理解，如果A用户访问过你的网站，然后你在半个小时后修改你域名的DNS服务器，那么在剩余的47个半小时内，A用户对你这个域名的所有请求都会发送到你*旧的*DNS服务器，而不是新的服务器。B用户在你修改DNS成功后访问你的网站，那么B用户将会得到*新的*DNS服务器地址，所有请求都发往*新的*DNS服务器，而不是*旧的*。明白这点后，那么你就会知道，为什么修改域名的DNS后需要48个小时来生效。你也会明白，为什么记录会一下变成新的，一下变成旧的：这是因为，你地方的DNS服务器一般是2-3台，但他们每台之间的记录并不同步。

另外，一些很小气的域名注册商，在你修改DNS服务器后，会停止对你域名的解析。那么你将会冒着用户无法访问你站点的风险。对这样的用户，我一般建议在迁移域名的前一天，在你域名注册商的DNS管理后台，先把你所有域名记录的TTL，从默认的3600改为36000（10个小时）以上，并且选择在晚上10点以后修改DNS，因为DNS记录会在用户关机、关闭ADSL MODEM后失效（当然，很有可能还会从地方电信的DNS得到旧记录）。设置10个小时以上是让用户在就算旧DNS服务器停止服务后，也能从本地的缓存里面得到记录。

具体如何修改DNS，我就不具体说了，每个域名注册商都不一样。有几点要注意的：

1.填写DNS的时候，记得要把旧的全部去掉，不要以为新旧留着会稳定，这样只会让你的用户得到错误的记录。

2.尽量多填DNS服务器，这样能让你的域名更加稳定。目前DNSPod有6台DNS服务器，一般的域名注册商允许你填写4个（新网），万网在客户面板里面只能填写2个，但在域名管理面板（diy.hichina.com）可以把6个都填写进去。

目前DNSPod的6台DNS服务器分别为（注意了，是.net不是.com）：

ns1.dnspod.net
ns2.dnspod.net
ns3.dnspod.net
ns4.dnspod.net
ns5.dnspod.net
ns6.dnspod.net

折腾了老半天，大家也够累了。这个时候，ISO应该下载好了吧？辛苦一点，先把盘给刻好了。在下一章奶罩将会带你进入神秘的Linux殿堂。

第 4 章 安装CentOS Linux

1.基本安装

把光盘塞进光驱，设置BIOS从光驱启动（别告诉我你不会），然后会看到启动界面，上面有一些提示，可以输入一些命令进行高级安装或者进入系统修复模式。不用管，直接按回车。



稍等一下，会出现一个界面要求你做光盘的完整性检查。一般来说现在刻出来的盘不会有坏的，除非是下载回来的ISO文件有问题。所以这里我们跳过。


跳过的方法是：按键盘的TAB键进行控件的选择，被选择上的控件会高亮，我们选择Skip（建议大家随时打开翻译工具或者网站对不懂的英文进行翻译），然后按空格键进行确认。



然后再等一下，我们就可以看到安装的图形界面了。



按下下一步（Next），选择安装语言。因为Linux对中文支持并不好，黑漆漆的字符界面如果显示中文的话会有乱码（虽然远程连接上去的时候可以看到，但服务器一旦出现故障，需要在机房操作的时候，你就会发现默认英文是多么重要），所以我们就选择默认的英文（English），然后按下一步。



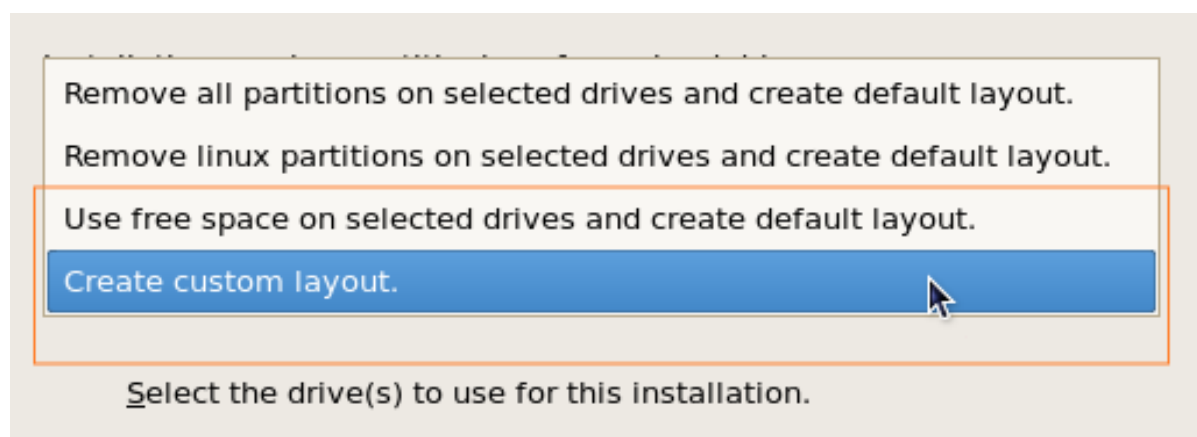
A vertical list of language options for system installation. The options are: Danish (Dansk), Dutch (Nederlands), English (English) (highlighted with a dark background), Estonian (eesti keel), and Finnish (suomi).

接着就是选择键盘样式。相信大家的键盘都是英文而不是日文的（玩过笔记本的都见过水货IBM吧？上面一撇一划的就是日文键盘），那么就留着默认U.S. English吧。什么？你刚好拿水货IBM来装？那还得用英文，否则打出来的字乱七八糟别怪我。

下一步，轮到选择安装方式了。因为我们是全新安装，所以就选择第一个Install CentOS，下一步。

接着是询问要如何分区了。坚决不要用一个分区安装所有文件，或者让系统自动划分。我们坚持所有分区都自己来划分。这样可以有效避免在单个分区出现问题的情况下会造成所有资料无法恢复的情况。（难理解？其实就跟Windows的C盘出问题一样。C盘出现严重问题无法读取，但可以使用工具盘启动系统，访问D、E等盘，把资料恢复出来）

我们在下拉菜单里面选择Create custom layout（创建自定义布局），下一步。



然后会看到当前机器上面硬盘的情况，Linux所有的设备都在目录/dev下面，普通的IDE硬盘的标识是hd（scsi、sata是sd），第一个硬盘是a，第二个是b，如此类推。所以这里如果你用的是普通IDE硬盘，然后接在第一根IDE线主盘的位置上，那么硬盘的标识就是/dev/hda。如果是sata硬盘，那么一般就是/dev/sda。如果光驱是IDE的，接在另外一根IDE线主设备口上，那么一般是/dev/hdc。

另外大家要对Linux的分区有认识。大家会觉得奇怪，Windows有C盘，D盘，为什么Linux没有？Linux的分区又是用什么一种方式存在的呢？

其实在Linux里面，每个分区都会作为一个目录对待。所有目录都从/（根）开始，树形发展。具体情况大家可以去参考相关资料，这里不多做阐述。这么做对于我们有一个最大的好处：某个目录分区空间用完了，我们只需要加装一个大硬盘，然后把这个硬盘整个分成一个区，在系统里面加载，把这个目录里面的文件全部拷贝进来，作为这个目录，这样就不怕空间不够了。

接着说说分区的文件系统。用Windows的朋友一定都知道fat32,ntfs吧？Linux下面最常用的文件系统是ext3和reiserfs，另外有比较旧的ext2，现在已经很少人用了，还有SWAP，交换分区。几种文件系统的优劣就不做评比了，文内一律使用ext3，因为CentOS对ext3支持比reiserfs要好点。

