An Ahp Based Server Selection Scheme For Migrating Crawler

[1]Ashlesha Gupta, [2]Manvi , [3]Amita

[1]Computer Engineering Department
J.C. Bose University of Science & Technology, Faridabad, India gupta_ashlesha@yahoo.co.in
[2]Computer Applications Department
J.C. Bose University of Science & Technology, Faridabad, India
manvi.siwach@gmail.com
[3]Computer Engineering Department
J.C. Bose University of Science & Technology, Faridabad, India
amita.arora@gmail.com

Article History: Received: 13 March 2020; Revised: 01 August 2020; Accepted: 05 August 2020; Published online: 28 August 2020

Abstract: Migrating crawlers are able to move to the resource which needs to be accessed in order to take advantage of local data access however this process incurs migration cost and have security risks. Therefore to increase the efficiency the crawler must be able to select the best possible server out of multiple copies of a web page present on WWW. In this work a novel credit based server selection mechanism has been proposed for appropriate server selection which considers network parameters like Network distance, Bandwidth availability, and existent load on the server to select a particular server for migrant. This technique provides enhanced security, minimum communication cost and efficient use of network resources.

Keywords: Migrating Crawler, AHP, Server-Selection, Efficient, Network resources

I. INTRODUCTION

WWW has evolved as an imperative information resource where tremendous data is available. Search engines act as an interface to access this huge information from WWW. The major components of the search engine are Crawler, Indexer and Page Ranker [1]. The crawler traverses the web, fetches documents and hyperlinks on web pages and stores them in a local repository [2]. In traditional crawling, the pages from all over the web are brought to the search engine side and then processed resulting in a lot of network traffic .A typical single process crawler however cannot keep pace with the rapid growth and dynamic nature of the World Wide Web necessitating the need to parallelize the crawling process. Since the parallel crawling instances run on a central machine, there are possibilities of overloading the system and may result in single point of failure. Therefore capabilities of mobile agents were utilized to develop migrating crawlers, where crawling instances called *migrants,* were moved to the data sources and all crawling tasks were performed at the web server itself. By migrating to the web server, the migrants download data much faster than from across the network because all the operations are being performed in the proximity of the data itself. Furthermore Migrating crawlers also support better network utilization, reliability and scalability.

In order to download data efficiently and quickly, Migrating crawlers [3] select the server which is at the least network distance for dispatching the migrants. However if the server to which the migrant is dispatched is already overloaded, assignment of new URLs to the same server will introduce network delays. Therefore, apart from considering network distance as a parameter for server selection, other factors need to be considered also like existent load, available bandwidth and security parameters so that the cost of migration can be compensated by network performance gains. In this paper an efficient AHP Based Credit based Server Selection technique is proposed, which is responsible for selecting appropriate servers based on the credit values of the servers. The proposed technique ensures that Servers with highest priority value are selected for migration and that the useful information is downloaded efficiently in minimum time.
The rest of the paper is organized as follows: Section II provides an insight to Existing Server Selection Schemes. Section III discusses the proposed algorithm and illustrates the working of the same.
Experimental set-up and results are summarized in Section IV. Section V includes the conclusion.

II. LITERTAURE REVIEW

The first Migrating crawler architecture was proposed by Odysseas [4]. He pointed out that the traditional centralized crawling model suffers from processing , DNS lookup and network bottleneck. He also observed that traditional centralized crawling cannot cope up with dynamic nature of web-sites. Therefore proposed a novel crawling architecture called UCYMicra was proposed .

In 2008, Dixit et.al [7] proposed Scalable Parallel Migrating crawler that utilizes the characterstics of both parallel and migrating crawlers for efficient download of web documents. The proposed crawler claimed to have improved time efficiency and better network bandwidth utilization.

To reduce Network load and bandwidth problem, Migrating crawlers were developed where all crawling tasks were performed at the web server itself. A migrating crawler [4] operates by distributing the downloading task among various instances. In order to download data efficiently and quickly selection of appropriate server is obligatory. Dajie et al [5] proposed use of Time and Weight factors for dispatching new URLs to the already existing Migrants on a web server. Servers where Migrants were having larger time factor were considered busy and were not assigned new URLs whereas Servers with high weight Migrants were assigned new URLs for crawling . The Time and Weight factor scheme was simple and efficient. However it suffered from Single point of failure problem and lacked scalability.

