Web Crawling

Christopher Olston and Marc Najork

Slides created by *Aécio Solano Rodrigues Santos* and *Nivio Ziviani*
based on the survey *Web Crawling* from *Foundations and Trends in Information Retrieval* (2010).

Summary

Introduction

Crawler Architecture

Crawl Ordering Problem

Summary

Introduction

Crawler Architecture

Crawl Ordering Problem

Introduction

- A *web crawler* (also known as a *robot* or a *spider*) is a system for the downloading of web pages.
- They are used for a variety of purposes:
  - Web Search Engines
  - Web Archiving
  - Web Data Mining
  - Web Monitoring

Introduction

- The basic algorithm is simple. Given a set of seed URLs:
    1. downloads all the web pages addressed by the URLs;
    2. extracts the hyperlinks contained in the pages;
    3. iteratively downloads the web pages addressed by these hyperlinks.
- Despite the apparent simplicity of this basic algorithm, web crawling has many inherent challenges.

Challenges

- **Scale**
  - The web is very large and continually evolving.
  - Crawlers that seek coverage and freshness must achieve extremely high throughput, which poses difficult engineering problems.
- **Content selection tradeoffs**
  - Crawlers do not purport to crawl the whole web, or keep up with all the changes.
  - Goals: acquire high-value content quickly, ensure coverage of all reasonable content, and bypass low-quality, irrelevant, redundant, and malicious content.
  - The crawler must balance coverage and freshness, while obeying constraints such as per-site rate limitations.

Challenges

- **Social obligations**
  - Crawlers should not impose too much of a burden on the web sites they crawl.
  - Without safety mechanisms a high-throughput crawler can inadvertently carry out a *denial-of-service attack*.
- **Adversaries**
  - Some content providers inject useless or misleading content into the corpus assembled by the crawler.
  - Motivation: financial incentives, for example (mis)directing traffic to commercial web sites.

Summary

Introduction

Crawler Architecture

Crawl Ordering Problem

Architecture Overview

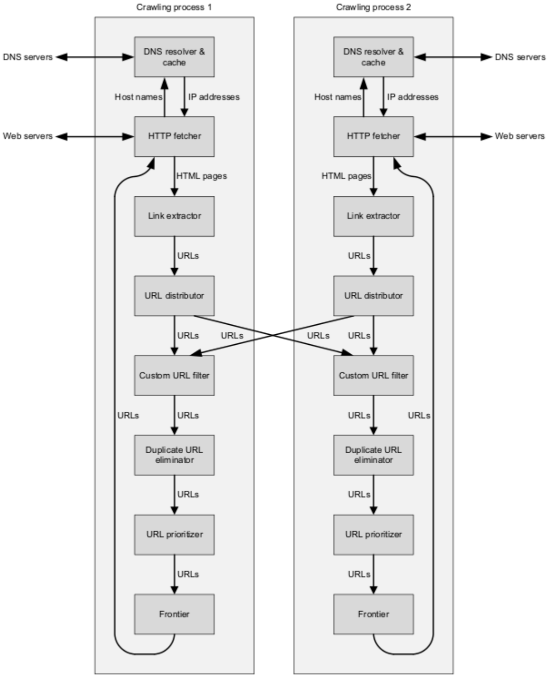
- Consists of multiple processes running on different machines connected by a high-speed network.
- Each process consists of multiple worker threads.
- Each worker thread performs repeated work cycles.
- At each work cycle, a worker obtains a URL from the **Scheduler** or *Frontier*, which organizes URLs according to priority and politeness policies.
- The worker thread invokes the **HTTP fetcher**:
  - It calls a DNS module to resolve the host component of the URL into the IP address of the corresponding web server
  - Connects to the web server and checks for robots exclusion files and attempts to download the web page.

Architecture Overview

- If the download succeeds the web page is stored in a repository.
- The **Link extractor** parses the HTML content of the page and extracts hyperlinks.
- The extracted URLs are passed to a **URL distributor**, which assigns each URL to a crawling process.
- This assignment is typically made by hashing the URLs host component, its domain, or its IP address (requires additional DNS resolutions).
- Most hyperlinks refer to pages on the same web site, thus assignment to the local crawling process is the common case.

Architecture Overview

- The URL passes through:
  - **Custom URL filter**: to exclude URLs belonging to "black-listed" sites and URLs with particular file extensions that are not of interest
  - **Duplicate URL eliminator**: maintains the set of all URLs discovered so far and passes on only never-before-seen URLs.
- The **URL prioritizer** selects a position for the URL in the Frontier or Scheduler, based on estimated page importance or rate of change.


