ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

Hybrid Cache Replacement Policyfor Proxy Server

SirshenduSekhar Ghosh[1], Dr. Aruna Jain[2]

Research Scholar, Dept. of Information Technology, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India[1]

Associate Professor, Dept. of Information Technology, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India[2]

Abstract: As the World Wide Web (WWW) usage has grown exponentially in last two decades, so has grown the recognition that Webproxy caching and Web prefetching techniques play an important role in reducing server loads, client response latencies, network traffic and bottlenecks. The sophisticated combinations of these two techniques cause significant performance improvements of the Web infrastructure and Quality of Service. The heart of a caching system is its page replacement policy, which needs to make efficient replacement decision when its cache is full and a new document needs to be stored. Several replacement policies based on recency, frequency, size, cost of fetching object and access latency have been proposed for improving the performance of Web caching by many researchers. However, it is difficult to have anomnipotentpolicy that performs best in all environments as each policy has different design rational to optimize different resources. In this paper we have implemented Hybrid Cache Replacement policy for proxy server, based on frequency, recency and the prefetch sequence of each object and also proposed the design for an integrated Cache-Prefetching System model which employs the above policy. The simulation results show that our proposed replacement policy performs better than other policies proposed in the literature in terms of Hit ratio, Byte hit ratio and Latency saving ratio.

Keywords: Web caching, Replacement policy, Web prefetching, Proxy Server, Hybrid Cache replacement policy.

I. INTRODUCTION

Due to explosive growth of the Webin terms of number of users [1] and number of Web applications [2], increase in Web latency, network congestion and server overloading are the crucial problems for enhancing the Web performance. Web caching is a widely deployed technique in the Web architecture which stores most popular Web objects already requested by users into a pool close to the client-side [3]. Web caching mechanisms are implemented at three levels: client level, proxy level and original server level. Amongst these, proxy servers play key role between users and Web servers in reducing the response time and thus saving network bandwidth. Therefore, for achieving better response time, an efficient caching approach should be built in a proxy server. It takes advantage of the Web object's temporal locality to reduce user-perceived latency, bandwidth consumption; network congestion and traffic.Kroeger et al. [4] divide the latency into internal and external taking into account the use of a proxy server on the Web architecture. Further, as it delivers cached objects form Proxy Servers; it reduces external latency and improves reliability as users can obtain a cached copy even if the remote Server is unavailable.

The efficiency of proxy caches is influenced to a significant extent by document Placement/Replacement algorithms. If a cache is full and an object needs to be stored, the policy will determine which object to be evicted to make room for the new object. In practical implementation a replacement policy usually takes place before the cache is really full. The goal of the replacement policy is to make the best use of available resources including disk space, processing speed, server load, and network bandwidth. In today's computing world where cache size is not a big issue but some significant problems such as updating the large cache which involves complexity and thus results in an increase in response time. Therefore, we must resort to an approach, which will predict the future users' requests and retain in cache the most valuable objects thus improving the Web latency.

Web prefetching is another extremely useful technique focused on latency reduction based on predicting the next future Web objects to be accessed by users and prefetching those in idle times. So, if finally the users request it, the object will be already available at the client's cache. This technique takes advantage of the spatial locality of Web objects and prevents bandwidth underutilization. Therefore, bottlenecks and traffic jams on the Web are bypassed and objects are transferred faster. Thus, proxies employing prefetching technique can effectively serve more users' requests, reducing the workload from the origin Servers. Consequently, protecting the origin Servers from the "flash crowd" events as a significant part of the Web traffic is diverted to the proxy Servers. As suggested by Marquez et al. [5], the prefetching technique has two main components: The prediction engine and the prefetchengine. The prediction engine runs a prediction algorithm to predict the next user's request and provide these predictions as hints to the prefetch engine. The prefetch engine handles the hints and

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

decides to prefetch them or not depending on some conditions like available bandwidth or idle time. Each engine can work at any element of the Web architecture.