Yuan Wan et al [6] designed and implemented a hashing based server assignment method. For scheduling, a URL was taken as input and a hash function was applied to it. URL was then assigned to that migrant whose ID matches with the generated hash code of the URL.The scheme however was less realistic and resulted in server load imbalances.

Dixit et. al[7] proposed nearest server selection scheme where first ICMP ping tool was employed to find the information about the region of the URL's IP address and the URL with the smallest probing time was then used to download the web pages. However, the scheme was not suitable and resulted in server imbalances and generated longer response times.

III. PROPOSED WORK

To overcome the deficiencies of the existing server selection schemes a novel Credit based Server Selection mechanism has been proposed. The proposed mechanism assigns each server a credit score based on various network parameters and server with the highest credit score is then selected for migration of migrants so that useful information can be downloaded efficiently in minimum time.

Factors for Server Selection Module:

Following factors are being used for calculating credit value of a Server:

- Network Distance: It is defined as the number of hops required to transfer data from source to destination.
- Network bandwidth: It is defined as the data rate supported by network connection.
- Server load: It is defined as the number of tasks and/or migrants already executing on the server.
- Trust values: It is based on reliability of communication path.

The algorithm for Server Selection Module is given in Figure 1. The proposed algorithm first selects a URL from URL Frontier list (Frontier list contains list of URLs to be visited). Then an inquiry message is sent to the local Regional Internet registries for getting the information for the subnets containing the IP addresses of the documents lying at more than one place. Probing messages are then sent to get information about Network Latency, Network Bandwidth, Server load and Trust values.

```
Server_Selecion()

Do forever

Step 1: Wait(crawlUrls)

        2: Pick the URL from url frontier list.
        3: Send inquiry message and retrieve server information in Server Info       Database
        4: Find an optimal server for picked URL
             4.1  First a preference matrix is created to find out relative importance of each criteria
             4.2  Score value of each server is calculated in terms of their contribution to each criteria
             4.3  A priority value for each of the server is calculated.

      5: Migrant is then dispatched to Server with the highest priority value.
     6: Signal(migrant_sel)
     7: End
```

Figure 1: Algorithm for Server Selection

The information is then stored in Server Info Database. Then the credit score for each of the server is calculated using Analytical Hierarchy Process. The analytic hierarchy process (AHP) is a structured technique for organizing and analyzing complex decisions based on mathematics and psychology [8].This technique has been used as it represents the most accurate approach for quantifying the weights of criteria.

The methodology of AHP is given below:

1. First an organizational model comprising of goals, criterion and alternatives is created.
2. Next relative importance of each criterion for achieving the goal is calculated.
3. Priority of each alternative w.r.t to each criterion is calculated.
4. Ranking of all alternatives is done to find the best alternative to meet the goals.

ILLUSTRATION

The process of selecting best server through AHP process is given below:

Step 1: Organisation Model Structure

The organisational model consists of three layers. First layer depicting goal which is in this case is selecting appropriate server, characteristics to be compared for server selection are represented in the second layer and finally different servers available for selection are shown at the bottom layer .

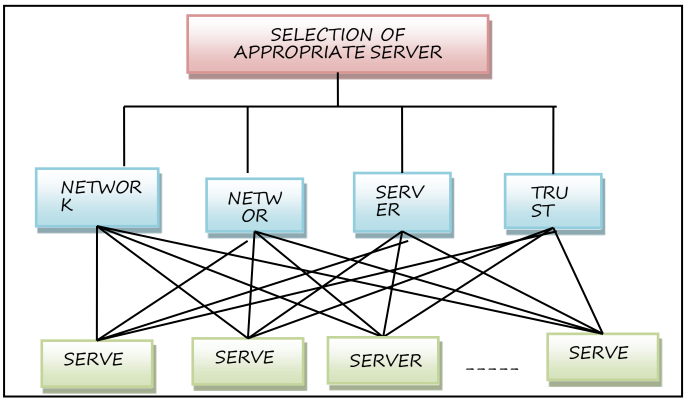

Figure 1: Organizational Model

Step2: Comparison Matrix

In this step relative importance of each characteristic of server is calculated. A matrix of relative importance is formed by making pair-wise comparison using following two steps:

