Domain Based Prefetching in Web Usage Mining

Dr. M. Thangaraj[1]
Associate Professor, Dept. of Computer Science
Madurai Kamaraj University
Madurai, India

Mrs. V. T. Meenatchi[2]
Lecturer, Dept. of Computer Applications
Thiagarajar College
Madurai, India

***Abstract*—In the current web scenario, the Internet users expect the web to be more friendly and meaningful with reduced network traffic. Every end user needs the channel with high bandwidth. In order to reduce the web server load, the access latency and to improve the network bandwidth from heavy network traffic, a model called Domain based Prefetching (DoP) is recommended, which uses the technique of General Access Pattern Tracking. DoP presents the user with several generic Domains with the top visited web requests in each Domain, which are retrieved from the web log file for future web access.**

Keywords—Latency; Domain; Prefetching; bandwidth; Network Traffic; Web Log File

I. INTRODUCTION

With the unprecedented growth of web, the users always perceive access latency. Intensive measures have been attempted to reduce the Latency. Prefetching is one such approach to reduce the average web access latency. Web Prefetching mainly deals with the ability to identify objects to be pre-fetched in advance. Prefetching is a complementary technique to Caching, which prefetches web documents, that tend to be accessed in near future, while the client is processing the previously retrieved web documents. Various studies have proposed mostly on History based Prefetching.

The interesting and useful access patterns can be analysed and discovered only when web usage data of the user is tracked. This can be achieved only through a branch of web mining, called Web Log Mining. An experiment with a Web Log File of an Educational Institution for predicting the future web requests is attempted here.

This study is divided into five sections each of which deals with a specific issue: Section 1 introduces the subject matter while Section 2 examines various issues associated with Prefetching. Section 3 deals with the Architecture and components of DoP while Section 4 presents the Experimental study and Performance Analysis and finally Section 5 records the concluding remarks.

II. RELATED WORK

Despite the rapid technological advancement in achieving high speed, users demand keeps growing for reducing the access latency. Some of the contributions based on Prefetching, focus on semantic locality, while several do not concentrate on content semantics and specific Domain categorization approach.

The following works do not concentrate on content semantics:

In order to improve the performance of client web object retrieval, the current web page's view time was used for acquiring the web objects of the future web pages. Markov Knapsack method as in [1] was used to define web application Centric Prefetching approach, which restricted the hyperlink domain of webpages to the web application. Though the model accurately represents the client's behavior, considering only the view time of the web page, it is not a wholesome approach.

The importance of preprocessing in Web Usage Mining and the format of the Server Log File is depicted in [14]. Learning algorithm called Fuzzy-LZ as in [7] mines the history of user access and identifies patterns of recurring accesses. To make prefetching decision, a prefetching algorithm based on Neural Network called Adaptive Resource Theory (ART) as in [18] uses bottom-up and top-down weights of the cluster-URL connections.

Various evaluations of analysis of Prefetching performance from user's perspective as in [11] is discussed and the author emphasizes the adaptation of prediction algorithm to the environment conditions. Graph based clustering algorithm as in[10] identifies the clusters of correlated web pages based on the users access pattern in order to improve the proxy server's performance. A group of Prefetching algorithms were reviewed as in [4] based on Popularity, Good Fetch, APL characteristics and Lifetime.

Sequential web access pattern mining as in [20] stores frequent sequential web access patterns in a Pattern tree. The web links generated through Pattern tree are used for recommendations, but they do not concentrate on Domain. User sessions are identified as in [3] and the web logs are cleaned. The user session sequence is generated through Maximum Forward Reference method. The study is defective as it does not focus on semantic locality and the user session sequence is not classified based on their web usage.

A web prefetching algorithm as in [9], particularly concentrated on user's perspective, which analyses the perceived latency with traffic increase and concludes that most likely predicted pages reduce latency. An intelligent solution to caching was proposed as in [17] to improve QOS of websites. It analyses the historical navigation of the website in log file using frequent closed item sets.

Web-object prediction model was developed as in [16] to empower the prefetching engine. It is built by mining the frequent paths from past web log data. Page Rank based Prefetching technique for accessing web page clusters as in [19] deals with the link structure of a requested page and determines the most important linked pages and also identifies the pages to be pre-fetched.