Linux的基本系统包括了下面几个目录

/ 根分区，如果你想把整个硬盘分一个区，那么就分这个区

/bin,/sbin 都是一些用到的命令

/boot 放置启动文件

/usr 包括一些命令、库、帮助等一些不常改变的文件

/usr/local 同上，一般新的程序都安装在这里，就跟Windows的Program files差不多

/var 都是一些经常变化的文件，比如log

/home 看名字就知道，用户住的地方

/etc 里面都是设置文件，跟注册表有点类似作用

/lib 共享库，so文件，作用类似windows的dll文件。打开里面会看到一些数字，其实是版本号

/root 老大住的地方，后面会提到

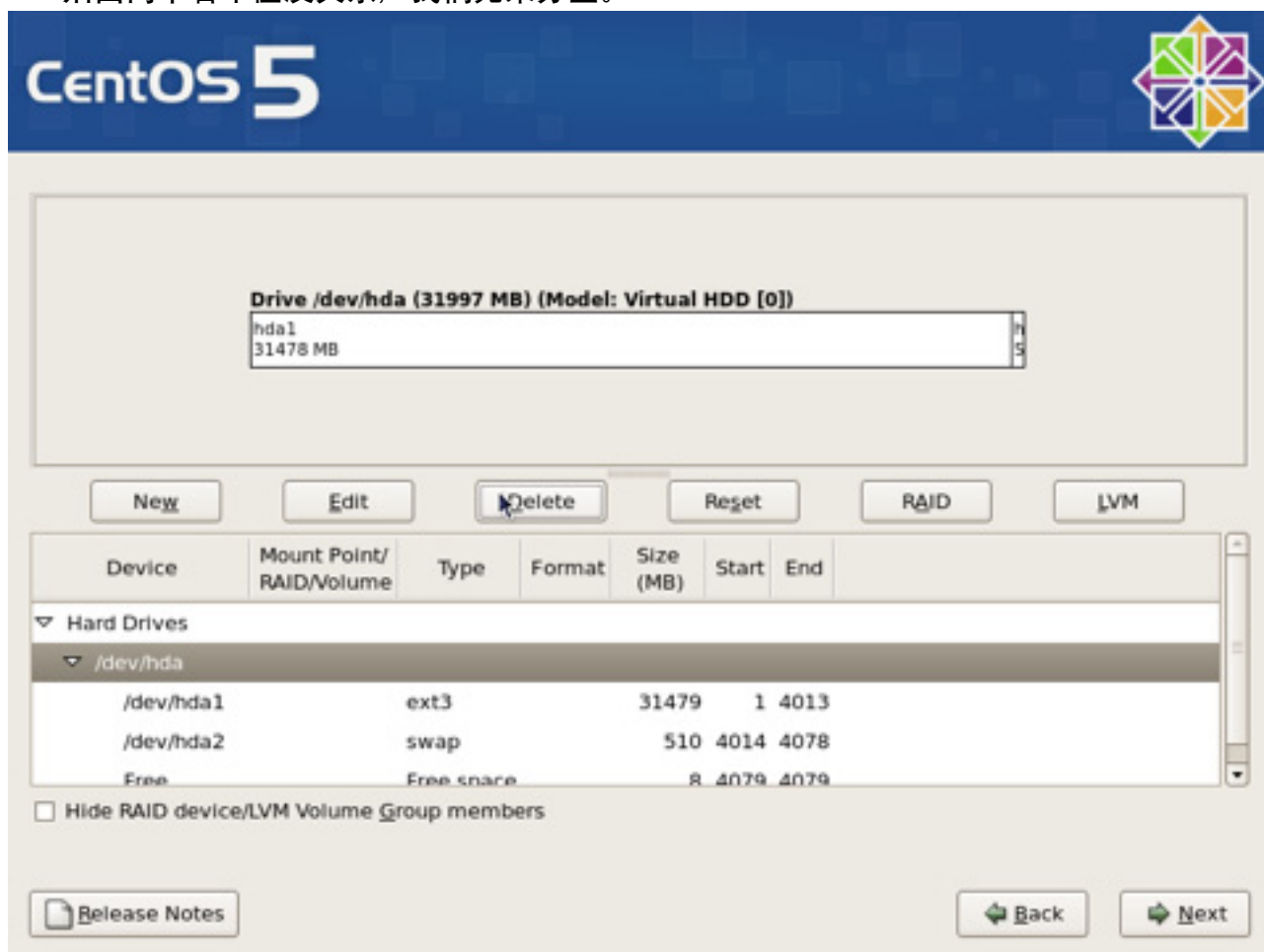
下面是几个比较特殊的目录

SWAP 看清楚了，前面没有/的。Swap是交换分区，也就是Windows里面的虚拟内存，但不一样的是基本不会用到。分区的时候只需要分一个，不需要设置挂载点，文件系统选择好就行。

/dev 存放设备系统，属于虚拟文件系统

/proc 存放有进程信息，同样是虚拟文件系统。可以通过此文件系统实时修改内核或者进程，也就是交互。比如可以实时查看内存的使用、CPU信息

后面两个看不懂没关系，我们先来分区。



上面的图片可以看到，硬盘原来存在分区，我们把/dev/hda选上，然后点上面菜单的Delete整个删除。

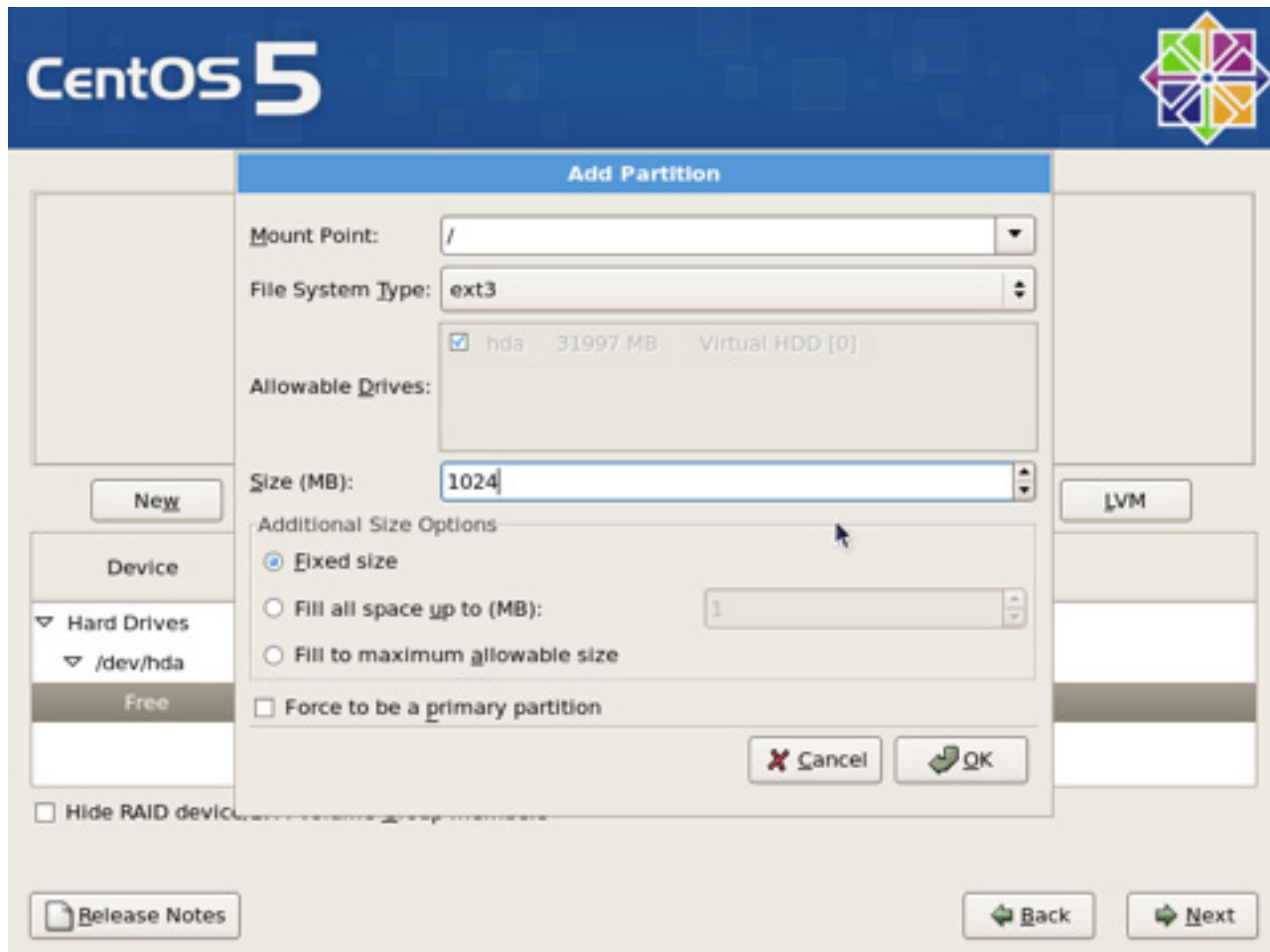
接着我们点一下菜单的New，开始分区，先分一个/，文件系统选择ext3，分区大小是1G，也就是1024M。

