CS4051

Information Retrieval

Week 14

Muhammad Rafi

May 06, 2024

Web Search Basics & Web Crawling

Chapter No. 19 and 20

Today's Agenda

- Web Search basic
- Background & History
- Web Characteristics
- The search user experience
- Economic Models
- Index Estimates
- Duplicate detection
- Web Crawler
- Conclusion

Web Search Basics

1. Top links to specialized searches
2. Search box
3. Click to search
4. Click to retrieve a single result
5. Link to Advanced Search
6. Click to set search preferences
7. Link to Google's language tools
8. Click to set Google as your browser home page

Web Search Basics

- Internet is a Client Server Architecture provides a bunch of services.
- Client
  - The client – generally a browser, an application within a graphical user environment
- Server
  - The server communicates with the client via a protocol HTTP
  - It is lightweight and simple, asynchronously carrying a variety of payloads (text, images and – over time – richer media such as audio and video files) encoded in a simple markup language called HTML (for hypertext markup language)
  - HTML – It is a markup language for the web. Connect different pages and content

Web Search – Client Server

- Browser
  - The first browser was developed by Tim Berners-Lee in 1990- very limited functionality
  - Mosaic was first GUI based browser in 1993 by Marc Andreesen
  - Marc started Netscape in 1994 and launch Netscape Navigator
  - Microsoft started IE in 1995 for free. 95% market share in 2002
  - Marc started Mozilla foundation and started Firefox in 2004 reached 23% market share in 2011

Web Search – Client Server

- HTTP
  - HTTP is an application protocol for distributed, collaborative, and hypermedia information systems.
  - HTTP/2, was standardized in 2015, and is now supported by major web servers and browsers.
  - HTTP Header contains a lot of fields for effective transfer of information.

Web Search – Client Server

Comparison of protocol stack changes delivered with each new version after HTTP/1.0

HTTP/1.1	HTTP/2	HTTP/3
• Some methods and response codes are added. • "Keep-Alive" becomes officially supported. "Host" header becomes supported for Virtual Domain. • Syntax and semantics are separated.	• Support of parallel request transmission by "stream" (elimination of ***HTTP*** HoL Blocking). • Addition of flow-control and prioritization function in units of "stream". • Addition of server-push function (send related file without request.)	• Lower protocol changes from TCP+TLS to UDP+QUIC • Streams and flow-control function are moved to QUIC. • Parallel request transmission is supported by QUIC stream (eliminating ***TCP*** HoL Blocking).

Web Search – Client Server

- HTTP
  - There are five groups of status codes which are grouped by the first digit:

    1xx—Informational.

    2xx—The request was successful.

    3xx—The client is redirected to a different resource.

    4xx—The request contains an error of some kind.

    5xx—The server encountered an error fulfilling the request.

Web Search – Client Server

- HTTPS
  - The secure version of HTTP protocol is HyperText Transfer Protocol Secure.
  - In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS) or Secure Sockets Layer (SSL)
  - Benefits of HTTPS
    - Customer information, like credit card numbers and other sensitive information, is encrypted and cannot be intercepted.
    - Visitors can verify you are a registered business and that you own the domain.
    - Customers know they are not suppose to visit sites without HTTPS, and therefore, they are more likely to trust and complete purchases from sites that use HTTPS.

Web Search – Client Server

- HTML
  - HTML 2.0 -1995; HTML 3.0 1997; HTML 4.0 1997
  - HTML 5.0 2014; XHTML vs. XML
- Server Side Scripting
  - A number of server side scripting available.
- Client Side Scripting
  - Generally UI and interaction with local machine, mostly Java Script
- Cascading Style Sheet (CSS)
  - CSS is a language that describes the style of an HTML document.

Web Search – Client Server

- HTTP Injection
  - HTML Injection also known as Cross Site Scripting. It is a security vulnerability that allows an attacker to inject HTML code into web pages that are viewed by other users.
  - HTTP Response Splitting
    - Web Application Vulnerability
    - Web Cache poisoning
    - Cross-User Defacement
  - HTTP Cross Site Scripting
  - Session Fixation