User behavior as in [21] is represented by sequence of consecutive web page accesses from proxy server access log. Indexing methods are used to organize the frequent sequences of the log. The introduction of semantics yields better results. The following citations perform prefetching based on semantics:

Reference [5] introduces a technique which predicts future requests based on Semantic preferences of past retrieved documents in a News Agent Prefetching system. The system extracts the document semantics by identifying keywords in their URL anchor texts. The anchor text for a current web page is associated with so many keywords. Need for more space to store a large set of keywords makes this approach disadvantageous. Selective Markov models as in [15] uses semantic information to prune its states in high order. The system uses semantic distance matrix to store all semantic distances among n webpages in the sequential database. A solution based on Sematic Web Mining was defined in [12] for the Website Key Object problem.

A Website core Ontology was represented for Web user interests. The drawback of the system is that the userr interests may change over the time period. Several methods of prefetching is explained in [13]. Basic scheme of Semantic Prefetching system is discussed. The paper [8] discusses how Semantic Web Mining improve the results of Web Mining.

A Semantic link Prefetcher as in [2], uses the current web page's hyperlink set to trace objects to be pre-fetched during the view time of the current webpage.

Reference [6] uses keyword based semantic prefetching in Internet News. It has taken the News domain alone for prediction. The system analyses the keywords found in the anchor tag for making semantic preferences. That system is known as the keyword method, which is taken up for comparative study.

The following section examines the proposed work, which overcomes the above mentioned problems.

III. PROPOSED WORK

Here a new architecture termed as Domain based Prefetching (DoP), as shown in Fig.1 is proposed.

A. DoP Architecture – An Overview

DoP architecture contains the following four main phases:

1) *PREPROCESS PHASE*
2) *CATEGORIZE PHASE*
3) *ONTOMAP PHASE*
4) *PREFETCHING PHASE*

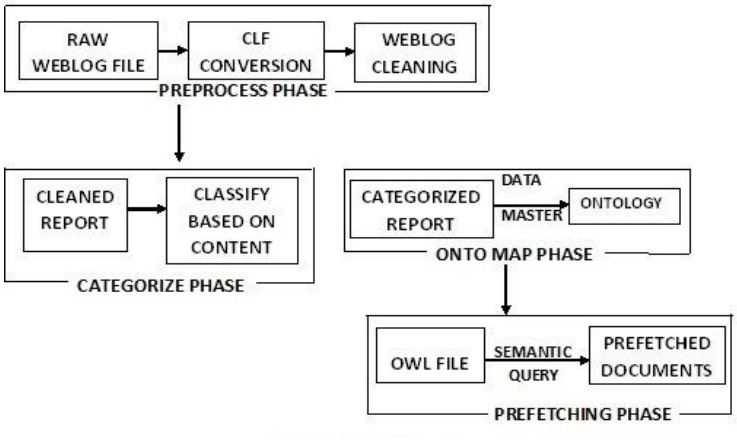


Fig. 1. DoP Architecture

The details of the phases are as follows:

1) *PREPROCESS PHASE:*

This phase concentrates on two main components, DoP Conversion and DoP Cleaning of the web log file.

a) DoP CONVERSION:

The web log file cannot be used as such. With the raw format of the web log file, no useful process can be executed. Of the various web log conversion formats available, the conventional one is the Common Log Format.

Since, the web server's log file does not follow any uniform format for storing entries, it needs to be converted into a format, which would be useful for further processing. This component accepts the raw web log file as its input and converts it into Common Log Format (CLF). Fig. 2 shows the

