Regulating Frequency of a Migrating Web Crawler based on Users Interest

Niraj Singhal[#1], Ashutosh Dixit[*2], R. P. Agarwal[#3], A. K. Sharma[*4]
[#]Faculty of Electronics, Informatics and Computer Engineering
Shobhit University, Meerut, India
[1]sonia_niraj@yahoo.com, [2]prajanag@gmail.com
2Department of Computer Engineering
YMCA University of Science and Technology, Faridabad, India
[3]dixit_ashutosh@rediffmail.com, [4]ashokkale2@rediffmail.com

***Abstract*-Due to the lack of efficient refresh techniques, current crawlers add unnecessary traffic to the already overloaded Internet. Frequency of visits to sites can be optimized by calculating refresh time dynamically. It helps in improving the effectiveness of the crawling system by efficiently managing the revisiting frequency of a website; and appropriate chance to each type of website to be crawled at appropriate rate. In this paper we present an alternate approach for optimizing the frequency of migrants for visiting web sites based on user's interest. The proposed architecture adjusts the frequency of revisit by dynamically assigning a priority of revisiting to a site by computing the priority based on previous experience that how many times the crawler founds changes in content in 'n' visits and the interest of the users shown in the websites.**

Keywords- Search Engine, Migrant, Frequency regulation, Dynamic priority, User interest

I. INTRODUCTION

According to [15], the maximum web coverage of any popular search engine is not more than 16% of the current web size. Even the first Google index in 1998 had 26 million pages only, which has increased to three billion in 2012. In 2009 Google had 146 million users monthly that has increased to 931 million in 2010 and one billion in 2011. Since the web is very dynamic and 52% of its contents change daily [12], to maintain the up-to-date pages in the collection, a crawler needs to revisit the websites again and again. Due to excessive revisits, the resources like CPU cycles, disk space, and network bandwidth etc., become overloaded and due to such overloads sometime a web site may crash. Study [8] reports that about 40% of current internet traffic and bandwidth consumption is due to the web crawlers.

Using migrants (i.e. migrating crawlers), the process of selection and filtration of web documents can be done at web servers rather than search engine side which can reduce network load caused by the web crawlers [8,13]. Frequency of visits to sites can be optimized by dynamically assigning a priority to a site, The computation of refresh time helps in improving the effectiveness of the crawling system by efficiently managing the revisiting frequency of a website; and appropriate chance to each type of website to be crawled at a fast rate.

The process of revisiting to a web site can further be improved by adjusting the frequency of visit by considering the interest of users shown for specifics websites. For example, the websites for which users show more interest be crawled at a faster rate as compared to those that are less or rarely surfed by the users.

II. RELATED WORK

A general web search engine (figure 1) has three parts; a crawler, indexer and query engine. Web search engines [13] employ crawlers to continuously collect web pages from the web. The downloaded pages are indexed and stored in a database. The indexer extracts all the uncommon words from each page and records the URLs where each word has occurred. The result is stored in a large table containing URLs; pointing to pages in the repository where a given word occurs. The query engine is responsible for receiving and filling search requests from users. It relies on the indexes and on the repository.

Web crawlers traverse the web on the search engine's behalf, and follow links to reach different pages to download them. Starting with a set of seed URLs, crawlers extract URLs appearing in the retrieved pages, and store pages in a repository. This continuous updation of database renders a search engine more reliable source of relevant and updated information. The crawler has to deal with two main responsibilities i.e. downloading the new pages, and keeping the previously downloaded pages fresh. However, good freshness can only be guaranteed by simply revisiting all the pages more often without putting unnecessary load on the internet. With the available bandwidth which is neither infinite nor free, it is becoming essential to crawl the web in a way that is not only scalable but also efficient, if some reasonable measure of quality or freshness is to be maintained.

