Asian Journal of Information Technology 7 (3): 87-99, 2008
ISSN: 1682-3915

Predictive Prefetching Framework Based on New Preprocessing Algorithms Towards Latency Reduction

[1]G. Arumugam and [2]S. Suguna
[1]Department of Computer Science, University of Madurai Kamaraj,
Madurai, Tamil Nadu, India
[2]Department of MCA, K.L.N. College of Engineering,
Pottapalayam Sivagangai (Dist.), Tamil Nadu, India

Abstract: World Wide Web is an important area for data mining research due to the huge amount of information. The success of the WWW depends on response time. Predictive prefetching is an important technique to reduce latency. To predict the user request, millions of web logs from client side or from server side or from proxy side need to be analyzed. Most of the existing predictive prefetching methods are based on URL dependency graph, Keyword search techniques and Link structure without considering all the factors for cleaning web logs. In this study, we propose a new predictive prefetching framework based on our new preprocessing algorithm for cleaning web logs and user session identification. Our analysis shows that the new cleaning algorithm reduces latency considerably than the cleaning techniques used by the existing predictive prefetching methods. The experimental results based on five different servers prove the importance of cleaning of web logs before using them for predictive prefetching process.

Key words: Web usage mining, predictive prefetching, latency, cleaning and user session identification process

INTRODUCTION

The WWW provides access to abundance of information for the Internet users. Users encounter many problems when interacting with the web (Kosala and Blockel, 2000). Some of them are filtering out relevant information from the search, extracting the existing hidden knowledge and providing individual based information by learning the interest of customers/individual users. Web mining techniques could be used to solve the above problems. The important data mining techniques (Han and Kamber, 2001) applied in the web domain include Association Rule Mining, Sequential Pattern Discovery, Clustering, Classification and Path Analysis. Web mining can be categorized into 3 areas of interest based on the part of the web to be mined: Web Content Mining, Web Structure Mining and Web Usage Mining. The important problem faced by the Internet users is latency. The latency for retrieving a web document depends on several factors like the network bandwidth, propagation time and the speed of the server and client computers. An immediate solution would be to increase the bandwidth. This is not a best solution since, without high investment, the infrastructure of web cannot be easily changed and also higher bandwidth tempts the users to create heavy documents and chocking the network again.

The literature survey indicate that the best solution to reduce latency is caching the web documents at various points in the network (client, proxy and server) using predicting and prefetching the files that are likely to be requested soon when the user is browsing through the currently displayed page. The main advantages of employing predictive prefetching are prevention of bandwidth underutilization and latency reduction. On the other hand, without a carefully designed predictive prefetching scheme several transferred documents may not be used by the client at all, thus, wasting the bandwidth (Pitko and Pirolli, 1999). The web logs utilized for predictive prefetching process contains many unwanted details. Existing algorithms concentrate towards predictive prefetching without giving more considerations for preprocessing of web logs. This in turn leads to some level of time constraint issues to execute the algorithm and thus increases the latency. So, we have developed a new preprocessing algorithm for any kind of server side logs and propose a new predictive prefetching method.

SURVEY OF RELATED WORKS

This study, provides an overview of research work related to web usage mining. Access histories of users visiting a web server are automatically recorded in client

Corresponding Author: G. Arumugam, Professor, Department of Computer Science, University of Madurai Kamaraj, Madurai, Tamil Nadu, India

and server side web access logs using any one of the log formats. Extended common log format is one of the universal server log format. Each entry represents a single request for a resource and contains the following details: <IP address> <Remote log name> <Authenticated User ID> <Date and time of the request> <URL request> <Status code> <Content length of response> <referrer> <Agent>. In this study we make analysis related to server and client side logs.

Padmanabhan (1996) and Sarukkai (2000), Fan *et al.* (1999), Loon and Bharghavan (1997), Cunha and Jaccoud (1997) use dependency graph for prediction and prefetching. Their prediction algorithm constructs a dependency graph that depicts the pattern of accesses to different files stored at the server. The graph has a node for every file that has ever been accessed. There is an arc from node A to B if and only if B was accessed within w (look ahead window size) accesses after A.

Prediction by Partial Match (PPM) model used by Cleary and Witten (1994) for web prefetching makes prefetching decisions by reviewing the URLs that clients on a particular server have accessed over some period. Since the tree records every accessed URL it takes up too much space in the server. This consumes more network bandwidth and increases the traffic overhead. Pitkow and Pirolli (1999) use Longest Repeating Subsequence (LRS) Model as a variation of the PPM model that builds common surfing patterns and regularities. This model assigns long branches to popular URLs that clients access frequently and shorter branches to less popular URLs. As the tree keeps only a small number of less frequently accessed URLs and ignores prefetching for many less frequently accessed URLs and overall prefetching hit rates can be low. Chen and Zhang (2003) use Popularity-Based Model that uses only the most popular URLs as root nodes and follows PPM and LRS models with faster hit ratio reduction.

Using path profiles Schechter *et al.* (1998) and Qiang Yang *et al.* (2001) have stated that the HTTP requests generated by today's dynamic web applications have a surprisingly high level of predictability.Using Keyword-Based Semantic Prefetching Xu *et al.* (2004) presented an approach to tolerate web access latency from the perspective of clients in the context of Internet news services. It exploits semantic preferences of client requests by analyzing keywords in the URL anchor texts of previously accessed articles in different news categories.

We note that all the above works ignored some of the following important issues for the prediction based on server side logs: Analysis of the importance for cleaning process, Analysis of aging factor for pages. Analysis of non_prefetchable items Approaches to prefetch objects that are newly created or never visited before Document size to be cached and cache utilization factors.

We have analyzed the seriousness of document duplication problem using the survey of Albers *et al.* (1999). There is a possibility for a document to reside in the browser's cache, in proxy cache and also in the server while accessing it from a browser. The server access happens when the document is not available locally and in proxy. The server sends the requested page along with prefetching hints using the server side logs. The existing prefetching methods missed to analyze this document duplication problem.

BASIC FRAMEWORK

This study proposes architecture as depicted in Fig. 1 to provide a solution for the problems with the existing predictive prefetching methodology. This architecture has the following processes: Cleaning Process, User Session Identification Process, Pattern Analysis Process, Predictor Engine Process, Prefetching Engine Process and Analyzer Process. The Cleaning process and User Session Identification Process are the preprocessing stages of the web usage mining. The output of cleaning algorithm is processed either by analyzer process to predict the user's next request or by user session identification process to identify the user sessions.

