A Novel Mobile Crawler System Based on Filtering off Non-Modified Pages for Reducing Load on the Network

Rajender Nath[1] and Satinder Bal[2]
[1]Department of Computer Science and Applications, Kurukshetra University, India
[2]Department of Computer Science and Applications, Vaish College of Engineering, India

Abstract: *The studies in the literature show that about 40% of the current Internet traffic and bandwidth consumption is due to web crawlers that retrieve pages for indexing by the different search engines. This traffic and bandwidth consumption will increase in future due to the exponential growth of the web. This paper addresses the problem of bandwidth consumption by introducing an efficient indexing system based on mobile crawlers. The proposed system employs mobile agents to crawl the pages. These mobile agent based crawlers retrieve the pages, process them, compare their data to filter out pages that are not modified after last crawl, and then compress them before sending them to the search engine for indexing. The experimental results of the proposed system are very encouraging.*

Keywords: *Search engine, mobile crawler, network resources, and internet traffic.*

Received March 17, 2009; accepted August 3, 2009

1. Introduction

According to [13], the World-Wide-Web (WWW) has grown from a few thousand pages in 1993 to more than several billion pages at present. Search engines maintain comprehensive indices of documents available on the web to provide powerful search facilities. According to [3], the web is very dynamic and 40% of its contents change daily. Web crawlers are used to recursively traverse and download web pages (Giving GET and POST commands) for search engines to create and maintain the web indices.

According to [11], the needs of maintaining the up to date pages in the collection cause a crawler to revisit the websites again and again. Due to this, the resources like CPU cycles, disk space, and network bandwidth, *etc.*, become overloaded and sometime a web site may crash due to such overloads on these resources.

One study [14] reports that the current web crawlers have indexed billion of pages and about 40% of current internet traffic and bandwidth consumption is due to the web crawlers that retrieve pages for indexing by the different search engines. According to [10], the maximum web coverage of any popular search engine is not more than 16% of the current web size.

Different approaches [6, 7, 12] were proposed and many systems were suggested to build web indices based on mobile crawlers powered by mobile agents. These mobile crawlers are transported to the site of the source where the data resides in order to filter out any unwanted data locally before transferring it back to the search engine. These mobile crawlers can reduce the network load caused by the crawlers by reducing the amount of data transferred over the network. The internet traffic and load on the resources (such as bandwidth, CPU cycles and memory) can be reduced only if the crawlers access those pages only that are actually modified.

In this paper, we present an alternate approach of URL delegation and filtration of the web pages. The approach assists the search engine to perform an efficient URL delegation to the mobile crawlers. These crawlers then move to the remote sites, filters out those pages that are not modified after the last crawl and send the modified pages to the search engine, for indexing, after compressing them.

The rest of the paper is organized as follows: The related work is discussed in section 2. Section 3 describes the problems with current crawling techniques. Section 4 describes the proposed architecture of the mobile crawler system. Sections 5 and 6 describe the URL delegation approach used by the system and the working of the proposed system respectively. Section 7 describes theoretical analysis of the proposed system. Sections 8 and 9 provide the experimental setup and results respectively. Finally, in Section 10, we conclude the paper.

2. Related Work

There is a little coverage in the literature on commercial crawlers. One study [2] is the major sources of information. Another study [2] provides information about the crawlers is the Harvest project, which investigates the web crawling in detail. A web crawler consumes a significant amount of network bandwidth and other resources by accessing the web resources at a fast speed. This affects the performance of the web server considerably. The [8] has suggested a set of guidelines for the crawler developers that have been published to handle this problem. A significant amount of resources of underlining network are consumed to build a comprehensive full text index of the web. Further, to maintain the indices of a search engine up-to-date, crawlers constantly retrieve the pages at a fast speed, which causes a huge load on to the web daily.

The studies of [4, 13], proposed distributed and parallel crawling systems to increase the coverage and to decrease the bandwidth consumption but these systems just distribute and localized the load but does not help much in reducing the load.