4030/Apr/201207:14:37pageview/cs497rej/et++/index.html 67.8.221.107 http://www.suksh.com/suk.html - (not authenticated) 200 1 0 0 0 0 0 3.21 k 71/Apr/2012 00:05:21 hit /seized/picts/(nonpage) 4.159.119.132 http://www.flowerfire.com/seized/reviews/blackmantle_sara_lipowitz.html - (not authenticated) 200 0 0 0 0 0 0 .34k 184 30/Apr/2012 09:10:40 page view /cs497rej/et++/index.html 127.8.21.10 - (not authenticated) 200 1 0 0 0 0 0 4.45 k 4 1/Apr/2012 00:04:26 spider /cs497rej/et++/src/(nonpage) 68.142.249.10 - - (not authenticated) 200 0 1 0 0 0 0 1.38 k 40 30/Apr/2012 07:14:37 page view /cs497rej/et++/index.html 17.8.221.107 http://gate.iitd.ac.in/jam.html - (not authenticated) 200 1 0 0 0 0 0 6.12 k 914 30/Apr/2012 17:14:20 page view /cs497rej/et++/index.html 67.8.21.17 http://www.travelsupermarket.com/c/cheap-flights/india.html - (not authenticated) 200 1 0 0 0 0 0 5.11 k 840 30/Apr/2012 12:08:17 page view/cs497rej/et++/index.html 67.8.221.107 http://photobucket.com/pic.html - (not authenticated) 200 1 0 0 0 0 0 3.21 k

sample web log file.

Fig. 2. Sample Web Log File

b) DoP CLEANING:

Web requests include spiders, web robots, files with different extensions other than html, the log entries generated for extremely long user sessions, log entry without proper URL address and requests with status code other than 200, 304, 306 with GET method.

Those web requests need not be considered for further processing and they need to be removed, since they are not useful in mining meaningful knowledge.

2) CATEGORIZE PHASE:

The purpose of this phase is to categorize the web log entries. The Categorization has been done for Semantic Prefetching. From the cleaned web log entries, the URL part is extracted and it is classified based on its content from the corresponding html file through the meta tag. i.e.,<meta name="description"content="">.

Categorization process always needs classifier or class label to perform classification. Here, classifier is the predefined domain name like News, Education, Shopping, Mail. Every entry that is being categorized is placed under the specified domain.

3) ONTOMAP PHASE:

This phase focusses on mapping the classified domains into the Web Ontology file, owl file. This is done through the configured plug in called Data Master of the Protege tool. Ontology will therefore contain the URL and its frequency is termed as HIT under its Domain name.

4) PREFETCHING PHASE:

The ranking of the web request is carried out by taking URL and the hit rate as the sort keys. Ranked web requests under each domain are stored. The prefetch list and purge list are maintained based on the Threshold value, which is based on the value of the hit rate.

All Phases of DoP architecture are interdependent. The basic work flow diagram is presented in Fig. 3, which clearly depicts the placement of Prefetching system in the proxy server.

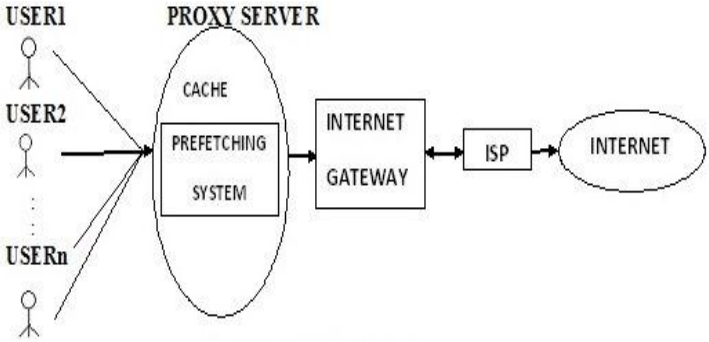


Fig. 3. Work Flow Diagram.

The prefetching system contains the popular web requests, predicted for every domain. User requests which match the predicted requests in the near future might be served from the proxy, without disturbing the original web server, which ensures reduction of the server load and access latency.

The detailed work flow diagram is shown in Fig.4.