Mount Point就是挂载点，输入/

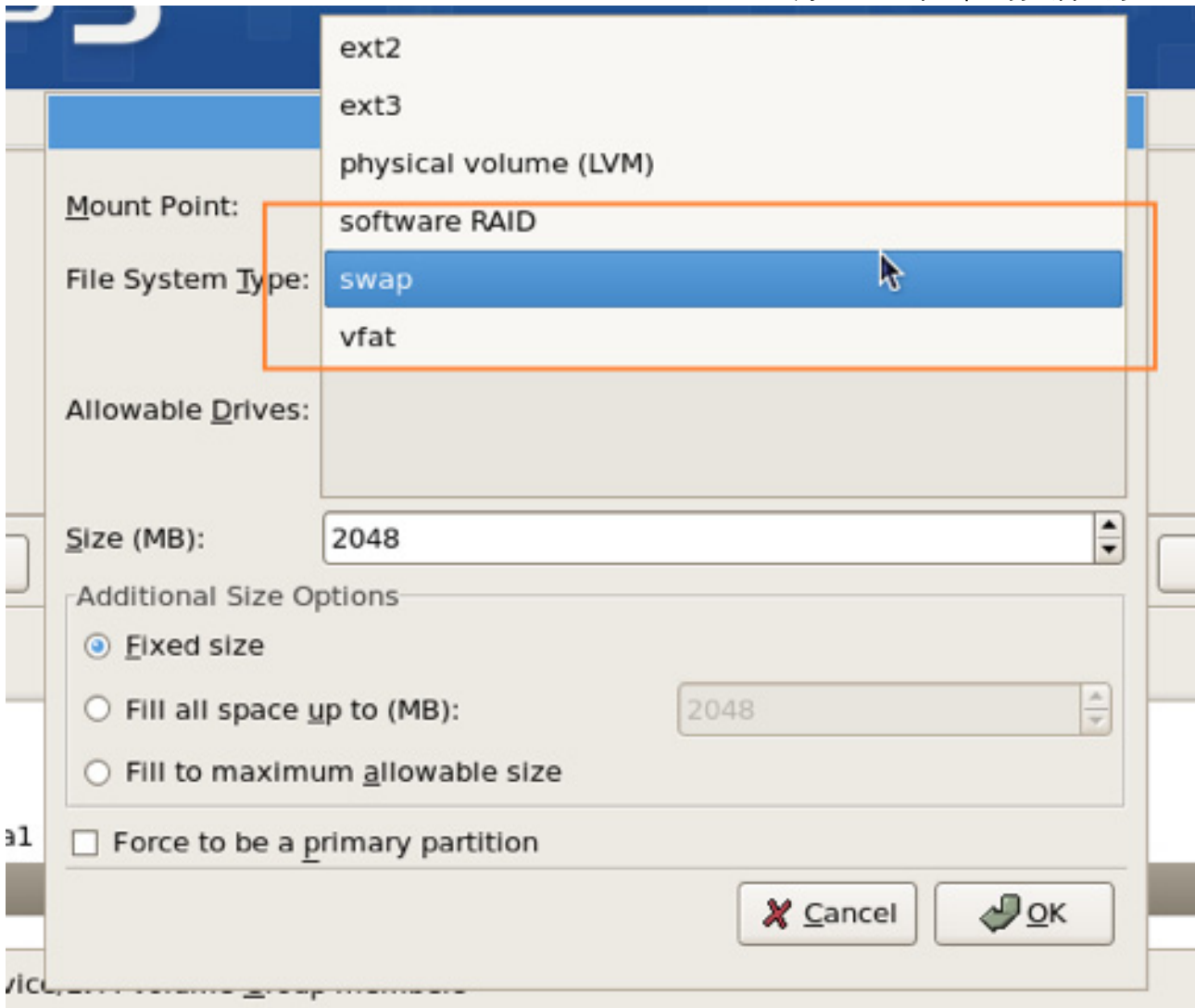
File System Type文件系统，下拉中选择ext3

Size分区大小，按照MB计算的，输入1024

完成后按OK



跟着分SWAP，点New，直接在File System Type选择swap，Size输入2048，按OK



剩余的分区大家按照下面来分

/ 1024

swap 2048

/usr 5120

/usr/local 5120

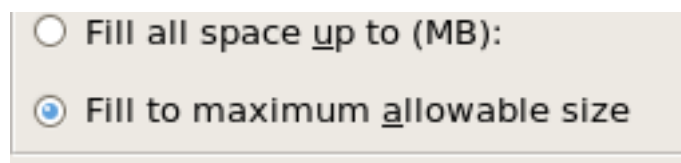
/var 5120

/tmp 2048

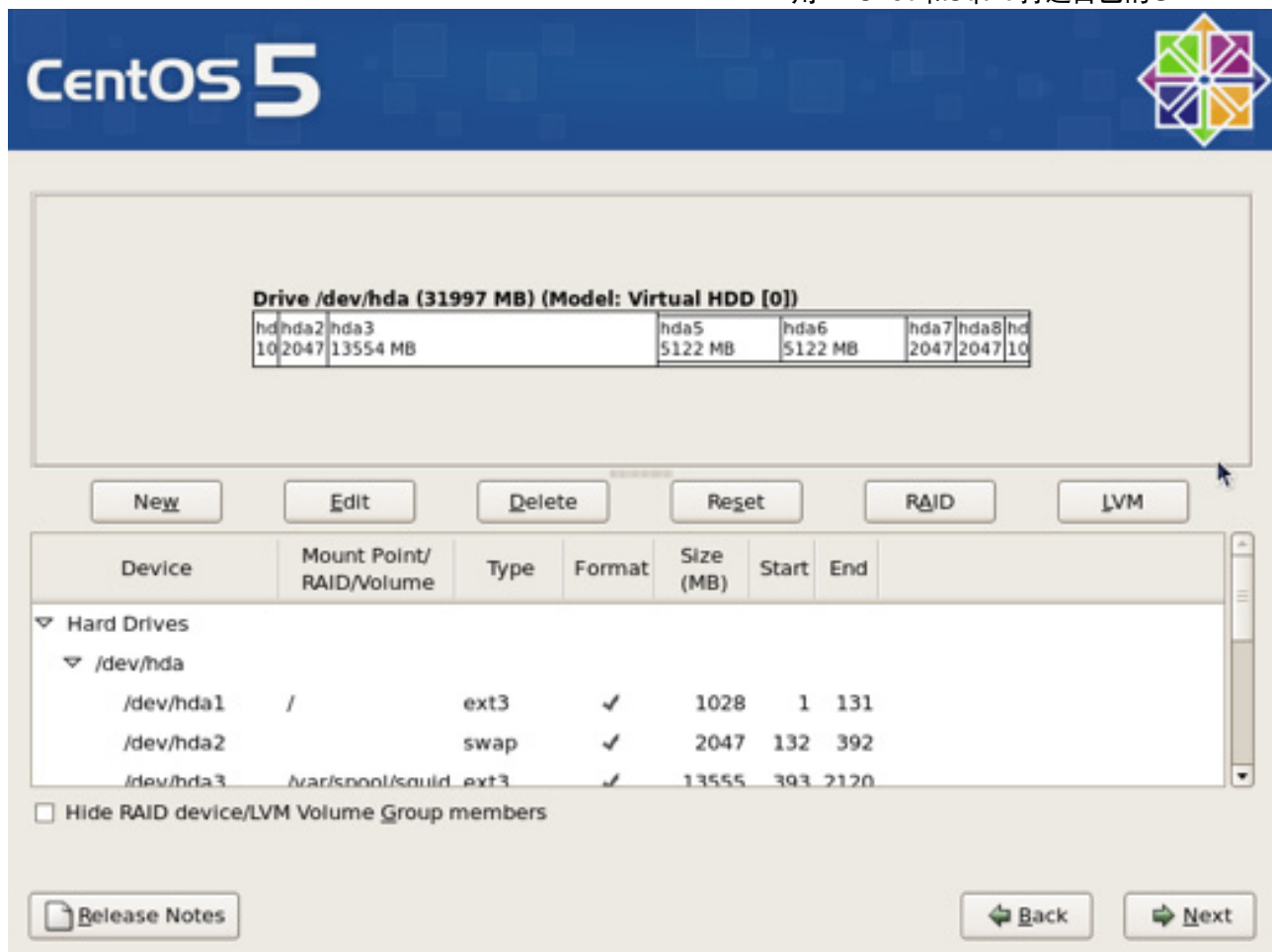
/usr/local/squid 1024

/var/spool/squid 所有剩余的空间

所有剩余的空间要怎么选？很简单，选上Fill to maximum allowable size即可。



分区分完以后，看起来像下面的样子



OK, 所有分区都完了以后, 我们继续下一步。

接着会出现一个安装GRUB的界面。GRUB是用来启动系统的, 包括双系统的启动都要用上, 装好后在启动的时候你会看到有一个菜单给你选择。这里不用更改什么, 直接下一步就好。

如果你的网卡能顺利被系统认出来的话, 这步就应该是让你配置网卡参数了。

点击Edit, 出现网卡配置页面。先把Use dynamic IP configuration(DHCP)去掉, DHCP是自动从网关分配IP, 因为我们的IP是固定的, 所以不需要。

