How Web Site and Server Design Affect the Ability to Properly Cache Objects in Client-Side Proxies

Stephen Pierzchala
August 21, 2002

Topics Covered

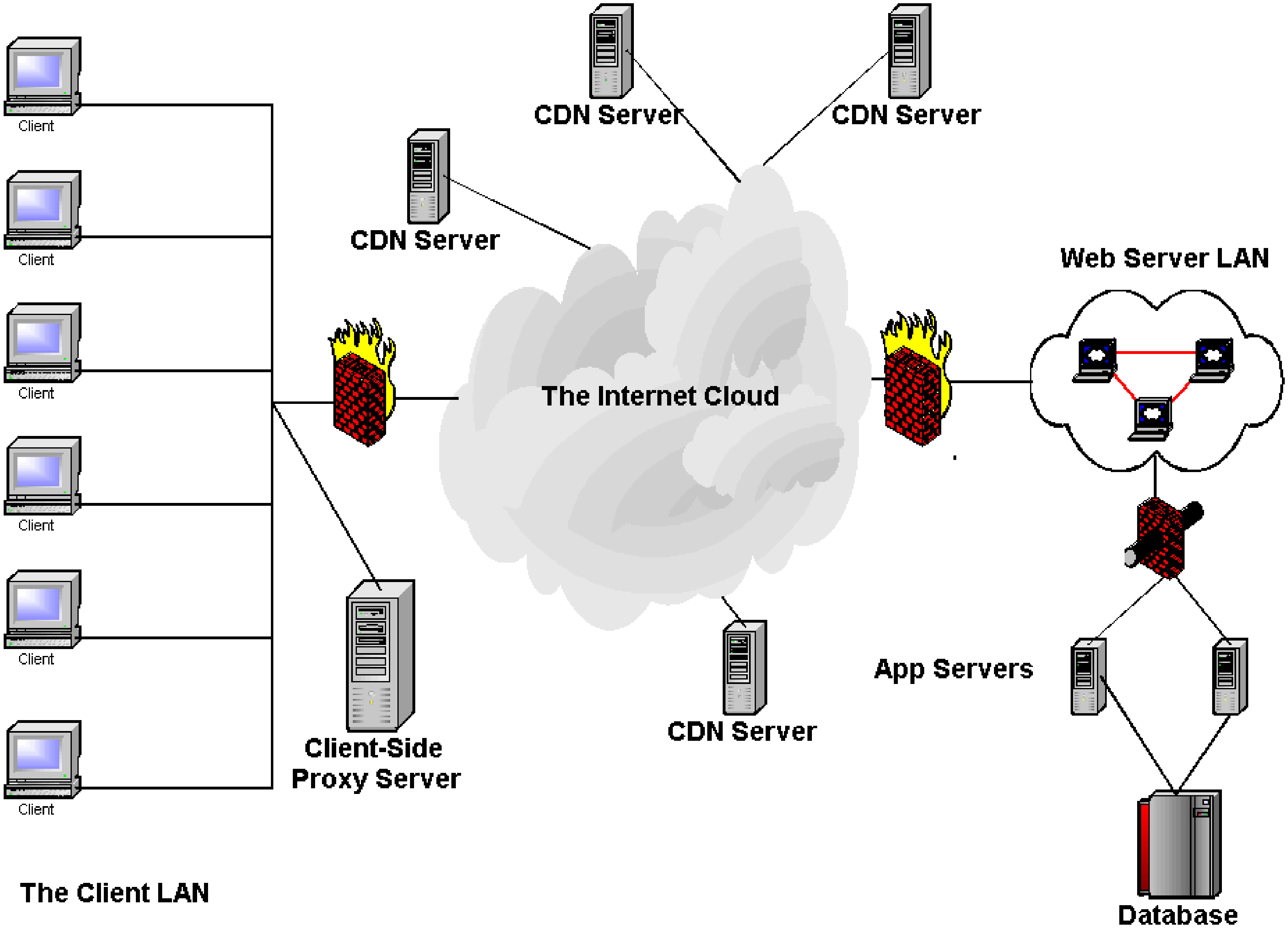
- Discussion of caching and its benefit to Web-based enterprises

- Analysis of cache-loading test results

Cacheability – What is it?

