ISSN 2347-4289

Review on Web Prefetching Techniques

Suvarna Temgire, Poonam Gupta

Computer Engineering Department, Raisoni College of Engineering, Pune, India;
Computer Engineering Department, Raisoni College of Engineering, Pune, India
Email: stemgire@gmail.com

ABSTRACT: Web prefetching is an important aspect to find the possibility of finding which object would be requested in near future. Demand of internet and easy accessibility of information, communication and flexibility had put gigantic pressures on the principal infrastructure of WWW. The World Wide Web is an immensely scattered and provides access to shared data with ease. Due to this there is a huge pressure on server with respect to information load, resulting in the compromise of service at the end user. Further, the load imbalances between the servers that arise from the severe nature of irregular web access essentially reflecting the underlying predictable and often unpredictable human nature is a concern from the resource deployment point of view. Besides, the media type adds further limitations to the whole process. But the cache management has many challenges in balancing the process of meeting the demands of the users on the one hand and ensuring optimal utilization of system resources on the other hand. Caching and pre-fetching is middle-aged technology widely used in many areas such as Database Systems and Operating Systems.

Keywords: Internet measurement, World Wide Web, Traffic analysis, Web Prefetching

1 INTRODUCTION

THE rapid growth in popularity of the WWW is leading to numbers of performance problems. The Internet is becoming increasingly congested and popular. Web sites are suffering from overload conditions due to the large number of concurrent accesses. As a result, considerable latency is often experienced in retrieving web objects from the Internet. Web prefetching mechanisms are beneficial to web users by hiding the download latencies. However, there is no comparison of web prefetching techniques that consider the potential seeming by the user as the key factor. This lack of comparison is due to difficulty to reproduce large amount of activities taking part in web pre fetching process. The motive is to build an effective prediction model which has been classified into two groupings as per data structure, which is PPM model and DG models. However, both models have their own shortcoming. To overcome their weaknesses, an improved algorithm is proposed based on exponential downward double dependency graph algorithm. Web prefetching method uses the spatial locality of Web matters evaluating the prefetching structures.With the advances of the Internet technologies, the number of Web sites and pages have increased rapidly in recent years. There are various estimates about the numbers of indexed Web sites, ranging from 100 to 500 million, and the number of indexed Web pages of over 20 billion. With such a huge collection, a quick Web search and response to user's query is important for effective utilization of the Web. A so called "eight second rule", one of the rules of thumb of Web latency metrics, states that most Internet users will get inpatient and are likely to abandon a site if the response time for a page exceeds eight seconds [12]. Reducing Web latency is particularly important for online businesses not to lose customers. Various approaches have been developed in improving the efficiency of Web servers, including improved hardware (speed, bandwidth) and software solutions (more suitable models and protocols, better algorithms). A commonly used and effective technique is prefetching that preloads some data to the local cache before it is actually requested anticipating that these data are to be requested by the user in the near future so that they will be readily available locally rather than retrieved from remote sites. Of course, the preloading process is to retrieve from remote sources, but it can be done without perceived delay from the user's point of view, simply because there is always a time gap between consecutive requests from the same user in the Web environment and the Web server can use this time gap to Pre-fetch the predicted pages. Successful prefetching will not only reduce the delays for users' requests for Web objects, but also result in less overall network traffic and lighter loads on the Web servers. The idea of prefetching can be traced back many years. It is traditionally used in operating systems for managing virtual memory to reduce disk accesses. Applying prefetching for latency reduction in Web object retrieval takes the issue to a whole different level. The Web is huge and dynamic and the client server environment is much more complex. Hence, more sophisticated models and algorithms are needed for Web prefetching that share the same goals: improving the response time by reducing the network traffic. In the literature, many Web prefetching techniques have been proposed and developed, each of which has its advantages and weaknesses. Different performance metrics have also been applied to measure the goodness of the techniques [13].And Pages has to be cautiously chosen in order to be prefetched [15].