然后IPv6我们也不用, Enable IPv6 support也去掉。

Activate on boot表示系统启动后自动启动网卡, 要选上。

因为这台服务器是网通使用, IP是2.2.2.2, 所以在Address里面就输入2.2.2.2, Prefix(Netmask)是子网掩码, 可以输入255.255.255.0或者24。24是一种名为CIDR的寻址方式, 具体可以去Google一下, 这里不深入解释。

配置好的网卡如下图

Edit Interface eth0

Configure eth0 - Realtek Semiconductor Co., Ltd. RTL-8029(AS)

Hardware address: 00:E5:5A:8F:E5:A1

☐ Use dynamic IP configuration (DHCP)

☒ Enable IPv4 support

☐ Enable IPv6 support

☒ Activate on boot

	Address	Prefix (Netmask)
IPv4:	2.2.2.2	255.255.255.0
IPv6:		

保存后，我们还需要设置主机名，这里就设置为cnc.naizhao.com吧。

CentOS 5

Network Devices

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix
<input checked="" type="checkbox"/>	eth0	2.2.2.2/24	Disabled

Hostname

Set the hostname:

☐ automatically via DHCP

☒ manually (e.g., host.domain.com)

Miscellaneous Settings

Gateway:

Primary DNS:

Secondary DNS:

然后还有一些杂项(Miscellaneous Settings), Gateway是网关, Primary DNS是主DNS, Secondary DNS是次要DNS, 根据具体需求设置上去。

跟着设置时区。点一下地图的右上角, 然后在上海的位置点一下, 左下角的下拉菜单会变为Asia/Shanghai, 下一步。

跟着就是设置root的密码。Linux跟windows不一样, windows可以有很多个管理员, Linux只能有一个root, 一般用户要行使管理员的身份, 必须要用su这个命令临时改变身份为管理员。

输入两次密码后确认, 下一步。

接着就是选择安装类型了。把Desktop什么的全部去掉, 在Server前面勾上。下面会有一行提示, 问你要在什么时候进行所安装的包的详细配置(You can further customize the software selection now, or after install via the software management application), 选上Customize now。下一步。

接着是安装包选择界面。左边会有Desktop Environments/Applications/Development等。按照下面的列表进行选择:

- * Desktop Environments
 - 全部去掉
- * Applications
 - Editors 选上
 - 其余全部去掉
- * Development
 - Development Libraries 选上
 - Development Tools 选上
 - 其余全部去掉
- * Servers
 - 全部去掉
- * Base System
 - Dialup Networking Support 去掉
- * 其他均为默认

下一步, 开始安装。等待安装完成后重启即可。

2. 初次启动

初次启动的时候, 会有一个设置代理, 就跟安装时要求检测CD完整性的界面差不多, 用上下键选择Firewall configuration, 按一下TAB键, 空格进入。

然后在Security Level选择Disabled, 也就是关闭防火墙, TAB键选择OK, 空格退出。之后系统会继续启动, 直到出现登录界面。

CentOS release 5 (Final)

Kernel 2.6.18-8.el5 on an i686

cnc login:

3. 远程连接

Linux的远程连接方式是ssh。ssh是一种经过加密的连接方式, 所有数据交换都会被加密, 就算服务器被arp攻击也不怕密码会被监听。

Windows下的ssh客户端, 我们一般使用putty。因为putty太好用了, 所以有台湾的网友进行了修改并且本地化, 发布出来的程序名字叫做pietty。

Pietty的官方网站是: <http://www.csie.ntu.edu.tw/~piaip/pietty/>

下载地址是: <http://www.csie.ntu.edu.tw/~piaip/pietty/stable/pietty0327.exe>

下载回来运行，然后输入服务器的IP，回车就可以连接了。如果是第一次连接服务器，程序还会提示你是否接受并保存服务器的证书，选择是即可。然后按照提示输入用户名root和密码。

4.Linux的基本命令

首先要注意：Linux是一个大小写区分的系统。不管你做什么事情都要注意大小写。cpu和Cpu是不一样的。

说几个比较常用的

reset 重启系统，需要root权限

poweroff 关机，同样需要root权限

ls 列出文件和目录。默认是当前目录，如果要列出/bin下面的文件，输入ls /bin

cd 进入某个目录。比如cd /bin可以进入/bin目录，cd ..进入上一级目录，cd abc进入当前目录下面的abc目录，cd -回到上一次进入的目录。

cp 拷贝文件或者目录。cp a b拷贝a文件为b文件

mkdir 建立目录。比如mkdir naizhao。如果要建立/a/b/c/d/e这个目录，但/a/b/c/d这几个目录都不存在，使用mkdir -p /a/b/c/d/e，那么会把这几个目录一并创建。

rm 删除文件或者目录。rm a删除a文件。要删除一个目录必须要用rm -rf a，表示删除a目录

mv 移动文件。不过大部分情况下用来给文件改名。因为linux下面没有专门改名的命令。mv a b把a文件改名为b。mv a ../把a文件移动到上一级目录

more 比较有用的命令。比如用了ls命令，但出来的文字太多，已经超过一个屏幕了，我要看开头部分的内容要怎么看？可以使用ls|more，这样就可以随意的翻页了。|是管道，负责把|前面命令的结果交给|后面的命令。管道可以使用多个，比如ls -lh|grep naizhao|more，也就是说把ls -lh的结果交给grep处理，然后grep把处理出来的结果交给more。

grep 也是比较常用的命令。负责把所需要的内容抽取出来。比如我ls -lh /lib，内容太多了，并不都是我所需要的，我只需要包含dns这几个字的内容，那么我就使用ls -lh /lib|grep dns。如果我需要不包括dns这三个字的内容呢？使用ls -lh /lib|grep -v dns

cat 不是猫。cat命令负责把一个文件里面的内容完整输出来。内容多的时候配合more使用。比如cat /proc/cpuinfo，会把cpuinfo里面的内容显示出来。

wc 噢，不是厕所，别误会。这是用来算数的。比如我要算一个文件里面有多少行，那么使用cat /proc/cpuinfo|wc -l，输出的数字就是cpuinfo文件的行数，-l参数是按照行(line)来计算。这个命令最常用的时候是什么呢？看网络连接，比如要看看当前系统是不是被别人SYN了，那么输入netstat -an|grep SYN|wc -l

netstat 查看网络状态，一般用来看系统当前有几个连接。比如看80端口有多少个连接，输入netstat -an|grep :80|wc -l

这些命令建议大家多点玩，不要怕。玩坏了系统大不了重装，反正上面现在也没资料。

5.VIM的基本使用

VI是所有Unix系统必带的工具。干什么用的呢？Windows的记事本，知道吧？这个东西比记事本还要高级几百倍，整个Linux基本就是在VI里面写的。VIM则是VI的改进版本，某些系统里面只带了VI，没有VIM。但这两个东西命令上是没有分别的，都是命令行里面敲入vi就可以启动。

我们要学习以下几个方面：打开文件、编辑文件、保存文件

我们以/etc/hosts文件为例，在里面添加一条1.1.1.1 www.naizhao.com的记录。

A. 打开文件

打开文件很简单，vi 文件名，比如vi /etc/hosts，然后会看到类似下面的内容

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          cnc.naizhao.com cnc localhost.localhost localhost
::1               localhost6.localhost6 localhost6
```

B. 编辑文件

编辑一共三个方面：插入（添加）、修改、删除

vi里面，可以通过键盘的方向键移动光标。我们移动到

```
::1          localhost6.localhost6 localhost6
```

这行，然后按下键盘的o（字母）键，会发现光标跑下面去了，自动新起一行。

然后我们输入1.1.1.1，按两次键盘的tab键，再输入www.naizhao.com，然内容看起来像下面这样

```
1.1.1.1      www.naizhao.com
```

然后，我们按两次键盘左上角的esc键。在按第一次的时候，你会发现光标向前跳了一下，马上再按第二次，如果你打开了音箱，你会听到当的一声响。

好了，添加新一行就是那么简单。

然后，我们再为1.1.1.1添加一个记录，tel.naizhao.com，怎么做呢？

移动光标到www.naizhao.com的m上，按一下键盘的i（insert的意思）键，这样编辑模式就会变为插入。嗯，敲个空格看看。啊，怎么把我的com给分开成co m了？我要的不是这个效果啊！

不急不急，按照前面的方法，按两次esc键，退出编辑模式，然后按一次键盘的u（undo）键。哦，还好，东西都回来了。原来i是在当前文字的前面插入啊。那再来一次，光标移动到m上面，按一下i，然后用键盘把光标向右移动一下，来到m的背后，敲一个空格，然后输入tel.naizhao.com，然后按两次esc键。嗯，看起来有模有样了。

喂，我怕文件被我破坏了，想要先保存一下，怎么办？

保存文件啊，也很简单啊。先输入一个冒号（:）。什么？不知道怎么输？按住shift键，敲一下回车键左边的分号；，这不，冒号出来了。然后冒号后面跟着一个w（write），看起来就像这个样子:w，回车。你会看到一行提示

