Analyzing Different Web Crawling Methods

Bhavin M. Jasani
Department of computer Science,
Saurashtra University, Rajkot, Gujarat (India).

C. K. Kumbharana
Department of computer Science,
Saurashtra University, Rajkot, Gujarat (India).

ABSTRACT
As we know that the no of internet users are increasing day by day at a enormous rate. To maintain the resource discovery of World Wide Web (WWW) is a crucial task in today's scenario. There are many algorithms and architectures have been introduced to make effective WWW resource discovery.

Keywords
WWW, Web Crawling, Web Tree, Web Spamming, Crawling Algorithms, Querying.

1. INTRODUCTION
The World Wide Web has become more accessible across the globe, the process of availability and accuracy of the information on the WWW is become more important. To maintain the resource discovery of World Wide Web (WWW) is a crucial task in today's scenario. The omnipresent use of the Web has raised important social concerns in the areas of privacy, restriction, and access to information. Due to the abundance of data on the web and different user perspective, information retrieval becomes a challenge . So the search engines have a bigger job of sorting out the results, in the order of interestingness of the user within the first page of appearance and a quick summary of the information provided on a page.

2. WEB CRAWLING BASICS
A Web crawler is an application or a set of instructions that scans the Web pages in a systematic and automated manner to categorize the information on the basis of user demand. Other terms for Web crawlers are ants, automatic indexers, bots, and worms or Web spider, Web robot, Web scutters.

This process is called Web crawling or spidering. Many sites, in particular search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting email addresses (usually for spam).

3. GENERAL CRAWLING STRATEGIES
There are many highly accomplished techniques in terms of Web crawling strategy. The researcher describe most relevant here:

3.1 Breadth-First Crawling
The Breadth-First search algorithm performs the unique search around the neighbor nodes(hyperlinks). It start by following the root node (Hyperlink) and scans the all the neighbor nodes at the initial level. If the targeted search is achieved then the scanning is stopped otherwise it leads to the next level.

Such types of algorithms are best suited where the branches are small and resultant objective is identical. When the branches or tree is very deep then this algorithm will not perform well, i.e. all path traversals leads to the same resultant node.

3.2 Depth First Crawling
The Depth First search algorithm starts searching the objective from the root node and traverse next to its child node, If there are more than one child node, then left most node is given highest priority and traverse deep until no more child node is present. Then it starts from the next unvisited node and then continues in a similar manner.

By using this algorithm the assurance of scanning of all node is achieved but when the number of child node is large then this algorithm takes more time and might go in to infinite.

3.3 Targeted Crawling
Search engines use random crawling process in order to target a certain type of page, e.g. pages on a specific topic or in a particular language, images, mp3 files, geo location, domain specific or scientific papers. In addition to these heuristics, more generic approaches have been suggested. They are based on the analysis of the structures of hypertext links and techniques of learning: the objective here is being to retrieve the greatest number of pages relating to a particular subject by using the minimum bandwidth. Most of the studies cited in this category do not use high performance crawlers, yet succeed in producing acceptable results.

3.4 Page Rank Algorithm
This algorithm works on the importance of the web pages. It calculates inlinks or backlinks to that page. Then the page rank is given to each page as per bellow formula.

PR(A) = (1-d)+d(PR(T1)/C(T1))+…PR(Tn)/C(Tn))

Where, PR(A) :-> Page Rank of Site.

d :-> damping factor.

T1,….Tn :-> no. of links.

After determining a page rank of a website the index has been generated to show the relevant on a website contain to the search keywords.

Table.1 : Comparison on Crawling Algorithms

Algorithm	Search Pattern	Benifits	Drawbacks
Breadth-First Crawling	Scans neighbor node from root level, if result not achieved then go to next level.	where the branches are small and resultant objective is identical.	When the branches or tree is very deep then goes into infinite.

Depth First Crawling	Scans from the root node and traverse next to its child leftmost node	the assurance of scanning of all node is achieved	Takes more time when the child node is large.
Targeted Crawling	Uses random (heuristics) crawling process	retrieves the greatest number of pages relating to a particular subject by using the minimum bandwidth.	Takes more time when specific topics are very large.
Page Rank Algorithm	works on the importance of the web pages. It calculates inlinks or backlinks to that page.	More accurate search result.	Difficult to manage and update page index repository.