2 MOTIVATION OF WORK

The contribution of understanding the pattern and behavior of internet is beneficial to predict and prefetch web objects in caching. These web objects stand for new comer requests is used to decrease response latency and increase efficiency of source management. Integrating web caching and prefetching to provide the proxy with opulent information, a Web server may purposely send all possible prefetching clues with various levels of assurances to the proxy. Without any control, a proxy will prefetch every implied object into its cache, despite that the confidences of some prefetching rules may be low. In this case, a significant portion of the cache content will be replaced because a proxy may concurrently serve a large amount of client requests and each of these requests may trigger certain prefetching rules. As a result, the state of the cache content will become insecure and the cache hit ratio will drop sharply. On the contrary, if the prefetching control is over authoritarian, a proxy will tend to discard some beneficial hints provided by the Web server, thus shaping down the advantage of Web prefetching. In view of this, the inspiration for our study is to design an innovative cache replacement algorithm, not only considering the caching effect in the Web atmosphere, But by calculating the prefetching techniques provided by prefetching structure. The contests of devising such a cache replacement algorithm are mainly due to the following sensations. First, the caching parameters of the implied object (i.e., the size, fetching cost, reference rate, invalidation cost, and invalidation frequency of the implied object) can affect the strength of the matching prefetching rule. Several algorithms based on Markov models and Web mining techniques are proposed in [1], [2], [3], [4], [5], [6], to develop prefetching rules from the server's access log. Therefore, when there be present spatial localities of user access patterns, the rules discovered at proxy servers are able to reflect the local user conduct more precisely than those discovered at Web servers. Basic variable will use in Prefetching:

Predictions: amount of objects predicted by the prediction engine.

Prefetchs: amount of objects prefetched by prediction engine.

Good Predictions: Amount of objects predicted that are subsequently demanded by the user.

Bad predictions: Those predictions that do not result in good predictions.

Prefetch hits: Amount of prefetched objects that are subsequently demanded by the user.

Objects not used: Amount of prefetched objects never demanded by the user.

User Requests: Amount of objects is user demand. Figure 1 shows realistically the relations between the variables defined above. A represents an object requested by the user is neither predicted nor prefetched. B represents a Good Prediction that has not been fetched, while C presents a prefetch hit, i.e. a Good prediction that has been prefetched. D is an Object not used, which is also a bad prediction. Finally E represents a bad Prediction. Analogously, define byte associated variables (Predictions, Prefetchs and so on) by replacing the objects with the corresponding size in bytes in their definition. Pre-fetching caching scheme constantly produces better performance than the one that does not have prefetching.

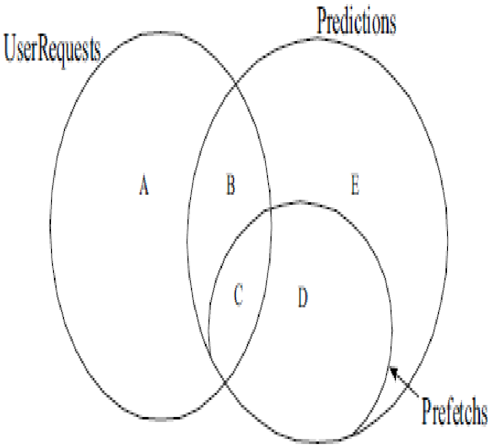

Fig 1. Sets representing the relation between user Requests, Prediction and Prefetchs

The benefit is more when cache size is smaller. At the end, the number of pages pre-fetched at each time is inspected. It is not true that more pages are pre-fetched, the better system performance. Therefore, it is important to carefully choose the number of pages that should be prefetched.

2.1 Web prefetching:

There are several ways of improving network traffic with the help of web caching. . That means in specific, it can decrease the bandwidth consumption, the network latency used by the client and the server load. Web caching considers several issues, which reduce risks, its ability and effectiveness, such as uniformity, dynamic objects and several security and legal issues. Alternatively, there were some researches that use mining techniques to understand the browsing behaviors of Internet users [7, 8]. Several studied considered prefetching popular documents in order to reduce perceivable network latency [9, 10, 11, 12, 13]. Some paper discussed techniques of an integrate model of prefetching and caching in a file system [9, 10]. This research paper uses web-log mining for judging the prediction and Prefetch web matters into web cache server. This method plays a vital role to extract useful knowledge and