Proposed preprocessing methodologies

Cleaning process: Using web logs we can predict user's next request without disturbing them. But one has to keep in mind that not all details / files available in web logs are appropriate for the purpose of mining navigation patterns. So, the information from the log needs cleaning before it can be used for prediction. The main objective is to identify only the valid and frequently requested HTML documents. Therefore unnecessary files are deleted using Phase 1 and 2 of cleaning algorithm (CLEALG).

Phase 1 algorithm:

- Remove files not containing HTML documents from the user requested page.
- Delete logs having codes other than 200, 304 and 306 with GET method.
- Delete logs generated for extremely long user sessions by search engine.
- Delete logs having no URL in the <URL request>.

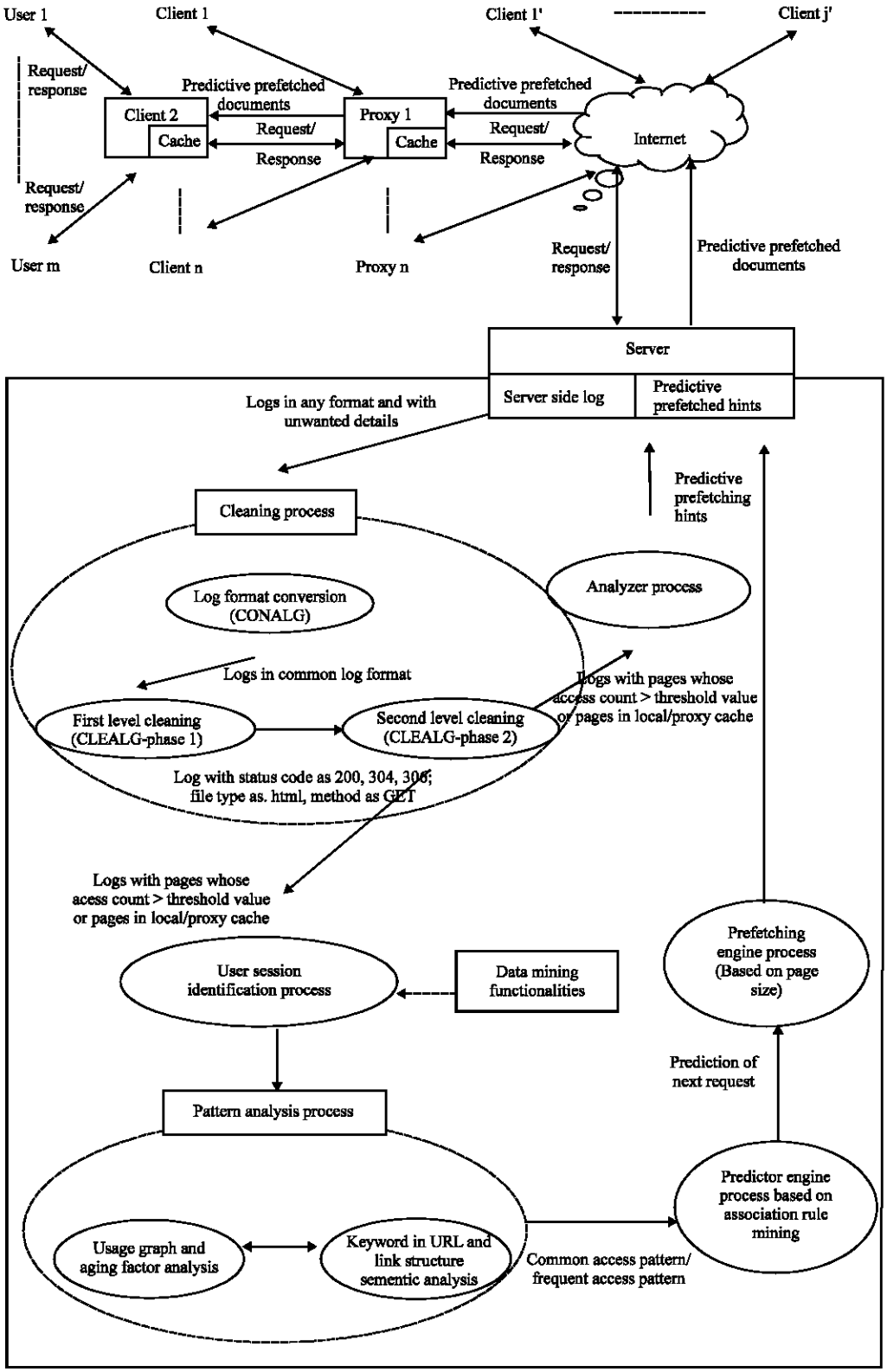

Fig. 1: Basic framework

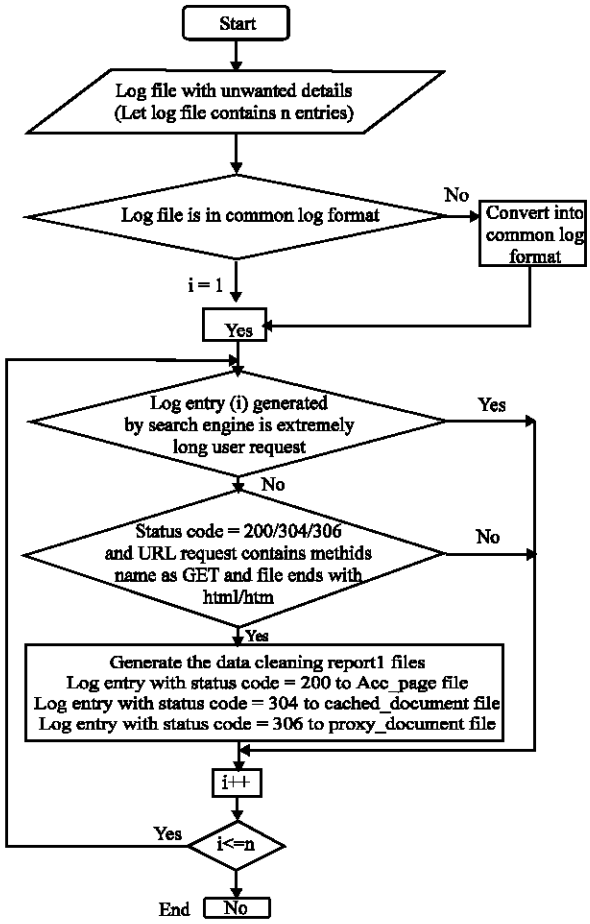

Fig. 2: Phase 1

- Delete entries related to Web robots request and System Request.

The detailed algorithm for Phase1 is depicted in Fig. 2.

