WEB CACHING USING RANDOMIZED ALGORITHM IN WEB PERSONALIZATION

[1]G.N.K.Suresh Babu and [2] Dr.S.K.Srivatsa

1 Apollo Engineering College, Chennai, Tamil Nadu, India

2 St.Joseph College of Engineering, Chennai, Tamil Nadu, India

E-Mail : gnksureshbabu@gmail.com

ABSTRACT

This paper defines the usage of our randomized algorithm in web personalization area. The web personalization is performing under four techniques the handicraft decision technique, hyperlink-based technique, content-based filtering, and collaborative filtering. In our paper we are considering these technique for improve our proxy server's performance. The proposed item specifies the implementation of randomized algorithm is for the website managers establish decision rules according to the statistics of users or session history. So the required files are filtered in an efficient manner according to our algorithm, moreover from this algorithm we improve our system's efficiency, frequency and memory.

Keywords: *Randomized algorithm, web personalization, proxy caching improvement, web cache, Content filtering.*

1. INTRODUCTION

The ordinary browser cannot display the required content for the particular user, in what they are interested. The personalization is concerned with the thing that the user should get the required content without going for the request explicitly.

In order to meet this approach, a browser is designed which can give the user what they want. Here the browser will maintain a database in which all the sites name will be stored and whenever the user visit a site, the browser will update the users profile automatically, like what site they navigate to and in what time and date they have visited that site etc., and if the user visits a site repeatedly it will increase the hit rate (A hit is a request for a file made by a user-agent. User-agents include web browsers and search engine indexing programs, or spiders)

So in an organization each person is interested in viewing the particular site they like most. So for providing the content for each user in an organization the browser will be having an user authentication, when ever the user is sign in, from that moment the user profile like the name of the user and the site they visited, the time in which the user is entered and the time of sign-out will be noted in that database, paralleling the hit rate of the user is also get increased. So whenever the user sign in to the browse at next time the hit rated page (i.e. the page visited by the particular person in an organization) will be shown automatically without an explicit request.

2. WEB PERSONALIZATION TECHNIQUES

Web Personalization is the process is followed under the following four techniques. They are:

1. The handicraft decision technique
2. Hyperlink-based technique
3. Content-based filtering
4. Collaborative filtering

The handicraft decision technique means that website managers establish decision rules according to the statistics of users or session history. To take advantage of these rules, the

recommendation system provides particular contents and web structures to particular sorts of users. This kind of system functions easily, but its efficiency is low and it is difficult to renew in a timely fashion. The hyperlink-based technique generally uses an algorithm related to diagram theory to discover the most representative elements provided by the user input or information request. Search engines mostly use this technique. The famous Google search engine is one notable example.

Content-based filtering examines the relationship between resources and users. It takes advantage of the similarity of information and users' interests to filter information. The weakness of the approach is that it is difficult to discern the quality and form of the information sought in current user sessions and it can only recommend information similar to the user's previously-identified interests, rather than advising on newly-identified resource needs arising from concurrent sessions. Content-based filtering establishes user profiles according to each user are visiting history and content searches and then classifies website content on this basis. When a user visits the website and seeks resources that match the established profile, the related information is retrieved.

Collaborative filtering compares the relationship between one user and another. It makes use of similarities among users to filter information and provides the similar user with commonly-sought information. Its advantage is that it can discover new information a user may be interested in but is not asking for. However, there are two problems: one is scarcity, since when the system is initiated the lack of system resource evaluations means the system will not easily identify similar users . The other is extensibility. As the system users and resources increase, the performance demands on the system will increase.

Compared to content-based filtering, collaborative filtering is more convenient for analyzing users' behaviors. What is analyzed is the click rate of the web page or the content in the web page, not the web page content, itself. Click rates could represent user favorites. It is generally thought that for similar web page content, users with similar click rates will exhibit similar visiting habits and usage patterns. Hence, systems can sort users based on the click rate of the web page content.

3. WEB CACHE

A *Web cache* sits between Web servers (or *origin servers*) and a client or many clients, and watches requests for HTML pages, images and files (collectively known as *objects*) come by, saving a copy for itself. Then, if there is another request for the same object, it will use the copy that it has, instead of asking the origin server for it again.

