मयि सर्वमिदं प्रोतं
B S A I T M
Since 2001

Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org
Volume 5, Special Issue 2, December (2015)

A crawling mechanism to maintain freshness of downloaded collection based on user perspective and page updation frequency

Shilpa Sethi
CE department, YMCAUST, Faridabad, India.

Ashutosh Dixit
CE department, YMCAUST, Faridabad, India.

Abstract – The World Wide Web is a huge source of dynamic information which gets updated on daily, weekly, monthly or yearly basis. Search engine uses crawlers to periodically pull remote web pages to maintain the updated database. Keeping in view the network congestion problem, the crawler is expected to visit the information sources in an optimized manner. In this paper, an efficient web crawling technique based on user's perspective and page updation frequency is being proposed. The pages which are more often demanded by the user are detected and accordingly the revisit interval for the page is reset

Index Terms –— search engine, web crawler, revisit interval, user behavior.

This paper is presented at International Conference on Recent Trends in Computer and information Technology Research on 25th& 26th September (2015) conducted by B. S. Anangpuria Instituteof Technology & Management, Village-Alampur, Ballabgarh-Sohna Road,Faridabad.

1. INTRODUCTION

The www contains millions of web sites accessed via internet. Each of these web sites contains large no. of hypertext documents. To retrieve the information from such a huge collection, the user needs some sort of automated tool that can connect to different web sites, search for the required information and gives the results to the user. These automated tools are known as 'search engines [3]'. When a user enters a query, the search engine apparently searches all the web sites on the internet and returns the results that match best with the user query. The kind of searching through takes many hours, much larger than one is willing to wait, but the search engine returns the results after a few milliseconds only. This is because the search engines gathers information much before it is actually needed with the help of a program called crawler. Basically, its the crawler who visits the different web sites in order to download the web documents and stores them in search engine's database[2]. So, it is important to note that when the user enters the query, the search engine does not search on internet, rather it searches its own database which has been in advance populated by the crawler [3]. But the main issue that needs to be addressed here is how often the crawler should visit the web in order to cope up with the dynamic nature of web [7]. As the contents on web changes at very fast rate, the crawler need to revisit the web as frequent as possible .But in present scenario, when internet has become the part of our life and internet usages are increasing day by day, revisiting the web sites not only causes network congestion, but also crawler will not be able to download the required information in time [1]. So, it becomes necessary to discover the factors that affect the revisit frequency of crawler. In this paper an efficient crawling technique based on user interest is being proposed.

The paper is structured as follow: section 2 gives the overview of crawling process and discusses an efficient crawler based on change frequency of web page. Section 3 describes the proposed technique. The result analysis is done in section 4. Section 5 concludes the paper document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website

2. RELATED WORK

This section provides the detail description of crawling process and a popular model for crawler revisit frequency

2.1 Working of web crawler

The general architecture of web crawler is shown in fig 1. It has four components namely URL frontier, downloader, parser and URL filter & duplicate eliminator [4]. The description of each component is given below.

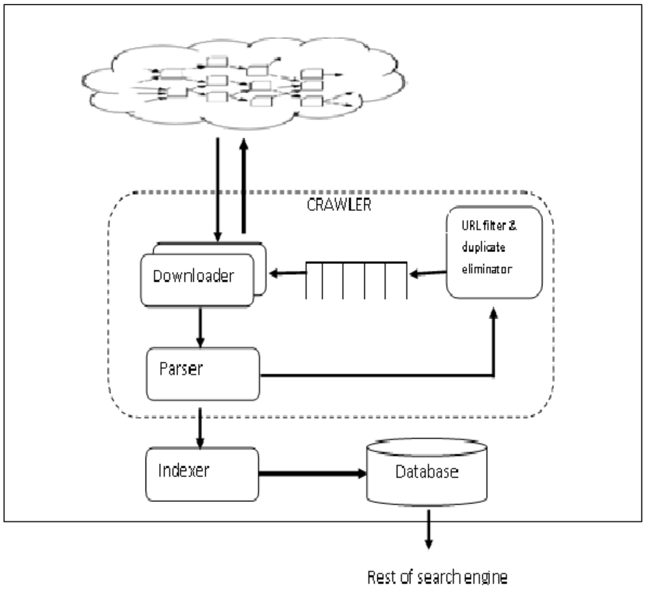

Fig 1 Working of crawler

URL frontier – it is a queue of unvisited URLs which is initially set by search engine administrator or specialized program [1]. The basic operation of any crawler starts by picking up a seed URL from the URL frontier. The page corresponding to fetched URL is downloaded from the web and links present in downloaded pages are added back to URL frontier for further crawling. The process of fetching and downloading the document continues until URL frontier is empty or some other conditions are met. Downloader- it fetches the URL from URL frontier and depending on the host protocol downloads the pages from corresponding web server. It then passes the page to parser module.