Phase 2 algorithm: In phase 2 of CLEALG pages that are accessed less frequently than the minimum threshold value can be deleted if they are not in the browser cache or in the proxy cache. The detailed algorithm for Phase 2 is depicted in Fig. 3.

After the cleaning stage of phase1 only valid HTML documents remain in a request sequence and these are sent as input for phase 2 algorithm. After this phase 2 stage only valid frequently accessed HTML documents remain in the request sequence. These can be used for the user session identification stage and as an input for the Analyzer Process to predict the next user request. In the earlier methods all the processes of phase1 have not been followed and hence there has not been much efficiency in the latency considerations.

User session identification process: There are several ways to identify individual visitors (Kosala and Blockeel, 2000). They are, using cookies, registration details and IP address. The various issues related to the above process are as follows:

- Violations of user's privacy as cookies allow intersession tracking of users.
- In case of user registration many users neglect pages with registration details.
- User IP address is complicated with the issues given below:
  - Existence of local and proxy server caches.
  - Clients web access from multiple hosts.
  - Sharing of one address by many clients with proxy servers.
  - Users not attending the system for long time

So the proposed user session identification algorithm (UIDALG) provides solution to the above issues with IP address.

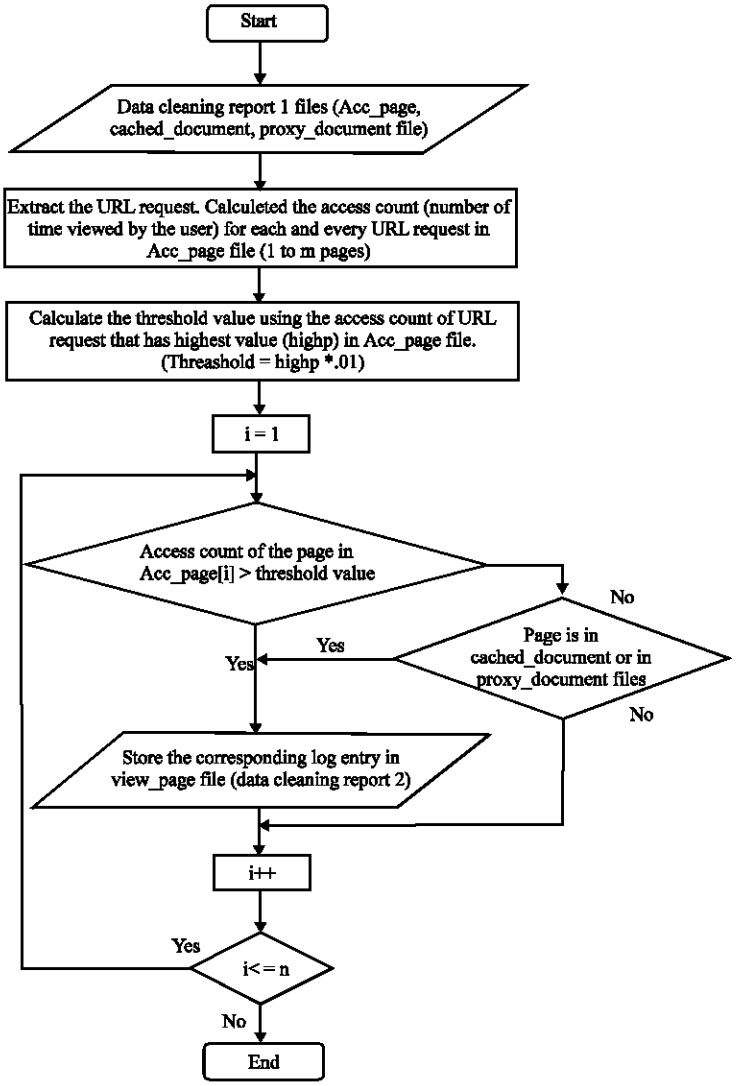

Fig. 3: Phase 2

The algorithm UIDALG: Consider two consecutive log entries (CUR, NEXT):

- If IP addresses are different then these two entries belong to different user sessions.
- Let IP addresses be the same and <Agents> be different. Then these two entries belong to different user sessions.
- Let IP addresses and <Agents> be the same. If the <referrer> page of NEXT log entry and the <URL request> page of CUR are same then declare that they belong to the same user session else do the following analysis:
- Set minimum timeout and maximum timeout. If consecutive accesses (CUR, NEXT) happen within the minimum timeout then they belong to the same user session else if it exceeds the maximum timeout then they belong to two different user sessions.
- If the heuristics of minimum timeout and maximum timeout are not useful for user session identification then use forward link analysis and backward link analysis to check whether the accessed pages are related with each other or not.

Case a: Forward link analysis:

- Check whether the page in <referrer> of NEXT is reachable from the page in <URL request> of CUR using the forward links within w access. Then these 2 entries belong to the same user session.

Case b: Backward link analysis:

- Check whether the page in <referrer> of NEXT is reachable from the page in <URL request> of CUR using the backward links within w access. Then these 2 entries belong to different user sessions.

- If the page in <URL request> of CUR is a
  - Hit Page then Remove the log entries in backward direction from <URL request> of CUR to the log entry with page whose <referrer> is equal to the <referrer> page of NEXT log entry.
  - View Page, it indicates the end of the user session with <URL request> of CUR as last page and the duplication of the log entries in backward direction from log entry with <URL request> as <referrer> of NEXT up to the log entry with <URL request> as home page.

Case c: Accessed pages that are not linked together with same IP address and agent are assumed as different user sessions.

If the user visiting pattern length is one (just visited the home page) then that pattern is removed from the user visiting patterns generated by the UIDALG algorithm. In this research, the algorithm UIDALG was tested with different samples and user visiting patterns are generated. The discovered patterns are analyzed in the next section to construct the usage graph to predict the next user request.

So far we have analyzed various processes involved in the preprocessing phase related to web usage mining. These processes need to be integrated with the other processes in the framework for the efficient pattern identification in the usage mining process. In what follows we briefly discuss various processes in the framework for the predictive prefetching activities.

PROPOSED FRAMEWORK

Frequent pattern identification and predictive prefetching process: The proposed heuristics to find frequent access path patterns using server side logs are based on:

- Probability Usage Graph (Padmanabhan, 1996) constructed from the user visiting patterns.
- Analysis of Keyword in URL (Xu and Ibrahim, 2004) with Link Structure of Server pages.
- Analysis of newly created or never visited pages.
- Analysis of Aging factor.

It is expected to improve the predictability ratio to provide the solution to the following important issues:

The Predictor Engine proposed uses Apriori algorithm to analyze the frequently accessed pages generated by the above algorithm to predict the user's next request. Let W(p) is the predicted future access for page based on above methods. Prefetching Engine gets the documents based on the recommendation by the Predictor Engine when the document size is within some threshold values.

