ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

Web Crawling Using Dynamic IP Address Using Single Server

Prabhat Kumar[1], **Niraj Singhal**[2]

M.Tech. Scholar, Shobhit University, Meerut, India[1]

Associate Professor, Shobhit University, Meerut, India[2]

Abstract: The major issue with web crawler is parallel execution of multiple requests using an individual IP address which can be easily traced by the website if multiple hits are observed from the same IP address. To overcome this, proxy servers are used which include discrete proxy IP address whenever the request is routed over the crawling server. Now multiple requests can be executed using same crawling server, to download the information from the same website without any blockage. Each request would contain one URL and one distinct proxy IP that routed to the crawling server by the request router. Now the routed request can fetch the information very fast with the parallel execution of requests by concealing identity from the tracking application. Data extracted from the website is stored into temporary database and Indexing is performed after the each URL seed receives complete response. The context of the documents collected by the crawling in the repository is extracted by the indexer using the context repository, thesaurus, repository, and documents are indexed according to their respective context.

Keywords: Crawling Servers, Discrete Proxy, Proxy Server, Web Crawler

I. INTRODUCTION

The web is continuously expanding with large collection of hypertext documents. Era of web has become in few years into the largest cultural endeavour of all times. The web is a distributed repository of information without a central point of control and thus can be seen as a vast, diverse, rapidly changing and unstructured database. The low cost of publishing information on the web is a key part of its success but implies that the searching of information on the web will always be inherently more difficult that searching information in local physical storage. It involves millions of detached websites and works on client-server based architecture that allows a user to set-off search by giving the keywords to any of the search engine that collects and reverts with the required results from the web.

A web search engine takes a user need, usually stated in the form of a few keywords, and returns a list of web pages that can be considered relevant for the given keywords. These web pages are a short list of usually few hundred items selected from a vast repository of thousands of millions of pages. The best part of web based searching is to find which documents in the web are related to a given query, because most queries are very broad, and there are thousands of pages relevant to most basic concepts. Search engine not only provide the most related document also it performs indexing on the web pages they have been collected into a repository. This process is similar to content mining process where the most relevant content is filtered from the cluster of documents collected into the repository.

The web can be considered as divided into two parts, the closed web and open web. Closed web comprises few high-quality controlled collections on which a search engine can fully trust. Open web comprises the vast majority of web pages, which lack an authority asserting their quality also the traditional information retrieval techniques, concepts and methods are the major challenges. Web crawling is a process by which a search engine gathers pages from the web, index them to fulfil user's query. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them. The main aim is to select the best collection of information in minimum possible time. The major issue with web crawler is parallel execution of multiple requests using an individual IP address which can be easily traced by the website if multiple hits are observed from the same IP address. In the proposed approach, proxy servers are used which include discrete proxy IP address whenever the request is routed over the crawling server. It allows execution of multiple requests using same crawling server.

II. WEB SEARCH ENGINES AND WEB CRAWLERS

A web search engine [1, 2, 9, 12] periodically downloads web pages using web crawler and indexes a sub-set of those. This index is used for searching and ranking in response to user queries. In modern web search engines, this view is extended with extra information concerning word frequencies and text formatting attributes, as well as metainformation about web pages including embedded descriptions and explicit keywords in the HTML mark-up.

Web crawler (also called web spider, bots, wanderers, web robots) is a computer program that is used to extract information from the web in systematic, methodical, automated manner. A web crawler (see figure 1) downloads the given URLs from the web, it extracts any hyperlinks contained in them and add them to the list of URLs to be crawled. It fetch the pages associated with these URLs, recursively continues to download the web pages identified by these hyperlinks.

Web crawlers are used in many other applications that process large number of web pages, such as web data mining, comparison shopping engine etc. The web search process has two main parts: off-line and on-line. The off-line part is executed periodically by the search engine, and consists in downloading a sub-set of the Web to build a collection of pages, which is then transformed into a searchable index. The on-line part is executed every time a user query is executed, and uses the index to select some candidate documents that are sorted according to estimation on how relevant they are for the user's need.