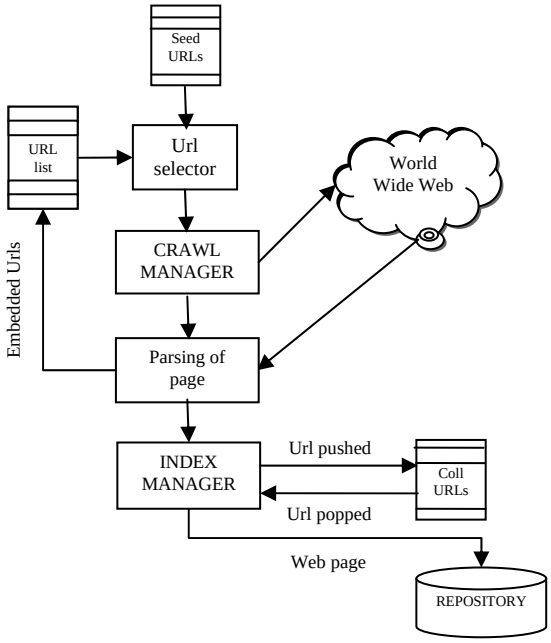

Fig. 1. A general web search engine

The centralized crawling techniques are unable to cope up with constantly growing web. Crawlers based on migrating agents become an essential tool for allowing such access that minimizes network utilization and also cop up with documents change. An agent is an autonomous entity that acts on behalf of others in an autonomous fashion, performs its actions in some level of pro-activity and reactivity, and exhibits some levels of the key attributes of learning, co-operation and mobility [15]. Agents can be classified according to the actions they perform, their control architecture, the range and effectiveness of their actions, the range of sensitivity of their senses and how much internal state they posses. Nwana [15] identifies seven types of agent i.e. collaborative agents, interface agents, migrating agents, information agents, reactive agents, hybrid agents and smart agents.

Migrating agents (or migrants) are computational software processes capable of roaming wide area networks such as WWW, interacting with foreign hosts, gathering information on behalf of its owner and coming back having performed the duties set by its user. Mobility allows an agent to move, or hop, among agent platforms. The agent platform provides the computational environment in which an agent operates. The platform from which an agent originates is referred to as the home platform, and normally is the most trusted environment for an agent. One or more hosts may comprise an agent platform, and an agent platform may support multiple computational environments, or meeting places, where agents can interact. They may cooperate or communicate with other agents making the location of some of its internal objects and methods known to other agents without necessarily giving all its information away.

The agent approach (as shown in figure 2) use the bandwidth of the network to migrate an agent to a platform, and allow it to continue to run after leaving a node, even if they lose connection with the node where they were created thereby provide the better utilisation on communication and allows parallel distributed applications. An agent can move on to other machines when necessary and can delegate tasks to other mobile agents in order to achieve real parallel applications. Various studies [12] have shown that distributed crawling methods based on migrating crawlers are an essential tool for allowing such access that minimizes network utilization and also keeps up with document changes.

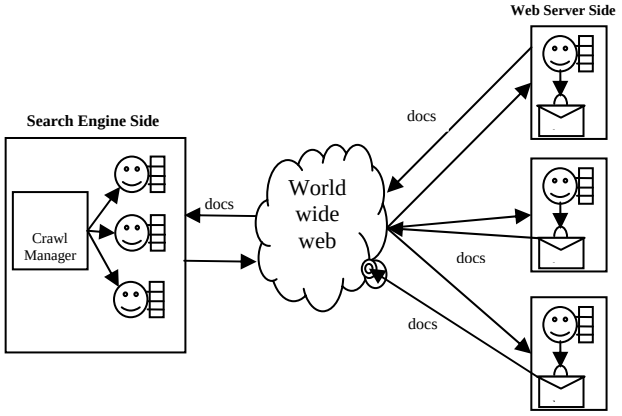

Fig. 2. Crawling with Migrants

Due to the deficiency in their refresh techniques [3], current crawlers add unnecessary traffic to the already overloaded Internet. Moreover there exist no certain ways to verify whether a document has been updated or not. Studies held on revisitation have demonstrated that 50% to 80% [9,12,19] of all Web surfing behavior involves pages that users have previously visited. While many revisits occur shortly after a page's first visit (e.g., during the same session using the back button), a significant number occur after a considerable amount of time has elapsed [16].