Client-Side Vs. Server Side Scripting

Difference between client-side scripting vs. Server side scripting

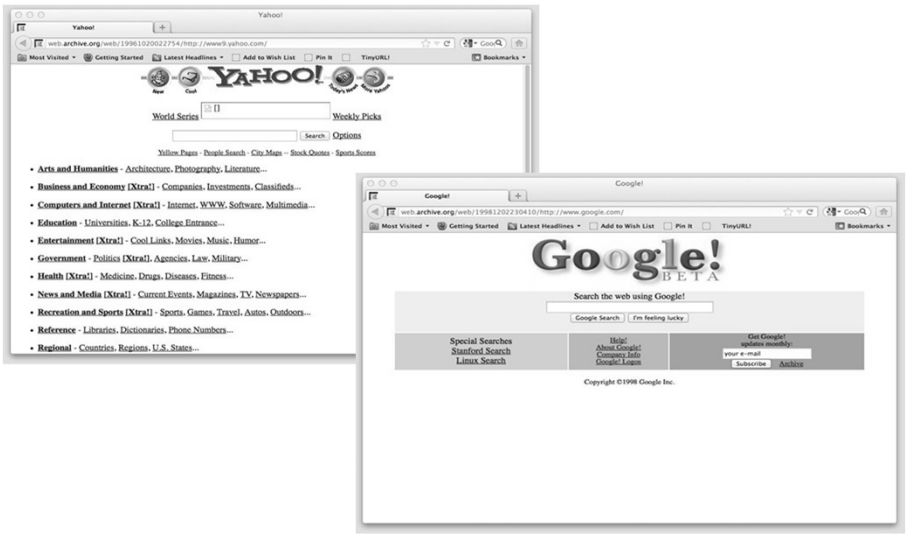
Client Side Scripting	Server Side Scripting
The client-side environment used to run scripts is usually a browser.	The **server-side environment** that runs a scripting language is a web server.
The source code is transferred from the web server to the user's computer over the internet and run directly in the browser.	A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser.
Advantages to client-side scripting including faster response times, a more interactive application, and less overhead on the web server.	The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores.
The Disadvantages of client-side scripting are that scripting languages require more time and effort, while the client's browser must support that scripting language.	The disadvantage of server-side processing is the page postback: it can introduce processing overhead that can decrease performance and force the user to wait for the page to be processed and recreated. Once the page is posted back to the server, the client must wait for the server to process the request and send the page back to the client.

Example
`<script>`
`document.getElementById('hello').innerHTML = 'Hello';`
`</script>`
	Example:
	`<h1 id="hello"><?php echo 'Hello'; ?></h1>`

Web Information Discovery

- Directories
  - Taxonomies populated with web pages in categories, such as Yahoo!
  - The user to browse through a hierarchical tree of category labels.
- Search Engines
  - Full-text index search engines such as Altavista, Excite and Infoseek
  - The user with a keyword search interface supported by inverted indexes and ranking mechanisms.

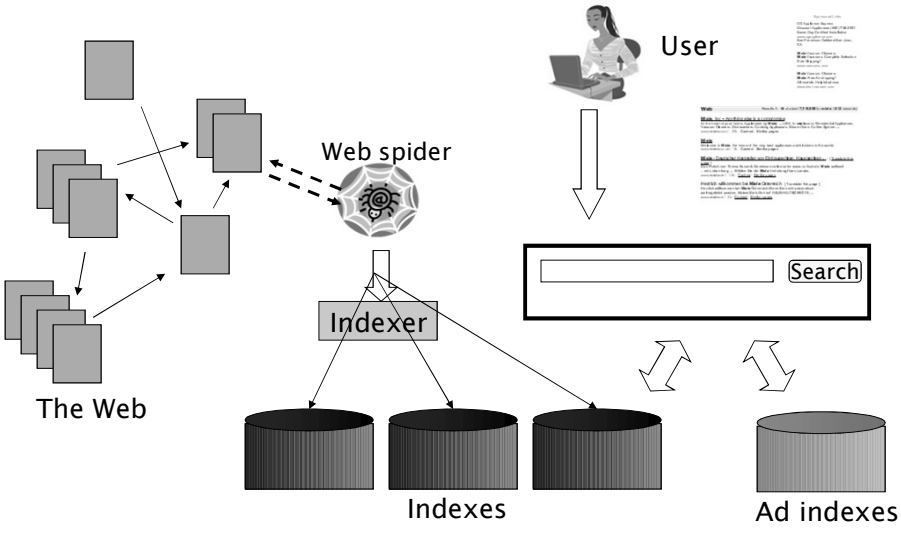
Web Information Discovery



Directories Vs. Search Engines

- **A directory allows you to explore and get what you want eventually.**
- **Use a directory to find cooking-related websites.**
- **Use a directory to find travel guides in a country.**
- **A search engine brings you to the exact page on the words or phrases you are looking for.**
- **Use a search engine to find a specific recipe, by providing the name of the ingredients.**
- **Use a search engine to find the transport trains schedule in Germany**

