B211 Internet Computing

Search Engines and Web Proxy Servers

Learning Objectives

1. Understand the role of search engines and web proxy servers.

2. Understand the technical operations of search engines and web proxy servers.

Lecture Outline

- Search Engines
  - Types of Search Engines
  - Components of Search Engines
  - How Search Engines create their indices
  - Power Searching
  - Current statistics on Search Engines
- Proxy Servers
  - What are web proxy servers?
  - How do proxy servers work?
  - Proxy Caching
  - Some example proxy server software

What are Search Engines?

- The most commonly used way to search for information on the web.
  - Any rich information source requires a way of easily searching for information - eg. encyclopedia, library.
- Search Engines keeps an indexed database of resources (mostly web pages) available on the web, and allows users to search this database using a web interface.
- Web sites with high listings in search engines get significantly more hits than the ones with lower listings.

Types of Search Engines

- The term “Search Engine” actually encompasses different types:
    - *Conventional Search Engines*
    - *Search Directories*
    - *Meta Search Engines*
- A lot of search engines today are combinations of the above (especially the first two).

Conventional Search Engines

- Conventional search engines obtain information on web-sites and pages by having programs (called *robots* or *spiders*) travel the web by following links.
- Some popular examples:
    - Google (http://www.google.com)
    - AltaVista (http://www.altavista.com.au)
    - Excite (http://www.excite.com)
    - InfoSeek (http://www.infoseek.com)
    - Lycos (http://www.lycos.com)

Components of a Search Engine

- *Robots, spiders* or c*rawlers* – automatically visits webpages, reads it and follow links to other pages.
- *Index* – the database which stores the information on the pages found by the robots
- *Search Engine Software* – for the interface and the process of user-requested searching.
- Different search engines implements the above in different ways.

Robots

- Implemented primarily as HTTP clients accessing pages on servers.
- How they work:
    1. Have a few predetermined web-page URLs to index
    2. Request those pages from their servers (just like any other normal web page access that browsers do)
    3. Index those pages by looking at their content.
    4. Generate a new set of URLs from links found in the downloaded pages.
    5. Repeat from (2).

Considerate Robots

- Search engines' robot programs usually have a special considerations like:
  - Avoid flooding a particular server with too many request at once - it usually takes a robot a few days to request all pages from a particular large site.
  - Avoid duplicate requests for the same page just because there are many links to that page
  - Avoid updating the same page too frequently - usually a robot has an algorithm (eg. how often has the page changed in the last few updates?) to work out how dynamic a page is, and returns at appropriate intervals.

Inconsiderate Robots

- If a search engine's robot does not have the above considerations, and makes too many requests to one web server, the web server administrator may block access from that robot.
  - Eg. by blocking all access from search engine site's IP address.

Robot Exclusion

- Robots may also be "requested" not to index a site by the web administrator of the site.
- Two common ways

1. The *Robot Exclusion Protocol*

2, Using the Robot META tag

The Robot Exclusion Protocol

- How it works:
  - Involves putting a *robot.txt* file (in plain ASCII text) in the main directory of the HTTP server (For example, when a robot first it visits a site ihaterobots.com.au, it will request and read the file http://ihaterobots.com.au/robot.txt)
  - The *robot.txt* file specifies which robot (ie. a specific client) should access which directories on the web server.
- The robot program may choose to ignore this file, but usually doesn't.

An example robot.txt file:

- The following file disallows all robots from visiting the site. It says:
  - for all web clients
  - do not traverse the top directory (and any directory below it)

```
User-agent: *
Disallow: /
```


Using the Robot META Tag

- Include a META tag in HTML pages
- Eg.

```
<META NAME="ROBOTS" CONTENT="NOINDEX">
```

- Many robots do not implement this, so there is a good chance it may be ignored.

Improving Ranking

- Search engines rank pages according to how they believe the pages are relevant to keywords
- They do not release their specific algorithms for determining ranking, but based on information supplied at the search engine sites, and studies correlating the rankings with the make-up of the pages, we can infer certain guidelines.

