

# Review: “Searching the Web” [Arasu 2001]

Gareth Cronin  
University of Auckland  
gareth@cronin.co.nz

*The authors of “Searching the Web” present an overview of the state of current technologies employed in the construction of search engines for the World Wide Web (WWW).*

*Through a combination of studies of the current literature and their own experiments, the authors describe a general Web search engine architecture and contrast some “crawling” and indexing strategies. The experiments are carried out on the “Stanford WebBase” using the contents of the Stanford University Website, consisting of around 225 000 pages.*

*Solutions to the challenges presented by the large scale of the WWW are explored.*

## 1. Introduction

The authors of “Searching the Web” present an overview of the state of current technologies employed in the construction of search engines for the World Wide Web (WWW).

Through a combination of studies of the current literature and their own experiments, the authors describe a general Web search engine architecture and contrast some “crawling” and indexing strategies. The experiments are carried out on the “Stanford WebBase” using the contents of the Stanford University Website, consisting of around 225 000 pages.

The scope of the paper is broad and gives somewhat brief descriptions of each of the topics it covers, but it does cite articles that provide excellent in-depth descriptions of many of the concepts.

The article is by a research group in the Department of Computer Science at Stanford University, California, U.S.A and appeared in the *ACM Transactions on Internet Technology*.

In the words of the ACM, the Transactions on Internet Technology (TOIT):

**“...brings together many computing disciplines including computer software engineering, computer**

**programming languages, middleware, database management, security, knowledge discovery and data mining, networking and distributed systems, communications, performance and scalability etc. TOIT will cover the results and roles of the individual disciplines and the relationships among them.” [ACM 2002]**

## 2. Background

As mentioned previously, “Searching the Web” cites quite a number of articles that cover various aspects explored in the paper in much greater depth.

There has been much research into the graph structure of the Web and ways to exploit classical graph algorithms, such as those found in most Computer Science textbooks. Of particular note are Jon M. Kleinberg’s 1999 paper [Kleinberg 1999] on analysing hyperlinks and hyperlinked documents to determine their degree of authority and the Coffman et al article proving a theorem of optimal performance and “freshness” when selecting web pages to update in a previously populated web page repository [Coffman 1998].

Some researchers who have built successful search engines have published case studies of their work [Brin 1998].

Much of the work on indexing, ranking and querying algorithms has been carried out commercially and remains commercially sensitive. It is therefore inaccessible to the general public and to academia [Arasu 2001].

## 3. Purpose

Current estimations of the size of the Web place it at over 1 billion pages. While it is possible for users to make use of “portals” and similar entry points that use human effort to categorise and list pages of interest, search engines provide a more flexible way of finding pages that are relevant to a user’s query [Arasu 2001].

The Web has become a powerful tool for both commercial and academic research, but it is only as useful as the search engines that are available to find relevant material amongst the millions of Web pages. Search engine Web interfaces are easily the most visited sites on the Web and are therefore of great commercial interest to advertisers and marketers who wish to expose their brands to the greatest audience possible. Although the latter is of questionable value, further understanding and improvement of Web searching technologies is of great importance.

An interesting case in point is that shortly after the September the 11<sup>th</sup> attacks on the World Trade Centre in the United States of America, around 6000 users a minute searched for “cnn” on the search engine “Google” [Google 2002]. Rather than simply typing the domain name, users trusted a search engine as their first point of contact with the Web [Wiggins 2002].

#### 4. General Structure

The article begins with a description of the general search engine architecture, comprising what is known as crawlers, indexers, repositories, analysers and the query engine.

The idea is that the “crawlers” navigate their way over Web pages, starting from some Uniform Resource Locator (URL) that constitutes a “hub” – a page that has links to many authoritative pages, where authoritative pages can be defined in a number of ways [Kleinberg 1999]. The crawler follows links that are considered worthwhile following and deposits a copy of the page being crawled into a repository. This process continues until for example, there is no more storage available in the repositories.

The indexers produce a reverse index of the terms found in the pages in the repository, associating a list of URLs with each term. A lexicon is generated that lists all the terms found in all pages along with some statistics such as the number of pages the term is found on; this is useful for page ranking during querying. At this point the repositories may be emptied and further crawling continues, or if the pages are to be cached on an ongoing basis, crawling stops. Some engines do hold cached copies of all pages they crawl, “Google” [Google 2002] holds every page in compressed form to allow researchers to download Web data on mass [Brin 1998].

