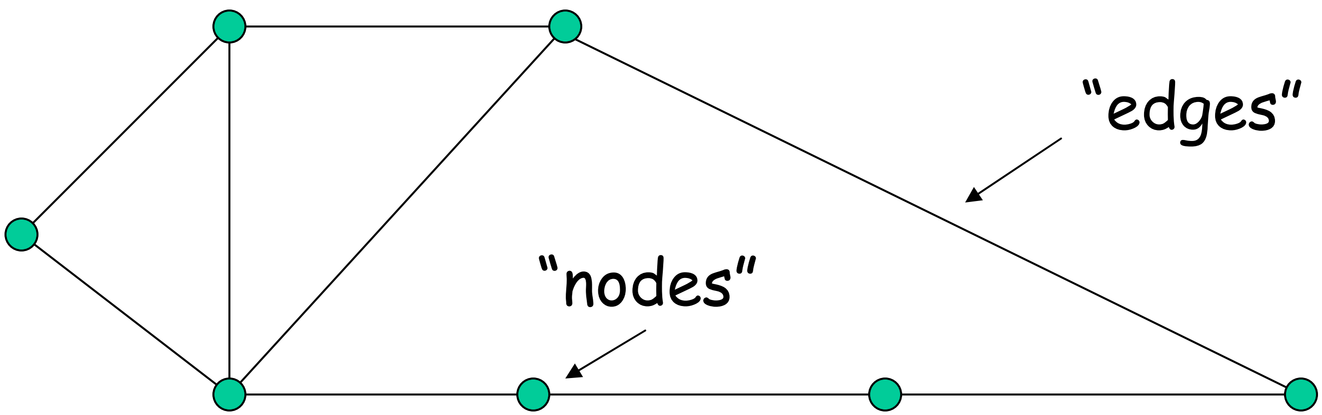
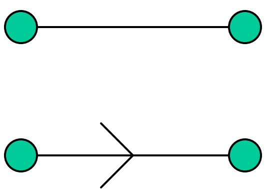
CPSC156: The Internet
Co-Evolution of Technology and Society

Lecture 10: February 15, 2007

Search

Graphs: An Important Abstraction





: Bidirectional "edges"

: Directed "arcs" or "links"

Graphs with directed links are called "digraphs."

Digraphs are Ubiquitous in Computer Science

- Used as *models* of systems
  - Nodes represent *components*.
  - Links represent *interactions* or *relationships*.
- Examples we've seen in CPSC156a:
  - Computer networks: Nodes represent computers (*e.g.*, hosts or routers), and links represent direct ("hardwired") connections.
  - The WWW: Nodes represent web pages, and links represent ... "links" (*e.g.*, html code pointing from one page to another).

Two Aspects of WWW Searching

- Analyze *contents* of pages
  - Text (*e.g.*, search terms)
  - Structure (*e.g.*, HTML tags)
- Analyze *structure* of WWW digraph
  - Links to page P indicate *interest* in the contents of of P.
  - *Importance* depends on *who* is interested.
  - Requires *global* analysis of digraph.

The WWW Digraph

- Massive, ***Distributed***, ***Explicit*** Digraph
- Many Billions of Nodes (Pages)
- Sparse: Average Degree (links per page) is 5-15.
- Can be *crawled* (*i.e.*, every node visited) in time linear in the total number of links (using classical methods).

“Hot” Research Area

- Graph Representation
- Duplicate Elimination
- Clustering
- Ranking Search Results

Finding Information on the Internet

The Internet is so successful partly because it is so easy to publish information on the World Wide Web.

- No central authority on what pages exist, where they exist, or when they exist.
- Too much to sort through, anyway.
- **Question: How do we find what we need on the web?**

WWW Search Engines

- **Answer:** Set up websites that people can use to search for information by performing a ***search query***.
- Not such an easy solution! In addition to the technical problems, we have these business questions:
  - How do people know about the search engine websites?
  - How do you make money off of this? (Especially now that the service is free.)

