ENHANCING PROXY BASED WEB CACHING SYSTEM USING CLUSTERING BASED PRE-FETCHING WITH MACHINE LEARNING TECHNIQUE

V.Sathiyamoorthi[1], P.Ramya[2]

[1]*Assistant Professor, Department of Computer Science, Sona College of Technology, Sona Nagar, Salem, Tamilnadu.*
[2]*PG Scholar, Department of Computer Science, Sona College of Technology, Sona Nagar, Salem, Tamilnadu.*

Abstract

Data Mining is a process of extracting knowledge from various data source. Web usage mining is the area of data mining which deals with the discovery and analysis of usage patterns from web data, specifically web logs, in order to improve web based applications. Considering these, Web caching is a Data mining technique which is used to reduce user perceived latency when user is accessing the web pages. Web pre-fetching is the scheme where web pages are pre fetched into the intermediate server (proxy) cache before user accessing it. These two techniques can complement each other since web caching exploits the temporal locality whereas web pre-fetching utilizes the spatial Locality of web objects. To improve the performance of web access, web caching and pre-fetching techniques are being integrated using clustering based pre-fetching algorithm. In this paper, pre-fetching using clustering technique is combined with SVM (Support Vector Machine)-LFU algorithm, a machine learning technique for web proxy caching. By analysis it is shown that SVM technique is better than clustering based pre-fetching technique using caching policy like LFU considering bandwidth utilization and access latency.

Keywords: *Web Caching, Web Pre-Fetching, Clustering, SVM Machine Learning Technique.*

----------***----------

1. INTRODUCTION

In today's world, the speed of distribution of information from one place to another is of vital importance. Web pages which are quite slow to download may create a lack of user's interest in accessing them. Hence to retain the viewership, we have to ensure that the pages are accessed quite fast. The traditional cache replacement techniques used often fail to increase the cache hit ratio. In this project different mechanisms are being proposed for improving the web cache performance. One such technique is integration of web pre-fetching and caching to anticipate the user's requests and also the clustering technique is used so that the pre-fetching engine performs faster and responds to the user's requests immediately with most related web objects. As web caching, pre-fetching and clustering technique plays a vital role here we discuss in detail on all data mining techniques. After the clustering process, it is combined with the SVM-LFU algorithm in order to show that the performance improvement using SVM would be comparatively better than implementing with clustering based pre fetching technique. The performance is measured using the performance metrics as HR (hit ratio) and BHR (byte hit ratio).In section 2 discussion is based with the techniques as Web Caching and Pre Fetching, in section 3 Clustering based Pre Fetching technique is discussed, section 4 deals with the Classifier technique i.e Support Vector Machine (SVM) and the combination of SVM and Cluster is discussed and final section deals with the performance of combining the SVM with Cluster is discussed.

2. WEB CACHING

Web caching is a technique which is used for caching as many web pages in the cache to improve network performance.web caching can be done either at the client side, or at the proxy server or at server side. Web caching aims to reduce network traffic, server load, and the user perceived retrieval latency. The main component of a caching system is its page replacement policy, which needs to make good replacement decisions when its cache is full and a new document arrived. In this work, web usage mining is used to optimize the existing web cache policy for better performance.

2.1. Caching Strategies

Caching strategies decide which document to remove from the cache when space is needed to new ones. Their goal is to make the usage of the cache as efficient as possible, by trying to optimize one or several of the performance criteria of caching cache hit rate, byte hit rate, or response time. Additionally the strategies themselves should work efficiently, i.e CPU and memory usage should also be as low as possible. To fully optimize the cache usage, all future requests must be known which impossible. Therefore the strategies use heuristics to

decide which documents to replace and which to keep. Few important factors that influence the replacement strategies are:

- **Recency**: time of the last reference to the object.
- **Frequency**: number of request to an object
- **Size:** size of the web object
- **Cost of retrieval:** cost to fetch an object from original server.

When these factors are taken into account it includes benefits as, reducing the cost of connecting to the internet. Reducing the Latency of today's WWW.