Bullot and Gupta [8] introduced data mining approach for optimizing performance of an Incremental Crawler. With the method presented, it is the user who chooses which pages the crawler must update. Kuppusamy and Aghila [10] involves user participation in larger extent in order to get the focused and more relevant information. Choudhari and Choudhari [17] address the scheduling problem and solution for the web crawlers with the objective of the optimizing the resources like freshness of repository and the quality of the index. They divided the web content providers into two parts i.e. active and inactive. For inactive content providers they use agents who continuously crawls the content providers and collect the update pattern of the content providers.

Glover et al [5] described a meta search engine architecture, that allows users to provide preferences in the form of an information need category. This extra information is used to direct the search process, providing more valuable results than by considering only the query. Using this architecture, identical keyword queries may be sent to different search engines, and results may be scored differently for different users. Malakar [18] presents a novel approach to personalised search using the concept of "iAGENT" an intelligent agent that assists a user to get relevant documents by modifying the query given by the user in accordance with the web pages previously visited. It presents a novel approach to personalise the search results and improve the relevancy rate.

Qiu et al [6] study how a search engine can learn a user's preference automatically based on her past click history and how it can use the user preference to personalize search results. They propose a framework to investigate the problem of personalizing web search based on users' past search histories without user efforts. Based on this correlation, they describe an intuitive algorithm to actually learn users' interests. They propose two different methods, based on different assumptions on user behaviors, to rank search results based on the user's interests we have learned.

Liu [11] proposes to measure page importance through mining user interest and behaviors from web browse logs. Unlike most existing approaches which work on single URL, here, both the log mining and the crawl ordering are performed at the granularity of URL pattern. The proposed URL pattern-based crawl orderings are capable to properly predict the importance of newly created (unseen) URLs. Vipul et al approach [20] is build on the basis of which a web crawler maintains the retrieved pages "fresh" in the local collection. Towards this goal the concept of Page Rank and Age of a web page is used. As higher page rank means that more number of users are visiting that very web page and that page has higher link popularity. Age of web page is a measure that indicates how outdated the local copy is. Using these two parameters a hybrid approach is proposed that can identify important pages at the early stage of a crawl, and the crawler re-visit these important pages with higher priority.

Dixit et al [2] propose an efficient approach for building an effective incremental web crawler with an approach for optimizing the frequency of visits to sites. The approach adjusts the frequency of visit by dynamically assigning a priority to a site. A mechanism for computing the dynamic priority for any site has been developed. An alternate approach [13] to manage the process of revisiting of a website, employs an ecology of crawl workers to crawl the web sites. Crawl manager extracts URLs from each queue of URLs and distribute them among crawl workers. The architecture manages the process of revisiting of a web site with a view to maintain fairly fresh documents at the search engine site. The computation of refresh time helps in improving the

effectiveness of the crawling system by efficiently managing the revisiting frequency of a website; and appropriate chance to each type of website to be crawled at a fast rate.

Based upon up-dation activity, documents can be categorized as a static web page which is not updated regularly, dynamically generated web pages e.g. database driven web page, very frequently updated parts of web pages e.g. news website, share market website, updated pages generated when website administrator updates or modifies its website. Keeping in view the above categorization, the crawler may visit a site frequently and after every visit its frequency of future visits may be adjusted according to the category of the site. The adjusted refresh rate/frequency can be computed by the following formula :

$t_{n+1} = t_n + \Delta t$ …Eqn. (1)

Where t_n: is current refresh time for any site.

t_{n+1}: is adjusted refresh time.

Δt: is change in refresh time calculated dynamically.

The value of Δt may be positive or negative, based upon the degree of success (pc) that the site contains the volatile documents. The degree of success is computed in terms of no. of hits by detecting the frequency of changes occurred in the documents on a site. For example, if the crawler encounters a document being updated six times out of its ten visits, the degree of success (pc) is assigned as 0.6 to that site.