Solutions (?) to Technical Problems

- How do we keep track of what pages are on the WWW?
  - Have a *crawler* or *spider* scan the web and links between pages to find new, updated, and removed pages.
- How do we store the content we find?
  - Design a way to map keywords in queries to documents so we can return a *usefully ordered list* to the user.
- What happens when pages are temporarily unavailable?
  - Use *caching*: keep a local copy of documents as we crawl the web. (Need lots of space!)

Solutions (?) to Technical Problems *(continued)*

- How do we store all the information?
  - Use a large network of disks (and maybe a clever method of compression) that can be easily searched.
- How do we handle so many different requests?
  - Use a *cluster* of computers that work together to process queries.

There is still ongoing research to find better ways to solve these problems!

Google History

- 1998: Founders Larry Page and Sergey Brin (Ph.D. students at Stanford) raise $1 million from family, friends, and angel investors. Google is incorporated Sept. 7. Site receives 10,000 queries per day and is listed in PC Magazine's top 100 search websites list.
- 1st half 1999: Google has 8 employees and answers 500,000 queries/day. Red Hat (Linux distributor) becomes first customer. Google gets $25 million equity

Google History (2)

- 2nd half 1999: 39 employees, 3 million queries/day. Partners with Virgilio of Italy to provide search services.

- 2000: Becomes largest web search engine, having indexed 1 billion documents. Answers 18 million queries/day. Gains more partners, including Yahoo! Starts web directory.

Google History (3)

- 2001: Acquires Deja.com's Usenet archive, adding newsgroups to Google's index. Improves and adds services including browser plug-ins, image searching, PDF searching, cell-phone and handheld compatibility, and queries and document searches in many languages. Advertising services used by over 350 Premium Sponsorship customers.

- Spring 2003: 3.3 billion web pages, 800 million newsgroup messages, and 425 million images indexed. Serves 200 million queries/day.

Google's Business Model

Scalable Search Services:

- Google provides customized search services for websites.
- Has become the dominant search engine, used by many portal and ISP websites as well as individuals.

Advertising:

- *Premium Sponsorship:* sponsored text links separated from search results; based on search category.
- *AdWords:* keyword-targeted, self-service advertising method. Choose keywords or phrases where text ads will appear to the right of the search result list.
- No banner ads or graphics!

Technical Highlights

- PageRank Technology: Linear-algebraic, objective calculations of the "importance" of a webpage.
  - Link from Page A to Page B is a "vote" for B.
  - Importance of A is factored into the vote.
  - Page owners cannot pay to have their PageRanks modified. (Note the difference between buying a "sponsored link" and getting a higher PageRank.)
  - Google employees can modify a PageRank in exceptional circumstances (*e.g.*, security threats).

Technical Highlights (2)

- Readings on how PageRank works:

  http://www.google.com/technology/index.html

  "Google's PageRank explained, and how to make the most of it," by P. Craven. http://www.webworkshop.net/pagerank.html

- **Hypertext-Matching Analysis**: The HTML tags are taken into account when examining the contents of a page. Headings, fonts, positions, and content of neighboring pages influence the analysis.