Some of the standard strategies used are,
LRU: Least Recently used (LRU) is the most common caching strategy. The items not used for the longest time are removed from the cache to make way for new ones. This algorithm leads to a high rate while causing little overhead.
LFU: Least frequently used (LFU) is the caching strategy where the items not used for long time are removed from the cache to make way for new ones. This algorithm lead to maximum number of hit rate compared to other caching techniques.
FIFO: First in first out (FIFO) is a caching technique which is used similar to stack operation. The item in the first entry will be removed first when new one comes if the cache memory is filled. This algorithm gives the average number of hit rate.
Apart from these some of the intelligent web caching algorithms are Intelligent Client Side Web Caching Scheme based on Least Recently used algorithm and Neuro-fuzzy system by Ali &Shamsuddin (2009)],NNPCR by cob & Elaarag (2008) and many more algorithms. In case of Least Recently used algorithm the important factor called the recency factor is being neglected.

The following figure shows the exact loction of proxy cache and the proxy server.proxy cache is the one which replaces the web pages according to the requests by the users whereas the proxy server serves the web page that is requested by the user,immediately if it is present in the server else it is forwarded to the cache and accordingly the replacement is being done.

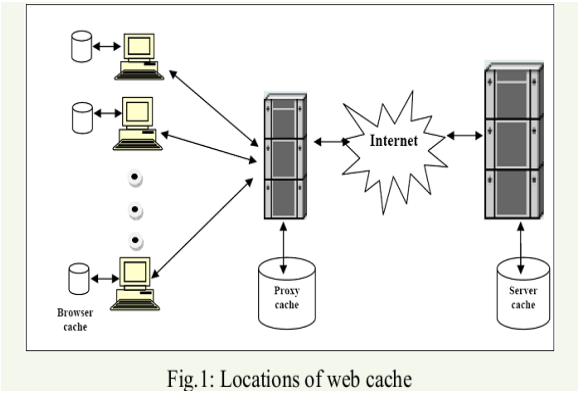

Fig.1: Locations of web cache

2.2. Performance Measures

There are standard metrics to analyze the efficiency and the performance of web caching techniques, they are Hit ratio(HR),Byte Hit Ratio(BHR) and Latency Saving Ratio (LSR) are the most widely used metrics used in evaluating the performance of web caching.

$$HR = \sum_{i=1}^{n} \delta i \, /N \qquad (1)$$

$$BHR = \sum_{i=1}^{n} bi\delta i / \sum_{i=1}^{n} bi \qquad (2)$$

$$LSR = \sum_{i=1}^{n} ti\delta i / \sum_{i=1}^{n} ti \qquad (3)$$

Where N is the total number of requests, t_i is the time taken for retrieval of object, and b_i is the size of the ith request.

2.3 Web Pre-Fetching

Web pre-fetching is a mechanism which is used for pre-fetching web pages into the cache before the actual request arrives. When combining web pre-fetching and web caching they can complement each other since the web caching technique make use of temporal locality, whereas web pre-fetching technique exploit the spatial locality of web objects. The problem is to determine which page has to be pre-fetched and cached. This problem is aggravated by the fact that there are wide spectrums of users and each has their own preferences. This work tries to solve the above problem by grouping the users based on their access patterns. Due to grouping of user's, we avoid the task of going into the individual preferences which will be quite a formidable task. The work presented here will integrate web caching and clustering based web pre-fetching scheme to improve the performance of proxy cache. Web pre-fetching scheme can be classified into two types: short term pre-fetching and long term pre-fetching schemes. A brief summary of the two approaches are as follows, they are short term pre fetching scheme and long term pre fetching scheme.

2.3.1. Short Term Pre-Fetching Scheme

The short term pre-fetching scheme pre-fetches web pages in the cache by analyzing the web cache's recent access history. Based on the analysis, the scheme computes the cluster of closely related web pages and pre-fetches clusters of web pages from the origin web servers (Chen, Qui, Chen, Nguyen & Katz, 2003).

2.3.2. Long Term Pre-Fetching Scheme

In Long term pre-fetching scheme, the popular web pages are identified by analyzing the global access pattern for the web page (Lee, An, & Kim, 2009) in this scheme, objects with high access frequencies and without longer update time intervals are more likely to be pre-fetched. Thus, web pre-fetching is a

pro active approach where the web pages are pre-fetched into the proxy server cache from the origin web servers.

2.3.3. Measuring the Performance of Pre-Fetching

To judge the success of a pre fetching system and to tune the parameters used, the performance of the system must be measured. The following criteria can be used to do this

- **Usefulness of Predictions/Pre-fetches**: the percentage of fetched pages that had already been predicted is pre-fetched.
- **Accuracy of Predictions/Pre**-fetches: The percentage of predicted or pre-fetched pages that were later actually requested by the user.
- **Practical accuracy of predictions**: the probability that one of the received predictions was correct.
- **Coverage**: the percentage of actual fetches which were preceded by the predictions.
- **Network traffic Increase**: The volume of network traffic with pre-fetching enabled / the volume of traffic without pre-fetching.