Guidelines for Improving Ranking

- Keywords appearing in titles, top of the page are weighted more.
- Keywords appearing in `<META NAME=… CONTENT=…>` tags can be useful, but only for selected search engines
  - These meta tags describes information in a form defined by what is called the *Dublin core*.
- Keywords appearing more often are weighted more…but almost all engines these days detect keyword *spamming* (needless repetitions of keywords) and penalise a page's ranking for doing it.

Guidelines for Improving Ranking (cont'd)

- Some engines (like Google) uses link analysis
  - How many other pages have links to this one? The more the better.
  - Google index pages linked by other pages even if it's robot hasn't visited it. It uses the words used by the links.
    - Eg. if a page links to "http://www.maths.org/" using the word "Mathematics", then Google can index "http://www.maths.org/" using the keyword "Mathematics" even if it hasn't visited the site yet.
- The above are only guidelines. In the end, there are NO GURANTEES!

Search Directories

- Search Directories create their listings by human manual entry into their databases. The entries are arranged in hierarchies of subject areas.
- Some popular examples:
  - Yahoo (http://www.yahoo.com)
  - LookSmart (http://www.looksmart.com)
  - eBlast (http:/www.eBlast.com)

Search Directories' Listings

- Pages are submitted manually using tools
  - eg. see listing at http://www.searchengines.com/URLsubmission.html
- Submissions are reviewed by editors.
  - Since there are no robots or spiders, no amount of META or ALT tags will make any difference in ranking.
  - The best strategy to get a high ranking is to describe the page as accurately as possible during manual submission.

Search Directories' Listings (cont'd)

- Some directories buys their indices from index generating services
  - eg. Yahoo buys their part of their index from Inktomi http://www.inktomi.com/)

Meta Search Engines

- Meta Search Engines search a few other search engines and returns the results of each.
- Some popular examples:
  - Dogpile (http://www.dogpile.com)
  - Metacrawler (http://www.metacrawler.com)
  - Beaucoup (http://www.beaucoup.com)

Power Searching

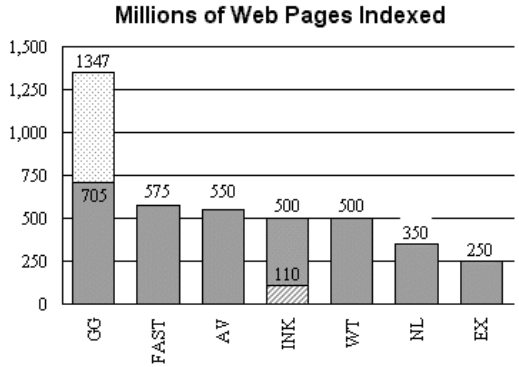
- Besides the standard keyword searches, most search engine interface offers power searching to help refine searches.
- Example power features:
  - Using boolean logic in search terms
    - Eg. “perl AND australia” looks for perl sites relevant to Australia.
  - Using special terms like include/exclude, any, all, etc
    - Eg. “+apache +modules -security” means to include pages on “apache” and “modules”, but exclude anything dealing with security

Power Searching (cont’d)

- Using wildcards like ? (for any character) or * (any sequence of characters
- Search pages within a certain date range
- Search titles of a page only instead of contents
- Display pages by relevancy, by dates, etc.

Search Engine Sizes

- Reported sizes of search engines (as of April 2001)
  - source: http://www.searchenginewatch.com/reports/sizes.html



Search Engine Sizes (cont'd)

- Key for previous graph:
  - GG=Google
  - WT=WebTop.com
  - AV=AltaVista
  - FAST=FAST,
  - NL=Northern Light
  - EX=Excite
  - INK=Inktomi
  - Go=Go (Infoseek)

Search Directory Sizes

- Reported sizes of search directories
  - http://www.searchenginewatch.com/reports/directories.html

Service	Editors	Subject Categories	Links...	As Of
Yahoo	100+	n/a	1.5-1.8 million	Aug-00
Open Directory	36,000	361,000	2.6 million	Apr-01
LookSmart	200	200,000	2 million	Aug-00
NBCi (Snap)	30	80,000	1.5 miliion	Dec-00

Does Size Really Matter?

