IJCSC International Journal of Computer Science and Communication Vol. 2, No. 1, January-June 2011, pp. 265-267

WEB CRAWLER - AN OVERVIEW

S.S. Dhenakaran[1] and K. Thirugnana Sambanthan[2]

[1]Department of Computer Science and Engineering, Alagappa University, Karaikudi, India, *E-mail: ssdarvind@yahoo.com*

[2]Department of IT, Indigrow Institute of Professional Studies, Coimbatore, India, *E-mail: shivaperuman@gmail.com*

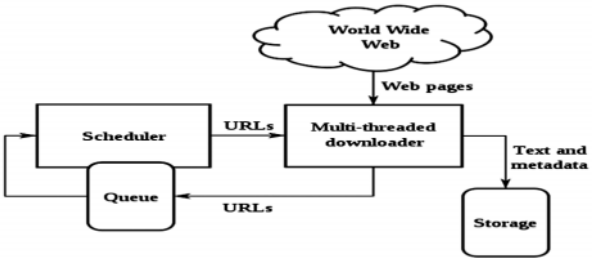
ABSTRACT

A **Web crawler** is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. Web crawling is an important method for collecting data on, and keeping up with, the rapidly expanding Internet. A vast number of web pages are continually being added every day, and information is constantly changing. This Paper is an overview of various types of Web Crawlers and the policies like selection, re-visit, politeness, parallelization involved in it. The behavioral pattern of the Web crawler based on these policies is also taken for the study. The evolution of these web crawler from Basic general purpose web crawler to the latest Adaptive web crawler is studied.

Keywords: Web Crawler, Behavior, Policies.

1. INTRODUCTION

WWW provides us with great amounts of useful information electronically available as hypertext. This large pool of hypertext is changing dynamically and semantically unstructured, making us finding the related and valuable information difficult. Therefore, a web crawler for automatic discovering of valuable information from the Web, or Web Mining is important for us nowadays. In reality, this web crawler is a program, which automatically traverses the web by downloading documents and following links from page to page. They are mainly used by search engine to gather data for indexing. Other possible applications include page validation, structural analysis and visualization, update notification, mirroring and personal web assistants/ agents etc. Web crawlers are also known as spiders, robots, worms etc.



Architecture of Web Crawler

There are important characteristics of the Web that make crawling very difficult:

- its large volume,
- its fast rate of change, and
- dynamic page generation.

The large volume implies that the crawler can only download a fraction of the Web pages within a given time, so it needs to prioritize its downloads. The high rate of change implies that by the time the crawler is downloading the last pages from a site, it is very likely that new pages have been added to the site, or that pages have already been updated or even deleted.

2. GENERAL-PURPOSE WEB CRAWLER

General-purpose web crawlers collect and process the entire contents of the Web in a centralized location, so that it can be indexed in advance to be able to respond to many user queries. In the early stage when the Web is still not very large, simple or random crawling method was enough to index the whole web. However, after the Web has grown very large, a crawler can have large coverage but rarely refresh its crawls, or a crawler can have good coverage and fast refresh rates but not have good ranking functions or support advanced query capabilities that need more processing power. Therefore, more advance crawling methodologies are needed due to the limited resources like time and network bandwidth.

3. BEHAVIOR OF WEB CRAWLER

The behavior of a Web crawler is the outcome of a combination of policies

- a *selection policy* that states which pages to download,
- a *re-visit policy* that states when to check for changes to the pages,
- a *politeness policy* that states how to avoid overloading Web sites, and
- a *parallelization policy* that states how to coordinate distributed Web crawlers.

3.1 SELECTION POLICY

Large search engines cover only a portion of the publicly-available part. As a crawler always downloads just a fraction of the Web pages, it is highly desirable that the downloaded fraction contains the most relevant pages and not just a random sample of the Web. This requires a metric of importance for prioritizing Web pages. The importance of a page is a function of its intrinsic quality, its popularity in terms of links or visits, and even of its URL.