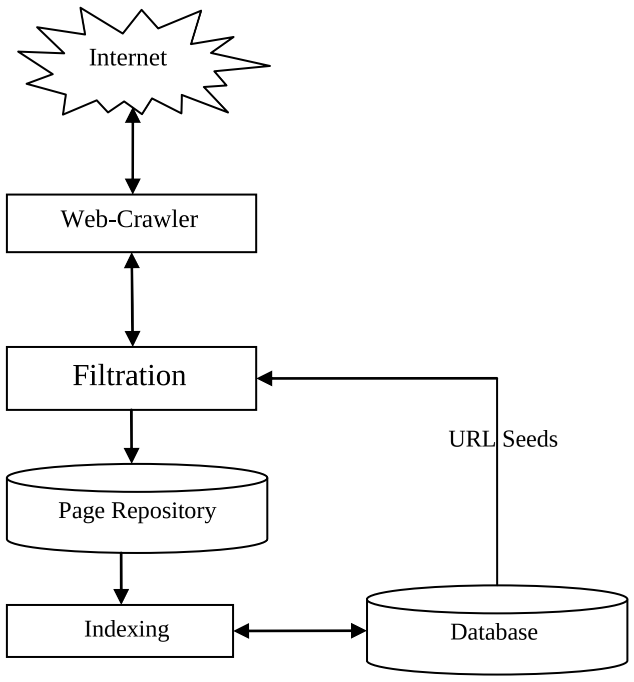

Figure 1. General architecture of a web crawler

III.RELATED WORK

Many researchers have looked at web search technology over the last few years, including crawling strategies, storage, indexing, ranking techniques and a significant amount of work on the structural analysis of the web and web graph. See [4] for overviews of some recent work and [5] for basic techniques. There has been some recent academic interest in the first issue, including work on strategies for crawling important pages first [6], crawling pages on a particular topic or of a particular type, re-crawling (refreshing) pages in order to optimize the overall "freshness" of a collection of pages, scheduling of crawling activity over time. The enormous size of the content on the web, it's very fast expansion rate, and the high frequency of page modifications render the web crawling problem as one of the biggest challenges in search engine design [2]. Most of the recent work on crawling strategies attempts to minimize the number of pages that need to be downloaded, or maximize the benefit obtained per downloaded page.

An exception is the work in [3] that considers the system performance of a focused crawler built on top of a general purpose database system, although the throughput of that system is still significantly below that of a high-performance bulk crawler. For proxy based web crawling of huge data many algorithms and techniques have already been proposed to completely crawl the web, crawler takes more than a week period of time. As the growth of the web exceeded all expectations, the research on web mining is growing more and more. Web mining research combines two of the activated research areas i.e. data mining and web mining. The web mining is a very advanced area for data mining research. Search engines that are based on web crawling framework also used in web mining to find the interacted web pages.

Proxy IP is used for faster crawling of web [7,8]. These also discuss 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 user. Intellect web bot 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 i.e., the text around a hyperlink within a web page. 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 users' pursuit. This work is accomplished by storing the users' name along with their search history in the server. For web crawling and indexing many algorithms has already been proposed. Cho and

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

Molina [10] gives two policies i.e. dynamic assignment and static assignment. Dynamic assignment allows seeds URLs to different server by which it could manage the load by distributing the URLs to different crawling servers. Here a central server is used which to transfer the workload among the distributed crawling servers for larger crawling. With this policy central system can add or remove the crawler from downloading processes accordingly. The two new configuration of dynamic assignment is further introduced by Shkapenyuk and Suel [10], in which a central DNS and central queues for each website is provided for smaller crawling processes while for larger crawling processes a DNS resolver and queues are distributed. Static assignment is a method in which a hashing function is used to transform URLs into a number that relates to the index of the corresponding crawling process. As there are external links that will go from a web site assigned to one crawling process to a website assigned to a different crawling process, some exchange of URLs must occur.

