An Enhance Approach for Dynamic Web Caching

Sujit K. Badodia
PG Scholar
Patel College of Science and Technology
Indore, India

Sachin Patel
Asst.Professor
Patel College of Science and Technology
Indore, India

ABSTRACT

There are two types of web projects first one is static web application. In static web application all the pages are linked together there is no connection from database. And other one is dynamic webapplication, for this type of application database is maintained and person can change according to their needs if he has an authentication. In both cases a large amount of data required to add over the pages that are visualized by any internet browser. Dynamic web applications become so popular that the increasing traffic volume threatens to overwhelm the networking capacity in place within corporate Intranets and on the Internet. Due to these increased traffic there is much need to improve the performance of web cache server using more flexible and effective manner. In this paper we propose, design and implement a new approach by which the performances of dynamic web applications are improved. Moreover it we compare the performance of our approach using performance evaluation of our implemented model.

Keywords

Web applications, cache management, networking, improvement, performance.

1. INTRODUCTION

Web servers provide two types of data static data from files stored at a server and dynamic data which are constructed by programs that execute at the time a request is made. The presence of dynamic data often slows down Web sites considerably. High performance Web servers can typically deliver several hundred static files per second, By contrast the rate at which dynamic pages are delivered is often one or two order of magnitudes slower.

Many Web sites today have an increasing need to serve dynamic content. Dynamic content is important for Web sites that provide rapidly changing information, e.g., sports Web sites must provide the latest information about sporting events, and financial Web sites must provide current information about stock prices. If the pages for such Web sites are generated dynamically by a server program that is executed [8] every time the pages are requested, the program can return the most recent version of the dynamic content, whereas if files are created to serve the pages statically, it may not be feasible to keep the files up to date. This is particularly true if there are a large number of files that need to be frequently updated. Dynamic content is also important for creating Web pages on the fly from databases. Search engines satisfy queries dynamically from databases. Web pages corresponding to product catalogs are often created dynamically from databases. Information personalized to individual users is also frequently created dynamically.

Improvement over dynamic data transfer from web server to client machine is an attractive and important task. But due to some network factors system response become slow. As above we can clearly see that dynamic data is changing per second or per user basis thus to create individual page for all these data is quite difficult task. Moreover it data is change according to the current request basis and it is done using data base and database is changed according to the request of user thus we need to implement a new method that improve the working and serving of dynamic data at client end more efficient manner.[1,8]

To solve this problem we propose and implement a web server cache server performance improvement system by which we can easily improve working with dynamic web pages this system is based on the concept of cache memory management scheme.

2. BACKGROUND

In study of different research papers and articles we found that the performance of dynamic web pages are depends upon the following facts.

1. To improve performance of any kind of service we required to add new resources
2. Resource implementation is a complex and expensive task
3. Discover new techniques of resource management
4. Design new techniques to improve performance of current domain using more than single methods
5. Use of optimized and hybrid methods to improve the performance of current system

Now here we provide some literature that provides us a guide line for improving the web server performance of the system.

2.1 Caching/Prefetching Architecture:

The caching/prefetching hierarchical structure envisions collaboration between proxy servers and content distribution servers, as well as between content distribution servers.[2] The lower level proxies cater to users by mainly servicing for their requests, and the higher level content distribution sites use prefetching on effective content distribution. They maintain the content collection using prefetching selection algorithms. New objects are added to the collection of data based on server assistance and user access.

Our prefetching selection algorithms use as input such global access [6] statistics as (1) estimates of object reference frequencies and (2) estimates of object lifetimes. This estimation can be well maintained by content distribution servers if they can collaborate between each other. Content servers collect user access statistics and publish information of objects popularity and patterns of usage; gather usage reports from their users, aggregate and analyse them and make them available to each other. They can also send prefetching hints to each other or actively push to each other objects that are likely to request in the near future. Whenever the replicated object is updated in the original server, the new version of it will be sent immediately to any cache that has subscribed to it. Although it is a key issue to maintain these global access statistics estimates for our prefetching algorithms.[7]

2.2 Prefetching Algorithms

The benefit of web prefetching is to provide low retrieval latency for users, which can be explained as high hit ratio.[4] Prefetching also increases system resource requirements in order to improve hit ratio. [3] Resources consumed by prefetching include server CPU cycles, server disk I/O's, and network bandwidth. Among them, bandwidth is likely to be the primary limiting factor. So, we add moderate bandwidth requirements as an important performance measure for a good web prefetching model. Our solution space for web prefetching spreads according to these two performance evaluation factors: hit ratio and band width consumption. There are four algorithms in our solution space together with demand caching and prefetching all objects in the cache. These prefetching algorithms are a set of threshold based algorithms according to different criteria for choosing which objects to pre-fetch in advance of requests from users. One of them attempts to maintain contents containing the most popular objects in the system.[1] The second approach mostly considers the network resource demands and tries to pre-fetch relatively static objects with a less frequent change rate. [3] In this way, it consumes the least bandwidth. The other two algorithms work in different ways, however both attempt to balance the benefit and the cost.

