CM0133 Internet Computing

Search Engines

Objectives

- *Recap our understanding of the WWW and the Internet*
- *Understand how a search engine works*
  - *Spiders, crawlers*
  - *Indexing*
  - *Ranking*
- *Search API – Yahoo BOSS*
- *This lecture slides have been adapted from http://courses.ischool.berkeley.edu/i141/f07/schedule.html*

How Do Search Engines Work?

- Say a you are using your computer at home (or in the lab) and want to find information about course CM0133?
- What happens when you:
  - Bring up a search engine home page?
  - Types a query?
- First we have to understand how the network works !
- Then we can understand search engines.

Revision: How Does the WWW Work?

- Let's say you received email with the address for the CM0133 Internet Computing web page
- You go to a networked computer, and launch a web browser.
- You types the address, known as a URL, into the address bar of the browser.
- What happens next?

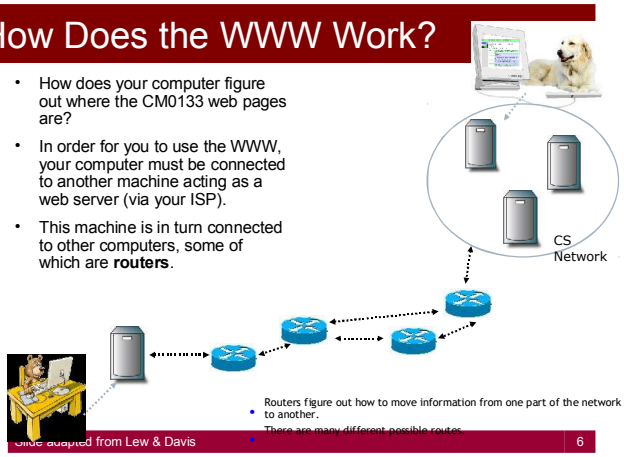
(URL stands for Uniform Resource Locator)

How Does the WWW Work?

- Say Florian Twaroch has written some web pages for the class CM0133 Internet Computing on his PC.
- He copied the pages to a directory on a computer of his local network at computer science in Cardiff.
- This computer runs a web server program, e.g. Apache.

How Does the WWW Work?

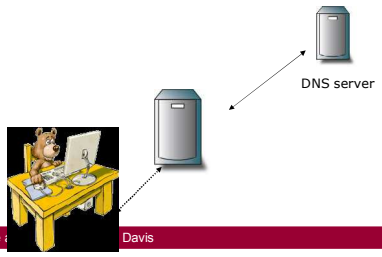
- How does your computer figure out where the CM0133 web pages are?
- In order for you to use the WWW, your computer must be connected to another machine acting as a web server (via your ISP).
- This machine is in turn connected to other computers, some of which are **routers**.



- Routers figure out how to move information from one part of the network to another.
- There are many different possible routes.

How Does the WWW Work?

- How does your server and the routers know how to find the right server?
- First, the URL has to be translated into a number known as an IP address.
- Your server connects to a Domain Names Server (DNS) that knows how to do the translation.

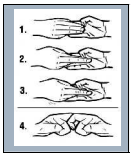


Converting Domain Names

- Domain names are for humans to read.
- The Internet actually uses numbers called **IP addresses** to describe network addresses.
- The Domain Name System (DNS) – resolves IP addresses into easily recognizable names
- For example:
  - 12.42.192.73 = *www.xyz.com*
- A domain name and its IP address refer to the same Web server.

How the Internet Works

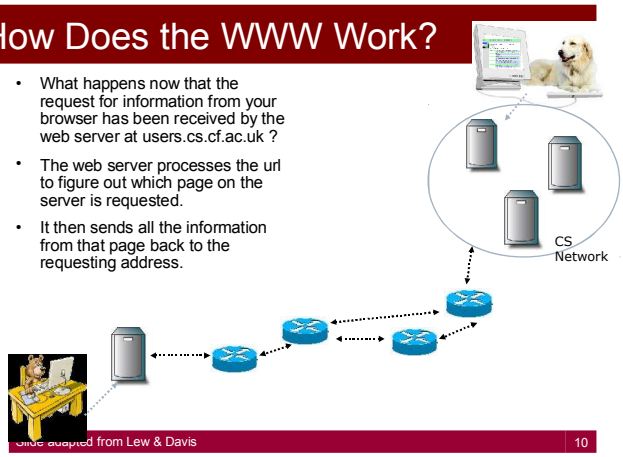
- Network Protocols:
  - Protocol – an agreed-upon format for transmitting data between two devices
    - Like a secret handshake
  - The Internet protocol is TCP/IP
  - The WWW protocol is HTTP



