International Journal of Computing and Business Research (IJCBR)

ISSN (Online) : 2229-6166

Volume 3 Issue 2 May 2012

INCREASING THE EFFICIENCY OF CRAWLER USING CUSTOMIZED SITEMAP

Dr. Bharat Bhushan
Head, Department of Computer Science
Guru Nanak Khalsa College,
Yamunanagar, Haryana

Meenakshi Gupta
Assistant Professor, Department of Computer Applications
Maharaja Agrasen Institute of Management and Technology,
Jagadhri, Haryana

Garima Gupta
Lecturer, Hindu Girls College,
Jagadhri, Haryana

Abstract: With the increasing dependency on Internet for searching information related to any field, the importance of web crawlers for a search engine is increasing. In order to make the relevant and timely information available, a web crawler has to visit not only the new websites but also has to maintain the freshness of already crawled web pages. This on the one hand is a time consuming process and on the other side further increases traffic on web server. In this paper we study how to increase the efficiency of a web crawler by not revisiting the web pages that have not been updated since its last visit. Here we make the use of sitemap of website maintained on web server with minor changes to it. This will help web crawler to find out web pages that have been updated or added since its last visit. We simulate the proposed approach on a website. Results show that this approach will be certainly helpful for improving the efficiency of web crawlers and saving the limited web resources.

Keywords: Web crawling, Search Engine, Sitemap, HTTP request, web pages.

1. INTRODUCTION

World Wide Web is a vast repository of information on almost every type of subject. In order to get the required information from so huge web, simply clicks in one's mind that make the use of a search engine. But one wonders, how the search engine can be able to retrieve the required information in just a fraction of seconds. For this search engine depends on web crawler.

Web crawlers are an important component of web search engines, where they are used to collect the corpus of web pages indexed by the search engine. Moreover, they are used in many other applications that process large numbers of web pages, such as data mining, comparison, shopping engines, and so on [7].

Web Crawler software doesn't actually move around to different computers on the Internet as viruses or intelligent agents do. It resides on a single machine. The crawler simply sends HTTP requests for documents to other machines on the Internet, just as a web browser does when the user clicks on links. All the crawler really does is to automate the process of following links [8, 10]. Details on crawling algorithms are kept as business secrets. When algorithms are published, there is often an important lack of details that prevents other from reproduce the work [1].

A number of web crawlers have been developed till date. The basic process that a web crawler follows is that firstly using some seed links it downloads the web pages. It makes the index on the basis of information extracted from these web pages such as title of web page, metadata defined in web page, contents of web page. From these web pages it further extracts the links and repeats the process. However it is very time consuming process and as the size of web is increasing very rapidly, so it becomes almost impossible for the web crawler to access whole of the web. Therefore, some technique is required so that web crawler can be able to crawl to the maximum possible amount of web if not the whole one.

When a visitor visits the site, he just follows the link available directly on the home page of the web site or the highlighted one. The visitor does not go many levels down in the links. So in order to guide the visitor, most of the sites these days provide sitemap. With the help of sitemap, users visiting the site may be able to know that what is where in the site. The sitemap provided

by the developer of website is basically for guiding the visitor of website but here we will make the use of this sitemap with slight modification for easing the work of web crawler. So that it takes less time during the crawling of the website and can do its work more efficiently. It will also reduce the traffic load on website. In 2010 Sun, et al analysis log file of many different web sites. They find average 50% of web request is generated by web crawler [5]. For implementing this technique the administrator of the website will have to provide some more information in the sitemap for the web crawlers.

2. RELATED WORK

Web crawlers are mainly used for the search engines. They have limited computer resources and limited time [3] and size of web is growing at a very fast rate. So it is not possible for crawlers to cover the complete web. They try to cover the most of the web. Further they have to frequently revisit the web pages in order to keep them updated. Various studies on this topic suggest that how web crawlers can do it in more efficient manner and save time.

One solution suggests the use of incremental crawler to improve the freshness of collection and to bring new pages more timely [4].

Other proposes query based approach to inform updates on website to web crawler using dynamic web page and HTTP GET request [5].

Another approach suggests that breadth-first search is a good crawling strategy, as it tends to discover high-quality pages early on in the crawl [2].