3. PROPOSED SYSTEM

There are various methods, tools and techniques are proposed and implemented to work with the cache improvement and enhancing the performance of the dynamic web contains. In this section we propose the actual method by which we are going to improve the performance of the web cache. Moreover it here first we discuss the factors that are affected by the web cache contains.

In order to provide the description of the work we tell about that some methods are depends upon the web server, some of them are related to web application architecture and some of them are based on the client server architecture to synchronize the application and improving the performance.

Our concept is based on cache management over the cache memory or the operating systems' memory management scheme, where we calculate the insight factor for the visited web sites.

Below given diagram is our proposed model to implement for the purpose to improve the web server performance. There are four effective and efficient methods are applied for improving the performance. These modules are listed and defined as.

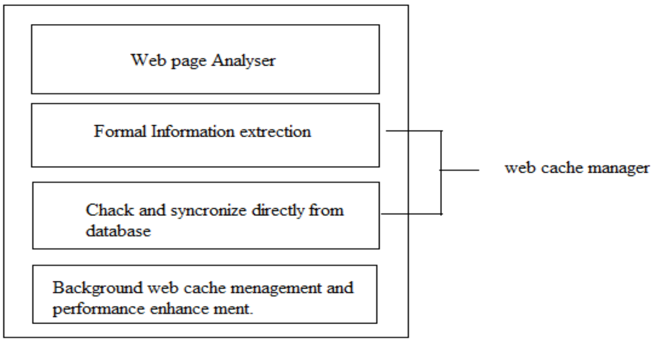

Web page Analyser
Formal Information extrection
Chack and syncronize directly from database
Background web cache menagement and performance enhance ment.
web cache manager

Fig 1 shows proposed system

1. **Web page analyser:** in this module we create and manage the individual page list that is visited by the users, here we extract the formal information that are contain over the pages visited.Such information is related to the source of data other web site references and other use full data contents.

2. Formal information extraction: in this part we calculate the most frequently visited web pages and less visited pages. There are two lists first those web pages that are frequently accessed and second are those that are rarely accessed. Insight factor is a time line that is responsible to get information of the web page access and its frequency.

3. Check and synchronize directly from data base: that is a new fast and efficient data delivery approach by which data is directly delivered to the client end and this process is demonstrated through our application which is promising to improve the performance of the dynamic web site performance.

4. Web cache management: here data cleaning and pruning process is take place. For that purpose when a page is expires its insight factor than it is removed from the cache server analysis list.

4. IMPLEMENTATION

Implementation of the proposed concept is performed using visual studio dot net platform. The dot net framework provides a rich IDE and a rich class library. This library enables to use the different concept of web and remote access.

How it is used?

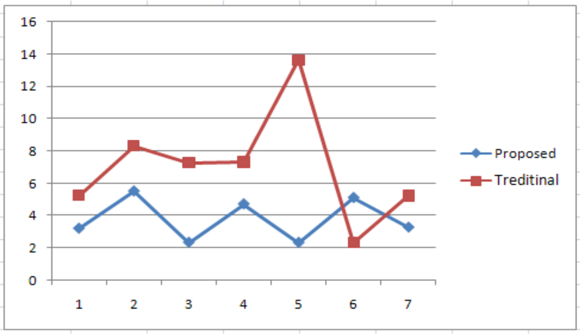
There is server GUI contains some navigation controls to start and end terminates the working of cache server. System start working when the OS starts in other words the application starts with system start-up. And start the collection of information from the user events over internet explorer.

There are two lists one is frequently accessible web continents and second is occasionally access web contents. In back ground all the above listed modules are takes place and there are not any kind of user invigilation is required all the process are triggered itself.