- Network Packets:
  - Typically a message is broken up into smaller pieces and re-assembled at the receiving end.
  - These pieces of information, surrounded by address information are called **packets.**

How Does the WWW Work?

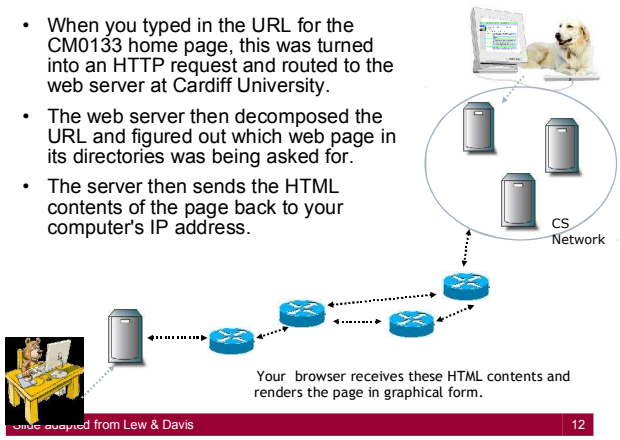
- What happens now that the request for information from your browser has been received by the web server at users.cs.cf.ac.uk ?
- The web server processes the url to figure out which page on the server is requested.
- It then sends all the information from that page back to the requesting address.



HTTP

- HTTP is the protocol used by the WWW
- When a user clicks on a hyperlink in their web browser, this sends an HTTP command to the Web server named in the URL
- This command usually is to “GET” the contents of the web page and return them to the user’s browser.
- It is a very simple protocol
  - It relies on the TCP/IP functionality

- When you typed in the URL for the CM0133 home page, this was turned into an HTTP request and routed to the web server at Cardiff University.
- The web server then decomposed the URL and figured out which web page in its directories was being asked for.
- The server then sends the HTML contents of the page back to your computer's IP address.



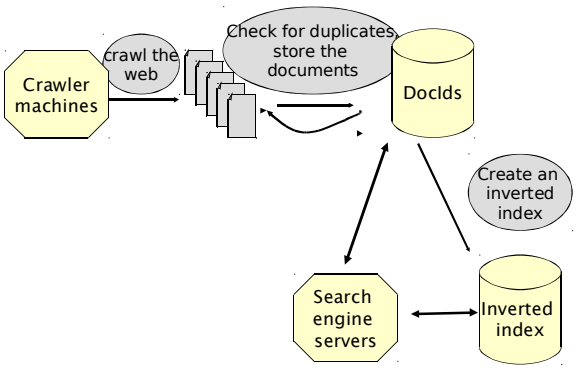
Your browser receives these HTML contents and renders the page in graphical form.

