# Using Squid to Reduce Bandwidth

Workshop on Optimization Technologies
for Low-Bandwidth Networks

Duane Wessels
Squid Cache Team
*wessels@squid-cache.org*

October 13, 2006

# Presentation Outline

- Web Proxies
- Caching
- Squid
- Squid ACLs
- Configuring Clients
- Delay Pools
- Log Files
- Anything Else

# Web Proxies

# Proxies: Introduction

- A proxy is an intermediary application that sits between clients and servers.
- Typically used to:
  - Control access
  - Record activities
  - Convert between protocols
- Proxies exist for many different protocols
  - SMTP/email
  - DNS
  - HTTP/Web
  - IRC/IM/Chat

# Proxies: Web Objects

- With a web proxy, we have the opportunity to cache and reuse certain server responses.
- When caching, we usually talk about "objects". For example:
  - An HTML page
  - A JPEG image
  - A PDF document
  - Flash animation
- Objects are identified by URL (Uniform Resource Locator)
- Note that sometimes the term "web page" refers to a number of different objects (HTML, images, etc).

# Caching

# Caching: Requests and Responses

- Browsers (clients) send HTTP requests
- Origin servers send HTTP responses (objects).
- A response may be cachable, depending on both request and response headers.
- For example, a response that required user authentication is usually not cachable.
- Origin servers may provide specific caching instructions:
  - Cachable/Uncachable
  - Cache validator
  - Freshness time-to-live (TTL)

# Caching: Fresh vs Stale

- Cached responses may be reused and sent to clients if they are still "fresh"
- Freshness is determined by:
  - An explicit TTL value from the server
  - Making a cache validation request, a.k.a. conditional request
  - Using simple heuristics (rules)
- Freshness/staleness may also be determined by client requirements or preferences.

# Squid

# Squid: Introduction

- Open Source
- Designed for Unix
- Runs on Windows also
- Speaks HTTP to clients (browsers)
  - not an email proxy
  - not an IRC/IM proxy
  - not a telnet proxy
  - sort of an FTP proxy

# Squid: Versions

- Squid-2.6 is current stable release
- Squid-3.0 is current development release
- BWMO book discusses some Squid-3.x features
- I will assume Squid-2.x here.

# Squid: Installation

- Easiest way is via OS software package installer.
- On Ubuntu use GUI, or:

  `sudo apt-get install squid`

- Or, get source code from `www.squid-cache.org`, `ftp.squid-cache.org`, or mirror site, then:
  - `tar xzvf squid-version.tar.gz`
  - `cd squid-version`
  - `./configure`
  - `make`
  - `make install`

# Squid: squid.conf

- See `/etc/squid/squid.conf`
- Lots of comments for documentation
- To see the relevant non-commented lines:

```
$ grep -v '^#' squid.conf | grep . | less
```

# squid.conf: Important Bits

```
http_port 3128

cache_dir ufs /var/spool/squid 1000 16 256
cache_mem 8 MB

cache_effective_user proxy

acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY

visible_hostname squid.domain.tld
```

# Squid: Starting at Boot

- Ubuntu Linux probably adds Squid to the startup scripts for you.
- If not, you can run

```
$ sudo update-rc.d squid defaults
```

- Or, to disable:

```
$ sudo update-rc.d -f squid remove
```

# Squid: Starting and Stopping

- `# /etc/init.d/squid stop`
- `# /etc/init.d/squid start`
- `# /etc/init.d/squid restart`
- `# /etc/init.d/squid reload`

# Squid: Logfiles

- `/var/log/squid/cache.log`
  Contains run-time status messages, warnings, and errors.
- `/var/log/squid/access.log`
  One line for each client request, including URL, bytes transferred, status code, and more.
- `/var/log/squid/store.log`
  Transaction log for objects that enter and leave the cache.
- Open a new terminal window and run:

  ```
  $ tail -f /var/log/squid/cache.log
  ```

- Open another new terminal window and run:

  ```
  $ tail -f /var/log/squid/access.log
  ```

# Configuring: Cache Disks

- The *cache_dir* directive(s) tell Squid how and where to store cached objects.
- Syntax:

  `cache_dir type path megabytes L1 L2`

- The default *type* is `ufs`, but `aufs` has better performance on Linux. `diskd` is good too.
- *path* can be anywhere on the filesystem, but is usually a dedicated disk or partition.
- *megabytes* is an upper limit on how much space Squid should use for this *cache_dir*. It should be less than 90% of the actual capacity.
- *L1* and *L2* specify the number of first- and second-level directories to use. Use `16` and `256` by default. These should not be changed after Squid has placed objects on the disk.

