ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 9, September 2016

An Adaptive Approach to Reduce Web Access Time

Bhavna Thakre[1], Harish Patidar[2]

ABSTRACT

Increasing popularity of the World Wide Web over the past few years has imposed a significant traffic burden upon the internet. The World Wide Web may be considered to be a large distributed information systems providing access to shared data. A mass research has done to improve the response time of web based system as the information is distributed over a geographical location. Web caching and pre-fetching are two important approaches used to reduce the noticeable response time perceived by users. An ideal pre-fetching caching scheme is a system that able to predict the next (number of next) requests and pre-load those into the cache. The pre-fetched objects are stored in a local cache to reduce the latency time. In this paper a methodology will be proposed for better handling of web caching and pre-fetching using Apriori algorithm & FP Growth algorithm. This paper provide an dynamic & adaptive approach for reduce web access time that use FP growth algorithms for frequent page generation, relative weighted rule for determining relative weight of each page with respect to other in order to enhance response and Apriori algorithm is used to store relative weight of page to their relative position for fast and efficient web pre-fetching for expediting users visiting speed.

1. INTRODUCTION

Rapid growth of web application has increased the researcher's interests in this era. All over the world has surrounded by the computer network. There is a very useful application call web application used for the communication and data transfer. An application that is accessed via a web browser over a network is called the web application. Web caching is a well-known strategy for improving the performance of Web based system by keeping Web objects that are likely to be used in the near future in location closer to user. The Web caching mechanisms are implemented at three levels: client level, proxy level and original server level [1, 2]. Significantly, proxy servers play the key roles between users and web sites in lessening of the response time of user requests and saving of network bandwidth. Therefore, for achieving better response time, an be visited efficient caching approach should be built in a proxy server. Web caching and Pre-Fetching are the most popular techniques that play a key role in improving the Web performance by keeping web objects that are likely to in the Near future closer to the client. Web caching can work independently or integrated with the web Pre-Fetching. The Web caching and Pre-Fetching can complement each other since the web caching exploits the temporal locality for predicting revisiting requested objects, while the web Pre-Fetching utilizes the spatial locality for predicting next related web objects of the requested Web objects [1]. Pre-Fetching is used as an attempt to place data close to the processor before it is required, eliminating as many cache misses as possible. Caching offers the following benefits: Latency reduction, Less Bandwidth consumption, Lessens Web Server load. Pre-Fetching is the means to anticipate probable future requests and to fetch the most probable documents, before they are actually requested. It is the speculative retrieval of a resource into a cache in the anticipation that it can be served from the cache in the near future, thereby decreases the load time of the object [3].

Web caching is usually transparent to the user and to the application designer, except for the possible improvement in response time [4]. The application designer, when planning the development of a system, usually will not have enough information to judge if a web cache is involved. Also, if this developer is not knowledgeable in network protocols, he or she will focus on the application functionality, i.e., the interface between the scripting language and the database and the assembly of the pre-defined response pages [5].

2. PREVIOUS WORK

In this study [6], author reviews the principles and some existing web caching & prefetching approaches. Firstly, they have reviewed principles and existing works of web caching. This includes the conventional and intelligent web caching. Secondly, types and categories of prefetching have presented and discussed briefly. Moreover, the history-based prefetching approaches have been concentrated and discussed with review of the related works for each approach in this survey. Finally, authors presented some studies that discussed integration of web caching and web prefetching together. Web caching and prefetching are two effective solutions to lessen Web service bottleneck, reduce traffic over the Internet and improve scalability of the Web system. The Web caching and prefetching can complement each other since the web caching exploits the temporal locality for predicting revisiting requested objects, while the web prefetching utilizes the spatial locality for predicting next related web objects of the requested Web objects. Thus, combination of the web caching and the web prefetching doubles the performance compared to single caching.

In this paper [7], the improved version of Apriori algorithm is proposed to overcome the deficiency of the basic Apriori algorithm. The basic Apriori algorithm follows bottom up approach which suffers from increased number of data base scan. The new proposed method follows top down approach which reduces the number of database scans. The improved version Apriori algorithm is more efficient which takes less time, less memory and hence reflects in high efficiency.