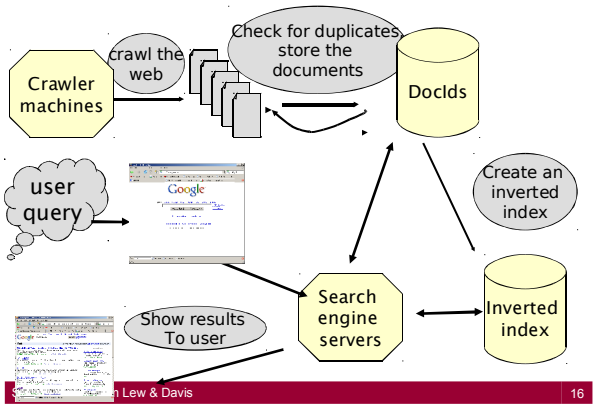
How Search Engines Work

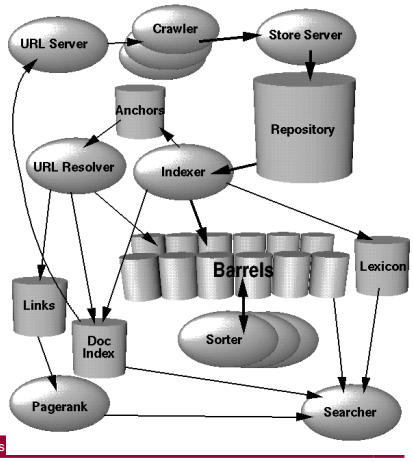
- There are MANY issues
- I'm only giving the basics today
- *This lecture slides have been adapted from* *http://courses.ischool.berkeley.edu/i141/f07/schedule.html*
- *Much more can be found on above pages*

How Search Engines Work

1) Gather the contents of all web pages (using a program called a **crawler** or **spider**)
2) Organize the contents of the pages in a way that allows efficient retrieval (**indexing**)
3) Take in a query, determine which pages match, and show the results (**ranking** and **display** of results)







1. SPIDERS / CRAWLERS

Spiders / Crawlers

- How to find web pages to visit and copy?
  - Can start with a list of domain names, visit the home pages there.
  - Look at the hyperlink on the home page, and follow those links to more pages.
    - Use HTTP commands to GET the pages
  - Keep a list of urls visited, and those still to be visited.
  - Each time the program loads in a new HTML page, add the links in that page to the list to be crawled.

Spider behaviour varies

- Parts of a web page that are indexed
- How deeply a site is indexed
- Types of files indexed
- How frequently the site is spidered

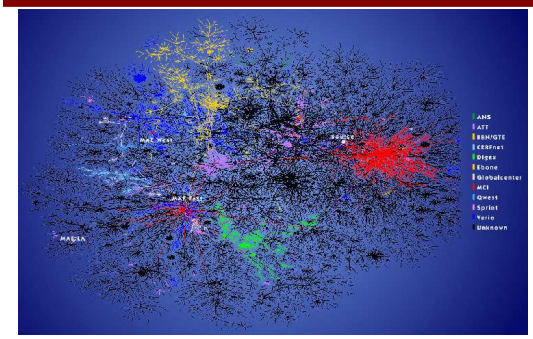
Four Laws of Crawling

- A Crawler must show identification
- A Crawler must obey the robots exclusion standard
  http://www.robotstxt.org/wc/norobots.html
- A Crawler must not hog resources
- A Crawler must report errors

Lots of tricky aspects

- Servers are often down or slow
- Hyperlinks can get the crawler into cycles
- Some websites have junk in the web pages
- Now many pages have dynamic content
  - The “hidden” web
  - E.g., schedule.berkeley.edu
    - You don’t see the course schedules until you run a query.
- The web is HUGE

The Internet Is Enormous



From http://www.nature.com/nature/webmatters/tomog/tomfigs/fig1.html

“Freshness”

- Need to keep checking pages
  - Pages change (25%,7% large changes)
    - At different frequencies
    - Who is the fastest changing?
    - Pages are removed
  - Many search engines **cache** the pages (store a copy on their own servers)

What really gets crawled?

- A small fraction of the Web that search engines know about; no search engine is exhaustive
- Not the “live” Web, but the search engine’s index
- Not the “Deep Web”
- Mostly HTML pages but other file types too: PDF, Word, PPT, etc.

