International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-2 Issue-6, May 2013

Study of WEBCRAWLING Polices

Anish Gupta, K. B. Singh, R. K. Singh

***Abstract*—Web crawler is a software program that browses WWW in an automated or orderly fashion, and the process is known as web crawling. A web crawler creates the copy of the visited pages so that when required later on, it will index the pages and processing becomes faster. This paper discuss the various techniques of the web crawling through which search becomes faster. In this paper studied has been done on the various issues important for designing high performance system. The performances and outcomes are determined by the given factors under the summarization criteria.**

***Index terms*—Web crawler, WWW - World Wide Web, URL - Universal resource locator, OPIC (On-line page importance computation), MIME – (Multipurpose Internet mail extension).**

I. INTRODUCTION

A Web crawler is a software program that browses the World Wide Web in a methodological fashion so that search becomes faster. This process is known as Web Crawling or Spidering. Web Crawlers are mainly used to create the copy of the visited pages so that when required it can be index and searching becomes faster. A Web crawler is types of web robots or bots i.e. an application that runs automated task over the internet. As the crawler visits the URLs, it identifies the entire hyperlink and adds them to the list of the URLs to visit, called the crawl frontier. These URLs are visited recursively from the frontier according to the policies. The number of possible URLs being generated by the server side software also made difficult for the web crawlers to avoid the replication of duplicate contents. To access any URLs their exists various possibilities, which depends upon the type of file extensions. The more the possibilities the more will be the combinations. The mathematical combinations creates the problem for the crawlers to search the unique contents as their exists endless combinations of relatively minor changes in the script. The behavior of the Web crawler depends upon the outcome of the combinations of certain policies.

- A *selection policy* that states which page is to be selected,
- A *re-visit policy* that states when to check for changes to the pages,
- A *politeness policy* that states avoiding of overloading Web sites, and
- A *parallelization policy* that states coordination of distributed Web crawlers.

The policies were discussed in detail along with merits and demerits in following sub sections.

Manuscript published on 30 May 2013.
*****Correspondence Author(s)
Anish Gupta, pursuing Ph.D from B.R. Ambedkar University Bihar, Muzzafarpur, (Bihar), India.
Dr. K. B. Singh, Associated with Institute of Physics(IOP), London, Indian Science Congress Association(ISCA), Kolkata, Indian Society of Atomic & Molecular Physic (ISAMP), India
Dr. Ram Kishore Singh, Associate Professor & Head of the Department of EC and IT in M. I. T. Muzaffarpur (Bihar), India.

A. Selection Policy:

In the year 1999 it has been identified that search engine index only utilized 19-20% indexed web [2], which go up to 40-70% by large scale search engine by 2005 [1]. A metric of importance for prioritizing web pages is required as the crawler download just of fraction, it important to have more relevant web pages rather than sample of web pages. The importance of pages is his quality, popularity in terms of links, and even the URL.

As the complete set of web pages cannot be identifies during crawling therefore designing a good selection policy is always a difficult as it has to work on partial information.

Pagerank strategy is useful if the crawler want to download the web page according to the page rank. The ordering metrics tested were breadth- first, backlink count and partial page rank.

Crawling strategy designed by Abiteboul , called OPIC (On-line page importance computation) is better than page computation, as each page is given initial sum of "cash" that is distributed among pages it points to. It downloaded the pages in the crawling frontier with higher amount of "cash". Later on experiments were carried out on pagerank strategies using depth first and breadth first, but depth first is found better that the others.

a. Focused crawling:

Focused crawling or topical crawling refers to the attempt of web crawlers to download pages that are similar to others. The problem encountered in focused crawling is to predict the similarity of the text of a given pages to the query before actually downloading the page. The performance of the focused crawling depends upon the number of links available in the specific topic to be searched [3][4][5][6].

i. *Restricting followed links:* Crawler may want to seek HTML pages and avoid other MIME types. Crawler may examine the URL and only request the URL ends with certain characters such as .html, .htm, .asp, .aspx, .php, etc. This strategy may skip various HTML web resources.

ii. *URL normalization:* URL normalization or URL canonicalization is the process of modifying and standardizing a URL in a consistent manner. This approach is generally being used by the crawlers to avoid the crawling more than once [7].

iii. *Path Ascending crawling:* Some crawlers want to access as many information from a particular web site. So this methodology was introduced to ascend to every path in each URL that is intent to crawl. Path ascending crawlers are also known as web harvesting [8].

iv. *Academic Focused Crawler***:** Is an example of focused crawlers, which crawls academic related documents [9][10].

