The World-Wide Web

Paolo Boldi
DSI
LAW (Laboratory for Web Algorithmics)
Università degli Studi di Milan

Web vs. Internet

World-Wide Web
(nickname: "the web")

popularly mistaken as a synonim of the Internet. Yet:

- it is just a service available on the Internet (though the most popular one)
- Internet pre-dates the web of many decades.

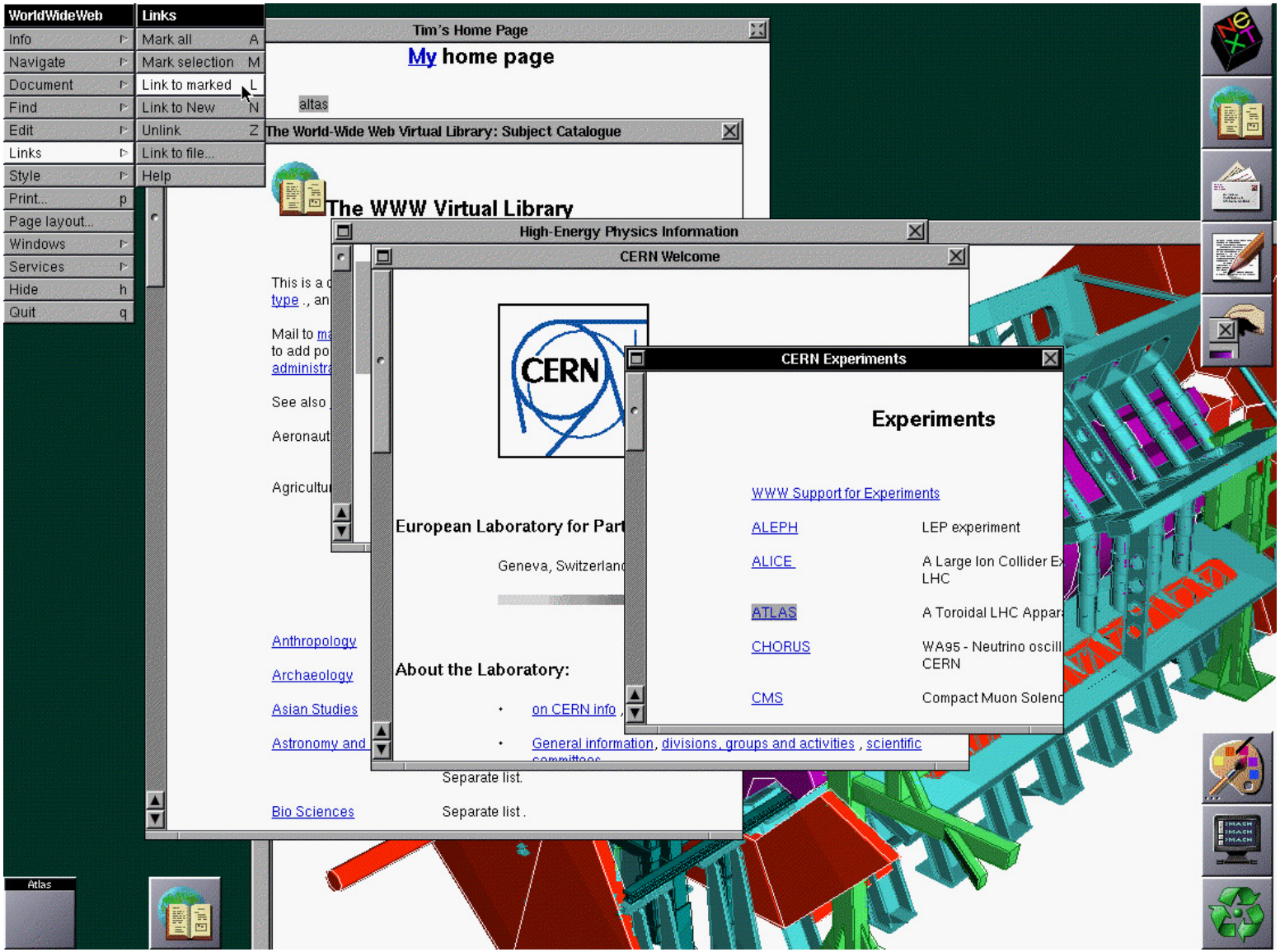
How the Web Was Born

- Tim Berners-Lee, late 1980s, was working at the CERN in Geneva when he invented
  - URLs
  - HTTP
  - HTML
- . . . in one word: he invented the web (and also wrote the first web browser and the first web server!)