Caching process: Several caching algorithms developed so far have been characterized by the replacement strategy they implement. It is proposed to extend the existing GDSF (Greedy Dual Size Frequency (Cherkasove, 1998)) cache replacement policy. In GDSF the ranking function for an object is computed as $K(p) = L+F(p)*C(p)/S(p)$. When replacing an object in a cache the object with the lowest key K(p) value is removed. This can be considered as making zero order prediction for future access frequencies because they use only the total access count as an estimate for the future prediction. So the ranking function for an object p can be rewritten as: $K(p) = L + (W(p) + F(p)) * C(p)/S(p)$ where W(p) is the predicted future access as in this study. One can also include the factor for handling document duplication problem between proxy server and client (Yang *et al*., 2001; Yang and Zhang, 2003; Xiao *et al*., 2004; Ramaswamy *et al*., 2004) before caching the next expected page.

EXPERIMENTATION AND COMPARATIVE STUDY

Experimentation: Web log cleaning process has been analyzed using five different server data as depicted in Table 1 with the help of CLEALG (Phase 1 and 2).

Experimentation steps

Step 1: In our study all the above server's web logs are in common log format except the sawmill server data. Table 2 contains original sawmill uploaded sample server logs.

So, this is converted into common log format. The conversion algorithm given below does this:

- Read the log file line by line.
- For each line split the fields (IPAddress, Date and Time, etc.) and store these fields into variables.

- Restore the above variables (fields) as per the common log format.

The above algorithm produces the server web log in common log format as shown in Table 3.

Analyzer algorithm (ANAALG) is developed to analyze the server logs. The statistics related to Sawmill server logs is given in Table 4.

In our CLEALG, it is enough to analyze the requests based on Status code and File types. So ANAALG is used to analyze the requests based on Status Code and File Types. Table 5 shows the analysis report of requests based on Status Code and File Types before cleaning the logs by CLEALG.

The Table 5 indicates that the original server logs have so many unwanted details for predictive prefetching process.

Step 2: Phase 1 of CLEALG is applied to produce the first level cleaning as depicted in Fig. 2. The output is:

Acc_page : Pages with status code 200 and file type as .html / .htm

Cached_pages : Pages with status code as 304 and file type as .html/. htm

Proxy_page : Pages with status code as 306 and file type as .html / .htm

Table 6 shows the analysis of requests based on Status Code and File Types for the above 3 cases after the first level cleaning by Phase1 of CLEALG.

Step 3: Now Phase 2 of CLEALG is applied to produce the second level cleaning as depicted in Fig. 3. The output of this phase excludes the pages that are accessed less frequently than the minimum threshold value if they are not the cached document or uses proxy.

Table 7 shows the analysis of requests based on Status Code and File Types for the above 3 cases after the second level cleaning by Phase 2 of CLEALG.

These resultant logs retained the necessary details for predictive prefetching. So it is enough to process 747 logs in Table 6 or 236 logs in Table 7 for the forthcoming stages instead of processing 17,689 logs in Table 5. This in turn improves the efficiency of predictive prefetching algorithm. Detailed bar graph for the analysis of status code for five servers depicted in Table 1 is shown in Fig. 4-8.

The output generated after the first level cleaning for the server depicted in Fig. 8 is shown in Table 8. Since all

Table 1: Sample server web logs details

1.	a) *www.sawmill.net* - Analyzed requests from Wed-26-Apr-2006 00:02 to Sun-30-Apr-2006 23:11 (4.96 days) b) *www.sawmill.net* - Analyzed requests from Wed-24-Apr-2002 00:22 to Tue-30-Apr-2002 23:59 (6.98 days). We analyzed around 7 crore Sawmill server web logs. Among them statistical report for 2 lakhs data are listed in this study.
2.	Philip.greenspun.com - Analyzed requests from Thu-06-Mar-2003 00:00 to Thu-06-Mar-2003 23:59 (1.00 days).
3.	www.jafsoft.com - Analyzed requests from Wed-26-Apr-2000 00:00 to Wed-26-Apr-2000 00:23 (few logs)
4.	www.group.slac.stanford.edu/wim/logfiles/info.html - Analyzed requests from Sat-08-May-1999 12:11 to Sat-08-May-1999 12:25 (0.01 days)
5.	Log Analyzer Sample log - Analyzed requests from Fri-31-Dec-1999 10:11 to Wed-05-Jan-2000 22:11 (5.50 days).

Table 2: Sawmill sample logs (original form)

9120 28/Apr/2006 11:43:15 page view /robots.txt 209.249.86.4 "*http://www.flowerfire.com/seized/reviews/a_game_of_thrones_tane_aikman.htm* " - (not authenticated) 200 10 0 0 0 0 92 b /Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)

9121 28/Apr/2006 11:43:15 page view /seized/reviews/firelord_sara_lipowitz.html 209.249.86.4 "*http://www.flowerfire.com/seized/reviews/a_game_of_thrones_tane_aikman.html*" - (not authenticated) 200 10 0 0 0 0 6.10 k /Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)

9122 28/Apr/2006 11:43:25 page view /seized/reviews/the_fledgling_sara_lipowitz.html209.249.86.4 - -(not authenticated) 200 10 0 0 0 0 4.65 k /Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)

9123 28/Apr/2006 11:43:49 spider/cs497rej/et++/src/ 68.142.250.115 - - (not authenticated) 200 0 10 0 0 0 92 b /Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)

9124 28/Apr/2006 11:44:16 page view /seized/reviews/the_golden_compass_sara_lipowitz.html209.249.86.4 - - (not authenticated) 200 10 0 0 0 0 4.99 k /Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)

Table 3: Sawmill server logs in common log format

209.249.86.4 - - [28/Apr/2006:11:43:15 -0500] "GET /robots.txt HTTP/1.0" 200 92 "*http://www.flowerfire.com/seized/reviews/a_game_of_thrones_tane_aikman.html*" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)"

209.249.86.4 - - [28/Apr/2006:11:43:15 -0500] "GET /seized/reviews/firelord_sara_lipowitz.html HTTP/1.0" 200 610 "*http://www.flowerfire.com/seized/reviews/a_game_of_thrones_tane_aikman.html*" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)"

209.249.86.4 - - [28/Apr/2006:11:43:25 -0500] "GET /seized/reviews/the_fledgling_sara_lipowitz.html HTTP/1.0" 200 465 "-" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)"

68.142.250.115 - - [28/Apr/2006:11:43:49 -0500] "GET /cs497rej/et++/src/ HTTP/1.0" 200 92 "-" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)"