Brandman et. al. suggest that if the meta-data includes the size, last-modified date and path of each available page then it can substantially save crawler's time[9].

3. CHALLENGES FOR WEB CRAWLER

Nowadays it is difficult to build a crawler that can accurately traverse the entire web because of its volume, pace of change and growth [12]. In this paper, we take into consideration the following challenges for the web crawler.

Coverage – It must be able to cover maximum percentage of the whole web.

Freshness – It must be able to keep latest updated copies of web pages at any time.

Utilisation of Network Bandwidth – It must be able to make the maximum utilisation of available network bandwidth.

Reduced Overloading – It must not overload the website during the process of crawling to find out the new or updated web pages.

4. PROPOSED APPROACH

Here we make the use of sitemap of website for making the working of web crawler more efficient. We consider it as extended sitemap. It will have one more field named 'LAST_UPDATED'. Its data structure is shown below:

4.1 Data Structure of Extended Sitemap

The proposed sitemap will be having the fields as shown below:

ANCHOR_TEXT	URL	LAST_UPDATED

For Example:

Contact Us	http://www.xyz.com/contact.html	15/06/2011 9:52:10 AM

However the date and time will be stored in milliseconds in order to find out more accurately that whether web page has been updated or not since the last visit or it is newly added one.

Ex. 15/06/2011 9:52:10 AM is equal to 1331007730000 milliseconds

This sitemap will guide web crawlers about a web page that whether to download it or not. To download the sitemap, web crawler will send HTTP request to web server like other normal web surfers. HTTP is a protocol with the lightness and speed necessary for a distributed collaborative hypermedia information system [11]. For visiting a website first time, web crawler will download the whole web site using sitemap. The web crawler will maintain an index of all the visited pages along with date and time of last visit. When it will revisit the website, it will download only those web pages that have been updated or added since last visit. To find out such web pages, web crawler will use the field 'LAST_UPDATED' in sitemap.

4.2 Steps for working of Web Crawler

1. Web crawler sends the HTTP request for downloading sitemap of website with the parameter 'XTND' to get the extended sitemap.
2. It receives extended sitemap.
3. It extracts required URLs from this sitemap by examining the field LAST_UPDATED.
4. It maintains a table of all visited web pages along with 'LAST_VISITED' field.

4.3 Steps for working of Web Server

1. The web developer updates an existing web page and also updates the LAST_UPDATED field in extended sitemap on web server.
2. The web developer creates a new web page and accordingly inserts a new entry in extended sitemap on web server with the date of creation in LAST_UPDATED field.

4.4 Time-Space Complexity of Given Solution

In the given solution web sites will have to provide extended sitemap. This is the sitemap with one more field named LAST_UPDATED. So a little bit, more memory space will be required for storing the sitemap. Further the website administrator has to take care that every time there is change in any of the web page, the sitemap must be updated accordingly. So from the administrator point of view it will require some extra time. However it will help to reduce the traffic of web crawlers on the website to a great extent. In addition, it will save a lot of time of web crawler by not visiting the web pages that have not been updated since the last visit. This solution will also help web crawler to improve the freshness of crawled websites more timely.

So the memory space and time required to maintain the extended sitemap is very less as compared to the time saved of web crawler as well as web server. Moreover, due to reduced traffic on web server, users will get faster access to website. From the point of view of user, the time needed to access the website is very important. If it is taking too much time to browse the website then either user becomes frustrated or switches to some other website that can be accessed faster.

5. ADVANTAGES

1. The extended sitemap will help the web crawler to find out web pages that have been updated after last visit. There will be no need to download the whole website to find out the updated web pages.
2. This will help to save the time of web crawler as well as reduce the traffic on websites generated due to web crawlers.
3. The web crawler will be able to utilise the saved time for crawling new websites. So in this way it will be possible to cover more percentage of the web.
4. For the suggested solution of extended sitemap there is no need to generate any special dynamic response for different web crawlers by the web server.

6. LIMITATIONS

Presently most of the websites provide sitemap to the users for easier navigation of website. However there are some websites who do not have sitemap. In this case to implement the suggested solution either the administrators of these websites will have to create the site map for their websites or the web crawler has to follow the traditional time-consuming way to crawl these websites.

