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

ABSTRACT

Caching and Prefetching are the two approaches for Web Caching. Again Web caching is a technology to reduce the response time, bandwidth uses and improving the network traffic etc. Web Prefetching tries to put the future used web objects into cache with higher probability of cache hit. In Web caching, Cache replacement algorithm is the core of it. So, good replacement policy would make effective management of cache memory utilization with higher probability of cache hits. General replacement policy like LRU, FIFO, LFU considering only the arrival time, but other parameters related to web objects should consider for deciding cacheable or not. This paper approaches a replacement policy with fuzzy inference system with input parameters Frequency, Latency and Bytesent of web objects. By considering these parameters, the replacement would have artificial intelligence in cache replacement policy.

Keywords

Web Caching; FIS (Fuzzy Inference System); Frequency; Latency; Bytesent.

1. INTRODUCTION AND RELATED WORKS

Users of internet are increasing day by day. But the bandwidth is limited. As more space of our bandwidth is used, latency of documents retrieved from the Internet increases. Despite the fact that the web is growing fast, the same documents get requested and the same web sites are visited repeatedly. We can take advantage of this to avoid the downloading redundant objects. That's why web caching is introduced to place the redundant objects closer to the clients that are likely to be used in the near future. But, we need to use the cache memory of proxy server [12] in a convenient way to get the good result of web caching.

Web pages can be cache at Client, Proxy Server and Original Server. Based on the location, caching can be Browser Caching, Client Side Proxy Caching, or Server Side Proxy Caching [1]. Browser caching is economical and effective way to improve the performance of the World Wide Web due to the nature of Browser Cache, which is closer to the user. In Client Side Caching, the web objects are placed near to the clients. And in server-side web caching, the cache pages is near to the server to reduce the Web server's load and, thus, to shorten the user perceived response time.

Three main important issues that have profound impact on caching management namely: Cache Replacement, Cache Consistency and Cache Prefetching [2].

Cache replacement algorithm is required when cache is full and need to delete or replace some web objects to give place for the new coming web objects. The cache replacement is the core of the web caching; hence, it is very necessary to create a good replacement algorithm for caching mechanisms achievement. In general, cache replacement algorithms are also called web caching algorithms [2]. There are some of the important cache replacement policies: first-in/first-out (FIFO), least recently used (LRU), least frequently used (LFU), and so on. They consider only one factor, which is Recency or arrival time of the web objects.

Other than Recency parameter, other parameters can be the page size, fetching delay, reference rate, invalidation cost, invalidation frequency of a Web object etc [15]. These parameters should consider whenever a good replacement algorithm required. Using these parameters into their designs, these cache replacement algorithms shows good performance improvement over the conventional ones like LRU, FIFO, LFU, and so on.

Then come to the Cache consistency. It is concern to determine that the contents in the cache should be up to date when the objects get changed on origin server. That means the original sever contents and proxy contents for some web objects should be same. Most frequent changing web objects are not good for Web Caching due to its frequent inconsistency state. Cache consistency is classified into 2 groups. One is weak cache consistency (Less freshness) and another strong cache consistency (High Freshness). Different examples of Weak consistency are TTL (Time-To-Live), PCV (Piggyback Client Validation), PSI (Piggyback Server Invalidation) etc. Different examples of strong consistency are Polling-every-time, Invalidation information from server, Lease in Client/server Interaction [3]. So, Which Cache Consistency algorithm would be applied is fully based on the application type and its needs.

Now comes to the point, Prefetching, which aims at predicting future requests for Web objects and bringing those objects into the cache in the background, before an explicit request is made for them. Web caching & Web Prefetching can complement each other since the Web caching technique exploits the temporal locality, whereas Web Prefetching technique utilizes the spatial locality [1] of Web objects. Different types of Prefetching methods are there. These are Prefetching hyperlinks, Association relation, History based, Page rank based [1].