The studies of [6, 7], proposed web crawling approach based on mobile crawlers powered by mobile agents. These mobile crawlers can exploit the information about the pages being crawled in order to reduce the amount of data that needs to be transmitted to the search engine. These mobile crawlers move to the resources that need to be accessed. After accessing a resource, mobile crawlers move on to the next server or to their home machine, carrying the crawling results in their memory. The main advantage of mobile crawling is localized data access, remote page selection, filtering and compression.

The study of [12], first designed and developed UCYMicra, (a distributed web crawling infrastructure) and then IPMicra, a mobile crawler based system. Their mobile crawlers move to the web servers, and perform the downloading of web documents, processing, and extraction of keywords and after compressing, transmit the results back to the central search engine. Further, these migrating crawlers remained in the remote systems and perform constant monitoring of all the web documents assigned to them for changes.

3. Problems with Current Traditional and Mobile Crawling Techniques

To build a comprehensive full text index of the web and to keep it up-to-date, significant resources of the underlying network are consumed by the crawlers. According to [5], it is estimated that web crawlers of big commercial search engines crawl up more than 10 million pages per day.

The major problem with the traditional crawling techniques is that there is no way to determine at the remote site, the pages that are actually modified after the last crawl and needs to be re-indexed. What ever can be done is at the search engine site. This will result in wastage of resources in indexing the pages that have already been indexed by the search engine. The studies of [6, 7, 12] introduced the mobile crawling techniques to overcome the problem of traditional crawling techniques as discussed above. In these techniques, mobile crawlers, crawl the web pages and stay in the remote system to monitor any change that can take place in the pages allotted to them. But these techniques have their own problems such as:

1. The mobile crawlers that always stay in the memory of the remote system occupy a considerable portion of it. This problem will further increase, when there are a number of mobile crawlers from different search engines. All these mobile crawlers will stay in the memory of the remote system and will consume lot of memory that could have otherwise been used for some other useful purposes.
2. It can also happen that the remote system may not allow the mobile crawlers to reside permanently in its memory due to security reasons
3. In case a page changes very quickly then the mobile crawler immediately accesses the changed page and sends it to the search engine to maintain up-to-date index. This will result in wastage of network bandwidth and CPU cycles *etc*.

To address these problems of the mobile crawling, the authors of the paper have proposed architecture for mobile crawling that makes use of frequency change estimators at the search engine site for efficient URL delegation and some new modules will help in filtering out those pages at the remote site that are not actually modified after the last crawl.

4. The Architecture of the Proposed Mobile Crawler System

As [6, 7, 9, 12] demonstrates, the mobile crawlers are constructed as mobile agents and are dispatched to web servers where they download the web documents, process them and extract keywords and URL's for comparison and filtration of non-modified pages , and finally, the pages actually modified are compressed and transmitted back to the Search Engine. The major components of our PMCS are:

1. Crawler Manager (CM).
2. Frequency Change Estimator Module (FCEM).
3. Statistics Database (SD).
4. Old Database File Module (ODBFM).
5. Comparator Module (CoM).
6. Analyzer Module (AM).
7. Remote Site/Server (RS).

Each Component is shown in Figure 1 and is briefly discussed below.

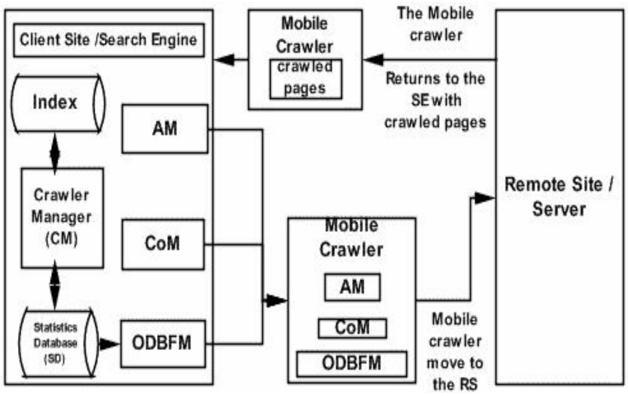

Figure 1. Proposed Mobile Crawler System.