Step 2.1: Assigning relative importance to each criterion

Since each criterion has a different measuring unit, therefore to simplify measurement a small workshop of Search Engine experts was conducted to help assign relative importance to each criterion. The experts based on their opinion have assigned importance to each criterion with respect to other criteria's. For example, expert's felt that Server load is 2 times better than Trust value. Based on expert's opinion, the order of importance of all the factors is given in Table 1

Table 1 :Relative Weight Table

Rating	Judgement
9	Extremely preferred
8	Very strongly to extremely preferred
7	Very Strongly preferred
6	Strongly to very Strongly preferred
5	Strongly preferred
4	Moderately to Strongly Preferred
3	Moderately Preferred
2	Equally to moderately Preferred
1	Equally Preferred

Step 2.2: Weight calculation of each Criterion

Weight of each criterion is calculated as follows:

1. Sum the values in each column
2. Divide each element by its column total
3. Take an average of elements in each row

This calculated weight is considered as relative importance of each criterion with respect to each other. For example the value of weight 0.4236 is derived as follows:

a) Values in each column are added i.e (1+1+1/2+1/5)= 2.7. Similarly other column values are added and we get values of 2.83, 5.5 and 11 respectively

b) Next, divide each element by its column total i.e (1/2.2+1/2.83+2/5.5+5/11)= 1.5417

c) Take an average of this value = 1.5417/4=0.3854

Similarly other weight values of each criterion are calculated. Table 2 shows the relative importance of each criterion.

Table 2: Criterion Comparison and Weight Table

Characteristic	Network Bandwidth	Network Distance	Server Load	Trust Value	Weight
Network Bandwidth	1	1	2	5	0.3854
Network Distance	1	1	2	3	0.339
Server Load	½	½	1	2	0.181
Trust Value	1/5	1/3	½	1	0.092
Total	2.7	2.83	5.5	11	

Step 3: Calculating Priority of Each Criterion

The priority is calculated by comparing each server with respect to all criterions in terms of weight . Suppose there are N server available and there are M criterions, then M number of NXN Matrices will be constructed. Example: Table 3 shows 3 different server alternatives with different values of criterion with an objective to select best server for Migrant migration. Priority of each criterion and selection of best server is calculated as follows:

Table 3: Server alternatives table

	Network Bandwidth	Network Distance	Server Load	Trust Value
Server1	12	4	10	2
Server2	6	8	5	3
Server3	3	16	20	4

Table 4, 5, 6 and 7 shows the comparison of three servers on the basis of Network Bandwidth, Network Distan9ce, Server Load and Trust value respectively. The weights are calculated by using steps explained in step 2.2. For example to calculate weight of S1 in Table 1.4 is calculated as follows: first the values in the column are added (i.e 1+1/2+1/3 =1.75) , Next, each element is divided by its column total (1/1.75+2/3.5+4/7=1.7142) and then take average (0.83/3= 0.5714). Similarly other weight values in all tables are calculated.

Table 4: Server comparison w.r.t to Network Bandwidth

Network Bandwidth	S1	S2	S3	Weight
S1	1	2	4	0.5714
S2	½	1	2	0.2857
S3	1/3	1/3	1	0.1745
	1.75	3.5	7	

Table 5: Server comparison w.r.t to Network Distance

Network Distance	S1	S2	S3	Weight
S1	1	2	3	0.546
S2	½	1	3/2	0.273

S3	1/3	2/3	1	0.182
	1.83	3.66	5.5	

Table 6: Server Comparison w.r.t to Server Load

Server Load	S1	S2	S3	Weight
S1	1	½	2	0.286
S2	2	1	4	0.571
S3	½	¼	1	0.143
	3.5	1.75	7	