Further website developers will have to create extended sitemap and keep it updated otherwise it may be the case that they have placed some important information on a web page to convey it to the users and the web page is ignored by web crawler just because the sitemap was not updated accordingly. So on the one hand users will be deprived of required information timey and on the other hand the related organization may have to bear great loss or face some untoward situation due to this.

7. SIMULATION OF PROPOSED APPROACH

The proposed solution is simulated on dummy educational website. The home page of website contains seven links. Further some of these links contain child links. As is shown in Figure 1 below:

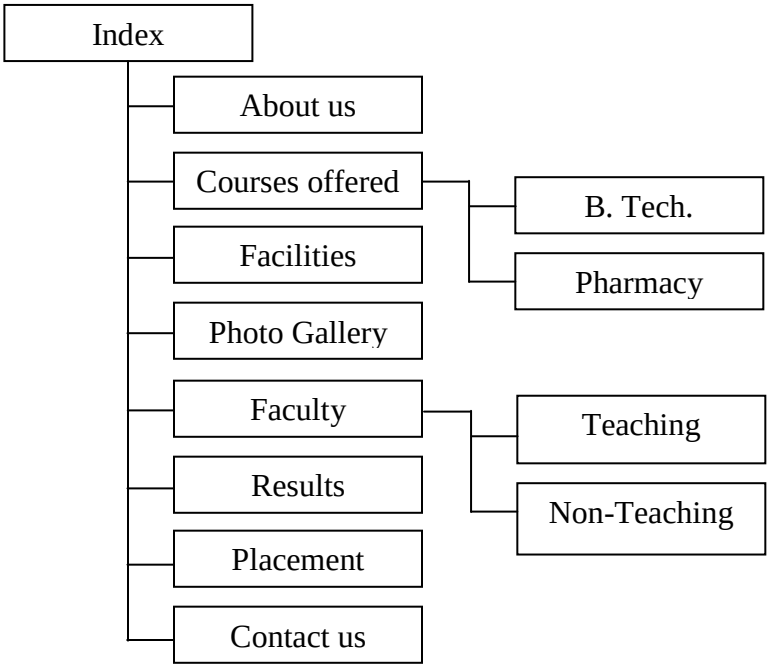

Figure 1: Structure of example website

When a web crawler will visit the website first time, it will download the complete website. Therefore, all the 13 web pages as shown in figure 1 will be downloaded. It will maintain a table of all the downloaded web pages as shown in Table 1. Here the field DOWNLOAD_TIME is taken just for comparison purpose between old approach and proposed approach used for downloading websites by web crawlers. Otherwise this field will not be maintained by the web crawler.

Table 1: Record of downloaded web pages maintained by web crawler

ANCHOR_TEXT	URL	DOWNLOAD_ TIME (in milliseconds)	LAST_VISITED (in milliseconds)
Home	http://www.xyz.com/index.html	538	1308111730000
About us	http://www.xyz.com/about.html	235	1308111730538
Courses Offered	http://www.xyz.com/courses.html	256	1308111730773
B. Tech.	http://www.xyz.com/courses/btech.html	327	1308111731029

Pharmacy	http://www.xyz.com/courses/pharmacy.html	423	1308111731356
Facilities	http://www.xyz.com/facilities.html	498	1308111731779
Photo Gallery	http://www.xyz.com/photo_gallery.html	890	1308111732277
Faculty	http://www.xyz.com/faculty.html	348	1308111733167
Teaching	http://www.xyz.com/faculty/teaching.html	459	1308111733515
Non Teaching	http://www.xyz.com/faculty/nonteaching.html	450	1308111733974
University Toppers	http://www.xyz.com/results.html	236	1308111734424
Placement	http://www.xyz.com/placement.html	365	1308111734660
Contact us	http://www.xyz.com/contact.html	287	1308111735025

Now when the web crawler schedules to revisit the website in order to maintain its freshness, it will firstly send the HTTP request with 'XTND' parameter to server for downloading the extended sitemap. In return web server will send the requested sitemap. Suppose at that time the sitemap consists of following entries as shown in Table 2.