These are some of the measures used to detect the performance of pre-fetching technique. Most of the existing pre-fetching techniques will employ single object pre-fetching technique, which can be handled by traditional cache replacement algorithms. But when clustering based pre-fetching technique is used, in which multiple objects are pre-fetched, so which cannot be handled by traditional algorithms. So the proposed cache replacement algorithm will consider this issue by modifying cache replacement algorithm and provide better performance.

3. CLUSTERING BASED PRE-FETCHING TECHNIQUE

As we had seen the web caching and pre-fetching process now we move into the process of integrating the web caching and pre-fetching technique using the data mining based clustering pre-fetching technique. Specifically to move into the integration process let us see the clustering Process.

3.1. Clustering Technique

Clustering is a process of dividing the given data into group of objects. It is an unsupervised learning Technique of hidden data in data mining. Each group can be clustered together if the objects are similar among themselves and dissimilar to objects of other groups. Clustering will assign the points to finite system of K subsets which is called a cluster. From machine learning perspective clusters correspond to patterns and these patterns are hidden. A computational requirement which deals with large database is known as cluster analysis. In this paper the clustering based pre-fetching scheme is being used in which the Web navigational graph (WNG) is constructed for each user independently using a user's session time interval. At the end of each time interval new navigational graph is constructed for each user based on content of log files. Each node in the WNG represents a web object requested by each user and each edge represents user's transition from one web object to another and a weight is assigned to each edge which represents the number of transitions between those objects.

The clustering algorithm gets the content of WNG as input. Support and confidence are the parameters used to keep track of frequently visited pages by user. A threshold value is fixed for those parameters and those edges which have values less than this threshold is removed. Support is defined as the frequency of navigation between two nodes u1 and u2. The confidence is defined as freq (u1, u2)/pop (u1) where pop (u1) is the popularity of u1.Popularity of the node is defined as the number of incoming edges into that node. The WNG is partitioned in to sub graphs by removing those edges having low support and confidence values. The nodes in each connected sub-graph are identified as a cluster. When a user requests any one of the nodes (Web objects) in a cluster and if it is found in the long- term cache or in the short-term cache, all the remaining nodes in that cluster can be pre-fetched in to the short-term cache by predicting that those objects will be requested by the user in the near future [20]. This pre-fetching is done during the browser idle time and this pre-fetching helps in reducing the user perceived latency time. If the access count of a Web object is greater than the threshold value then that Web object is moved from the short term cache to long term cache. LFU technique is used for removal of pages from the short term cache if sufficient space is not available for caching a new Web object.

3.1.1. Preprocessing Step

A log file generated by the proxy server consists of the following fields,

- Machine IP address making the request
- Cache hit/ Cache miss and Response code
- Size of the requested object measured in bytes.
- URL of the requested page
- Information if the request is redirected to other server.

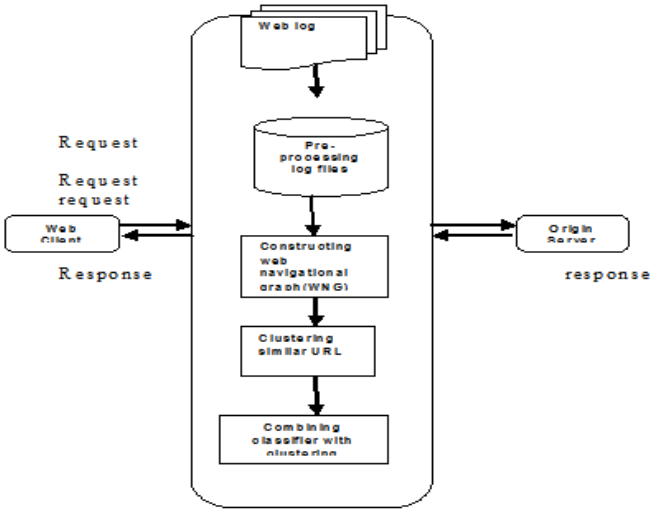

Fig.2. Architectural diagram of proposed work

3.1.2. Web Navigation Graph (WNG)

A weighted directed Web graph (x,y) is used to represent the requests of each user, where each node x represents a Web object and each edge y represents a user's transition from one Web object to another. The weight of each edge represents the number of transitions in the set. To make the size of the WNG manageable, the edges are removed whose connectivity between two Web objects is lower than a specified threshold. Support and confidence are the two parameters that determine the connectivity between two objects [1].