Web Search

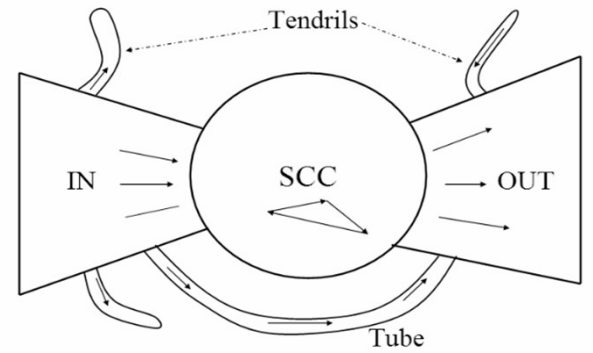


Web Characteristics

- Web User Interaction
- Web as a Graph
- Web Spam

Web Characteristics

- Web as a Graph

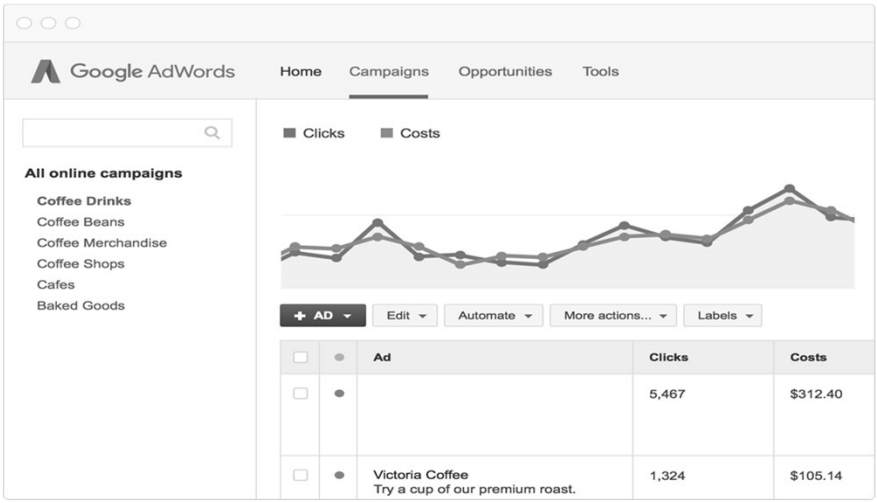


  - There are three major categories of web pages that are IN, OUT and SCC

Web Economic Model

- Advertisement Model for Revenue
- Unit of Measurement
  - CPM, CPC, CPI, CPD, CPP
- Complex Advertisement Models
  - AdWords
    - Ads
    - Search terms
    - Daily budget

AdWords



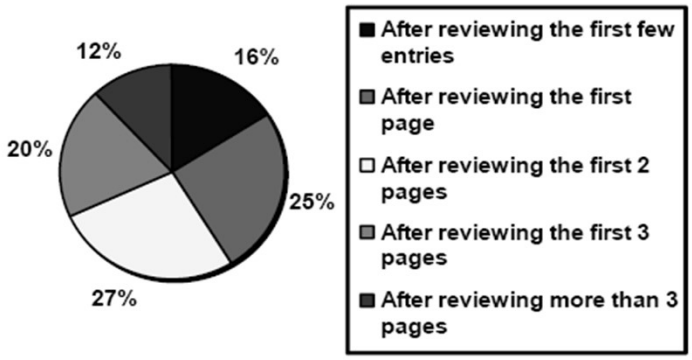
Sec. 19.4.1

User Needs

- Need [Brod02, RL04]
  - **Informational** – want to learn about something (~40% / 65%)
    `Low hemoglobin`
  - **Navigational** – want to go to that page (~25% / 15%)
    `United Airlines`
  - **Transactional** – want to do something (web-mediated) (~35% / 20%)
    `Seattle weather`
    `Mars surface images`
    `Canon S410`
    - Access a service
    - Downloads
    - Shop
  - **Gray areas**
    `Car rental Brazil`
    - Find a good hub
    - Exploratory search "see what's there"

How far do people look for results?

"When you perform a search on a search engine and don't find what you are looking for, at what point do you typically either revise your search, or move on to another search engine? (Select one)"



(Source: iprospect.com WhitePaper_2006_SearchEngineUserBehavior.pdf)

User Experience

- User Queries
  - 3-4 Keywords
  - Seldom uses syntax operators (Free Text Queries)
- Search Engines: Google identified two principles that helped it grow at the expense of its competitors
  - Relevance
  - Simple Interface
- Which Search engine is Bigger?