In our study we only consider the cache replacement policy for cache servers located between clients and origin servers, acting as a proxy. There has been significant amount of research work carried out in the past for enhancing the performance of Web infrastructure by integrating Web caching and prefetching. Kroeger et al. [4] suggest that the use of Caching can reduce up to 26% of Latency; and the use of Prefetching can improve the Web performance up to 57%. Motivating by the wealth of research in replacement policies of Web caching [3,6,7,8] as well as the benefits of Web prefetching[5,14,15,17,19,21], in this paper we have proposed a prefetch enhanced hybrid cache replacement policy based on both frequency and recency along with prefetch sequence of each Web object. Frequency-based polices [6] use object popularity (or frequency count) as the primary factor. The rationale behind is that different Web objects have different popularity values, and only a small set of popular objects account for most of the total requests. Therefore, by trying to keep those objects with high frequency counts in the cache, most requests can be satisfied. This category of polices is suitable for systems in which the popularity distribution of objects is highly skewed, or in which there are many requests to Web sites having objects with very steady popularity (rarely changing abruptly). LFU is a simple policy that evicts the least frequently referenced object first. On the other hand, Recency-based policies use recency as the primary decision making factor. Most of the policies in this category are LRU variants which evicts the least recently referenced object first. It is designed on the assumption that a recently referenced document will be referenced again in the near future. These policies perform particularly well when Web request streams exhibit high temporal locality i.e. many clients have a common set of Web objects in which they are interested. This is particularly popular because of its simplicity and fairly good performance in many situations.

We combine the most popular cache replacement policies, LFU and LRU which have been effectively adopted by proxy servers by integrating a prefetching mechanism. In our proposed policy the cache space is logically divided into two portions: Normal cache queue (with LFU) and Prefetch cache queue (with LRU). The prefetch sequence of Web objects will be stored separately in the Prefetch cache queue. Considering prefetch sequence can have some benefits and may become a powerful tool for specifying replacement conditions especially for applications like Web caching.

The rest of the paper is organized as follows: Section 2 reviews the related work and outlines the motivation and contribution of this work. Section 3 describes the prefetch enhanced hybrid cache replacement policy. Section 4 explains the proposed Integrated Cache-Prefetching system architecture. Section 5 provides the results and discussion and finally concluding remarks in Section 6.

II. RELATED WORK

Web caching has been studied with many different angles by the research community over the years.Balamash et al. [7] and Poplipniget al. [8] give an overview of various replacement algorithms. They conclude that GDSF outperform when cache size is small.

Williams et al. [9] discussed that SIZE outperforms than LFU, LRU and several LRU variations in terms of different performance measures; cache hit ratio and byte hit ratio. In their experiments, they fail to consider object frequency in decision making process.

Rachid et al. [10] proposed a strategy called class-based LRU. C-LRU works as recency-based as well as size-based, aiming to obtain a well-balanced mixture between large and small documents in the cache, and hence, good performance for both small and large objects requests. The caching strategy class-based LRU is a modification of standard LRU.

Triantafillou et al. [11] employ CSP (Cost Size Popularity) cache replacement algorithm which utilizes the communication cost to fetch Web objects, object's sizes, their popularities, an auxiliary cache and a cache admission control algorithm. They conclude that LRU is preferable to CSP for important parameter values, accounting for the objects' sizes does not improve latency and/or bandwidth requirements, and the collaboration of nearby proxies is not verybeneficial.

Rassul et al. [12] present two modified LRU algorithms and compare their performance with the LRU. Their results indicate that the performance of the LRU algorithm can be improved substantially with very simple modifications.

Griffioen et al. [13] paid attention to the modeling of Web prefetching and caching on file system. The research assumed that prefetching and caching share the same cache space, and showed that integrated Web prefetching and caching can improve the performance of cache system.

Cao et al. [14] presented a model of integrated Web prefetching and caching on file system, and based on which, she made performance study and simulative validation. Simulations illustrated that the integrated model could reduce the elapsed times of the applications by up to 50%.

Yang et al. [15] [16] presented an integrated architecture for Web object caching and prefetching. The Web object prediction model was built by mining the frequent paths from past Web log data, and prefetching algorithm named Pre-GDSF was implemented. Experimental shows that integrated Web prefetching and caching system can have a better performance than those without prefetching mechanism.