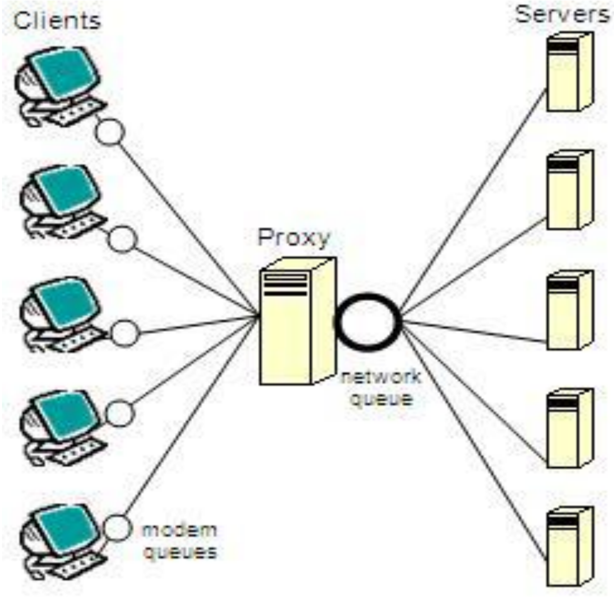

Fig 2. Proxy server implementation

if it found that the requested object is already stored in its cache, returns the object to the user. Otherwise, it goes to the original server instead of the user, grabs the object, stores it in its cache, and returns the object to the user. An advantage of Web proxy caching and pre-fetching is that all clients within the Local Area Network (LAN) can share objects stored in the cache. In order to compute the benefits of pre-fetching, the common currency that will be used to measure the benefit must be selected. Since the ultimate goal of this study is to mask document latency, latency is selected as the currency and to effort to minimize it.

2.2 Pre-fetching and correlation

Pre-fetching is the operation of fetching information from remote Web servers even previously it is requested. Objects such as images and hyperlink pages that are cited in a Web page (say a HTML page) may be fetched well before they are really called for. It should be noted that a tradeoffs occurs. Web objects are pre-fetched assuming they will be requested quickly. An accurate selection of such objects would lead to a decrease in the access latency, whereas inaccurate picks would only result in wasted bandwidth. Correlation between Web objects can be obtained by studying the link structures of Websites. Moreover, Web pages on related topics are often accessed together because of users' particular interests. The prediction algorithm in this study is based on the prediction model described by Padmanabhan and Mogul [9]. A remarkable difference is that our prediction model is time-based, which means the prediction window is a specific time period instead of a number of requests. In order to present the links a correlation database is used to store information of pre-fetch information such as size, links, and date access and protocols type. The database is divided into two tables, which are transaction and category as shown in Figure 3. The transaction table stores all links and category table is differentiating a type of transaction involved. This information can be analyzed further in a data warehouse. The data warehouse currently is implemented in Analysis Service SQL Server 2005. Web Cache management framework to utilized resource services based on prediction based model. By integrating weight value for predicting future requests, it is possible to improve both hit rate and byte hit rate while reducing the response latency. The varieties techniques would be considered and apply algorithm analysis to improve existing scheme with mechanism of self- discovery. Web cache management policies and impact used in the market will be follows and extend to dynamic prefetching strategy on application development.

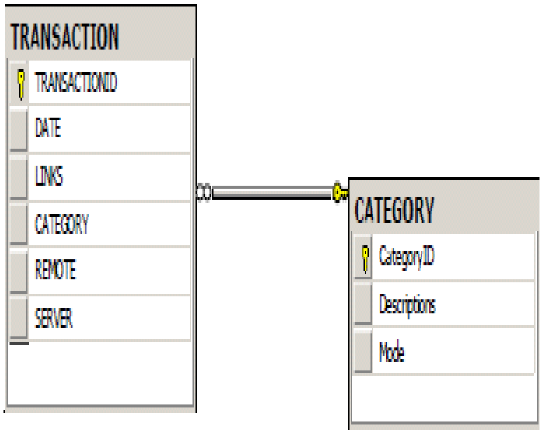

Fig 3. Database design

The helpfulness of pre-fetching and caching algorithm depends and may vary on the usage and the type of search that is available. For example, Google has implemented the most frequent user search. By the time user finishes typing the word they are looking for, a list of all the hits is listed. This algorithm can be well implemented in a social network Website such as Facebook, Twitter, MySpace and etc. Users can use many applications in the social network Website, where sometime they may lose their way using it. By implementing this algorithm, social network Website users can benefit these criteria other than implementing it in a proxy server alone. The implementation of the mechanisms as seen can be widely used to refine and improve the access to many million Websites in the world as a proxy server.

3 CATEGORIES OF PREFETCHING APPROACHES

Web prefetching has been extensively studied. Roughly speaking, major approaches fall in the following three categories: probability based, clustering based and using weight-functions.