In this study [8], authors provide a survey about the research in the area of Web mining's today structure and tomorrow view. They point some confusion between data mining and web mining. Web data is growing at a significant rate. Web Mining is fertile area of research. Many Successful applications exist. They also suggest the subtask of web mining & future of web mining. They also work for the process mining and try to combine usage mining with structure mining. Authors also go for the mining from cloud. Whenever they work on mining over cloud computing that time they hesitate for the cost but that come very less by cloud mining. So, they can say that cloud mining can seen as future of web mining.

In this paper [9], authors presented a semantic web prefetching for making the predictions of web objects to be prefetched. The scheme provides efficient predictions when users are visiting web pages that contain information related to a specific topic of interest. Since predictions are done based on the computed probability value of anchor texts, it is essential that user's use this scheme only when browsing the web pages for similar content. In case the user visits web pages in random without looking for specific content, then the scheme will provide weak predictions resulting in unnecessary prefetching of web objects. The prefetching scheme helps to minimize the user access latency when satisfying the user requests by achieving good hit rates across different session durations.

In this paper [10], the literature survey in the area of web mining is being provided. The paper basically focuses on the methodologies, techniques and tools of the web mining. The basic emphasis is given on the three categories of the web mining and different techniques incorporated in web mining. The paper explains the web mining subtasks and web mining taxonomy as a base. Then after, three literature review tables are being provided on web content, web structure and web usage mining. The survey came up with the pros and cons of the web mining techniques. It clears the scope of the web mining and presents a better analysis and comparison of web mining and its types.

In this paper [11], author have given brief introduction about the different Web prefetching techniques namely Prediction by partial match, Predictive web prefetching Model based predictive pre-fetching, Semantic web prefetching, Link pre-fetching, Domain top approach, Data prefetching, Content prefeching, Context based prefetching, Proxy cache prefetching, Dynamic web pre-fetching and Greedy-Dual-Size pre-fetching are analyzed and discussed. The web prefetching scheme focus on the property spatial locality of web objects. These techniques are applied to reduce the network traffic and improve the user satisfaction. Web prefetching and caching can also be integrated to get better performance.

In The paper [12], presents a framework for the prefetching and prediction in web. The framework has few merits over the previously proposed approaches, like as web latency reduction and web traffic reduction. According to the framework, previous web requests of the user will be extracted from the proxy web log. From this web log, strong rules will be generated using FP Growth algorithm. These rules will be used to prefetch the upcoming requests of the current user. Simulation performed using RapidMiner5 on data set named "pa.sanitizedaccess.20070110.gz" collected from ftp://ircache.net. The real world implementation of the framework will surly result into the improvement of cache hit ratio and web congestion control.

The main disadvantage of the proposed method is the complexity to implement and high computation needed for the proxy server [13]. Only web caching provides gives higher cache hit rather than caching & Prefetching. The whole architecture is only for static web objects. Recent web sites generally contain dynamic web documents. But, Dynamic web documents are tough to cache due to its low degree of reusability and strong dependent on backend data. There plan is to caching the Dynamic Web documents. The dynamic web documents may be cache by caching the data rather than the web documents.

In this paper [14], authors find results that FP-Growth algorithm is used for finding the most frequently access pattern generated from the web log data. By using the concept of web usage mining they can easily find out the user's interest and they can modify their web site more valuable and more easily accessible for the behavior users. The main focus of the empirical analysis and comparison is to identify factors. In their comprehensive study and defined different factors, they find results that the Apriori and FP-growth method is efficient and scalable for mining both long and short frequent patterns. In future the both algorithm can be extended to web content mining, web structure mining.

By the use of FP Growth algorithm in association rule mining they reduced the response time of most of the user queries thereby increasing proxy server hit ratio, it improves the overall performance of three tier web architecture. The user query is forwarded to proxy server through the listener and then a predictor maintains web log history and passes the hints to FP Growth block where association is applied on the loaded data set using association rule mining. This proposed framework [15], helps in reducing network congestion and it also helps in providing security to the main server as by increasing the hit ratio in proxy server somehow they kept the user away from main server. By using this framework they can also reduce DOS (Denial of service) attacks up to some extent. This framework will also reduces the time for which a network channel is allocated to a user in TCP, as soon as a user gets results to its queries, the allocated channel will become free and network traffic can be reduced. The merits of this framework includes: increase in proxy server cache hit ratio. The simulation of this framework is performed using Data-Applied tool on data set named Web Click stream collected from data-applied.com. Actual realization of this framework results in improved network architecture.