```
"/etc/hosts" 5L, 248C written
```

表示文件保存完成。

然后，呃，还有修改。如果我发现某个字打错了，比如com的o，我打成0了，我要怎么改？

先把光标移动到0上面，然后按键盘的r（replace），然后按一下o，哈哈，改过来了，赶紧:w保存先。

嗯。。。如果我不需要tel.naizhao.com，我要怎么删除啊？

删除的英文是什么来着？delete吧？鼠标移动到tel的t上面，按一下键盘d，然后按一下右方向键，嘿嘿，t被删掉了。但是，我要删除整个url，一个一个删过去，是否麻烦了点？嗯，有点道理，跟我一起数数吧。1,2,3,4,5...tl.naizhao.com一共有14个字符，那我就先按一下键盘的d，然后用数字键输入14（带有特殊符号那行，不是小键盘）（呃，怎么输入后看不到的？），然后按一下右方向键，嗯，世界清净了。真完美。

另外，还有一些比较方便的快捷键，比如输入dd，删除当前行，输入gg，会回到文件的开头位置，输入G（也就是shift+g）会到文件的末尾，ctrl+d到下一页，ctrl+b到上一

页，shift+a在文件末尾插入，shift+d删除当前光标位置一直到行末的文字，这些都等大家慢慢琢磨。

C.保存文件

保存文件刚才说了一个:w，如果我想直接保存退出，要怎样呢？可以使用:wq (write&quit)，或者直接:x。

但是，文件刚才被我改到一塌糊涂，我不想保存了，我要直接退出，怎么退？

如果文件没有被改变过，可以直接输入:q退出。但如果文件被改动过，但又没有保存，:q会报错，那么可以在q前面加上一个!，强行退出，就像这个样子:!q

如果文件只读，你要强行写入，当然可以在w的前面加上!了。

6.设置安装更新

国外的东西在国内有个很麻烦的地方：镜像都在国外。所以，要升级系统，必须要把镜像改为国内的私人镜像。于是乎，刚才所学到的VIM知识又用上了。

CentOS的升级工具为yum，镜像配置文件在/etc/yum.repos.d/CentOS-Base.repo，那么，我们用vi打开吧。

vi /etc/yum.repos.d/CentOS-Base.repo

呃，里面好多网址啊，难道要我一个一个都改过去？岂不是累死我？

不怕不怕，记事本都有查找替换呢，vi当然也少不了这个功能了。

yum默认是从mirrorlist=这行获取镜像列表的，既然我们指定了镜像地址，那么就需要把这行注释了。注释的方法是在前面加上#，也就是#mirrorlist=这样。

赶紧开始操作，vi里面查找替换的格式大概是这样:%s/a/b/g，也就是查找a，替换成b，g则是全文件查找。把公式套用过来，就是这个样子

:%s/mirrorlist=##mirrorlist=/g

嗯嗯，输入回车看看，都替换掉了吧？哈哈。

接着，指定url地址要把#baseurl=前面的#去掉，根据刚才的例子，得到

:%s/#baseurl=/baseurl=/g

又OK了一个，剩下最后一个了。

国内私人搭建的centos镜像，比较有名的是be10.com，网址是http://mirror.be10.com/

打开上面的网址，可以看到孤零零的一个centos目录，对比一下配置文件里面的url

http://mirror.be10.com/centos/

http://mirror.centos.org/centos/

找不同游戏开始了，看起来，两个url的差别就是be10.com和centos.org了，也就是说我只要把centos.org替换成be10.com就可以了。

:%s/centos.org/be10.com/g

大功告成，赶紧保存退出，然后输入

yum update

更新吧。

更新的过程中会有互动节目，有几个地方会让你输入y来确认是否要安装更新包，直接输入y回车确认就好。

忙活了老半天，大家都累的够呛了，让机器慢慢更新，我们喝杯咖啡泡个功夫茶去。接下来的事情还多着呢。

第 5 章 安装Squid的前期准备

从本章开始，大家将会学到如何在Linux下面安装、编译程序，还会学到程序编译的优化方法，最后会通过源代码编译的方式把Squid安装上。

1.Linux下面程序安装的基本知识

A.程序安装包的分类

平时大家在Windows下面安装程序，一般都是直接运行安装程序，然后安装程序会把编译好的二进制文件拷贝到系统里面，最终完成安装过程。

Linux下面，安装程序的自由度比Windows要大。因为Linux下面大部分东西都是开源的，所以程序的提供方式一般是二进制包或者源代码包。

二进制包，同样会有不同的打包和管理方式（类似于zip、rar，但更加先进）。最常见的两种软件包为rpm（redhat、suse等系统采用的包）和deb（debian、ubuntu）。机器上面安装的软件包多了，自然需要一个软件来进行管理、更新。所以就有了yum（rpm）、apt（deb）这样的软件包管理、升级软件。

源代码包，这样的包比较简单，一般都是直接把源代码打包成一个压缩文件，后缀一般是tar.gz或者tar.bz2

通过包管理软件安装软件比较简单，设置好软件的安装源后，可以使用下面的命令安装软件：

```
yum install php (yum)
```

```
apt-get install php5(apt)
```

如果不知道需要安装的软件包确切名字，可以使用下面的方法搜索：

```
yum search php(yum)
```

```
apt-cache search php(apt)
```

本文章主要探讨源代码包的安装。

B.通过源代码包安装程序

一般情况下，程序从源代码到能正常使用，需要经过三个步骤：1.根据系统环境对源代码进行配置（configure）；2.编译程序（make）；3.安装程序（make install）。

a.源代码配置

通过运行configure脚本，可以对所需要编译的程序的一些基本环境、所使用到的命令、库等文件进行自动查找，并且生成用于编译文件所需要的Makefile文件。因为Linux有太多的发行版本了，每个版本的环境都不一样，所以需要用到这个脚本。通过configure脚本还可以对程序的模块进行自定义，可以禁用或者激活某个功能。

b.编译程序

根据configure脚本生成的Makefile文件，对源代码进行编译、链接，生成二进制文件。但这个时候的二进制文件一般来说还不能使用。

c.安装程序

通过make install命令，把编译好的二进制文件安装到相应的路径，这个时候程序才真正可以使用。

C.程序安装到哪里？

一般编译安装的程序都会安装到/usr/local下面。比如php就会被安装到/usr/local/bin/php，php.ini会被安装到/usr/local/lib/php.ini，等等。如果需要指定安装目录，在configure的时候指定--prefix参数，会把所有文件都安装到prefix目录中。比如./configure --

prefix=/usr/local/php。那么会把所有文件都安装到这个目录，最后看起来会是
/usr/local/php/bin/php和/usr/local/php/lib/php.ini

D.如何得到configure的参数？

常用的configure参数有--prefix，可以通过./configure --help得到更多的参数。一般带有--enable-、--with-开头的参数都是启用某个功能，--disable-、--without-开头的都是禁用某个功能。

E.如何删除安装好的程序？

Linux删除程序跟Windows不太一样。Windows建议使用卸载功能卸载程序，Linux下通过包管理软件安装的程序也可以直接用命令删除，例如：

```
yum remove php(yum)
apt-get remove php5(apt)
```

如果通过源代码包安装的程序，一般会带有make uninstall，通过这个命令可以删除所安装的文件。如果不带有make uninstall，那么可以直接把程序安装目录删除。

注意：如果是依赖库（比如php如果要使用到mysql的功能，必须先安装mysql，然后在配置php的时候指定mysql库的路径，那么php就是依赖了mysql），删除被依赖的程序，那么用到依赖库的程序都将会无法使用（比如删除mysql后，php将无法正常使用mysql部分的功能）。

F.还有个make clean是干什么用的？

make clean用来打扫战场。把所有编译时留下的临时文件、编译后的二进制文件等都清理掉。一般建议在make install后make clean，方便下次重新编译和节省空间。

2.源代码编译时的优化

用Windows的朋友都知道，平时最头痛的事情就是程序占用CPU太高，而且消耗内存特别大。这是因为Windows的程序都是一些“通用”程序，并没有针对特定的平台和特定的CPU进行优化。在Linux下，通过源代码来编译二进制文件，这个问题将得到有效的改善。我们可以通过添加优化参数，让程序针对某个CPU型号和某个系统进行最大优化，减少文件大小、CPU使用率、内存使用率。

但是，通过指定优化参数来编译的程序，将不再具有跨系统跨平台的能力。就算两台机器系统版本一样，只要CPU不一样，程序都无法运行。程序只能在所编译的机器上运行。

一般情况下，优化参数都是通过export命令设定CFLAGS和CXXFLAGS，然后configure的时候会自动读入，make的时候会自动使用所选的优化参数。

例如，普通Pentium 4的CPU（显示的型号为Intel(R) Pentium(R) 4 CPU XXXXMHz，cpu family: 15，model: 0/1/2）就可以输入下面的命令