4. CRAWLING ALGORITHMS

We now discuss a number of crawling algorithms that are suggested in the literature. Note that many of these algorithms are variations of the best-first scheme. The difference is in the heuristics they use to score the unvisited URLs with some algorithms adapting and tuning their parameters before or during the crawl.

4.1 Naive Best-First Crawler

A naive best-first represents the collection of fetched URLs as a vector. In the study of crawler evaluation the Naïve Best-First Crawler was one of the most evaluated algorithm by the authors. In this algorithm the cosine similarity of the page with the query or description provided by the user is calculated. Then weight can be generated for unvisited URLs on each page by this cosine value. After that the URL is inserted to the vector based on the cosine weight. The crawler iterates by picking the best URL in the vector to crawl and returns new unvisited URLs that are again inserted in to the vector based on the cosine weight of the parent page.

4.2 SharkSearch

SharkSearch [15] is a more aggressive version of FishSearch [12] with some improvements. In Fish-Search crawling, the crawler search more broadly in the areas of the web where number of relevant page found is more. At the same time the crawler skip the areas where the relevant pages are not found. SharkSearch uses a similar valued function to measure the relevance as opposite to the binary relevance function of Fish-Search. In addition of these, SharkSearch has a more sophisticated concept of potential scores for the links in the crawl frontier. The Potential score of links is influenced by anchor text, link context, and inherited score from ancestors(incoming and outgoing links URLs of the page).

4.3 Focused Crawler

Chakrabarti has invented a focused crawler based o a hypertext classifier *et al.* [9, 6]. The main aim of the crawler is to categorize the crawled pages in to different topic based categories. To begin, the crawler requires a topic classification such as Yahoo. User can also provide their interested search keywords or URLs. Examples provided by the users get categorized in to different categories of classification. The crawler uses the Bayesian classifier to set the probability of a page that the page will belong to which category in the classification. Then the crawling process is similar to that of Naïve Best-First Crawling, it picks up the page with a highest match with the user query and starts the crawling for unvisited URLs found from that page and inserts in to the vector for further classification.

4.4 Context Focused Crawler

Sometimes if we are looking for specific topic i.e. "Computer Architecture", that word or topic may not be on the home page of computer science website. To reach at the topic we need to go first to the home page of computer science website then move to faculty pages which may lead to specific topic. To estimate the relevance of the link distance between a crawled page and the specific page the Context Focused Crawler is used, unlike focused crawler, the Context Focused Crawler is so much advance.

Such pages have given low priority in naive best-first crawler and may never crawl it again, this crawler can estimate that the relevant page of "Computer Architecture" is two link away from the "About Computer" page then the home page of computer science website. And the highest priority can be given to About Computer by using context graph of layers corresponding to seed page.

4.5 InfoSpiders

In InfoSpider [21, 23] algorithm, an adaptive population of agents perform the search for pages relevant to the topic given by the search query. By using an adaptive query list and a neural network each agent follows the crawling loop to decide which links to crawl next. The algorithm provides an exclusive frontier for each agent.

While each thread has its own frontier to fetch the pages, the crawler was limited to following the links on the current page and it was outperformed by the naïve best-first crawler on a number of evaluation standards. As taking inspiration from the naïve best-first algorithm, many improvements have been made in the InfoSpider. And the redesigned version has been found to outperform on crawling task that are longer than ten thousand pages [23].

An agent consists of a list of search keywords and a neural network to evaluate new links. The occurrence of keyword is weighted while traversing each link based on the nearness of the link from the given keyword. After fetching new page, the agent receives "energy" in proportion to the similarity between its keyword and the new page.A back-propagation algorithm is used to learn prediction of similarity estimation.

In this section we have presented a variety of crawling algorithms, most of which are variations of the best-first scheme. The readers may pursue Menczer et. al. [23] for further details on the algorithmic issues related with some of the crawlers.

5. A WEB CRAWLING SYSTEM

In order to set my work in this field in context as per review of above search strategies and algorithms, listed below are definitions of services that should be considered the minimum requirements for any large-scale crawling system.

Flexibility: as we know the structure of a website is changing and improving frequently, Any web crawling algorithm must be able to fit in desired scenarios with a minor or no change.

