

Squid 中文权威指南

(第 2 章)

译者序:

本人在工作中维护着数台 Squid 服务器, 多次参阅 Duane Wessels (他也是 Squid 的创始人) 的这本书, 原书名是 "Squid: The Definitive Guide", 由 O'Reilly 出版。我在业余时间把它翻译成中文, 希望对中文 Squid 用户有所帮助。对普通的单位上网用户, Squid 可充当代理服务器; 而对 Sina, NetEase 这样的大型站点, Squid 又充当 WEB 加速器。这两个角色它都扮演得异常优秀。窗外繁星点点, 开源的世界亦如这星空般美丽, 而 Squid 是其中耀眼的一颗星。

对本译版有任何问题, 请跟我联系, 我的Email是: yonghua_peng@yahoo.com.cn

彭勇华

目 录

第 2 章 获取Squid	2
2.1 版本和发布	2
2.2 使用源代码	2
2.3 预编译的二进制文件	3
2.4 匿名CVS	3
2.5 devel.squid-cache.org	4

第 2 章 获取 Squid

2.1 版本和发布

Squid 开发者定期发布源代码。每一个发布版有一个版本号，例如 2.5.STABLE4。版本号的第三部分以 STABLE 或 DEVEL（短期开发版本）开头。

也许你能猜到，DEVEL 版本倾向于拥有更新，更试验性的功能。但也许它们有更多的 bugs。无经验的用户不应该运行 DEVEL 版本。假如你选择运行一个 DEVEL 版本，并且遇到了问题，请将问题报告给 Squid 维护者。

在一段时间的开发期后，Squid 版本号变为 STABLE。该版本适合于普通用户。当然，即使稳定版可能也有一些 bugs。高的稳定版本（例如 STABLE3,STABLE4）应该 bugs 更少。假如你特别关心稳定性，你应该使用这些最近发布版本中的一个。

2.2 使用源代码

为什么你不能 copy 一份预编译的二进制代码到你的系统中，并且期望它运行良好呢？主要理由是 squid 的代码需要知道特定操作系统的参数。实际上，最重要的参数是打开文件描述符的最大数量。Squid 的 ./configure 脚本在编译之前侦察这些值。假如你获取一个已编译的使用某个参数值的 squid 到另一个使用不同参数值的系统中，可能会遇到问题。

另一个理由是许多 squid 功能在编译时必须被激活。假如你获取一个别人已编译的 squid 文件，它不包含你所需要的功能，那么你又得再编译一遍。

最后，共享库的问题可能使得在系统之间共享可执行文件困难。共享库在运行时被装载，如已知的动态链接一样。squid 在编译时会侦察你系统中的 C 库的某些功能（例如它们是否被提供，是否能运行等）。尽管库功能不常改变，但两个不同的系统的 C 库之间可能有明显的区别。如果两个系统差别太大，就会对 Squid 造成问题。

获取 squid 的源代码是很容易的。请访问 squid 的首页：<http://www.squid-cache.org>。首页有链接指向不同的稳定版和开发版。假如你不在美国，那么请访问 squid 的众多镜像站点中的一个。镜像站点通常以 "wwwN.CC.squid-cache.org" 命名，N 是数字，CC 是国家的两位代码。例如，www1.au.squid-cache.org 是澳大利亚的镜像站点，在主页上有链接指向不同的镜像站点。

每一个 squid 发布版分支（例如 Squid-2.5）有它自己的 HTML 页面。该页面有链接指向源代码，以及与其他发布版的差别。假如你从一个发布版升级到下一个，你应该下载这些差别文件，并且打上补丁，请见 3.7 章节中的描述。每个版本的发布页描述新功能和重要的改进，也有链接指向已经修正的 bugs。