- The off-loading of Web site content-delivery responsibilities onto network edge caches **OR** client network proxy servers
- This presentation focuses on the latter method

Simple Picture of the Internet



Client-Side Network Administrator

Cacheability Objectives:

- Reduce Bandwidth usage and cost
- Control Content coming into network
- Client-Side Proxy server does both of these and is much more efficient than browser-level cache mechanisms

Server-Side Network Administrator

Cacheability Objectives:

- Reduce Bandwidth usage and cost
- Reduce strain on limited hardware resources
- Control TTL of data
- Client-Side Proxy server provides help with the first two items, and the Web server administrator controls the third

Content and Data Administrators

- The wild card in this scenario

- Want to ensure freshness of data

- For these administrators…
  - **ALL DATA IS DYNAMIC**
  - Caching = **FOREVER**
    - Content that is cached outside their servers means the client never comes back to the origin server in their mind
    - **They Must Track Site Visitation!**

- These are the forces that must be brought back from the Dark Side

CDNs

- Content Delivery (or Distribution) Networks

- Advanced network of edge caches that move content closer to the requesting clients
  - Great for large e-commerce enterprises
  - Too expensive for small- or medium-sized businesses, non-profits, NGOs, governments or educational institutions

Client-Side Proxy Servers

- The term “Proxy server” will be used in this presentation to represent any device on a LAN that caches data for a large number of clients
    - Proxy Servers
    - Cache Devices
    - Firewall/Proxy combinations

- Using client-side proxy servers and server caching messages, Web sites can build their own CDNs

Client-Side Proxy Servers

- There are two methods used by proxy servers to determine the cacheability of a requested object:

1. ***Explicit Caching Using Server Header Message***
2. ***Implicit Caching Using Proxy Configuration***

Server Header Message Caching

- Client-side Proxy server caches objects based on the explicit message appearing in the server header

- ***Expires*** and ***Cache-Control*** messages determine exactly how long the object can be cached for

Server Header Message Caching, cont’d

```
HTTP/1.1 200 OK
Date: Tue, 02 Jul 2002 20:35:56 GMT
Server: Apache
Cache-Control: max-age=86400
Expires: Wed, 03 Jul 2002 20:35:56 GMT
Last-Modified: Thu, 06 Jun 2002 21:46:08 GMT
ETag: "13dee-2550-3cffd820"
Accept-Ranges: bytes
Content-Length: 9552
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
```

Server Message Caching, cont'd

Important Caching Messages:

- ***Cache-Control: no-cache***
  - Tells caches and proxy servers to please not cache
- ***Cache-Control: no-store***
  - Tells caches and proxy servers to ***definitely*** not cache
- ***Cache-Control: max-age=X***
  - Tells caches and proxy servers to cache the item for X seconds
- ***Cache-Control: max-age=X, must-revalidate***
  - Tells caches and proxy servers to cache the item for X seconds and then re-validate the object with the origin server
- ***Expires: [DATE]***
  - Indicates the date and time when the content expires. Usually equal to Server Date *plus* max-age

Server Message Caching, cont’d

- ***Pragma: no-cache***
  - This is **NOT** a valid Server Response header
  - Client Request ***ONLY***
  - Large number of sites use this to try and prevent caching – proxy servers ignore this when it is returned by a Web server
  - The correct Server Response headers to send to prevent caching:

Cache-control: no-cache

OR

Cache-control: no-store

Implicit Caching

- Client-side Proxy server caches objects based on internal settings for objects with no explicit cache setting

- ***Last-Modified*** and ***ETag*** messages are used by the proxy to determine how long the object can be cached for before being re-validated

- Implicit Caching causes most ***304 Not Modified*** messages in Web server logs