Index Size & Estimate

- Capture / Recapture Method
  - Suppose that we could pick a random page from the index of E1 and test whether it is in E2's index and symmetrically, test whether a random page from E2 is in E1.
  - These experiments give us fractions x and y such that our estimate is that a fraction x of the pages in E1 are in E2, while a fraction y of the pages in E2 are in E1.
  - Then, letting |Ei| denote the size of the index of search engine Ei, we have $x|E1| \approx y|E2|$,from which we have the form we will use $|E1|/|E2| \approx y/x$

Index Size & Estimate

- Sampling Methods
  - Random Searches
  - Random IP addresses
  - Random Walks
  - Random Queries
- Actual Estimate is quite challenging

Duplicate / Near Duplicate Detection

- Web pages are mirrored for redundancy and high availability, hence while indexing for web search engine we may come up for duplicate (identical copy). Checksum is a common method to detect a duplicate.
- Near Duplicate – not identical, but a portion is common, based on pre-set threshold we can filter out the near duplicates.
- Shingling - Given a positive integer k and a sequence of terms in a document d, define the k-shingles of d to be the set of all consecutive sequences of k terms in d.

Shingling

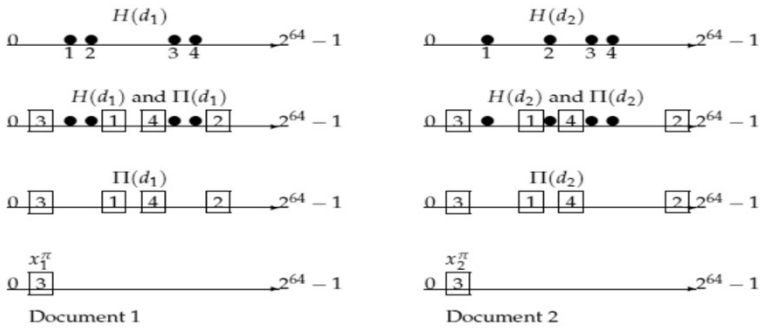
- To find a near duplicate, a shingling approach is used. If there are many common shingling for some k in a pair of documents, its contents will be the same.
- Consider a sentence below

  a rose is a rose is a rose.
- Its shingling set Z = {a rose is a ; rose is a rose ; is a rose is ; a rose is a ; rose is a rose } , which has |Z|=5
- Overlap, by Jaccard = 2/5

Near-Duplicate Scaled Approach

- A pair-wise approach seems unavoidable for using shingling overlap to detect near duplicate.
- We can perform better, by using a large integer Hash Function and doing Hashing for shingling patterns.

Near-Duplicate Scaled Approach



► **Figure 19.8** Illustration of shingle sketches. We see two documents going through four stages of shingle sketch computation. In the first step (top row), we apply a 64-bit hash to each shingle from each document to obtain $H(d_1)$ and $H(d_2)$ (circles). Next, we apply a random permutation Π to permute $H(d_1)$ and $H(d_2)$, obtaining $\Pi(d_1)$ and $\Pi(d_2)$ (squares). The third row shows only $\Pi(d_1)$ and $\Pi(d_2)$, while the bottom row shows the minimum values x_1^π and x_2^π for each document.

Web Crawler

- Web crawling is the process by which we gather pages from the Web to index them and support a search engine.
- The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them.
- web crawler is sometimes referred to as a spider.

Feature a Crawler MUST provide

- Robustness: The crawler must be robust to deal with a large number of linked pages from a website. Sometime server traps a crawler, the crawler must identify these traps.
- Politeness: Web servers have both implicit and explicit policies regulating the rate at which a crawler can visit them. These politeness policies must be respected.

Feature a Crawler Should provide

- Distributed: The crawler should have the ability to execute in a distributed fashion across multiple machines.
- Scalable: The crawler architecture should permit scaling up the crawl rate by adding extra machines and bandwidth.
- Performance and efficiency: The crawl system should make efficient use of various system resources including processor, storage, and network bandwidth.