Except these, Web log mining is also used for Prefetching [5] [6] [7] [8] and other adaptive Prefetching algorithms are already there [2]. So, we can conclude that web caching and Prefetching both are the two techniques use to reduce the response time perceived by the user. Recently artificial intelligence is applied in almost every field to get human like thinking work done by machine. Fuzzy logic, Neural Network, Genetic Algorithms all is having their roots in Artificial Intelligence.

2.PROPOSED ARCHITECTURE

Only one parameter Recency or arrival time is not sufficient for good replacement of web objects from Proxy Server Cache memory. Some examples of arrival based cache replacement algorithm are LRU, LFU, and FIFO etc. Here three parameters of the web objects are used for cache replacement policy. These are Frequency, Latency and Bytesent. The First parameter is Frequency. Frequency mean number of cache hits of the web objects for a period of time in the proxy server. Higher Frequency value would give higher probability of cache hits and providing quick retrieval of web objects to clients from Proxy Server. The second parameter is Latency. Latency means time to get the requested web object from server. If the web objects required higher Latency, it would be given higher priority of cache. Because, if the high Latency web objects are place into cache of proxy server, the repetitive client requests would be deliver quickly to the clients from proxy server thus providing quick retrieval. The third parameter **Bytesent** means the size of the web object or the byte sent by the server to the Proxy Server for the requested web object. The Bytesent or the size of the object is crucial for replacement policy. If the object is more size, then it would be given high priority of cache, because higher web object size required more bandwidth for transmission. So, it would be better to place higher size web objects into Cache of proxy server which is near to the clients and serving repetitive client request from proxy server, thus reduces bandwidth.

The proposed architecture cab be explained as

- When the users navigate web pages, all web objects are placed in Proxy Server cache primarily. The information Latency and Bytesent of the web objects are saved in proxy server data store. The proxy server maintains cache hits of the each web objects for a period of time and it is acting as Frequency of the web objects. After a period of time, the cache memory of proxy server would filled up. When new web objects request come for placing in the cache of the proxy server, then the replacement policy is applied for freeing space.
- The Sugeno type Fuzzy Inference System is employed here to find a rank of web objects. This rank helps in replacement policy. The rank is based on the input values of Byte Sent, Frequency and Latency. The low valued rank web objects would be removed first from cache of proxy server. If all the web objects consider high rank then, arbitrarily any web objects would removed for new coming web objects into cache memory of Proxy Server.

3. FIS STRUCTURE

The Fuzzy Inference System (FIS) can be developed set of IF Then rules and determine the membership function for the inputs. Here Sugeno type FIS is used. The sugeno based fuzzy Inference System can be explained as

$$\text{IF } (x_1 \text{ is } A_1)\text{AND}(x_2 \text{ is } A_2)\text{AND}(x_3 \text{ is } A_{3)} \ldots \ldots \text{AND } (x_n \text{ is } A_n)$$

$$\text{THEN} \quad y = f(x_{1,}\ x_2, x_3 \ldots x_n)$$

Where $x_1, x_2, x_3 \ldots \ldots x_n$ are input variables, $A_1, A_2, A_3, \ldots A_n$ are fuzzy sets and y is either constant or linear function of input variables.

In this proposed model, parameters or inputs used here in the FIS are Frequency means number of cache hits for particular period of time, Latency is the fetching delay or retrieval delay from server and Bytesent is the size of the web object or byte sent from original server to proxy server. Fig 1 Show the FIS structure.