There are two main reasons that Web caches are used:

- To **reduce latency** - Because the request is satisfied from the cache (which is closer to the client) instead of the origin server, it takes less time for the client to get the object and display it. This makes Web sites seem more responsive.
- To **reduce traffic** - Because each object is only gotten from the server once, it reduces the amount of bandwidth used by a client. This saves money if the client is paying by traffic, and keeps their bandwidth requirements lower and more manageable.

4. KINDS OF WEB CACHES

Browser Caches

If you examine the preferences dialog of any modern browser (like Internet Explorer or Netscape), you'll probably notice a 'cache' setting. This lets you set aside a section of your computer's hard disk to store objects that you've seen, just for you. The browser cache works according to fairly simple rules. It will check to make sure that the objects are fresh, usually once a session (that is, the once in the current invocation of the browser).This cache is useful when a client hits the 'back' button to go to a page they've already seen. Also, if you use the same navigation images throughout your site, they'll be served from the browser cache almost instantaneously.

Proxy Caches

Web proxy caches work on the same principle, but a much larger scale. Proxies serve hundreds or thousands of users in the same way; large corporations and ISP's often set them up on their firewalls. Because proxy caches usually have a

large number of users behind them, they are very good at reducing latency and traffic. That's because popular objects are requested only once, and served to a large number of clients. Most proxy caches are deployed by large companies or ISPs that want to reduce the amount of Internet bandwidth that they use. Because the cache is shared by a large number of users, there are a large number of *shared hits* (objects that are requested by a number of clients). Hit rates of 50% efficiency or greater are not uncommon. Proxy caches are a type of *shared cache*.

Replacement algorithm design

N: Total number of Documents
M: Next Least useful document
Eviction: Retrieval the requesting document
Sample: Method for selecting the retrieval option

```
If (eviction)
{
        if (first_iteration)
   {

sample(N);

evict_least_useful;

keep_least_useful(M);

   }

else
 {
        sample(N-M);

evict_least_useful;

keep_least_useful(M);

 }
}
```

Replacement algorithm designing includes two iteration models. in the first iteration step the N-documents is randomly picked from the cache and among that N-documents the least useful document is evicted and then next M least useful document is retained and in subsequent iteration the N-M documents is randomly picked from cache And it is Appended to the M previously retained documents and among the N-samples the least useful document is evicted and M next least useful document are retained.

Combine Results of randomized algorithm and Personalization Techniques:

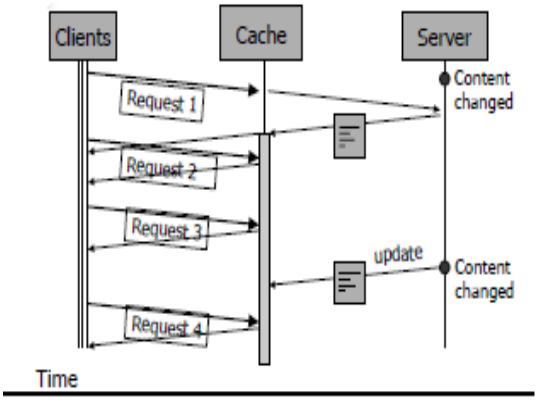


Fig (1). Update

1. Server remembers where its pages are cached
2. When a page is modified, server notifies the caches and gives them an updated version of the page (usually used together with application layer multicast)
3. Caches always assume that they have the

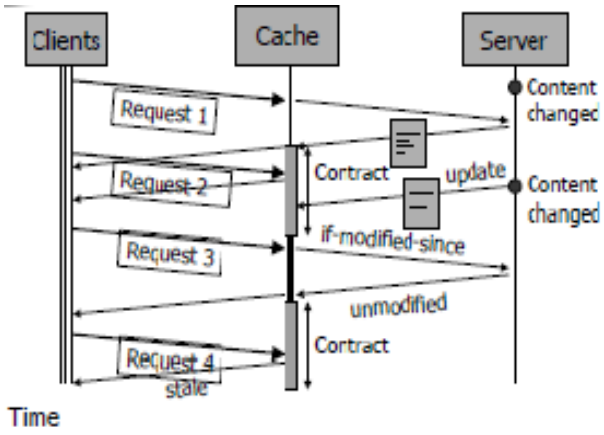


Fig (2)

1. Invalidation / Update traffic can get very large
2. To limit this traffic server gives invalidation or update contracts to caches
3. Contracts have expiration time
4. When content changes server notifies only those caches whose contract has not expired

