How Should Crawler Update The Stored Web pages In Database?

Web has become one of the basic necessities in human life since its advent. Every human in present era uses the web daily in direct or indirect form. Due to this the size of the web is increasing day by day. So for the better end user experience, search engines are using different types of web crawler at different geographical locations to maximize the coverage area and the amount of information on the web. Web crawler is a program in search engines which collects the web pages from the World Wide Web.

The process of crawling begins with the set of seed URLs having large number of outgoing links. These outgoing links are downloaded and URLs are extracted from them to be traversed further. The downloaded URLs are stored in repository .So there is need arises to keep the repository updated. This problem comes with the idea of page revisit policy to be linked with the crawling system. Page revisit policy is used to keep the freshness of the repository as high as possible. In this paper a module called freshness checker is used as the revisit module with the focused crawling system. This module is used to detect the changes in the web page. This module can detect page structural change, page content change and image change.

How should crawler update the stored web pages in database? Which revisit policy should be planned for efficient crawling system? Web crawling is a continuous process and it takes several days or even months to crawl the fraction of the web. But web is very dynamic in nature and it keeps on changing. Some web pages change very frequently within days while some are kept unchanged for several days. So to keep the collection of downloaded webpage updated they should be revisited regularly. It is very hard to synchronize page revisit policy with the crawling process as the size of the web is very large.

Ample amount of new information is updated each day on the web which results in creating new web pages or updating existing web pages. Humongous size of the web makes it difficult for the web crawler to detect these changes on regular basis. But for the better experience of the user collected URLs must be updated on regular basis. So, a separate module should be incorporated in the crawling system to detect these changes and this module is named as freshness checker module. In this paper, freshness checker module can detect the structural change, content change and image change in the web page. This module helps in maintaining the freshness of repository and also reducing the consumption of bandwidth.

Change detection of web page in Focused Crawling system

According to the studies, for a short span of time the ratio of change in web content with the total web content is significantly smaller. In authors from their study of 720,000 pages of dataset concluded that around 70 percent of the web remains unchanged for 30 days but some of the domains like .com changes very frequently. These studies show that page revisit policy should be based on domain of the web page. But there are chances that page does not change when it is revisited based on the type of domain. On the basis of such considerations, the algorithm uses a different color image multiplied by the weighting coefficients of different ways to solve the visual distortion, and by embedding the watermark, wavelet coefficients of many ways, enhance

the robustness of the watermark.

So for the efficient web crawler, need arises for a page revisit policy to be incorporated in crawling system. Some of the proposed page revisit policies are as follows:

- Revisit policy in is based on the frequency of change in content. Refresh rate of web page is computed dynamically according to which URLs are distributed among several URL queues. URLs from the queues are selected at every n2 * 20 µ sec where n (1…..n) is number assigned to the queue based on the refreshing time of URL.
- In algorithm is proposed in which page score is used to depict which URL has to visit first. Page score of the URL is calculated as the ratio of page rank with the age of the page where age of page is the difference in the present time and last modified time. Pareto's principle of 80-20 is applied in which top 20% URLs having high page score are kept in B collection and the frequency of revisiting these pages are high than that of A collection URLs.
- Authors in suggest a technique to maintain the freshness of crawled pages stored in the repository. The proposed technique encompasses compression techniques to store crawled pages. The compression utility has been found by surveying different techniques of compression for the best compression ratio, its compression efficiency is about 74.5%.
- A page update policy has been proposed in that minimizes the carbon footprints and energy consumption of servers. This is achieved by lowering the number of requests to web servers. The approach calculates the optimal policy for crawling on the basis of values of page staleness and greenness indicators.
- In web page changes are broadly classified in four major categories namely 1.Content /Semantic changes 2. Presentation/Cosmetic changes 3.Structural changes 4.Behavioral changes. The node of HTML tree structure of web page should contain ID, CHILD, PARENT, LEVEL, CONTENT VALUE information. Structural change is depicted when any new tag is created or deleted then the initial set and later set have different values. Content change is recognized by comparing the product of R.M.S value and ASCII value of the characters in text.