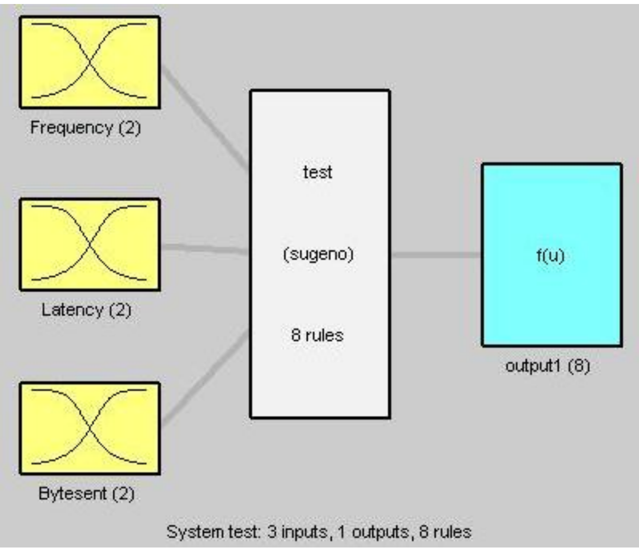


Fig 1: FIS and its input parameters and output

The membership function of each input is divided into two regions namely, small, and large. Here all the membership functions are bell shape member function. The curves of membership functions are shown at fig 2.

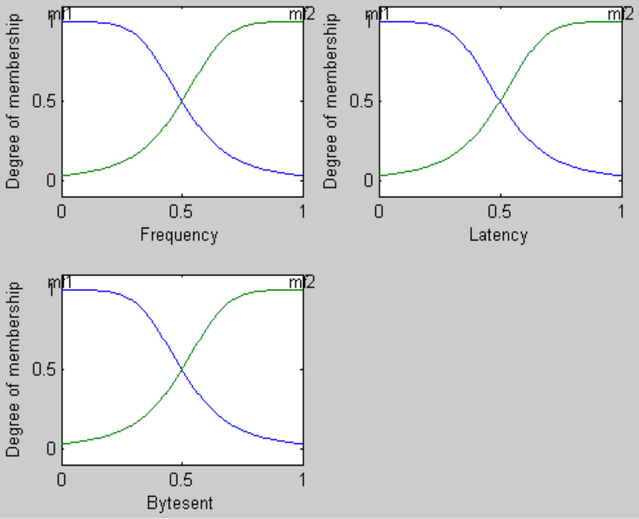


Fig 2: Bell shaped membership functions of inputs of FIS. The inputs are ByteSent, Frequency, and Latency.

Membership function Range from [0 1], that means the input values must be between 0 and 1.

The inputs are the ratio of $\frac{\text{Current Value}}{\text{Maximum Value}}$ of the web objects different parameter.

Here the inputs,

$$\text{Bytesent} = \frac{\text{Current Bytesent Value}}{\text{Maximum Bytesent Value}}$$

$$\text{Latency} = \frac{\text{Current Latency Value}}{\text{Maximum Latency Value}}$$

And

$$\text{Frequency} = \frac{\text{Current Web object FrequencyValue}}{\text{Maximum Frequency Value}}$$

If $\frac{\text{Current Value}}{\text{Maximum Value}} > 1$ then, the input would be considered as 1 and the Maximum value would be updated with the Current Value .

Since it is Sugeno type FIS and it has 3 variables and each variables having 2 regions "Small" and "Large". So, the combination of rules is $\mathbf{2^3}$ i.e. **8.** The outputs would be total 8 numbers of constant type membership functions.

FIS Information:

Type: Sugeno
Type of input Member Function: Bell function
No of Member Functions: 2 MFs for each input.
Type of output Member Function: Constant.
Number of fuzzy rules: 8

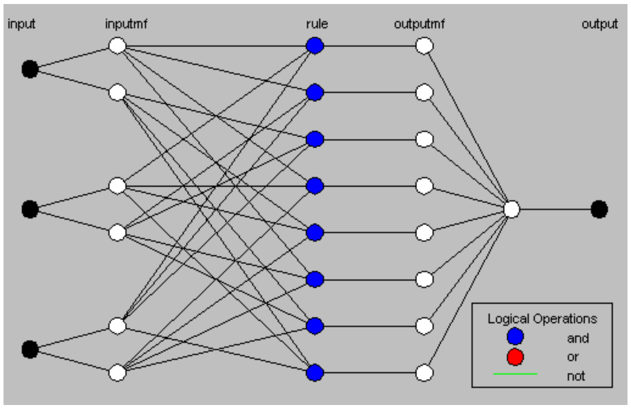


Fig 3: FIS structure with different layers.