Crawling process 1
Crawling process 2
DNS servers
DNS resolver & cache
Host names
IP addresses
Web servers
HTTP fetcher
HTML pages
Link extractor
URLs
URL distributor
Custom URL filter
Duplicate URL eliminator
URL prioritizer
Frontier
DNS servers
DNS resolver & cache
Host names
IP addresses
Web servers
HTTP fetcher
HTML pages
Link extractor
URLs
URL distributor
Custom URL filter
Duplicate URL eliminator
URL prioritizer
Frontier

Key Design Points

- Any web crawler needs to keep track the URLs.
- Operations that must be supported:
  - Adding a new URL
  - Retrieving a URL
  - Marking a URL as downloaded
  - Testing whether the set contains a URL
- In-memory data structures (e.g., trees or sorted lists) exceed the amount of memory available on a single machine.

Key Design Points

- Solution: maintain the data structure on disk.
- However, cost of accessing items in the set (e.g., set membership test) involves an expensive disk seek.
- Every modern web crawler splits the crawl state into:
  - **URL-seen test** or **Duplicate URL eliminator**: maintains the set of URLs that have been discovered (whether downloaded or not)
  - **Scheduler**: maintains the set of URLs that have yet to be downloaded.

Scheduler Data Structure and Politeness

- A straightforward implementation is a First-in-First-out (FIFO) queue, which results in a breadth-first traversal of the web graph.
- However, this simple approach has drawbacks:
  - Most hyperlinks refer to another page on the same web server, resulting in many consecutive HTTP requests to that server.
  - A barrage of requests is considered “impolite”, and may end in a denial-of-service attack.

Scheduler Data Structure and Politeness

- One solution: multithreaded or distributed crawler that issues many HTTP requests in parallel.
- Avoid issuing multiple overlapping requests to the same server.
  - Maintain a mapping from web servers to crawling threads, e.g., by hashing the host component of each URL.
  - Each crawling thread has a separate FIFO queue, and downloads only URLs obtained from that queue.

Scheduler Data Structure and Politeness

- Politeness policy: space out requests to each web server according to that server's capabilities.
  - E.g., delay subsequent requests by a multiple (say $10\times$) of the time it took to download the last page from that server.
- This policy ensures that:
  - The crawler consumes a bounded fraction of the web server's resources.
  - Fewer pages will be downloaded from slow or poorly connected web servers than from fast, responsive web servers.

Scheduler Data Structure and Politeness

- It may be desirable to prioritize pages according to their estimated usefulness, based on:
  - PageRank
  - Traffic they receive
  - Reputation of the web site
  - Rate at which the page has been updated in the past
  - Etc.

URL Seen Test

- The second major data structure: **URL-seen test** (UST) or the **duplicate URL eliminator** (DUE)
- It keeps track of the set of URLs that have been previously discovered and added to frontier.
- UST needs to support:
  - Insertion
  - Set membership testing
- In a continuous crawling setting, it must also support deletion to cope with URLs that no longer point to a valid page.

URL Seen Test

- In a disk-based hash table, each lookup requires a disk seek, severely limiting the throughput.
- Caching popular URLs can increase the throughput.
- However, the average web page contains close to hundred links and each link needs to be tested for novelty.
- Implementations performing random file accesses perform poorly, but those that perform streaming sequential reads or writes achieves good throughput.

URL Seen Test

- The IRLbot crawler [84] uses:
- Batch of URLs arriving at the DUE is written to disk, distributed over multiple files keyed by the prefix of each hash.
- Once the size of the largest file exceeds a certain threshold, the files that together hold the batch are read back into memory one by one and merge-sorted into the main URL hash file on disk.

Auxiliary Data Structures: Robots File

- *Robots Exclusion Protocol*: allows an administrator to bar crawlers from crawling the site, or pages within the site.
- Provides a file at URL `/robots.txt` containing rules that specify which pages the crawler is allowed to download.
- A crawler should check whether the site supplies a `/robots.txt` file, and if so, adhere to its rules.
- To avoid repeatedly requesting `/robots.txt`, crawlers cache the results of previous requests of that file.

Auxiliary Data Structures: DNS

- URLs contain a host component (e.g., www.yahoo.com), which is “resolved” using the Domain Name Service (DNS) protocol.
- DNS requests can take quite a long time due to the request-forwarding nature of the protocol.
- Therefore, crawlers often maintain their own DNS caches.