High Performance: The implementation of the system should be cost effective. It should be able to work on a minimum hardware availability. The system needs to be scalable with a minimum of 1000/ second to millions of pages so the quality and assurance are crucial for maintaining the high performance.

Error Acceptance: During web crawling process the system interacts with various aspects. As the system crawls from one server to another some protocol problems may occur. There may be the invalid or unstructured HTML code on the crawling page or some critical emerge, the system should be able to accept or ignore such problems and keep crawling process continue. As crawling process may take several days or week the chances of system or process failure, the crawling system should be able to restart or keeping data loss to a less.

Maintainability and Configurability: The web crawling system should have a proper configuration interface so that the crawling process can be monitored. the interface should contain statistics like download speed, no of pages crawled, no of running crawling instances, size of data stored. Monitoring interface can adjust the speed and on off instance of a crawler, add or delete system nodes and supply the black list of domains not to be visited, etc.

In a large distributed system like the Web, users find resources by following hypertext links from one document to another. When the system is small and its resources share the same fundamental purpose, users can find resources of interest with relative ease. However, with the Web now encompassing millions of sites with many different purposes, navigation is difficult.

The typical design of search engines is a "cascade", in which a Web crawler creates a collection which is indexed and searched. Most of the designs of search engines consider the Web crawler as just a first stage in Web search, with little feedback from the ranking algorithms to the crawling process. This is a cascade model, in which operations are executed in strict order: first crawling, then indexing, and then searching. The researcher's approach is to provide the crawler with access to all the information about the collection to guide the crawling process effectively. This can be taken one step further, as there are tools available for dealing with all the possible interactions between the modules of a search engine, as shown in Figure 1.

Fig.1 : Cyclic architecture for search engines, showing how different components can use the information

Generated by the other components. The typical cascade model is depicted with thick arrows.

6. EVALUATION OF CRAWLERS

As we discussed different crawling algorithms, there are different parameters to evaluate each crawler. Because the distinctiveness and necessitate of each crawler is different. There may be the different inputs for different algorithms. In a general sense, a Web Crawler may be evaluated on its capability of retrieving qualitative pages.

The size of a Web is enormously increasing and retrieval of a good page from such huge repository is a problem for web crawler in real life situation. The coverage of large number of topics is important for any topical based crawler. While performing experiment users may examine the significance of crawled pages that where the crawl was successful or not. The success of crawl is depended on the number of crawl process. To obtain a good result the crawl area should be large enough to analyze the success ratio.

7. CONCLUSION

Due to dynamic change of Web and its information structure, the algorithms of web crawling are changing to provide more accurate search results. Now a days Topical Crawlers are becoming more useful tools for topic based or focused information search areas of Web. The usage of hypertext structure of the web and link hierarchy is the challenge for future web crawlers for focusing on the specific topic or subject on the Web. By separating the HTML page structure in to different parts like page title, metatags, body content, link tags, headings etc. crawler can find more relevant page for user query.

To prevent the illegal access of web page and restricting a portion of entire web application form web crawler, www has introduces a concept called robot.txt file. It is not an official standard backed by a standards body, or owned by any commercial organization. It is not enforced by anybody, and there no guarantee that all current and future robots will use it. Consider it a common facility the majority of robot authors offer the WWW community to protect WWW server against unwanted accesses by their robots. The latest version of this document can be found on http://www.robotstxt.org/wc/robots.html.

WWW Robots (also called wanderers or spiders or crawlers) are programs that traverse many pages in the World Wide Web by recursively retrieving linked pages. For more information see the robots page. By improving the details of robot.txt file we can get more information related to particular website.

8. REFERENCES

[1] C. C. Aggarwal, F. Al-Garawi, and P. S. Yu. Intelligent crawling on the World Wide Web with arbitrary predicates. In WWW10, Hong Kong, May 2001.

[2] B. Amento, L. Terveen, and W. Hill. Does "authority" mean quality? Predicting expert quality ratings of web documents. In Proc. 23rd Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2000.

[3] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan. Searching the Web. ACM Transactions on Internet Technology, 1(1), 2001.

[4] K. Bharat and M.R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In

Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998.

[5] Sergey Brin and Lawrence Page. The anatomy of a large-scale hyper textual Web search engine. Computer Networks and ISDN Systems, 1998.