The Fig 3 FIS structure can be explained as Layer-1: Inputs Layer, Layer-2: Membership function for inputs, Layer-3: Rules, Layer-4: Output Membership function, Layer-5: Weighted sum output and Layer-6: Output.

4. IMPLEMENTATION

The response time, Delay of web objects can be obtained from the Log file of the proxy server. The Log file can be look like as in Fig 4.

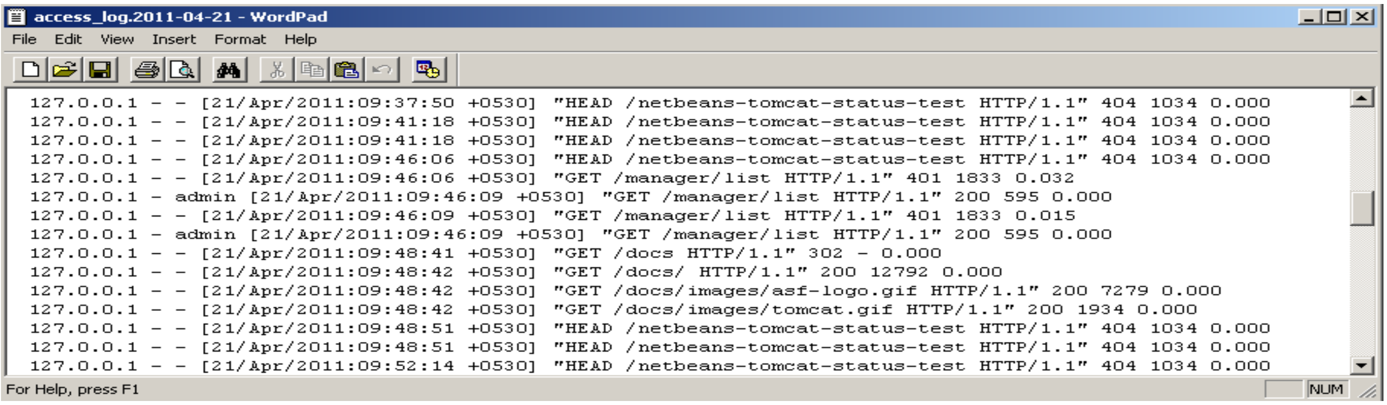
Log file which maintains a history of web object requests and use to analysis the user request.

The general contents of log files are

%a - Remote IP address
%A - Local IP address
%b - Bytes sent, excluding HTTP headers, or '-' if zero
%B - Bytes sent, excluding HTTP headers
%h - Remote host name (or IP address)
%H - Request protocol
%l - Remote logical username
%m - Request method (GET, POST etc.)
%p - Local port on which this request was received
%q - Query string (prepended with a '?' if it exists)
%r - First line of the request (method and request URI)
%s - HTTP status code of the response
%S - User session ID
%t - Date and time, in Common Log Format
%u - Remote user that was authenticated (if any), else **'-'**
%U - Requested URL path
%v - Local server name
%T - Time taken to process the request, in seconds
%I - current request thread name (can compare later with stack traces)

The pattern code is placed in configuration file of the proxy web server. The proxy server would create a log file with the required following data.