Incremental Web Crawling

- Web crawlers can be used to:
  - Assemble one or more static snapshots of a web corpus (batch crawling)
  - Perform *incremental* or *continuous crawling*: download newly discovered pages and re-download previously crawled pages.
- Efficient incremental crawling requires a few changes to the major data structures of the crawler.
  - The DUE should support the deletion of URLs that are no longer valid (e.g., that result in a 404 HTTP return code).
  - URLs are retrieved from the scheduler and downloaded as in batch crawling, but they are subsequently reentered into the scheduler.

Summary

Introduction

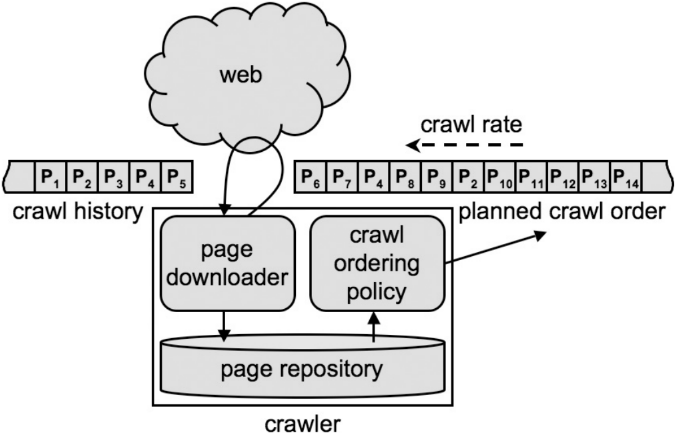
Crawler Architecture

Crawl Ordering Problem

Crawl Ordering Problem

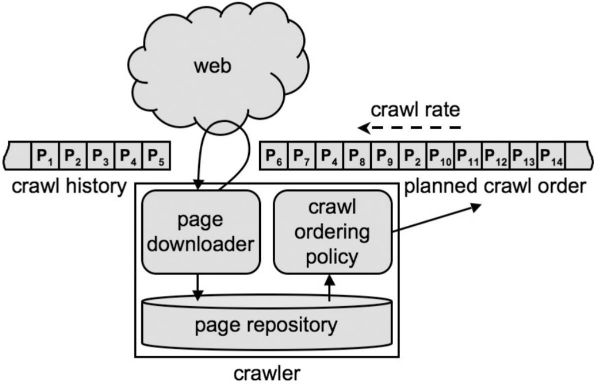
- The crawl order is extremely important:
  - due to the growth rate of new content
  - due to dynamically generated content
- A good crawler order must consider:
  - **Coverage.** The fraction of desired pages that the crawler acquires successfully.
  - **Freshness.** The degree to which the acquired page snapshots remain up-to-date, relative to the current “live” web copies.

Simplified Crawl Ordering Model



Model

- At a given point in time, some historical crawl order has already been executed (P1, P2, P3, P4, P5 in the diagram).
- Some future crawl order has been planned (P6, P7, P4, P8, ...).



Model

- Pages downloaded by the crawler are stored in a *repository*.
- The future crawl order is determined, at least in part, by analyzing the repository.
- The content of a web page is subject to change over time.
- It is sometimes desirable to re-download a page that has already been downloaded, to obtain a more recent snapshot of its content.

Model: Two Approaches

- **Batch crawling**:
  - The crawl order does not contain duplicate occurrences of any page
  - The entire crawling process is periodically halted and restarted as a way to obtain more recent snapshots of previously crawled pages.
- **Incremental crawling**:
  - Pages may appear multiple times in the crawl order.
  - Crawling is a continuous process that conceptually never terminates.
- Most modern commercial crawlers perform incremental crawling, which is more powerful because it allows re-visitation of pages at different rates.

Web Characteristics

- One notable study is by Broder et al. [26], which uncovered a “bowtie” structure consisting of:
  - a central strongly connected component (the core).
  - a component that can reach the core but cannot be reached from the core.
  - a component that can be reached from the core but cannot reach the core.
- In addition, there are a number of small, irregular structures such as disconnected components and long “tendrils”.

Web Characteristics

- In an earlier study, Broder et al. [28] showed that there is an abundance of near-duplicate content of the web.
- Using a corpus of 30 million web pages collected by the AltaVista crawler, they found that:
  - 29% of the pages were more than 50% similar to other pages in the corpus
  - 11% of the pages were exact duplicates of other pages.
- Sources of near-duplication include mirroring of sites (or portions of sites) and URL synonymy.