# Squid: Create Swap Directories

- After adding a *cache_dir*, you need to initialize it with this command:

```
$ sudo squid -z
2006/10/12 09:48:24| Creating Swap Directories
$
```

# Squid: Memory

- Squid likes to use a lot of memory.
- Each object in the cache takes about 54 bytes in memory. More on 64-bit processors.
- Also need memory for temporary buffers, in-memory objects, DNS caches, and various other data structures.
- Don't get too excited about the *cache_mem* directive. It probably doesn't do what you think, and doesn't affect performance that much.
- Increase *cache_mem* only after Squid has been filled up with objects and running for a few weeks.

# Squid ACLs

# ACLs: Very Important

- It is very important that you allow access to Squid from only your local users or people that you trust.
- An open proxy will be abused by people that download pornography, post hateful messages, and make purchases with fraudulent credit cards.
- Those transactions will be traced back to your IP address.
- Your IP address may be listed in an Open Proxy blacklist.

# ACLs: The Defaults

- Here are the default Squid ACLs:

```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563      # https, snews
acl SSL_ports port 873          # rsync
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https, snews
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT
```

# ACLs: Continued

```
http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
#http_access deny to_localhost
http_access allow localhost
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
http_access deny all

http_reply_access allow all

icp_access allow all
```

# ACLs: Allow Local Clients

- First, define the *acl*:

```
acl our_networks src 192.168.1.0/24 192.168.2.0/24
```

- Then, allow the access, just before the final *deny* rule:

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
http_access allow our_networks
http_access deny all
```

# ACLs: Proxy Authentication

- Give each user a username and password.
- Allows you to easily track down individuals that may cause problems.
- IP-based controls become less important
- Can utilize existing user databases, such as LDAP, Windows NTLM, Unix passwd, Unix PAM, and more.
- Incompatible with transparent caching.

# ACLs: NCSA Proxy Auth

- NCSA refers to the password file format originally developed for the NCSA (now Apache) HTTP server.
- Add these lines to squid.conf:

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm "Squid Caching Proxy"
auth_param basic credentialsttl 1 hour
auth_param basic casesensitive off
```

- And add these lines to the ACL section:

```
acl ProxyUsers proxy_auth REQUIRED
http_access allow ProxyUsers
```

- You need *htpasswd* program from Apache to create and manage the password file.
  - or from `http://www.squid-cache.org/htpasswd/`

# Configuring Clients

# Clients: Configuring the Proxy

- To maximize caching, you want as many clients as possible to send requests to Squid.
- Users cannot always be trusted to:
  - do the configuration
  - do it correctly
  - leave the setting alone
- Blocking output port 80 on the firewall is a good way to force users through the proxy cache.
- But not the only way.

# Clients: Manual Configuration

- Browsers have proxy settings under “Tools” or “Preferences” or “Settings”.
- See if you can find the configuration for your browser.
- In a corporate or university environmet, you may be able to pre-configure workstations with proxy settings.

# Clients: PAC

- Proxy Auto Configuration is a script (javascript), residing at an easy-to-remember URL.
- The browser executes the javascript function before each request.
- The PAC script returns a proxy (or list of proxies) that the browser should use for the request.
- Useful for simple load-balancing of multiple proxies.
- Also useful if proxy/proxies are unstable.