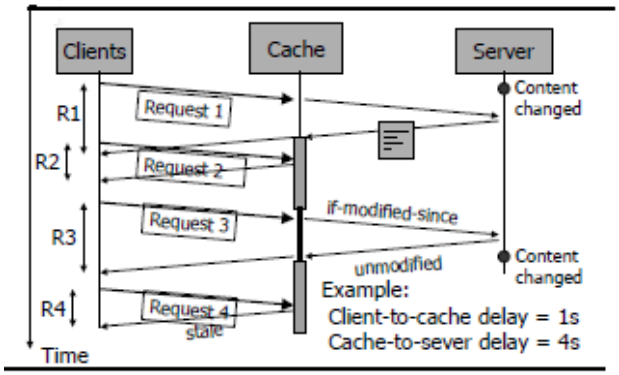
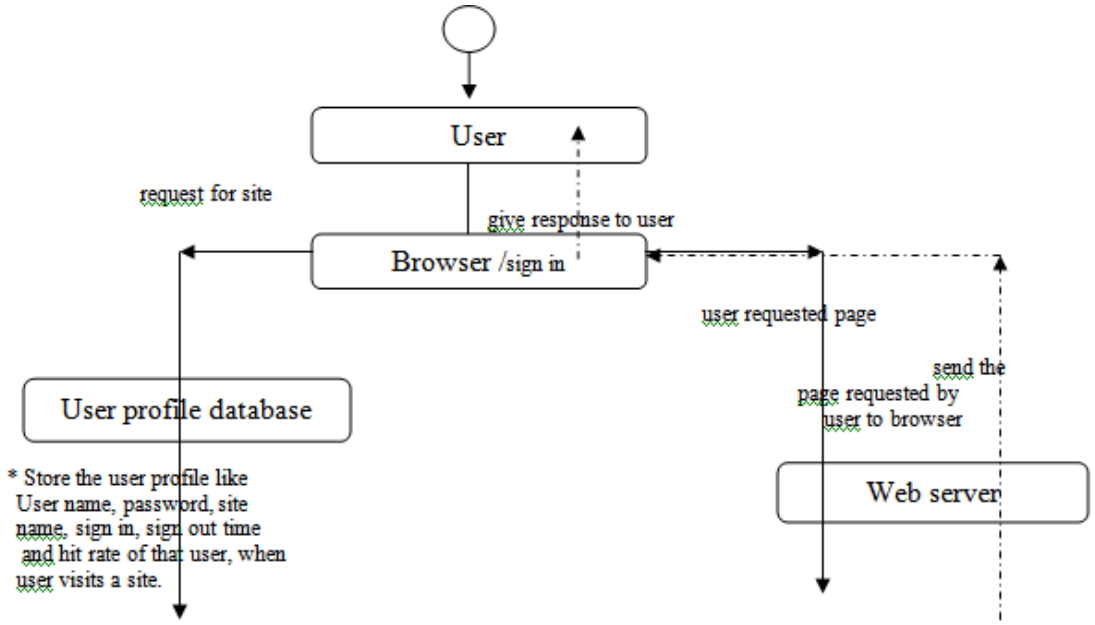
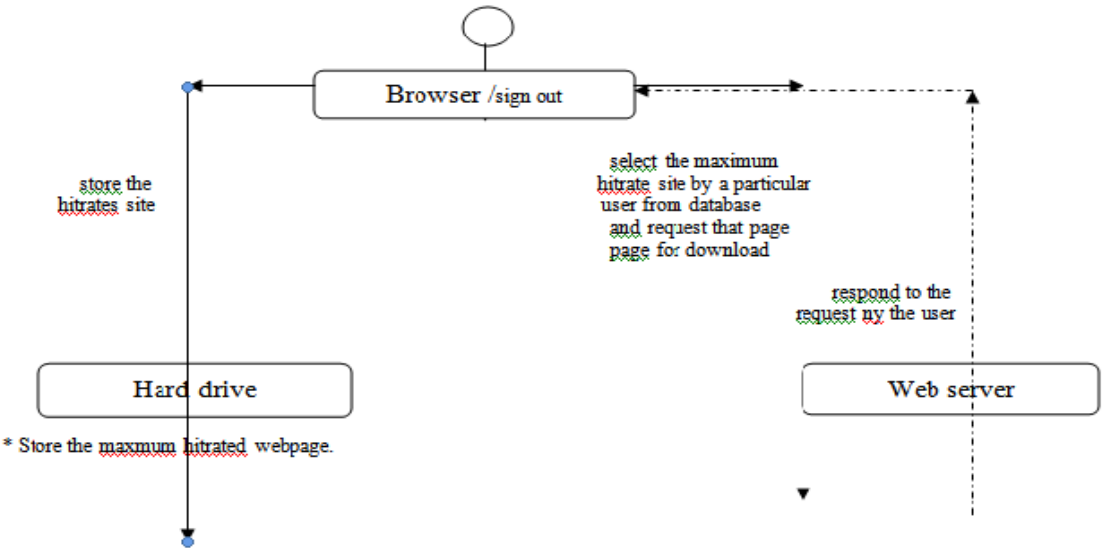