A unit step function u(pc) has been employed for the computation of Δt, which can be defined as follows

$\Delta t = \{(1-pc/pg)*u\ (pc-pg) + (1-pc/pl)*u\ (pl-pc)\}*tn$ Eqn. (2)

Where pg, pl are the boundary conditions i.e. upper and lower threshold values of pc respectively,

$$u(x) = \begin{cases} 1 & \text{if } x>0 \\ 0 & \text{otherwise} \end{cases}$$

From all above it is evident that user's interest plays an important role while calculating revisit frequency of a website. In this paper an alternate approach for optimizing the frequency of migrants for visiting web sites based on user's interest is presented. The proposed architecture manages the process of revisiting to a web site with a view to maintain fairly fresh documents at the search engine site. The approach adjusts the frequency of revisit by dynamically assigning a priority of revisiting to a site by computing the priority based on previous experience that how many times the crawler found changes in content in 'n' visits and the interest of the users shown in the websites. The pages visited by the users more, be given high priority as compared to those that are less or rarely visited.

III. PROPOSED WORK

In traditional crawling the pages from all over the web are brought to the search engine side and then processed, and only after analyzing the page it can be concluded that whether the page is useful or not. Studies suggest that most of the times, the downloaded page is not useful in the sense that it has not updated since its last crawl. In such cases the efforts made to send an HTTP request to the web server and bringing the page to the search engine side seems to be useless and also causing unnecessary load on to the internet.

In the migrating approach of crawling [12,14], migrants are allowed to migrate to a destination host where the interactions, downloading and processing of documents can take place locally on to the web server itself as shown in figure 3. The main concern is to move the computations to the data rather than the data to the computations. By migrating to the location of the resource, a migrant can interact with the resource much faster than using HTTP across the network.

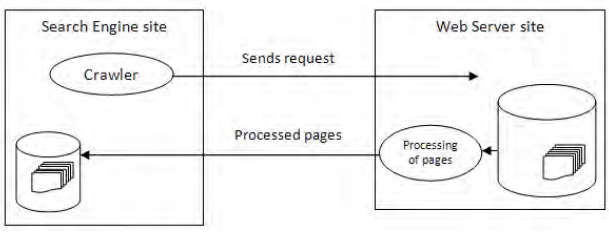

Fig. 3. Crawling with Migrants

Normally, while calculating revisit frequency dynamically, based on previous experience that how many times the crawler found changes in content of a website in 'n' visits. Here the websites that change at same rate needs

to be crawled at the same rate. However, in practice the user's interest in the websites changing at the same rate need not be the same.

In the proposed approach it is suggested that the websites in which user shows more interest be crawled a faster rate as compared to those in which user shows less or rare interest. For example as shown in figure 4, consider ten documents whose contents get changed at the same rate, then these need to be revisited at the same rate. However, as all users do not show same level of interest in all the documents i.e., Doc4, Doc6, Doc7, Doc9 are not visited by any user, Doc1, Doc2, Doc3, Doc5, Doc10 are visited by one user only, and Doc8 is visited by 4 users. So, it is proposed that Doc8 be visited at faster rate than Doc1, Doc2, Doc3, Doc5, Doc10, and than Doc4, Doc6, Doc7, Doc9.

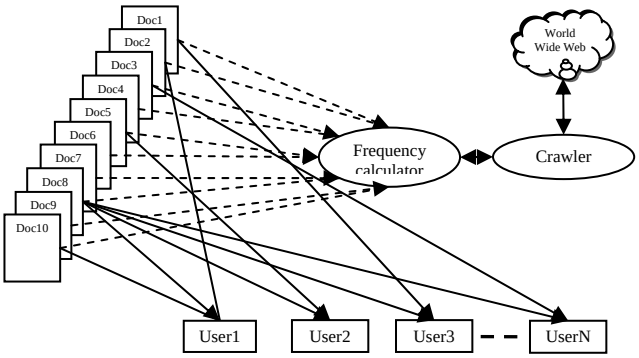

Fig. 4. Finding user's interest

Now, considering the user's interest shown in websites, the revisit frequency be calculated giving equal weightage to the change in contents and to the interest of the user. Now, Equ. (1) can be rewritten as,

$$t_{n+1} = t_n + (\Delta t1 + \Delta t2) \qquad \text{Eqn. (3)}$$

Where,

t_n: is current refresh time for any site.

t_{n+1} : is adjusted refresh time.

$\Delta t1$: is change in refresh time calculated dynamically for change in contents as per Equ. 2.