[6] S. Chakrabarti. Mining the Web. Morgan Kaufmann, 2003.

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

[8] S. Chakrabarti, K. Punera, and M. Subramanyam. Accelerated focused crawling through online relevance feedback. In WWW2002, Hawaii, May 2002.

[9] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: A new approach to topic-specific Web resource discovery. Computer Networks, 1999.

[10] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through URL ordering. Computer Networks, 1998.

[11] B.D. Davison. Topical locality in the web. In Proc. 23rd Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2000.

[12] P. M. E. De Bra and R. D. J. Post. Information retrieval in the World Wide Web: Making client-based searching feasible. In Proc. 1st International World Wide Web Conference, 1994.

[13] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused crawling using context graphs. In Proc. 26th International Conference on Very Large Databases (VLDB 2000).

[14] D. Eichmann. Ethical Web agents. In Second International World-Wide Web Conference, 1994.

[15] 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 WWW7, 1998.

[16] J. Johnson, T. Tsioutsiouliklis, and C.L. Giles. Evolving strategies for focused web crawling. In Proc. 12th Intl. Conf. on Machine Learning (ICML-2003),Washington DC, 2003.

[17] J. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM, 1999.

[18] V. Kumar, A. Grama, A. Gupta, and G. Karypis. Introduction to Parallel Computing: Design and Analysis of Algorithms. Benjamin/Cummings, 1994.

[19] H. Lieberman, F. Christopher, and L. Weitzman. Exploring the Web with Reconnaissance Agents. Communications of the ACM, August 2001.

[20] A.K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. Information Retrieval,2000.

[21] F. Menczer and R. K. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the Web. Machine Learning, 2000.

[22] F. Menczer, G. Pant, M. Ruiz, and P. Srinivasan. Evaluating topic-driven Web crawlers. In Proc. 24th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2001.

[23] F. Menczer, G. Pant, and P. Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. To appear in ACM Trans. on Internet Technologies, 2003.

[24] http://dollar.biz.uiowa.edu/~fil/Papers/TOIT.pdf.

[25] G. Pant. Deriving Link-context from HTML Tag Tree. In 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003.

[26] rajagopalan. Automatic resource list compilation by analyzing hyperlink structure.

[27] M. Porter. An algorithm for suffix stripping. Program, 1980.

[28] G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[29] Steven S. Skiena, The Algorithm design Manual.

[30] Ben Coppin, Artificial Intelligence Illuminated.

[31] **[Berners-Lee 1992]:** Berners-Lee, T., Cailliau, R., Groff, J.F. and Pollermann, B. **World-Wide Web:** the information universe. Electronic Networking: Research, Applications and Policy.

[32] **[Bush 1945]:** Bush, V. As We May Think. Atlantic Monthly, 1945.

[33] **[Coombs 1990]:** Coombs, J.H., Hypertext, Full Text, and Automatic Linking. In SIGIR, (Brussels, 1990).

[34] **[DeRose 1999]:** DeRose, S.J. and van Dam, A. Document structure and markup in the FRESS hypertext system. Markup Languages: Theory & Practice.

[35] **[Frisse 1988]:** Frisse, M.E. searching for information in a hypertext medical handbook. Communications of the ACM.

[36] **[Nelson 1981]:** Nelson, T. Literary Machines. Mindful Press, Sausalito, 1981.

[37] **[Nelson 1988]:** Nelson, T.H. Unifying tomorrow's hypermedia. In Online Information. 12th International Online Information Meeting Learned Info, Oxford, UK, 1988.

[38] **[van Dam 1969]** van Dam, A., Carmody, S., Gross, T., Nelson, T., and Rice, D., A Hypertext Editing System for the 360. In Conference in Computer Graphics, (1969), University of Illinois.

[39] **[Van Dam 1988]** van Dam, A. Hypertext '87 Keynote Address. Communications of the ACM.

[40] **Crawling the Web:** Gautam Pant, Padmini Srinivasan, and Filippo Menczer, Department of Management Sciences, School of Library and Information Science

[41] The University of Iowa, Iowa City IA 52242, USA.

[42] **WebCrawler:** Finding What People Want: Brian Pinkerton[2000]

[43] **Effective Web Crawling:** Carlos Castillo [2004]

[44] www.wikipedia.com