Smart Approach to Reduce the Web Crawling Traffic of Existing System using HTML based Update File at Web Server

Shekhar Mishra
Department of Computer Science and Engineering Radharaman Institute of Technology & Science, Bhopal, Madhya Pradesh, India.

Anurag Jain
Department of Computer Science and Engineering Radharaman Institute of Technology & Science, Bhopal, Madhya Pradesh, India.

Dr. A.K. Sachan
Department of Computer Science and Engineering Radharaman Institute of Technology & Science, Bhopal, Madhya Pradesh, India.

ABSTRACT

Web crawler is used for downloading information from web. Web pages are changed without any notice. Web crawler frequently revisits websites to check updates. It is expected that 40% of present internet traffic is because of web crawling. In this paper we propose a file which maintains the list of updated URLs of web pages of web site. Format of file is based on HTML. Crawler will only visit the UPDATE File, and need not have to revisit the full website to know the updates. This scheme can easily implement on today's system with little modification on web application and web crawler. In simulator we test proposed method; using a website of 13 pages for experiment. Experiment results shows that this scheme is very promising.

General Terms

Approach to reduce Web Crawling Traffic.

Keywords

Web Search Engine, Web, Web Crawler. Web Crawling Traffic.

I. INTRODUCTION

Main component of search engine is Web Crawler. Web crawler is automatic program that browses the web & download information [1]. Search engine uses this downloaded web pages to store in repository, this repository is used to generate the results of search. Web crawler is also used in many other services like search engines, digital library online marketing, web data mining & search for personal information such as emails, numbers, address for marketing and spam mails.

Size of web is increasing at very high rate. Google announced that Google have revealed one trillion unique URL in May 2009.Over 109.5 million Websites operating [2].

Yuan and Harms in 2002 review the log file of web server at department of computer Science at University of Alberta and find that maximum 40.6% total web traffic is due to hits by web crawler [13]. Crawlers are not actual user so the heavy crawling traffic is not good for websites and network. It is most horrible for websites new in business.

Bal and Nath in 2010 perform experiment on web. They download home pages of 100 different web sites daily for 30 days. They find 52% pages change every day. 48% of web pages don't change daily [6].

Web managers can direct web crawler by using Robots Exclusion Protocol. It is a convention to avoid Web Crawler from accessing all or part of a Website which is openly viewable. Protocol use "robot.txt" file. Web crawler fined this file at server and follows the directions in robot.txt. Robot.txt file is maintained by Web manager [3, 4].

In this paper we propose, using a list of URLs of updated pages in UPDATE File. Therefore web crawler will only check this UPDATE file for updates. Format of file is HTML based file. This method will reduce the traffic of Web Crawling.

In this paper, Section II is regarding the related work done for reducing the crawling traffic. Section III shows the proposed approach with figure. Section IV is describing the simulator use for trial. Experiment results and tables are at Section V. Section VI mathematical formula to compare our idea with old approach. Section VII shows the calculation with graph. Conclusion and Reference are at last.

II. RELATED WORK

To reduce the crawling traffic research is going in various different areas. In our survey we identify those areas such as network level, crawler level, web server and web crawler coordination.

In crawler level research work, priorities for different types of web pages are given according to their altering rate. Performance of crawling is based on following periodic page changing priorities is incremental crawling [9, 10].

One very interesting approach is to setup Active Routers at the strategic key position in network. These active routers capture the content of web pages from underling traffic and keep them for indexing [13].

Other approach is to place position crawler at different geographical positions. These crawlers execute their work locally in there geographical area [7].

One proposed solution is to shrink the crawling traffic by coordination between crawler and web server. Special query language and web services are proposed. Crawler sends the query to Web server and web server will response back to give details about web pages updates and removals [8].

In Mobile crawling idea, place the mobile crawler at the web server. That small web crawler resides on web server, perform their operations on web server and send updates in return [6].

Researches show that some crawlers do not obey the moral accepted behavior. Researchers have developed a method to calculate ethicality of web crawler [11].