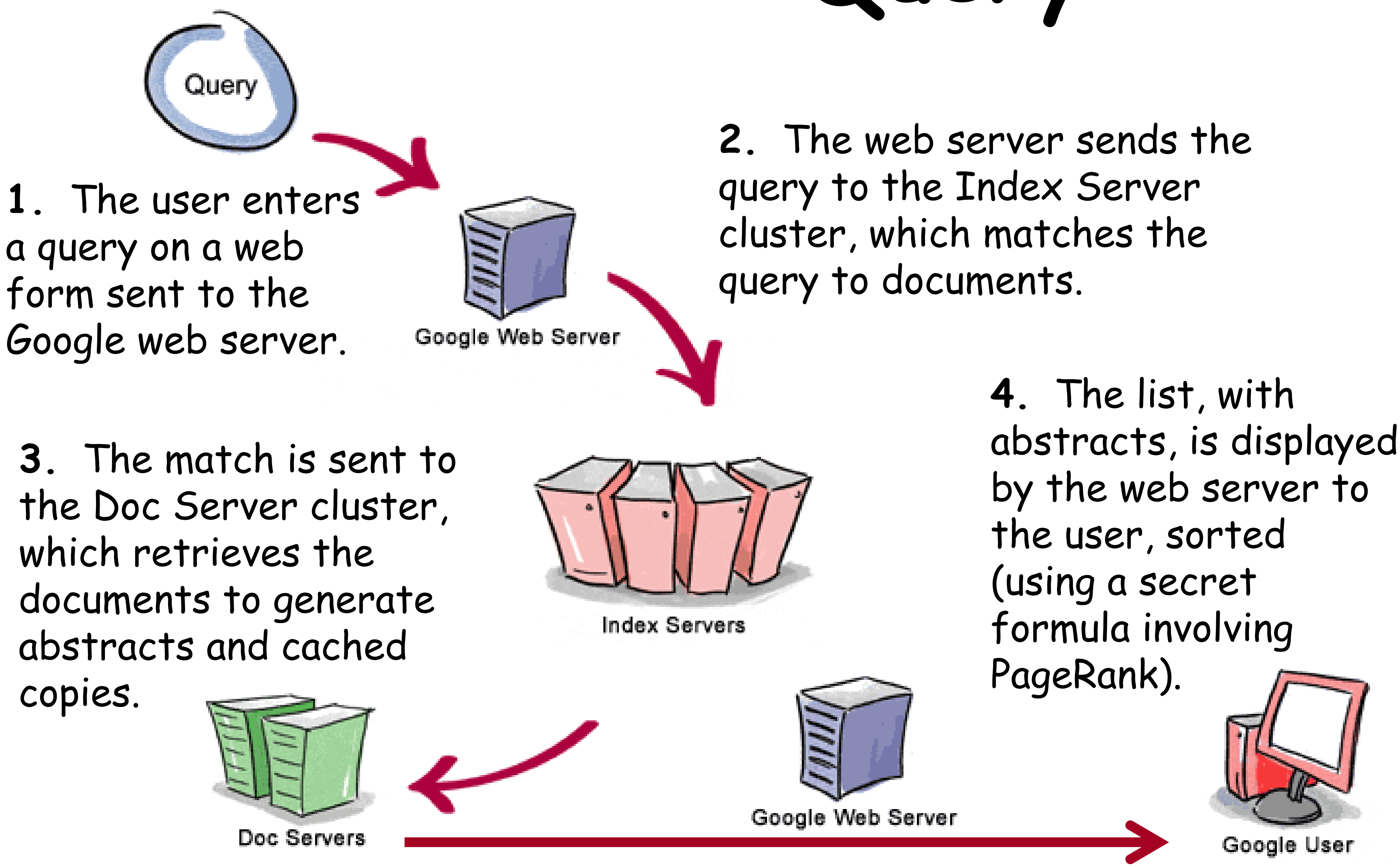
Technical Highlights (3)

- **Scalable Core Technology:** Calculations are performed by the largest commercial Linux cluster of over 10,000 servers.
  Can grow with the Internet!
- **Complex-File Searching:** Google can now index files in “non-Internet” formats, *e.g.*:
  - PostScript, PDF (Adobe)
  - Word, Excel, PowerPoint, Works (Microsoft)
  - WordPro, 1-2-3 (IBM/Lotus SmartSuite)
  - MacWrite
  - Rich Text (RTF), plain text

Technical Highlights (4)

- **Bayesian Spelling-Suggestion Program:** Offers suggestions for misspelled words in queries, making searching easier. ("*Did you mean…?*")
- **Internationalization:**
  - Google is developing technology to index pages with complex scripts, *e.g.*:
    - Some East Asian languages have no spaces between words.
    - Hebrew and Arabic are written right-to-left; Chinese is sometimes top-to-bottom.
  - Google has a translation engine and provides its interface in many languages.
  - Current research question: How to detect the language(s) of a page?