In this paper [16], author used SPRINT- a decision tree induction classification technique, instead of naive bayes. Initially anchor text is found and using lexical analyser, tokens are counted. Then after, SPRINT is applied. On the results of SPRINT, patterns are compared with the threshold. If they are greater than the threshold, they are prefetched and stored in the cache, else discarded. Results of the proposed approach is compared with the previous ones using Mat lab 5.0 and found that proposed work provides more accurate results.

This survey paper [17], author study concept of frequent item sets and web usage mining generally serve as building blocks for various patterns in many real-life applications. User perceived latency's from several sources such as bandwidth, speed, overhead, accessing the web page etc. Most of the existing algorithms find unconstrained frequent item sets from traditional static transaction databases consisting of precise data. However, there are situations in which ones are uncertain about the contents of transactions. There are also situations in which users are only interested in some subsets of all the mined frequent item sets. Furthermore, a flood of data can be easily produced in many situations. If we use the prefetching with caching then the performance of cache is improved. Prefetching fetches objects that are likely to be accessed in the near future and store them in advance thus the response time of the user request is reduce.

3. PROPOSED ARCHITECTURE

Web log file is used to capture client request at server side, whether web log file is an rough set having higher degree of noise so it is need to pre process log file before taking any decision over web pre fetching then we applying frequent mining pattern with FP Tree rules. Which is obtaining on the basis of input and on the basis of these rules apriori algorithm will apply on it. Thus this paper presents a new idea for the interpretation of Web pre-fetching from the given usage items.

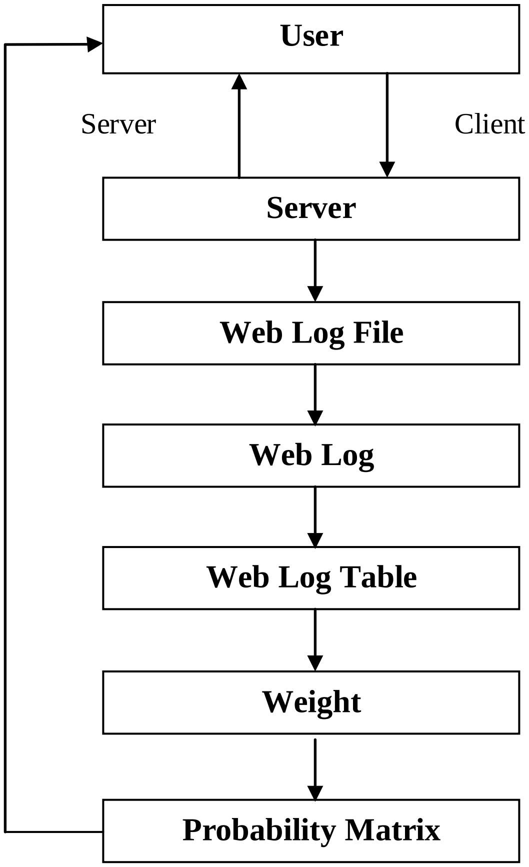

Figure 1: Flow of Data

The approach works on the basis is web mining with the combination of apriori algorithm. The flow of data is demonstrated in figure 1.

Proposed Scheme for web page prediction can be easily understand by the architectural diagram as show in figure 2. Where proposed diagram having two different layers, front and back. Front layer is use to grape web information i.e. web transaction information in web log. Whereas backed layer used to analyses this information and generate resultant apriori model for future web page prediction.

3.1 FRONTEND LAYER

Front layer responsible for capturing client web access behavior over web log file whereas back layer use this historical information as a input to analyses client web behavior for web page pre fetching. Once analysis over log file is completed, back layer generate resultant apriori model.

After completion of apriori model if any client A requests a web page P1 web server performing two different job over that request before replying. first redirect that request to transaction probability matrix of backend layer, transaction probability matrix reply number of most frequent pre fetch page index number having higher relative weight where number of pre fetch page depend upon catch size i.e. as per requirement.

3.2 BACKEND LAYER

Backend layer is use to pre- process and refine raw log file and generate apriori model. Back layer having following step. Data cleaning, user and session identification, data integration and so on are main important part of log pre processing.