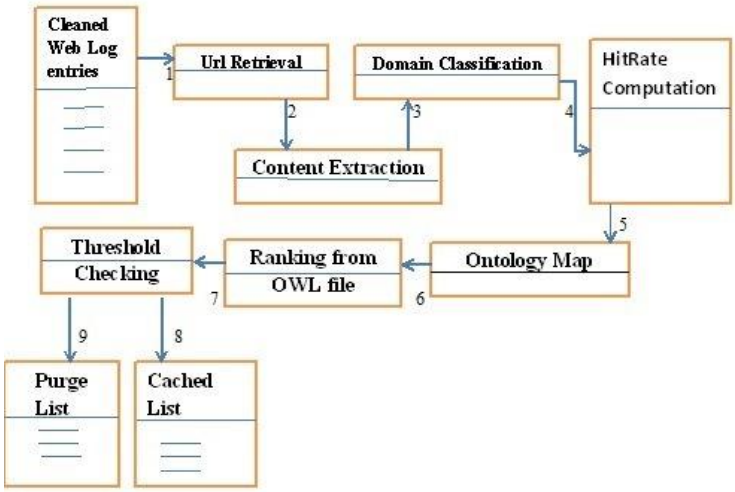


Fig. 4. Detailed Work Flow Diagram

B. Algorithms

The Categorization algorithm- "Categorize" takes the cleaned log file as an input and produces the web ontology file as an output.

Every web request from the cleaned log file is being scanned and the URL segment is tokenized. From the <meta content tag> of the web request, the keywords are fetched and checked with the predefined keyword list. Once there is a keyword match, the web request's URL is stored under its domain, which are then mapped into the Ontology to create 24 distinct classified domains with their corresponding web request's URL which is then mapped into the Ontology.

```
Algorithm      : Categorize(cl,ol)
Data Structure : Table
Input          : cl – Cleaned Log File
Output         : ol – owl File containing classified Domains.
/* i  : 1<=i<=rq  (rq- web entries)
   n  : Total no. of Domains
   url : http request
   Dk  : Array of stored keywords
   K1  : Array of extracted keyword
   hl  : html file of the request
   D   : Domain Table
   Ol  : web ontology file
   kw  : keywords list.*/
for each rq i in cl
{
 tokenize rqi →url;
}
 for each hl of rqi →url
{
  add rqi → kw from <meta name="description" content=" "> to k1[];
for each Dk in n
{
```

```
if (Dk[] = = k1[])
add rqi →url to D;
add D to ol;
}
}
```

The Prefetch algorithm takes web ontology file as its input and produces prefetch cache and purge list as the output. The classified domain contains large numbers of related web requests, of same type. For those entries, the frequency count is computed and stored as the hit rate. The web object's i.e., the url with the corresponding hit rate is then ranked based on the hit rate as the sort key. The sorted web requests are stored under its corresponding domain.

```
Algorithm: Prefetch(ol,dc,pl)
Input    : ol - Web Ontology file with Domains
Output   : dc - cached requests;   pl - purge list
/*url    : http request
min_th   : minimum threshold value
D        : Domain containing classified requests
hr       : hitrate
freq_ct  :function to compute the frequency of web requests
u[]      : Array of http requests
n        : Number of url in D*/
for each url in D
{
  hr  = freq_ct(url);
  sort(url,hr);
  add m to ol;
for each url →hr in ol
{
 if (url →hr <= min_th)
  add url →hr to pl;
else
 add url →hr to dc;
}
}
freq_ct(url)
{
 cn:=0;
for each url in D
{
 if(url == u[])
  cn++;
}
  return;
}
sort(url,hr)
{
m := urli →hr;
for each i in n
{
for each urli in D

{
  if (urli+1→hr > m)
  m:= urli+1 →hr;
}
}
return;
}
```

Threshold value is based on the web object's hit rate. The web object which exceeds the minimum threshold value is stored in the prefetch cache while others are stored in the purge list.

Maintenance of prefetch cache and purge list enables the prefetch cache to contain the most popular web requests and enables the purge list to check periodically with the stored purge list. This is done to permanently remove some web requests, which are consistently retained in the least rank. The purge list is maintained to improve the cache efficiency, since cache can hold only limited web objects.

IV. Experimental Results

DoP approach has been implemented with the use of JAVA, Protégé. The set of experiment explores the web log entries with its various attributes on performance. All experiments were done in Intel Core i5 2.67 GHz with 4 GB RAM, running Windows 7. As an input dataset, the Web Log file of an Educational Institution was analysed. The Log file contained around 1,80,000 entries , collected for a period of 1 year period.

The objectives of the experiments are as follows:

- To improve the proxy server's efficiency. This in turn will reduce the web server load.
- To reduce the user access latency, since the predicted requests are served from the cache, when user request is matched.
- The DoP system suggests the top popular websites in each domain. The web requests under each domain gives clarity to the user when surfing the web.