Teng et al. [17] developed IWCP (Integration of Web Caching and prefetching) algorithm by integrating Web caching and Web prefetching which outperforms LNC-R-

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

W3-Pc algorithm in terms of delay saving ratio and hit ratio. But their algorithm was developed only for client-side proxies.

Bouras et al.[18] present an extended study about a prefetching technique and its impact on the Proxy Cache Server in a real WAN environment (i.e. university campus). The later proposal contributes with many useful considerations (e.g. log analysis, session estimation, Web object types) to take into account when prefetching is applied.

Kroeger et al. [4] suggests that the use of caching can reduce up to 26% the latency; also the use of prefetching can improve the Web performance up to 57%. Furthermore, the combined use of caching and prefetching can reduce the latency perceived up to 60%. Nevertheless, Dom`enech et al. [19], taking into account the current Web generation, point out a theoretical upper bound of 97% of latency reduction when prediction is done in a collaborative manner between proxies and servers.

Dom`enech et al. [19] studied the impact of the Web architecture on the limits of latency reduction. They concluded that latency reduction depends on the predictor location: it can be reduced by 36%, 54%, and 67% when the predictor is located at the server side, client, or proxy, respectively. Latency reductions higher than 90% could be obtained if the predictor works collaboratively at different elements of the architecture.

Bhawnaet al. [20] analyzed Dynamic Nested Markov Model on three Prefetching and Caching schemes: Prefetching only, Prefetching with Caching and Prefetching from Caching for modeling the Web Log and to predict the next access Web page. The predicted Web page will be prefetched and cached in some Cache to save user's navigation time to provide better navigational services to the Web users.

Shi et al. [21] present a model to control the prefetch requests in the Proxy Cache server side. Their mechanism tries to prevent the cache pollution caused by the prefetched objects. Therefore, if the prefetched objects replace the most popular objects cached and the cache hit ratio is decremented, the mechanism reduces the prefetch request to avoid this effect.

Jyoti et al. [22] presented an approach that will predict the user page access before user accessing them. They have used higher order markov model for predicting user next request. The main drawback is that, it will predict only one object at a time.

González et al. [23] evaluated the performance of various cost based algorithm by using different content types like audio, video and image and so on.

Nair et al. [24] presents a dynamic pre-fetching technique implemented at proxy server in which Web caching and prefetching techniques are integrated. Using the technique cache hit ratio is increased to 40%-75% and subsequently latency is reduced to 20%-63%.

III. PREFETCHENHANCED HYBRID CACHE REPLACEMENT POLICY

In our proposed Prefetch enhanced Hybrid cache replacement policy, cache space is divided into two partitions: Normal cache queue (with LFU as the replacement policy) and a Prefetch cache queue (using the LRU policy). This partitioning of the cache space aims at isolating the effects of document mispredictions and aggressive prefetching. It achieves this by dedicating part of the cache space to exploit the temporal locality of the request stream (on-demand requests) and the rest of the cache space is dedicated to exploit the spatial locality (prefetch requests).

The relative size of the partitions should reflect the "amount" and type of the locality of the request stream. The hinted Web objects are downloaded from the server and stored separately in the Prefetch cache queue until it is full based on the assumption that the requests occurred in the recent past will likely to occur in the near future too. It is maintained to eliminate the caching effect in Normal cache queue due to temporal locality in the user's browsing patterns. LRU algorithm is used to manage the Web objects stored in the Prefetch cache queue by selecting least recently accessed Web objects for purging to provide space for accommodating newly prefetched Web objects.

When a client request arrives, the cache manager checks whether the page is available in Normal cache queue or in the Prefetch cache queue. If a hit occurs in the Prefetch cache queue, the pre-fetched document will be moved to the Normal caching partition, replacing the least frequently used document. Web objects will not be stored in both the caches (normal and prefetch) at the same time.