3.1 Probability Based Prefetching

In paper [14] Web object prefetching approaches according to Probability, The central problem for Web prefetching is the prediction algorithm. When a request comes, a decision needs to be made on which page would mostly likely to be requested next time. Probability based prediction is a natural approach. Probabilities are calculated using the history access data. This method assumes that the request sequence follows a pattern (is not random) and the probabilities are trying to follow this pattern. One of the advantages for this approach is the number of pages prefetched can be controlled. In [16], some data structure need for record probability for this purpose tree structure is use. Another probability based method is the multidimensional matrix scheme proposed in [18]. Instead of using the tree structure, [18] presents a matrix structure to store sequences of probabilities. The advantages of this model include parallel and sequential search in prediction. The algorithm does not need training data and entries of the multi-dimensional matrices are dynamically updated at run time. It is shown that the latency reduction can be dramatically increased when the search path and the number of prefetched pages are selected at the optimal.

3.2 Clustering Based Prefetching

In [14], Web object prefetching approaches according to Clustering based prefetching methods make decisions using the information about the clusters containing pages that have been previous fetched, anticipating that pages that are "close" to those previously fetched pages are more likely to be requested in the near future. Support vector machine (SVM) is a data mining based classification algorithm. The method is to use hyperplanes to separate data in different classes. This idea is adopted in [19] to develop a SVM-based online learning algorithm to deal with the web prediction problem. This online learning algorithm is based on incremental chunk for LS-SVM (Least Square Support Vector Machines) classifier. The training of the LSSVM can be placed in a way of incremental chunk avoiding large scale matrix inverse but maintaining the precision when training and testing data. The online algorithm is especially useful for the large data set and practical applications where the data come in sequentially. The clustering based prefetching presented in [20] effectively integrates caching and prefetching. Specifically, Web caching and prefetching can complement each other since the first one exploits the temporal locality whereas the second one utilizes the spatial locality of the Web objects. In [20], the clustWeb scheme for clustering inter-site Web pages is introduced. The clusters from the algorithm are obtained by partitioning the Web navigational graph using association rule mining techniques and the connectivity among Web pages in the graph. With this clustering scheme (called clustPref), each time a user requests an object, the proxy fetches all the objects which are in the same cluster with the requested object. The simulation results indicated that the graph-based clustering approach significantly improved network performance with higher byte hit rate (BHR) than other methods. Assisted by data mining techniques, the Prefetching Candidate Mining (PCM) is presented in [21]. PCM processes user requests; repeated request sequences are recorded in first-order Markov model. To limit the amount of memory used, PCM ranks stored request sequences according to their likely to yield useful predictions. As a result they obtain a prefetching algorithm that can use a pre-allocated amount of memory. Furthermore the amount of memory available to the algorithm can be modified at run time. This prefetching algorithm is suitable for those devices that have low memory, such as PDA and smart phone. In [22], a neural network model is applied to classify user groups. The idea is to use the ART1 based clustering algorithm to group users according to their Web access patterns. The advantage of using the ART1 is that it adapts to the change in users Web access patterns over time without losing information about their previous Web access history. With this approach, each cluster of users is represented by a prototype vector that is a generalized representation of URLs frequently accessed by all the members of that cluster. The degree of similarity between the members of each cluster can be adjusted by changing the value of a vigilance parameter.

3.3 Weight-Function Based Prefetching

In [14] although the models presented in Probability Based Prefetching and Clustering Based Prefetching has been shown to be efficient, they only consider the request patterns and mainly the request probabilities. The cost of the network traffic and server workload as the overhead of the programs was not considered. To consider factors other than just the probabilities (such as size, priority), a function that involves multiple factors is needed. Several approaches in this direction have been proposed include the following:

(1) Web page size consideration
(2) Prediction by partial match model (PPM)
(3) Mining Web logs

In [15], the prefetching function involves size of the web page. Bigger size pages have less chance for prefetching and smaller pages have more advantages.The idea is to consider the cost of the prefetching algorithm. In case of failure, the algorithm would not increase too much on the network traffic. The prediction by partial match model (PPM) proposed in [17] attempts to capture the changing user request patterns and fit the memory. The structure is based on a noncompact suffix tree. It incrementally inserts the newest user request and deletes the oldest one. A sliding window is used to control the number of user requests. The concept of entropy is introduced and a maximum entropy principle is used to model the outgoing probability distribution of every node in the tree. Their approach predicts the user's next request based on the nodes with low entropy. The prediction function combines lower entropy, larger prediction accuracy rate and the longest match rule to predict the user's next request.