Implicit Caching, cont'd

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Connection: keep-alive
Set-Cookie: SITESERVER=ID=BLAH_BLAH_BLAH
Date: Tue, 02 Jul 2002 20:45:25 GMT
Content-Type: image/gif
Accept-Ranges: bytes
Last-Modified: Wed, 10 Jan 2001 20:49:12 GMT
ETag: "01435c9467bc01:907"
Content-Length: 43
```

How to Sabotage Caching

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Connection: keep-alive
Set-Cookie: SITESERVER=ID=BLAH_BLAH_BLAH
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Date: Tue, 02 Jul 2002 20:45:25 GMT
Content-Type: image/gif
Accept-Ranges: bytes
Last-Modified: Wed, 10 Jan 2001 20:49:12 GMT
ETag: "01435c9467bc01:907"
Content-Length: 43
```

- The server header above is from a real e-commerce server
- An ***Expires*** date in the past is no better than setting ***EXPIRES = Date***
  - Same effect if ***Cache-control: max-age=0*** – in fact, in many servers, this automatically generates an ***Expires = Date*** header

How to Sabotage Caching – Part 2

```
HTTP/1.1 200 OK
Server: Netscape-Enterprise/3.6 SP3
Date: Tue, 02 Jul 2002 21:03:00 GMT
Content-type: image/gif
Etag: "8b548-2e39-3c33017d"
Last-modified: Wed, 02 Jan 2002 12:47:57 GMT
Content-length: 11833
Accept-ranges: bytes
```

- Why isn’t this **EXPLICITLY** cacheable?
- Content has not changed since January 2002!

How to Cache

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Tue, 02 Jul 2002 21:27:04 GMT
P3P:CP="BUS CUR CONo FIN IVDo ONL OUR PHY SAMo TELo"
Connection: close
Expires: Tue, 01 Jul 2003 21:27:05 GMT
Cache-Control: max-age=31449600
Content-Type: text/css
HMServer: BLAH_BLAH_BLAH
```

- This is the server header returned for a CSS file

- However, they prevent persistent connections with the ***Connection: close*** message

How to Cache – Part 2

```
HTTP/1.0 200 OK
Connection: Keep-Alive
Expires: Tue, 09 Jul 2002 15:39:43 GMT
Cache-Control: public, max-age=300, no-transform
MIME-Version: 1.0
Date: Tue, 09 Jul 2002 15:34:43 GMT
Server: AOLserver/3.4
Content-Type: text/html
Content-Length: 9794
```

- This file is ***VERY*** cacheable, with a clearly defined TTL
- The ***no-transform*** directive indicates that this object **MUST NOT** be modified by intermediate caches

The Problem

- Very few sites set explicit caching messages in the server headers
  - This applies to both cacheable and non-cacheable items

The Result

- If a site sets explicit caching information, the proxy server knows exactly when the file is valid for serving or in need of re-validation from the server

- If a server sets no explicit caching information, the proxy server uses its internally configured mechanisms to determine how long the file will be valid for
  - May be longer than the Web Server wants
  - May be less than the Client-side Network Administrator wants

The Result, cont'd

- If a server sets zero or negative caching times, the object **SHOULD BE** in a constant state of change, with each hit returning a different result
  - Dynamic HTML
  - PHP, ASP, JSP, CFM, SHTML
  - **SHOULD NOT BE SET FOR STATIC IMAGES THAT NEVER CHANGE NAME, SIZE, OR LOCATION**

- Using explicit caching messages, a Web site can have very tight control of how long objects are cached, while reaping the benefits of lower bandwidth usage and server loads.

Configuring Caching

- How easy is it to configure caching directives?
  - One line in an Apache Directory Container Directive
  - One line in an IIS Directory Settings Box

- How discrete are caching directives?
  - Can be set for entire Web Document Tree
  - Can be set directory by directory
  - Can be set by file type
  - Can be set for individual files

Configuring Caching, cont'd

- Setting ***Expires*** header in Apache

```
<Directory "/home/webdev/htdocs/documents">
    Options Indexes FollowSymLinks
    AllowOverride None
    ExpiresDefault A600
    Order allow,deny
    Allow from all
</Directory>
```