Abiteboul designed a crawling strategy based on an algorithm called OPIC (On-line Page Importance Computation).[1] In OPIC, each page is given an initial sum of "cash" that is distributed equally among the pages it points to. It is similar to a Pagerank computation, but it is faster and is only done in one step. An OPIC-driven crawler downloads first the pages in the crawling frontier with higher amounts of "cash". Experiments were carried in a 100,000-pages synthetic graph with a power-law distribution of in-links. However, there was no comparison with other strategies nor experiments in the real Web.

Daneshpajouh *et al.*[2] designed a community based algorithm for discovering good seeds. Their method crawls web pages with high PageRank from different communities in less iteration in comparison with crawl starting from random seeds. One can extract good seed from a previously-crawled-Web graph using this new method. Using these seeds a new crawl can be very effective.

3.1.1 Topic-focused Web Crawling

Topic-Focused Web Crawling **[chakra99, michelan00, chakra02]** initiation was motivated by the fact the Web is huge with an unprecedented scaling problem, but most people are only interested in a small fraction of the Web. The main objective is to only crawl on a small fraction of the Web to discover the set of pages covering a certain topic. This is essential because of the finite crawling resources such as time, network bandwidth and storage.

3.1.2 URL Normalization

Crawlers usually perform some type of URL Normalization in order to avoid crawling the same resource more than once. The term URL Normalization, also called *URL* Canonicalization, refers to the process of modifying and standardizing a URL in a consistent manner. There are several types of normalization that may be performed including conversion of URLs to lowercase, removal of "." and ".." segments, and adding trailing slashes to the non-empty path component.[3]

3.1.3 Path-ascending Crawling

Path-ascending crawlers are also known as Web harvesting software, because they're used to "harvest" or collect all the contents from a specific page or host. Some crawlers intend to download as many resources as possible from a particular web site. So Path-Ascending Crawler was introduced that would ascend to every path in each URL that it intends to crawl. [4]Cothey found that a path-ascending crawler was very effective in finding isolated resources, or resources for which no inbound link would have been found in regular crawling.

3.2 RE-VISIT POLICY

The Web has a very dynamic nature, and crawling a fraction of the Web can take weeks or months. By the time a Web crawler has finished its crawl, many events could have happened, including creations, updates and deletions. From the search engine's point of view, there is a cost associated with not detecting an event, and thus having an outdated copy of a resource. The most-used cost functions are freshness and age[5].

Freshness: This is a binary measure that indicates whether the local copy is accurate or not. The freshness of a page p in the repository at time t is defined as:

$$F_p(t) = \begin{cases} 1 & \text{if } p \text{ is equal to the local copy at time } t \\ 0 & \text{otherwise} \end{cases}$$

Age: This is a measure that indicates how outdated the local copy is. The age of a page p in the repository, at time t is defined as:

$$A_p(t) = \begin{cases} 0 & \text{if } p \text{ is not modified at time } t \\ t - \text{modification time of } p & \text{otherwise} \end{cases}$$

Coffman *et al.* worked with a definition of the objective of a Web crawler that is equivalent to freshness, but use a different wording: they propose that a crawler must minimize the fraction of time pages remain outdated. They also noted that the problem of Web crawling can be modeled as a multiple-queue, single-server polling system, on which the Web crawler is the server and the Web sites are the queues. Page modifications are the arrival of the customers, and switch-over times are the interval between page accesses to a single Web site. Under this model, mean waiting time for a customer in the polling system is equivalent to the average age for the Web crawler. [6]The objective of the crawler is to keep the average freshness of pages in its collection as high as possible, or to keep the average age of pages as low as possible. These objectives are not equivalent: in the first case, the crawler is just concerned with how many pages are out-dated, while in the second case, the crawler is concerned with how old the local copies of pages are.