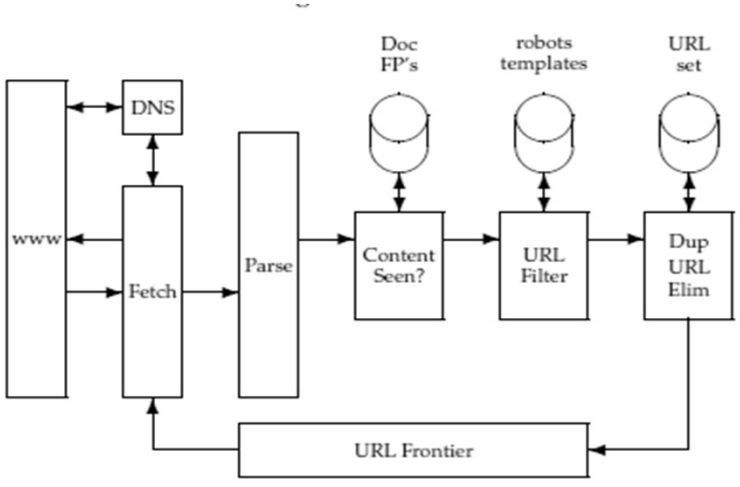
Feature a Crawler Should provide

- Quality: Given that a significant fraction of all web pages are of poor utility for serving user query needs, the crawler should be biased toward fetching “useful” pages first.
- Freshness: In many applications, the crawler should operate in continuous mode: It should obtain fresh copies of previously fetched pages.

Feature a Crawler Should provide

- Extensible: Crawlers should be designed to be extensible in many ways – to cope with new data formats, new fetch protocols, and so on. This demands that the crawler architecture be modular.

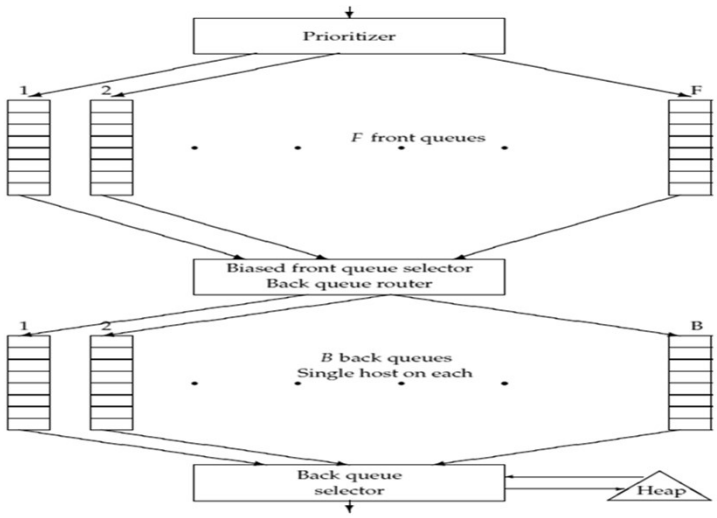
Architecture of a Crawler



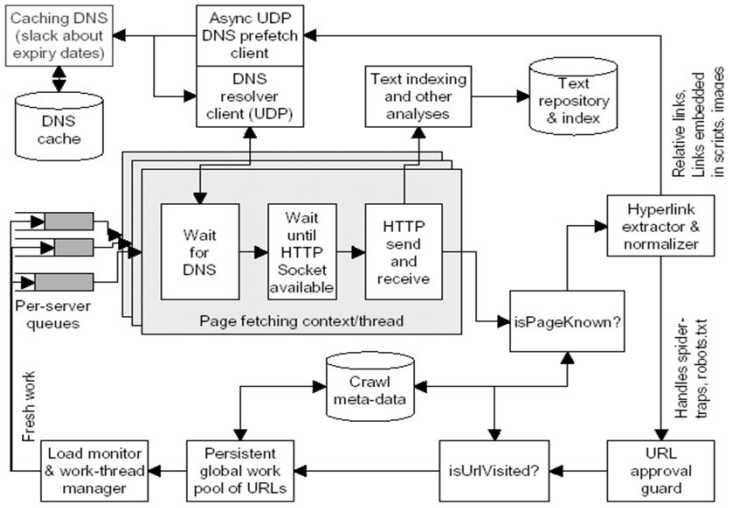
Architecture of a Crawler

- URL Frontier: containing URLs yet to be fetches in the current crawl. At first, a seed set is stored in URL Frontier, and a crawler begins by taking a URL from the seed set.
- DNS: domain name service resolution. Look up IP address for domain names.
- Fetch: generally use the http protocol to fetch the URL.
- Parse: the page is parsed. Texts (images, videos, and etc.) and Links are extracted.

URL frontier



Typical Crawler



Architecture of a Crawler

- Distributed Indexes
  - By term (global Indexes)
  - By document (Local Indexes)
- Connectivity Server
  - URL are transformed into Integers values
  - In-Link and Out-Link states are maintained.
  - Ordering of URL based on Host, lexicographic ordering, etc