Taxonomy of Crawl Ordering Policies

- Next figure presents a high-level taxonomy of published crawl ordering techniques.
- The first group of techniques focuses exclusively on ordering pages for first-time downloading, which affects *coverage*.
  - Can be applied either in the batch crawling scenario, or in the incremental crawling scenario in conjunction with a separate policy of the second group.
- The second group of techniques governs re-downloading of pages to maintain *freshness*.
- Techniques in the third group consider the combined problem of interleaving first-time downloads with re-downloads, to balance coverage and freshness.

Taxonomy of Crawl Ordering Policies

Technique	Objectives		Factors considered		
	Coverage	*Freshness*	*Importance*	*Relevance*	*Dynamicity*
Breadth-first search [43, 95, 108]	√				
Prioritize by indegree [43]	√		√		
Prioritize by PageRank [43, 45]	√		√		
Prioritize by site size [9]	√		√		
Prioritize by spawning rate [48]	√				√
Prioritize by search impact [104]	√		√	√	
Scoped crawling (Section 4.2)	√			√	
Minimize obsolescence [41, 46]		√	√		√
Minimize age [41]		√	√		√
Minimize incorrect content [99]		√	√		√
Minimize embarrassment [115]		√	√	√	√
Maximize search impact [103]		√	√	√	√
Update capture (Section 5.2)		√		√	√
WebFountain [56]	√	√			√
OPIC [1]	√	√	√		

References

[1] S. Abiteboul, M. Preda, and G. Cobena, “Adaptive on-line page importance computation,” in Proceedings of the 12th International World Wide Web Conference, 2003.

[2] E. Adar, J. Teevan, S. T. Dumais, and J. L. Elsas, “The web changes everything: Understanding the dynamics of web content,” in Proceedings of the 2nd International Conference on Web Search and Data Mining, 2009.

[3] Advanced Triage (medical term), http://en.wikipedia.org/wiki/Triage# Advanced triage.

[4] A. Agarwal, H. S. Koppula, K. P. Leela, K. P. Chitrapura, S. Garg, P. K. GM, C. Haty, A. Roy, and A. Sasturkar, “URL normalization for de-duplication of web pages,” in Proceedings of the 18th Conference on Information and Knowledge Management, 2009.

[5] C. C. Aggarwal, F. Al-Garawi, and P. S. Yu, “Intelligent crawling on the world wide web with arbitrary predicates,” in Proceedings of the 10th International World Wide Web Conference, 2001.

[6] D. Ahlers and S. Boll, “Adaptive geospatially focused crawling,” in Proceedings of the 18th Conference on Information and Knowledge Management, 2009.

[7] Attributor. http://www.attributor.com.

[8] R. Baeza-Yates and C. Castillo, “Crawling the infinite web,” Journal of Web Engineering, vol. 6, no. 1, pp. 49–72, 2007.

References

[9] R. Baeza-Yates, C. Castillo, M. Marin, and A. Rodriguez, “Crawling a country: Better strategies than breadth-first for web page ordering,” in Proceedings of the 14th International World Wide Web Conference, 2005.

[10] B. Bamba, L. Liu, J. Caverlee, V. Padliya, M. Srivatsa, T. Bansal, M. Palekar, J. Patrao, S. Li, and A. Singh, “DSphere: A source-centric approach to crawling, indexing and searching the world wide web,” in Proceedings of the 23rd International Conference on Data Engineering, 2007.

[11] Z. Bar-Yossef and M. Gurevich, “Random sampling from a search engine’s index,” in Proceedings of the 15th International World Wide Web Conference, 2006.

[12] Z. Bar-Yossef, I. Keidar, and U. Schonfeld, “Do not crawl in the DUST: Different URLs with similar text,” in Proceedings of the 16th International World Wide Web Conference, 2007.

[13] L. Barbosa and J. Freire, “Siphoning hidden-web data through keyword-based interfaces,” in Proceedings of the 19th Brazilian Symposium on Databases SBBD, 2004.

[14] L. Barbosa and J. Freire, “An adaptive crawler for locating hidden-web entry points,” in Proceedings of the 16th International World Wide Web Conference, 2007.

[15] L. Barbosa, A. C. Salgado, F. de Carvalho, J. Robin, and J. Freire, “Looking at both the present and the past to efficiently update replicas of web content,” in Proceedings of the ACM International Workshop on Web Information and Data Management, 2005.