```
export CHOST="i686-pc-linux-gnu"
export CFLAGS="-march=pentium4 -O2 -pipe -fomit-frame-pointer"
export CXXFLAGS="${CFLAGS}"
```

CPU型号等信息可以通过输入下面的命令得到

```
cat /proc/cpuinfo
```

将会输出类似下面的信息

```
processor      : 0
vendor_id     : AuthenticAMD
cpu family    : 15
model         : 47
model name    : AMD Athlon(tm) 64 Processor 3200+
stepping      : 2
cpu MHz       : 2000.336
cache size    : 512 KB
fdiv_bug      : no
hlt_bug       : no
f00f_bug      : no
coma_bug      : no
fpu           : yes
fpu_exception : yes
cpuid level   : 1
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
mmx fxsr sse sse2 syscall nx mmxext fxsr_opt lm 3dnowext 3dnow up pni lahf_lm ts fid vid ttp tm
stc
bogomips      : 4002.57
```

上面的信息显示了一颗AMD Athlon 64 3200+的CPU。

CPU优化参数可以从下面地址获得（有些cpu family和model没有，可以用相同型号的代替）

http://gentoo-wiki.com/Safe_Cflags

3. 编译Squid的前期准备

我们需要先了解一个叫做“文件描述符”的东西。我们知道，人能同时干的事情是有限的。文件描述符就是一个限制最大上限的东西。文件描述符用来限制一个程序能同时打开的文件数目，默认是1024。也就是说，如果不修改文件描述符，那么一个程序只能同时打开1024个文件。1024这个数字，一般程序已经够用了，但Squid不行。Squid会同时打开成千上万个文件，以保证最高的效率和响应速度。

想象一下下面两种情况：a.用户每次访问squid，squid会按照需要打开文件，然后读取文件内容再返回给用户。b.squid预先打开访问频率高的文件，用户访问squid，squid直接把内容返回给用户。相对来说，后面的方法可以对用户的请求进行更加快的反应。

要更改文件描述符的大小，必须要修改两个文件。

```
/usr/include/bits/typesizes.h
```

```
/usr/include/linux/posix_types.h
```

用vi打开上面的文件（如果忘记如何使用，请参考前面章节），查找

```
#define __FD_SETSIZE 1024
```

把1024改为65536，然后保存。

为什么是65536，不是更加大？因为这是Linux能接受的最大数值。

刚才编辑的这两个文件是C/C++程序里面的头文件，编译squid的时候会被自动引用。除了这两个文件以外，我们还需要对当前环境进行设置。

环境，也就是你用ssh登录到系统时的一些设定。每个登录进程都可以进行单独的设置，在没有把设置写入环境配置文件(.profile, .bash_rc)的情况下，关闭登录进程后环境设置会丢失。

举个例子，你使用pietty开了两个窗口，使用同一个帐号密码，同时登录到同一台服务器上面，然后在其中一个登录进程中使用export命令，那么只会在这个登录进程中生效，对另外一个登录进程不起作用。

明白后，我们说一下ulimit命令。ulimit用来设置当前环境的一些资源限制。刚才所说的，这个命令是对环境进行设置，所以退出当前登录进程后命令将会失效。

我们输入下面的命令

```
ulimit -Hs 65536  
ulimit -n 65536
```

H参数是硬性限制，s是堆栈上限，n是文件描述符上限。

最后，我们使用wget把squid的源代码下载回来。

```
wget http://www.squid-cache.org/Versions/v2/2.6/squid-2.6.STABLE13.tar.gz
```

wget是unix下一个支持断点续传的下载工具。会有一些比较实用的功能，比如把别人整个网站都下载回来（像平时用的小偷吧？）。

第 6 章 编译并安装Squid

首先使用tar把源代码压缩包解压

```
tar zxvf squid-2.6.STABLE13.tar.gz
```

解压后，我们得到一个名为 squid-2.6.STABLE13 的目录。进入目录

```
cd squid-2.6.STABLE13
```

在configure前，我们必须要先设置cflags，这里我们假设CPU是intel core duo的，cpu family 6,model 14。通过 http://gentoo-wiki.com/Safe_Cflags#Intel_Core_Solo.2FDuo 可以找到对应的优化参数

```
CHOST="i686-pc-linux-gnu"  
CFLAGS="-march=prescott -O2 -pipe -fomit-frame-pointer"  
CXXFLAGS="${CFLAGS}"
```

然后通过export命令设置

```
export CHOST="i686-pc-linux-gnu"  
export CFLAGS="-march=prescott -O2 -pipe -fomit-frame-pointer"  
export CXXFLAGS="${CFLAGS}"
```

设置完成后，我们可以通过env命令来查看是否设置成功。

然后，我们开始configure源代码。使用下面的命令，把squid安装到/usr/local/squid目录中去。

```
./configure --prefix=/usr/local/squid --enable-follow-x-forwarded-for --enable-storeio=aufs,ufs  
--with-maxfd=65536 --with-pthreads --enable-dlmalloc --enable-poll --enable-stacktraces --enable-  
removal-policies=heap,lru --enable-delay-pools
```

对应参数的作用可以使用./configure --help得到。

这里大家要注意下，configure前面还有一个./，这个代表当前目录。意思是当前目录下面的configure文件。linux如果要执行当前目录的文件，必须要加上./。如果要执行当前目录下的下级目录里面的文件，可以不用加，比如bin/run。当然，./bin/run也是一样的效果。

接着我们会看到屏幕向下翻滚，出现一大堆checking for ...的字样。一直等到下面的文字出现，并且停止，那么configure就算完成了。

```
config.status: creating tools/Makefile  
config.status: creating include/autoconf.h  
config.status: executing depfiles commands  
[root@cnc squid-2.6.STABLE13]#
```

然后，我们输入make编译源代码，又是一行行的英文翻滚。是否有点感觉做黑客的味道？我们要做的还是继续等待下面的文字出现

```
make[2]: Leaving directory `/root/squid-2.6.STABLE13/tools'  
make[1]: Leaving directory `/root/squid-2.6.STABLE13/tools'  
make[1]: Entering directory `/root/squid-2.6.STABLE13'
```

```
make[1]: Nothing to be done for `all-am'.
make[1]: Leaving directory `/root/squid-2.6.STABLE13'
[root@cnc squid-2.6.STABLE13]#
```

这样源代码就算编译完成了。然后我们运行make install安装，等待下面的提示

```
make[2]: Leaving directory `/root/squid-2.6.STABLE13'
make[1]: Leaving directory `/root/squid-2.6.STABLE13'
[root@cnc squid-2.6.STABLE13]#
```

这里要注意：如果任何一个步骤出现带有error之类的警告，那么就是编译出错，需要检查到底是什么步骤出现问题并且改正。如果严格按照本文来编译，一般是不会出现错误的。另外，cflags参数使用不当同样会造成编译错误。

如果出现编译错误，必须要先make clean，然后重新make。（如果错误是在configure这一步就不需要了）

到这里，squid就算是编译安装完毕了。我们可以到/usr/local/squid目录看看，里面已经躺着一大堆文件了。

```
[root@cnc squid-2.6.STABLE13]# cd /usr/local/squid
[root@cnc squid]# ls -lh
total 72K
drwxr-xr-x 2 root root 4.0K Jul  7 02:27 bin
drwxr-xr-x 2 root root 4.0K Jul  7 02:27 etc
drwxr-xr-x 2 root root 4.0K Jul  7 02:27 libexec
drwx----- 2 root root 16K Jun 15 08:09 lost+found
drwxr-xr-x 3 root root 4.0K Jul  7 02:27 man
drwxr-xr-x 2 root root 4.0K Jul  7 02:25 sbin
drwxr-xr-x 4 root root 4.0K Jul  7 02:25 share
drwxr-xr-x 3 root root 4.0K Jul  7 02:25 var
[root@cnc squid]#
```

接着我们要做的是配置squid。

第 7 章 配置Squid

1. 设置hosts表

首先我们要明白一下squid在CDN中扮演的角色。squid在CDN中其实只是一个代理 + 缓存。跟一般的代理服务器类似，squid代替用户向真正有内容的服务器进行请求，并且缓存下来。所不同的是，squid是作为一个类似apache一样的web服务器存在，用户访问的时候并不感觉到是访问了一台代理服务器，而是一个真实的web服务器。一般情况下，我们把这种方式称之为“反向代理”。

所以，我们需要让squid知道内容要到哪里去取。因为我们已经把域名naizhao.com设置了智能DNS，而且这台squid服务器也是网通的，按照正常的dns解析，会把naizhao.com解析为2.2.2.2这个IP（也就是网通机器本身）。但这个IP上面并不存在任何所需要的内容，所有需要的内容其实是在1.1.1.1上（源服务器）。所以我们需要为naizhao.com固定一个IP，也就是1.1.1.1。

让系统得到这个IP的方法有两种，一种是设置hosts表，另外一种是在系统里面搭建一个dns服务器。Windows下大家应该会经常去编辑到hosts文件，linux下同样也存在这个文件，我们在前面的vim教程中已经提到过。搭建dns系统这个方法工程太过庞大，还是编辑hosts文件相对来说比较简单，所以我们就采用编辑hosts表的方法。

linux下面的hosts表位于/etc/hosts，格式跟Windows大致相同，打开后我们把下面一行加到末尾