Fig (3). The Average Response Time

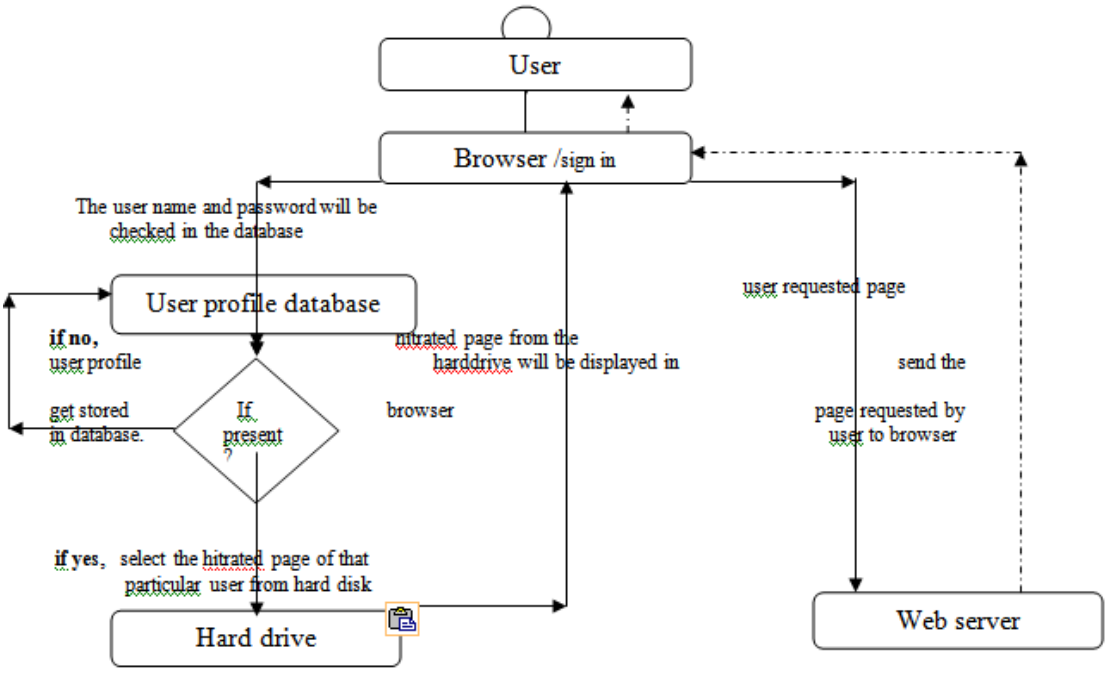
1. Invalidation / Update traffic can get very large
2. To limit this traffic server gives invalidation or update contracts to caches
3. Contracts have expiration time
4. When content changes server notifies only those caches whose contract has not expired