4.1. Crawler Manager

The CM performs various tasks. The major tasks of CM are mobile crawler generation, delegation of URL's to the mobile crawler for crawling purposes based on the information taken from the FCEM, construction/ Updating the data of ODBFM , updating the SD , receiving the crawled pages from the RS and indexing them. The task of decompressing the pages (to be indexed) received from the RS, in compressed form, is also performed by the CM. After generation and URL delegation to the mobile crawlers, these mobile crawlers are sent to the designated RS's for crawling purposes by the CM.

4.2. Frequency Change Estimator Module

This module is the part of the client site and remains there. The FCEM is used to calculate the probability of the page change. This module maintains the page change frequency of every page at the SE and updates this information in SD every time a page is crawled. This information about the pages give a good idea to the SE that what page is to be re-crawled and when. The CM uses this information for URL delegation to the mobile crawlers. This information helps the CM in deciding which URL's are delegated and when to the mobile crawler. Thus, the FCEM filters all those pages at the client site that has low probability of change and reduces the work of the crawler. The concept of Estimation of Frequency (Existence of change) is used in our proposed system to establish the probability of page change [3].

4.3. Statistics Database

This module is maintained at the SE and contains information about all the pages crawled by the mobile crawlers. This module has five fields:

1. *Name of URL*- This field stores the names of the pages that are indexed by the SE.
2. *Last modified date*- This field stores the date of the corresponding page on which it is last time modified at the RS.
3. *Number of URL's in the page*- This field contains information about the total number of URL's/the number of links to the other pages in the corresponding page.
4. *Number of keywords in the page*- This indicates the number of keywords a page contains.
5. *Frequency of change*- This indicates the frequency of page change for each page and is updated by the FCEM after every crawl.

4.4. Old Database File Module

The CM constructs this module one for each mobile crawler. It contains statistics about each HTML page to be crawled. This statistics is taken from the SD. This module is sent with the corresponding mobile crawler on to the RS. This module has four fields:

1. *Name of URL*
2. *Last Modified Date*
3. *Number of URL's in the page*
4. *Number of keywords in the page.*

4.5. Comparator Module

The major function of this module is to compare the statistics of each HTML page taken from ODBFM with the statistics collected by the AM of the corresponding HTML page at the remote site to filter out the pages that are not modified. The CoM first compares last modified date of the HTML page (if available) in the log file with the last modified date in the ODBFM. If it mismatches, then this page is sent to the SE for indexing. If it is missing, the CoM compares other field's data (umber of URL's in the page, umber of keywords in the page) taken from the AM for each crawled page with the statistics available in the ODBFM of the corresponding page one by one. It helps in deciding whether the retrieved page is candidate to be sent back to the SE or not for indexing. It is assumed that crawled page is modified if any one of the three items (last modified date, number of URL's or number of keywords in the page) is changed. The page that is modified is stored for sending it to the SE.

4.6. Analyzer Module

The analyzer module performs the same task at both the sites (i.e., on the SE and RS. At the SE constructs the most part of the SD. It scans and analyses all the pages crawled by the mobile crawler and extracts and counts the number of URL's (preceded by href tag) and number of keywords (not unique but except general words) in each page. Initially, it creates this data by scanning and analyzing each page crawled by the mobile crawler and then updates this data by scanning the web pages received from the mobile crawler after every crawl.

Initially, it is sent at the RS with the mobile crawler and a copy of it is stored in the secondary memory of RS for future use by the mobile crawler. Here the AM scans and analyses each HTML page one by one that is delegated to the mobile crawler for crawling and collects the number of URL's and number of keywords in each page. The AM then send this data to the CoM to compare it with the statistics of that HTML page in the ODBFM for filtering those pages that are not modified after last crawl.