Let : $<xi,xj>$ be an edge from node xiR to node xjR. Support of G, denoted by $freq(xi,xj)$ is defined as the frequency of navigation steps between$xitoxj$R. Confidence of g is defined as $freq(xi,xj)/pop(xi)$, where $pop(xi)$ is the popularity ofxiR. By this definition, the support value of the edge $(x3,x4)$ for the user X in Figure 3 is$(x3,x4)$ = 1 and confidence value is $freq(x3,x4)/pop(x3)$ = 0.5. If the support threshold chosen is very less, too many less important user's

Transitions for clustering may be included and if the chosen threshold value is high, many interesting Transitions that occur at low levels of support may be missed.

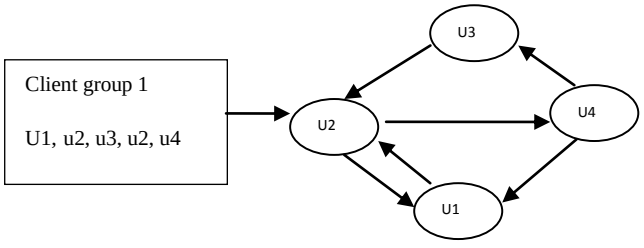

Fig.3. Web Navigational Graph (WNG)

3.1.3. Web Clustering Algorithm

The algorithm for clustering inter-site Web pages is described below [20]. A weighted directed Web graph (x,y) that represents the access patterns of a user is used. This graph is partitioned into sub graphs by filtering edges with low support and confidence values. The nodes in each connected sub graph in the remaining navigational graph will form a cluster. The inputs to this clustering algorithm will be Web navigational graph, the number of users, support threshold and confidence threshold. All the edges with support or confidence value less than the corresponding threshold values are removed. BFS (Breadth First Search) algorithm is applied to the navigational graph. BFS takes a node in the graph (called as source) and visits each node reachable from the source by traversing the edges. It outputs a sub-graph that consists of the nodes reachable from the source. This procedure is applied for all the nodes of the graph. All the nodes in each connected sub-graph forms a cluster. The time complexity of BFS is $(|x| + |y|)$ where $|x|$the number of nodes is and $|y|$ is the number of edges in the graph [1].

Input: G (u, v);
Unmark all nodes u;
Choose some starting node x;
Mark x;
List L = X;
Sub graph T= x;
While L nonempty;
{Choose some node y from front list;
Visit y ;}
For each unmarked neighbor w
{Mark w;
Add it to end of list;
Add edge yw to T ;}

Fig: 4 BFS Algorithm

3.2. Clustering Based Pre-Fetching

The following are the steps that take place in the proposed pre-fetching method

- A user requests a web object.
- Using the IP address, the proxy identifies the user and maps the user to a particular user group. Given that the Clusters of Web objects are known, the proxy searches inside the existing clusters of that user group to find in which cluster the requested object exists.
- All the remaining objects from the selected cluster are Pre-fetched from the origin server by the proxy and they are loaded into the short-term cache during the browser idle time.
- The proxy sends to the user his/her requested object.

The following describes in detail about the BFS algorithm with support and confidence value.

```
Input Parameters:
Confidence,support,
G(i)=navigation graph,
K=number of client groups:
Begin
For i=1 to K
{ CutWithConfidence(G(i),Confidence);
CutWithSupport(G(i),Support);
TraverseWithBFS(G(i));
}
End
```

```
Procedure
CutwithConfidence(G(i),Confidence)
{
For j=1 to num-of-edges
If (edge[j] <Confidence) remove edge[j];
}
Procedure CutwithSupport(G(i), Support)
{
For j=1 to num-of-edges
If (edge[j] <Support) remove edge[j];
}
Procedure Traverse With BFS (G(i))
{
int c1[i]=0;
Repeat
{C[c1[i]]=BFS(G(i));// C; a list array of traversed nodes
C1[i]++;
} until (all nodes of G(i) are traversed);
}
```

Fig.5 Clust Web Algorithm

4. PROPOSED WORK COMBINING CLUSTERING WITH SVM-LFU

4.1. Intelligent Web Proxy Caching Algorithms

Compared to traditional caching approaches, intelligent Web caching methods are more efficient.