References

[16] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates, “Link- based characterization and detection of web spam,” in Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web, 2006.

[17] A. Benczúr, K. Csalogány, T. Sarlós, and M. Uher, “Spamrank — fully automatic link spam detection,” in Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, 2005.

[18] K. Bharat and A. Broder, “Mirror, mirror on the web: A study of host pairs with replicated content,” in Proceedings of the 8th International World Wide Web Conference, 1999.

[19] K. Bharat, A. Broder, J. Dean, and M. Henzinger, “A comparison of techniques to find mirrored hosts on the WWW,” Journal of the American Society for Information Science, vol. 51, no. 12, pp. 1114–1122, 2000.

[20] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” Communications of the ACM, vol. 13, no. 7, pp. 422–426, 1970.

[21] P. Boldi, B. Codenotti, , M. Santini, and S. Vigna, “UbiCrawler: A scalable fully distributed web crawler,” Software — Practice & Experience, vol. 34, no. 8, pp. 711–726, 2004.

[22] P. Boldi, B. Codenotti, M. Santini, and S. Vigna, “Structural properties of the African web,” in Poster Proceedings of the 11th International World Wide Web Conference, 2002.

[23] P. Boldi, M. Santini, and S. Vigna, “Paradoxical effects in pagerank incremental computations,” Internet Mathematics, vol. 2, no. 3, pp. 387–404, 2005.

References

[24] C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz, “The Harvest information discovery and access system,” in Proceedings of the 2nd International World Wide Web Conference, 1994.

[25] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” in Proceedings of the 7th International World Wide Web Conference, 1998.

[26] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, “Graph structure in the web,” in Proceedings of the 9th International World Wide Web Conference, 2000.

[27] A. Broder, M. Najork, and J. Wiener, “Efficient URL caching for World Wide Web crawling,” in Proceedings of the 12th International World Wide Web Conference, 2003.

[28] A. Z. Broder, S. C. Glassman, and M. S. Manasse, “Syntactic clustering of the web,” in Proceedings of the 6th International World Wide Web Conference, 1997.

[29] M. Burner, “Crawling towards eternity: Building an archive of the world wide web,” Web Techniques Magazine, vol. 2, no. 5, pp. 37–40, 1997.

[30] J. Callan, “Distributed information retrieval,” in Advances in Information Retrieval, (W. B. Croft, ed.), pp. 127–150, Kluwer Academic Publishers, 2000.

[31] J. Callan and M. Connell, “Query-based sampling of text databases,” ACM Transactions on Information Systems, vol. 19, no. 2, pp. 97–130, 2001.

References

[32] J. P. Callan, Z. Lu, and W. B. Croft, “Searching distributed collections with inference networks,” in Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995.

[33] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg, “Automatic resource compilation by analyzing hyperlink structure and associated text,” in Proceedings of the 7th International World Wide Web Conference, 1998.

[34] S. Chakrabarti, M. van den Berg, and B. Dom, “Focused crawling: A new approach to topic-specific web resource discovery,” in Proceedings of the 8th International World Wide Web Conference, 1999.

[35] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, “Structured databases on the web: Observations and implications,” ACM SIGMOD Record, vol. 33, no. 3, pp. 61–70, 2004.

[36] K. Chellapilla and A. Maykov, “A taxonomy of JavaScript redirection spam,” in Proceedings of the 16th International World Wide Web Conference, 2007.

[37] H. Chen, M. Ramsey, and C. Yang, “A smart itsy bitsy spider for the web,” Journal of the American Society for Information Science, vol. 49, no. 7, pp. 604–618, 1998.

[38] S. Chien, C. Dwork, R. Kumar, D. R. Simon, and D. Sivakumar, “Link evolution: Analysis and algorithms,” Internet Mathematics, vol. 1, no. 3, pp. 277–304, 2003.

[39] J. Cho and H. García-Molina, “The evolution of the web and implications for an incremental crawler,” in Proceedings of the 26th International Conference on Very Large Data Bases, 2000.

References

[40] J. Cho and H. García-Molina, “Parallel crawlers,” in Proceedings of the 11th International World Wide Web Conference, 2002.

[41] J. Cho and H. García-Molina, “Effective page refresh policies for web crawlers,” ACM Transactions on Database Systems, vol. 28, no. 4, pp. 390–426, 2003.