- Search Engine sizes matters if searching for *obscure* keywords, since the larger coverage means *significantly* higher chance of including the required pages.

- But when searching *popular* keywords, the engines' sizes do not necessarily mean good results - it is the relevancy of the results that count
  - Does it really matter if the search engine returns you 50,000 hits rather than 25,000?
  - Wouldn't you rather get "good" top few hits?

Size vs Relevancy

- Examples of tests run on various search engines on different type of searches, see
  - http://www.searchenginewatch.com/reports/sizetest.html

References

- For materials and developments on search engines:
  - http://www.searchenginewatch.com/
- A useful page on how to get high search engine rankings for your page is
  - http://www.searchengines.com/
- Some further information on robot exclusion:
  - http://info.webcrawler.com/mak/projects/robots/exclusion.html

What are Web Proxy Servers?

- Web proxy servers are HTTP servers which allows clients from behind a firewall to access external servers.
- The proxy server can resides on the firewall machine, or in front of it - ie. on the other side of the firewall from the clients.

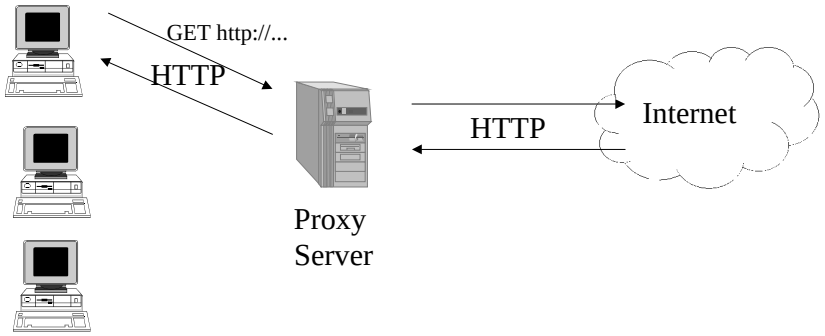
Why have a Proxy Server?

- A proxy server is useful to:
  - Allow clients on non-Internet machines (eg. not having an IP-address) to access the Internet
  - permit and restrict client access to the Internet
  - Cache retrieved documents from the Internet
  - Convert data to HTML format readable by a browser.

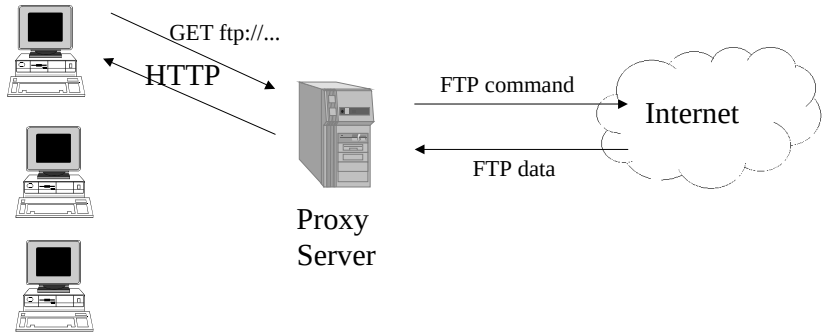
How a Proxy Server works:

- The process:
  - Client sends a request to the proxy server (usually on its own local LAN) using HTTP.
  - Proxy server retrieves the documents using an appropriate protocol (HTTP, FTP, gopher, etc), if not there already.
  - Proxy server sends the information back to the client in HTTP.