IV. PROPOSED INTEGRATED CACHE-PREFETCHING SYSTEM ARCHITECTURE

The proposed Integrated Cache-Prefetching model is implemented at proxy server as it can serve wide range of clients. In todays' demanding situation, deployment of proxy server between Web clients and Web server is essential to minimize server overload, bottlenecks and user access latency. Fig.1 illustrates the proposed

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

integrated system architecture.

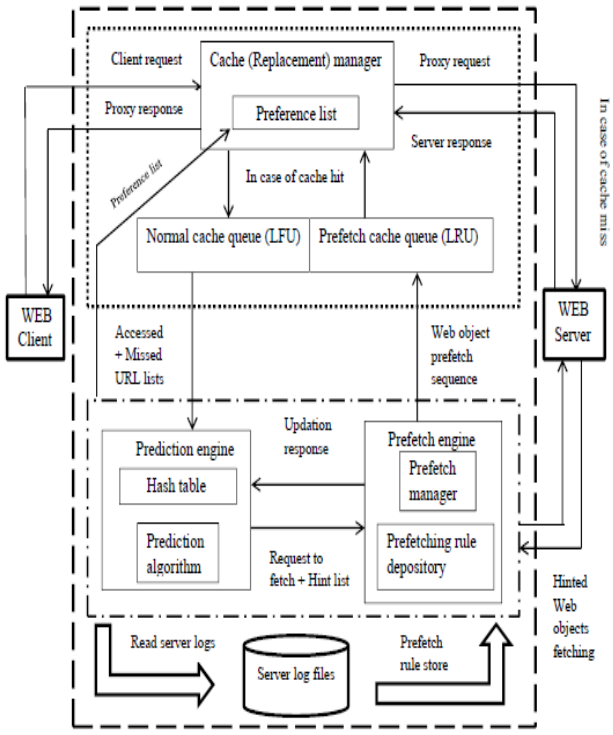

Fig. 1 Proposed proxy server based prefetch enhanced Hybrid cache replacement model

The steps shown in Fig.1 can be explained as follows:

1. Web clients issue object requests which are sent to the Cache manager and it authenticate the users by user names and passwords by checking the stored Preference list.
2. The cache is logically partitioned into two portions, Normal cache queue (with LFU) and Prefetch cache queue (with LRU). The Cache (Replacement) manager calls the Hybrid replacement algorithm that collects information of the requested object in cache and makes the replacement decision.
3. Cache manager checks if the object is found at cache (either in Normal cache or Prefetch cache). If found considered as Hit and send it to the client in response with minimal latency, otherwise miss and sends the request to the Web server.
4. The Cache manager also prepares lists of accessed and missed URLs and sends to the Prediction engine.
5. Prediction engine stores in a Hash table both the lists and their weight information. It also runs a Prediction algorithm to predict the next user's request and provide these predictions as hints to the Prefetch engine.
6. The Prefetch manager in the Prefetch engine handles the Hints, generates and stores Prefetching rules at Prefetching rule depository by discovering users' Web page access patterns by reading the proxy server's access log periodically. Thus the users' Preference list of objects can be generated and supplied to the Cache manager.
7. Prefetch manager also decides whether to prefetch from Web server or not depending on certain conditions like the available bandwidth or the idle time and automatically starts downloading the hinted Web objects from the Web server and send as Prefetch sequence to the Prefetch cache. It also sends the updation information as response to the Prediction engine.

V. RESULTS AND DISCUSSION

The data set for testing our proposed Hybrid Replacement Cache Policy is obtained from proxy server of Birla Institute of Technology (BIT), Mesra, Ranchi, Jharkhand which is extremely popular among students, faculty members and staffs of as many twenty five departments along with various administrative sections, hostels and quarters. We constructed a trace-driven simulation to study our proposed model using a set of student/faculty traces from the university. A trace collected by proxy server, referred to as proxy logs and contains information about Web documents accessed by users. In our experimentthe proxy traces refer to the period from 12/Sept/2011:11:45:04 to 26/Sept/2011:00:00:02 of two weeks.The trace is composed of 11,388 nodes and 1,165,845 Web requests with averageof 2,300 users per day. The simulations were performed at different network loads.