Table 7: Server Comparison w.r.t to Trust Value

Trust Value	S1	S2	S3	Weight
S1	1	1	4/3	0.364
S2	1	1	4/3	0.364
S3	¾	¾	1	0.273
	2.75	2.75	3.66	

Step 4: Next the consistency of the given information corresponding to given criterion and available server alternatives is evaluated using the values of Consistency Ratios. Consistency ratio determines how consistent the judgements have been relative to large samples of purely random judgements. It is defined as the ratio of consistency index (CI) and random consistency index (RI) .

CR= CI/RI

CI is calculated by using equation 1(A)

Consistency Index= (λmax-n)/(n-1) (1(A))

Random Index is the consistency index of a randomly generated pair-wise comparison index comparison matrix. It depends on the number of elements being compared and takes on following values as shown in Table 8

Table 8: Random Index Table

N	1	2	3	4	5	6	7	8	9	10
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

If the value of consistency ratio is smaller or equal to 10%, it is acceptable. For values higher than 10%, there is need to revise the judgement. For example the consistency ratio of Table 8 is calculated as follows

CI= (1.75*0.5714)+(3.5*.2857)+(7*.1745)= 3.2213

RI= .58

Therefore CR= 3.2213/.58* 100= 0.055

Since the CR value is less than 0.1 , therefore the matrix is consistent and the information contained in it is acceptable. Similarly consistency ratio of all other matrices is calculated.

Step 5: After the weights of each server corresponding to given criterion are calculated a priority matrix is generated and rank of each server is calculated. Table 5.8 shows the priority matrix for three server alternatives.

The priority of Server1 = ((0.5714*0.3854)+(0.546*0.339)+(0.286*0.181)+(0.364*0.092))=0.3306. Similarly priority of Server2 , Server 3 are calculated. Table 1.9 shows the priority matrix for three servers.

Table 9: Priority Matrix

Criterion Weight	0.3854	0.339	0.181	0.092	
Crtiterion → ↓ Server	Network Bandwidth	Network Distance	Server Load	Trust Value	Priority
S1	0.5714	0.546	0.286	0.364	0.4903
S2	0.2857	0.273	0.371	0.364	0.3031
S3	0.1745	0.182	0.143	0.273	0.1797

After the priorities are calculated, Server with highest priority is given Rank 1 and is most suitable for migration of the migrant. Table 10 shows the Ranking of each Server on the basis of their priority:

Table 10: Ranking of Servers

Server	Priorities	Rank
S1	0.4903	1
S2	0.3031	2
S3	0.1797	3

From the given example it is thus concluded that Server 1 is most appropriate for the transfer of the migrant. The crawler manager waits for the signal crawl URL and will then dispatch the migrant to server 1 for crawling of information.

IV. EXPERIMENTAL EVALUATION

The performance of proposed AHP based server selection method was compared with a Conventional method of server selection. In conventional method since only network distance was used as a parameter for server selection therefore it lead to uneven load distribution where some servers were overloaded and some were under- utilized and hence resulted in more crawling time. Whereas proposed AHP based server selection mechanism, achieves load balancing and reduction in crawling time and proper utilization of server resources.

The experiment was conducted on four systems of different configurations. Crawling manager was running on one machine and other machines were having instances of migrating crawler. The three machines were having similar index of web pages. The Crawler-Manager was responsible for distributing the migrants on different machines. Server parameters of Network Bandwidth, Server load and Trust values were changed in three test cases while keeping Network distance same every time . Following are the results of the experiments conducted

.

	Migrating Crawler based on AHP based Server Selection Scheme	**Migrating Crawler based on Nearest Server Selection Scheme**
Total Links Crawled	4456	3743
Links Saved	4021	3692
Time Taken(in secs)	14300	14300

The efficiency of Migrating crawler based on proposed AHP based Server Selection Scheme can be calculated in terms of URLs crawled and efficient utilization of network resources.