Unethical crawlers are causing serious problem for website. One such trouble is denial of services to real users. It adds extra expenditure to run the website. Crawler copies copyright property and private information of user [12].

III. APPROACH USED

We propose the use of new file to inform updates to web crawler. This new file contains URLs of updated pages. File is place at the root of website. We name the file UPDATE file. When pages in website are update, web manager puts the URLs of updated pages on UPDATE file. Crawler will only visit the UPDATE file and updated pages for updates, instead of visiting the full website.

UPDATE FILE: It is normal file accepted by all parties. We believe that all web crawlers understand HTML and all Web Managers can maintain the HTML based file. So proposed format of file is HTML based file. It can be HTML, JSP, ASP, etc. File containing the URLs in order to their time of updates. New updates are at the top and old updates are at the bottom. File is maintained by web manager (Website owner).

Algorithm for Crawling

1. Crawler visits all web pages of website for first time (Following Robot.txt)
2. For updates crawler only visit UPDATE file.(Figure 1)
3. Crawler checks updates of UPDATE file with its own last visit.
4. If updates in file are new for crawler, crawler visits the updated pages and download pages for indexing.

Algorithm for Managing update file at Website

1. Web manager update the web pages in the website
2. Web manager put the valid URLs of updated pages on update file, new updates on top

Benefits:

This system can work on existing system with little modification. No extra skills, software or hardware needed for both parties.

Incentive for individual web sites

a) This system reduces the crawling traffic of web Server and network. It is more effective for sites having large number of pages. For example if website has 1000 pages and website update are of two pages only, web crawler will visit all 1000 pages to discover the updated pages. In our approach crawler will only visit the UPDATE file and visit only those two updated pages.

b) If there are no new updates, then crawler will only visit the UPDATE file and will leave the website.

c) This approach gives the power to Web Manager (website owner) to tell the crawler what is new in website.

Incentive for individual web Crawler

a) Web Crawler will work fast and search engine will give more updated results, as crawler is only allowed for checking the updated files. Instead of checking all the files and pages of the Website.

b) The crawler visits and find the updates in constant number of page (Visiting update file + number of updates).Method is independent from the number of pages in the website. In normal method crawler visit every page in website to know the updates.

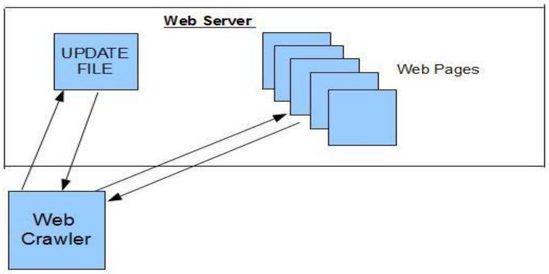


Figure 1 Illustration crawling using UPDATE file.

Limitation

a) This system will only work when both web crawler and website agree to follow rules.

b) Website manager should update "UPDATE file" as it update web page.

c) Web crawler should only check the UPDATE file for updates.

Difference between proposed scheme with other protocols

Sitemapes protocol: Main objective of protocol is to tell search engine about hyperlinks map of website. Protocol tells about some Meta data of web pages to web crawler. Protocol is based on XML. Meta data Field includes "lastmod" field. This field represent day and time of modification of web page. "Changefreq" field tell web crawler about changing frequency of web page [5].

This protocol can be used to tell the web crawler about the updates on web site. But both useful fields "lastmod" and "changefreq" are optional. Protocol is based on XML and not every web Crawler understands XML.

In our proposed approach we use HTML based file. We believe that every web crawler understand HTML. Main objective of this approach is to tell the web crawler about Updates.

Robots Exclusion Protocol: protocol is use to direct crawlers. Protocol doesn't provide Updates to crawler.