5. The URL Delegation Approach Used by Proposed Mobile Crawler System

The CM delegates the URL's to the mobile crawlers. It is assumed that the pages of one web site generally remain on a single server. So the CM allocates only those pages to the mobile crawler that belongs to the same server. Further, the pages are allocated to the mobile crawlers on the basis of information taken from the SD. Also ODBFM is sent with each crawler that contains information about the pages to be crawled. This information is taken by the crawler manager from the SD and stored in the ODBFM during URL delegation process. To store this information for the URL under consideration in the ODBFM, the CM works as given below.

If the probability of page change is less than or equal to 0.10, then it is assumed that enough information about page change frequency is not available and the page needs to be accessed to obtain more information. So there is no need to put any data in the ODBFM. The mobile crawler does not send these pages to the AM for scanning and analysis to collect the statistics about them. The mobile crawlers directly retrieve, and store these pages for sending them to the Client Site/Search Engine.

If the probability of page change (taken from the SD) comes out to be greater than 0.10 and less than 0.80 then the corresponding page's URL with last modified date, number of URL's in the page and number of keywords are stored in the ODBFM. If the page change probability is greater than or equal to 0.80, it has been assumed that it is most certain that the page is being modified so there is no need to put the data regarding last modified date, number of URL's and number of keywords in the ODBFM. The AM of mobile crawler does not scan and analyses these pages to collect the statistics. The mobile crawlers directly retrieve, and store them for sending to the SE.

By directly retrieving and indexing those pages whose page change probability is <= 0.10 or >= 0.8, some load on the network has been reduced. This reduction is achieved by not storing any information about these pages in the ODBFM. It also saved CPU time of the RS used in the analysis of these pages.

.

6. Working of Proposed Mobile Crawling System

The complete working of PMCS is described in Figure 2. The mobile crawlers used in PMCS takes the ODBFM, the CoM and the AM with it on the RS from where the pages are to be crawled. The mobile crawler accesses those pages one by one whose URL's are given in the ODBFM.

The CoM, first of all, compares the last modified date of the page in the ODBFM with the last modified date of the web page retrieved from the log file (if available) of the RS, if it is matched than the page is not a candidate to be sent to the SE for indexing because the page was not updated since last modification, but if mismatch occurs then the page is sent to the SE for indexing.

But the last modified date is not available with all the web pages. So, all those pages whose last modified date is not available are passed to the AM. The AM scans and analyses all these pages one by one and counts the number of URL's and number of keywords in them. Firstly, the value of URL's are compared. If the mismatch occurs the page is indexed and if the URL count is same then another comparison is done because it is not possible to decide by the equal number of URL count that whether the page is modified or not as there may be the situation where the number of links added to a page is equal to the number of the links deleted or the target location of the link may be changed. So, another comparison has to be made about the number of keywords in that web page. If the number of keywords is not same, then the page is indexed else it is not indexed.

7. Analysis of the Proposed Mobile Crawling System (PMCS)

In this section, it is shown that the proposed approach is theoretically correct and will actually provide the desired benefits. The following abbreviations are used in this section.

$þ_t$: Total number of web pages indexed by Traditional Crawlers (TC).

$þ_m$: Total number of web pages indexed by PMCS.

P_t : Number of web pages whose probability of change is ≤ 0.10.

P_{te} : Number of web pages whose probability of change lies between 0.10 & 0.80.

P_e : Number of web pages whose probability of hange is ≥ 0.80.

P_{av} : Average page change probability of the pages, whoes Individual Probability of change lies between 0.10 to 0.80.

