UGC Approved Journal

IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

ISO 3297:2007 Certified

Vol. 5, Issue 9, September 2017

Design and Development of Intellect Webbot- A Novel approach scheme and Utility of Intelligent Agent based Web Crawler

Mrs. P. Jegadeeshwari[1], Dr. D. Sengeni[2], Mrs. R. Nithya[3]

Assistant Professor, CK College of Engineering and Technology, Cuddalore[1, 3]

Associate Professor, CK College of Engineering and Technology, Cuddalore[2]

Abstract: To completely crawl the World Wide Web, web crawler takes more than a week period of time. This paper focuses on role of agents in providing intelligent crawling over the web. The role of building a proxy server at application level is clearly discussed. The web pages need to be cached for providing better response time. Within this time, there are changes occurred to various pages, so it cannot always be able to provide the updated content to the use. Intellect Webbot will reduce the latency taken for search results by enabling various agents, providing more updated links to the user, also the adeptness to view users' bookmarks anywhere through our system. Moreover, this system has distributed intelligent agents, which is used to index the web pages in the server with the updated information. The actual scenario is the user going to give the keyword in terms of query to this system. The system contains several agents such as Link repository agent, Regional crawler agent, link maintenance agent, and bookmark agent. The results from this system are the list of URLs along with description about that page. The link in the result page is called context link. Forming the context link, based on the user given keyword and the related link that are available in the link repository, should be made and that would be included in the result page as a list of context links. Unlike other search engines, crawler provides context links to the user, according to the user's pursuit. This work is accomplished by storing the users' name along with their search history in the server. The dynamic web cache management scheme is being tested across 30 nodes and its results are discussed. The proposed intelligent crawler is compared with LLI and dynamic web cache scheme and results are discussed. The results achieved from these experiments confirm the efficiency and adaptability of the proposed crawler.

Keywords: Intellect Webbot, World Wide Web, Web Crawler, Intelligent Agent.

I. INTRODUCTION

The number of web pages available in the internet is tremendously growing day to day and since this is the case, searching relevant information in the internet is hark task. Usually, searching information in the World Wide Web can be done by searching the list of links crawled and sorted based on type of the web page or contents of the web page. Today, the main problems of search engines are the size and the rate of change in the web page daily. Web crawler takes approximately more than a week to crawl all the web pages in internet once. The changes to the web pages are very frequently occurred today, so providing more relevant information for a specific request by a search engine is a challenging issue.

Intellect Webbot will overcome from above mentioned problems. Each agent will perform its own task in order to crawl the web information. The user needs to give their keyword for search, the crawler will search the links related with the help of link resolver agent to identify the Universal Resource Locators available in the link repository. Intellect Webbot helps to crawl the pages available in the internet using distributed intelligent agents, which is used to index the web pages in the server with the updated information and produce the effective list of links as a result based on quality metrics such as speed, quality information, or related information.

A crawler is a program that retrieves and stores pages from the Web, commonly for a Web search engine. A crawler often has to download hundreds of millions of pages in a short period of time and has to constantly monitor and refresh the downloaded pages [3]. Context of a hyperlink or link context is defined as the terms that appear in the text around a hyperlink within a Web page. Link contexts have been applied to retrieval and categorization tasks of web information. Topical or focused Web crawlers have a special reliance on link contexts [2]. Crawling the web using a single node should be a very difficult task and time consuming, so the updated content of the web cannot be displayed while searching through search engines. Enabling the agent based crawler distributed at various locations. So the crawling process will be exercised in faster manner. Finally, agents will coordinate and integrate the results from various locations.

UGC Approved Journal
IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2017

II. EXISTING METHODS

Latent Linkage Algorithm (LLI) can be used for efficiently retrieving web pages with lesser response time [17]. The system can be built an efficient hyper link-based Algorithm to find the relevant links for a given web page (URL). The algorithm is advantaged with linear Algebra Theories to reveal deeper relationship among the web page to identify relevant links more precisely and efficiently. The algorithm presented is supposed to be the best and effective to enhance web searches and the techniques implemented are best supported and recommended for web-related researches.