5. RESULTS

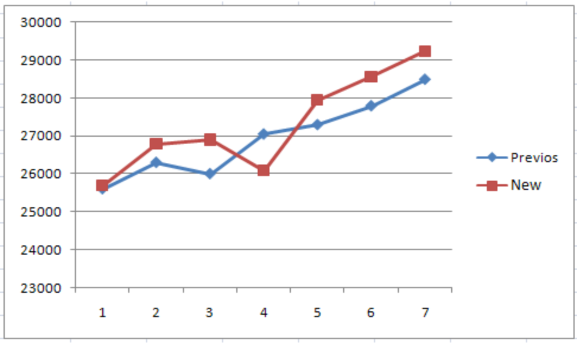
After implementation of the system we provide the here respective results and their conclusions for that purpose here we provide the performance parameters and their outcomes in this section. The performance factor is response time that is providing in seconds between traditional and our proposed system.

No of experiment	Proposed model	Traditional model
1	3.21	5.28
2	5.53	8.32
3	2.34	7.29
4	4.73	7.32
5	2.33	13.63
6	5.12	2.34
7	3.28	5.23



Memory Allocation:

No of experiment	Previous	New
1	25600	25700
2	26300	26800
3	26000	26900
4	27050	26090
5	27300	27950
6	27800	28580
7	28500	29240



The above given chart and graph show our experiment with traditional system and our designed system. And we found that after system implementation the web applications that are served over client machine are performed more efficient than previously designed systems.

6. CONCLUSION

As we proposed the scheme for improving the dynamic web performance is achieved. And the implementation of the system is successfully completed. Additionally we found the effective results from the proposed and designed system that system is performed much better than traditional model of performance enhancement.

In future we work with the same model and work to enhance the application design for the data delivery of the dynamic web application.

7. REFERENCES

[1]. Improving Web Server Performance by Caching Dynamic Data Arun Iyengar and Jim Challenger IBM Research Division T. J. Watson Research Center P. O. Box 704 Yorktown Heights, NY 10598 {aruni, challngr}@watson.ibm.com.

[2]. Techniques for efficiently serving data and dynamic data at webservers using internet and intranet technology Prof. S. N. Gujar1, Prof. G.R.Bamnote2, Prof. R.S.Apare2, Prof. M.A.Pund2, Mr. S.R.Gupta2 1Smt. Kashibai Navale College of Engineering, Pune. India Email: satish_gujar@yahoo.com 2Smt. Kashibai Navale Colege of Engineering, Pune.India Email: ravi_apare@yahoo.co.in 2Prof. Ram Meghe Institute of Technology and Research, Badnera India Email:{ pundmukesh@rediffmail.com,nilguptacse@gmail.com, grbamnote@rediffmail.com}

[3]. Methodologies for Generating HTTP Streaming Video Workloads to Evaluate Web Server Performance, Jim Summers, Tim Brecht University of Waterloo jasummer,brecht@cs.uwaterloo.ca Derek Eager University of Saskatchewan eager@cs.usask.ca Bernard Wong University of Waterloo bernard@cs.uwaterloo.ca

[4]. Cache-based Compaction: A New Technique for Optimizing Web Transfer Mun Choon Chan Thomas Y.C. Woo Networking Software Research Department Bell Laboratories {munchoon, woo}@research.bell-labs.com

[5]. Intelligent Web Caching Using Neuro computing and Particle Swarm Optimization Algorithm Sarina Sulaiman1, Siti Mariyam Shamsuddin2, Fadni Forkan3, Ajith Abraham4 Soft Computing Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia. 4 Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway. sarina@utm.my1, mariyam@utm.my 2, fuelcon@gmail.com3, ajith.abraham@ieee.org

[6]. An Optimization Technique of Web Caching using Fuzzy Inference System Anish Kumar Saha Assistant Professor CSE Department NIT, Agartala, INDIA Partha Pratim Deb M.Tech CSE Netaji Subhash Engg College West Bengal, India Moutushi Kar, D. Rudrapal Assistant Professor CSE Department NIT, Agartala, INDIA

[7]. Web Log Mining for Improvement of Caching Performance Rudeekorn Soonthornsutee1, Pramote Luenam

[8]. Sync Kit: A Persistent Client-Side Database Caching Toolkit for Data Intensive Websites

[9]. Edward Benson, Adam Marcus, David Karger, Samuel Madden {eob,marcua,karger,madden}@csail.mit.edu MIT CSAIL

[10]. Vcache: Caching Dynamic Documents Vipul Goyal1 Sugata Sanyal Dharma P. Agrawal Department of Computer Science & Engg School of Technology & Computer Science Center for Distributed and Mobile Computing, ECECS Institute of Technology, Banaras Hindu University Tata Institute of Fundamental Research University of Cincinnati Varanasi-221005, India Mumbai-400005, India Cincinnati, OH 45221-0030, USA vipulg@cpan.org sanyal@tifr.res.in dpa@ececs.uc.edu