第 3 章　Squid 代理服务

代理服务器是在 Web 浏览器（如 Internet Explorer）和 Internet 之间起媒介作用的计算机。Squid 作为应用层代理服务软件，主要提供缓存加速、应用层过滤控制的功能。本章将介绍有关代理服务器的基本信息及 Squid 的安装、运行与配置等内容。

3.1　基 本 信 息

在安装 Squid 服务器之前，需要了解搭建该服务的一个环境。在实际应用中，安装一个服务的系统在硬件和软件方面都有些要求。下面介绍 Squid 的基本信息，包括网卡配置、软件包、进程、端口等内容。

3.1.1　网卡配置文件：/etc/sysconfig/network-scripts/ifcfg-XXX

下面为安装 Squid 服务的主机设置一个固定的 IP 地址为 192.168.1.1。

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
HWADDR=00:0C:29:88:77:96
IPADDR=192.168.1.1
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
```

3.1.2　软件包：squid

下面以表格的形式列出了 RedHat Linux 中 Squid 服务的 squid 软件包位置及源码包下载地址，如表 3.1 所示。

表 3.1　软件包位置

软件包类型	位　　置
RHEL6 RPM	光盘：/Packages
RHEL5 RPM	光盘：/Server
源码包	http://www.squid-cache.org/

本章讲解安装 Squid 软件的方法，适合 REHL5.X～6.4 的所有版本。不同版本的软件包名，如表 3.2 所示。

表 3.2　不同发行版本的软件包

RHEL 6.4	squid-3.1.10-16.el6.i686.rpm
RHEL 6.3	squid-3.1.10-1.el6_2.4.i686.rpm
RHEL 6.2	squid-3.1.10-1.el6_1.1.i686.rpm
RHEL 6.1	squid-3.1.10-1.el6.i686.rpm
RHEL 6.0	squid-3.1.4-1.el6.i686.rpm
RHEL 5	squid-2.6.STABLE21-6.el5.i386.rpm

3.1.3　进程名：squid

Squid 服务启动后，会自动运行一个名为 squid 的进程。可使用以下命令查看：

```
[root@localhost ~]# ps -eaf | grep squid
root      8981     1  0 17:30 ?    00:00:00 squid -f /etc/squid/squid.conf
squid     8983  8981  0 17:30 ?    00:00:00 (squid) -f /etc/squid/squid.conf
squid     8985  8983  0 17:30 ?       00:00:00 (unlinkd)
root      8990  7183  0 17:30 pts/0    00:00:00 grep squid
```

3.1.4　端口：3128

Squid 服务运行后，默认监听 3128 号端口。可以使用以下命令查看：

```
[root@localhost ~]# netstat -antp | grep squid
tcp        0      0 :::3128                 :::*                    LISTEN      8983/(squid)
```

3.1.5　防火墙所开放的端口号：3128

当 Squid 服务搭建成功后，需要使用客户端测试验证服务器是否正常。在现实使用情况下，防火墙都是开启的，此时有可能不允许客户端访问。所以需要在防火墙中开放 Squid 服务的 3128 端口允许客户端进行访问。

```
iptables -I INPUT -p tcp --dport 3128 -j ACCEPT
```

3.2　构建 Squid 服务

代理服务的种类非常多，如果按所支持的协议来分，可以分为 HTTP 代理、FTP 代理、SSL 代理、POP3 代理、SOCKS 代理等。其中，HTTP 代理（也称为 Web 代理）的应用最广泛，本节主要以 HTTP 代理为例，介绍代理服务的运行机制、安装、配置等内容。

3.2.1　运行机制

Squid 服务具体是如何工作的，下面分别介绍代理服务器的作用、构成、工作流程 3

部分内容。

1．代理服务器的作用

代理服务器一般构建在内部网络和 Internet 之间，负责转发内网计算机对 Internet 的访问，并对转发请求进行控制和登记。代理服务器作为连接 Intranet（局域网）与 Internet（广域网）的桥梁，在实际应用中有着重要的作用。利用代理，除了可以实现最基本的连接功能外，还可以实现安全保护、缓存数据、内容过滤和访问控制等功能。

2．代理服务器构成

多台客户机通过内网与 Web 代理服务器连接。Web 代理服务器除了与内网连接外，还有一个网络接口与外网连接。代理服务器的架构如图 3.1 所示。

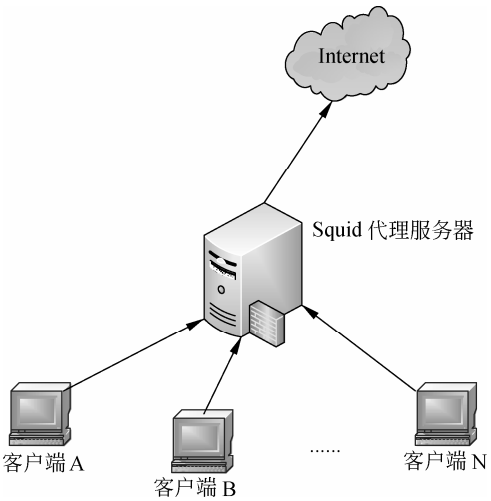

图 3.1　Squid 代理服务器的构成

在图 3.1 中，Squid 代理服务器是客户端与 Internet 网进行连接的桥梁。Squid 代理服务器简单分为 3 部分，即客户端、代理服务器、Internet。客户端通过向代理服务器发送请求与 Internet 网建立连接。

3．代理服务器的工作流程

代理服务器的工作流程如图 3.2 所示。当客户端访问 Internet 网中的 Web 服务器时，客户端首先向代理服务器发送 HTTP 请求。如果发现所请求的数据在缓存中已经存在，则直接把这些数据发送给客户端。如果代理服务器在缓存中找不到所请求的数据，则会转发这个 HTTP 请求到客户端要访问的 Web 服务器。Web 服务器响应后，把数据发给了代理服务器，代理服务器再把 Web 服务器响应的数据转交给客户端，同时把这些数据存储在缓存中。于是，下次客户端再次请求同样的数据时，代理服务器就直接用缓存中的数据进行响应，而不需要再次向 Web 服务器请求数据。这样也提高了数据传输的速度。

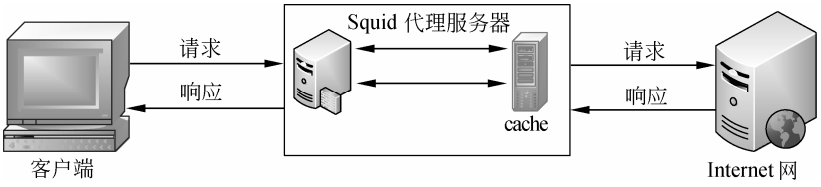

图 3.2　代理服务器工作流程图

当然，如果客户端每一次请求的数据在代理服务器的缓存中都没有，都需要通过代理服务器向 Internet 上的 Web 服务器请求，则比客户端自己直接请求时的速度要慢。但由于能加快后续访问的速度，因此，从整体来说，速度的提高还是很明显的。

3.2.2　搭建服务

默认情况下，在安装 RHEL 时并未安装 Squid 服务器程序，在使用 Squid 服务前，首先必须将 Squid 服务器程序安装到系统中。为了方便安装，本节使用光盘中自带的 RPM 包安装 Squid 代理服务。具体操作步骤如下：

（1）使用如下命令查看 squid 软件是否安装。