如果 web 访问不可行，你能从 <ftp://ftp.squid-cache.org> 的 FTP 服务器获取源代码，或者使用其他 FTP 镜像。要获取当前版本，请访问 [pub/squid-2/DEVEL](ftp://pub/squid-2/DEVEL) 或 [pub/squid-2/STABLE](ftp://pub/squid-2/STABLE) 目录。FTP 镜像也在许多国家有，你能用同样的国家代码去猜测一些 FTP 镜像站点，例如 [ftp1.uk.squid-cache.org](ftp://ftp1.uk.squid-cache.org)。

当前的 Squid 发布版本大约 1M 大小。在下载完压缩的打包文件后，你能继续第 3 章。

2.3 预编译的二进制文件

一些 Unix 发布版可能预包含了 Squid 的编译版。对 Linux 系统，你可以找到 Squid 的 RPM 包。通常 squid RPM 包含在你所买的 Linux 光碟里。Freebsd/Netbsd/OpenBSD 也在它们的 ports 或者 packages 里面包含了 squid。

虽然 RPM 或者预编译的 packages 能节省你一些时间，但它们也有一些弊端。就像我提过的一样，在你开始编译 squid 之前，某些功能必须被激活或禁止。而你安装的预编译的包可能不包含你想要的特定功能。而且，squid 的 ./configure 脚本侦察你系统中的特定参数，这些在你系统中的参数可能与编译它的机器的参数不同。

最后，假如你想对 squid 打补丁，你必须等某个人编译更新的 RPM 或 packages，或者你还得自己找源代码编译。

我强烈建议你从源代码编译 squid，当然怎样选择由得你。

2.4 匿名 CVS

你能匿名访问 squid 的 CVS 文件（只读）以保持你的源代码同步更新。使用 CVS 的有利面是你能够轻易获取当前运行版本的补丁。这样就容易发现近来改变了什么。

将这些补丁打到你所运行的版本中，有效的保持你的源代码和官方版本的同步。

CVS 使用树型索引系统，树干叫做头分支。对 Squid 而言，这里也是所有的新改变和新功能的存放之地。头分支通常包含试验性的，也许不太稳定的代码。稳定的代码通常在其他分支上。

为了有效的使用 squid 的匿名 CVS，你首先应知道版本和分支是怎样被标明不同的。例如，版本 2.5 分支被命名为 SQUID_2_5。具体的发布有长的命名，例如 SQUID_2_5_STABLE4。为了得到 squid 版本 2.5.STABLE4，请使用 SQUID_2_5_STABLE4 标签；使用 SQUID_2_5 得到最近的 2.5 分支的代码。

为了使用 squid 匿名 CVS 服务，你首先必须设置 CVSROOT 环境变量：

```
csh% setenv CVSROOT :pserver:anoncv@cv.squid-cache.org:/squid
```

或者，对 Bourne shell 用户：

```
sh$ CVSROOT=:pserver:anoncv@cv.squid-cache.org:/squid
```

```
sh$ export CVSROOT
```

然后你就可以登陆到服务器：

```
% cvs login
```

```
(Logging in to anoncv@cv.squid-cache.org)
```

```
CVS password:
```

在提示符下，敲入 anoncv 作为密码。现在你可以用这个命令检查源代码树：

```
% cvs checkout -r SQUID_2_5 -d squid-2.5 squid
```

-r 选项指定获取修订标签。省略-r 选项你将获得头分支。-d 选项改变存放文件的顶级目录名。假如你省略-d 选项，顶级目录名就与模块名字一样。最后的命令行参数（squid）是要检查的模块名字。

一旦你检查完 squid 源代码树，你能运行 cvs update 命令去升级你的文件，和保持文件同步。其他命令包括：cvs diff, cvs log, 和 cvs annotate。

想获取更多 CVS 知识，请访问：<http://www.cvshome.org>

2.5 devel.squid-cache.org

Squid 的开发者维持一个独立的站点，当前运行在 SourceForge，提供了试验性的 squid 功能。请检查它们在 <http://devel.squid-cache.org>。在这里你能发现许多正在开发的工程，它们还未集成到 squid 的官方源代码里。你能通过 SourceForge 的匿名 CVS 服务来访问这些工程，或者下载与标准版本不同的差别文件。