$\Delta t2$: is change in refresh time calculated as per user's interest shown in websites.

A unit function for user's interest can be defined as,

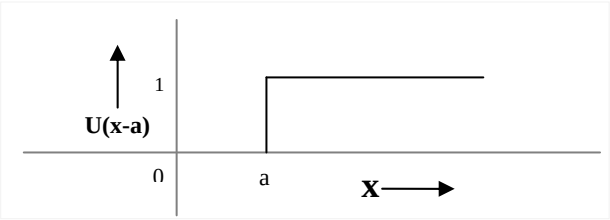

Fig. 5. Unit function for user's interest (average interest a)

$$\text{and } u(x-a) = \begin{cases} 1 & \text{if } x >= a \\ 0 & \text{otherwise} \end{cases}$$

Here, it is assumed that users show a minimum interest (less than a) in each website whose information is contained in the database. In such case the crawler crawls all the websites with normal frequency. However, as user's interest increases in a specific website or suddenly a set of users starts surfing a specific website then it is suggested that frequency of revisit be calculated using following formula as per Equ. (4).

$$\Delta t2 = \{(1-pc/pg)*u(pc-pg+a) + (1-pc/pl)*u(pl-pc+a)\} * t_n \qquad \ldots\text{Eqn. (4)}$$

Where pg, pl are the boundary conditions i.e. upper and lower threshold values of pc respectively, and pc is given a threshold value .5 for example.
Consider three websites, S1, S2 and S3 which have same revisit frequency. Let Δt1, the refresh time based on change in the contents is computed with following data set, and is same for all three websites.
Consider the data given below:

tn = 100 units
Pl = 0.3
Pg = 0.7
Pc = say 0.85

Δt1 would be computed as given below:
Δt1= {(1-0.85/0.7)*1+(1-0.85/0.3)*0} *100 = - 150/7 =-21.42
Refresh time is decreased
From the above analysis, it is concluded that for sites S1,S2 and S3 if changes in pages are found at same frequency, then revisit frequency is decreased by 21.42 units for all, i.e. if current refreshing time is 100 units then new refreshing time would be 78.58 units for all. Moreover, even if users show different level of interest in the above three websites, the revisit frequencies for all three websites remain same.
Now, let equal weightage is given to the change in contents and the users interest. Following examples show computation of refresh time for three cases by taking different sets of data.

Case 1: let tn = 50 units
Pl = 0.5
Pg = 0.8
Pc = say 0.7
Δt2 for sites S1,S2 and S3 would be:
Δt2= {(1-0.7/0.8)*0 + (1-0.7/0.5)*0} * 50 = 0

The new refresh time
tn+1 = tn +Δt1 +Δt2
tn+1 = 100 -10.71 +0=89.29 units
Refresh time is decreased

Case 2: tn = 50 units
Pl = 0.5
Pg = 0.8
Pc = say 0.3
Δt2 for sites S1,S2 and S3 would be:
Δt2= {(1-0.3/0.8)*0 + (1-0.3/0.5)*1} * 50 = 20 units
The new refresh time
tn+1 = tn +Δt1 +Δt2
tn+1 = 100 -10.71 + 20= 109.29 units
Refresh time is increased

Case 3: tn = 50 units
Pl = 0.5
Pg = 0.8
Pc = say 0.9
Δt2 for sites S1,S2 and S3 would be:
Δt2= {(1-0.9/0.8)*1 + (1-0.9/0.5)*0} * 50 = -6.25
The new refresh time
tn+1 = tn +Δt1 +Δt2
tn+1 = 100 -10.71 -6.25= 83.04 units
Refresh time is decreased

Here, it is also evident from the above analysis that on considering the user's level of interest in the websites, the refreshing time to revisit is not same.

IV. PERFORMANCE ANALYSIS

With the increase in the availability of web pages on the Internet, the major problem faced by the present search engines is the difficulty in information retrieval. It is a challenging task to identify the desired pages from amongst the large set of web pages found on the web. Problem grows exponentially with further increase in the size of the Internet. The number of web pages which have gone under change increases as the web grows, as shown in figure 6.