IV. SIMULATION

For experiment we use dummy website on our local computer. Site is deployed on normal web J2EE server. There are three levels in our experimental website. Each page is pointing to three child pages by hyperlink. General structure of our experimental website is shown in figure 2. Experimental website is having Total 13 pages. We use Crawler to visit website. Crawler is coded in java. Single thread crawler is use for experiments. We record the time in milliseconds in normal crawling for each page. Then the same crawler is directed to the UPDATE file. All pages are JSP and almost having same size. Format of the UPDATE file is JSP.

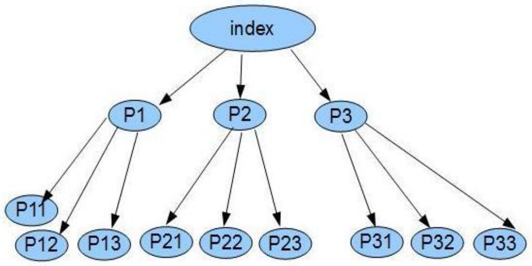


Figure 2 structure of website used for experiment

V. EXPERIMENT

First we perform the normal crawling on our website. The results obtained shown in Table1.

Table 1 (Normal Visiting of Crawling in website)

Index	URL (Crawler site Visiting order)	Starting Time of Crawler (in millisecond)	Time after page Download (in millisecond)	Total time to download(in millisecond)
1	http://localhost:9254/CrawWeb/Index.jsp	1287486308224	1287486308396	172

2	http://localhost:9254/CrawWeb/p3.jsp	1287486308224	1287486308692	468
3	http://localhost:9254/CrawWeb/p2.jsp	1287486308224	1287486308942	718
4	http://localhost:9254/CrawWeb/P1.jsp	1287486308224	1287486309223	999
5	http://localhost:9254/CrawWeb/p33.jsp	1287486308224	1287486309457	1233
6	http://localhost:9254/CrawWeb/p31.jsp	1287486308224	1287486309722	1498
7	http://localhost:9254/CrawWeb/p32.jsp	1287486308224	1287486310096	1872
8	http://localhost:9254/CrawWeb/p21.jsp	1287486308224	1287486310315	2091
9	http://localhost:9254/CrawWeb/p22.jsp	1287486308224	1287486310564	2340
10	http://localhost:9254/CrawWeb/p23.jsp	1287486308224	1287486310814	2590
11	http://localhost:9254/CrawWeb/p13.jsp	1287486308224	1287486311048	2824
12	http://localhost:9254/CrawWeb/p11.jsp	1287486308224	1287486311235	3011
13	http://localhost:9254/CrawWeb/p12.jsp	1287486308224	1287486311454	3230

Second we perform experiment using our proposed approach. We change the content of the pages(s) in experimental website. We put the URL of updated page(s) in UPDATE file. Then we direct crawler to UPDATE file. We perform six experiments. First 3 experiments we use update single page at different level. For last three experiments we update 2 and 3 pages.

Results we obtained in experiment are in table 2.

Column L = Contain experiment number.
Column M = Contain the updated page(s).
Column N = Contain URL of pages visited by crawler
Column O = Contain the start time of Crawler (Millisecond)
Column P = Contain the time to reach that page. (Millisecond)
Column Q = Time spend to visit particular page (Millisecond)

Table 2

L	M	N	O	P	Q
1	index	http://localhost:9254/CrawWeb/update.jsp	1287490380435	1287490380669	234
		http://localhost:9254/CrawWeb/index.jsp		1287490380747	312
2	P1	http://localhost:9254/CrawWeb/update.jsp	1287490577650	1287490577837	187
		http://localhost:9254/CrawWeb/P1.jsp		1287490577915	265
3	P23	http://localhost:9254/CrawWeb/update.jsp	1287490730645	1287490730817	172
		http://localhost:9254/CrawWeb/p23.jsp		1287490730895	250
4	P11 and P23	http://localhost:9254/CrawWeb/update.jsp	1287477109254	1287477109426	172
		http://localhost:9254/CrawWeb/p11.jsp		1287477109722	468
		http://localhost:9254/CrawWeb/p23.jsp		1287477109987	733
5	P11, P22 and P33	http://localhost:9254/CrawWeb/update.jsp	1287477421879	1287477422237	358
		http://localhost:9254/CrawWeb/p33.jsp		1287477422549	670
		http://localhost:9254/CrawWeb/p22.jsp		1287477422783	904
		http://localhost:9254/CrawWeb/p11.jsp		1287477422846	967
6	P1, P2 and P31	http://localhost:9254/CrawWeb/update.jsp	1287478638170	1287478638357	187
		http://localhost:9254/CrawWeb/p31.jsp		1287478638435	265
		http://localhost:9254/CrawWeb/p2.jsp		11287478638608	438
		http://localhost:9254/CrawWeb/P1.jsp		1287478638800	630