Out of the above here, only **Response time** (%T) acting as **Latency** and **Byte sent** (%B) is retrieve for each web objects. The **Frequency** for each web objects is maintained by proxy server. The proxy server counts the number of cache object hits for a particular period of time acting as Frequency for that web object.



```
127.0.0.1 - - [21/Apr/2011:09:37:50 +0530] "HEAD /netbeans-tomcat-status-test HTTP/1.1" 404 1034 0.000
127.0.0.1 - - [21/Apr/2011:09:41:18 +0530] "HEAD /netbeans-tomcat-status-test HTTP/1.1" 404 1034 0.000
127.0.0.1 - - [21/Apr/2011:09:41:18 +0530] "HEAD /netbeans-tomcat-status-test HTTP/1.1" 404 1034 0.000
127.0.0.1 - - [21/Apr/2011:09:46:06 +0530] "HEAD /netbeans-tomcat-status-test HTTP/1.1" 404 1034 0.000
127.0.0.1 - - [21/Apr/2011:09:46:06 +0530] "GET /manager/list HTTP/1.1" 401 1833 0.032
127.0.0.1 - admin [21/Apr/2011:09:46:09 +0530] "GET /manager/list HTTP/1.1" 200 595 0.000
127.0.0.1 - - [21/Apr/2011:09:46:09 +0530] "GET /manager/list HTTP/1.1" 401 1833 0.015
127.0.0.1 - admin [21/Apr/2011:09:46:09 +0530] "GET /manager/list HTTP/1.1" 200 595 0.000
127.0.0.1 - - [21/Apr/2011:09:48:41 +0530] "GET /docs HTTP/1.1" 302 - 0.000
127.0.0.1 - - [21/Apr/2011:09:48:42 +0530] "GET /docs/ HTTP/1.1" 200 12792 0.000
127.0.0.1 - - [21/Apr/2011:09:48:42 +0530] "GET /docs/images/asf-logo.gif HTTP/1.1" 200 7279 0.000
127.0.0.1 - - [21/Apr/2011:09:48:42 +0530] "GET /docs/images/tomcat.gif HTTP/1.1" 200 1934 0.000
127.0.0.1 - - [21/Apr/2011:09:48:51 +0530] "HEAD /netbeans-tomcat-status-test HTTP/1.1" 404 1034 0.000
127.0.0.1 - - [21/Apr/2011:09:48:51 +0530] "HEAD /netbeans-tomcat-status-test HTTP/1.1" 404 1034 0.000
127.0.0.1 - - [21/Apr/2011:09:52:14 +0530] "HEAD /netbeans-tomcat-status-test HTTP/1.1" 404 1034 0.000
```

Fig 4: Log file Contents format.

When the cache memory is filled up with web objects, then the replacement algorithm would apply for freeing space for new coming web objects. The FIS would give a rank for each web objects. The rank range is [0 1] and the higher rank web objects would give high priority to be cache and whereas the low priority web objects would be removed early.
The ruler view is shown as in fig 5.

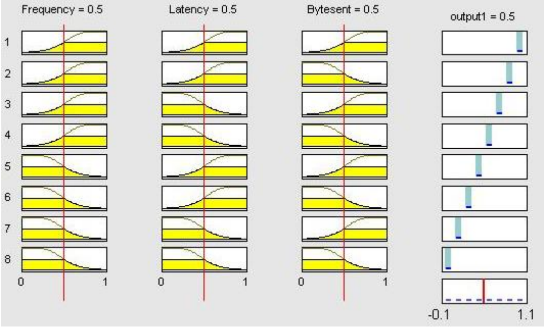


Fig 5: Ruler view of Inputs with Outputs.

The simulator is done with JDK 1.5 Netbeans IDE and Matlab 2009. JMatlink is used to Connection between JDK 1.5 and it is free software and can be downloaded from the link http://jmatlink.sourceforge.net/. The Fuzzy Logic Toolboxes in Matlab 2009 is used to design the Fuzzy Inference System.

5. CONCLUSIONS AND FUTURE WORK

The membership functions of the Fuzzy Inference System need to refine its curves based on live data. So, live data required. Next plan is to implement in practical place to check feasible or not. It would be better to use a trained Fuzzy Inference System and Adaptive Neuro Fuzzy Inference System would be the solution for it. So, next plan is to modify the approach with ANFIS (Adaptive Neuro Fuzzy Inference System) to refining the membership function curves based on input to output data. The propose paper is considering only Web Caching, but Web Prefetching would be another approach in which the anticipated future web objects are placed in cache. Next plan is to propose innovative Prefetching algorithm and integrate both the technique Web Caching and Prefetching into the same Proxy Web Server. It is mentioned that the limitation of this approach required extra computation overhear for the proxy server and implementation complexity.