Hit Ratio (HR), Byte Hit Ratio (BHR) and Latency Saving Ratio (LSR) are the most widely used metrics in evaluating the performance of Web caching. HR is defined as the percentage of requests that can be satisfied by the cache. BHR is the number of bytes satisfied from the cache as a fraction of the total bytes requested by user. LSR is defined as the ratio of the sum of download time of objects satisfied by the cache over the sum of all downloading time.

Let N be the total number of requests (objects).

δ_i= 1, if the request i is in the cache, while

δ_i= 0, otherwise.

Mathematically, this can be expressed as follows:

$$HR = \frac{\sum_{i=1}^{N} \delta i}{N} \qquad \text{(i)}$$

$$BHR = \frac{\sum_{i=1}^{N} bi\,\delta i}{\sum_{i=1}^{N} bi} \quad \text{(ii), where } b_i \text{ is the size in bytes of the } i^{th} \text{ requested object,}$$

$$LSR = \frac{\sum_{i=1}^{N} ti\,\delta i}{\sum_{i=1}^{N} ti} \qquad \text{(iii), where } t_i \text{is the time to download the } i^{th} \text{referenced object from server to the cache.}$$

A high HR indicates the user's satisfaction and defines an increased user servicing. On the other hand, a high BHR and LSR improve the network performance and reduce

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

the user-perceived latency (i.e. bandwidth savings, low congestion etc.).

We did the trace-driven simulation to compare the performance of our proposed prefetch enhanced hybrid cache replacement policy with some well-known cache replacement algorithms. We choose three basic replacement algorithms, FIFO, LRU and LFU, which respectively consider three most basic factors, reference number, last access time and access frequency of objects.