3.2.1 WEB LOG PREPROCESSING

Log file used to capture client server behavior over the network at any time i.e. what page has been requested by which client and when all this information has been capture in web server log file. Along with that important information there is also some inconsistent data like noise, null value and other error information which is not so important for web personalization so in order improve web mining result its need to refine web log file before mining

3.2.2 WEB LOG TABLE

After web log preprocessing the recognized useful tokens are stored in the database. Normalization is also performed in order to remove the redundancy.

3.2.3 WEIGHT ASSIGNMENT

Weight assignment concept is being used mapping any web page with their entire relevant page having. higher relative weight

$$R_w = \frac{Number\ of\ occurance\ of\ page\ x\ and\ page\ y\ together}{Number\ of\ occurance\ of\ page\ x} ..1$$

Relative weight of any page y with respect to x means probability of page y request after page x is being calculated by dividing number of occurrence of page x with page y together with number of occurrence of page y.

3.2.4 PROBABILITY MATRIX

This step is concern with relative positioning of relative weight evaluate in previous step, whereas relative position is drive from apriori algorithm. Actually data representation in apriori algorithm is very efficient in both representation and retrieval.

The frequent patterns are extracted with the weight values. The weighted support is estimated and used for the pages. As suggested in Algorithm below.

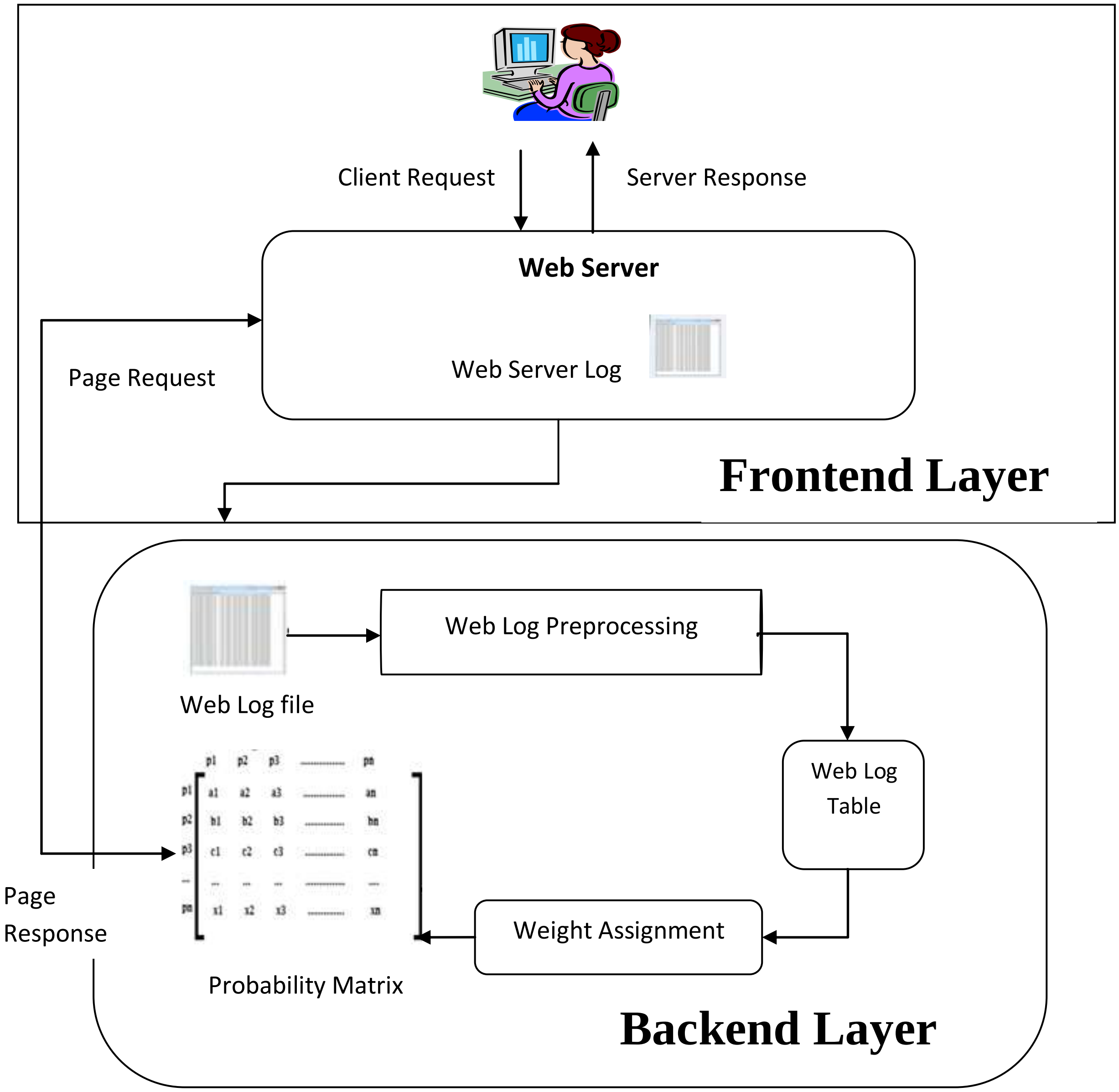

Figure 2: Architecture of proposed algorithm

ALGORITHM

Step1- Log Pre Processing – Token reorganization and removal of unwanted attribute and store in data base.

Step 2- preparing web log table - After web log preprocessing the recognized useful tokens are store in the database. Normalization is also performed in order to remove the redundancy

Step 3- Weight Assignment - For each page Pi this step evaluate weight for each of its relative page Pj

$$W_{Pi,pj} = \frac{total\ \ number\ \ of\ \ occurance\ \ of\ \ Pi\ \&\ Pj\ \ toget\ her}{total\ \ number\ \ of\ \ occurance\ \ of\ \ Pi}$$

Step 4- Apriori Model- Assign relative weight to two dimensional table as apriori model suggest

$$P_{I,J}^{M} = \begin{bmatrix} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \end{bmatrix} = W_{pi,pj}$$

4. SIMULATION AND RESULT