4 WEB PERFORMANCE INDEXES TAXONOMY

To the better understanding of the meaning of those indexes, it classify them into three main categories (see Figure 4), attending to the system feature they evaluate:

Category 1: prediction related indexes.
Category 2: resource usage indexes.
Category 3: end-to-end perceived latencies indexes.

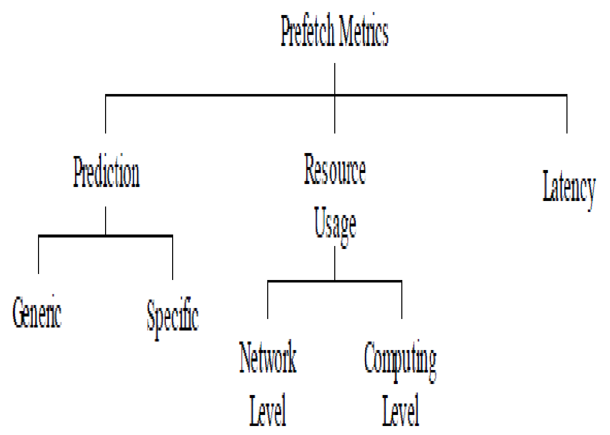

Fig 4. Prefetching metrics taxonomy

The first category is the main one when comparing prediction algorithms performance and includes those indexes which quantify both the efficiency and the efficacy of the algorithm (e.g., precision). The second category quantifies the additional cost that prefetching incurs (e.g., traffic increase or processor time). This cost may become really high; thus, it must be taken into account when comparing prefetching techniques, thus those indexes can be seen as complementary measures. Finally, the third category summarizes the performance achieved by the system from the user point of view. Prefetching techniques must take care of the cost increase because they can negatively impact on the overall system performance (traffic increase, user perceived latencies). Therefore, the three categories are closely related since, in order to achieve a good overall performance (category 3), Prefetching systems must trade off the aggressiveness of the algorithm (category 1) and the cost increase due to prefetching (category 2). Different definitions for the same index can be found in the literature (e.g., precision) and this fact increases the heterogeneity of the research efforts. Here, the definition is considered more precise and appropriate for evaluation purposes. In the cases where several names match the same definition, we can select the most appropriate index name our point of view. The goal of this section is not only to help the understanding of the indexes but also to discuss their usefulness, distinguishing those used for comparison purposes in any prefetching systems from those applicable to a particular prefetching technique (i.e. specific). Specific indexes are only found in the Category 1. Performance Indexes: One of the most important steps in a performance evaluation Study is the correct choice of the performance indexes. The algorithms performance has been evaluated by using the main metrics related to the user's perceived performance, prefetching costs and prediction performance. Latency per page ratio: The latency per page ratio is the ratio of the latency that prefetching achieves to the latency with no prefetching. The latency per page is calculated by comparing the time between the browser initiation of an HTML page GET and the browser reception of the last byte of the last embedded image or object for that page. This metric represents the benefit perceived by the user, which will Increase (Tr): The bytes transferred through the network when prefetching is employed divided by the bytes transferred in the non-prefetching case. This metric includes both the extra bytes wasted by prefetched objects that the user will never use, and the network overhead caused by the transference of the prefetch hints. The variant Object Increase treasures this cost in amount of objects. Both indexes evaluate the costs that prefetching incurs to achieve the benefits. They are better as lower their value is. The ratio of requested objects by the user those were previously prefetched. This metric is the prediction index that better explains the latency per page ratio, i.e., this is the bene_t of the prefetching from the prediction point of view. It ranges from 0 to 1, being 1 its best value. The variant Byte The percentage of the bytes requested previously prefetched. Precision (Pc): The ratio of prefetch hits to the total number of objects prefetched. It ranges from 0 to 1, being 1 its best value.

CONCLUTION

In this paper, we review and categorize different Web prefeching models and parameter. These papers gave me fruitful information on how to preload data, different model techniques for prefetching in Web Domain. After that, we learnt which object are frequently used and how it predicts that object , which object to be prefetched, which are removed from cache according to prefetching techniques and also category and measurement of prefetching.

REFERENCES

[1] M. Deshpande and G. Karypis, “Selective Markov Models for Predicting Web-Page Accesses,” Proc. First SIAM Int’l Conf. Data Mining, 2001.