Whenever the client request the web page the request will be forwarded to the server. The server will forward it to the web server via ISP and DNS. The response from the web server will be stored as cookie in the web browser for later use. An HTTP cookie is a packet of information sent by a web server to the web browser and then sent back by the browser each time it accesses that server. The existing server's uses data structures And BLOB data base management system. BLOB database is used to store massive volume of data. Some databases require you to store large amounts of data as a BLOB rather than text. The advantages of this technique are that it's extremely simple to dump the data into the table and equally simple to extract it back out. There are no keys to manage for this table. Some major drawbacks are that you probably won't be able to do any useful text a unique searching, and you may have difficulties locating a specific document since there's nothing to identify document within the table.

III. PROPOSED SYSTEM

In the proposed system we implement a proxy server, which makes use of randomized algorithm that combines the benefit of both RR scheme and utility function. This avoids the need for data structure. RR scheme is used to evict the documents randomly. The utility function assigns to each page a value based on-recentness of use-frequency of use-size of page-cost of fetching. RR scheme would replace the least recently used web documents.

The second step is to build a intelligent webrobot using agents for different functionalities. We have implemented LLI algorithm and a model for dynamic web caching and compared the results with proposed intelligent webrobot.

IV. INTELLECT WEBBOT: ANALYSIS & DESIGN

The architecture of web crawler should be broadly divided into two architectures, one is cyclic architecture as in Fig.1 and another one is detailed architecture as in Fig.2. First step is the collecting pages from the World Wide Web, and then indexing all the pages. If any search keyword given, crawler will be able to give a list of relevant links by searching in this index. The followingFig.1 clearly explicates the architecture.

Initially, crawler collects or indexes all the pages from World Wide Web, based on the downloaded information, crawler prepares database called link repository. Repository has the type of web page with corresponding link identification number. If a user request some information related to a specific topic crawler will search based topic given by the user. Then it indexes all the pages as a links based on the ranking score to the particular link.

In this system user-specific results are exercised by sustaining the user's details along with their search history. The scenario is that the when user enters the keyword for search, the system first checks in user's cache for result. If result is not sufficient for the user, it will get results from link repository. Fig.3 clearly depicts the working of Intellect Webbot.