```
[root@localhost ~]# rpm -q squid
```

输出信息如下：

```
package squid is not installed
```

输出的信息说明软件包 squid 没有安装。

（2）挂载 RHEL 6.4 系统光盘，并安装其中的 squid-3.1.10-16.el6.i686.rpm 软件包。

```
[root@localhost ~]# mount /dev/cdrom /mnt/cdrom/
mount: block device /dev/sr0 is write-protected, mounting read-only
[root@localhost ~]# cd /mnt/cdrom/Packages/
[root@www Packages]# rpm -ivh squid-3.1.10-16.el6.i686.rpm
warning: squid-3.1.10-16.el6.i686.rpm: Header V3 RSA/SHA256 Signature, key
ID fd431d51: NOKEY
Preparing...          ########################################### [100%]
   1:squid            ########################################### [100%]
```

3.3　文 件 组 成

当某一个服务安装后，会自动创建一些目录和文件。下面来看一下 Squid 代理服务所创建的文件，如表 3.3 所示。

表 3.3　Squid代理服务中的文件

目　　录	文 件 名	文 件 类 型	功 能 说 明
/etc/httpd/conf.d/	squid.conf	配置文件	和 Web 的代理捆绑在一起
/etc/logrotate.d	squid	配置文件	日志转储文件
/etc/pam.d	squid	配置文件	PAM 认证
/etc/rc.d/init.d	squid	可执行文件	代理服务的守护进程
/etc/squid	cachemgr.conf	配置文件	监控代理服务器
	errorpage.css	配置文件	代理服务错误页面的样式表改编自免费 CSS 模板设计
	mime.conf	配置文件	定义 MIME-TYPE 文件
	msntauth.conf	配置文件	MSNT 认证的配置文件
	squid.conf	配置文件	代理服务的主配置文件
/etc/sysconfig/	squid	配置文件	命令参数配置文件
/usr/bin	squidclient	可执行文件	一个简单的 HTTP Web 客户端
/usr/sbin/	squid	可执行文件	代理缓存服务器

续表

目　　录	文　件　名	文 件 类 型	功 能 说 明
/var/log/squid	access.log	普通文件	默认访问日志文件
	cache.log	普通文件	缓存日志文件
	squid.out	普通文件	记录配置文件中出现的问题

在表 3.3 中列出了该服务所有的文件，下面以图的形式表示出这些文件的工作流程。代理服务工作过程中发挥作用的文件流程如图 3.3 所示。

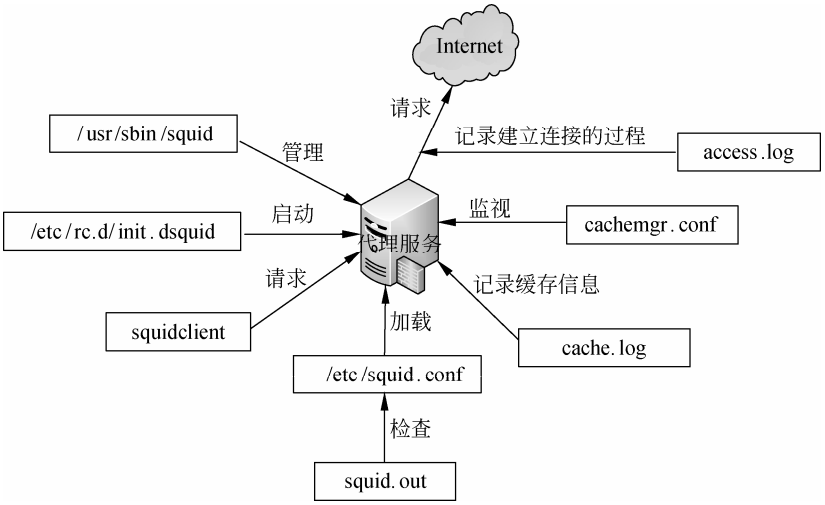

图 3.3　Squid 服务流程图

3.4　配置文件：/etc/squid/squid.conf

Squid 服务默认的配置文件位于/etc/squid/squid.conf。该文件中包含大量的注释内容，为相关的配置项提供了详尽的解释和说明。下面仅对最常用到的一些配置项做出解释。充分了解这些配置项的作用，将有助于用户在实际生活中灵活应用配置代理服务。

3.4.1　访问控制列表选项：acl

访问控制列表（Access Control List）简称 ACL。这些列表中包含了一定的过滤和控制条件，然后只要针对这些列表设置是 allow（允许）或 deny（拒绝）就可以实现访问控制了。

在 squid.conf 配置文件中，HTTP 的访问控制主要由 acl 和 http_access 配置项共同实现，两个配置项分别用来定义控制的条件（列表）和实施控制。

acl 配置项用于设置访问控制列表的内容，可以为每组特定的控制目标指定一个名称。每一行 acl 配置定义一个访问控制列表，格式如下。