A HTTP Request using a Proxy Server:



An FTP Request using a Proxy Server:

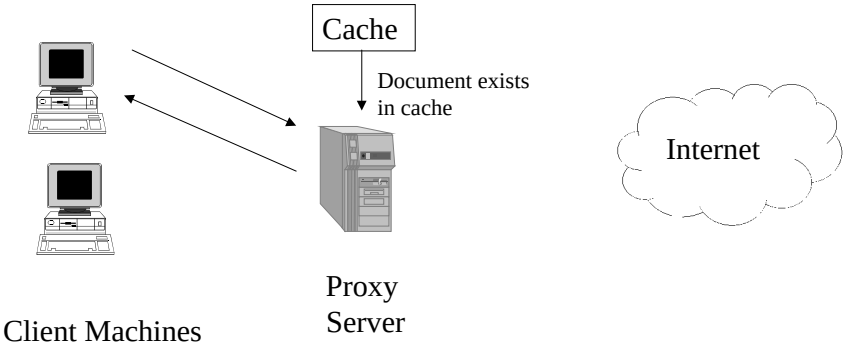


Caching Documents

- Proxy servers usually cache retrieved documents retrieved - for efficiency.
- Clients served by the proxy server gets sent the cache copies instead of retrieving the original copies again
- Proxy configured to decide:
  - Which documents are used frequently enough to justify caching?
  - How long to keep the documents in cache before fetching up-to-date copies?

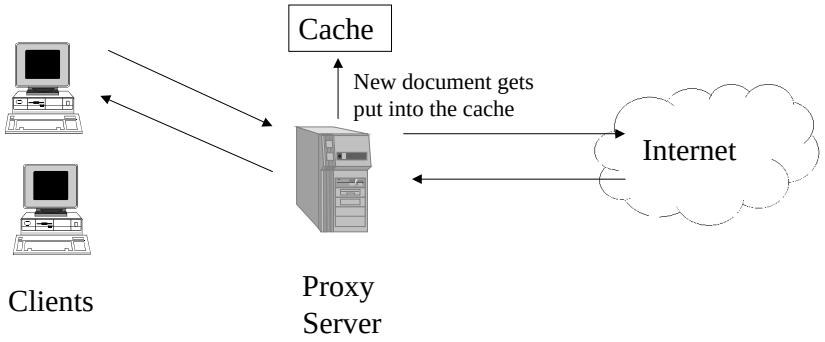
How Proxy Caching Works:

- If a document exists in a the proxy's cache:



How Proxy Caching Works:

- If a document doesn't exist in a the proxy's cache:



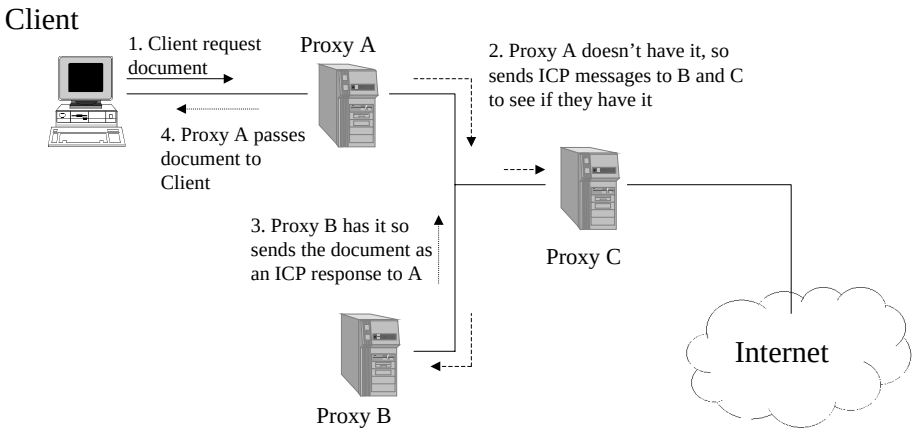
Proxy Server Caching

- Passive Caching
  - proxy server waits for local clients to make requests, then cache the appropriate retrieved documents
- Active Caching
  - during low activity, proxy server fetches documents it believes local clients would want
- In environments (LANs) with multiple proxies, protocols for managing information about their caches are implemented on the proxies:
  - Internet Cache Protocol (ICP)
  - Cache Array Routing Protocol (CARP)