VI. DATA ANALYSIS

In our experiment we observe in normal crawling the time to visit particular updated web page depending on order of crawling and the total number of pages visited by crawler to reach that particular page. We use two modified formulas to show the comparison between old and proposed approach.

Time Calculation

Let (Ti) is a time to download the i th page in website.

1. Time needed to visit the updated page (i) in website is X = T1+ T2+ T3......... Ti; T1, T2... Are pages before the Ti.
2. Time needed to know the total number of updated pages, crawler have to visit every page in Website. Its total time needed to visit the complete website. Let website have total (N) pages. Then time needed to know the number of updated pages in site is Y = T1 + T2....... TN

Using proposed approach crawler only visit the web UPDATE file and visit only those links that are new for crawler.

1. Time need to visit particular page is X1 = time to visit UPDATE file + time to visit pages previous to particular page using update file + time needed to visit that particular page.
2. Time needed to know the number of updated pages is Y1= (time needed to download the UPDATE file) + (time need to download the particular updated pages.)

Speed up ratio = S = (pages visit with old method)/ (pages visit with new method),

If value of S is greater than 1, system is S times faster than old approach. If value is less than 1, system is slower then old approach.

Page Wise calculation

In normal Crawling

1. To know the total number of Updates in Website crawler visits every pages in website. Z = total number of pages is site.

2. To reach the update page crawler visits U = number of pervious pages visit + Updated page.

VII. CALCULATION

Following table shows results. Table 3 shows speed up ratio base on time.

Table 3

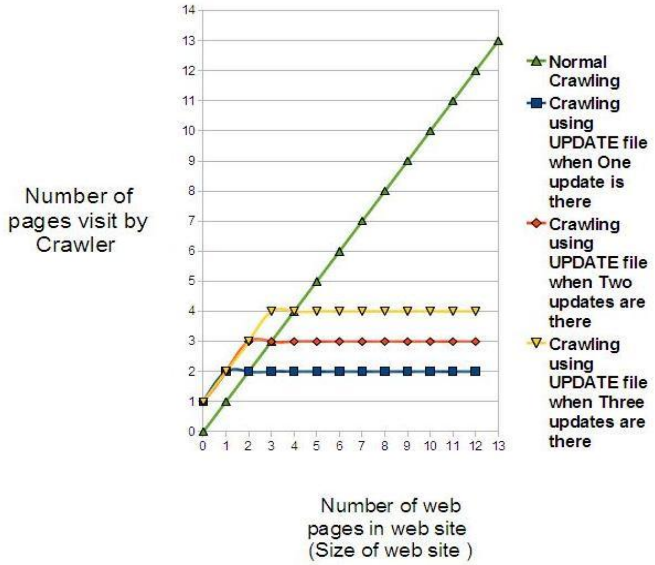
Experiments'	Updated Page	Time to visit updated page in normal crawling (in milliseconds)	Time to visit updated page using UPDATE file	Time to visit full website to know total number of updates in normal crawling	Speed up ratio to visit the update page	Speed up ratio to know the total number of update in website
1	Index	172	312	3230	.5512	10.352
2	P1	999	265	3230	3.7698	12.18867
3	P23	2590	250	3230	10.36	12.92
4	P11 & P23	3011	733	3230	4.1077	4.4065
5	P11, P22 & P33	3011	967	3230	3.1137	3.3402
6	P1,P2 & P31	1498	630	3230	2.3777	5.1269