```
acl 列表名称 列表类型 列表内容 ...
```

其中，“列表名称”为用户自行指定，“列表类型”必须使用 Squid 预定义的值，“列表内容”即控制的具体对象，根据对应的列表类型进行设置。每个列表类型的内容可以包含多个值，各个值之间使用“或”的关系，只要满足其中任何一个值就可以匹配成功。

Squid 预定义的列表类型有很多种，常用的包括源地址、目标地址、访问时间、访问端口等，如表 3.4 所示。

表 3.4　常用的几种acl列表类型

列 表 类 型	列表内容示例	含义/用途
src	192.168.1.168/32 192.168.1.0/255.255.255.0 192.168.1.0-192.168.3.0/24	客户端的 IP 地址或网络段、地址范围
dst	www.playboy.com 216.163.137.3/32	用户访问的目标主机名或者 IP 地址
port	80 8000 8080 21	用户访问的目标端口
srcdomain	.benet.com .accp.com	客户端来源域（根据 IP 地址作反向解析）
dstdomain	.qq.com .msn.com verycd.com	用户访问的目标域，匹配域内所有站点
time	MTWHF 8:30-17.30 12:00-1300 AS	用户上网的时间段 字母表示一星期中各天的英文缩写 M-Monday、T-Tuesday W-Wednesday、H-Thursday、F-Friday A-Saturday、S-Sunday
Maxconn	15	客户端的并发 HTTP 连接数
url_regex	url_regex -i ^rtsp://^mms:// url_regex -i ^emule://	用户访问的整个 URL 网址，可以使用正则表达式，加-i 表示忽略大小写
urlpath_regex	urlpath_regex -i sex adult nude urlpath_regex -i \.mp3$\.rar$	匹配用户访问的 URL 路径（部分），可以使用正则表达式

（1）定义源地址类型的访问控制列表。

```
acl all src 0.0.0.0/0.0.0.0
acl localhost src 127.0.0.1/255.255.255.255
acl LAN1 src 192.168.1.0/24
acl LAN2 src 192.168.2.0/24
acl PC1 src 192.168.1.66/32          //定义单个 IP 地址
```

（2）定义来源域类型的访问控制列表。

```
acl lan_Domain srcdomain .benet.com .accp.com
```

（3）定义目标地址类型的访问控制列表。

```
acl to_localhost dst 127.0.0./8
acl Black_IP dst 61.143.79.86/32 21723.45.77/32
```

（4）定义目标域类型的访问控制列表。

```
acl Black_HOST www.xxxx.com www.adult.com //squid 在启动时会尝试解析为 IP 地址
acl Black_Domain dstdomain .qq.com .msn.com .gamezone.net
```

（5）定义 HTTP 并发连接数限制类型的访问控制列表。

```
acl Max10_Conn maxconn 10
acl Max20_Conn maxconn 20
```

（6）定义按正则表达式定义用户请求访问的 Web 对象类型的访问控制列表。

```
acl Black_URL url_regex -i ^rtsp://^mms://^emule://
acl Illegal_Words urlpath_regex -i sex adult fake
acl MediaFile urlpath_regex -i \.mp3$ \.mp4$ \.rmvb$ \.rm$ \.mov$ \.mpg$
```

（7）定义用户时间类型的访问控制列表。

```
acl Lunch_Hours time MTWHF 12:00-13:00
acl Work_Hours time MTWHF 08:30-17:30
```

3.4.2　设置 acl 访问权限：http_access

针对各种 acl 列表，使用 http_access 配置项控制其访问权限，允许（allow）或者拒绝（deny）。http_access 配置行必须在对应的 acl 定义之后。每一行 http_access 配置确定一条权限控制规则，格式如下：

```
http_access allow | deny [!]aclname......
```

一般格式如下：

```
http_access allow 或 deny 列表名......
```

在每一条 http_access 规则中，可以同时包含多个 acl 列表名，各个列表之间使用“与”的关系，只有满足所有 acl 列表对应的条件才会进行限制。在 acl 列表前可以添加“！”符号设置相反的条件。

在 squid.conf 文件中，将按照 http_access 各条规则的顺序进行扫描，如果找到一条相匹配的规则就不再向后搜索。因此，访问控制规则的顺序非常重要。以下两种默认情况需要注意。

- 没有设置任何规则时：Squid 服务将拒绝客户端的请求。
- 有规则但找不到相匹配的项：Squid 将采用与最后一条规则相反的权限，即如果最后一条规则是 allow，那么就拒绝客户端的请求，否则允许该请求。

通常情况下，把最常用到的控制规则放在最前面，以减少 Squid 的负载。在访问控制的总体策略上，建议使用“先拒绝后允许”或“先允许后拒绝”方式，在最后添加一条“http_access allow all”或者“http_access deny all”。

在模板配置文件中已经附带了一些配置并且有相应的解释。用户可以参考这些例子设置需要的访问权限。

```
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
```

```
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost

# And finally deny all other access to this proxy
http_access deny all
```

3.4.3　设置代理服务监听的地址和端口：http_port

http_port 是配置代理服务器最重要的一个选项。它决定了服务器使用哪种代理方式。该选项的设置格式有以下 3 种：

```
http_port 端口号
http_port IP 地址:端口号 transparent
http_prot IP 地址:端口号 vhost
```

其中，这 3 种格式的含义如下。

- http_port 端口号：用来设置支持基本代理功能。默认监听的端口号为 3128。
- http_port IP 地址：端口号 transparent：用来设置支持透明代理功能。这里的地址指的是代理服务器的地址。
- http_prot IP 地址：端口号 vhost：用来设置支持反向代理功能。这里的地址指的是内网中服务器的地址。

3.4.4　指定可见的主机名：visible_hostname

在设置代理服务器时，必须要配置该选项，否则会报错“WARNING: Could not determine this machines public hostname. Please configure one or set 'visible_hostname'.”。该选项的格式如下：

```
visible_hostname 主机名
```

3.4.5　对邻居的请求限制：hierarchy_stoplist

许多使用缓冲堆叠的用户，想控制或限制 Squid 发送到邻居缓冲的请求。这时就可以设置该选项。配置文件中默认的该列表是：