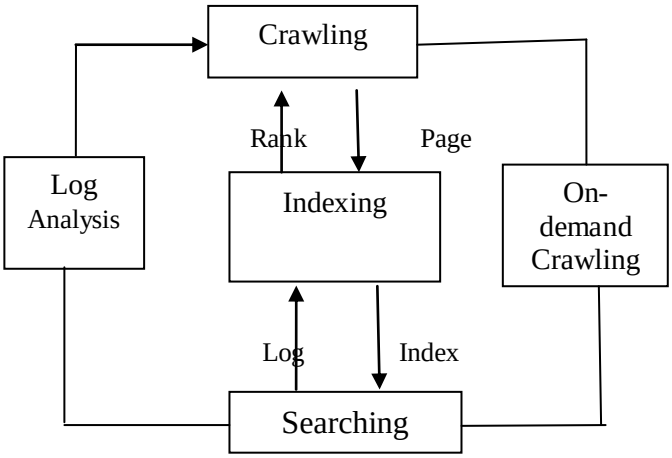

Figure 1: Cyclic Architecture For Search Engine

UGC Approved Journal
IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2017

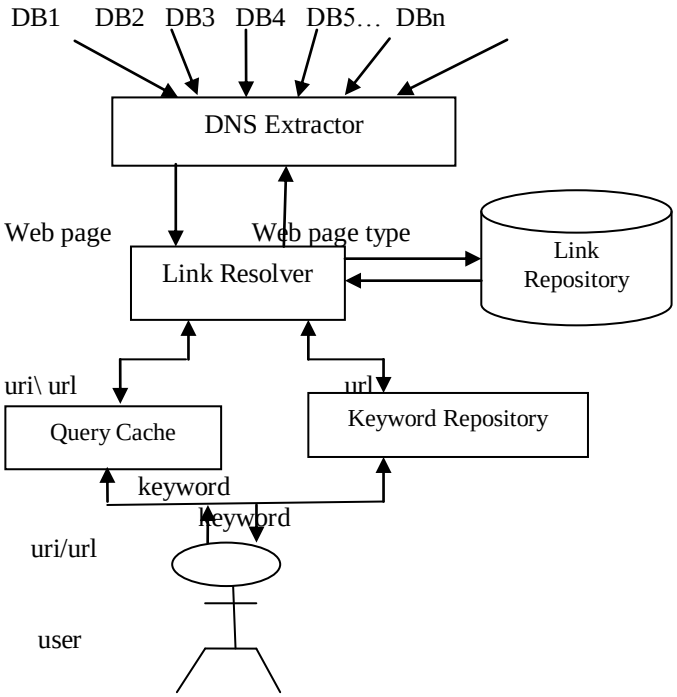

Figure 2: Detailed Architecture of Web Crawler

V. DIFFICULTIES IN CRAWLING

There are two important characteristics of the Web that generate a scenario in which web crawling is very difficult: its large volume and its rate of change. The large volume implies that the crawler can only index a fraction of the Web pages within a given time, so it needs to prioritize work. The high rate of change implies that by the time the crawler is indexing the last pages from a site, it is very likely that new pages have been added to the site, or that pages have already been updated or even deleted. Here, user gives the keyword in order to search the web. But with this system we make the crawler to first search in the cache and then the link repository so as to reduce the latency.
Based on the link id crawler will retrieve the full name and description of the web page with the help of DNS Extractor. Also, it will maintain the keyword along with the resulted list of links in the Query cache for future use; this is primarily used for fast access.

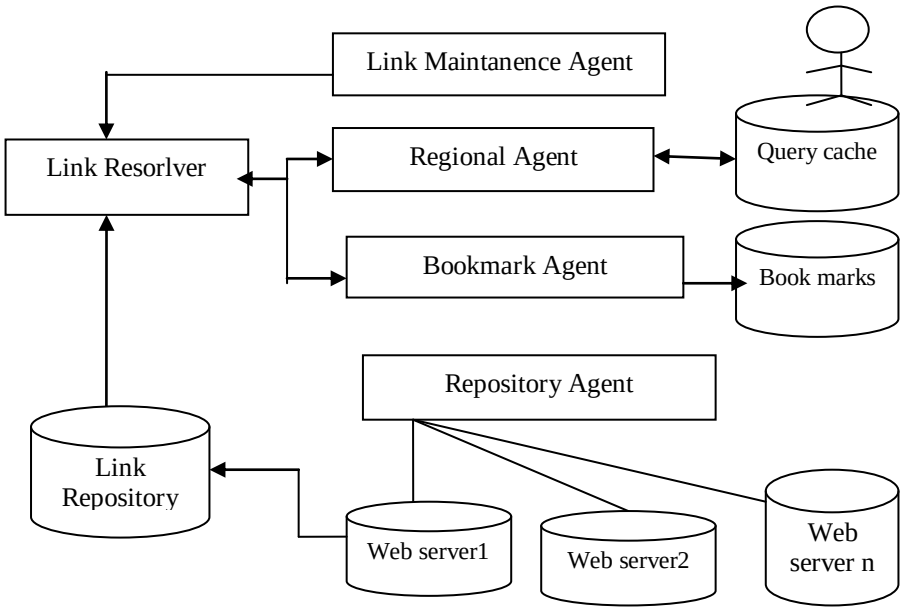

Figure 3: Intellet Webbot Architecture

UGC Approved Journal
IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2017

VI. INTELLEECT WEBBOT ARCHITECTURE

The architecture contains four agents such as link repository agent, Regional crawler agent, Link maintenance agent, and bookmark agent. Figure 3 depicts the proposed architecture; double lined rectangle denotes the proposed agents in our architecture.

Role of Agents
The role of agents is to provide the faster indexing of web pages in link repository by implementing distributed web crawlers. By enabling agent control we store the user search history and bookmark results. Moreover, this system sustains the link updation often as s usage a separate agent for this purpose is exercised here.

Link Repository Agent
This agent runs on the server, which collects the list of keywords from WebPages stored in webserver and indexes it in the link repository. This will have the list of keyword s and corresponding URL. It ignores the unwanted phrases and prepositions, ect while accumulating the keywords. Whenever a user gives a search request, keywords will be matched in link repository and corresponding URLs will be sent to the user.