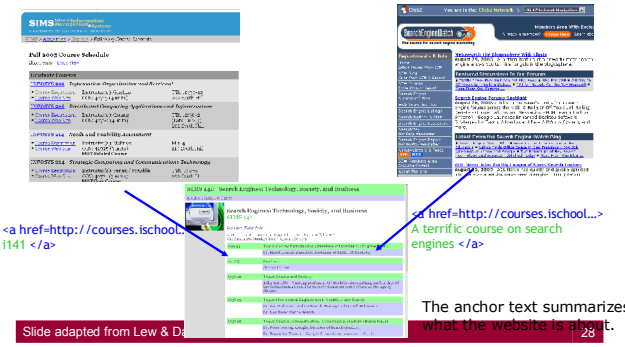
2. INDEXING

Index (the database)

Record information about each page

- List of words
  - In the title?
  - How far down in the page?
  - Was the word in boldface?
- URLs of pages pointing to this one
- Anchor text on pages pointing to this one

The importance of anchor text



The anchor text summarizes what the website is about.

Inverted Index

- How to store the words for fast lookup
- Basic steps:
  - Make a “dictionary” of all the words in all of the web pages
  - For each word, list all the documents it occurs in.
  - Often omit very common words
    - **“stop words”**
  - Sometimes **stem** the words
    - (also called **morphological analysis**)
    - cats -> cat
    - running -> run

Inverted Index Example

- T_0 = "it is what it is"
- T_1 = "what is it"
- T_2 = "it is a banana"

```
"a":      {2}
"banana": {2}
"is":     {0, 1, 2}
"it":     {0, 1, 2}
"what":   {0, 1}
```

A term search for the terms "what", "is" and "it" would give the set :

$$\{0,1\} \cap \{0,1,2\} \cap \{0,1,2\} = \{0,1\}$$

http://en.wikipedia.org/wiki/Inverted_index

Inverted Index Example

- With the same texts, we get the following full inverted index
  - pairs are document numbers and local word numbers
  - "banana": {(2, 3)} means the word "banana" is in the third document (T2), and it is the fourth word in that document (position 3).

```
"a":      {(2, 2)}
"banana": {(2, 3)}
"is":     {(0, 1), (0, 4), (1, 1), (2, 1)}
"it":     {(0, 0), (0, 3), (1, 2), (2, 0)}
"what":   {(0, 2), (1, 0)}
```

If we run a phrase search for "what is it" we get hits for all the words in both document 0 and 1. But the terms occur consecutively only in document 1.

Inverted Index Example

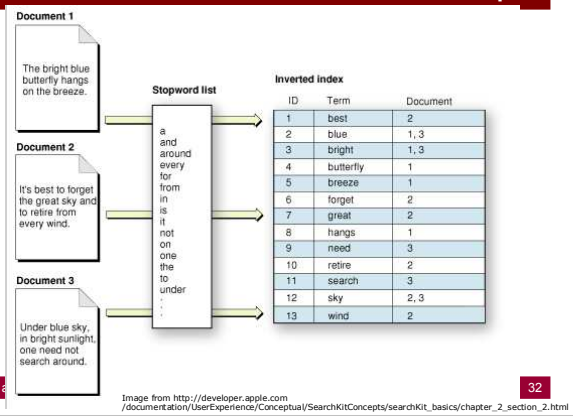


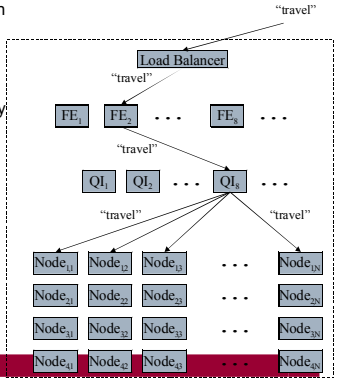
Image from http://developer.apple.com /documentation/UserExperience/Conceptual/SearchKitConcepts/searchKit_basics/chapter_2_section_2.html

Inverted Index

- In reality, this index is HUGE
- Need to store the contents across many machines
- Need to do optimization tricks to make lookup fast.

Query Serving Architecture

- Index divided into segments each served by a node
- Each row of nodes replicated for query load
- Query integrator distributes query and merges results
- Front end creates a HTML page with the query results



3. RANKING

Results Ranking

- Search engine receives a query, then
- Looks up the words in the index, retrieves many documents, then
- Rank orders the pages and extracts "snippets" or summaries containing query words.
  - Most web search engines assume the user wants all of the words (Boolean AND, not OR).
- These are complex and highly guarded algorithms unique to each search engine.