A Historical Document

```
From: Tim Berners-Lee (timbl@info.cern.ch)
Subject: WorldWideWeb: Summary
Newsgroups: alt.hypertext
View: (This is the only article in this thread) | Original Format
Date: 1991-08-06 13:37:40 PST

In article (6484@cernvax.cern.ch) I promised to post a short summary
of the WorldWideWeb project. Mail me with any queries.

WorldWideWeb - Executive Summary

The WWW project merges the techniques of information retrieval and
hypertext to make an easy but powerful global information system.

The project started with the philosophy that much academic information
should be freely available to anyone. It aims to allow information
sharing within internationally dispersed teams, and the dissemination
of information by support groups.
...
```

A Historical Screenshot



What we talk about when we talk about the Web

- A definition of the web, anybody?
- "The set of documents identified by URLs with the HTTP protocol"
- Not really precise, though...

Web Ingredients: 1) URLs

URL = Universal Resource Locator: a compact way to name a resource (e.g.: a file) available on the Internet.

A URL is a string that compactly specifies:

- the address (IP or symbolic name) of the machine where the resource is located (*authority*)
- the protocol that should be used when contacting the machine to get the resource
- if needed: the username/password that should be used to get the resource
- the full name (i.e.: name + path) of the resource.

Web Ingredients: 1) URLs (cont'd)

Example:

`ftp://joebozo:bl123@internet.address.edu/foo/bar/file.gz`

means

- connect to the machine named `internet.address.edu`
- using the FTP protocol
- use `joebozo` as username and `bl123` as password
- get the file named `foo/bar/file.gz`

Web Ingredients: 2) HTTP

HTTP = HyperText Transfer Protocol: a protocol that is particularly well-suited to transfer hypertext (but can really be used to transfer anything)

To understand this, let us consider what happens when you type the following in the Address text-area of your browser (IE/Netscape/Mozilla/...):

`http://mathworld.wolfram.com/topics/AppliedMathematics.html`

Web Ingredients: 2) HTTP (cont'd)

`http://mathworld.wolfram.com/topics/AppliedMathematics.html`

Your browser (client) contacts the remote machine (server)

- it consults its DNS machine to translate the symbolic machine name (`mathworld.wolfram.com`) into a numeric IP address (`140.177.205.23`)
- a TCP connection is established to the remote machine
- works more or less like a telephone call

Web Ingredients: 2) HTTP (cont'd)

http://mathworld.wolfram.com/topics/AppliedMathematics.html

The client issues a request; in this case, it will be something like:

```
get /topics/AppliedMathematics.html http/1.1
host: mathworld.wolfram.com
```

. . . plus a final newline to mean that the request is ended

Web Ingredients: 2) HTTP (cont'd)

http://mathworld.wolfram.com/topics/AppliedMathematics.html

The server answer may be something like:

```
HTTP/1.1 200 OK
date: Fri, 17 May 2002 09:47:47 GMT
server: Apache/1.3.20 (Unix)
connection: close
cache-control: no-cache
content-length: 22085
content-type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>  ...
```

And then the connection (usually) is closed: that's why HTTP is "stateless".

The Web as a Graph

You can think of the Web as a (directed) graph:

- its nodes are the URLs
- there is an arc from node x to node y iff the page with URL x contains a hyperlink towards URL y.

This is called the *Web graph*.

How Large Is It?

It is *extremely* difficult to evaluate the Web size!

- about 8/11 billions of nodes (excluding the *dark mass*)
- about 300/500 billions of arcs
- about 400/600 millions of hosts
- about 2,000 millions of users.

Dark mass = pages that cannot be reached: according to prudent estimates, currently indexed web is just the 16-20% of the "real" web!

BE CAREFUL! Our knowledge of the Web is only indirect! (it is what we can collect using web crawlers, or interrogating search engines...)

What Is the Shape of the Web?