209.249.86.4 - - [28/Apr/2006:11:44:16 -0500] "GET /seized/reviews/the_golden_compass_sara_lipowitz.html HTTP/1.0" 200 499 "-" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)"

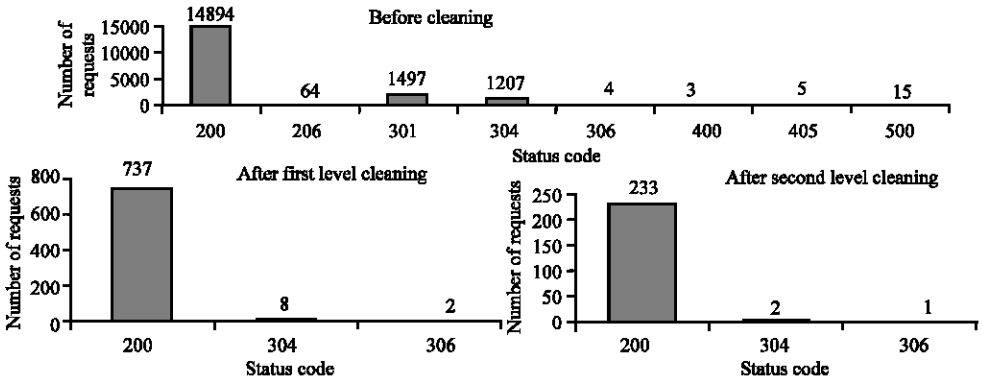

Fig. 4: Sawmill.net cleaning report for the requests from Wed-26-Apr-2006 00:22 to Sun-30-Apr-2006 23:11 (4.96 days)

Table 4: Sawmill server logs overall statistics

Details	Frequency (before cleaning)	Frequency (after first level cleaning)	Frequency (after second level cleaning)
Successful requests	16.165	747	236
Average successful requests per day	3.255	150	46
Successful requests for pages	15.794	737	233
Average successful requests for pages per day	3.181	150	46
Failed requests	23	nil	nil
Redirected requests	1.501	10	3
Distinct files requested	405.	218	49
Distinct hosts served	2.666	283	117
Corrupt log file lines	9.	nil	nil
Data transferred	17.99 MB	573.41 KB	216.19 KB
Average data transferred per day	3.62 MB	115.52 KB	43.58 KB

Table 5: Status code and file type based analysis before cleaning

Reqs	Status code	Reqs	File type / extension
14894	200 OK	15102	[directories]
64	206 Partial content	792	.html [Hypertext Markup Language]
1497	301 Document moved permanently	15	.mov [Quick Time movie]
1207	304 Not modified since last retrieval	1730	[no extension]
4	306 Switch proxy	34	.c
3	400 Bad request	8	.gz [Gzip compressed files]
5	405 Method not allowed	8	.tar.gz [Compressed archives]
15	500 Server error		

Table 6: Status code and file type based analysis after first level cleaning

Pages types	Reqs	status Code	File type
Acc_Pages	737	200 OK	.html
Cached_Pages	8	304 Not modified since last retrieval	.html
Proxy_Pages	2	306 Switch proxy	.html

Table 7: Status code and file type based analysis after second level cleaning

Pages types	Reqs	Status code	Extension
Acc_Pages	233	200 OK	.html
Cached_Pages	2	304	.html
Proxy_Pages	1	306	.html

the pages in Table 8 are accessed only once, the probability for future access to all pages are the same. So, data cleaning report 2 for this sample data is empty. Data cleaning report 2 for the sample data of server depicted in Fig. 9 is also empty.

Similarly, analysis report based on file type is also generated and analyzed. In Fig. 10-14 piechart for the server data before applying CLEALG is depicted. The output of CLEALG Phase1 and 2 consists of only html files. So, we excluded the piechart descriptions for the output generated by the Phase 1 and 2 cleaning algorithms. Since predictive prefetching process needs only html documents this analysis indicates the need for cleaning process.

The complete chart of file type and status code analysis reports are depicted in Table 9. Thus, the experimental result shows the importance for cleaning of web logs.

Comparative study: Our literature survey covers the research papers related with predictive prefetching from 1994-2004. Very few papers have analyzed about the cleaning process. The maximum level of cleaning applied with the methods proposed in the study (Padmanabhan, 1996; Cleary and Witten, 1994; Pitkow and Pirolli, 1999; Cheng and Zhang, 2003; Schechter *et al*., 1998; Yang *et al*., 2001; Xu and Ibrahim, 2004; Sarukkai, 2000; Fan *et al*., 1999; Loon and Bharghavan, 1997; Cunha and Jaccoud, 1997), is removal of logs with other materials such as .gif, .css, .jpg, .js, etc and removal of logs if their status code is 4xx/5xx/3xx/1xxx/1xx. Let us name it as PREVALG. The main difference between the PREVALG with CLEALG is listed below:

- CLEALG stores logs whose status code is 200 and access method is GET but PREALG stores logs with status code 2xx and any kind of access method.

Table 8: Web logs after the first level cleaning for the server "Group.slac.stanford.eu"

1Cust216.tnt1.santa-monica.ca.da.uu.net - - [08/May/1999:12:15:19 -0700] "GET /grp/pao/seminar.html HTTP/1.0" 200 2766 "http://www.slac.stanford.edu/slac/sciinfo.shtml" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)"
1Cust216.tnt1.santa-monica.ca.da.uu.net - - [08/May/1999:12:15:37 -0700] "GET /grp/pao/semsearch.html HTTP/1.0" 200 4421 "http://www.slac.stanford.edu/grp/pao/seminar.html" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)"
1Cust216.tnt1.santa-monica.ca.da.uu.net - - [08/May/1999:12:19:02 -0700] "GET /detailed.html HTTP/1.0" 200 33577 "http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)"
1Cust216.tnt1.santa-monica.ca.da.uu.net - - [08/May/1999:12:20:53 -0700] "GET /gen/meeting/ssi/next/index.html HTTP/1.0" 200 9887 "http://www.slac.stanford.edu/gen/meeting/ssi/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)"
1Cust216.tnt1.santa-monica.ca.da.uu.net - - [08/May/1999:12:22:06 -0700] "GET /grp/pao/tour.html HTTP/1.0" 200 7263 "http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /grp/pao/tour.html - "HTTP/1.0"
1Cust216.tnt1.santa-monica.ca.da.uu.net - - [08/May/1999:12:24:25 -0700] "GET /welcome/mission.html HTTP/1.0" 200 1671 "http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)"
1Cust216.tnt1.santa-monica.ca.da.uu.net - - [08/May/1999:12:24:42 -0700] "GET /gen/edu/education.html HTTP/1.0" 200 13095 "http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)"