# Clients: Sample PAC Script

```
function FindProxyForURL(url, host)
{
    if (isPlainHostName(host))
        return "DIRECT";
    if (!isResolvable(host))
        return "DIRECT";
    if (url.substring(0, 5) == "http:")
        return "PROXY random.us.ircache.net:3128; DIRECT";
    if (url.substring(0, 4) == "ftp:")
        return "PROXY random.us.ircache.net:3128; DIRECT";
    if (url.substring(0, 7) == "gopher:")
        return "PROXY random.us.ircache.net:3128; DIRECT";
    return "DIRECT";
}
```

# Clients: WPAD

- Web Proxy Auto Discovery
- A way for clients to automatically find the PAC script.
- Browser does DNS lookup for *wpad.domain.tld*.
- If DNS lookup succeeds, makes HTTP request for `http://wpad.domain.tld/wpad.dat`
- Works well if your clients have hostnames in your DNS domain.
- ...and if its enabled by default.
  - it is in IE
  - what about firefox?
- Also a DHCP option. For ISC DHCP server:

```
option wpad "http://example.com/path/to/proxyconfig.pac";
```

# Clients: HTTP Interception

- a.k.a. Transparent Caching
- A router or switch redirects outbound HTTP (port 80) traffic to the proxy
- Proxy spoofs IP addresses and pretends to be the origin server.
- Client sends normal HTTP requests.
- Proxy rewrites HTTP request so that hostname is in the URL, then processes normally.
- Captures all HTTP traffic without configuring any clients.
  - even viruses!!

# Clients: HTTP Interception

- Doesn't work with proxy authentication
- or RFC 1413 "ident" servers
- Squid configuration is a little tricky
- and prone to forwarding loops
- Complicated if you want to run an HTTP server on the same box

# Interception: Linux Configuration

- add to `/etc/sysctl.conf`

  ```
  net.ipv4.ip_forward = 1
  ```

- Add an *iptables* rule:

  ```
  $ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
          -j REDIRECT --to-port 3128
  ```

# Interception: Cisco Configuration

- Define a route-map that sets the next-hop to Squid's IP address:

```
route-map proxy-redirect permit 10
 match ip address 110
 set ip next-hop 192.168.0.1
```

- Define an access-list that matches port 80 packets and does not match those coming from Squid itself:

```
access-list 110 deny   tcp any any neq www
access-list 110 deny   tcp host 192.168.0.1 any
access-list 110 permit tcp any any
```

- Apply the route-map to the outgoing interface:

```
interface FastEthernet0/0
 ip policy route-map proxy-redirect
```

# Interception: Squid Configuration

- As of Squid-2.6, the configuration is simple:

```
http_port 3128 transparent
```

# Interception: WCCP

- The Web Cache Coordination Protocol
- Can distribute load between multiple caches
- Better at dealing with cache failures
- Uses GRE to encapsulate packets

# WCCP: OS Configuration

- Configure a GRE tunnel endpoint in `/etc/network/interfaces`:

```
iface gre0 inet static
        address 172.16.98.1
        netmask 255.255.255.252
        broadcast 172.16.98.3
```

- Enable the interface:

```
$ ifup gre0
```

- Configure iptables:

```
$ iptables -t nat -A PREROUTING -i gre0 -p tcp --dport 80 \
        -j REDIRECT --to-port 3128
```

- Warning: above is untested!

# WCCP: Cisco Configuration

- To enable WCCPv1:

```
ip wccp version 1
ip wccp web-cache
interface FastEthernet0
 ip wccp web-cached redirect out
```

# Interception: Etc

- There are many more ways to accomplish HTTP interception
- Linux/BSD-based routers
- Running Squid on your router
- Fancy and expensive Layer 4 switches

# Delay Pools

# Delay Pools: Introduction

- Squid's Delay Pools can be used to rate-limit traffic coming into the the cache (cache misses).
- Cache hits are never delayed
- Implemented as "bandwidth buckets"
- Squid delays reading a response until bandwidth is available in the bucket.
- Access control rules determine which responses are subject to delays.