A. Performance Metrics:

To reduce the access latency, the following four main metrics are vital for prefetching. They are Hit rate, ByteHitRate, Waste Ratio and Byte Waste Ratio.

- HitRate: The percentage of the requested objects serve from prefetching cache.
- ByteHitRate: The percentage of the requested objects serve from the prefetching cache in terms of size.

- WasteRatio : The percentage of undesired documents in the prefetching cache.
- ByteWasteRatio: The percentage of undesired documents in the prefetching cache in terms of size.

The Coverage and Accuracy metrics are also employed.

- Coverage: It is the measure to evaluate the efficiency of prefetcher in satisfying the future object request demand.
- Accuracy: It is the measure of the total prefetched objects, actually used to satisfy the user requests from the prefetched objects.

B. Equations:

Domainwise Coverage is calculated by using the formula given in (1),

$$C_i = C_n / n \quad (1)$$

where,

C_i is the coverage metric.

C_n is the total number of objects in the specific domain d_i.

n is the total number of web objects in cleaned log file.

Domain wise Accuracy is computed as given in (2),

$$A_i = u(d_i) / C_n \quad (2)$$

where, $u(d_i)$ is the total number of objects used in each domain.

The Hit rate percentage(*hr)* is computed as given in (3),

$$hr = 100 * A_i \quad (3)$$

where, A_i is the Accuracy.

Parameters taken for our study is given in Table 1.

TABLE I. PARAMETERS USED IN DOP SYSTEM.

Parameter Name	Description
Wl	Web log file ranges from 1 to 1,80,000 entries.
d_n	Domain Name is of string value (News, Education, Advertisement)
N	Total no. of domains, for the study is 24, which may be increased
Ul	http request of the log file.
Pc	prefetch cache, file that stores the popular web requests
Pl	purge list, file that stores the web requests, to be removed after threshold consideration.
min_th	Minimum threshold value based on hit rate.
max_th	Maximum threshold value based on hit rate.

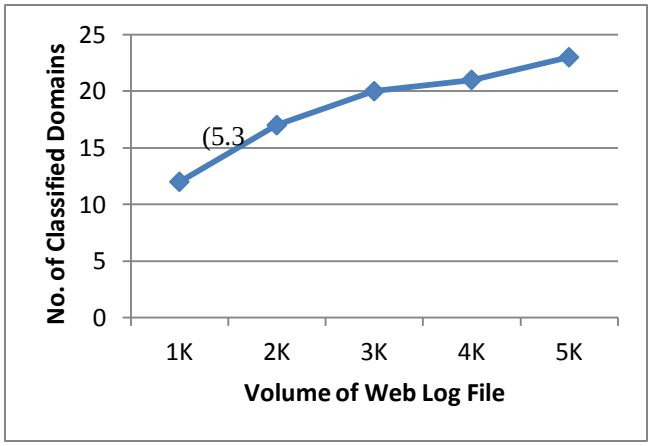
Fig. 5 presents the data size of the web log file with the variation in Throughput. Throughput is the time measured in millisecond, which includes the total time taken for the Log file cleaning, CLF conversion, Log entries categorization, Ontology mapping and Prefetching.

Fig. 5. Throughput Analysis

(5.1)

Categorization efficiency is achieved only when the log entries are correctly classified under its domain.

Fig.6 shows the no. of classified domains with the corresponding log entries. This study has 24 fixed domains for Categorization.



(5.3

Fig. 6. Categorization Analysis

Fig. 7 clearly shows the distribution of the web requests hit rate. This is processed from the whole log file. Since it is a Educational Institution Log file, major distribution is towards Education category and Job Search.

From this visualisation, one could easily find the top most popular domain and the least used one. 10%of Others category shows the ratio of the unclassified web requests with the total web requests. The reduced percentage in Others category reveals the categorization effectiveness.