[42] J. Cho and H. García-Molina, “Estimating frequency of change,” ACM Transactions on Internet Technology, vol. 3, no. 3, pp. 256–290, 2003.

[43] J. Cho, J. Garcíıa-Molina, and L. Page, “Efficient crawling through URL ordering,” in Proceedings of the 7th International World Wide Web Conference, 1998.

[44] J. Cho and A. Ntoulas, “Effective change detection using sampling,” in Proceedings of the 28th International Conference on Very Large Data Bases, 2002.

[45] J. Cho and U. Schonfeld, “RankMass crawler: A crawler with high personalized PageRank coverage guarantee,” in Proceedings of the 33rd International Conference on Very Large Data Bases, 2007.

[46] E. G. Coffman, Z. Liu, and R. R. Weber, “Optimal robot scheduling for web search engines,” Journal of Scheduling, vol. 1, no. 1, 1998.

[47] CrawlTrack, “List of spiders and crawlers,” http://www.crawltrack.net/ crawlerlist.php.

[48] A. Dasgupta, A. Ghosh, R. Kumar, C. Olston, S. Pandey, and A. Tomkins, “The discoverability of the web,” in Proceedings of the 16th International World Wide Web Conference, 2007.

References

[49] A. Dasgupta, R. Kumar, and A. Sasturkar, “De-duping URLs via rewrite rules,” in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008.

[50] B. Davison, “Recognizing nepotistic links on the web,” in Proceedings of the AAAI-2000 Workshop on Artificial Intelligence for Web Search, 2000.

[51] G. T. de Assis, A. H. F. Laender, M. A. Gonçalves, and A. S. da Silva, “A genre-aware approach to focused crawling,” World Wide Web, vol. 12, no. 3, pp. 285–319, 2009.

[52] J. Dean and M. Henzinger, “Finding related pages in the world wide web,” in Proceedings of the 8th International World Wide Web Conference, 1999.

[53] P. DeBra and R. Post, “Information retrieval in the world wide web: Making client-based searching feasible,” in Proceedings of the 1st International World Wide Web Conference, 1994.

[54] M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles, and M. Gori, “Focused crawling using context graphs,” in Proceedings of the 26th International Conference on Very Large Data Bases, 2000.

[55] C. Duda, G. Frey, D. Kossmann, and C. Zhou, “AJAXSearch: Crawling, indexing and searching web 2.0 applications,” in Proceedings of the 34th International Conference on Very Large Data Bases, 2008.

[56] J. Edwards, K. S. McCurley, and J. A. Tomlin, “An adaptive model for optimizing performance of an incremental web crawler,” in Proceedings of the 10th International World Wide Web Conference, 2001.

References

[57] D. Eichmann, “The RBSE spider — Balancing effective search against web load,” in Proceedings of the 1st International World Wide Web Conference, 1994.

[58] D. Fetterly, N. Craswell, and V. Vinay, “The impact of crawl policy on web search effectiveness,” in Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2009.

[59] D. Fetterly, M. Manasse, and M. Najork, “Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages,” in Proceedings of the 7th International Workshop on the Web and Databases, 2004.

[60] D. Fetterly, M. Manasse, M. Najork, and J. L. Wiener, “A large-scale study of the evolution of web pages,” in Proceedings of the 12th International World Wide Web Conference, 2003.

[61] R. Fielding, “Maintaining distributed hypertext infostructures: Welcome to MOMspider’s web,” in Proceedings of the 1st International World Wide Web Conference, 1994.

[62] A. S. Foundation, “Welcome to Nutch!,” http://lucene.apache.org/nutch/.

[63] W. Gao, H. C. Lee, and Y. Miao, “Geographically focused collaborative crawling,” in Proceedings of the 15th International World Wide Web Conference, 2006.

[64] GigaAlert, http://www.gigaalert.com.

[65] D. Gomes and M. J. Silva, “Characterizing a national community web,” ACM Transactions on Internet Technology, vol. 5, no. 3, pp. 508–531, 2005.

References

[66] L. Gravano, H. García-Molina, and A. Tomasic, “The effectiveness of GlOSS for the text database discovery problem,” in Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, 1994.

[67] M. Gray, “Internet growth and statistics: Credits and background,” http://www.mit.edu/people/mkgray/net/background.html.

[68] D. Gruhl, L. Chavet, D. Gibson, J. Meyer, P. Pattanayak, A. Tomkins, and J. Zien, “How to build a WebFountain: An architecture for very large-scale text analytics,” IBM Systems Journal, vol. 43, no. 1, pp. 64–77, 2004.