Following table 4 shows speed up ratio base on number of page visit

Table 4

Experiments'	Updated page	Number of Pages visit to reach updated page. In normal crawling	Number of page visit using UPDATE file	Speed up ratio to visit the update page	Speed up ratio to know the total number of update in website
1	Index	1	2	1/2 =.5	13/2 = 6.5
2	P1	4	2	4/2 = 2	13/2 = 6.5
3	P23	10	2	5	13/2 = 6.5
4	P11 & P23	10	3	10/3	13/3 = 4.333
5	P11, P22 & P33	12	4	3	13/4 = 3.25
6	P1,P2 & P31	6	4	1.5	13/4 = 3.25

In our calculation we find that this scheme is very promising. Experiments show our scheme download 6.5 times less pages than old scheme when there is one update. Idea is more effective for web sites have large number of pages. Graph 1 shows our results.



Graph1. Graph showing crawler visit updated pages in constant number of page visit

VIII. CONCLUSION

This approach gives the power to the website manager to tell the web crawler what is new. Web manager don't have to install new software on web server that put extra load on website. Scheme can be automatic using some software that can work under the control of web manager. There is possible use of this method for deep web mining. There are many unknown drawbacks and applications of this idea. Some useful Meta data of pages can also be provided to crawler with this sachem.

REFERENCES

[1] "Web crawler", From Wikipedia, http://en.wikipedia.org/wiki/Web_crawler

[2] "World Wide Web", From Wikipedia, http://en.wikipedia.org/wiki/World_Wide_Web

[3] "Robots Exclusion Protocol", http://www.robotstxt.org/robotstxt.html

[4] "Robots exclusion standard", Wikipedia http://en.wikipedia.org/wiki/Robots_exclusion_standard

[5] "Sitemaps", from Wikipedia, http://en.wikipedia.org/wiki/Sitemaps

[6] Bal.S and Nath.R,"Filtering the web pages that are not modified at remote site without downloading using mobile crawler". Information Technology journal 9(2)2010 ISSN 1812- 5638, Asian Network for Sciencetific information. (pp: 376-380)

[7] Cambazoglu, B.B.; Junqueira, F.; Plachouras, V.; Telloli, L., "On the feasibility of geographically distributed web crawling." (ISBN: 978-963-9799-28-8) In the proceedings of Third International ICST Conference on Scalable Information Systems, ICST, Vico Equense, Italy (2008)

[8] Chandramouli A and Gauch. S. "A Co-operative Web Services Paradigm for Supporting Crawlers", In the proceedings of Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications) - RIAO 2007, 8th International Conference, Carnegie Mellon University, Pittsburgh, PA, USA, May 30 - June 1, 2007.

[9] McCurley S. Kevin "Incremental Crawling" Google Research http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/en//research/pubs/archive/34403.pdf

[10] Sharma A.K, Dixit. A and Singhal N. "Design of a Priority Based Frequency Regulated Incremental Crawler" 2010 International Journal of Computer Applications (ISSN: 0975 – 8887) Volume 1 – No. 1. (pp: 42-47)

[11] Sun. Y, Councill G. Isaac and Giles C. Lee, "The Ethicality of Web Crawlers", in the proceedings of 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Toronto Canada august 2010. (pp: 668-675)

[12] Thelwall. M and Stuart. D, "Web crawling ethics revisited: Cost, privacy and denial of service". Journal of the American Society for Information Science and Technology. 2006. Volume 57, Issue 13 November 2006. (pp: 1771 - 1779)

[13] Yuan, X.M. and J. Harms, "An efficient scheme to remove crawler traffic from the internet." Proceedings of the 11th International Conference on Computer Communications and Networks, Oct 2002. 14-16, IEEE CS Press, (pp: 90-95).