Table 2: Extended sitemap maintained on web server

ANCHOR_TEXT	URL	LAST_UPDATED (in milliseconds)
Home	http://www.xyz.com/index.html	1272964276000
About us	http://www.xyz.com/about.html	1272964523000
Courses Offered	http://www.xyz.com/courses.html	1272964705000
B. Tech.	http://www.xyz.com/courses/btech.html	1272964720000
Pharmacy	http://www.xyz.com/courses/pharmacy.html	1272964736000
Facilities	http://www.xyz.com/facilities.html	1276682653000
Photo Gallery	http://www.xyz.com/photo_gallery.html	1276683322000
Faculty	http://www.xyz.com/faculty.html	1273576124000

Teaching	**http://www.xyz.com/faculty/teaching.html**	**1308473192000**
Non Teaching	http://www.xyz.com/faculty/nonteaching.html	1273576523000
University Toppers	**http://www.xyz.com/results.html**	**1308473495000**
Placement	**http://www.xyz.com/placement.html**	**1308473616000**
Our Alumnus	**http://www.xyz.com/alumni.html**	**1309334004000**
Contact us	http://www.xyz.com/contact.html	1272964754000

The updated entries in this table since the last visit of the web crawler are shown in bold letters just for reference purpose. By observing above sitemap we get that a new web page 'http://www.xyz.com/alumni.html' has been added to the website. Further the web pages named as 'teaching.html', 'results.html' and 'placement.html' have been updated since the last visit by the web crawler.

So the web crawler does not need to download the complete website. It will filter this sitemap to find out the web pages that have been updated or added since last visit. For this it will compare the LAST_VISITED field with LAST_UPDATED field for every web page and take in to consideration only those web pages where LAST_UPDATED field's value is more than LAST_VISITED field's value.

So in given example the web crawler will just download only four web pages as shown in Table 3. The DOWNLOAD_TIME required to visit these web pages is shown, as previously mentioned, just for comparison purpose.

Table 3: Web pages requiring re-download

ANCHOR_TEXT	URL	DOWNLOAD_ TIME (in milliseconds)	LAST_VISITED (in milliseconds)
Teaching	http://www.xyz.com/faculty/teaching.html	515	1309453215000
University Toppers	http://www.xyz.com/results.html	351	1309453215515
Placement	http://www.xyz.com/placement.html	360	1309453215866
Our Alumuns	http://www.xyz.com/alumni.html	673	1309453216226

So after revisiting the website, the table updated by the web crawler will be as shown in Table 4. Here we have not shown DOWNLOAD_TIME field as it is not actually maintained by web crawler.

Table 4: Updated record of downloaded web pages maintained by web crawler

ANCHOR_TEXT	URL	LAST_VISITED (in milliseconds)
Home	http://www.xyz.com/index.html	1308111730000
About us	http://www.xyz.com/about.html	1308111730538
Courses Offered	http://www.xyz.com/courses.html	1308111730773
B. Tech.	http://www.xyz.com/courses/btech.html	1308111731029
Pharmacy	http://www.xyz.com/courses/pharmacy.html	1308111731356
Facilities	http://www.xyz.com/facilities.html	1308111731779
Photo Gallery	http://www.xyz.com/photo_gallery.html	1308111732277
Faculty	http://www.xyz.com/faculty.html	1308111733167
Teaching	**http://www.xyz.com/faculty/teaching.html**	**1309453215000**
Non Teaching	http://www.xyz.com/faculty/nonteaching.html	1308111733974
University Toppers	**http://www.xyz.com/results.html**	**1309453215515**
Placement	**http://www.xyz.com/placement.html**	**1309453215866**
Contact us	http://www.xyz.com/contact.html	1308111735025
Our Alumnus	**http://www.xyz.com/alumni.html**	**1309453216226**

So with old approach,

Total time required to download the complete website = $\sum pt_i$ (where, i = 1 to n)

Where *n = total number of web pages in a website*

pt = time required to download web page

Therefore, total time required to re-download given dummy website =

538+235+256+327+423+498+890+348+**515**+450+**351**+**360**+287+**673** = 6151

Here the figures in bold are the download time of updated or newly added web pages since last visit of web crawler to the website.