6. REFERENCES

[1]. Wei-Guang Teng, Cheng-Yue Chang, and Ming-Syan Chen "Integrating Web Caching and Web Prefetching in Client-Side Proxies" Fellow, IEEE.

[2]. Jaeeun Jeon, Gunhoon Lee, Ki Dong Lee and Byoungchul Ahn, "An Adaptive Prefetching Method for Web Caches" Yeungnam University, School of Electrical Engineering and Computer Science.

[3]. L. Y. CAO and M. T. OZSU "Evaluation of Strong Consistency Web Caching Techniques" University of Waterloo, School of Computer Science, Waterloo, ON, Canada N2L 3G1.

[4]. Antonis Sidiropoulos, George Pallis, Dimitrios Katsaros, Konstantinos Stamos,Athena Vakali, Yannis Manolopoulos "Prefetching in Content Distribution Networks via Web Communities Identification and Outsourcing" World wide Web (2008) 11:39–70; Springer.

[5]. Qiang Yang, Henry Haining Zhang, Ian T.Y. Li, and Ye Lu "Mining Web Logs to Improve Web Caching and Prefetching" School of Computing Science Simon Fraser University Burnaby, BC, Canada V5A 1S6 (qyang, hzhangb, tlie, yel)@cs.sfu.ca

[6]. Areerat Songwattana "Mining Web logs for Prediction in Prefetching and Caching" School of Engineering and Technology, Asian Institute of Technology Rangsit, Pathumthani, 12000, Thailand areerat@rangsit.rsu.ac.th, sareerat@gmail.com

[7]. Qiang Yang and Haining Henry Zhang "Web-Log Mining for Predictive Web Caching".

[8]. Francesco Bonchi[1], Fosca Giannotti[2], Giuseppe Manco[3],Chiara Renso[4],Mirco Nanni[5],Dino Pedreschi[6],Salvatore Ruggieri[7] "Data Mining for Intelligent Web Caching".[1,2,3,4] CNUCE-CNR - Institute of Italian National Research Council[5,6,7] Department of Computer Science, University of Pisa

[9]. Víctor J. Sosa Sosa[1], Gabriel González S[2]., Leandro Navarro[3], "Building a Flexible Web Caching System" [1, 2] Centro Nacional de Investigación y Desarrollo Tecnológico, Interior Internado Palmira S/N, Cuernavaca,Morelos, México.Universitat Politècnica de Catalunya (UPC)Jordi Girona, 1-3, D6-105, Campus Nord,Barcelona, Spain.

[10]. Sarina Sulaiman[1], Siti Mariyam Shamsuddin[2], Fadni Forkan[3], Ajith Abraham[4] "Intelligent Web Caching Using Neuro computing and Particle Swarm Optimization Algorithm"[1,2,3]Soft Computing Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia.[4]Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway.

[11]. Michael Chau, and Hsinchun Chen "Incorporating Web Analysis Into Neural Networks: An Example in Hopfield Net Searching" IEEE

[12]. S. V. Nagaraj "CACHING AND ITS APPLICATION" Chapter:1 VARIOUS FLAVORS OF WEB CACHING, page 3-16

[13]. Konstantinos Stamos, George Pallis, and Athena Vakali "Integrating Caching Techniques on a Content Distribution Network" Konstantinos Stamos, George Pallis, and Athena Vakali, Department of Informat Aristotle University of Thessaloniki, 54124, Thessaloniki, Greece.

[14]. S. V. Nagaraj "CACHING AND ITS APPLICATION" Chapter:11 PREFETCHING, page 105-122

[15]. Waleed Ali & Siti Mariyam Shamsuddin "Neuro-Fuzzy System in Web Client-side Caching" Faculty of Computer Science & Information System, University Technology of Malaysia

[16]. Josef Schmidbauer and Hilmar Linder "Utilizing Layered Multicast for Web Caching" Department of Scientific Computing, Paris-Lodron University of Salzburg, Salzburg, Austria