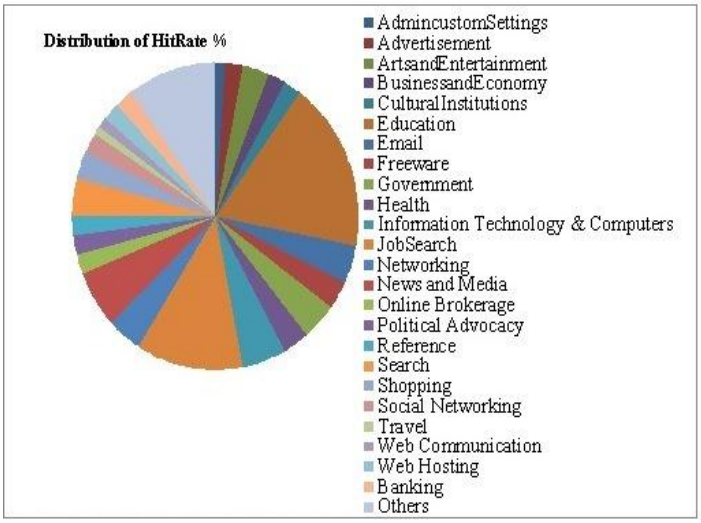


Fig. 7. Overall hitrate distribution for 24 domains

B. Comparitive Study:

In this section, DoP method is compared with the KW method, which considers the News domain. To be generic, the proposed system takes 24 domains into account. Major 4 metrics of prefetching were considered for comparing the DoP method with Keyword based method. Fig. 8 shows the comparative study of DoP with Keyword based method in terms of Hit ratio.

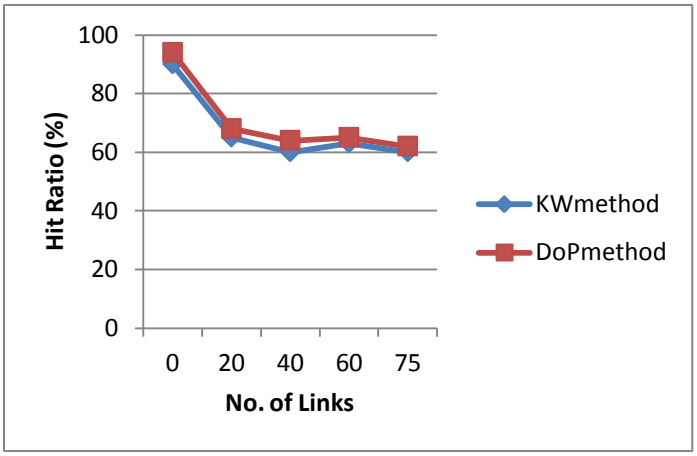


Fig. 8. Growth of Hit Rate against No.of Links.

From Fig. 8, considerable improvement in hit rate of DoP method is clearly learnt. Fig. 9 shows the comparison of DoP with Keyword based method in terms of byte hit rate. The byte hit rate is based on individual web object size. The increased percentage in byte hit rate of DoP method, shows that large number of objects have been requested and fetched from the web log file.

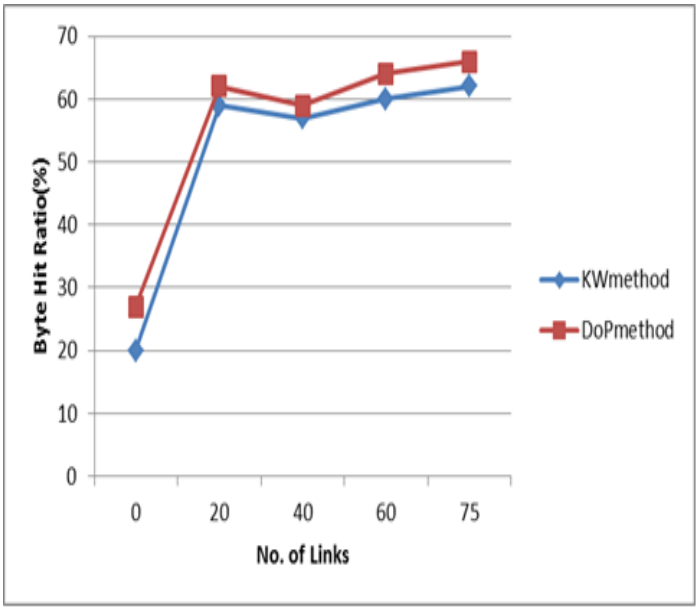


Fig. 9. Performance on ByteHinRate

In DoP method, the no. of undesired document s in theprefetch cache is computed with the help of the purge list. Fig. 10 shows the Waste Ratio comparison.