- The (known part of the) Web graph has a bowtie-like shape...
- Subdomains (e.g., country-code domains [`.fr`, `.it`, ...], large intranets etc.) are similar: a fractal-like graph
- It is *highly dynamic*

What Is the Shape of the Web?

Giant component (core)

- A strongly-connected component containing about 30% of the pages
- Estimated diameter: directed $\rightarrow$ 20/30; undirected $\rightarrow$ 10/17
- "It's a small-world!"

What Is the Shape of the Web?

Left-hand side

- Ancestors of the giant component in the scc DAG
- About 25%
- Estimating its size is very difficult (how can you get there???)
- They are the Web pariahs: they are there, but no one wants to link them!

What Is the Shape of the Web?

Right-hand side

- Descendants of the giant component in the scc DAG
- About 25%
- Contains all “documents” with no hyperlinks in them (e.g., pure text, word/PDF/PostScript with no links etc.)

What Is the Shape of the Web?

Tubes, tendrils and isolated components

- Remaining nodes (20%)

Anarchy in the Web: Search Engines

Due to its dimensions and its anarchic hierarchy, it is extremely difficult to find (high-quality, pertinent) information.

Currently, two solutions:

- directories (*à la* dmoz): human-selected pages, categorized by content
- search engines (*à la* Google): databases containing a "complete" and updated copy of the Web, that allow you to make full-text search on the page contents

Search engines are by far the most popular way to search for information in the Web.

Hostile presences in the Web

The Web sports a number of hostile behaviors that one should be aware of:

- *dark web*: not necessarily "hostile", but difficult (impossible) to index/search
- *search-engine traps*: portions of the web that "entrap" search bots; sometimes unintentionally
- *spam*: pages that do not provide useful information but are there for some other reason (e.g., providing traffic/rank/popularity to sites that don't deserve it)
- *irrelevant/misleading information*: often unintentional, but extremely difficult to deal with

Web spam

In the web, "spam pages" are there to mislead search engine, so that they assign an undeservedly high rank to some pages (making them believe that they are more important that they should be).

- SEO (Search Engine Optimization) is the politically-correct term used for spamming...
- *Boosting techniques* (aimed at obtaining high ranks): term spamming, link spamming (honeypots, infiltrating web directories, blog posting, buying expired domains, creating link farms)
- *Hiding techniques* (to hide boosting from humans): content hiding (white-on-white), cloaking, redirection.

Web business models

Examples of different business models:

- *Advertising*: in the form of banners etc.
- *Brokerage*: bringing buyers and sellers together (get paid by sellers and possibly by buyers)
- *Infomediary*: collecting information from sellers and aiding buyers in the choice
- *Merchant*: sales goods produced by someone else, possibly through auctions
- *Manufacturer*: direct sale of goods
- *Subscription*: provides a service with periodic fee payment
- *Utility*: provides a service under a pay-as-you-go approach

Web advertising

Web advertising is the main revenue for most search engines.

- Typically the SERP (Search Engine Result Page) contains some *sponsored results*
- Google AdWords, for example, adopts a PPC (Pay-Per-Click) policy
- Some search engine allow a service to put sponsored ads on your own pages (getting paid for the space you allow): e.g., Google AdSense
- Often, sophisticated techniques are adopted to decide how/when to select an ad for display and how much to charge for clicking it

PPC and click frauds

PPC (Pay-Per-Click) brought to the new phenomenon of click frauds. Different kinds of perpetrators:

- *Ad networks* (acting as middleman between advertisers and publishers)
- *Competitors of advertisers*
- *Competitors of publishers*
- *Friends of publishers*
- *Vandals*

Sometimes these behaviors cannot be sued because they are perfectly legal.

Search engines

Allows users to enter *queries* to search for relevant web pages:

- First engines started in 1993-95: Lycos, AltaVista, Yahoo! (then, a directory)
- Currently, the most popular ones are: Google (1998, 82%), Yahoo! (2004, 7%), Baidu (2000, 5%), Microsoft Bing (ex-MSN, 1998, 4%)
- Meta-search engine (redirect a query to many search engines)

Search engine features

Features related to search-engine quality:

- Coverage
- Freshness
- Ranking quality
- User-interface, query language richness and alike
- Further services: query suggestions, result clustering, etc.