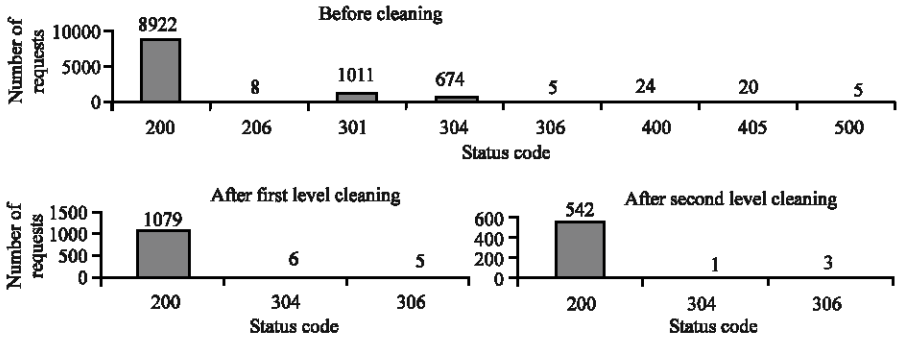

Fig. 5: Sawmill.net, cleaning report for the requests from Wed-24-Apr-2002 00:22 to Tue-30-Apr-2002 23: 59 (23: 59days)

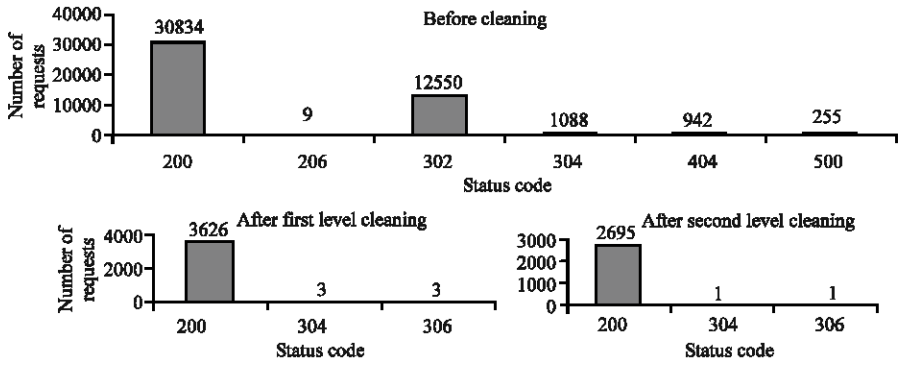

Fig. 6: Philip.greenspun.com cleaning report for the requests from Thu-06-Mar-2003 00: to Thu-06-Mar-2003 23:59 (1 days)

- PREVALG neglects the information supplied by the web logs with status code 304 and 306. But, our CLEALG stores and proves the necessity of storing cached document (304) and proxy level document (306) details.
- Logs for extremely long user sessions that are generated by search engines are also deleted in CLEALG. If there is no URL in the URL request of log that can also be deleted in CLEALG. But these details are retained in PREVALG.
- CLEALG keeps the page if it is accessed frequently than the minimum threshold value or in the browser cache or in the proxy cache. This improves the predictability ratio. In PREVALG there is no such analysis.

To prove the efficiency of CLEALG than the PREALG, number of bytes to be processed to predict the next request is taken into account. Figure 15 shows the comparative analysis between CLEALG and PREALG.

Consider the sawmill server log requests from Wed-26-Apr-2006 00:02 to Sun-30-Apr-2006 23:11 (4.96 days). Initially, number of bytes to be processed to predict the next requests in this sever is 17990 KB. But, after PREVALG (existing algorithm) is applied with that server

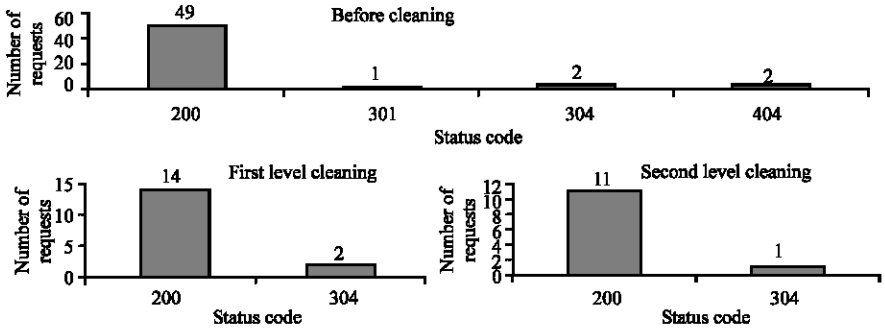

Fig. 7: Log analyzer-sample log, cleaning report for the requests from Fri-31-Dec-1999 10:11 to Wed-05-Jan-2000 22:11 (5.50 days)

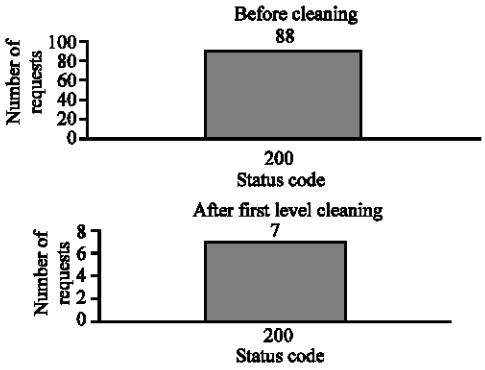

Fig. 8: Group.slac.stanford.eu, cleaning report for the requests from Sat-08 Ma-1999 12:11 to Sat-08-May-1999 12:25 (0.01 days-few logs)

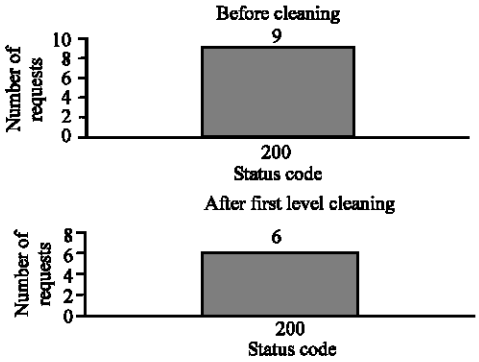

Fig. 9: www.jafsoft.com, cleaning report for the requests from Wed-26 Apr-2000 00:00 to Wed-26-Apr-2000 00:23 (0.02 days-few logs)

