CHAPTER-4
REVISIT FREQUENCY REGULATION

4.1 INTRODUCTION

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.

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:-

\[ t_{n+1} = t_n + \Delta t \] ……(1)

Where,

- \( t_n \) : is current refresh time for any site.
- \( t_{n+1} \) : is adjusted refresh time.
- \( \Delta t \) : is change in refresh time calculated dynamically.

The value of \( \Delta t \) may be positive or negative, based upon the degree of success (\( p_c \)) 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 ($p_c$) is assigned as 0.6 to that site.

A unit step function $u(p_c)$ has been employed for the computation of $\Delta t$, which is defined as follows:

$$\Delta t = \{(1-p_c/p_g)*u(p_c-p_g) + (1-p_c/p_l)*u(p_l-p_c)\}*t_n \quad \ldots \ldots \ldots (2)$$

Where $p_g$, $p_l$ are the boundary conditions i.e. upper and lower threshold values of $p_c$ respectively. The value of unit function $u(x)$ for any value of $x$ is defined as,

$$u(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad \ldots \ldots \ldots (2A)$$

From all above it is evident that user’s interest plays an important role while calculating revisit frequency of a website. Here, an alternate approach for optimizing the frequency of migrants for visiting websites based on user’s interest is presented.

The proposed architecture manages the process of revisiting to a website 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.

### 4.2 REGULATING FREQUENCY OF MIGRATING CRAWLER BASED ON USERS INTEREST

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, 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. 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. 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, consider ten documents (see Figure 4.1,) whose contents get changed at the same rate, then these needs 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 then Doc4, Doc6, Doc7, Doc9.

![Diagram of User Interest](image)

**Figure 4.1:** 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, Eqn. (1) can be rewritten as,

\[ t_{n+1} = t_n + (\Delta t_1 + \Delta t_2) \] \hspace{1cm} (3)

Where,

- \( t_n \) : is current refresh time for any site.
- \( t_{n+1} \) : is adjusted refresh time.
- \( \Delta t_1 \) : is change in refresh time calculated dynamically for change in contents as per Eqn. (2).
- \( \Delta t_2 \) : 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 shown in Figure 4.2.

Figure 4.2: Unit function for User’s Interest (Average Interest ‘a’)

\[ u(x-a) = \begin{cases} 
1 & \text{if } x \geq a \\
0 & \text{otherwise}
\end{cases} \] \hspace{1cm} (3A)
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 Eqn. (4).

$$\Delta t_2 = \{(1-p_c/p_g)u(p_c-p_g+a) + (1-p_c/p_l)u(p_l-p_c+a)\} \times t_n$$ ......(4)

Where $p_g$, $p_l$ are the boundary conditions i.e. upper and lower threshold values of $p_c$ respectively, and $p_c$ is given a threshold value .5 for example.

Consider three websites, S1, S2 and S3 which have same revisit frequency. Let $\Delta t_1$, 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:-

- Let, $t_n = 100$ units
- $P_l = 0.3$
- $P_g = 0.7$
- $P_c$ = say 0.85

$\Delta t_1$ would be computed as given below:-

$$\Delta t_1 = \{(1-0.85/0.7)*1+(1-0.85/0.3)*0\} \times 100 = -150/7 = -21.42 \text{ units}$$

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 frequency for all three websites remains 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, \( t_n = 50 \) units

\[ P_1 = 0.5 \]

\[ P_g = 0.8 \]

\[ P_c = \text{say 0.7} \]

\( \Delta t_2 \) for sites S1, S2 and S3 would be,

\[ \Delta t_2 = \{(1-0.7/0.8)*0 + (1-0.7/0.5)*0\} * 50 = 0 \text{ units} \]

The new refresh time,

\[ t_{n+1} = t_n + \Delta t_1 + \Delta t_2 \]

\[ t_{n+1} = 100 - 10.71 + 0 = 89.29 \text{ units} \]

Refresh time is decreased.

**Case 2:**

Let, \( t_n = 50 \) units

\[ P_1 = 0.5 \]

\[ P_g = 0.8 \]

\[ P_c = \text{say 0.3} \]

\( \Delta t_2 \) for sites S1, S2 and S3 would be,

\[ \Delta t_2 = \{(1-0.3/0.8)*0 + (1-0.3/0.5)*1\} * 50 = 20 \text{ units} \]

The new refresh time,

\[ t_{n+1} = t_n + \Delta t_1 + \Delta t_2 \]

\[ t_{n+1} = 100 - 10.71 + 20 = 109.29 \text{ units} \]

Refresh time is increased.
Case 3:

Let, \( t_n = 50 \) units

\( P_1 = 0.5 \)

\( P_g = 0.8 \)

\( P_c = \) say 0.9

\( \Delta t_2 \) for sites S1, S2 and S3 would be,

\( \Delta t_2 = \{(1-0.9/0.8)*1 + (1-0.9/0.5)*0\} * 50 = -6.25 \)

The new refresh time,

\( t_{n+1} = t_n + \Delta t_1 + \Delta t_2 \)

\( t_{n+1} = 100 -10.71 -6.25= 83.04 \) 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, as shown in Figure 4.3.

Figure 4.3: Computation of Refresh Time

The frequency of migration is inversely proportional to the refresh time, and in the above three cases will not be same, as shown in Figure 4.4
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 4.5.

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 4.6.
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 (see Figure 4.7).

**Figure 4.6:** Effect of Web growth on quality of pages

![Graph showing the effect of Web growth on quality of pages](image)

**Figure 4.7:** Analysis of proposed architecture (Web size vs Network traffic)

![Graph showing network traffic vs web size](image)
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 (see Figure 4.8).

![Graph showing comparison between quality and web size for proposed and traditional crawling]

**Figure 4.8**: 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.

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. This work presents that revisiting to a website 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.

Migrating crawlers (mobile processes) are capable of roaming the web, interacting with web servers that host web pages, gathering information on behalf of its owner and coming back having performed the duties set by its user. Distributed migrating crawlers minimize network utilization and also keep up with document changes. Keyword searching is the most common form of text search on the Web and, the present search engines are unable to search for the keywords based on sense behind the words. An improvement in the
index construction, in which related terms based on perception help the search engines to provide better results to the user based on his mental vision.

The next chapter presents use of migrating crawling agents in managing highly volatile web data and perception based index construction.