```
hierarchy_stoplist cgi-bin ?
```

这样，任何包含问号或 cgi-bin 字符串的请求匹配该列表，变成不可层叠。默认 Squid

直接发送不可层叠的请求到原始服务器。因为这些请求不会导致无法连接到网络，它们通常是邻居的缓冲来承担责任的。

3.4.6　设置缓冲数据时使用的目录参数：cache_dir

配置文件中默认的配置如下：

```
cache_dir ufs /var/spool/squid 100 16 256
```

这条配置选项中 cache_dir 是关键字，其他都是参数。其中，UFS（UNIX File System，UNIX 文件系统）是 Squid 最早使用的缓存文件的格式，也是 Squid 内建的存储格式类型；/var/spool/squid 是缓存数据的默认存放目录；后面 3 个数字依次表示该缓存目录可以使用的磁盘空间大小（单位为 MB）、一级子目录个数、二级子目录个数。如果代理用户数量较多，可以适当增大缓存目录的大小。

按此行配置初始化后的 Squid，将会在/var/spool/squid/目录下创建 16 个一级子目录（名称为 00、01、02、.....、0F），在每个一级子目录下创建 256 个二级子目录（名称为 00、01、02、.....、F0、F1、F2、.....、FF）。缓存下来的文件数据将保存到上述目录中。

在应用环境中可以根据实际情况适当扩大缓存目录的容量和子目录数。

- **access_log** /var/log/squid/access.log squid：指定日志文件的保存位置和记录格式（squid），该日志文件用于记录有哪些客户端通过代理访问过哪些 Web 对象等信息。
- **visible_hostname** porxy.benet.com：设置代理服务器可用的完整主机名，在 Squid 初始化或者启动服务时可能会用到。
- **dns_testnames** www.google.com www.sina.com.cn www.163.com：为了确保能够正常提供 Web 代理服务，squid 服务在启动时，可以通过该项设置测试 DNS 解析工作是否存在障碍。按从左到右的顺序，只要成功解析出一个域名，就不再测试后面的其他域名。如果管理员确认 DNS 解析没问题或者不需要 DNS 解析，建议注释掉该项配置，以加快服务初始化的速度。

3.4.7　定义 dump 的目录：coredump_dir

配置文件中默认的配置如下：

```
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
```

该选项定义了 dump 的目录为/var/spool/squid。

3.4.8　间接地控制磁盘缓存：refresh_pattern

refresh_pattern 用于确定一个页面进入 cache 后，在 cache 中保存的时间。refresh_pattern 规则仅仅应用到没有明确时间限制的响应。它的语法格式如下：

```
refresh_pattern [-i] regexp min percent max [options]
```

该语法中各选项含义如下。

- regexp：区分正则表达式的大小写。如果使用“-i”选项可以不用区分大小写。
- min：设置过时响应的最低时间限制。如果某个响应驻留在缓冲里的时间没有超过这个最低限制，那么它不会过期。
- percent：在最低和最高时间限制之间的响应。它是使用（LM-factor）算法计算过期时间的。
- max：设置存活响应的最高时间限制。如果某个响应驻留在缓冲里的时间高于这个最高限制，那么它必须被刷新。

在主配置文件 squid.conf 中，默认的有如下配置：

```
# Add any of your own refresh_pattern entries above these.
refresh_pattern ^ftp:          1440    20%     10080
refresh_pattern ^gopher:       1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .              0       20%     4320
```

3.4.9　设置缓冲功能的内存空间：cache_mem

设置用于缓存功能的内存空间大小，可以使用 MB 作为单位，主要用于保持访问较频繁的 Web 对象。一般来说将这个参数设置为物理内存的 1/4～1/3 比较合适，具体视服务器的实际性能和负载而定。下面看一个配置例子。

```
cache_mem 64MB
```

3.4.10　设置保存到高速缓冲的容量：maximum_object_size

允许保存到高速缓存的最大对象（文件）大小，可以使用 KB 作为单位。超过指定容量的文件将不会被缓存，而是直接转发给用户。如果需要对音频、视频等较大的文件进行缓存，可以适当增加该参数的值。下面看一个配置例子。

```
maximum_object_size 4096 KB
```

3.5　日 志 文 件

Squid 服务运行后，会自动创建 access.log、cache.log、squid.out 这 3 个日志文件。这 3 个文件分别保存了代理服务 3 种情况下的日志信息。下面介绍这 3 个配置文件。

3.5.1　访问日志文件：/var/log/squid/access.log

该文件记录了 Squid 的访问日志信息。如果想查看客户端是否通过 Squid 代理服务进行了网络连接，在该日子文件中有详细的记录。下面看一段通过 Squid 访问日志记录的信息：

```
1369906537.667              4   192.168.2.100   TCP_MISS/403   4308   GET
http://192.168.1.100/ - DIRECT/192.168.1.100 text/html
1369906537.722             18   192.168.2.100   TCP_MISS/200   2177   GET
http://192.168.1.100/icons/apache_pb2.gif - DIRECT/192.168.1.100 image/gif
1369906537.732              1   192.168.2.100   TCP_MISS/404   594   GET
http://192.168.1.100/favicon.ico - DIRECT/192.168.1.100 text/html
1369906540.734              2   192.168.2.100   TCP_MISS/404   594   GET
http://192.168.1.100/favicon.ico - DIRECT/192.168.1.100 text/html
```

日志信息描述了具体哪个客户端通过代理服务器访问到一台 Web 服务器的记录。如 192.168.2.100 是客户端的地址，192.168.1.100 是 Web 服务器的地址。

3.5.2　缓存日志文件：/var/log/squid/cache.log

该文件中记录了所有高速缓存的行为。该文件就相当于一个临时存储器，可以提高用户访问信息的速度。缓存文件可以定期清理，不影响系统使用。下面看一段记录的信息：