```
1.1.1.1      naizhao.com www.naizhao.com cnc.naizhao.com
```

如果已经按照前面的教程，添加过1.1.1.1这条记录的，修改为上面的样子。

2. 设置squid

squid的配置文件位于/usr/local/squid/etc/squid.conf，打开后会有英文的详细说明。暂时我们不需要去理解这个文件里面各个配置的含义，相信大家看到一大堆英文都会头大的。大家只需要知道如何把squid按照自己的需求跑起来，并且不出错就好。

我们先用rm命令把squid.conf删掉。不用怕删掉后找不回来，还有一个默认的squid.conf.default在呢。一般的linux程序，包括apache等，都会有一个.default的默认文件，方便大家恢复默认设置，或者作为一个修改的参考。

squid.conf删掉后，我们再用vi squid.conf，新建这个文件，然后把下面的内容帖进去，保存

```
http_port 80 vhost vport=80
acl apache rep_header Server ^Apache
broken_vary_encoding allow apache
cache_mem 1024 MB
maximum_object_size 4096 KB
minimum_object_size 0 KB
maximum_object_size_in_memory 256 KB
cache_dir ufs /var/spool/squid 20480 16 256
cache_swap_low 80
cache_swap_high 97
strip_query_terms off
request_body_max_size 5 MB
memory_pools on
memory_pools_limit 50 MB
access_log none
cache_log /var/log/squid/cache.log
```

```

cache_store_log none
pid_filename /var/run/squid.pid
hosts_file /etc/hosts
refresh_pattern ^ftp:      1440  20%  10080
refresh_pattern ^gopher:   1440  0%   1440
refresh_pattern -i naizhao\.com      240  100%  1440 ignore-reload
refresh_pattern -i www\.naizhao\.com  240  100%  1440 ignore-reload
refresh_pattern -i .gif 180 20% 10080 override-expire ignore-reload reload-into-ims
refresh_pattern -i .jpg 180 20% 10080 override-expire ignore-reload reload-into-ims
refresh_pattern .          120  50%   1440
acl all src 0.0.0.0/0.0.0.0
acl p-manager proto cache_object
acl s-localhost src 127.0.0.1/255.255.255.255
acl d-localhost dst 127.0.0.0/8
acl d-domains dstdomain .naizhao.com
acl p-ssl port 443 563
acl p-safe port 80 443 563
acl m-conn method CONNECT
acl m-purge method PURGE
acl n-maxconn maxconn 15
http_access allow p-manager s-localhost
http_access allow m-purge
http_access deny !p-safe
http_access deny m-conn !p-ssl
http_access deny n-maxconn
http_access allow p-manager
http_access allow d-domains
http_access deny all
http_reply_access allow all
acl r-url urlpath_regex realtime
cache_deny r-url
icp_access allow all
follow_x_forwarded_for allow all
acl_uses_indirect_client offrange_offset_limit -1
dns_timeout 2 seconds
forward_timeout 10 seconds
connect_timeout 10 seconds
peer_connect_timeout 6 seconds
read_timeout 10 seconds
request_timeout 6 seconds
persistent_request_timeout 16 seconds
cache_mgr silence.z@gmail.com
cache_effective_user nobody
cache_effective_group nobody
visible_hostname cnc.naizhao.com
logfile_rotate 0
always_direct allow all
error_directory /usr/local/squid/share/errors/Simplify_Chinese
coredump_dir /var/spool/squid

```

需要说明的几个参数:

cache_mem 1024 MB (squid使用的内存大小, 注意必须要留下500M左右的内存给系统使用, 建议系统内存2G, 然后分配1G给squid)

cache_dir ufs /var/spool/squid 20480 16 256 (缓存的目录, 放在/var/spool/squid下面, 大小是20G, 第一级目录是16个, 第二级目录是256个。因为一个目录下面放置的文件和目录数量有限制, 所以这里必须要设置多个目录, 以适合squid存放大量文件的需要。目录数千万别设置太大, 否则会引起系统死机)

memory_pools_limit 50 MB (内存池, 用来存放缓存文件的链表, 方便高速查找文件存放的对应位置, 不需要太大。不懂就别管了)

cache_log /var/log/squid/cache.log (log文件的存放位置, 必须要有读写权限, 稍后说到)

```
refresh_pattern -i naizhao\.com      240 100% 1440 ignore-reload
refresh_pattern -i www\.naizhao\.com  240 100% 1440 ignore-reload
refresh_pattern -i .gif 180 20% 10080 override-expire ignore-reload reload-into-ims
refresh_pattern -i .jpg 180 20% 10080 override-expire ignore-reload reload-into-ims
```

(上面的几行指定了文件缓存的时间, 可以指定某个域名, 也可以指定某个文件类型。有更加多的域名或者文件类型就按照需要添加吧。-i是忽略大小写, naizhao.com和.gif是文件类型, 240是最小缓存的时间, 1440是最大缓存时间, 都是分钟。这些先不用管, 有需要的时候调整就好了。我们的口号是: 先跑起来)

acl d-domains dstdomain .naizhao.com (允许访问的域名列表。如果要跑多个网站就添加多几个, 用空格分开。naizhao.com前面有个英文的., 表示所有*.naizhao.com都允许。如果没有这个点, 那么就只允许naizhao.com, 访问www.naizhao.com就会提示访问被拒绝。)

cache_mgr silence.z@gmail.com (管理员的邮件地址, 改为自己的)

```
cache_effective_user nobody
cache_effective_group nobody
```

(用来跑squid的用户和用户组。为了避免使用root帐号造成的不安全因素, 大部分linux程序都会先用root运行, 然后马上把运行的用户从root切换到其他用户, 这样就算程序有漏洞被黑了, 黑客也只能得到这个普通用户的权限, 对系统的危害将大大降低。保持默认即可)

visible_hostname cnc.naizhao.com (服务器输出的机器名字, 多台squid分布的时候这个很重要, 通过设置多个不同的名字, 很容易就可以找出哪台服务器有问题)

coredump_dir /var/spool/squid (squid挂掉后, 临终遗言要放到哪里。不用管, 一般人看不懂)

修改好配置文件后, 我们还需要做一些最后的工作。

先建一个log目录, 用来存放log文件。按照squid.conf的设置, 我们输入下面的命令

```
mkdir /var/log/squid
chown nobody:nobody /var/log/squid
```

chown命令是设置所有者和组。刚才说了, squid将会使用用户nobody组nobody来运行。nobody:nobody冒号前面的是用户, 后面的是用户组。

还有squid的缓存目录，同样需要设置权限为nobody:nobody

```
chown nobody:nobody /var/spool/squid
```

查看命令是否成功，可以使用ls -lh

```
ls -lh /var/spool/
```

```
.....  
drwxr-xr-x 3 nobody nobody 4.0K Jun 15 08:19 squid
```

剩下最后一步，初始化squid的缓存目录结构。（也就是刚才说的16个一级目录，256个二级目录。本身这些文件夹不存在，需要通过squid的命令建立）

squid的命令在/usr/local/squid/sbin下，我们进入这个目录，然后看看squid命令的帮助

```
./squid -h
```

```
Usage: squid [-hvzCDFNRYX] [-d level] [-s | -l facility] [-f config-file] [-u port] [-k signal]
```

```
-d level  Write debugging to stderr also.
```

```
-f file   Use given config-file instead of  
         /usr/local/squid/etc/squid.conf
```

```
-h       Print help message.
```

```
-k reconfigure|rotate|shutdown|interrupt|kill|debug|check|parse  
    Parse configuration file, then send signal to  
    running copy (except -k parse) and exit.
```