# Delay Pools: Buckets

- Three types of buckets:
- Individual
- Network
- Aggregate
- Buckets have 2 parameters:
- Capacity (in bytes)
- Fill rate (in bytes/sec)

# Delay Pools: Classes

- Three types of classes: 1, 2, 3
- A class 1 pool has a single, Aggregate bucket
- A class 2 pool has an Aggregate bucket, and 256 Individual buckets.
- A class 3 pool has an Aggregate bucket, 256 Network buckets, and 65,536 Individual buckets.

# Delay Pools: Back to Buckets

- Aggregate bucket applies to all clients.
- Class 2 Individual buckets are indexed by 4th octet of client IPv4 address.
- Class 3 Network buckets are indexed by 4th octet of client IPv4 address.
- Class 3 Individual buckets are indexed by 3rd and 4th octets of client IPv4 address.

# Delay Pools: Defining a Pool

- Syntax:

```
delay_pools count
delay_class index class
delay_parameters index R/C [R/C [R/C]]
delay_access index allow ...
```

- For example:

```
delay_pools 1
delay_class 1 3
delay_parameters 1 500/1000 600/1800 700/4000
delay_access 1 allow our_network
```

# Delay Pools: Notes

- Delay pool state is lost whenever you reconfigure Squid
  - a bug...
- Does not guarantee fairness
  - Clients on connections with low-numbered file descriptors may receive more than their fair share.
- Squid can only delay reading data from the kernel. The packets will still arrive over the network, subject to TCP window sizes and such. TCP data will be delayed only if the entire HTTP response is larger than the TCP window size.
- Delay pools assume certain network addressing schemes
  - Class 3 pool sort of assumes a /16 address block, for example.
- Don't forget to convert between bytes/sec and bits/sec.

# Log Files

# Logs: Location

- Ubuntu places Squid logs in `/var/log/squid`.
- Ubuntu automatically rotates Squid logfiles every day.
- See `/etc/logrotate.d/squid`

# Logs: cache.log

- Look here for warnings from Squid
- Error messages if Squid happens to crash
- Can enable more verbose debugging messages if necessary. See *debug_options* in `squid.conf`.
- Can also send these messages to syslog with *-s* command line option.

# Logs: access.log

- One line per client request.
- Ten fields
  1. timestamp (end of request, in unix epoch seconds)
  2. elapsed time (milliseconds)
  3. client IP address
  4. cache stats and HTTP status
  5. number of bytes sent to client
  6. request method
  7. request URL
  8. username
  9. hierarchy code and next-hop address
  10. content-type
- An example:

```
1159854402.086    804 127.0.0.1 TCP_MISS/200 5886
  GET http://www.google.com/search? -
  DIRECT/66.249.89.104 text/html
```

- Note query terms are stripped by default

# Logs: Cache Status Codes

- TCP_HIT : A valid copy of the requested object was in the cache.
- TCP_MISS : The requested object was not in the cache.
- TCP_REFRESH_HIT : The requested object was cached but STALE. The IMS query for the object resulted in "304 not modified".
- See Squid FAQ for others.

# Logs: HTTP Status Codes

- 200 OK
- 204 No Content
- 206 Partial Content

- 301 Moved Permanently
- 302 Moved Temporarily
- 303 See Other
- 304 Not Modified

- 400 Bad Request
- 401 Unauthorized
- 403 Forbidden
- 404 Not Found
- 407 Proxy Authentication Required

- See Squid FAQ for others.

# Logs: store.log

- Low-level details of objects saved to, and removed from, the disk cache.
- `1159858729.772 SWAPOUT 00 00000E8B 036F4D5F20A9E51D29AF9B724697DA`
  `200 1159858729 1158918355 -1 image/jpeg 4160/4160`
  `GET http://rosie.ripe.net/images/istanbul.jpg`
- 13 fields
  1. timestamp
  2. SWAPOUT or RELEASE
  3. cache_dir index
  4. cache file number
  5. cache key (MD5)
  6. HTTP status code
  7. HTTP Date
  8. HTTP Last-Modified
  9. HTTP Expires
  10. Content-type
  11. Content-length and bytes written
  12. request method
  13. URL

# Logs: Analysis Tools

- Calamaris
- Webalizer
- More at `http://www.squid-cache.org/Scripts/`

The End