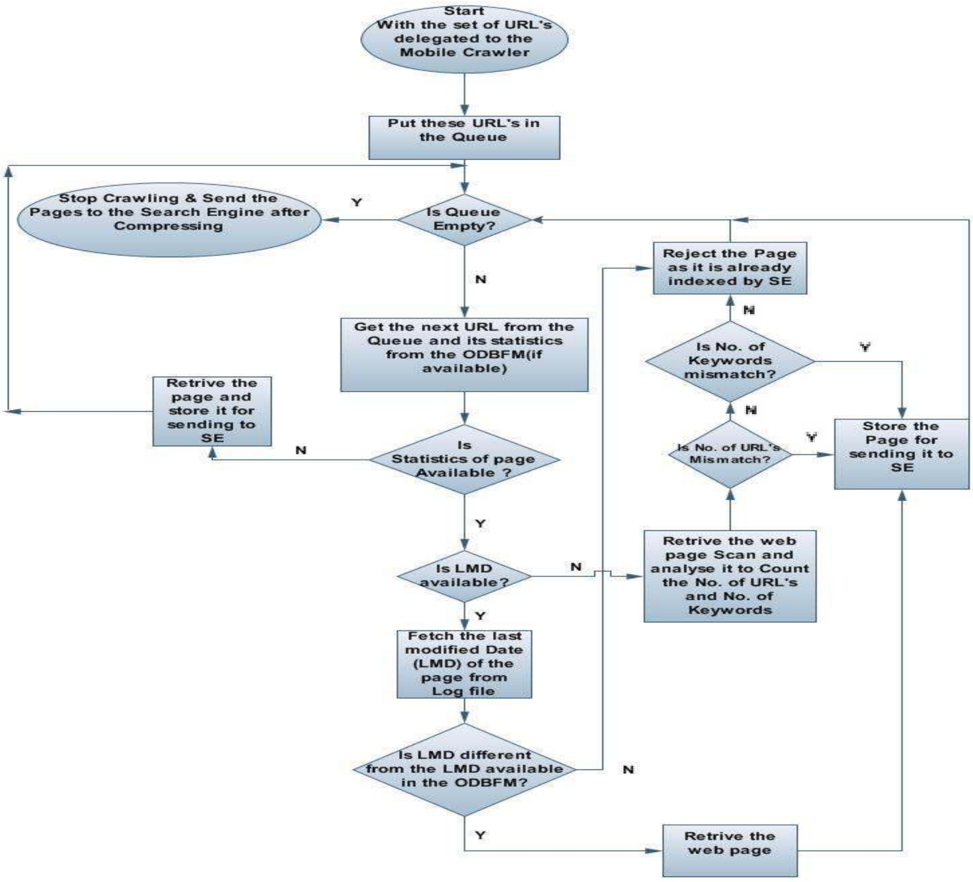

Figure 2. Working of PMCS.

In case of TC, the total numbers of pages indexed are

$$þ_t = P_t + P_{av} * P_{te} + P_e \quad (P_{av} = 1) \qquad (1)$$

The number of web pages indexed by PMCS can be

$$þ_m = P_t + P_{av} * P_{te} + P_e \quad (P_{av} \neq 1) \qquad (2)$$

Difference in the number of web pages indexed

$$= þt - þm \qquad (3)$$

The load reduced on the network by the PMCS (in KB) is

$$= ((þ_t - þ_m) * \text{size of each page (in KB)}) \qquad (4)$$

This means load reduced on the network is directly proportional to the difference in the number of pages indexed by the PMCS and the TC.
Thus, the % Load reduced by the PMCS