To reduce the overhead due to the exchange of URLs between crawling processes, the exchange should be done in batch, several URLs at a time, and the most cited URLs in the collection should be known by all crawling processes before the crawl. An effective assignment function must have three main properties: each crawling process should get approximately the same number of hosts (balancing property), if the number of crawling processes grows, the number of hosts assigned to each process must shrink (contra-variance property), and the assignment must be able to add and remove crawling processes dynamically.

The next section presents a novel approach that uses discrete proxy IP address whenever the request is routed over the crawling server. Now multiple requests can be executed using same crawling server, to download the information from the same website without any blockage. Each request would contain one URL and one distinct proxy IP that routed to the crawling server by the request router. Now the routed request can fetch the information very fast with the parallel execution of requests by concealing identity from the tracking application. Data extracted from the website is stored into temporary database and Indexing is performed after the each URL seed receives complete response. The context of the documents collected by the crawling in the repository is extracted by the indexer using the context repository, thesaurus, repository, and documents are indexed according to their respective context.

IV. PROPOSED WORK

Crawling process (see figure 2) is carried out in three phases .i.e. Application server generates, loads and sends request to the BOT framework. It starts with request generation at the application server where the request is generated. In second phase, a random proxy is selected and inserted in executing request such that data is crawled from the website without being detected and stored over the crawling server in repository; here the filtration is performed by the framework based on the context. In third phase two services works correspondingly for data transporting, data is transported to the physical storage using the response sending service at crawling server and response receiving service at centralized database receive the data and stored into physical storage

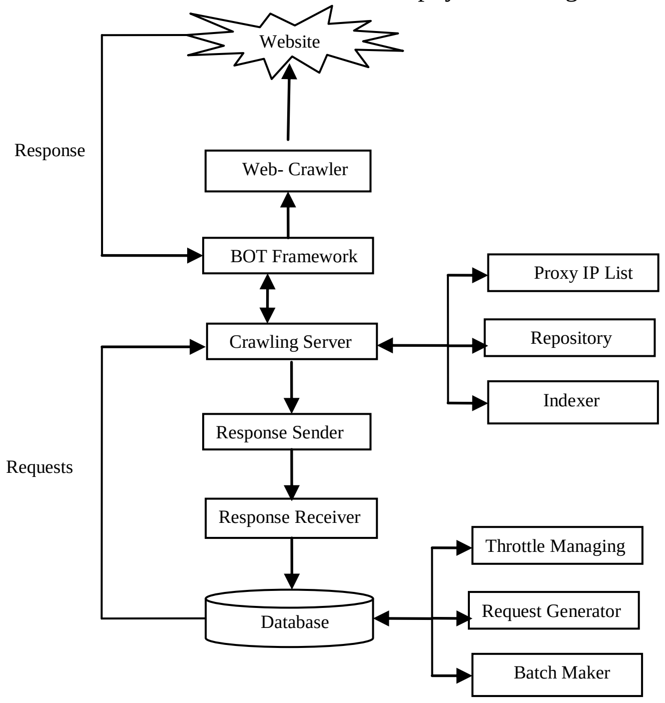

Figure 2. Architecture of web crawling using multiple IP Address

Various components of proposed crawling system are as follows:

❖ **Crawler:** It is stated as a computer program dedicated to download Hypertext Markup Language (HTML) or Extendable Mark-up language (XML) pages from the internet.

❖ **Bot Framework:** Each time bot downloads an HTML page; framework takes it and extracts its contents. Framework filters the resulting text and converts them into a suitable format that can be suitable for indexing and moves to the storage.

❖ **Indexer:** When the crawling process starts, each time a return the response indexer took the responses from the framework in an appreciate structure and provide a distinct key called index value

❖ **Page Repository:** It is temporary storage of the web pages being crawled and forward to the framework which filtration of the required information done.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

❖ **Flexibility:** As mentioned, one would be able to use the system in a variety of scenarios, with as few modifications as possible.

❖ **Centralized Database:** A centralized database is used to maintain all the data in one place, main role of centralized database is to provide the connectivity and facilities to the various resources.