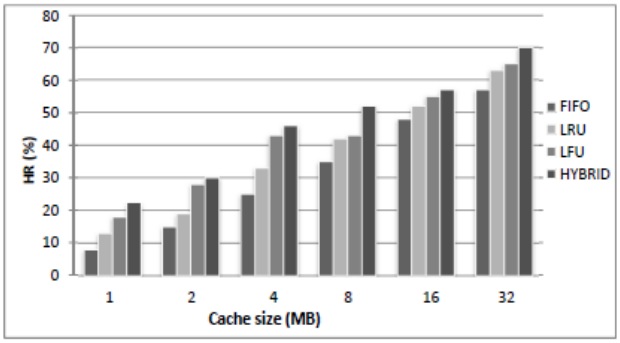

Fig. 2Hit ratio (HR) for the proxy trace

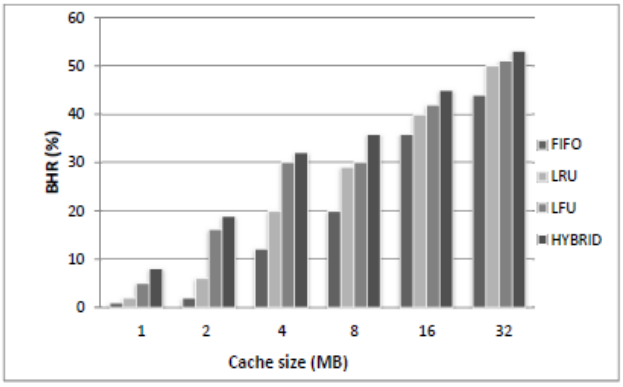

Fig. 3Byte hit ratio (BHR) for the proxy trace

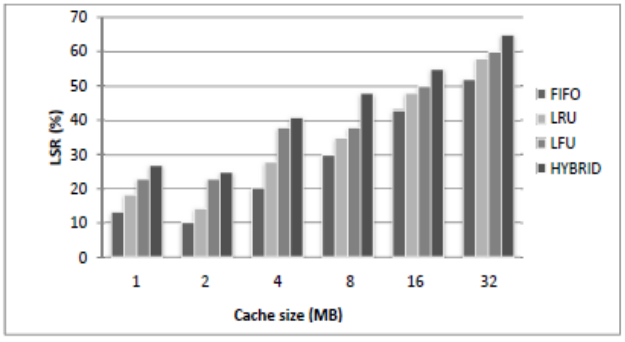

Fig. 4Latency saving ratio (LSR) for the proxy trace

We have done the trace-driven experiment with an increase in size of the cache and compared performances of FIFO, LRU, and LFUwith our proposed Hybrid policy in terms of hit ratio, byte hit ratio and latency saving ratio. As shown in Fig.2 through 4, with an increase in size of the cache our proposed hybrid policy outperforms the other policiesfor all performance metrics. Results indicate that our proposed Hybrid policy can improve the performance in terms of HR up to 21%, in terms of BHR up to 16% and in terms of LSR up to 18% compared to FIFO, LRU and LFU.

VI. CONCLUSION

In this paper, we addressed an integrated cache-prefetching system model to be deployed at proxy server using a hybrid cache replacement policy. The proposed scheme efficiently integrates Web caching and prefetching whereas the cache space is partitioned into Normal cache queue (with LFU as the replacement policy) and a Prefetch cache queue (using the LRU policy). LRU and LFU are two very simple and widely used Web cache replacement policies. In the recent years, several more efficient replacement policies such as LRU-Min, LRU-Threshold, SIZE, Lowest Latency First, Hyper-G, Greedy-Dual-Size (GDS), Lowest Relative Value (LRV), LNC-R-W3, Size-adjusted LRU (SLRU), Least Unified-Value (LUV), Hierarchical Greedy Dual (HGD), Smart Web Caching etc. have been proposed. But, these advanced policies require more knowledge about the workloads and are generally more difficult to implement. We used trace driven simulation to evaluate the performance of the proposed policyand compare it with three other traditional policies. The main attraction of LRU and LFU is their simplicity and by integrating these with Web prefetching, our proposed model improves the proxy server performance. Current Web prefetching technique is still far from achieving its maximum performance, mainly due to the accuracy of prediction algorithm. High accuracy prediction model can enhance the Prefetching system as well as our proposed integrated Cache-Prefetching system performance to a better extent. Further work is going in this direction. Also by surveying our previous work [25] on Web caching and prefetching, we notice that there are still some open problems in Web caching such asproxy placement, cache routing, dynamic data caching, fault tolerant, security, etc. The research frontier inWeb performance improvement lies in developing efficient, scalable, robust, adaptive, stable Web cachingscheme that can be easily deployed in current and future network.

REFERENCES

[1] http://www.internetworldstats.com/stats.htm
[2] http://www.worldwideWebsize.com/
[3] SarinaSulaiman, S.M.Shamsuddin, A. Abraham, S. Sulaiman, "Web Caching and Prefetching: What, Why, and How?", IEEE, 2008.
[4] T. M. Kroeger, D. D. Long, and J. C. Mogul, "Exploring the bounds of Web latency reduction from caching and prefetching," in Procc. of the 1st USENIX Symp. on Internet Technologies and Systems, Monterey, USA, 1997.
[5] J.Marquez, J. Dom`enech, J. A. Gil, and A. Pont, "An intelligent

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

technique for controlling Web prefetching costs at the server side", IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008.

[6] A.K.Y. Wong, "Web Cache Replacement Policies: A Pragmatic Approach", IEEE Network magazine, 20(1), (2006), pp.28–34.

[7] A.Balamash, M.Krunz, "An Overview of Web Caching Replacement Algorithms", IEEE Communications Surveys & Tutorials, Second Quarter, 2004.

[8] Poplipnig, S. and Böszörmenyi, L.,"A Survey of Web Cache Replacement Strategies", ACM Computing Surveys, Vol. 35, Ner 4, pp. 374-398, 2003.

[9] S.Williams, M.Abrams, C.R. Standbridge, G.Abdulla and E.A.Fox, "Removal Policies in Network Caches for World-Wide Web Documents", Proceedings of the ACM Sigcomm96, August, 1996, Stanford University.

[10] Boudewijn R. Haverkort, Rachid El AbdouniKhayari, RaminSadre, "A Class-Based Least Recently Used Caching Algorithm for World-Wide Web Proxies", Computer Performance Evaluation / TOOLS 2003: 273- 290.