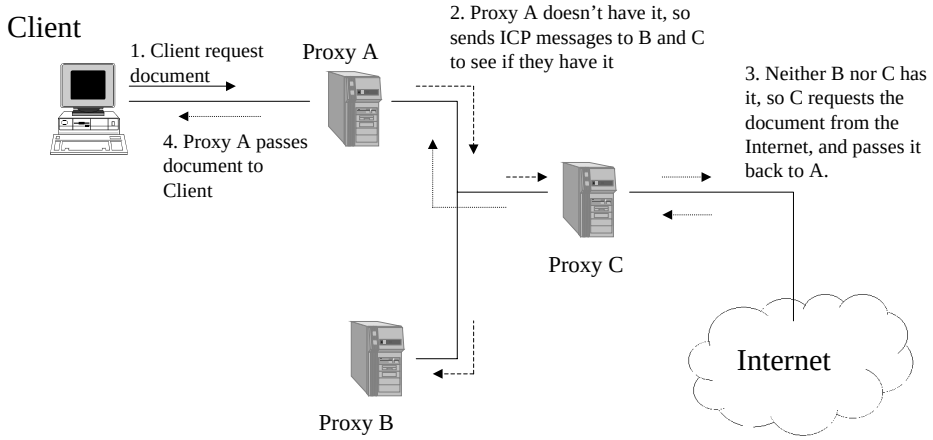
The Internet Cache Protocol (ICP)

- ICP specifies a message format for communication between proxies
- Exchange information about whether a certain page exists on a proxy's cache
- Proxies uses 3 ports:
  1. For HTTP requests by clients
  2. For exchanging ICP messages with other proxies
  3. For retrieving pages stored on another proxy's cache

An example of how ICP works:



Another example of ICP:



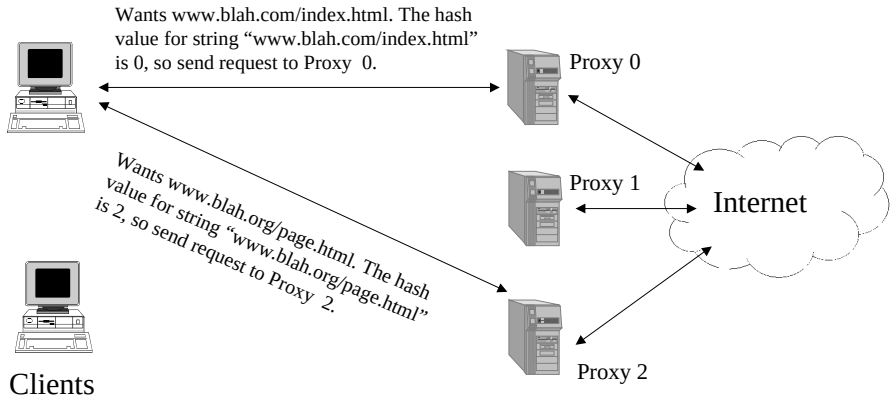
How CARP works:

- All proxies in the environment arranged as an "array".
- All URLs are designated to belong to one and only one proxy in this array, based on computing a hash function
- Clients determine which proxy should own the URL (by computing the hash function) and queries the particular proxy.

How CARP works:

- The specified proxy either
  - sends the requested document to the client if it has it in its cache, or
  - fetches the document from the URL address
- Advantages:
  - All clients can calculate exactly which proxy server handles which URL.
  - The load is distributed evenly (if the the hash function is good) between all proxy servers.

How CARP works:



Some Example Proxy Server Software:

- Squid
  - WWW proxy cache packages
  - Supports ICP and CARP
- Harvest
  - as above
- Microsoft Proxy Server
  - firewall and caching functionalities
- Netscape Proxy Server
  - Caching and virus scans