```
2013/07/15 16:37:02| storeDirWriteCleanLogs: Starting...
2013/07/15 16:37:02|   Finished.  Wrote 0 entries.
2013/07/15 16:37:02|   Took 0.00 seconds (  0.00 entries/sec).
2013/07/15 16:37:02| logfileRotate: /var/log/squid/access.log
```

3.5.3　Squid 的配置出现的问题文件：/var/log/squid/squid.out

该文件记录了代理服务的配置文件 squid.conf 的问题，如果该配置文件有错误配置项会在该文件中记录下来。下面是一段错误记录信息：

```
squid: ERROR: No running copy
2013/06/20 15:56:11| WARNING: No units on 'cache_mem 64MB', assuming 64.00 bytes
2013/06/20 15:56:11| WARNING: No units on 'maximum_object_size 4096KB',
assuming 4096.00 bytes
```

3.5.4　日志转储参数：/etc/logrotate.d/squid

squid 文件的主要作用就是告诉 logrotate 读入存放在/etc/logrotate.d 目录中的日志转储参数，方便用户集中管理。该文件不需要做任何修改，采用默认配置就可以。

3.6　可执行文件

Squid 服务的可执行文件是用来控制 squid 进程和端口的。当该文件运行时，squid 进程和端口也就会被监听。下面来看下这些脚本文件。

3.6.1　可执行程序文件：/usr/sbin/squid

Squid 服务安装成功后，在/usr/sbin 目录下有一个可执行程序文件 squid。它可以用来

启动 Squid 服务，语法格式如下：

```
squid [选项]
```

常用选项含义如下。

- -a port：指定新的 http_port 值。该选项覆盖了来自 squid.conf 的值。但是，在 squid.conf 中能指定多个值，-a 选项仅仅覆盖配置文件里的第一个值。
- -d level：让 squid 将它的调试信息写到标准错误中。
- -f file：指定另一个配置文件。
- -F：让 squid 拒绝所有的请求，直到它重新建立起存储元数据。
- -h：显示用法。
- -i：安装为 Windows 服务。
- -k function：指示 squid 执行不同的管理功能。功能参数有 reconfigure、rotate、shutdown、interrupt、kill、debug、check，parse。其中，reconfigure 表示重新加载配置文件；rotate 表示关闭日志、重命名和再次打开日志；shutdown 表示关闭 squid 进程；interrupt 表示立刻关闭 squid，不必等待活动会话完成；kill 表示发送 KILL 信号给 squid，这是关闭 squid 的最后保证；debug 表示将 squid 设置成完全的调试模式；check 表示简单的检查运行中的 quid 进程，返回的值显示 squid 是否在运行；parse 表示简单的解析 squid.conf 文件，如果配置文件包含错误，进程返回非零值。
- -N：阻止 squid 变成后台服务进程。
- -R：阻止 squid 在绑定 HTTP 端口之前使用 SO_REUSEADDR 选项。
- -s：将日志记录到 syslog 进程。
- -v：显示版本信息。
- -V：激活虚拟主机加速模式。
- -X：强迫使用完整调试模式。
- -z：初始化缓存目录。

【实例 3-1】使用 squid 命令启动代理服务。执行命令如下：

```
[root@www ~]# squid -z
 [root@www ~]# squid -k reconfigure
```

3.6.2　控制服务文件：/etc/rc.d/init.d/squid

Squid 文件通过使用 service 命令的 start、stop、restart 参数来启动、关闭、重启 Squid 代理服务。该文件也可以使用它的绝对路径带 start、stop、restart 参数来控制服务的运行。启动 Squid 服务的命令如下：

```
[root@www ~]# service squid start
正在启动 squid:                                           [确定]
```

或者

```
[root@www ~]# /etc/init.d/squid start
正在启动 squid:                                           [确定]
```

3.7　其他配置文件

在前面以分类的形式介绍了相关配置文件的作用及内容。该服务还有其他几个文件，下面详细介绍与代理服务有关的文件。

3.7.1　命令参数配置文件：/etc/sysconfig/squid

该文件中记录了 Squid 服务主配置文件的位置、关闭服务的等待时间等信息。下面查看该文件默认的内容。

```
[root@www ~]# vi /etc/sysconfig/squid
# default squid options
SQUID_OPTS=""