Details about intelligent caching methods are found in [4] and that of conventional replacement policies are found in [23]**.** In our work, the proxy cache is divided in to short-term cache and long-term cache. The Web objects requested for the first time are loaded in to short-term cache. LFU algorithm is used for managing short-term cache. Those objects visited more than once from the short- term cache are moved to long-term cache. In case of BPNN and NNPCR caching technique, the recency factor which is considered as an important factor is being ignored in these techniques. Moreover the LRU caching technique when used ignores the frequency factor during the replacement of cache content. From the above studies it is observed that intelligent caching technique can be employed either individually or can be combined with LFU technique.

4.2. Support Vector Machine (SVM)

SVM (Support Vector Machine) is a technique useful for data classification. SVMs are supervised learning models associated with learning algorithms for analyzing data and recognizing patterns which are used for classification and regression analysis. It takes a set of input data and predicts to which one of the two possible classes, the given input data belongs to. Given a set of training datasets, each marked as belonging to one of two classes, a SVM training algorithm constructs a model that classifies new data inputs into one category or the other. SVM determine the hyper-plane to do binary division that is used to find the linear boundary between two classes such as Positive (Class1) and Negative (Class0) [5]. The hyper-plane is placed in between two classes and it is oriented like, the distance between the plane and data point in each class is maximized. The nearest data points are called as Support Vectors. In an arbitrary dimensional space a separating hyper plane can be written,

$$w.x + b=0$$

Where, b is bias, w the weights and$x$is the input vector.

The decision function can be written as,

$$D(x) = sign\ (w.x + b)$$

If the sign of the decision function is positive the object is classified as Class 1 (Positive Class), otherwise the object is classified as Class 0 (Negative Class)

$$w.x + b=1\ (Class\ 1)$$

$$w.x + b=-1\ (Class\ 0)$$

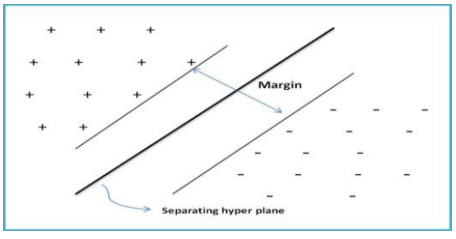

Fig.6 SVM

SVM classifier is used to classify the Web objects as class 0 or class1, while moving them from short term cache to long term cache. When a user requests for a Web object, simultaneous search is made in short term cache as well as in the long term

cache. If the requested Web object is available in the short-term cache, its access count is increased by one. If the access count is greater than the threshold value then that web object is given as input to SVM classifier for classification. If it is classified as class 0, it is moved to the bottom of long-term cache. If it is classified as class 1, it is moved to the top of long-term cache. If there is no space in the long-term cache, the Web object(s) placed at the bottom of the long-term cache are removed to provide space for this new Web object. Meanwhile a copy of this web object is given to the requested user. If that Web object is found in the long-term cache, classification of that Web object is done again. If it is re-classified as class 1, it is moved to the top of the cache (long-term cache) else it is moved to the bottom of that cache. It is then given to the requested user and pre-fetching of other web objects if any belonging to that cluster in to the short-term cache is initiated If cache miss occurs, the requested object is searched in all the clusters generated by Clustering algorithm for that user. If it is found in any one of those clusters, that Web object is fetched from the original server and it is sent to the user as well as placed in to the short term cache. Other Web objects that are present in that cluster will be pre-fetched during browser idle time and will be placed in the short-term cache. If the requested object is not found in any of the clusters of that user, then the required Web page is fetched from the original server. A copy of it is placed in the short term cache and it is sent to the requested user. Thus by combining Web caching and pre-fetching it is possible to increase the hit ratio, decrease user perceived latency and reduce the origin server load.

5. PERFORMANCE EVALUATION

5.1. Dataset

The scheme explained in this paper is tested with a dataset. The dataset is obtained from a proxy server installation www.ircache.net. The raw proxy server log file for the data set contained the details of more than 1 million requests. The log cleaning process was applied on the dataset, and moreover the dataset contained about 610,634 entries fir for analysis. 70% of all the requests have been used for the user's access pattern analysis, creating training dataset and testing.[1].The remaining 30% will be used for the testing the scheme.

5.2. Performance Metrics

- **Hit Ratio(HR)**
  HR is the percentage of the total number of requests served by the cache over the total number of requests given.