[69] Z. Gyöngyi and H. García-Molina, “Web Spam Taxonomy,” in Proceedings of the 1st International Workshop on Adversarial Information Retrieval, 2005.

[70] Y. Hafri and C. Djeraba, “High performance crawling system,” in Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval, 2004.

[71] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, “Measuring index quality using random walks on the web,” in Proceedings of the 8th International World Wide Web Conference, 1999.

[72] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, “On near- uniform URL sampling,” in Proceedings of the 9th International World Wide Web Conference, 2000.

[73] M. R. Henzinger, R. Motwani, and C. Silverstein, “Challenges in web search engines,” SIGIR Forum, vol. 36, no. 2, pp. 11–22, 2002.

References

[74] M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, “The shark-search algorithm — An application: Tailored web site mapping,” in Proceedings of the 7th International World Wide Web Conference, 1998.

[75] A. Heydon and M. Najork, “Mercator: A scalable, extensible web crawler,” World Wide Web, vol. 2, no. 4, pp. 219–229, 1999.

[76] International Workshop Series on Adversarial Information Retrieval on the Web, 2005.

[77] Internet Archive, http://archive.org/.

[78] Internet Archive, “Heritrix home page,” http://crawler.archive.org/.

[79] K. Jarvelin and J. Kekalainen, “Cumulated gain-based evaluation of IR techniques,” ACM Transactions on Information Systems, vol. 20, no. 4, pp. 422–446, 2002.

[80] J. Johnson, K. Tsioutsiouliklis, and C. L. Giles, “Evolving strategies for focused web crawling,” in Proceedings of the 20th International Conference on Machine Learning, 2003.

[81] R. Khare, D. Cutting, K. Sitakar, and A. Rifkin, “Nutch: A flexible and scalable open-source web search engine,” Technical Report, CommerceNet Labs, 2004.

[82] J. Kleinberg, “Authoritative sources in a hyperlinked environment,” Journal of the ACM, vol. 46, no. 5, pp. 604–632, 1999.

[83] M. Koster, “A standard for robot exclusion,” http://www.robotstxt.org/ orig.html, 1994.

References

[84] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov, “IRLbot: Scaling to 6 billion pages and beyond,” in Proceedings of the 17th International World Wide Web Conference, 2008.

[85] L. Lim, M. Wang, S. Padmanabhan, J. S. Vitter, and R. C. Agarwal, “Characterizing web document change,” in Proceedings of the International Conference on Advances in Web-Age Information Management, 2001.

[86] L. Liu, C. Pu, W. Tang, and W. Han, “CONQUER: A continual query system for update monitoring in the WWW,” International Journal of Computer Systems, Science and Engineering, vol. 14, no. 2, 1999.

[87] B. T. Loo, O. Cooper, and S. Krishnamurthy, “Distributed web crawling over DHTs,” UC Berkeley Technical Report CSD-04-1305, 2004.

[88] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, “Google’s deep-web crawl,” in Proceedings of the 34th International Conference on Very Large Data Bases, 2008.

[89] M. Mauldin, “Lycos: Design choices in an internet search service,” IEEE Expert, vol. 12, no. 1, pp. 8–11, 1997.

[90] O. A. McBryan, “GENVL and WWWW: Tools for taming the web,” in Proceedings of the 1st International World Wide Web Conference, 1994.

[91] F. Menczer and R. K. Belew, “Adaptive retrieval agents: Internalizing local context and scaling up to the web,” Machine Learning, vol. 39, pp. 203–242, 2000.

References

[92] F. Menczer, G. Pant, and P. Srinivasan, “Topical web crawlers: Evaluating adaptive algorithms,” ACM Transactions on Internet Technology, vol. 4, no. 4, pp. 378–419, 2004.

[93] G. Mohr, M. Stack, I. Ranitovic, D. Avery, and M. Kimpton, “An introduction to Heritrix, an open source archival quality web crawler,” in Proceedings of the 4th International Web Archiving Workshop, 2004.

[94] M. Najork and A. Heydon, “High-performance web crawling,” Technical report, Compaq SRC Research Report 173, 2001.

[95] M. Najork and J. L. Wiener, “Breadth-first search crawling yields high-quality pages,” in Proceedings of the 10th International World Wide Web Conference, 2001.