Time Complexity: Proposed methodology for taking decision concerning Pre-Fetch page having additional overhead time needed to evaluating the request from any client. FP Growth tree concept uses Tree to store relative weight and time taken for taking decision about Pre-Fetch page is O (Log N) where N is height of tree, whereas proposed technique uses two dimension table that take O (1) for Pre-Fetching single page, The Table 1 gives the detailed time analysis between proposed technique, FP Growth Tree and Apriori Algorithm.

Space Complexity: In terms of space complexity proposed approach is moderate because it needs large space as compared to FP Growth Tree methodology but much lesser than plain Apriori algorithm . Table 2 shows the space complexity proposed technique, FP Growth Tree and Apriori algorithm. Here the table shows that the space complexity of proposed method is much lesser than apriori algorithm but a little bit more than FP Growth Tree model.

Table 1: Time Comparison

No. of Pages	**Proposed Technique** (Time in ms)	**FP Growth Tree** (Time in ms)	**Apiori Algorithm** (Time in ms)
1	1	0	1
10	1	1	1
50	1	1.698970004	1
100	1	2	1
150	1	2.176091259	1
200	1	2.301029996	1
250	1	2.397940009	1
300	1	2.477121255	1
350	1	2.544068044	1

Table 2: Space Comparison

No. of pages	**Proposed Technique** (space in Kb)	**FP Growth Tree** (space in Kb)	**Apriori Algorithm** (space in Kb)
2	4	4	6
4	16	8	60
6	36	12	210
8	64	16	504
10	100	20	990

5. CONCLUSION

There are large number of web application has been used for the various purpose. These applications should be good at its response time. Therefore the web caching and web Pre-Fetching are the approaches which can be applicable for the enhancing the response time of a web application.

This work is an assessment on these approaches. These concepts come under the web mining so this paper are also covered the web mining. This paper also throws some light on the previous work in the related field. In this paper related work section describe that the web Pre-Fetching can apply on any web application. This approach can apply with various strategies.

This work using the Apriori algorithm and FP-Growth tree are give better results as compare to previous work in terms of Pre-Fetch ratio, accuracy of Pre-Fetching, space complexity and time complexity. The implementation is also showing that it is easy to apply in order to Pre-Fetch the pages of a web site.