Coverage and size estimation

Given two search engines A and B, how can we estimate their relative coverage? A variant of the capture-recapture method used in ecology.

- let $P[A \cap B \mid A]$ be the probability that a URL is indexed by both, given that it is indexed by A; then

$$P[A \cap B \mid A] \approx |A \cap B|/|A|$$

  and similarly for $P[A \cap B \mid B]$
- so

$$|A|/|B| \approx P[A \cap B \mid A]/P[A \cap B \mid B]$$

- to compute this one needs: i) *sampling* (a procedure to draw, uniformly at random, a page from the index of a given search engine) ii) *checking* (a way to determine if a given page is indexed by another search engine)
- After that, one can use the *declared search engine size* to obtain actual sizes

Coverage and size estimation: i) sampling

- Build a lexicon (e.g., using DMOZ.com)
- Bin words by frequency, and submit one-word queries from different bins
- Every time, select a URL at random from the SERP

Coverage and size estimation: ii) checking

- Take the top k most discriminating words from the page
- Submit the query obtained in this way, and check if the URL is in the result set (CAREFUL: URL normalization is in the way!)
- Every time, select a URL at random

Pages vs. URLs

What is the relation between pages and URLs? Do different URLs always correspond to different pages?

- No, for many reasons:
  - Mirroring or quasi mirroring (small differences, like dates etc.)
  - Synonym URLs (via redirection or direct server translation)
  - Presence of user IDs or other similar tricks
- Number of URLs and number of (distinct) pages are quite different!
- This fact cannot be ignored when building a search engine

(Near-)duplicate detection

The problem has two aspects:

- Given a URL, decide whether it corresponds to a page that has already been crawled or not *before trying to dowload it*; e.g., learn some URL-rewrite rule that determine this with high probability
- After a page has been crawled, decide quickly if it is a duplicate (or quasi-duplicate) of an existing one
- Exact duplicates are easy to find (e.g., keeping an MD5 of the crawled pages [pure text or HTML?])
- Near duplicates require more interesting techniques...

Shingling

The problem of detecting near-duplicates is interesting, because it allows us to introduce some techniques that have a general appeal.

Given a document D, and for a fixed parameter k, define $S(D)$ the set of *k-shingles* of D, that is, the set of k-grams appearing in the document D.

If D is *a rose is a rose is a rose* and $k = 3$, then $S(D)$ is $\{$`"a rose is"`, `"rose is a"`, `"is a rose"`$\}$.

Define the *similarity* between document D and document D' as $J(S(D), S(D'))$, where J is the Jaccard coefficient:

$$J(A, B) = \frac{A \cap B}{A \cup B}.$$

First simplification: hashing

Instead of computing $J(S(D), S(D'))$ (the overlap between the two sets of shingles), we map shingles into integers using some hash function $h : \Sigma^* \to \{0, \ldots, M-1\}$, and compute $J(h(S(D)), h(S(D')))$ instead.

If M is large (say $M = 2^{64}$), the probability that two shingles collide (i.e., are mapped to the same integer) is very low, so the Jaccard coefficient is going to be the same:

$$J(S(D), S(D')) \approx J(h(S(D)), h(S(D'))).$$

Let us write $H(D)$ for $h(S(D))$. So now the problem is the computation of the Jaccard coefficient between two sets of integers in the universe $\Omega = \{0, \ldots, M-1\}$.

Second simplification: min-wise permutation

Let $A, B \subseteq \Omega = \{0, 1, \ldots, M-1\}$, and let Π be the set of all $M!$ permutations of Ω.

Theorem

If π is drawn uniformly at random from Π

$$P[\min(\pi(A)) = \min(\pi(B))] = J(A, B).$$

So one way to estimate $J(A, B)$ is to draw a number of permutations, say $\pi_1, \pi_2, \ldots, \pi_N$, and count how many times $\min(\pi_i(A)) = \min(\pi_i(B))$.

Third simplification: sketching

Fix N permutations $\pi_1, \ldots, \pi_N$ and define the *sketch* of a document D as

$$\sigma(D) = \langle \min(\pi_1(H(D))), \ldots, \min(\pi_N(H(D))) \rangle.$$

$S(D, D')$ is approximated by the fraction of indices i such that $\sigma_i(D) = \sigma_i(D')$.