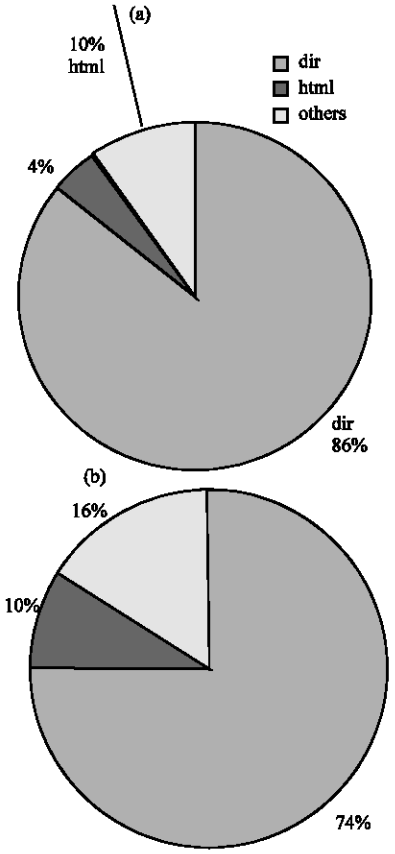

Fig. 10 (a,b): Analysis report based on file type before cleaning, www.sawmill.net analyzed requests from Wed-26-Apr-2006 00:02 to Sun-30-Apr-2006 23:11 (4.96 days)

logs, it is enough to process 660 KB of data (without cached documents details). CLEALG phase1's output stated that 573 KB of data (including cached documents) details to be processed for predictive prefetching process and CLEALG phase 2's output stated that it is enough to process 216 kb (almost half of PREVALG) data for predictive prefetching process. In general, CLEALG reduces the amount of data to be processed by the prefetching algorithm into more than half of PREVALG, while at the same time it maintains all the necessary details and additional supportive information like access count for pages, browser and proxy level cached document details for predictive prefetching process.

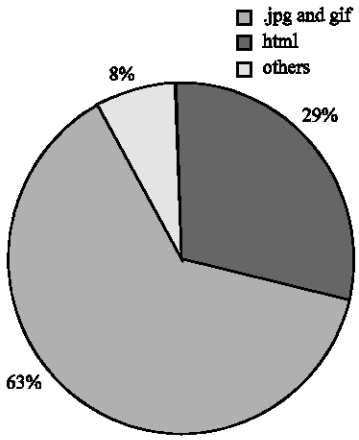

Fig. 11: www.sawmill.philips.green pun.com analyzed requests from Thu-06-Mar-2003 00:00 to Thu-06-Mar-2003 23:59 (1.00 days)

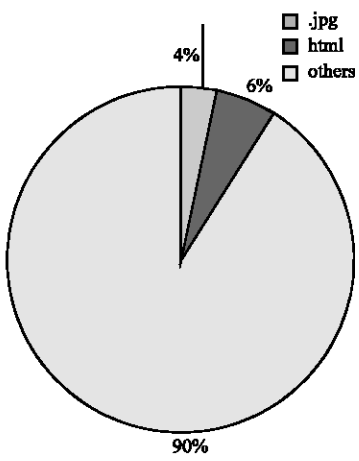

Fig. 13: Log analyzer sample log analyzed requests from Fri-31-Dec-1999 10:11 to Wed-05-Jan-2002 22:11 (5.50 days)

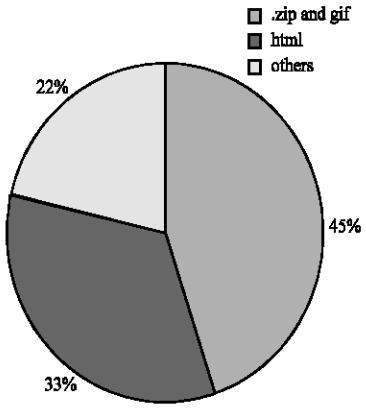

Fig. 12: www.sawmill.jsoft.com analyzed requests from Wed-26-Apr-2000 00:00 to Wed-26-Apr-2000 00:23 (0.02 days)-feew log

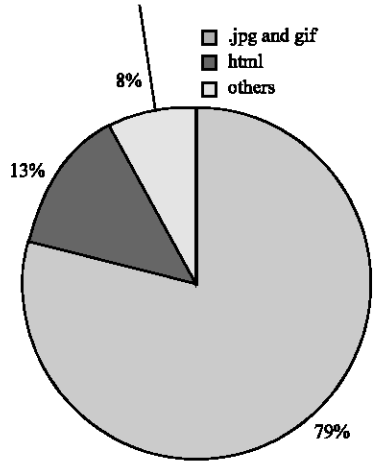

Fig. 14: www.group.slac.stanford.edu.wim/log files analyzed requests from Sat-08-May-1999 12:11 to Sat-08-May-1999 12:25 (0.01 days)

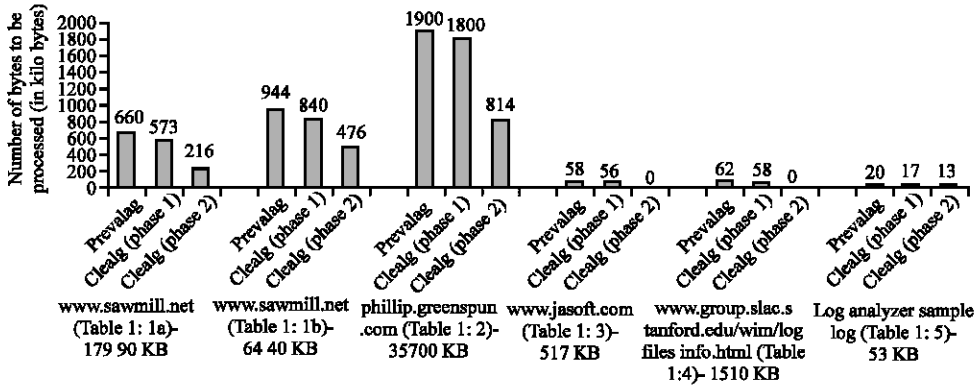

Fig. 15: Comparitive chart between exisiting cleaning algorithm PREALG and our algorithm CLEALG for the server data listed in Table 1-based on number of bytes to be processed for predictive prefetching process