$$= ((þ_t - þ_m) / þ_t) * 100 \qquad (5)$$

To prove the above stated fact lets calculate the P_{av} *i.e.*, the average page change probability, when the individual page change probability lies between 0.10 to 0.80 for different cases (*i.e.*, average case, best case and worst case). It is assumed that 100 pages are to be delegated to the mobile crawler for crawling purpose. The average case (page change probability):- it is assumed that $P_t = 10$, $P_e = 10$ and $P_{te} =80$ .Further it is assumed that of these 80 pages, 20 pages have page change probability 0.15 and 40 pages have page change probability 0.45 and rest 20 pages have page change probability 0.55. Then P_{av} will be $P_{av} = (20*15 + 40*45 + 20*55) / 80*100 = 3300/8000 =0.4125$ thus, on the average approximately 41 pages change in this case, that saves 39 pages on the average. The best case (page change probability) exists when all the pages lies in P_{te} and the average page change probability of all the pages is just above 0.10 *i.e.*, $P_t =0$, $P_e = 0$, $P_{te} =100$ and $P_{av} = 0.11$, this is the best case, since more pages are filtered at the remote site due to low probability of page change. The worst case (page change probability) exists when all the pages lie in P_{te} and the average page change probability of all the pages is just below 0 .80 *i.e.*, $P_t =0$, $P_{te} =100$, $P_e =0$ and $P_{av} = 0.79$, this is the worst case, since less pages are filtered at the RS because of high probability of change.

Following two extreme cases are worth noting: In first case, all the pages have the probability under 0.10 *i.e.*,

$$P_t =100, P_{te} =0, P_e = 0.$$

In the second case, all the pages have the probability over 0.80 *i.e.*,

$$P_t =0, P_{te} =0, \quad P_e =100.$$

In both the cases, all the pages are sent back to the SE for indexing by the PMCS.

8. Experimental Setup

A virtual environment has been set up to perform the experiments. The Remote Site/Server and the SE have Intel processor clocked at 3.06 GHz with 1GB of RAM. Both the machines support java environment and Tahiti server is installed on them to support the Java Aglets. Both the machines have Windows XP operating system. These two machines are connected through high-speed LAN. One hundred (100) sites (such as .com, .org, .edu *etc*.) are selected and home pages of these sites are downloaded and stored on the RS. The mobile crawlers have been developed using Java Aglets on Client server/Search Engine and URL's are delegated to them for crawling. The experiment was performed for 30 days. The crawler developed by the authors of this paper visited the Remote Site (RS) and analyzed the pages for modification after the last crawl and returned only those that were actually modified. Following types of data were collected during the experiment.

1. Number of pages that were modified after the last crawl.
2. The parameters responsible for modification-change in URL count, change in keyword count.
3. The number of pages retrieved directly by the crawler.

9. Experimental Results and Discussion

This section shows the results of the PMCS. To perform the experiment, 100 pages were selected. On an average, each page was of 8 KB. The selected pages were downloaded through the experimental set up discussed above on the remote site for 30 days. The crawler that is constructed on the SE visits the RS and performs the analysis of the pages and compares the data obtained with the data in the ODBFM. The crawler returns to the SE with the pages that are actually modified. The following results are the average of the data collected for 30 days. Following three parameters are considered for comparing the proposed mobile crawler system with that of traditional approaches.

1. Page change behavior.
2. Load on the network.
3. Bandwidth preserved.

1. Page Change behavior - It was found that on an average 37 pages changed due to change in the URL count in the page, 4 pages changed due to change in number of keywords only, 22 pages are retrieved directly as their probability of change is either less than 0.10 or greater than 0.80 and rest of the pages do not change. The bar chart in Figure 3 shows the average number of pages retrieved due to different change criteria of pages in PMCS.

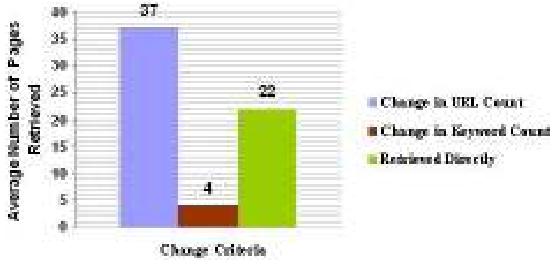

Figure 3. Average number of pages retrieved.