(i) Efficient Utilization of Network Resources: It is based on proper utilization of network resources. Based on the results obtained of three test cases it was observed Conventional Migrating crawler always selected Server which at least distance without considering its bandwidth or load. As a result it causes load imbalances. On the other hand migrating crawler based on proposed server selection mechanism selects different Machines for Migrant migration and properly utilizes all the network resources. The graphs in Figure 3 reflect the effect of proposed mechanism on load distribution.

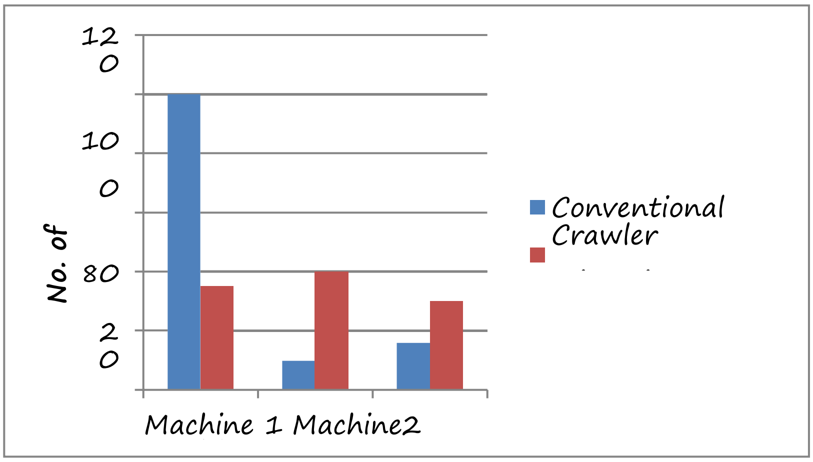

Figure 3: Comparison of Load Distribution

ii) Efficiency crawling : It is based on the value of URLs crawled in particular time. Proposed Migrating crawler using AHP Server Selection Scheme crawled more links. Furthermore due to uneven load distribution the conventional crawler's takes longer time to process and send information back to the crawling manager and hence crawls less number of URLs. Figure 4 shows the effect of improper load distribution on number of URLs crawled by conventional migrating crawler.

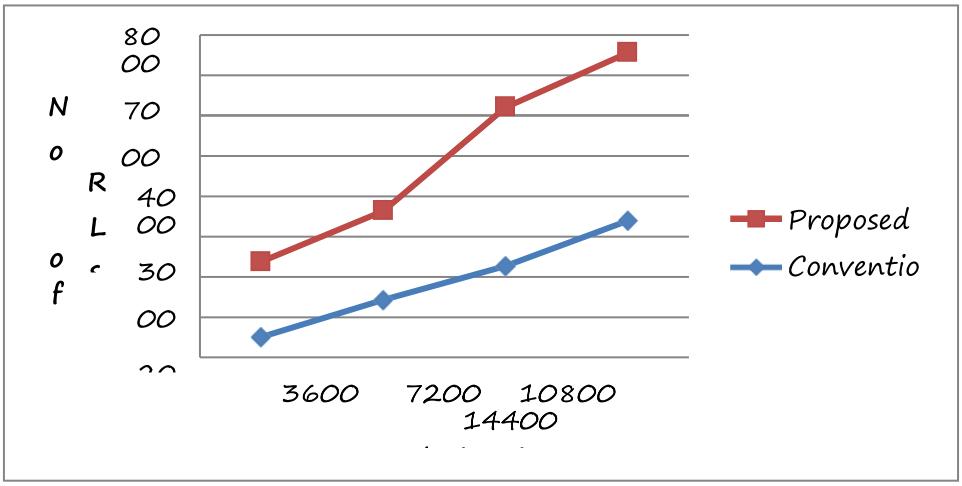

Figure 4: Comparison of number of URLs crawled

Hence it is proved that the migrating crawler based on proposed AHP based server selection method outperforms conventional migrating crawler based on nearest server selection method in terms of both efficiency and utilization of network resources.