Parser – It extracts the text and link information from the downloaded pages. The text information is passed to the indexer, which apply some sort of indexing algorithm to build search engine's database. The extracted link information is passed to URL filter & duplicate elimination module.

URL filter & duplicate elimination module- in order to enhance the accuracy of crawling process, the extracted links are examined by filtering module. The URL filter module determines whether to include or exclude the URL in URL frontier based on certain crawling techniques. For example, the focused crawler downloads the pages from specific domains only (say, .ac or .gov only) whereas a depth sensitive crawler poses limit on crawling depth . It also assigns the priority to each URL based on some prioritization scheme.

The working of web crawler shows that the main task of crawler is to download the new documents and update the existing document after a specific interval [5].The paper focus on determining the crawling interval so that the updated user specific information can be downloaded in an optimized way. In the next section a mathematical model for crawler revisit frequency [2] has been described, which forms the platform for the proposed crawling method.

2.2 Calculation of revisit frequency of crawler based on page updation frequency (RFCUF)

Dixit et al [1] proposed a novel mechanism for calculating the revisit frequency of crawler based on updation frequency of a web page. They pointed out that since all pages do not change in same time period, e.g. news; share market web pages etc are updated frequently as compared to pages related to pay commission. So, all pages are needless to be refreshed at same frequency. Further, it is also pointed out that revisit frequency is not directly proportional to page updation frequency; it increases up to a certain threshold value after that remains constant till second threshold value and then starts decreasing as shown in fig2

The method dynamically adjusts the frequency of future visit to a web page by using the following formulas.

$f_{n+1} = f_n + \delta f$ ....(1a)

Where:

- f_n denotes current revisit frequency of nth web page
- f_{n+1} denotes the updated revisit frequency of nth web page for future.

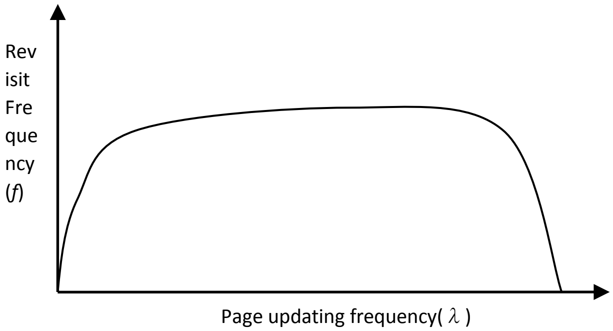

Fig 2 : Graph showing changing nature of revisit frequency

The δf can be computed using a unit step function which is defined as follow:

δf= [{f_n×(λ_i /λ_{i-1} -1) × u (λ_i – λ_l)× u (λ_m – λ_i) × u (λ_g- λ_I)}+ {f_n×(1- λ_i/λ_g)×u(λ_i– λ_g)×u(1- λ_i}]
...(1b) Where:

- $\lambda_{l,}$ λ_g , λ_m denotes the lower ,upper and middle threshold boundary values for page updation frequency respectively.
- λ_i denotes the change frequency of page at i period of time.
- $\lambda_{i\ -1}$ denotes the change frequency of page at i-1 period of time.
- u(x) denotes a unit step function ; u(x)=1 , if x>0 ,otherwise u(x)=0

Comparing the revisit frequency of two web pages

To explain how the revisit frequency of each page is refreshed separately, let us consider two pages Pn and Pm respectively

Case1: To calculate the revisit frequency of a page p_n at i+1 th interval of time, let us assume the revisit frequency if page P_n at ith interval of time denoted by f_i = 20 times/ unit time , page updation frequency λ_i = 0.2 and λ_{i-1} =0.15 , the boundary conditions for page updation frequency are: λ_l = 0.1, λ_m = 0.5 and λ_g = 0.9 ,

Using eq^n (2), δf would be computed as given below:

δf = [{20×(0.2 / 0.15 -1)×u (0.2 – 0.1)×u (0.5- 0.2)×u (0.9- 0.2)} + {20×(1- 0.2 / 0.9)×u (0.2 – 0.9) ×(1-0.2}]. = 7

So, the revisit frequency for page p_n at i+1 th interval of time can be computed by eqn(1a) as follow :

f_{i+1} = 20+7= 27 times/ unit time

Case2: To calculate the revisit frequency of a page p_m at i+1 th interval of time , let us assume the revisit frequency if page P_m at ith interval of time is also f_i = 20 times/ unit time ,but page updation frequency λ_i = 0.3 and λ_{i-1} =0.2 , the boundary conditions for page updation frequency are: λ_l = 0.1, λ_m = 0.28 and λ_g = 0.7,

Using eq^n (2), δf would be computed as given below:

δf = [{20×(0.3 / 0.2 -1)×u (0.3 – 0.1)×u (0.28- 0.3)×u (0.7- 0.3)} + {20×(1- 0.3 / 0.7)×u (0.3 – 0.7) ×u(1-0..3)}] = 0

So, the revisit frequency for page p_m at i+1 th interval of time can be computed by eqn(1a) as follow :

f_{i+1} = 20+0=20 times /unit time

So it is observed that the revisit frequency of Pn is increased whereas the revisit frequency of Pm remains the same.