# Time to wait for Squid to shut down when asked. Should not be necessary
# most of the time.
SQUID_SHUTDOWN_TIMEOUT=100

# default squid conf file
SQUID_CONF="/etc/squid/squid.conf"
```

其中，以“#”开头的内容为注释信息，也是对下面信息的解释。

3.7.2　PAM 认证文件：/etc/pam.d/squid

PAM 认证是 Linux 服务器系统最主要的安全认证模式，掌握 PAM 认证对于加强系统安全非常重要。在代理服务器中该配置文件不需要修改，默认即可。

3.7.3　监视性能文件：/etc/squid/cachemgr.conf

该文件通过管理 cachemgr.cgi 脚本监视 Squid 服务的性能。该文件不需要做任何的修改。如果使用源码包安装 Squid 软件，需要使用 cachemgr.cgi 的网页管理功能，编译时需要使用“--enable-cachemgr-hostname”选项。

3.7.4　定义 MIME-TYPE 文件：/etc/squid/mime.conf

该文件用来定义 MIME-TYPE，该文件定义的格式是：正则表达式的内容、类型、图标内容、编码、模式。该文件保持默认的配置即可，一般情况下不需要定义。

3.7.5　MSNT 认证的配置文件：/etc/squid/msntauth.conf

/etc/squid/msntauth.conf 是 MSNT 的身份验证配置文件。该文件已经默认定义了 MSNT

的认证信息，不需要进行修改了。

3.7.6　和 Web 的代理捆绑在一起：/etc/httpd/conf.d/squid.conf

该文件主要用来配置与 Web 的代理相关的配置。默认的使用 ScriptAlias 指令将 CGI 程序的 cachemgr.cgi 文件限定在另外的目录/usr/lib/squid/中，用该文件来监控 squid 的问题。该文件保持默认的配置就可以了。

3.8　实 例 应 用

为了更深地了解 Squid 的配置，下面以实例的形式来演示传统代理、透明代理、反向代理的配置过程。

3.8.1　配置 Squid 实现基本的代理功能

标准代理即普通的代理服务，一般以提供 HTTP、FTP 代理为主，需要客户端在浏览器中指定代理服务的地址和端口号（默认端口号为 3128）。对于企业的局域网来说，通过代理服务器同样可以接入 Internet，但一般只能访问 Web 网站和 FTP 站点。同时，通过代理服务器的缓存机制，局域网用户访问 Web 站点的速度可以得到显著提高。

当客户端通过代理服务器请求 Web 页面时，代理服务器会首先检查自己的高速缓存。如果有客户端需要的页面，则直接从高速缓存中读取页面并返回给客户端浏览器，如图 3.4 所示。如果缓存中没有该页面，则代理服务器向 Internet 中发送请求，获得返回的 Web 页面以后，将数据保存至高速缓存并返回给客户端浏览器。缓存加速的对象主要是文字图像等静态的 Web 对象。

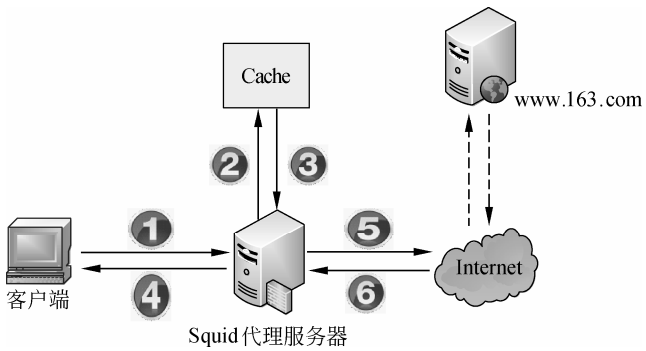

图 3.4　代理服务的缓存加速机制

通过引入缓存加速机制，当客户端在不同的时候访问同一 Web 对象，或者不同的客户端访问相同的 Web 对象的时候，就可以直接从代理服务器的缓存中获得结果。这样就大大减少了向 Internet 提交重复数据访问的过程，加快了客户端的 Web 访问速度。同时，代理

服务器可以在这个“代理访问”过程中加入过滤和控制。

在初次配置 Squid 代理服务时，主要注意两个地方即可：其一，设置好完整的主机名，例如 visible_hostname www.benet.com，如果没有正式可用的完整主机名，也可以将主机名指定为 localhost.localdomain，以避免在 Squid 检查主机名时发生错误。其二，注意添加 http_access allow all 的访问策略，以允许客户端使用代理服务。

实验需求描述如下：

（1）在 Linux 网关主机上启用 Squid 代理服务，为局域网用户（192.168.1.100/24）访问 Internet 网站提供缓存加速，如图 3.5 所示。

（2）调整 squid.conf 配置文件，禁止所有用户通过代理下载超过 10MB 大小的文件。

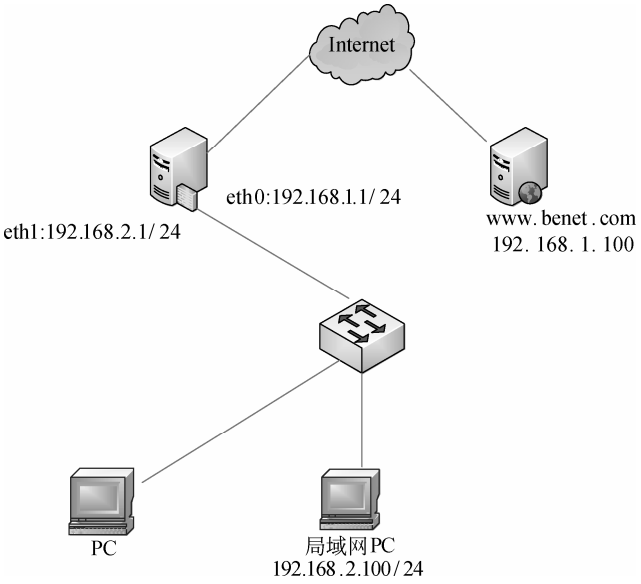

图 3.5　配置 Squid 实现基本的代理功能

【实例 3-2】根据上述环境配置 Squid 实现基本的代理功能（传统代理）。具体操作步骤如下。

（1）配置 Squid 代理服务器端。

```
[root@www ~]# vi /etc/squid/squid.conf
http_port 3128
visible_hostname www.benet.com              //指定可见的主机名
http_access allow all    //查找修改此行，否则应放在 http_access deny all 行之前
```

（2）初始化并启动代理服务。

```
[root@www ~]# service squid start
正在启动 squid：.                                          [确定]
```

如果不使用系统服务脚本，也可以直接调用 squid 程序。

```
[root@www ~]# squid -z                          //-z 选项用于初始化缓存目录
2013/05/30 15:09:40| Creating Swap Directories
[root@www ~]# squid -k reconfigure          //需要重新加载配置文件时使用此命令
```

（3）修改客户端浏览器设置指定所使用代理服务器的 IP 地址、端口。

（4）通过 Squid 访问日志查看客户端的访问记录（tail -f 用于跟踪文件变化，按 Ctrl+C 中止）。

```
[root@www ~]# tail -f /var/log/squid/access.log
...
1369899684.010              4   192.168.2.100   TCP_MISS/403   4278   GET
http://192.168.1.100/ - DIRECT/192.168.1.100 text/html
```

3.8.2　配置透明代理

透明代理（Transparent Proxy）提供与传统代理相同的功能和服务，其“透明”之处在于：客户端不需要在浏览器中指定代理服务器的地址和端口号，代理服务对客户端用户来说是“透明”的，用户甚至并不知道自己已经在使用代理服务了。

在很多企业网络中，代理服务器往往也就是局域网接入 Internet 的网关。因此，管理员就有机会将局域网访问 Web 站点的数据转交（Redirect，重新定向）给网关本节的代理服务程序。而这个数据转交的工作由防火墙策略来完成，局域网的客户机并不需要知道具体的实现过程。

在配置透明代理之前，需要了解一下设置 iptables 的重定向策略。

iptables 防火墙有一个名为 REDIRECT（重定向）的数据包处理策略，可以在防火墙主机内部转发数据包。简单地说，这个策略可以将符合条件的数据包交给本机中某个特定端口（如代理服务的 3128 端口）上的服务进程进行处理。

REDIRECT 只能在 nat 表的 PREROUTING 或 OUTPUT 链以及被其调用的链中使用。通过“--to-ports 端口号”的形式指定映射的目标端口。

【实例 3-3】将从 eth1 网卡进入、源 IP 地址属于 192.168.1.0/24 网段且访问 TCP 协议 80 端口（Web）页面数据包，重定向转交给运行在本机 3128 端口上的服务（Squid）进行处理。