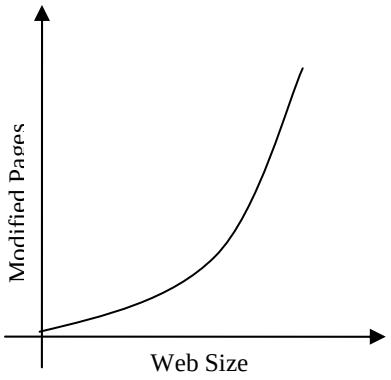

Fig. 6. Effect of Web growth on updation of pages

With this increase in web size the crawler traffic will definitely be more, and quality of pages in the collection starts declining as shown in figure 7.

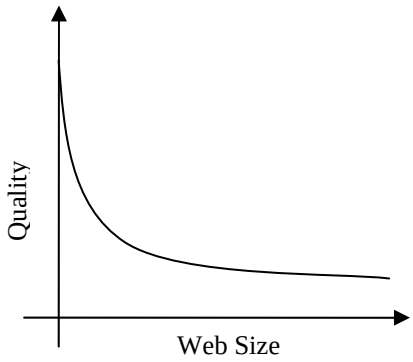

Fig. 7. Effect of Web growth on quality of pages

The architecture given in this work effectively takes into consideration the computation of refresh time dynamically based on user's interest. The proposed download mechanism is independent of the size of the Internet as it is based on the self adjusting refresh time based strategy. Since, only those web pages are retrieved which have undergone updation and in which users have shown their interest, the system would continue to give modified pages only irrespective of the size of the Internet (figure 8). Moreover, since the pages are relevant in the sense that they have gone under updation and users have shown their interest in these pages, the traffic on network will be reduced and hence only quality pages are retrieved (figure 9).

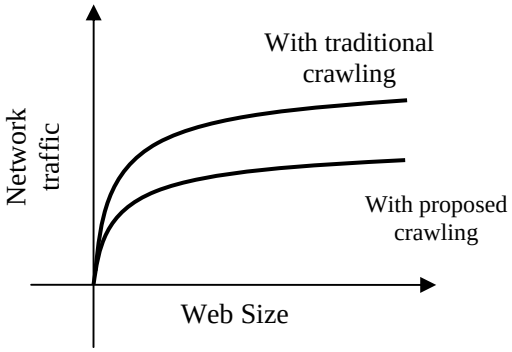

Fig. 8. Performance analysis of proposed architecture Web size vs Network traffic

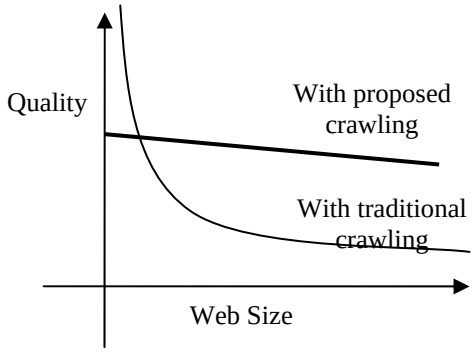

Fig. 9. Performance analysis of proposed architecture Web size vs Quality of collection

Considering various performance parameters like quantity of web pages downloaded, their quality and the network traffic, the proposed mechanism definitely holds an edge above the present conventional crawling strategies based on fixed refresh time.

V. CONCLUSION

Frequency of visits to sites can be optimized by dynamically assigning a priority to a site, The computation of refresh time helps in improving the effectiveness of the crawling system by efficiently managing the revisiting frequency of a website; and appropriate chance to each type of website to be crawled at a fast rate. In this paper we presented that revisiting to a web site can further be improved by adjusting the frequency of visit by considering the interest of users shown for specifics websites. The websites for which users show more interest be crawled at a faster rate as compared to those that are less or rarely surfed by the users.