❖ **Throttling:** It is important not to put too much load on a single web server, by observing a time-out period between requests. It is also desirable to do domain-based throttling, to make sure that requests are balanced between different second- or higher-level domains. There are several motivations for this. Some domains may have a fairly slow network connection, but large number of web servers, and could be impacted by a crawler

Algorithm for the proposed system is as follows:

❖ **Step-1:** Request Generator application works at the application server, it is responsible to check that whenever the crawling is initialized it generates the requests on the basis of requested parameter. Batch maker application also run at the application server and is responsible for making bunch of the generated requests.

❖ **Step-2**: Routing server is used to distribute the load at the multiple servers during the data extraction. While we are using the single server for extraction. At crawling server throttle managing service is used to manage the number of maximum request can be executed at a time based on the capacity of the website. It is done by analysing the load that a website can accept at a time

❖ **Step-3**: Request Puller is an application which accepts Request to be crawled on websites and handle accordingly and it is the same application which runs in infinite loop and dedicated to accept the request from central db to crawling database according their status provided.

❖ **Step-4:** Proxy manager is a web service which consists of available proxy IP and it includes individual Proxy IP address with each executing request.

❖ **Step-5:** Request processor takes the request from the crawling server and initiates the bot framework which loads the DLL for the respective website and start collecting data.

❖ **Step-6:** With receiving the response from the Internet, data is stored into the repository which is already deployed at crawling server. Data messaging and filtering is done on available at repository for faster query response for better retrieval from database

❖ **Step-7:** Send data Application is responsible for sending the data from repository to the physical storage in XML or HTML format.

❖ **Step-8:** This crawling process last until all seed URLs is visited and data for the same is stored into database.

VI.IMPLEMENTATION AND RESULTS

This implementation is to provide faster and dedicated crawling for larger data users and presented some preliminary experiments. Due to the parallel request acceptance capability of the website multiple request execution is expected to provide better utilization of resources compared to a single (sequential) request execution implementation.

To implement this C# language has been used as it support multithreading environment. Programming language such as java or c# both support multithreading which is a major part in this implementation, as a thread is a sequential flow of control within a program, a program or process can have multiple threads running concurrently all of which may share data or run independently while performing tasks. The multithreaded implementation allows one thread to use the network connection to retrieve documents, while another thread can use the CPU, and another can access cache information on the local disk. Threads are useful because that they allow to perform multiple tasks at once in a single program or process. Advantages of threads include exploiting parallelism in multiprocessor architectures, reducing execution time by being able to perform tasks while waiting on a blocking system calls such as I/O operation, and avoiding freezing.

This process is aid to improve the larger crawling from the dedicated website by making an illusion to the website that that IP hit are coming from distinct IP and is not doing by any program. By analysing the behaviour and maximum hit limit of the website, one can setup a static throttle value (No of Hit/Min) and using specific proxy IP address with each executing request to make the crawling work faster and smoothly, now the request is getting initiated using single server and can be executed without being traced-up by the tracking tool used by the websites. As each request is including different IP address which means that these are not coming from the same system/machine. By this strategy crawling process will continue longer and faster.

Table 1 shows a log file of the website, here the max hits allowed limit was analysed before and on the basis of maximum request execution limit of the website. Throttle value is analysed and assigned; this value will control the execution of concurrent request w.r.t time. In this implementation 7 distinct Proxy IP has been used which belongs to different countries servers. As the hit count performed by each Proxy IP address is lesser than the maximum hit limit allowed limit such that no blockage is done and the entire crawling get completed using same crawling server but different Proxy IP. Benefit of this implementation comes out with faster and cost efficient crawling by using single crawling server. In the below (Table 1) the max hit allowed limit is 50, So by using 7 distinct Proxy IP and single crawling server and we have

performed 7 times more crawling of the maximum allowed limit per minute.
Maximum open Limit= Number of Crawling Server * Number of Proxy IP *Throttle Limit