```
-s | -l facility  
    Enable logging to syslog.
```

```
-u port   Specify ICP port number (default: 3130), disable with 0.
```

```
-v       Print version.
```

```
-z       Create swap directories
```

```
-C       Do not catch fatal signals.
```

```
-D       Disable initial DNS tests.
```

```
-F       Don't serve any requests until store is rebuilt.
```

```
-N       No daemon mode.
```

```
-R       Do not set REUSEADDR on port.
```

```
-S       Double-check swap during rebuild.
```

```
-X       Force full debugging.
```

```
-Y       Only return UDP_HIT or UDP_MISS_NOFETCH during fast reload.
```

我们可以看到-z命令就是建立缓存目录的，所以我们输入

```
./squid -z
```

```
2007/07/07 07:52:03| Creating Swap Directories
```

没有任何错误提示，说明目录已经建立成功了，我们可以用ls /var/spool/squid看看效果。

最后，我们让squid跑起来

```
./squid -Nd 10
```

```
2007/07/07 07:53:50| Starting Squid Cache version 2.6.STABLE13 for i686-pc-linux-gnu...
2007/07/07 07:53:50| Process ID 3135
2007/07/07 07:53:50| With 1024 file descriptors available
2007/07/07 07:53:50| Using poll for the IO loop
2007/07/07 07:53:50| Performing DNS Tests...
2007/07/07 07:53:50| Successful DNS name lookup tests...
2007/07/07 07:53:50| DNS Socket created at 0.0.0.0, port 1026, FD 4
2007/07/07 07:53:50| Adding nameserver 211.147.6.3 from /etc/resolv.conf
2007/07/07 07:53:50| Adding nameserver 61.233.9.9 from /etc/resolv.conf
2007/07/07 07:53:50| Unlinkd pipe opened on FD 8
2007/07/07 07:53:50| Swap maxSize 20971520 KB, estimated 1613193 objects
2007/07/07 07:53:50| Target number of buckets: 80659
2007/07/07 07:53:50| Using 131072 Store buckets
2007/07/07 07:53:50| Max Mem size: 1048576 KB
2007/07/07 07:53:50| Max Swap size: 20971520 KB
2007/07/07 07:53:50| Store logging disabled
2007/07/07 07:53:50| Rebuilding storage in /var/spool/squid (DIRTY)
2007/07/07 07:53:50| Using Least Load store dir selection
2007/07/07 07:53:50| Set Current Directory to /var/spool/squid
2007/07/07 07:53:50| Loaded Icons.
2007/07/07 07:53:52| Accepting accelerated HTTP connections at 0.0.0.0, port 80, FD 7.
2007/07/07 07:53:52| Accepting ICP messages at 0.0.0.0, port 3130, FD 9.
2007/07/07 07:53:52| WCCP Disabled.
2007/07/07 07:53:52| Ready to serve requests.
2007/07/07 07:53:52| Done scanning /var/spool/squid (0 entries)
2007/07/07 07:53:52| Finished rebuilding storage from disk.
2007/07/07 07:53:52|      0 Entries scanned
2007/07/07 07:53:52|      0 Invalid entries.
2007/07/07 07:53:52|      0 With invalid flags.
2007/07/07 07:53:52|      0 Objects loaded.
2007/07/07 07:53:52|      0 Objects expired.
2007/07/07 07:53:52|      0 Objects cancelled.
2007/07/07 07:53:52|      0 Duplicate URLs purged.
2007/07/07 07:53:52|      0 Swapfile clashes avoided.
2007/07/07 07:53:52| Took 2.0 seconds ( 0.0 objects/sec).
2007/07/07 07:53:52| Beginning Validation Procedure
2007/07/07 07:53:52| Completed Validation Procedure
2007/07/07 07:53:52| Validated 0 Entries
2007/07/07 07:53:52| store_swap_size = 0k
2007/07/07 07:53:53| storeLateRelease: released 0 objects
```

打开浏览器，输入http://2.2.2.2，会看到Access Denied的错误提示，说明squid已经正常跑起来了。

因为指定了-d参数，所以现在squid会把所有的调试信息都输出终端。按下键盘的ctrl+c，然后输入killall squid杀死squid，可以终止squid的进程。

第 8 章 测试并运行Squid

1. 前期准备

想要测试Squid是否正常，必须要先把www.naizhao.com这个域名解析到2.2.2.2这个IP上。跟上一章一样，如果你是网通用户就不需要做任何操作，DNSPod会给你返回2.2.2.2这个IP。如果你是电信用户，或者解析出来的IP不是2.2.2.2，那么就要修改系统的hosts表，对应记录添加进去。跟上一章不一样，上一章修改的是Squid服务器的hosts表，你现在需要修改的是你用来测试Squid的系统的hosts表。

除了修改hosts表外，我们还需要一个名为curl的小工具。一般的Linux/Unix系统都附带了這個工具，如果使用Windows，那么需要到下面的地址下载一个，并且放到D:\（或者其他地方，自己喜欢）。

<http://curl.haxx.se/latest.cgi?curl=win32-nossl>（32位系统）

<http://curl.haxx.se/latest.cgi?curl=win64-nossl>（64位系统）

下载回来解压后，我们只需要一个curl.exe。

2. 运行Squid

我们继续用上一章的方法，用调试方式启动squid

```
cd /usr/local/squid/sbin
```

```
./squid -Nd 1
```

然后打开浏览器，输入<http://www.naizhao.com>，看输出的网页是否正确。

如果能显示出正常的页面，那么说明squid已经正常运行了。但我们还需要确认一下，所需要的页面是否已经被squid缓存了。

3. 测试Squid

我们使用curl，输入下面的命令（Windows用户请使用命令行来运行curl）

```
curl -I http://www.naizhao.com/
```

```
HTTP/1.0 200 OK
```

```
Date: Sun, 08 Jul 2007 12:16:27 GMT
```

```
Server: NaiZhao Web Server/1.0.0
```

```
Last-Modified: Wed, 13 Dec 2006 06:11:14 GMT
```

```
ETag: "7665b-1983-4417e480"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 6531
```

```
Vary: Accept-Encoding
```

```
Content-Type: text/css
```

```
Age: 29155
```

```
X-Cache: MISS from cnc.naizhao.com
```

```
Via: 1.0 cnc.naizhao.com:80 (squid/2.6.STABLE13)
```

```
Connection: close
```

我们留意输出的内容，下面一行内容表示并没有命中缓存。也就是并不是从缓存中读取所需文件。一般第一次访问一个文件的时候，出现MISS是正常的，因为本来文件就不存在于缓存中。

```
X-Cache: MISS from cnc.naizhao.com
```

然后，我们重新运行一次命令，继续看输出的内容

```
curl -I http://www.naizhao.com/
```

```
HTTP/1.0 200 OK
Date: Sun, 08 Jul 2007 12:16:27 GMT
Server: NaiZhao Web Server/1.0.0
Last-Modified: Wed, 13 Dec 2006 06:11:14 GMT
ETag: "7665b-1983-4417e480"
Accept-Ranges: bytes
Content-Length: 6531
Vary: Accept-Encoding
Content-Type: text/css
Age: 29155
X-Cache: HIT from cnc.naizhao.com
Via: 1.0 cnc.naizhao.com:80 (squid/2.6.STABLE13)
Connection: close
```

可以看到，这次的输出已经是HIT了。也就是表示，内容是从缓存中读取的。

如果第二次访问仍然是MISS，那么你就需要检查下网页是否限制了缓存。可以从下面几个方面入手：

Last-Modified: Wed, 13 Dec 2006 06:11:14 GMT 检查这行是否每次访问都变化。如果一直变化，说明页面是动态的，并不合适squid缓存。squid在每次请求的时候都会检查这行，判断页面是否已经过期，并且对过期的页面重新获取。

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0 这行是负责控制缓存。no-store、no-cache、must-revalidate等都是要求squid禁止缓存内容，并且在每次请求的时候都去验证页面是否过期。

Expires: Thu, 19 Nov 1981 08:52:00 GMT 过期时间如果大于当前时间，也会被认为是页面已经过期，需要重新获取。这样的页面也不会被缓存。

Pragma: no-cache 这句也是禁止缓存。

大家可以按照上面的内容进行自查自纠，根据实际需求对网站进行修改即可。确认没有问题后，我们就可以让Squid正式跑起来了。

4. 正式运行Squid

```
ulimit -Hs 65536
```

```
ulimit -n 65536
```

```
./squid
```

直接使用上面的命令就可以让squid跑起来了。

跑起来后，输入<http://www.naizhao.com>看看访问是否正常。

5. 让Squid跟随系统启动

CentOS让Squid跟随系统启动很简单，只需要编辑/etc/rc.local，在最后加上下面内容即可

```
ulimit -Hs 65536
```

```
ulimit -n 65536
```

```
/usr/local/squid/sbin/squid
```