[96] A. Ntoulas, J. Cho, and C. Olston, “What’s new on the web? The evolution of the web from a search engine perspective,” in Proceedings of the 13th International World Wide Web Conference, 2004.

[97] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, “Detecting spam web pages through content analysis,” in Proceedings of the 15th International World Wide Web Conference, 2006.

[98] A. Ntoulas, P. Zerfos, and J. Cho, “Downloading textual hidden web content through keyword queries,” in Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, 2005.

[99] C. Olston and S. Pandey, “Recrawl scheduling based on information longevity,” in Proceedings of the 17th International World Wide Web Conference, 2008.

References

[100] V. J. Padliya and L. Liu, “Peercrawl: A decentralized peer-to-peer architecture for crawling the world wide web,” Georgia Institute of Technology Technical Report, 2006.

[101] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: Bringing order to the web,” Technical Report, Stanford University, 1998.

[102] S. Pandey, K. Dhamdhere, and C. Olston, “WIC: A general-purpose algorithm for monitoring web information sources,” in Proceedings of the 30th International Conference on Very Large Data Bases, 2004.

[103] S. Pandey and C. Olston, “User-centric web crawling,” in Proceedings of the 14th International World Wide Web Conference, 2005.

[104] S. Pandey and C. Olston, “Crawl ordering by search impact,” in Proceedings of the 1st International Conference on Web Search and Data Mining, 2008.

[105] S. Pandey, K. Ramamritham, and S. Chakrabarti, “Monitoring the dynamic web to respond to continuous queries,” in Proceedings of the 12th International World Wide Web Conference, 2003.

[106] G. Pant and P. Srinivasan, “Learning to crawl: Comparing classification schemes,” ACM Transactions on Information Systems, vol. 23, no. 4, pp. 430–462, 2005.

[107] G. Pant and P. Srinivasan, “Link contexts in classifier-guided topical crawlers,” IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 1, pp. 107–122, 2006.

[108] B. Pinkerton, “Finding what people want: Experiences with the WebCrawler,” in Proceedings of the 2nd International World Wide Web Conference, 1994.

References

[109] S. Raghavan and H. García-Molina, "Crawling the hidden web," in Proceedings of the 27th International Conference on Very Large Data Bases, 2001.

[110] U. Schonfeld and N. Shivakumar, "Sitemaps: Above and beyond the crawl of duty," in Proceedings of the 18th International World Wide Web Conference, 2009.

[111] V. Shkapenyuk and T. Suel, "Design and implementation of a high- performance distributed web crawler," in Proceedings of the 18th International Conference on Data Engineering, 2002.

[112] A. Singh, M. Srivatsa, L. Liu, and T. Miller, "Apoidea: A decentralized peer- to-peer architecture for crawling the world wide web," in SIGIR Workshop on Distributed Information Retrieval, 2003.

[113] Q. Tan, Z. Zhuang, P. Mitra, and C. L. Giles, "A clustering-based sampling approach for refreshing search engine's database," in Proceedings of the 10th International Workshop on the Web and Databases, 2007.

[114] T. Urvoy, T. Lavergne, and P. Filoche, "Tracking web spam with hidden style similarity," in Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web, 2006.

[115] J. L. Wolf, M. S. Squillante, P. S. Yu, J. Sethuraman, and L. Ozsen, "Optimal crawling strategies for web search engines," in Proceedings of the 11th International World Wide Web Conference, 2002.

[116] B. Wu and B. Davison, "Identifying link farm spam pages," in Proceedings of the 14th International World Wide Web Conference, 2005.

References

[117] P. Wu, J.-R. Wen, H. Liu, and W.-Y. Ma, “Query selection techniques for efficient crawling of structured web sources,” in Proceedings of the 22nd International Conference on Data Engineering, 2006.

[118] Yahoo! Research Barcelona, “Datasets for web spam detection,” http://www.yr-bcn.es/webspam/datasets.

[119] J.-M. Yang, R. Cai, C. Wang, H. Huang, L. Zhang, and W.-Y. Ma, “Incorporating site-level knowledge for incremental crawling of web forums: A list-wise strategy,” in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009.

[120] S. Zheng, P. Dmitriev, and C. L. Giles, “Graph-based seed selection for web- scale crawlers,” in Proceedings of the 18th Conference on Information and Knowledge Management, 2009.

[121] K. Zhu, Z. Xu, X. Wang, and Y. Zhao, “A full distributed web crawler based on structured network,” in Asia Information Retrieval Symposium, 2008.