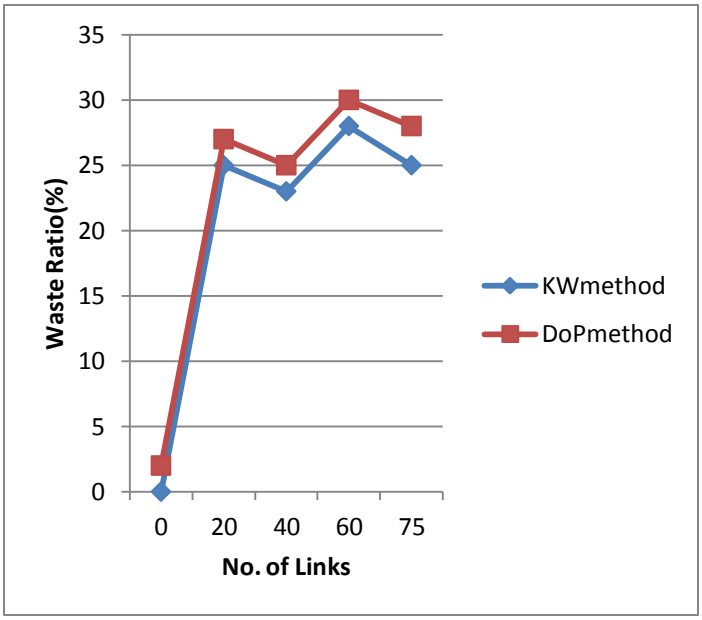


Fig. 10. Waste Ratio Analysis

The associated size of the undesired web objects that reside in the prefetch cache is the Byte Waste Ratio as shown in Fig.11.

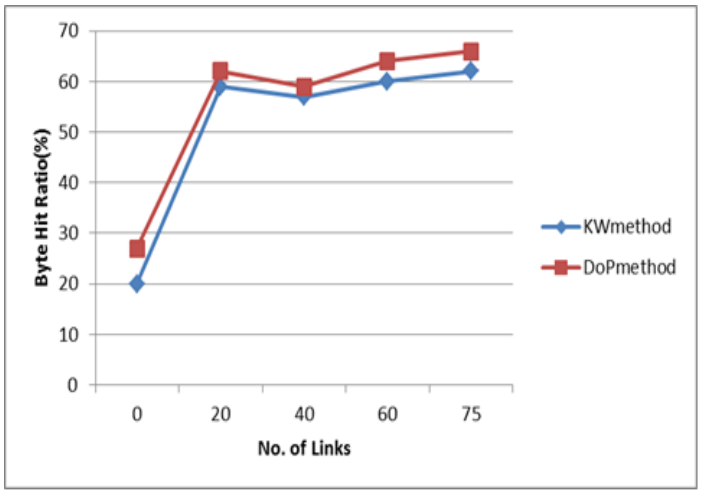


Fig. 11. Byte Waste Ratio Analysis

V. CONCLUSION

This study has presented architecture of Domain based Prefetching in Semantic Web and gives importance to the need for Content Prefetching and Domain wise Prefetching. The system facilitates the user request from relevant cluster. The performance aspect shows the DoP method which outperforms the existing method with varied domains and achieves the hit rate of 80%. The system considerably reduces the access latency. Since the log file of an educational institution was taken as the main platform, the set of users and their interests do not vary in different sectors. The user access patterns are almost decisive in nature. Currently this web ontology file is used for mapping of web log entries, SPARQL queries were used for retrieval and prediction.

This Research may further be extended by periodical analysis of the other domain. Instead of owl file representation, RDF structures may be used for representing the log file and the individual entries of the log file may be annotated. The current study focusses mainly the generic domains, further if an individual domain is separately analysed, there is a lot of scope to prove with more constructive results. There is innumerable number of areas available for further exploration in Prefetching.

REFERENCES

[1] Alexander, P.Pons,"Improving the performance of client web object retrieval," The journal of the Systems and Software 74, 2004 pp. 303-311, doi: 10.1016/j.jss.2004.02.030.

[2] Alexander, P. Pons, "Object Prefetching Using Semantic Links," The DATABASE for Advance in Information Systems," 2006 Vol.37, No. 1.