REFERENCES

[1] Alexandros Ntoulas, Junghoo Cho and Christopher Olston, "What's new on the Web ? The Evolution of the Web from a Search Engine perspective", in proceedings of the World-Wide Web Conference (WWW), May 2004.
[2] Ashutosh Dixit and A. K Sharma, "Self Adjusting Refresh Time Based Architecture for Incremental Web Crawler", International Journal of Computer Science and Network Security (IJCSNS), Vol. 8, No.12, Dec 2008.
[3] J. Cho and H. Garcia-Molina, "Estimating Frequency of Change", Technical report, DB Group, Stanford University, Nov. 2001.
[4] Dirk Lewandowski, "Web searching, search engines and Information Retrieval, Information Services & Use", 25 (2005) 137-147, IOS Press, 2005.
[5] Eric J. Glover, Steve Lawrence, Michael D. Gordon, William P. Birmingham, and C. Lee Giles, "Improving Web searching with user preferences", communications of the ACM December 2001/Vol. 44, No. 12, ACM 0002-0782/01/1200, pp. 97-102, 2001
[6] Feng Qiu, Junghoo Cho, "Automatic Identification of User Interest For Personalized Search", International World Wide Web Conference Committee (IW3C2).WWW 2006, Edinburgh, Scotland, ACM 1595933239/06/0005, May 23–26, 2006.
[7] Gulli A. and Signorini A., "The Indexable Web is More than 11.5 billion pages", in proceedings of the Special interest tracks and posters of the 14th international conference on World Wide Web, Chiba, Japan, pp. 902-903, 2006.
[8] Hadrien Bullot and S K Gupta, "A Data-Mining Approach for Optimizing Performance of an Incremental Crawler", in proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03).
[9] Herder, E, "Characterizations of user Web revisit behavior", in proceedings of Workshop on Adaptivity and User Modeling in Interactive Systems", 2005.
[10] K.S. Kuppusamy and G. Aghila, "FEAST - A Multistep Feedback Centric, Freshness Oriented Search Engine", in proceedings of 2009 IEEE International Advance Computing Conference (IACC 2009).
[11] Minghai Liu, Rui Cai, Ming Zhang, and Lei Zhang, "User Browsing Behavior-driven Web Crawling", CIKM'11, Oct. 24–28, 2011, Glasgow, Scotland, UK., ACM 978-1-4503-0717-8/11/10s.
[12] Nath R., Bal S., and Singh M., "Load Reducing Techniques an the Websites and other Resources: A comparative Study and Future Research Directions," Computer Journal of Advanced Research in Computer Engineering, Vol. 1, No. 1, pp. 39-49, 2007.
[13] Niraj Singhal, Ashutosh Dixit and A. K. Sharma, "Design of A Priority Based Frequency Regulated Incremental Crawler", published in proceedings of International Journal of Computer Applications (IJCA), Vol. 1, No. 1, Article 8, pp 47-52, Harvard Press US 2010. ISSN: 0975–8887, 2010.
[14] Niraj Singhal, R.P. Agarwal, Ashutosh Dixit and A. K. Sharma, "Information Retrieval from the Web and Application of Migrating Crawler", 978-0-7695-4587-5/11 © 2011 IEEE, DOI 10.1109/CICN.2011.99, pp. 476-480, Print ISBN: 978-1-4577-2033-8.2011, Oct. 2011.
[15] Nwana H. S., "Software Agents: An Overview," Knowledge Engineering Review, Cambridge University Press, 1996.
[16] Obendorf, Hartmut, H, Weinreich, E. Herder, and M. Mayer., "Web page revisitation revisited: Implications of a long-term click-stream study of browser usage", in proceedings of CHI '07, 2007.
[17] Rahul Choudhari and Ajay Choudhari, "Increasing Search Engine Efficiency using Cooperative Web", in proceedings of International Conference on Computer Science and Software Engineering, 2008.
[18] Sompa Malakar, "iAGENT: A Novel and Intelligent Assistant to Personalised Search", International Journal of Computer Applications (0975 – 8887) Vol. 1, No. 1, 2010.
[19] Tauscher, L. and S. Greenberg, "How people revisit Web pages: Empirical findings and implications for the design of history systems", International Journal of Human-Computer Studies, 47(1):97-137, 1997.
[20] Vipul Sharma, Mukesh Kumar, Renu Vig, "A Hybrid Revisit Policy For Web Search", Journal of Advances in Information Technology, Vol. 3, No. 1, Feb. 2012, pp. 36-47.