IP Address	Country	Time	Throttle Limit/Min	Max Hit Allowed/Min	Status
192.xxx.x.123	IN	2013-10-02 00:00:01.000	30	50	Open
192.xxx.x.10	IN	2013-10-02 00:00:01.000	30	50	Open
193.xxx.x.55	BR	2013-10-02 00:00:01.000	30	50	Open
182.xxx.x.190	FR	2013-10-02 00:00:01.000	30	50	Open
192.xxx.x.103	UK	2013-10-02 00:00:01.000	30	50	Open
182.xxx.x.83	CA	2013-10-02 00:00:01.000	30	50	Open
132.xxx.x.77	BR	2013-10-02 00:00:01.000	30	50	Open

Table 1. Log file of IP tracking tool

VII. CONCLUSION AND FUTURE SCOPE

The paper presents solution to the issue caused due to parallel execution of multiple requests using an individual IP address. The architecture and implementation details of proposed crawling system are discussed along with preliminary results. It presents complete crawling process by using the proxy server for better optimization of crawling process and tried to escape the IP blockages process used by several website to restrict the large crawling. This process is an aid to improve the larger crawling from the dedicated website by making an illusion to them that the requests are coming from different servers and are diverse .This process provides faster and dedicated crawling for larger data users.

REFERENCES

[1] Franklin, curt, "How Internet Search Engines Work", www.howstuffworks.com, 2002.
[2] D. Eichmann, "The rbse spider–balancing effective search against web load", Proceedings of First International Conference on World Wide Web, pp. 113–120, 1994.
[3] Fabrizio Silvestri, Raffaele Perego and Salvatore Orlando, "Assigning Document Identifiers to Enhance Compressibility of Web Search Engines Indexes", Proceedings of SAC, 2004.
[4] K. Bharat, A. Broder, M. Henzinger, P. Kumar, and S. Venkatasubramanian,"The connectivity server: Fast access to linkage information on the web", Proceedings of 7th International World Wide Web Conference, pp. 43-49, May 1998
[5] R. Baeza-Yates and B. Rebeiro-Neto, "Modern Information Retrieval", Addision Wesley, 1999.
[6] J. Cho, H. Garcia-Molina, and L. Page, "Efficient crawling through url ordering", Proceedings of 7th International World Wide Web Conference, May 1998.
[7] Shailesh A. Patel, Dr. Jayesh M. Patel, "Web Crawler: An Intelligent Agent through Intellect Webbot", Proceedings of Indian Journal of Applied Research, Sep 2012.
[8] Subhendu Kumar Pani, Deepak Mohapatra, and Bikram Keshari Ratha, "Integration of Web mining and web crawler
relevance and State of Art", International Journal on Computer Science and Engineering, March 2010.
[9] Niraj Singhal, Ashutosh Dixit, and A. K. Sharma, "Design of A Priority Based Frequency Regulated Incremental Crawler", International Journal of Computer Applications, Vol. 1, No. 1, Article 8, pp. 47-52, 2010.
[10] Junghoo Cho and Hector Garcia-Molina,"Distributed web crawling", Proceedings of http://www.en.wikipedia.org (2002).

BIOGRAPHIES

Prabhat Kumar received his B.Tech. in Computer Science and Engineering from Uttarakhand Technical University, Dehradun and now pursuing his M.Tech. in Computer engineering from Shobhit University, Meerut. His area of interests includes Data Structure and Network Security. His current research focus is Web information retrieval.

Dr. Niraj Singhal received his M.Tech. in Computer Engineering (first class with honours) and Ph.D. in Computer Engineering and Information Technology from Shobhit University Meerut in 2009 and 2013 respectively. He is the member of several International and National bodies and, reviewer and member of advisory board for several International and National journals. He has guided several students at postgraduate level for their research work. He has authored two handbooks with laboratory manuals for undergraduate students and a series of computer science books for school students as per CBSE/NCERT guidelines. He has more than forty research publications to his credit in National and International journals/conferences of repute. Presently he is working as Associate Professor in the Faculty of Electronics, Informatics and Computer Engineering, Shobhit University, Meerut. His area of interest includes system software, network technologies and web information retrieval.