[3] Arumugam, G and S.Suguna, "Predictive Prefetching Framework Based on New Preprocessing Algorithms Towards Latency Reduction," Asian journal of Information Technology 7(3) , 2008 pp. 87-99, issn: 1682-3915.

[4] Bin Wu and Ajay D. Kshemkalyani, "Objective - Optimal Algorithms for Long-Term Web Prefetching," Proc. of IEEE Transaction on Computers, 2006 Vol. 55, No. 1.

[5] Cheng-Zhong Xu and Tamer I.Ibrahim, "Semantics-Based Personalized Prefetching to Improve Web Performance," Proc. of the 20th IEEE Conf. on Distributed Computing Systems, 2000 pp. 636-643.

[6] Cheng-Zhong Xu **and** Tamer I.Ibrahim, "A keyword-based semantic prefetching approach in Internet news services," Proc. of IEEE Transaction on Knowledge and Data Engineering, 2004 doi: 10.1109/TKDE.2004.1277820

[7] Daby M. Sow, David P. Olshefski, Mandis Beigi and Gurudth Banavar, "Prefetching Based on Web Usage Mining," International Federation for Information Processing 2003, LNCS 2672, pp.262-281.

[8] Gerd Stumme, Andreas Hotho and Bettina Berndt, "Semantic Web Mining State of the art and future directions," Elsevier Journal of Web Semantics 2006 doi: 10.1016/j.websem.2006.02.001.

[9] George Pallis, Athena vakali and Jaroslav pokorny, "A Clustering - based Prefetching scheme on a Web cache environment," Computers andElectricalEngineering2008pp.309-323, doi:10.1016/j.compeleceng.2007.04.002.

[10] Joseph Domenech, Ana Pont, Jose A. Gil and Julio Sahuquillo, "Guidelines for Evaluating and Adapting Web Prefetching Techniques,".XVII Jornadas De Paralelismo – Albacete 2006, Spain.

[11] Joseph Domenech, J.A. Pont, J. Sahuquillo and J.A Gil, "A User-focused evaluation of web prefetching algorithms,"Computer Communications2007pp.2213-224,doi:10.1016/j.comcom.2007.05.003.

[12] Juan D. Velasquez, Luis E. Dujovne and Gaston L'Huillier," Extracing significant Website Key Objects: A Semantic Web Mining Approach," Elsevier Journal of Engineering Applications of Artificial Intelligence 2011, doi: 10.1016/j.engappai.2011.02.001.

[13] Lenka Hapalova and Ivan jelinek, "Semantic web access prediction," Proc. Of International Conference on Computer Systems and Technologies 2007, ISBN: 978-954-9641-50-9.

[14] Marathe Dagadu Mitharam, "Preprocessing in Web Usage mining," International Journal of Scientific & Engineering Research, February 2012 Vol. 3, No.2.

[15] Nizar, R. Mabroukeh and C.I. Ezeife, " Semantic-rich Markov Models for Web Prefetching," Proc. of IEEE International Conference on Data Mining Workshops 2009, doi: 10.1109/ICDMW.2009.18.

[16] Qiang Yang and Henry Hanning Zhang, "Integrating Web Prefetching and Caching Using Prediction Models," World Wide Web 2001 pp. 299-321

[17] Samia Saidi and Yahya Slimani, "Enhancing Web Caching Using Web Usage Mining Techniques," Springer- Verlag Berlin Heidelberg 2010 pp. 425-435.

[18] Toufiq Hossain Kazi, Wenying Feng and Gongzhu Hu, "Web Object Prefetching: Approaches and a New Algorithm," Proc. Of IEEE International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2010, doi:10.1109/SNPD.2010.28.

[19] Victor Safronov and Manish Parashar, "Optimizing Web Servers Using PageRank Prefetching for Clustered Accesses," World Wide Web: Internet and Web Information Systems 2002 pp. 5, 25-40.

[20] WANG Xiao-Gang and LI Yue, "Web Mining Based on User Access Patterns for Web Personalization," Proc. of ISECS International Colloquium on Computing, Communication, Control and Management 2009.

[21] Yi-Hung Wu and Arbee L. P. Chen, "Prediction of Web Page Accesses by Proxy Server Log," World Wide Web 2002 Vol. 5 No. 1, pp. 67-88.