3.3 POLITENESS POLICY

Crawlers can retrieve data much quicker and in greater depth than human searchers, so they can have a crippling impact on the performance of a site. Needless to say, if a single crawler is performing multiple requests per second

and/or downloading large files, a server would have a hard time keeping up with requests from multiple crawlers. The use of Web crawlers is useful for a number of tasks, but comes with a price for the general community. The costs of using Web crawlers include:

- network resources, as crawlers require considerable bandwidth and operate with a high degree of parallelism during a long period of time;
- server overload, especially if the frequency of accesses to a given server is too high;
- poorly-written crawlers, which can crash servers or routers, or which download pages they cannot handle; and
- personal crawlers that, if deployed by too many users, can disrupt networks and Web servers.

3.4 PARALLELIZATION POLICY

A Parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as the same URL can be found by two different crawling processes.

4. ADAPTIVE CRAWLER

Adaptive crawler **[edward00]** is classified as an incremental type of crawler which will continually crawl the entire web, based on some set of crawling cycles. The adaptive model used would use data from previous cycles to decide which pages should be checked for updates. Adaptive Crawling can also be viewed as an extension of focused crawling technology. It has the basic concept of doing focus crawling with additional adaptive crawling ability. Since the web is changing dynamically, adaptive crawler is designed to crawl the web more dynamically, by additionally taking into consideration more important parameters such as freshness or up to date-ness, whether pages are obsolete, the way pages change, when pages will change, how often pages change and etc. These parameters will be added into the optimization model for controlling the crawling strategy, and contribute to defining the discrete time period and crawling cycle. Therefore, it is expected that more cycles the adaptive crawler goes in operation, more reliable and refined will the output results.

5. CONCLUSION

This paper describes about Different types of Web crawler and the policies used in the web crawlers. The evolution of these web crawlers are studied. A web crawler is a way for the search engines and other users to regularly ensure that their databases are up to date. Web crawlers are a central part of search engines, and details on their algorithms and architecture are kept as business secrets. When crawler designs are published, there is often an important lack of detail that prevents others from reproducing the work. There are also emerging concerns about" Search Engine Spamming", which prevent major search engines from publishing their ranking algorithms. New Modification and extension of the techniques in Web crawling should be next topics in this area of research.

REFERENCES

[1] Abiteboul Serge, Mihai Preda, and Gregory Cobena (2003). "Adaptive On-line Page Importance Computation". *Proceedings of the 12th International Conference on World Wide Web*.

[2] Shervin Daneshpajouh, Mojtaba Mohammadi Nasiri, Mohammad Ghodsi, "A Fast Community Based Algorithm for Generating Crawler Seeds Set".

[chakra 99] Soumen Chakrabarti, and Martin Van Den Berg, and Byron Dom, "Focused Crawling: a New Approach to Topic-specific {Web} Resource Discovery". *Computer Networks*, **31 (11-16)**, pp. 1623-1640, 1999.

[michelan00] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C. Lee Giles, and Marco Gori, "Focused Crawling using Context Graphs", *26th International Conference on Very Large Databases*, VLDB 2000, Cairo, Egypt 2000

[chakra02] S. Chakrabarti, K. Punera, M. Subramanyam, "Accelerated Focused Crawling through Online Relevance Feedback", *World Wide Web (ACM)*, Hawaii 2002.

[3] Pant Gautam, Srinivasan Padmini, Menczer Filippo, (2004). "Crawling the Web" *In Levene, Mark; Poulovassilis, Alexandra. Web Dynamics: Adapting to Change in Content, Size, Topology and Use*. Springer. pp. 153-178.

[4] Cothey Viv, (2004). "Web-crawling Reliability". *Journal of the American Society for Information Science and Technology*, **55 (14)**, 1228-1238.

[5] Cho Junghoo, Hector Garcia-Molina, (2000). "Synchronizing a Database to Improve Freshness". *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. Dallas, Texas, United States:

[6] Jr. E.G. Coffman, Zhen Liu, Richard R. Weber, (1998). "Optimal Robot Scheduling for Web Search Engines". *Journal of Scheduling*.

[edward00] J. Edwards, K. McCurley, and J. Tomlin, "An Adaptive Model for Optimizing Performance of an Incremental Web Crawler", *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, May 2001.