It is obvious from the bar chart in Figure 3 that out of 100 pages on the average only 63 (= 37+4+22) pages have been changed and on the average remaining 37 pages remained unchanged. It can be seen that the proposed mobile crawler system sends 63 pages not 100 pages as in case of traditional crawler thus saving the load of 37 pages on the network**.**

2. Load on the network-The load on the network is 800 KB when TC are used. But PMCS retrieved on the average 63% pages only and an additional 30 KB load caused by the additional components of the mobile crawler. Thus, the network load is little more than half when mobile crawler sends the pages to the client site without compressing. It is approximately one forth when mobile crawler sends the pages after compressing. Using standard compressing tools (e.g. WINZIP) the pages can be compressed up to 30% of the original size. The load can be further reduced if the additional components are also compressed. The bar chart in Figure 4 shows the comparison of load on the network (in KB) for different types crawlers used.

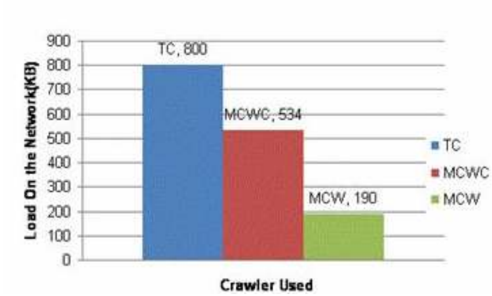

Figure 4. Mobile crawling benefits.

Three types of crawlers are used in the experiment - TC, Mobile Crawler Without Compression (MCWC), and Mobile Crawler With compression (MCW). It is quite obvious from the bar diagram in Figure 4 that the PMCS reduces the load little more than half of when compression is not used and it is approximately one fourth when compression is used. Thus in both of the cases, the PMCS performs much better than TC.

3. Bandwidth preserved - According to Shannon-Hartley theorem, the data rate must be twice the bandwidth of the channel. So, if it is assumed that the channel is of 4 KHz without noise (ideal case) then the data rate of this channel is 8 KBPS. Thus it takes 100 seconds to send the data using TC. It takes approximately 67 seconds to send the filtered data (only those pages which are actually modified) using MCWC and it takes 24 seconds to send the same data when pages are compressed before sending i.e. in case of MCW. The bar chart in Figure 5 shows the comparison of bandwidth that is available to send additional data when PMCS is used.

The bar chart in Figure 5 clearly shows that the PMCS preserves bandwidth by reducing the traffic on the network by way of filtering the unmodified pages at remote site. The PMCS preserves a little more than half of the bandwidth when compression is used and approximately three fourth of bandwidth is preserved as compared to TC. This bandwidth can be utilized for sending additional data.

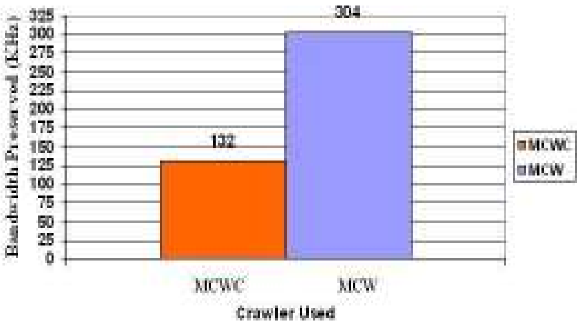

Figure 5. Comparison of bandwidth preserved.

10. Conclusions

The PMCS, developed by the authors of this paper has been found more efficient. The efficiency is in terms of reduction of load on the network, preserving bandwidth and saving memory than the TC. The authors have developed a mobile crawler in Java based on the proposed approach and created the virtual environment to perform the experiment for validating the results. The experiment was aimed to compare the proposed mobile crawler system with that of traditional crawler system on the following three parameters – page change behavior, load on network and bandwidth preservation. Experimental results have shown that the PMCS is more efficient than the TC systems. It was also found that on an average 63% of the total pages change (according to our experiment) and hence only 63% of the total pages need to be sent for indexing thus reducing 37% pages on the network. Experimental results have further shown that the proposed system can reduce the load on the network to one-fourth through the use of compression and consequently preserving the bandwidth of the network as compared to the traditional crawler system.