Some ranking criteria

- For a given candidate result page, use:
  - Number of matching query words in the page
  - Proximity of matching words to one another
  - Location of terms within the page
  - Location of terms within tags e.g. <title>, <h1>, link text, body text
  - Anchor text on pages pointing to this one
  - Frequency of terms on the page and in general
  - Link analysis of which pages point to this one
  - (Sometimes) Click-through analysis: how often the page is clicked on
  - How "fresh" is the page
- Complex formulae combine these together.

Measuring Importance of Linking

- PageRank Algorithm
  - Idea: important pages are pointed to by other important pages

  - Method:
    - Each link from one page to another is counted as a "vote" for the destination page
    - But the importance of the starting page also influences the importance of the destination page.
    - And those pages scores, in turn, depend on those linking to them.

Image and explanation from http://www.economist.com/science/tq/displayStory.cfm?story_id=3172188

Measuring Importance of Linking

- Example: each page starts with 100 points.
- Each page's score is recalculated by adding up the score from each incoming link.
  - This is the score of the linking page divided by the number of outgoing links it has.
  - E.g, the page in green has 2 outgoing links and so its "points" are shared evenly by the 2 pages it links to.
- Keep repeating the score updates until no more changes.

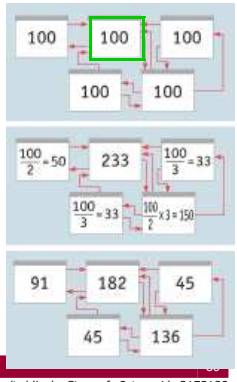


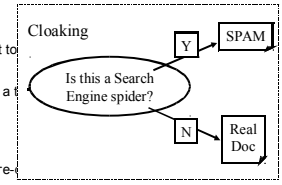
Image and explanation from http://www.economist.com/science/tq/displayStory.cfm?story_id=3172188

Manipulating Ranking

- Motives
  - Commercial, political, religious
  - Promotion funded by advertising budget
- Operators
  - Search Engine Optimizers
  - Web masters
  - Hosting services
- Forum
  - Web master world (www.webmasterworld.com)

A few spam technologies

- **Cloaking**
  - Serve fake content to search engine robot
  - *DNS cloaking:* Switch IP address. Impersonate
- **Doorway pages**
  - Pages optimized for a single keyword that re-direct to
- **Keyword Spam**
  - Misleading meta-keywords, excessive repetition of a
  - Hidden text with colors, CSS tricks, etc.
- **Link spamming**
  - Mutual admiration societies, hidden links, awards
  - *Domain flooding:* numerous domains that point or re-
- **Robots**
  - Fake click stream
  - Fake query stream
  - Millions of submissions via Add-Url



Meta-Keywords =
"… London hotels, hotel, holiday inn, hilton, discount, booking, reservation, sex, mp3, britney spears, viagra, …"

Paid ranking

Pay-for-inclusion

- Deeper and more frequent indexing
- Sites are not distinguished in results display

Paid placement

- Keyword bidding for targeted ads

Know your search engine

- What is the default boolean operator? Are other operators supported?
- Does it index other file types like PDF?
- Is it case sensitive?
- Phrase searching?
- Proximity searching?
- Truncation?
- Advanced search features?

Keyword search tips

- There are many books and websites that give searching tips; here are a few common ones:
  - Use unusual terms and proper names
  - Put most important terms first
  - Use phrases when possible
  - Make use of slang, industry jargon, local vernacular, acronyms
  - Be aware of country spellings and common misspellings
  - Frame your search like an answer or question
- For more, see http://www.googleguide.com/

Links

- www.searchengineland.com
- www.searchenginewatch.com
- www.searchenginejournal.com
- www.searchengineshowdown.com
- http://battellemedia.com
- http://cs.nyu.edu/courses/fall02/G22.3033-008/lec1.html
- http://cs.nyu.edu/courses/fall02/G22.3033-008/
- http://courses.ischool.berkeley.edu/i141/f07/schedule.html