A critical look at the available literature indicates that although calculating the revisit frequency of a page based on page updation frequency reduces the network congestion problem but, it may not be necessary that a page, whose updation frequency is more, is in demand by the user. As revisiting an undermanded page at higher frequency are unnecessarily increasing the network traffic so, it become vital to consider the perspective of user while adjusting the revisit frequency of a page.

The proposed crawling mechanism discussed in the next section overcomes the above short coming by incorporating the page usages information with page updation frequency.

3. PORPOSED CRAWLING MECHANISM

An efficient web crawling technique based on user perspective and page updation frequency is proposed here as shown in fig 3. It takes the user interest [6] on a page into account with the aim to determine the importance of freshness of a page for the user. The major components of proposed model are as follows: User interface, PDF calculator, Query processor, Revisit frequency calculator, and Crawler. The detail description of each component is given in following sections.

3.1 User Interface

The user submits its search need in the form of query here. The user interface passes the query terms to query processing module to retrieve the pages which matches best with the user query from search engine database. After receiving the sorted list of Pages from query processing module, presents the results back to user. When the user clicks any of the links from the presented list, It sends a signal "something to monitor "to PDF calculator.

3.2 PDF calculator

After receiving the signal from user interface it starts recording the various facts about the importance of a page from user perspective. For this it assigns a page demand factor to each page clicked by the user. The PDF of a page Pi can be computed by the eqn (2) as follow.

PDF(Pi)= click(pi) +time spent(pi) +action(Pi)
.. ..(2)

Where:

- Click(pi) denotes click weight of page Pi
- Time(pi) denotes the time weight on page pi
- Action(pi) denotes the action weight of page pi

Initially click weight of all the pages are set to 0. If the no. of clicks on a page is less than 10000(click threshold value) then click(pi) is incremented by 0.01[2].

The time is also an important factor while analysing the importance of page from user' point of view. So, time weight (pi) can be computed by comparing the relevancy of page pi with respect to any page P which had been viewed for the longer time as given in eqn (3).

$$\text{time(pi)}=\frac{\text{time spent on page pi}}{\text{highest time spent on any page p}} \quad ....(3)$$

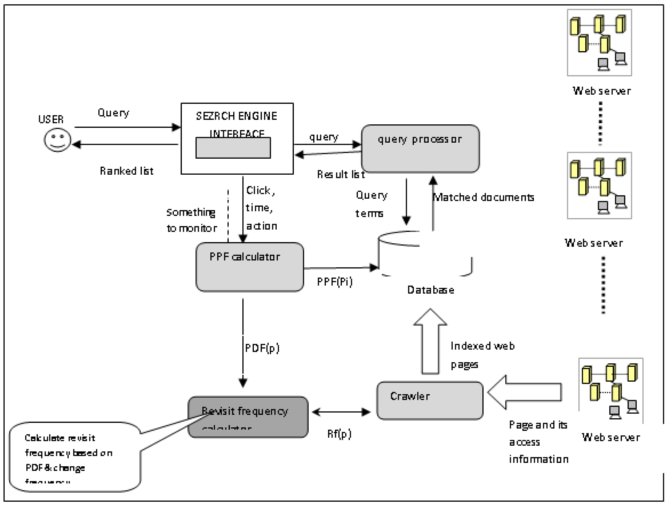

Fig 3 :Proposed Architecture

The action weight can be assigned using table 1. It is observed that a page which is printed has higher utility at present than saving the page. A page which is sent is having less utility among all.

Table 1 Action weight

Action	Weight
Print	0.4
Save	0.3
Bookmark	0.2
Send	0.1
No action	0

The PDF information is used to compute the revisit frequency of a page as a page which has higher importance for the users must be fresh in its contents. Initially , The PDF of each page is set to zero and it keeps on updating as the demand of this page increases from one user to another.

3.3 Revisit frequency calculator

This module calculates the revisit frequency of each crawled page on the basis of page demand and changing frequency of page. The revisit frequency Rfn+1 can be computed by eqn(4) as follows.

$$Rfn+1= Rfn+\Delta Rf \qquad .....(4)$$

Where:

- Rfn is current revisit frequency.
- ΔRf is change in frequency computed by eqn (5)
- Rfn+1= is adjusted revisit frequency.

$$\Delta Rf\ (p)=0.3\ PDF(P)\ +0.7\ \ \delta f(p) \qquad .....(5)$$

The revisit frequency calculator takes the value of PDF(p) from PDF calculator . It calculates the page changing probability represented by δf(p) by using eqn (1b).