Table 9: Analysis report based on status code and file type

Server name	Status code report						File type report					
Log Analyzer Sample Log - Analyzed requests from Fri-31-Dec-1999 10:11 to Wed-05-Jan-2000 22:11 (5.50 days)	Before cleaning		First level cleaning		Second level cleaning		Before cleaning		First level cleaning		Second level cleaning	
	Req.	Status code	Req.	Status code	Req.	Status code	Req.	File type	Req.	File type	Req.	File type
	49	200	14	200 OK	11	200	49	.html	16	.html	12	.html
	1	301	2	304	1	304	2	.jpg				
	2	304					3	.other				
	2	404										
www.group.slac.stanford.edu/wim/logfiles/info.html- Analyzed requests from Sat-08-May-1999 12:11 to Sat-08-May-1999 12:25 (0.01 days)	88	200	7	200	Nil	Nil	13	.jpg	7	.html	Nil	Nil
							57	.gif				
							6	.html				
							12	.other				
http://www.jafsoft.com/ - Analyzed requests from Wed-26-Apr-2000 00:00 to Wed-26-Apr-2000 00:23 (0.02 days)	9	200	2	200	Nil	Nil	1	.zip	2	.html	Nil	Nil
							2	.html				
							3	.gif				
							1	.jpg				
							2	.other				
philip.greenspun.com - Analyzed requests from Thu-06-Mar-2003 00:00 to Thu-06-Mar-2003 23:59 (1.00 days)	30834	200	3626	200	2695	200	10257	.jpg	3632	.html	2697	.html
	9	206	3	304	1	304	3141	.gif				
	12550	302	3	306	1	306	3633	.html				
	1088	304					8189	no ext				
	942	404					1015	.ps				
	255	500					4079	.dir				
							15364	.other				
www.sawmill_net - Analyzed requests from Wed-26-Apr-2006 00:02 to Sun-30-Apr -2006 23:11 (4.96 days)	14894	200	737	200	233	200	15102	.dir				
	64	206	8	304	2	304	792	.html	747	.html	236	.html
	1497	301	2	306	1	306	15	.mov				
	1207	304					34	.c				
	4	306					8	.gz				
	3	400					1738	.other				
	5	405										
	15	500										
www.sawmill.net - Analyzed requests from Wed-24-Apr-2002 00:22 to Tue-30-Apr -2002 23:59 (6.98 days).	8922	200	1079	200	542	200	7901	.dir	1090	.html	546	.html
	8	206	6	304	1	304	1103	.html				
	1011	301	5	306	3	306	85	.pl				
	674	304					216	.no ext				
	5	306					2	.qt				
	24	400					184	.class				
	20	408					33	.java				
	5	500					1145	.other				

CONCLUSION

The analysis indicated that the existing web technology faces so many problems. Some of the problems are addressed in this study. One among them is latency. It has been proved that the predictive prefetching is the best solution to reduce the latency. Our algorithm for cleaning web logs improves the predictive prefetching algorithm by providing efficient web logs. In the forth-coming papers prefetching algorithms are to be modified according to the above preprocessing algorithms and thus caching algorithms may be modified according to this predictive prefetching factor.

REFERENCES

Albers, S., S. Arora and S. Khanna, 1999. Page Replacement for General Caching problems. Proc. 10th Ann. ACM-SIAM symp. Discrete Algorithms (SODA), pp: 31-40.

Chen, X. and X. Zhang, 2003. A Popularity-Based Prediction Model for Web Prefetching. Published by the IEEE Computer Society, http://www.cs.wm.edu/hpcs/WWW/HTML/publications/papers/TR-03-1.pdf.

Cherkasova, L., 1998. Improving WWW Proxies Performance with Greedy Dual Size Frequency Caching Policy. HP Technical Report.

Cleary, J.G. and I.H. Witten, 1994. Data Compression Using Adaptive Coding and Partial String Matching. IEEE. Trans. Commun., 32 (4): 396-402.

Cunha, C.R. and C.F.B. Jaccoud, 1997. Determining WWW User's Next Access and its Applications to perfetching. Proc. Int. Symp. Comput. Comm., pp: 6-11.

Fan, L., P. Cao, W. Lin and Q. Jacobson, 1999. Web Prefetching between Low_Bandwidth Clients and Proxies Potential and Performance. Proc. Sigmetrics, pp: 178-187.

Han, J. and M. Kamber, 2001. Copy right 2001 by Academic Press, Data Mining Concepts and Techniques. Published by Elsevier, a division of Reed Elsevier India Private Limited, New Delhi-110 024, India, Original ISBN: 1-55860-489-8.

Loon, T.S. and V. Bharghavan, 1997. Allevating the lantency and Bandwidth problems in WWW Browsing. Proc. USENIX Symp. Internet Technol. Sys., pp: 219-230.

Kosala, R. and H. Blockeel, 2000. Department of computer science. Katholieke Universiteit Leuven, Celestijnenlian 200A, B-3001 Heverlee, Belgium. Web mining Research: A Survey, http://arxiv.org /PS_cache/cs/pdf/0011/0011033. pdf.

Pitkow, J. and P. Pirolli, 1999. Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. Proc. Usenix Technical Conf. Usenix, pp: 139-150.

Padmanabhan, V.N., 1996. University of California at Berkeley and Jeffrey and C. Mogul, Digital Equipment Corporation, Western Research Laboratory, Using Predictive Prefetching to Improve World Wide Web Latency. Comput. Comm. Rev., 26 (3): 22-36.

Sarukkai, R., 2000. Link Prediction and Path Analysis using Markov Chains. Proc. 9th Int. World Wide Web Conf., 2000. Kosala, R. and H. Blockeel, Department of computer science, Katholieke Universiteit Leuven, Celestijnenlian 200A, B-3001 Heverlee, Belgium, Web mining Research: A Survey.

Ramaswamy, L. and L. Liu, 2004. An Expiration age-based document placement scheme for cooperative web caching. IEEE. Trans. Knowledge and Data Eng., 5 (16): 585-599.

Schechter, S., M. Krishna and Michael D. Smith, 1998. Using path profiles to predict HTTP requests. Proc. 7th Int. World Wide Web Conf. Also Appeared in Comput. Networks and ISDN Sys., 20: 457-467.

Yang, Q., H.H. Zhang and T. Li, 2001. Mining Web Logs for prediction Models in WWW Caching and Prefetching. 7th ACM SIDDD Intl Conference on Knowledge Discovery and Data Mining KDD, Sanfrancisco, California, USA.

Yang, Q. and H. Zhang, 2003. Web-log mining for predictive web caching. IEEE. Tran. Knowledge and Data Engineering, Vol. 14, No. 15.

Xu, C.Z., I. Ibrahim, Keyword-Based Semantic Prefetching Approach in Internet News Services. IEEE. Trans. Knowledge and Data Eng., 16 (5): 601-611.

Xiao, L., X. Zhang, A. Andizejak, and S. Chen, 2004. Building a large and efficient hybrid peer-to-peer internet caching system. IEEE. Trans. Knowledge and Data Eng., 6 (16): 754-768.