V. CONCLUSION

In this paper a novel Credit based server selection mechanism has been proposed. The proposed mechanism considers network parameters like Network distance, Bandwidth availability, and existent load on the server and reliability factors to calculate the credibility of each server. The experimental results conforms that the proposed technique outperforms other server selection technique based on nearest possible server in terms of efficient utilization of network resources and fast crawling of web pages.

VI. FUTURE WORK

To effectively utilize network bandwidth migrating mobile agents called Migrants are dispatched to download data in the proximity of the data itself. Since web sites store data on multiple sites therefore factors like network-load, trust, Server load and Network distance etc are being used to select the best server to dispatch the migrant so that fast crawling of documents may be achieved. This work can be amalgamated in future with web usage mining techniques along with user preferences for ranking web pages so that the user gets desired results in less number of clicks

References

1. Dirk Lewandowski “Web searching, search engines and Information Retrieval, Information Services & Use 25(2005).
2. Saini, C., & Arora, V. (2016, September). Information retrieval in web crawling: A survey. In Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on (pp. 2635-2643). IEEE.
3. Kumar, M., Bhatia, R., & Rattan, D. (2017). A survey of Web crawlers for information retrieval. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(6).
4. Odysseas Papapetrou, Stavros Papastavrou , George Samaras, “ Distributed Indexing of the Web using Migrating Crawlers”, PhD. Thesis, Computer Science Detartment, University of Cyprus.
5. Ge, D., & Ding, Z. (2014, December). A Task Scheduling Strategy Based on Weighted Round-Robin for Distributed Crawler. In Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on (pp. 848-852). IEEE
6. Y.Wan and H.Tong, “ URL Assignment Algorithm of Crawler in Distributed Crawler in Distributed System Based on Hash”,2008 IEEE International Conference on Networking Sensing and Control, 2008
7. Ashutosh Dixit, A.K. Sharma : Design of Scalable Parallel Migrating Crawler Based on Augmented Hypertext Documents. Ph.D. Thesis, MDU, May 2010
8. Punj, Deepika, and Ashutosh Dixit. "Design of a Migrating Crawler Based on a Novel URL Scheduling Mechanism using AHP." International Journal of Rough Sets and Data Analysis (IJRSDA) 4.1 (2017): 95-110.
9. [A. Gupta, A. Dixit and A. K. Sharma, "Prospective Terms Based Architecture for Migrating Crawler," *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, Mathura, 2012, pp. 915-919, doi: 10.1109/CICN.2012.168.
10. Ashlesha Gupta, Ashutosh Dixit and A.K. Sharma, “Recent Trends In Effective Design Of Search Engines” Volume 4, Special Issue(1) , Journal of Network communications and Emerging Technologies , September- 2015, ISSN:2395-5317.
11. Manvi, Ashutosh Dixit, Komal Kumar Bhatia, Bhumika Wadhwa, “Generating domain specific ontology for retrieving hidden web data”, IEEE International Conference ISCON,2014, GLA University, Mathura,1-2 March 2014.
12. Manvi, Komal Kumar Bhatia and Ashutosh Dixit, “A novel design of hidden web crawler using ontology”, International Journal of Engineering Trends & Technology (IJETT), August 2015, ISSN: 2231-5381 DOI: 10.14445/22315381/IJETT-V26P204. DBLP indexed
13. Joachim Hammer, “Using Mobile Crawlers to Search the Web Efficiently”, Dept of Computer & Information Science & Eng. University of Florida Gainesville, U.S.A.
14. Agichtein, Eugene, Eric Brill, and Susan Dumais. "Improving web search ranking by incorporating user behavior information." Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006.
15. Singhal, N., Dixit, A., Agarwal, R.P. et al. A reliability based approach for securing migrating crawlers. Int. j. inf. tecnol. 10, 91–98 (2018). https://doi.org/10.1007/s41870-017-0065-0
16. Faizan Farooqui M., Rizwan Beg M., Qasim Rafiq M. (2013) A Critical Review of Migrating Parallel Web Crawler. In: Meghanathan N., Nagamalai D., Chaki N. (eds) Advances in Computing and Information Technology. Advances in Intelligent Systems and Computing, vol 177. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31552-7_63