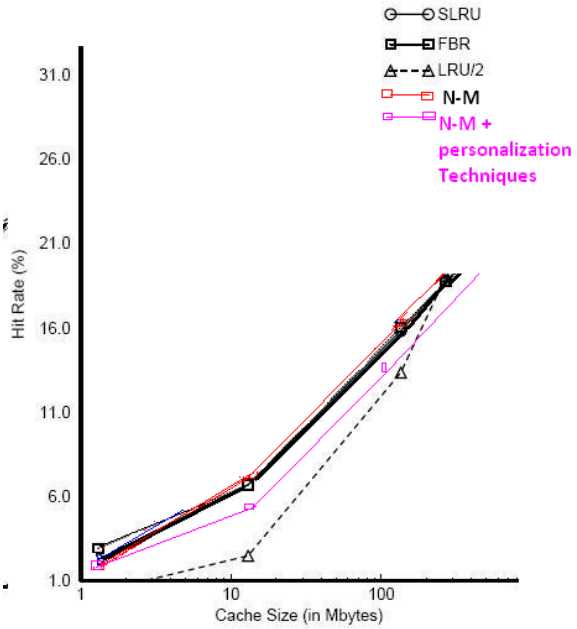
When the user sign in for the first time, he can navigate to any site what he wants to visit and the response for the request is provided from the web server, in meantime the user details will be stored in the database (e.g. site navigation with time and date factor along with his/her credentials) until the user sign-out.



When the user sign-out the browser will automatically check for the hit rated page and it will store the hit rated page to a local hard drive. This is achieved by the browser as, it will request that page automatically from the web server using the maximum hit rate.



when the user sign in for the next time, it will check whether there is any hit rated page for that particular user, if it present, that page will be automatically displayed by the browser to the user. The user can also navigate to any site he wants. Again his details will be updated in the database. This will be applicable for the entire user in an organization, as each person has their own username and password.



The Hit ratio for our algorithm is implemented; here we compare our algorithm with SLRU (second Least recently used), FBR and LRU/2. The ratio of our algorithm has slight difference from the other algorithms retrieval. In the above figure we describe the hit ratio along with our cache size. The N-M with personalization techniques is implemented above has great variation than others.

5. CONCLUSION

Web caching works because of popularity Ñ the more popular a resource is, the more likely it is to be requested in the future. In one study spanning more than a month, out of all the objects requested by individual users, on average close to 60 percent of those objects were requested more than once by the same user. Likewise, much content is of value to more than one user. In fact, of the hits recorded in another caching study, up to 85 percent were the results of multiple users requesting the same object. Three features of Web caching make it attractive to all Web participants, including end users, network managers, and content creators.

- Caching reduces network bandwidth usage, which can save money for both content consumers and creators
- Lessens user-perceived delays, which increases user-perceived value; and
- Lightens loads on the origin servers, saving hardware and support costs for content providers and providing consumers a shorter response time for nonrated resources.

REFRENCES:

[1] Aggarwal C, J.L. Wolf, and P.-S. Yu, "Caching on the World Wide Web," IEEE Trans. Knowledge and Data Eng., vol. 11, no. 1, pp. 94- 107, Jan. /Feb. 1999.

[2] Bahattab, A.A. Bodnar, B. Kraft, G. Evens, M. Comput. Technol. Dept., Jeddah Coll. of Electron. Tech., Saudi Arabia; "Producing a high hit ratio and low search time in forwarding routing tables by predicting IP addresses"

[3] "Efficient and Anonymous Web-Usage Mining for Web Personalization" Cyrus Shahabi Farnoush Banaei-Kashani

[4] "Algorithm of Mining Sequential Patterns for Web Personalization Services", Cui Wei Wu Sen Zhang Yuan Chen Lian-chang

[5] A Web Caching Primer Brian D. Davison Rutgers, The State University of New Jersey (USA) IEEE. Reprinted from *IEEE Internet Computing* ,Volume 5,Number 4, July/August 2001

[6] Proxy Cache Replacement Algorithms: A History-Based Approach Department of Informatics, Aristotle University

[7] Web Caching and Zipf-like Distributions: Evidence and Implications Lee Breslau, Pei Cao, Li Fan, Graham Phillips, Scott Shenker. IEEE INFOCOM, VOL. XX, NO. Y, MONTH 1999

[8 Web Mining for Web Personalization MAGDALINI EIRINAKI and MICHALIS VAZIRGIANNIS Athens University of Economics and Business

[9] Oracle9*i*AS Web Cache Overflow Vulnerability, Reference Date: October 18, 2001

[10] Performance evaluation of Web Cache Pawan Kumar Choudhary and Kishor S. Trivedi Duke University, Durham, NC 27708

[11] Sitaram Iyer Antony Rowstron Peter Druschel Squirrel: A decentralized peertopeer web cache ACM Symposium on Principles of Distributed Computing (PODC 2002)