```
[root@www ~]# iptables -t nat -I PREROUTING -i eth1 -s 192.168.1.0/24 -p
tcp --dport 80 -j REDIRECT --to-ports 3128
```

实验需求描述如下：

（1）在代理服务器（192.168.2.1/24）中启用 Squid 服务，并添加透明代理支持。

（2）该服务器同时作为局域网内（192.168.2.0/24）各主机的网关服务器，如图 3.4 所示。

（3）设置 iptables 防火墙规则，将局域网用户访问 Internet 网站的数据包进行重定向，交给 Squid 服务处理。

【实例 3-4】根据上述环境配置 Squid 实现透明代理服务。具体操作步骤如下。

（1）修改 squid.conf 配置文件，添加透明代理支持（其他基本功能配置略）。

```
[root@www ~]# vi /etc/squid/squid.conf
http_port 192.168.2.1:3128 transparent
```

（2）重新加载 Squid 服务配置。

```
[root@www ~]# service squid reload
```

（3）设置 iptables 规则，将访问 HTTP 的数据重定向给代理服务器。

```
[root@www ~]# iptables -t nat -I PREROUTING -i eth1 -s 192.168.2.0/24 -p
tcp --dport 80 -j REDIRECT --to-ports 3128
```

（4）确认各客户端的 IP 地址、默认网关地址等设置正确，浏览器中不需要设置代理（如果已经设置则改回不使用代理）。

（5）在客户端浏览器重新访问网页，同时在代理服务器上跟踪 Squid 访问日志记录。如果能正常访问网页且 access.log 有访问记录，说明透明代理配置成功。

```
[root@www ~]# tail -f /var/log/squid/access.log
...
1369899684.010              4   192.168.2.100   TCP_MISS/403   4278   GET
http://192.168.1.100/ - DIRECT/192.168.1.100 text/html
```

3.8.3　配置反向代理

反向代理（Reverse Proxy）也同样提供缓存加速，只不过服务的对象反过来了。传统代理也好，透明代理也好，大多是为局域网用户访问 Internet 中的 Web 站点提供缓存代理；而反向代理恰恰相反，主要为 Internet 中的用户访问企业局域网内的 Web 站点提供缓存加速，是一个反方向的代理过程，如图 3.6 所示，因此称为反向代理。

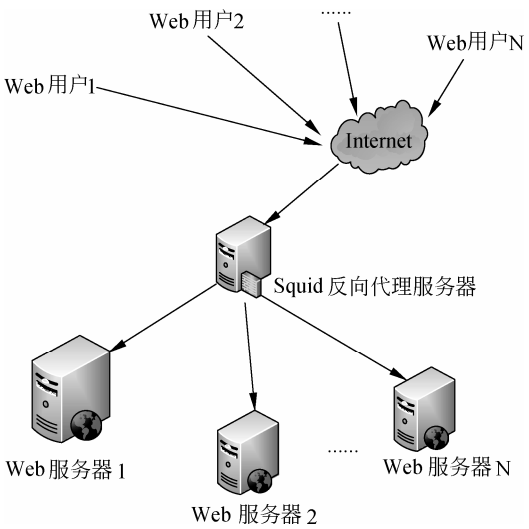

图 3.6　反向代理的缓存加速

在 squid.conf 文件中，实现反向代理服务最基本的选项有两处。其一，在 http_port 配置项后边增加 vhost 选项；其二，使用 cache_peer 配置项指定后台真正提供 Web 服务的主机（有时称为上游服务器）的 IP 地址端口等参数。

使用 cache_peer 配置项指定上游 Web 服务器主机的位置，配置行格式如下：

```
cache_peer Web服务器地址 服务器类型 http端口 icp端口 [可选项]
```

其中，服务器类型对应到目标主机的缓存级别，上游 Web 主机一般使用 parent（父服

务器）；icp 端口用于连接相邻的 ICP（Internet Cache Protocol）缓存服务器（通常为另一台 Squid 主机），如果没有，则使用 0；可选项是提供缓存时的一些附件参数，例如 originserver 表示该服务器作为提供 Web 服务的原始主机，weight=n 指定服务器的优先权重，n 为整数，数字越大优先级越高（默认为 1）；max-conn=n 指定反向代理主机到该 Web 服务器的最大连接数。

注意：vhost 与 transparent 不要同时使用。

实验需求如下：

（1）公司使用两台 Web 服务器（192.168.2.10/24192.168.2.20、24）实现负载分担，并在前端使用 Squid 做反向代理加速，如图 3.7 所示。

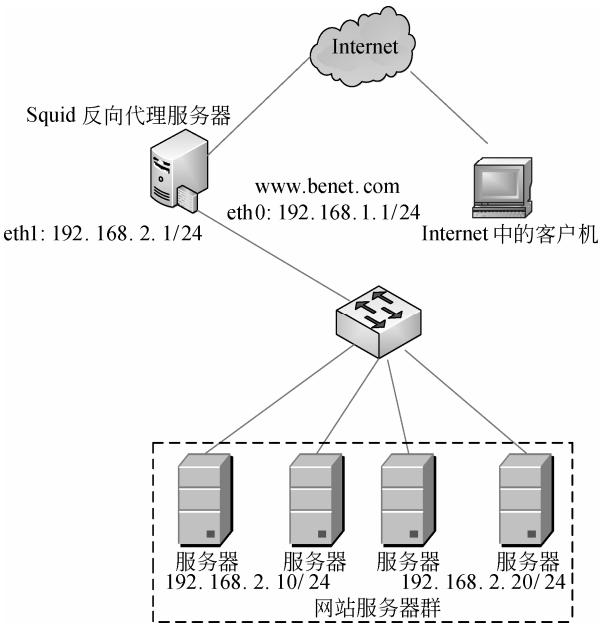

图 3.7　配置 Squid 实现反向代理加速

（2）Internet 用户直接访问的是 Squid 反向代理服务器（将监听端口修改为 80）。

（3）通过 Squid 代理服务器间接访问实际的网站服务器。

【实例 3-5】根据上述环境配置 Squid 实现反向代理，操作步骤如下。

（1）修改 squid.conf 配置文件。