Regional Crawler Agent
The regional crawler agent is responsible for the grouping of the recently searched keywords and the corresponding URLs for a particular user. These details will be updated in the server whenever the user logs out of our system. Once the user enters a keyword and initiates search operation, the request is sent to the cache. The matching process will be done in cache to list the recently fetched results. If not found, the request is sent to link repository to list the results After returning the results the user log is updated with recently searched information. The cache is maintained by using the LRU algorithm.

Link Maintenance Agent
The Link Maintenance Agent IS RESPONSIBLE FOR THE PERIODIC UPDATION PROCESS. It will periodically check for updates in the web server and sustains the update on links lost and links redirected. It will send the links present in the cache to the server and stores the updated content in the user log. Depending upon the working environment and the bandwidth utilization factor the threshold period is to be set.

Bookmark Agent
The bookmark agent is responsible for maintaining the bookmarked URLs by the user. The bookmarked URLs are stored corresponding with the username in the server. Whenever the user gets the resultant links, user can bookmark the URL and the user can access the bookmarked URLs anywhere through our system.

CRAWING POLICIES
The behavior of a web crawler is the outcome of a combination of the following policies:

- A selection policy that states which pages to download.
- A re-visit policy that states when to check for changes to the pages.
- A politeness policy that states how to avoid overload websites.
- A parallelization policy that states how to coordinate distributed web crawlers.

Selection policy
As the size of the web is large, even very famous and large search engine covers the portion of the internet. Not even one search engine searches more than 16% of web content. It is highly desirable that, from that crawled portion, finding relevant information. A good selection policy is required in order to do the best crawling task.

This requires a metric of importance for prioritizing Web pages. The importance of a page is a function of its intrinsic quality, its popularity in terms of links or visits, and even of its URL. Designing a good selection policy has an added difficulty: it must work with partial information, as the complete set of Web pages is not known during crawling.

Cho et al. [11] Made the first study on policies for crawling scheduling. Their data set was a 180,000-pages crawl from the Stanford.edu domain, in which a crawling simulation was done with different strategies. The ordering metrics tested were breadth-first, backlink- count and partial page rank. One of the conclusions was that if the crawler wants to download pages with high Pagerank early during the crawling process, then the partial Pagerank strategy is the better, followed by breadth-first and backlink-count. However, these results are for just a single domain.
Najork and Wiener[12] performed an actual crawl on 328 million pages, using breadth-first ordering. They found that a breadth-first crawl captures pages with high Pagerank early in the crawl. The explanation given by the authors for this

UGC Approved Journal
IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2017

result is that "the most important pages have many links to them from numerous hosts, and those links will be found early, regardless of on which host or page the crawl originates.

Abiteboul et al.[16] designed a crawling strategy based on an algorithm called OPIC(On-line Page Importance Computation). In OPIC, each page is given an initial sum of "cash" which is distributed equally among the pages it points to. It is similar to a Pagerank computation, but it is faster and is only done in one step. An OPIC-driven crawler downloads first the pages in the crawling frontier with higher amounts of "cash". Experiments were carried in a 100,000-pages synthetic graph with a power-law distribution of in-links. However, there was no comparison with other strategies nor experiments in the real Web.

Boldi et al[14] used simulation on subsets of the Web of40 million from the domain and 100 million pages from the WebBase crawl , testing breadth-first against random ordering and an omniscient strategy. The winning strategy was breadth-first, although a random ordering also performed surprisingly well One problem is that the WebBase crawl is biased to the crawler used to gather the data They also showed how bad PageRank calculations carried on partial subgraphs of the Web, obtained during crawling, can approximate the actual Pagerank

The importance of a page for a crawler can also be expressed as a function of the similarity of a page to given query. This is called "focused crawling". The main problem in focused crawling is that in the context of a Web crawler, we would like to be able to predict the similarity of the text of a given page to the query before actually downloading the page. A possible predictor is the anchor text of links; this was the approach taken by pinkerton in a crawler developed in the early days of the Web. To use the complete content of the pages already visited to infer the similarity between the driving query and the pages that have not been visited yet. The performance of a focused crawling depends mostly on the richness of links in the specific topic being searched, and crawling usually relies on a general Web search engine for providing starting points.

Re-visit policy
The Web has a very dynamic nature, and crawling a fraction of the Web can take a long time, usually measured in weeks or months. By the time a Web crawler has finished its crawl, many events could have happened. The events are characterized as creations, updates and deletions.