Where as in the proposed solution there is no need for the web crawler to download the complete website in order to improve the freshness. Only the updated or newly added web pages since the last visit will be required to download.

So using proposed approach,

Total time required to revisit the website = $\sum qt_i$ (where, i = 1 to ua)

Where *ua = total number of web pages updated or newly added to the website*

qt = time required to download updated or newly added web page

So total time required to refresh given dummy website = 515+351+360+673 = 1899

Difference = Total time to download the complete website – Total time to download only the updated or newly added web pages i.e. 6151 – 1899 = 4252. The difference is more than 50% which means on an average less than 50% of web pages on a website change.

So when revisiting the website instead of downloading it completely in order to increase the freshness it is worth to take some time to download the extended sitemap and filter it to find out that which web pages have been updated or newly added since the last visit. This will increase the efficiency of web crawler as well reduce the traffic on website due to web crawler.

8. CONCLUSION

In this paper we focussed on reducing the increasing web traffic on web servers due to web crawlers. We proposed an approach that by making minor changes in sitemap of a website, we can increase the efficiency of a web crawler to a great extent. We simulated our approach and found that while revisiting the websites, if a web crawler can find that which web pages have been updated or newly added since last visit, then there is no need to download the complete website every time. With the proposed solution it will be less time consuming for web crawlers to maintain the freshness of downloaded websites used by search engines. This on the one hand will help web crawlers to cover more of the web and on the other side it will reduce the traffic on websites due to them.

9. REFERENCES

1 Castillo, C., Marin, M., Rodrguez A. and Baeza-Yates, R. (2004), "Scheduling Algorithms for Web Crawling", WebMedia & LA-Web, pp 10-17.

2. Najork, M. and Wiener J. L. (2001), "Breadth-First Search Crawling Yields High-Quality Pages", WWW'01, 10th International World Wide Conference, pp. 114-118.
3. Cho, J., Gracia-Molina, H. and Page, L. (1998), "Efficient Crawling Through URL Ordering", Computer Networks and ISDN Systems (0169-7552), Volume 30, No. 1-7, pp 161-172.
4. Cho, J. and Gracia-Molina, H. (2000), "The Evolution of the Web and Implications for an Incremental Crawler", In Proceedings of 26th International Conference on Very Large Databases (VLDB), Cairo, Egypt, pp 200-209.
5. Mishra, S., Jain, A. and Sachan, A. K. (2011), "A Query based Approach to Reduce the Web Crawler Traffic using HTTP Get Request and Dynamic Web Page", International Journal of Computer Applications (0975-8887), Vol. 14, No. 3, pp 8-14.
6. Lawrence, S. and Giles, C. L. (1999), "Accessibility of Information on the Web", Published in Nature, Vol. 400, pp 107-109.
7. Najork, M. (2009), "Web Crawler Architecture", Encyclopaedia of Database Systems, Part 23, pp 3462-3465.
8. Blum, T., Keislar, D., Wheaton, J. and Wold, E. (1998), "Writing a Web Crawler in the Java Programming Language", http://java.sun.com.developer/technicalArticles/ThirdPa rty/WebCrawler/
9. Brandman, O., Cho, J., Garcia-Molina, H. and ShivaKumar, N. (2000), "Crawler-Friendly Web Servers", In Proceedings of the Workshop on Performance and Architecture of Web Servers (PAWS), Santa Clara, CA.
10. Bhatia, M.P.S. and Gupta, D. (2008), "Discussion on Web Crawlers of Search Engine", Proceedings of 2nd National Conference on Challenges & Opportunities in Information Technology, RIMT-IET, pp 227-230.

11. Tyagi, N. and Gupta, D. (2010), "A Novel Architecture for Domain Specific Parallel Crawler", Indian Journal of Computer Science and Engineering (0976-5166), Vol. 1, No. 1, pp. 44-53.
12. Baeza-Yates, R. and Castillo, C. (2002), "Balancing Volume, Quality and Freshness in Web Crawling", In Soft Computing Systems – Design, Management and Applications, HIS, Santiago, Chile, pp 565-572.