Life of a Query



Hub-and-Authority Framework

The next eight slides are a linear-algebraic interlude for mathematically inclined students. They are not required reading for CPSC 156.

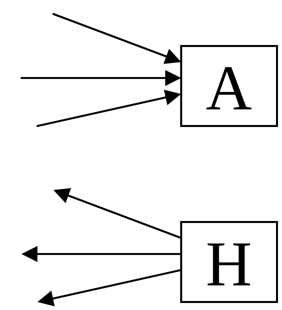
The Hub-and-Authority Framework

- Linear-algebraic interlude for technically minded students.
- NOT required for the exam!
- Introduced simultaneously with Google's PageRank.
  - Like PageRank, uses "wisdom" implied by WWW links.
  - Like PageRank, has provable mathematical properties.
  - Specific algorithm differs from that of PageRank.
- Invented by Jon Kleinberg, then at IBM, now at Cornell.
- See http://www.cs.cornell.edu/home/kleinber/ for many related papers.

"Abundance" Problem

http://www.cs.cornell.edu/home/kleinber/auth.pdf

- Given a query find:
  - Good Content ("Authorities")
  - Good Sources of Links ("Hubs")
- Mutually Reinforcing
- Simple (Core) Algorithm



$$T \hat{=} \{n \text{ Pages}\},\ A \hat{=} \{\text{Links}\}$$

$X_p \in \Re^{\geq 0},\ p \in T$ non-negative "Authority Weights"

$Y_p \in \Re^{\geq 0},\ p \in T$ non-negative "Hub Weights"

I operation Update Authority Weights

$$X_p \leftarrow \sum_{(q,p) \in A} Y_q$$

O operation Update Hub Weights

$$Y_p \leftarrow \sum_{(p,q) \in A} X_q$$

Normalize: $\sum_{p \in T} X_p^2 = \sum_{p \in T} Y_p^2 = 1$

Core Algorithm

Z ← (1,1,…,1)

X ← Y ← Z

Repeat until Convergence

- Apply I /* Update Authority weights */
- Apply O /* Update Hub Weights */
- Normalize

Return Limit (X*, Y*)

Convergence of $(X^i, Y^i) \triangleq (OI)^i(Z,Z)$

$A \triangleq$ n x n "Adjacency Matrix"

Rewrite I and O:

$X \leftarrow A^T Y$; $Y \leftarrow AX$

$X^i = (A^T A)^{i-1} A^T Z$; $Y^i = (AA^T)^i Z$

AA^T Symm., Non-negative and Z = (1,1,..., 1) $\Rightarrow$

$$X^* \triangleq \lim_{i \to \infty} X^i = \omega_1(A^T A)$$

$$Y^* \triangleq \lim_{i \to \infty} Y^i = \omega_1(AA^T)$$

Whole Algorithm (k,d,c)

q ⇒ Search Engine ⇒ |S| ≤ k

Base Set T:

(In S, S → , → S) and ≤ d links/page

Remove "Internal Links"

Run Core Algorithm on T

From Result (X,Y), Select

C pages with max X* values

C pages with max Y* values

Examples (k= 200, d=5)

q = censorship + net

- www.EFF.org
- www.EFF.org/BlueRib.html
- www.CDT.org
- www.VTW.org
- www.ACLU.prg

q = Gates

- www.roadahead.com
- www.microsoft.com
- www.ms.com/corpinfo/bill-g.html

[Compares well with Yahoo!, Galaxy, etc.]

Approach to "Massiveness": Throw Out Most of G!!

- Non-principal Eigenvectors correspond to "Non-principal Communities"
- Open (?):

  Objective Performance Criteria

  Dependence on Search Engine

  Nondeterministic Choice of S and T

Assignments

- Written assignment due February 22, 2007 (http://zoo.cs.yale.edu/classes/cs156/assignments/assignment3.html)
- Reading assignment:
- http://www.newyorker.com/printables/fact/070205fa_fact_toobin
- http://aei-brookings.org/admin/authorpdfs/page.php?id=1251
- http://www.policybandwidth.com/doc/googleprint.pdf