This paper is to demonstrate that web Pre-Fetching is an effective solution to reduce web latency perceived by the users and that it can be implemented easily and efficiently in the current real environment. On the basis of result analysis work using the FP growth tree and Apriori algorithm shows that, it gives better results as compare to previous work i.e.

- Improved time Complexity as compared to FP Growth tree.
- Improved space complexity as compared to the apriori algorithm.
- Reduces Web Latency.
- The implementation also shows that it is easy to apply in order to pre-fetch the page of a web site.

REFERENCES

[1] K. Ramu, Dr. R. Sugumar and B. Shanmugasundaram “A Study on Web Prefetching Techniques” Journal of Advances in Computational Research: An International Journal Vol. 1 No. 1-2 (January-December, 2012)

[2] Waleed Ali, Siti Mariyam Shamsuddin, and Abdul Samad Ismail “A Survey of Web Caching and Prefetching”, Int. J. Advance. Soft Comput. Appl., Vol. 3, No. 1, March 2011

[3] Daesung Lee and Kuinam J. Kim, “A Study on Improving Web Cache Server Performance Using Delayed Caching”, IEEE 2010, pp 1-5.

[4] Bhawna Nigamand Dr. Suresh Jain, “Analysis Of Markov Model On Different Web Prefetching And Caching Schemes”, 978-1-4244-5967-4/10/ ©2010 IEEE

[5] M. Junchang, G. Zhimin, “Finding Shared Fragments in Large Collection of Web Pages for Fragment-based Web Caching”, Fifth IEEE International Symposium on Network Computing and Applications (NCA'06) 2006.

[6] Waleed Ali, Siti Mariyam Shamsuddin, and Abdul Samad Ismail “A Survey of Web Caching and Prefetching” ISSN 2074-8523; Copyright © ICSRS Publication, 2011, Int. J. Advance. Soft Comput. Appl., Vol. 3, No. 1, March 2011.

[7] Suneetha K R and Krishnamoorti R, “Web Log Mining using Improved Version of Apriori Algorithm”, in International Journal of Computer Applications (0975 – 8887) Volume 29– No.6, September 2011.

[8] Kavita Sharma, Gulshan Shrivastava, and Vikas Kumar “Web Mining: Today and Tomorrow” 2011 IEEE 978-1-4244-8679-3/11.

[9] Dr. K. Ramu, “Improving the Performance of Web Using Semantic Prefetching Methodology”, International Journal of Advanced Research in Computer and Communication Engineering, vol. 1, issue 4, June 2012.

[10] K.Ramu, Dr.R.Sugumar and B.Shanmugasundaram “A STUDY ON WEB PREFETCHING TECHNIQUES” Journal of Advances in Computational Research: An International Journal, Vol. 1 No. 1-2 (January-December, 2012).

[11] N. Sharma and S.K. Dubey, “FP tree use in prefetching”, Proc. of Int. Conf. on Advances in Computer Science, AETACS, Elsevier, 2nd December, 2013, pp 555-561.

[12] Anish Kumar Saha, Hrituparna Paul and Partha Pratim Deb, “Integration of Caching and Prefetching in Proxy Server” published on vol 3 Issue 3, 2014 in International Journal of Engineering Research & Technology.

[13] Avadh Kishor Singh, Ajeet Kumar and Ashish K. Maurya “An Empirical Analysis and Comparison of Apriori and FP- Growth Algorithm for Frequent Pattern Mining” 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (lCACCCT), pp 1599-1602.

[14] Devender Banga and Sunitha Cheepurisetti, “PROXY DRIVEN FP GROWTH BASED PREFETCHING”, in International Journal of Advances in Engineering & Technology, July, 2014.

[15] Neha Sharma and Dr. Sanjay Kumar Dubey “Semantic Based Web Prefetching Using Decision Tree Induction” IEEE 2014 5th International Conference- Confluence The Next Generation Information Technology Summit (Confluence), pp 132-137.

[16] Devendra Parmar, "SURVEY ON WEB USAGE MINING AND PRE-FETCHING", in International Journal of Advance Engineering and Research Development, Volume 2,Issue 3, March -2015.

[17] Mr. Swapnil S. Chaudhari and Prof. Poonam Gupta, "PROXY SIDE WEB PREFETCHING SCHEME FOR EFFICIENT BANDWIDTH USAGE: DATA MINING APPROACH", in International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 09 | Dec-2015.