```
[root@www ~]# vi /etc/squid/squid.conf
http_port 192.168.1.1:80 vhost vport
cache_peer 192.168.2.10 parent 80 0 originserver
cache_peer 192.168.2.20 parent 80 0 originserver
visible_hostname www.benet.com
```

（2）重新启动 squid 服务。如果在 80 端口已经运行 httpd 服务，注意先关闭。

```
[root@www ~]# service squid restart
```

（3）在 Internet 中的客户端主机上，使用浏览器访问反向代理服务器的地址（如

192.168.1.1）。

（4）在 Squid 反向代理服务器上，查看 access.log 访问日志，验证反向代理是否成功。

```
[root@www ~]# tail /var/log/squid/access.log
//......省略部分内容
1369965652.982                2   192.168.1.10   TCP_MISS/403   4306   GET
http://192.168.1.1/ - FIRST_UP_PARENT/192.168.2.10 text/html
1369965672.675                1   192.168.1.10   TCP_MISS/304   273   GET
http://192.168.1.1/icons/apache_pb2.gif - FIRST_UP_PARENT/192.168.2.10 -
```

其中，192.168.1.10 是测试的 Internet 中客户端主机的 IP 地址。

3.9　测 试 服 务

搭建一个服务就是为了能够很好地应用在网络环境中。本节将讲述在 Windows、Linux 客户端下测试 Squid 代理服务器的功能。

3.9.1　Windows 客户端

在前面实例的配置下，现在使用 Windows 客户端测试标准代理服务器。配置步骤如下。

（1）在 IE 浏览器中依次选择“工具”|“Internet 选项”，打开“Internet 选项”对话框，如图 3.8 所示。

（2）选择“连接”选项卡，然后单击“局域网设置（L）”按钮，打开如图 3.9 所示对话框。

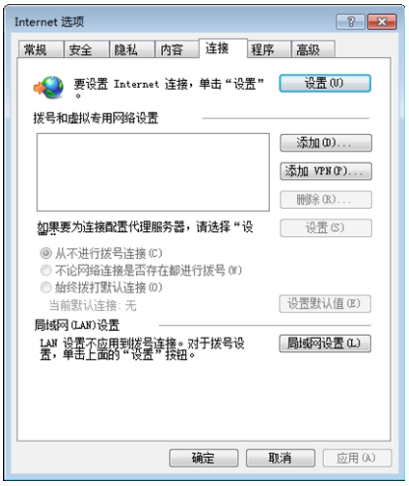

图 3.8　“Internet 选项”对话框

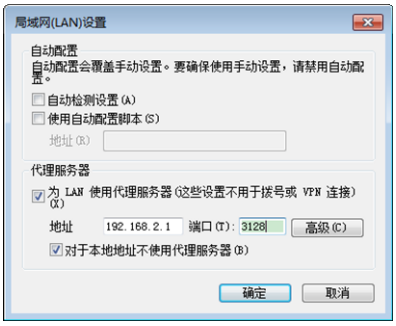

图 3.9　“局域网（LAN）设置”对话框

（3）选中“局域网（LAN）设置”对话框中“代理服务器”选项区域下的复选框，并且在“地址”和“端口”文本框中输入代理服务器的地址和端口。这里的代理服务器地址和端口分别是 192.168.2.1 和 3128。然后单击“确定”按钮。

（4）这时就可以在浏览器中输入 Web 服务器地址进行访问了。访问到的页面如图 3.10

所示。

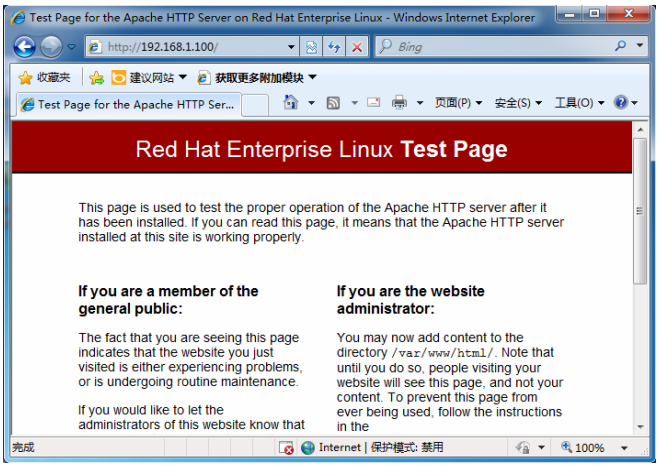

图 3.10　Web 服务器测试页面

对于透明代理和反向代理在客户端的配置比较简单，这里就不再介绍了。测试结果可以从日志文件中查看，日志文件前面已介绍不再赘述。

3.9.2　Linux 客户端

在前面实例的配置下，现在使用 Linux 客户端测试标准代理服务器。配置步骤如下。

（1）在 Firefox 浏览器中依次选择“编辑”|“首选项”命令，打开“Firefox 首选项”对话框，如图 3.11 所示。

图 3.11　“Firefox 首选项”对话框

（2）在其中选择“高级”选项卡，接着依次选择“网络”|“设置”选项，打开如图 3.12 所示对话框。

（3）在其中选中“手动配置代理：（M）”单选按钮，在“HTTP 代理：（X）”和“端

口：（P）”文本框中输入代理服务器的地址 192.168.2.1 和端口号 3128。然后，单击“确定”按钮，代理服务器客户端设置成功。

图 3.12 “连接设置”对话框

（4）测试代理服务器是否配置正确，在客户端通过访问 Web 服务器进行测试服务器的设置。如配置成功，测试结果如图 3.13 所示。

图 3.13　Web 服务测试页面