```
HTTP/1.1 200 OK
Date: Tue, 09 Jul 2002 21:31:28 GMT
Server: Apache
Cache-Control: max-age=600
Expires: Tue, 09 Jul 2002 21:41:28 GMT
Last-Modified: Sun, 07 Jul 2002 04:38:48 GMT
ETag: "abad1-c030-3d27c5d8"
Accept-Ranges: bytes
Content-Length: 49200
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: application/postscript
```

Configuring Caching, cont’d

- Setting ***Cache-contol*** header in Apache

```
<Directory "/home/webdev/htdocs/documents">
    Options Indexes FollowSymLinks
    AllowOverride None
    Header set Cache-control "max-age=600, must-revalidate"
    Order allow,deny
    Allow from all
</Directory>
```

```
HTTP/1.1 200 OK
Date: Tue, 09 Jul 2002 21:39:24 GMT
Server: Apache
Cache-control: max-age=600, must-revalidate
Last-Modified: Sun, 07 Jul 2002 04:38:48 GMT
ETag: "abad1-c030-3d27c5d8"
Accept-Ranges: bytes
Content-Length: 49200
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: application/postscript
```

Analysis of Proxy Server Effectiveness

Test Setup

- Data gathered from two sites July 3-8, 2002
  - Linux server on Keynote Systems internal LAN
  - Linux server on AT&T Broadband Network
- Proxy "loaded" using GNU WGET
  - Static list of 90 Web Pages retrieved using a CRON job
- Squid Proxy Server used for caching objects
  - http://www.squid-cache.org

Test Results

- With no tweaking or filtering, Squid is able to serve approximately 60-85% of requested objects directly from cache
    - Of these, 5%-7% were revalidation requests (***304 Not Modified***)
- However, only 40%-60% of the total bytes are served by the cache

Test Results, cont'd

	Cache Hits	%	Cache Misses	%
Keynote LAN	191352	85.44	32594	14.55
ATTBI Network	165088	83.57	32414	16.41

	KB Hits	%	KB Miss	%
Keynote LAN	516075	60.81	332602	39.19
ATTBI Network	429703	58.78	301370	41.22

Parent Cache Peering

- Proxy server peered into the NLANR Proxy Mesh
    - Caches used:

Palo Alto	pa.us.ircache.net
San Jose	sj.us.ircache.net
Silicon Valley	sv.us.ircache.net
San Diego	sd.us.ircache.net

- **http://www.ircache.net/** for more information

Parent Cache Peering, cont’d

- Proxy Misses are sent up to the parent caches for attempted retrieval or revalidation

- 35%-40% of local Cache Object Misses were retrieved or revalidated from the parent caches

- 20%-25% of Cache Miss Kilobytes were retrieved or revalidated from parent caches

Parent Cache Peering Test Results

	Cache Hits	%	Cache Misses	%	Cache Misses Served from Parent	%
Keynote LAN	191352	85.44	32594	14.55	n/a	n/a
Keynote LAN w/ Parent Caching	128805	85.87	21192	14.13	14378	41.11

	KB Hits	%	KB Miss	%	KB Misses Served from Parent	%
Keynote LAN	516075	60.81	332602	39.19	n/a	n/a
Keynote LAN w/ Parent Caching	357518	61.41	224625	38.59	56954	21.05

Log Analysis Online

AT&T Broadband Data:

http://www.pierzchala.com/cache_study/squid_home.html

Keynote LAN Data:

http://www.pierzchala.com/cache_study/squid_keyn.html

Keynote LAN With Parent Caches Data:

http://www.pierzchala.com/cache_study/squid_keyn_parent.html

Proxy Servers Work!

- The test results show that even with a limited subset of user visited Web sites, a large amount of content can be served from a proxy server

- Making pages and object explicitly cacheable ***where relevant*** can reap high returns in bandwidth preservation and server capacity

- A very inexpensive way to do more with less

Thank you

Questions?

Contact Info:

Stephen Pierzchala

stephen@pierzchala.com