Creations: When a page is created , it will not be visible on the public Web space until it is linked, so we assume that at least one page update-adding a link to the new Web page-must occur for a Web page creation to be visible. A Web crawler starts with a set of starting URLs, usually a list of domain names, so registering a domain name can be seen as the act of creating URL. Also, under some schemes of cooperation the Web server could provide a list of URLs WITs without the need of a link.

Updates: Page changes are difficult to characterize: an update can be either minor, or major. An update is minor if it is at the paragraph or sentence level, so the page is semantically almost the same and references to its content are still valid. On the contrary, in te case of a major update, all references to its content are not valid anymore. It is customary to consider any update as major, as it is difficult to judge automatically if the page's content is semantically the same.

Deletions: A page is deleted if it is removed from the public Web, or if all the links to that page are removed. Note that even if all the links to a page are removed, the page is no longer visible in the Web site, but it will still be visible by the Web crawler. It is almost impossible to detect that a page has lost all its links, as the Web crawler can never tell if links to the target page are not present, or if they are only present in pages that have not been crawled Undetected deletions are more damaging for a search engine's reputation than updates, as they are more evident to the user. Search engine performance reports that on average 53% of the links returned by search engines point to deleted pages.

Cost functions: From the search engine's point of view, there is a cost associated with not detecting an event, and thus having an outdated copy of a resource. The most used cost functions are freshness and age.

Freshness: This is a binary measure that indicates whether the local copy is accurate or not.
Age: This is a measure that indicates how outdated the local copy.

Strategies: The objective of the crawler is to keep the average freshness of pages in its collection as high as possibly, or to keep the average age of pages as low as possible. These objective are not equivalent: in the first case, the crawler is just concerned with how many pages are out-dated, while in the second case, the crawler is concerned with how old the copies of pages are.

UGC Approved Journal
IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2017

Two simple re-visiting policies

(a) **Uniform policy**: This involves re-visiting all pages in the collection with the same frequency, regardless of their rates of changes

(b) **Proportional policy:** This involves re-visiting more often the pages that change same frequently. The visiting frequency is directly proportional to the change frequency.

VII. IMPLEMENTATION

Using java and swing concept, the working model for our paper has been developed. In the working model, it is assumed there are two or more web servers present

Link repository agent crawls each and every web page, upholds the keywords and store it in link repository

Regional crawler agent groups the recently searched keywords and the corresponding URLs for a particular user. The table is maintained by using the LRU algorithm.

Link maintenance agent does the periodic updation process. It will periodically check for updates in the WebServer and sustains the update on links redirected in link repository as well as user log.

Bookmark agent maintains the bookmarked URLs by the user. The bookmarked URLs are stored corresponding with the username in the server. Whenever the user clicks the bookmark button, that particular URL is stored in bookmark table.

VIII. RESULTS AND DISCUSSIONS

The proposed system was tested with a test bed of 30 users trying to access similar links:

The system utilizes the bandwidth and directs request and retrieves the WebPages using Proxy server software that is designed for use. Using the proposed intelligent crawler the response time for accessing keywords in WebPages is listed below:

Keyword	server response in time(ms)	cache response	Web site
Vista	203	16	www.microsoft.com
Yahoo	4.56	26	www.yahoo.com
Gmail	4.2	16	www.google .com
Vlb	2.4	27	www.vlb.edu
Infoline	2.2	100	Indiainfoline.com
Cricket	3.5	120	Cricinfocom

IX. FUTURE WORK

This paper describes a new architecture for agent based intelligent crawler In order to clearly explain the architecture, working model for Intellect Webbot architecture has also been shown. Implementing any crawler in a real time is very tough task than the designing work. It requires getting various permissions the owners of the web servers administrators. It needs huge storable node for keeping the details of the user and the URLs.

REFERENCES

[1] Alex Homer, Dave Sussman, Rob Howard, Brian Francies, KARLI Watson and Richard Anderson, Professional ASP.NET 1.1 FROM Wrox publishing.

[2] Birukov. A,Enrico Blanzieri and paolo Giorgini, Implict: An Agent Based Recommendation System john wiley & Sons,1999.