Utility indexes may also be generated by analysers, indexing on such values as the number of images in each page, or the size of pages.

On the initial crawl when the repositories are empty, the crawling process is carried out in full. For subsequent crawls, an update strategy must be chosen to

update indexes where any of the content of the currently indexed pages has changed.

User queries are processed by the query engine, which must use the indexes to decide which pages are most relevant the query. These pages are usually returned in a ranked fashion, where the pages that are thought to be most relevant to the query are listed before those that are less relevant. Relevance can be based on different factors that are explained later.

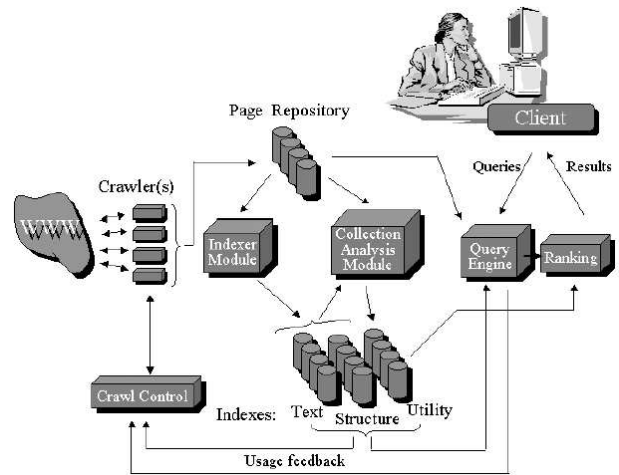


Figure 1 - General Web Server Architecture [Arusu 2001]

#### 5. Issues and Challenges

The description of the architecture above raises many questions, many of which are addressed by the article.

Most of the challenges surround the issue of scale. As mentioned earlier, according to some studies the Web is currently estimated at over a billion pages, and “Searching the Web” places the size of textual data alone in the tens of terabytes. This means that considerable computing power and storage resources by today’s standards are required to deal with the volumes of data. The authors note that well-known information retrieval methods are not designed to cope with this sort of scale.

The “bow tie” structure of the Web [Broder 2000] is mentioned early in the article, but its significance is not discussed further. This is a pity, as the concept of searching around a central core of pages (the centre of the bow tie) and perhaps how to include the outer pages (the pages that can reach the core but are not reached) is an interesting one.

#### 6. Crawling and Storage Challenges

There are practical limits on the time taken to crawl and index pages before the information being indexed becomes out of date. The concept of how “fresh” an index is presents a constant challenge both in how to measure the “freshness” and how to maintain it given the constraints of speed and storage available.

Because a search engine can not practically download all pages on the Web, there must be some method of selecting which pages to download. In theory these should be the most “important” pages, where “important” can be defined in many ways. The authors describe the PageRank and HITS algorithms explained briefly later in this paper.

Taking advantage of distributed computing is an obvious way to improve the speed of crawling, indexing and analysing. The authors of the article do not address this in depth and comment that more research needs to be done in this area.

Storage must be sufficiently scalable to cope with the predicted growth of the Web; some studies place the rate of growth at around a doubling every eight months [Kobayashi 2000]. Again, distribution of storage nodes is a solution here.

The storage must also be able to cope with both sequential and random access to deal with indexing methods and user queries respectively. Deletion of index items must be supported to enable updates where Web pages are found to be no longer accessible. Physical page organisation is important in order to make trade-offs between the different access methods.

The authors’ own experiments in another paper [Hirai 2000] find that a “log-based” scheme is optimal, where incoming pages are appended to a contiguous “log” and b-tree indexes are used for speeding random access.

Two basic choices for updating methods are outlined in the article. A distinction is made between a batch-mode and steady crawler, where updates are either periodic or are restricted to a certain time period e.g. monthly. A further distinction is made between partial or complete updates. A complete crawl involves performing the same operation as the initial crawl when the repository was empty, whereas a partial crawl involves deciding in which pages to look for updates and refreshing only those.

The authors suggest using either a batch-mode crawler with complete crawls that utilises a “shadowing repository” or a steady crawler with “in-place updates”. “Shadowing” refers to keeping a separate collection of updates to the original collection of pages in the repository as opposed to in-place updates that are made directly to the original collection.

An interesting and somewhat wider issue briefly raised in the article is the load placed on web servers being crawled. If this is ignored by engine designers, server administrators may begin to ban web robots from their sites. One partial solution is a protocol developed quite recently for marking web servers with a small text file (robots.txt) that informs visiting robots which directories they should not visit [Robots 2002].