[11] P. Triantafillou and I. Aekaterinidis, "Web Proxy Cache Replacement: Do's, Don'ts, and Expectations", Proc. of the second IEEE Int. Symposium on Network Computing and Applications (NCA'03), 2003.

[12] Rassul A, Y.M. Teo and Y.S. Ng, "Cache Pollution in Web Proxy Servers", IEEE, 2003.

[13] J. Griffioen, R. Appleton, "Reducing file system latency using a predictive approach", In Proc. of USENIX Summer Conference, 1994, pp. 197-207.

[14] P. Cao, E.W. Felten, A.R. Karlin, K. Li, "A Study of Integrated Prefetching and Caching Strategies", In Proc. Of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, pp. 188-197, 1995.

[15] Q. Yang, H. Zhang, "Integrating Web Prefetching and Caching Using Prediction Models", World Wide Web, pp. 299-321, 2001.

[16] Q. Yang, J. Z. Huang, N. Michael, "A Data Cube Model for Prediction-Based Web Prefetching", Journal of Intelligent Information Systems, 2003, vol.20 (1), pp.11-30.

[17] W.-G. Teng, C.-Y. Chang, and M.-S. Chen, "Integrating Web caching and Web prefetching in client-side proxies," IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 5, pp. 444–455, 2005.

[18] C. Bouras, A. Konidaris, and D. Kostoulas, "Predictive prefetching on the Web and its potential impact in the wide area." World Wide Web, vol. 7, no. 2, pp. 143–179, 2004.

[19] J. Dom`enech, J. Sahuquillo, J. A. Gil, and A. Pont, "The impact of the Web prefetching architecture on the limits of reducing user's perceived latency," in Proc. of the International Conference on Web Intelligence. IEEE, 2006.

[20] Bhawna Nigam and Dr. Suresh Jain, "Analysis of Markov Model on different Web Prefetching and Caching schemes", IEEE, 2010.

[21] L. Shi, B. Song, X. Ding, Z. Gu, and L. Wei, "Web prefetching control model based on prefetch-cache interaction," in Proc. First International Conf. on Semantics, Knowledge and Grid SKG '05, 2005.

[22] JyotiPandey, AmitGoel, Dr. A K Sharma "A Framework for Predictive Web Prefetching at the Proxy Level using Data Mining" IJCSNS,VOL.8 No.6, 303-308, June 2008.

[23] F.J. González-Cañete, E. Casilari, A. Triviño-Cabrera "A content-type based evaluation of Web Cache replacement policies" IADIS International Conference Applied Computing 2007.

[24] Achuthsankar S. Nair, Jayasudha J.S, "Dynamic Web Pre-fetching Technique for Latency Reduction", IEEE, 2007.

[25] SirshenduSekharGhosh and Dr. Aruna Jain, "Web Latency Reduction Techniques: A Review Paper", IJITNA vol.1 No.2 pp 20 – 26, September, 2011.

BIOGRAPHY

SirshenduSekharGhosh has completed his M.Tech in IT from BESU, Shibpur, West Bengal, India in June 2010. He has done MCA, M.Sc in Mathematics and B.Sc. Hons. in Mathematics. He has more than 4 years of teaching experience. Currently he is pursuing his full-time Ph.D. research work in IT Dept. of BIT Mesra, Ranchi, Jharkhand, India. His research interest is of Internet Technology and Web Mining.

Dr. Aruna Jain has received her Ph.D. from BIT Mesra, Ranchi, Jharkhand, India in the year 2009. She has done M.Tech in Computer Science and M.Sc. in Physics. She has published around 30 papers in reputed Journals and National and International Conferences. She has acted as resource person in various National and International conferences and editorial board member in reputed Journals. Her fields of Research are Computer Networks & Security, Data Mining, Soft Computing, and Web Engineering. She has more than 20 years teaching experience. Currently she is working as Associate Professor in Department of IT, BIT Mesra, Ranchi, Jharkhand, India and guiding Ph.D. research scholars.