- **Byte Hit Ratio(BHR)**
  BHR is the percentage of the number of bytes corresponding to the requests served by the cache over the total number of bytes requested.

Generally HR and BHR are calculated using different values of support and confidence using SVM and LFU pre-fetching. The performances for both of these methods for various cache sizes are given in the table below. Moreover increase in the support and confidence value leads to increase in number of web objects in the created clusters. users experience decreased access latency while using SVM pre-fetching because of increase in percentage of cache hits. this ensures lesser load on the original server.

6. CONCLUSIONS AND FUTURE WORK

In this paper the web access is improved by preprocessing the log file and constructing web navigational graph. Moreover clustering of Web objects in the WNG is done using a clustering algorithm. Frequency of frequently used Web objects is monitored by Support and Confidence values. The pre-fetching of requested web objects is done during the idle time either from short term cache or long term cache. Using access count a Web object in short-term cache is moved to long-term cache after classification using SVM algorithm. A cache miss in both the caches leads to fetching of that object from the original server and a copy of it being placed in the short-term cache. The SVM-LFU algorithm is compared with that of clustering based pre-fetching technique and the performance would be shown to better in SVM than clustering based pre-fetching.

This work can be further enhanced with the content distribution network to improve the performance of the server overall using some of the web caching and pre-fetching technique integrated.

REFERENCES

[1] G.Pallis, A.Vakali, and J.Pokorny,"A Clustering-based Pre-fetching Scheme on a Web Cache Environment". Computers and Electrical engineering, 34(4),(2008).pp.309-323.

[2] Ali W, Shamsuddin S M, Ismail A S, "Intelligent Web Proxy Caching Approaches Based on Machine Learning Techniques". *Decision Support Systems 53 (2012)* 565-579.

[3] W.Ali, S.M.Shamsuddin, A.s.Ismail,"A Survey of Web Caching and Pre fetching". International journal of advances in soft computing and its application (2011).

[4] S.Podlipnig, L.Boszomenyi,"A Survey of Web Cache Replacement Strategies",CM computing surveys 35(2003) 374-398.

[5] C.W.Hsu, C.C.Chang, C.J.Lin,"A Practical Guide to Support Vector Classification". A online guide for using LIBSVM tools, 2009.

[6] Pallis.G,Vakali A."Insight and Perspectives for Content Delivery Networks". Communication of ACM (CACM) 2006; 49(1):101-6.

[7] Ten, W, Chang,CY,& Chen,MS.(2005)."Integrating Web Caching and Web Pre fetching in Client Side

Proxies". Parallel and Distributed systems,IEEE Transactions on 16(5),444-455.

[8] R.Sudhakarapandian,"Modified ART1 Neural Networks for cell formation using production data".Key Bridge Marriott,Washington DC,USA August 23-26,2008.

[9] Akshay Shenoy,"Improving the Performance of a Proxy Server using Web Log Mining".San Jose State University,4-1-2011.

[10] Ali W, Shamsuddin S M, Ismail A S, "Web Proxy Cache Content Classification Based on Support Vector Machine", *Journal of Artificial Intelligence* 4 (2011) 100–109..

[11] Chen X, Zhang X. "Popularity-based PPM: an effective Web Prefetching technique for high accuracy and low storage." *In: Proceedings of the international conference on parallel processing. Canada, Vancouver; 2002.*

[12] Chen Y, Qiu L, Chen W, Nguyen L, Katz R H. "Efficient and Adaptive Web Replication using Content Clustering." *IEEE J Selected Areas Communication 2003*; 21(6):979–94.

[13] Pallis G,Angelis L,Vakali A."Validation and Interpretation of Web Users' Session Clusters."Information Processing and Management 2007; 43(5):1348-67.

[14] Web reference: http://www.wikipedia.com/svm..

[15] Ali W,Shamsuddin S M "Intelligent Client-Side Web caching Scheme based on least recently used algorithm and neuro-fuzzy system".in:W.Yu,H.He,N.Zhang (Eds.),Advances in Neural Networks-ISSN 2009,Springer,Berlin/Heidelberg,2009,pp.70-79..

[16] Yang Q,Zhang H."Integrating Web Pre-fetching and caching using prediction models".World wide web 2001;4(4):299-321.

[17] Chen T,"Obtaining the optimal cache document replacement policy for the clustering system of an EC website",*European Journal of Operational Research* 181(2007)828-841.

[18] cherkasova L,"Improving WWW Proxies Performance with greedy-Dual-Size-Frequency