[2] B. Lan, S. Bressan, B.C. Ooi, and K. Tan, “Rule-Assisted Prefetching in Web Server Caching,” Proc. 2000 ACM Int’l Conf. Information and Knowledge Management, 2000.

[3] A. Nanopoulos, D. Katsaros, and Y. Manolopoulos, “Effective Prediction of Web-User Accesses: A Data Mining Approach,” Proc. Workshop Web Usage Analysis and User Profiling (WebKDD), 2001.

[4] V. Padmanabhan and J.C. Mogul, “Using Predictive Prefetching to Improve World Wide Web Latency,” ACM SIGCOMM Computer Comm. Rev., vol. 26, no. 3, 1996.

[5] J. Pitkow and P. Pirolli, “Mining Longest Repeating Subsequence to Predict World Wide Web Surfing,” Proc. Second USENIX Symp. Internet Technologies and Systems, 1999.

[6] Q. Yang, H.H. Zhang, and I.T. Li, “Mining Web Logs for P. Cao and S. Irani, Cost-aware www proxy caching algorithms, In USENIX Systems, Monterey, CA, Dec. 1997.

[7] Z. Su, Q. Yang, Y. Lu and H. Zhang, What next: A prediction system for web requests using n-gram

sequence models. In Proceedings of the First International Conference on Web Information System and Engineering Conference, Hong kong June 2000, pp. 200-207.

[8] M. Gerry and H. Haddad, Evaluation of web usage mining approaches for user's next request prediction, in: Proceedings of the 5th ACM International Workshop on Web Information and Data Management, New Orleans, Louisiana, USA 2003, pp. 74-81.

[9] W. Wu and H. Lu, Efficient prediction of web accesses on a proxy server, in: Proceedings of the 11th ACM International Conference on Information and Knowledge Management, 2002, pp.169-176.

[10] Q. Yang, H. H. Zhang and T. Li, Mining web logs for V. Padmanabhan and J. Mogul. Using predictive prefetching to improving www caching, The Seventeenth International Conference on very large Database, Sept. 1991, pp.255-264.

[11] Yin-Fu Huang and Jhao-Min Hsu, Mining web logs to improve hit ratios of prefetching and caching, Knowledge-Based Systems, Science Direct, 2006.

[12] Peter M. Broadwell. Response time as a per formability metric for online services. Technical Report CB/CSD- 04-1324, University of California at Berkeley, Berkeley, California, 2004.

[13] Josep Dom`enech, Jos´e A. Gil, Julio Sahuquillo, and Ana Pont. Web prefetching performance metrics: A survey. Performance Evaluation, 63(9-10):988C1004, 2006.

[14] Toufiq Hossain Kazi, "Web Object Prefetching: Approaches and a New Algorithms", ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel / Distributed Computing., PP.115-120, 11th 2010.

[15] Wenying Feng, "Machine Learning Prediction and Web Access Model", 31st Annual International Computer Software and Applications Conference(COMPSAC 2007), 2007.

[16] Wenying Feng, Shushuang Man, and Gongzhu Hu. Markov tree prediction on Web cache prefetching. In Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, volume 208 of Studies in Computational Intelligence, pages 105–120. Springer, Daegu, South Korea, May 26-28, 2009.

[17] Zhijie Ban, Zhimin Gu, and Yu Jin. An online ppm prediction model for web prefetching. In Proceedings of the 9th annual ACM international workshop on Web information and data management, pages 89–96. ACM,2007.

[18] Wenying Feng and Karan Vij. Web cache prefetching by multi-dimensional matrix. In Proceedings of 2008 Advanced Software Engineering and Its Applications, pages 265–270, 2008.

[19] Zhili Zhang, Changgeng Guo, Shu Yu, De Yu Qi, and Songqian Long. Web prediction using online support vector machine. In 17th IEEE International Conference on Tools with Artificial Intelligence, pages 451–456.IEEE Computer Society, 2005.

[20] George Pallis, Athena Vakali, and Jaroslav Pokorny. A clustering-based prefetching scheme on a Web cache environment. Computers and Electrical Engineering, 34(4):309–323, 2008.

[21] Qinghui Liu and Roberto Solis-Oba. Web prefetching with machine learning algorithms. In International Conference on Internet Computing, pages 142–148, 2008.

[22] Santosh K. Rangarajan, Vir V. Phoha, Kiran Balagani, Rastko Selmic, and Sitharama S. Iyengar. Adaptive neural network clustering of Web users. IEEE Computer, pages 34–40, 2004.