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

b. Re-visit Policy:

Web has a dynamic nature, crawling a web can take a long time. By the time creation, updating and deletions may take place [11] [12]. From the view of search engine, there is a cost associated with not detecting an event. Mostly used cost functions are age and freshness.

i. *Age*: The age of a page p in the repository, at time t is defined as:

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

ii. *Freshness:* It indicates whether the local copy is accurate or not. The page p in the repository at time t is defined as:

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

Crawlers should have kept the average freshness of the page in the collection as high as possible.

c. Politeness Policy:

If the single crawler generates multiple requests in a fraction of seconds, than the server efficiency is reduced if the same request is generated from the multiple crawlers. A partial solution of the problem is robot exclusion protocol. It's a protocol standard for administrator to indicate which part of the web server a crawler should not access [13][14][15][16].

d. Parallelization Policy:

Parallel crawler is one which crawls multiple processes in parallel. The target is to maximize the download rate and to avoid repetition.

B. Architecture

Architecture should be optimized, along with good crawling strategies. It's easy to build slow crawler that download less number of pages, building high performance system is a challenge in system design, I/O, robustness and manageability [17]. Web crawler algorithm and design is being kept secret, but for high performance system the architecture is given in figure 1.

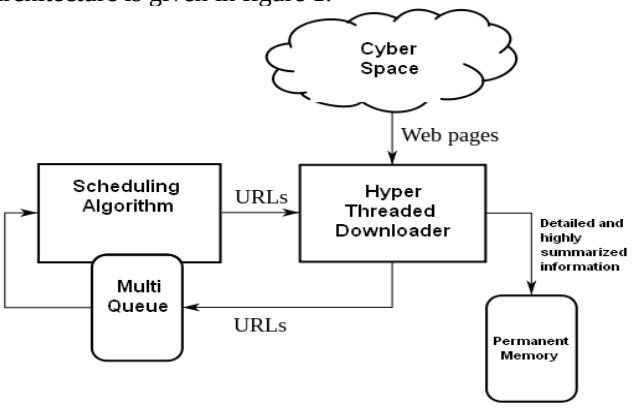

Figure 1: Architecture of Web Crawler

II. CONCLUSION OF STUDY

Various policies are being described to design high performance system. We can identify that crawler generates the request for the web pages, these request form the queue which is being processed by using various scheduling algorithm. Each policy has a certain issue which is too kept in mind by the crawler in order to design the high performance system.

III. CONCLUSION

All the policies are to be implemented in parallel or collectively in order to design high performance system. We should use the scheduling algorithm in rotation according to the requirement or need of the crawler so that efficiency can be improved. Permanent memory should more be focused on path ascending crawling in order to achieve freshness and age of the webpage.