## 7. Indexing and Analysis Challenges

A link index to provide structural information for analysis is useful, e.g. to aid in finding related pages to a given initial search result. The article notes that graph theory can help with efficiently implementing a link index by using adjacency lists to record page “neighbourhood” information.

The text index that relates search terms to pages is usually built using inverted indexes. The authors do point out some alternative methods such as “suffix arrays” and “signature files” and cite articles where these are explored [Manber and Myers 1990] [Faloutsos and Christodoulakis 1984]. The authors suggest a structure where index entries contain the term, the location and a “payload”. The payload contains data such as how “important” to the page the term is to aid in page ranking of returned search result sets. For example, words that appear between bold tags (<b></b>) within a page may be considered more important.

Utility indexes are additional indexes used for implementing search engine features. The authors’ example is that of a search engine that allows searching of a particular set of domains. In this instance, an index mapping URLs to domains could be maintained to improve the performance of the ranking process.

## 8. Metrics for Ranking and Freshness

The article presents a set of metrics for calculating the “freshness” of a set of pages collected in a repository, i.e. how up-to-date the pages are. These metrics can be used to compare different refresh strategies. The authors point the reader to another of their articles proving a theorem on the optimal refresh policy for a given page, based on the frequency of change to the page’s contents [Garcia-Mollina 2000].

A simple definition of freshness is presented, where freshness is the “...fraction of the local collection that is up-to-date.” [Arusu 2001] or:

$$F(S;t) = \frac{1}{N} \sum_{i=1}^N F(E_i;t)$$

Where  $S = \{e_1 \dots e_N\}$  is the set of  $N$  pages and  $F(e_i;t) = 1$  if  $e_i$  is consistent with its real-world counterpart at time  $t$  else  $F(e_i;t) = 0$

Being able to make this calculation is of course dependent on knowing the rate of change for each page in the collection. This is not easy in the practical sense in that all pages would need to be measured several times over a period of time to calculate their rate of change. The article posits that a uniform refresh policy of refreshing pages at regular intervals regardless of their rate of change will always be superior to any proportional rate of updating, so long as the update frequencies follow a Poisson process. They refer again to the aforementioned article [Garcia-Mollina 2000] in which they provide a detailed proof of this. They also refer to another as yet unpublished study they have made where they find that a Poisson process is a reasonable model for the frequency of change of Web pages.

“Searching the Web” points out that traditional information retrieval methods for ranking pages rely on the pages being self-descriptive, i.e. there is an assumption that the keywords forming a user’s query are found in the text of pages relevant to that query. This is not true of Web pages on the whole, for example:

**“One could begin from the query ‘search engines’, but there is an immediate difficulty in the fact that many of the natural authorities (Yahoo!, Excite, AltaVista) do not use the term on their pages. This is a fundamental and recurring phenomenon—as another example, there is no reason to expect the home pages of Honda or Toyota to contain the term ‘automobile manufacturers’.” [Kleinberg 1999]**

For this reason other methods of ranking should be used.

A common system that is employed by the popular “Google” [Google 2002] search engine is the PageRank scheme [Brin 1998]. PageRank is an extension of the “citation count” measurement, where the more links that exist to a page, the more important the page is considered. The PageRank algorithm works on the basis that more links are more important still if the page they link from is in turn more important. It is calculated using eigenvalues on a matrix representation of the Web graph. The authors direct interested readers to the original article by Page et al [Page 1998].

Another important algorithm described is the HITS (Hypertext Induced Topic Search) algorithm. HITS is based around the concept of “hubs” that join together “authoritative” information sources, where the authority is calculated heuristically from the sets of pages that the hubs connect [Kleinberg 1999]. A number of less popular methods are also mentioned, but the driving factor behind all of the methods is that the link structure of the Web presents a wealth of useful information that can be analysed.

## 9. Future Directions

The authors state that the study of the analysis of the link structure of the Web as an area where there is much potential for further research. They also raise the possibility of investigating query logs and click streams as a method of improving Web searching. Query logs can be analysed through interpreting past queries and associated result sets to track which information users select from those results and so which information is more likely to be relevant to a given query. Click stream analysis follows a similar method, except that the actual clicks on individual items on a web site are recorded and analysed.