References

[1] Bowman C., Danzig P., Hardy D., Manber U., and Schwartz M., "Harvest: A Scalable, Customizable Access System," *Technical Report CU-CS-732-94,* Department of Computer Science, University of Colorado, Boulder, 1994.

[2] Brin S. and Page L., "The Anatomy of a Large Scale Hypertextual Web Search Engine," *Technical Report*, Stanford University, Stanford, CA, 1997.

[3] Cho J. and Garcia-Molina H., "Estimating Frequency of Change", *ACM Transactions on Internet Technology (TOIT)*, vol. 3, no. 3, pp. 256-290, 2003.

[4] Cho J. and Garcia-Molina H., "Parallel Crawlers," *in Proceedings of the Eleventh International Conference on World Wide Web*, Honolulu, Hawaii, USA, pp. 124-135, 2002.

[5] Cho J., Garcia-Molina H., and Page L., "Efficient Crawling Through URL Ordering," *Technical Report,* Computer Science Department, Stanford University, Stanford, CA, USA, 1997.

[6] Fiedler J. and Hammer J., "Using Mobile Crawlers to Search the Web Efficiently," *International Journal of Computer and Information Science*, vol. 1, no. 1, pp. 36-58, 2000.

[7] Fiedler J. and Hammer J., "Using the Web Efficiently: Mobile Crawling," *in Proceeding of the 7th International Conference of the Association of Management (AoM/IAoM) on Computer Science*, San Diego, CA, pp. 324-329, 1999.

[8] Koster M., "Guidelines for Robot Writers", A Web Document, *http://wsw.nexor.co.uk/mak/doc/robots/guidelines.html*.

[9] Lange D. and Oshima M., *Programming and Deploying Java Mobile Agents with Aglets*, Addison Wesley, 1998.

[10] Lawrence S. and Giles C., "Accessibility of Information on the Web," *Nature*, vol. 400, no. 6740, pp. 107-109, 1999.

[11] Nath R., Bal S., and Singh M., "Load Reducing Techniques an the Websites and other Resources: A comparative Study and Future Research Directions", *Journal of Advanced Research in Computer Engineering*, vol. 1, no. 1, pp. 39-49, 2007.

[12] Papapetrou O. and Samaras G., "Minimizing the Network Distance in Distributed Web Crawling," *CoopIS/DOA/ODBASE, OTM Confederated International Conferences,* Agia Napa, Cyprus, pp. 581-596, 2004.

[13] Shkapenyuk V. and Suel T., "Design and Implementation of A High Performance Distributed Web Crawler," *in Proceedings of the 18th International Conference on Data Engineering*, San Jose, California. IEEE CS Press, pp. 357-368, 2002.

[14] Yuan X. and Harms J., "An Efficient Scheme to Remove Crawler Traffic from the Internet," *in Proceedings of the 11th International Conferences on Computer Communications and Networks*, pp. 90-95, 2002.

Rajender Nath is a reader in the Department of Computer Science and Applications, Kurukshetra University, Kurukshetra. He did his PhD on the topic of software components classification and retrieval from Kurukshetra University, Kurukshetra. He has more than 20 years of teaching experience. Currently, he is teaching MTech (CSE) and MCA classes and actively engaged in his research work in the areas of software reuse, search engines, MANETS and biometrics. He has published more than 50 technical papers in international/national journals/conferences.

Satinder Bal is an assistant professor in the Department of Computer Science and Applications, Vaish College of Engineering, Rohtak. He did his post graduation from Maharshi Dayanand University, Rohtak. He has more than 10 years of teaching experience. Currently, he is doing PhD from Kurukshetra University, Kurukshetra. He has published more than 15 technical papers in international/national journals/conferences. He has published 8 books on various subjects with one having international edition.