REFERENCES

1. Gulli, A.; Signorini, A. (2005). "The indexable web is more than 11.5 billion pages". *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM Press.. pp. 902–903. doi:10.1145/1062745.1062789.
2. Lawrence, Steve; C. Lee Giles (1999-07-08). "Accessibility of information on the web". *Nature* **400** (6740): 107. Bibcode 1999Natur.400..107L. doi:10.1038/21987. PMID 10428673.
3. Menczer, F. (1997). ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery. In D. Fisher, ed., Machine Learning: Proceedings of the 14th International Conference (ICML97). Morgan Kaufmann
4. Menczer, F. and Belew, R.K. (1998). Adaptive Information Agents in Distributed Textual Environments. In K. Sycara and M. Wooldridge (eds.) Proc. 2nd Intl. Conf. on Autonomous Agents (Agents '98). ACM Press
5. Chakrabarti, S., van den Berg, M., and Dom, B. (1999). Focused crawling: a new approach to topic-specific web resource discovery. Computer Networks, 31(11–16):1623–1640.
6. Pinkerton, B. (1994). Finding what people want: Experiences with the WebCrawler. In Proceedings of the First World Wide Web Conference, Geneva, Switzerland.
7. 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. ISBN 978-3-540-40676-1. Retrieved 2009-03-22.
8. Cothey, Viv (2004). "Web-crawling reliability". *Journal of the American Society for Information Science and Technology* **55** (14): 1228–1238. doi:10.1002/asi.20078.
9. Jian Wu, Pradeep Teregowda, Madian Khabsa, Stephen Carman, Douglas Jordan, Jose San Pedro Wandelmer, Xin Lu, Prasenjit Mitra, C. Lee Giles, Web crawler middleware for search engine digital libraries: a case study for citeseerX, In proceedings of the twelfth international workshop on Web information and data management Pages 57-64, Maui Hawaii, USA, November 2012.
10. Jian Wu, Pradeep Teregowda, Juan Pablo Fernández Ramírez, Prasenjit Mitra, Shuyi Zheng, C. Lee Giles , The evolution of a crawling strategy for an academic document search engine: whitelists and blacklists, In proceedings of the 3rd Annual ACM Web Science Conference Pages 340-343, Evanston, IL, USA, June 2012.
11. 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: ACM. pp. 117–128. doi:10.1145/342009.335391. ISBN 1-58113-217-4. Retrieved 2009-03-23.
12. Jr, E. G. Coffman; Zhen Liu, Richard R. Weber (1998). "Optimal robot scheduling for Web search engines". *Journal of Scheduling* **1** (1): 15–29. doi:10.1002/(SICI)1099-1425(199806)1:1<15::AID-JOS3>3.0.CO;2-K.
13. Cho, J. and Garcia-Molina, H. (2003). Effective page refresh policies for web crawlers. ACM Transactions on Database Systems, 28(4).
14. Koster, M. (1995). Robots in the web: threat or treat? ConneXions, 9(4).
15. Heydon, Allan; Najork, Marc (1999-06-26) (PDF). *Mercator: A Scalable, Extensible Web Crawler*. Retrieved 2009-03-22.[dead link]

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

16. Dill, S., Kumar, R., Mccurley, K. S., Rajagopalan, S., Sivakumar, D., and Tomkins, A. (2002). Self-similarity in the web. ACM Trans. Inter. Tech., 2(3):205–223.
17. Shkapenyuk, V. and Suel, T. (2002). Design and implementation of a high performance distributed web crawler. In Proceedings of the 18th International Conference on Data Engineering (ICDE), pages 357-368, San Jose, California. IEEE CS Press.

AUTHOR PROFILE

Anish Gupta, is B.E(Hons) and M.Tech, he is pursuing Ph.D from B.R. Ambedkar University Bihar, Muzzafarpur, Bihar. Has got 12 years of teaching experience, he has written two books on "Data Structure Using C" and "Human Computer Interaction". He has published four international research papers and three national. He is member of IEEE.

Dr. K. B. Singh, born in 1976, passed B. Sc. Physics (Hons.), M. Sc. Physics with Electronics as special paper, Ph. D. Physics, PGDCA, D. Sc. Physics (Pursuing),being placed in first class with distinction and Gold medalist from B. R. A. Bihar University and indulged in teaching and research. At present he is serving G. P. Darbhanga as a Assistant Professor. He has attended several national and international conferences. He is associated with Institute of Physics(IOP), London, Indian Science Congress Association(ISCA), Kolkata, Indian Society of Atomic & Molecular Physic (ISAMP), India, Optical Society of Ammerica (OSA), Delhi Chapter, Bihar Mathematical Society (BMS), Bihar, Indian Society for technical Education (ISTE), ISMAMS, LASSI, National Academy of Science(NAS), India as life member. He has published several research paper, article, and review in journals of national and international repute. He is chief editor of Journal of Physical Sciences.

Dr. Ram Kishore Singh, born in 1952, passed B. Sc. Engg., M. Sc. Engg., Ph. D. in Electrical Engg from B. R. A. Bihar University and indulged in teaching and research. At present he is serving as a Associate Professor and head of the Deptt. Of EC and IT in M. I. T. Muzaffarpur. He has attended several national and international conferences. He is associated with Institute of Engineers, Kolkata,India as life member. He has published several research paper, article, and review in journals of national and international repute.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)