3.4 Crawler

It maintains the priority list of URLs to be crawled next on the basis of revisit interval. It takes the revisit interval of each page from revisit frequency calculator module Based on the priority, it fetches the pages from different web servers and index them in search engine database. it dynamically calculates the change frequency of page. It also passes the change frequency information to revisit frequency calculator.

3.5 Query processor

It takes the query from user interface; remove the non functional terms from the query. Based on the functional terms retrieves the documents from search engine database. Apply the pageRank [9, 10] algorithm to sort the matched pages and returns the ranked list of URLs to user interface.

4. RESULTS AND DISCUSSIONS

The analysis of sample dataset of 100 URLs has been conducted to identify the demand factor of each URL which can be further utilized in revisit calculation mechanism. The proposed system is implemented using JAVA and SQL .The concentration has been done to record the changing frequency of revisit with respect to user demand factor and page updation rate.

It has been observed that revisiting frequency of a page increases with increase in page demand and increase in page updation frequency up to a certain limit then remains almost constant after a cut off value. The graph shown in fig 4 compare the revisit frequency of page based on proposed mechanism and RFCUF disused in section 2.(B)

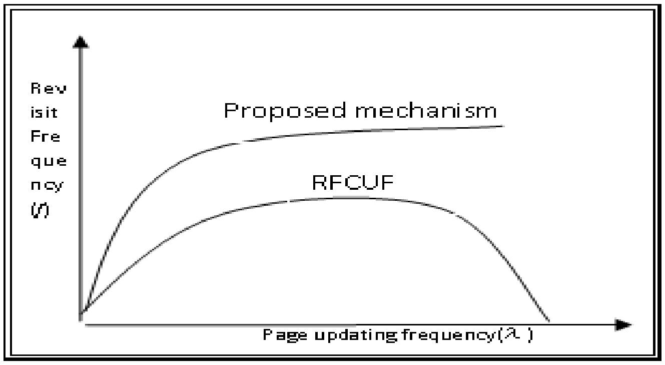

Fig 4: Comparison of proposed system with RFCUF

5. CONCLUSION

With the increasing usage of web, user wants the fresh information related to its area of interest. The no. of clicks, time spent and action performed on a page reflects the user interest in the page. As compared to previous proposed mechanism for calculating revisit frequency of a page, the proposed mechanism help to manage revisiting frequency based on user interest which was not earlier included

REFERENCES

[1] Ashutosh Dixit, A.K.Sharma "A mathematical model for crawler revisit frequency" Advance Computing Conference (IACC), 201010.

[2] Yadu Nagar, Niraj Singhal " A user search history based approach to manage revisit frequency of incremental crawler" international journal of computer application(0975-887) 2013

[3] 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.

[4] TriuptiV Udapure , Ravindra D Kale " study of web drawler and its different types" IOSR-JCE eISSN 2278-0661 , p-ISSN 2278-8727 (2014)

[5] 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.

[6] Shlpa Sethi, Ashutosh Dixit" Design of personalized search system based on user interest and query structuring "Proceedings of 2nd International Conference on Computing for Sustainable Global Development Page(s): 1346 - 1351 Print ISBN: 978-9-3805-4415-1

[7] Vipul Sharma , Mukesh Kumar " A hybrid revisit policy for web search" Journal of advance in information technology 2012

[8] Jianchao Han , Nick Cercone" A weighted fressness metric for maintaining search engine local repository" conference web intelligence 2004 proceeding of IEEE

[9] Sethi Shilpa, Dixit Ashutosh " A comparative study of link based pge ranking algorithm" International journal of advance technology in engineering and science(IJATES) ISSN:2348-7550, volume-3, special issue- 01, May 2015

[10] Mittal Ankur, Sethi Shilpa " A novel approach to page ranking mechanism based on user interest" International journal of advance technology in engineering and science(IJATES) ISSN:2348-7550, volume-3, special issue- 01, May 2015

Authors

. **Shilpa Sethi** received her M. Tech. in Computer Engineering from MD University Rohtak, in the years 2009 and MCA from Kurukshetra University in the year 2005. She is presently serving as Assistant Professor in the department of computer engineering at YMCA University of Science & Technology, Faridabad Haryana. Her research interests include Internet Technologies and web mining.

Ashutosh Dixit received his PhD and M. Tech. in Computer Engineering from MD University Rohtak, in the years 2010 and 2004 respectively. He is presently serving as Associate Professor in the department of computer engineering at YMCA University of Science & Technology, Faridabad Haryana. He has published around 80 research papers in various International journals and conferences. His research interests include Internet Technologies, Data Structures and Mobile and Wireless networks.