The article raises another interesting issue regarding sites where content is not accessible via hyperlinks, but only through filling out and submitting HTML forms. Current Web search engines are not capable of filling out forms and as the proportion of dynamic to static content grows, research into ways of performing this function during the crawl process could become necessary. I believe that this is perhaps not such a serious issue, as the convention of using hyperlinks is well entrenched and I have certainly not witnessed a move towards a form-only paradigm.

## 10. Conclusion

“Searching the Web” gives a broad overview of current general Web search engine architecture and identifies some challenges and possible solutions to problems facing the designers of such engines. All of the challenges stem from the over-arching problem of the mammoth scale of the Web. Both the current size and the rate of growth of the WWW are very large by information retrieval standards and so special ways of dealing with such a scale have been and will continue to be found.

The Web becomes a more important tool for information-finding and research every day and its user community grows substantially every year. The Web’s

performance is only as good as the programs that index and search it, therefore research into search engine improvements gains importance by the day.

## References

- [ACM 2002] *Association for Computing Machinery (ACM)*, <http://www.acm.org>, "Transactions on Internet Programming", [http://portal.acm.org/browse\\_dl.cfm?linked=1&part=transaction&idx=J780&coll=portal&dl=ACM&CFID=1974065&CFTOKEN=28973806](http://portal.acm.org/browse_dl.cfm?linked=1&part=transaction&idx=J780&coll=portal&dl=ACM&CFID=1974065&CFTOKEN=28973806), 2002.
- [Arasu 2001] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke and Sriram Raghavan. "Searching the Web", *ACM, ACM Transactions on Internet Technology*, vol. 1, no. 1, August 2001, pp 2-43, U.S.A.
- [Brin 1998] Brin S, Page L. "The anatomy of a large-scale hypertextual Web search engine", *Elsevier, Computer Networks & Isdn Systems*, vol.30, no.1-7, April 1998, pp.107-17. Netherlands.
- [Broder 2000] Broder A, Kumar R, Maghoul F, Raghavan P, Rajagopalan S, Stata R, Tomkins A, Wiener J. "Graph structure in the Web", *Elsevier, Computer Networks: the International Journal of Distributed Informatique/Computer Networks*, vol.33, no.1-6, June 2000, pp.309-20. Netherlands.
- [Coffman 1998] Coffman EG Jr, Zhen Liu, Weber RR. "Optimal robot scheduling for web search engines", *Journal of Scheduling*, vol.1, no.1, June 1998, pp.15-29. Publisher: Wiley, UK. [Faloutsos and Christodoulakis 1984]
- [Garcia-Mollina 2000] Junghoo Cho, Garcia-Molina H. "Synchronizing a database to improve freshness", *ACM, Sigmod Record (Acm Special Interest Group on Management of Data)/Sigmod Record*, vol.29, no.2, June 2000, pp.117-28. USA.
- [Google 2002] Google Search Engine, <http://www.google.com>, 2002.
- [Hirai 2000] Hirai J, Raghavan S, Garcia-Molina H, Paepcke A. "WebBase: a repository of Web pages", *Elsevier, Computer Networks: the International Journal of Distributed Informatique/Computer Networks*, vol.33, no.1-6, June 2000, pp.277-93. Netherlands
- [Kleinberg 1999] Kleinberg JM. "Authoritative sources in a hyperlinked environment", *ACM, Journal of the ACM*, vol.46, no.5, Sept. 1999, pp.604-32. USA.
- [Kobayashi 2000] Kobayashi M, Takeda K. "Information retrieval on the Web", *ACM, ACM Computing Surveys*, vol.32, no.2, June 2000, pp.144-73. USA.
- [Manber and Myers 1990] Manber U, Myers G. "Suffix arrays: a new method for on-line string searches", *SIAM*

*Journal on Computing*, vol.22, no.5, Oct. 1993, pp.935-48. USA.

[Page 1998] Page, L., Brin, S., Motwani, R., and Winograd, T. 1998. **The pagerank citation ranking: Bringing order to the web**. Technical Report. Computer Systems Laboratory, Stanford University, Stanford, CA.

[Robots 2002] [www.robotstxt.org](http://www.robotstxt.org), <http://www.robotstxt.org>, "A Standard for Robot Exclusion", <http://www.robotstxt.org/wc/norobots.html>, 2002.

[Wiggins 2002] Richard W. Wiggins, *First Monday*, <http://www.firstmonday.dk>, "The Effects of September 11 on the Leading Search Engine", [http://www.firstmonday.dk/issues/issue6\\_10/wiggins/index.html](http://www.firstmonday.dk/issues/issue6_10/wiggins/index.html), 2002.