

Rough Neuro-PSO Web Caching and XML Prefetching for Accessing Facebook from Mobile Environment

Sarina Sulaiman¹, Siti Mariyam Shamsuddin¹, Ajith Abraham²

¹*Soft Computing Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia*

²*Machine Intelligence Research Labs (MIR Labs)
Scientific Network for Innovation and Research Excellence
P.O. Box 2259, Washington 98071-2259, USA
sarina@utm.my, mariyam@utm.my, ajith.abraham@ieee.org.*

Abstract

Prefetching and Web caching have been known as techniques to increase the speed of Web loading process. Previous studies have been conducted to infuse artificial intelligence such as Artificial Neural Network (ANN) into Web caching. In this paper, we propose a new hybrid technique based on combination of ANN and Particle Swarm Optimization (PSO) for classification Web object either to cache or not and generate rules from log data by using Rough Set technique on proxy server (Rough Neuro-PSO). It is needed because mobile context has limited resources like speed and memory. Our method is by using XML file for prefetching which is saved into mobile memory. Prefetching that used xml file is much faster to be searched or accessed. In Web caching side, we enhance the speed by using Rough Neuro-PSO in order to choose the log. At the end of this paper, we present a new framework that is believed to speed up the access of Web page in mobile environment context.

1. Introduction

Prefetching is a technique used to reduce bandwidth load and user-perceived latency. Furthermore, Prefetching can also be used for traffic shaping and idle-time processing [1,2]. There are lots of elements contributed in reducing Web speed such as: heterogeneous network connectivity and the distance between server and client. Web caching has been introduced as a technique that is able to trim down network and server load. By injecting prefetching Web into prefetching local caches, it can act as agent to

decrease the latency [3,4].

In this paper we will describe the potential works on Web caching and prefetching which have yet to be handled by previous researchers. This paper is divided into 7 sections. Section 2 presents the previous and current work on Web caching and prefetching. Section 3 critical analysis on prefetching is presented. Section 4 will give some description on Web caching. Section 5, will talk on Web caching and prefetching on mobile environment. Section 6, will present a new framework that become our main contribution. Finally, the discussion and conclusion are explained in section 7.

2. Related works

Lots of researchers have begun to search and observe the way of making the interaction between mobile device and server much easier. Harding et al. in [5] proposed a novel architecture for communication between mobile device and server by adding a feature for preparation in mobile side. The interaction on mobile device is very rich and almost similar to desktop application. This feature acts as preparation tools for users. Therefore, the users only communicate with server after they prepared the data. The data that will be transferred over to a mobile device and a server and bandwidth load will also decrease. It will also increase the speed over mobile and server.

Many researchers have looked for ways to improve current caching techniques. Padmanabhan and Mogul [1] proposed predictive model to be used as server hint. The proposed model is equipped with server that is able to create Markov model by predicting probability of object A will be tag along, with next n requests and

object B (n is a parameter of the algorithm). The server will use the model to produce a forecast for subsequent references and throw the forecast to the client. The client will use forecast result to prefetch an object on server if only that object is not in the client cache. The simulation has been done is able to reduce the latency until 45%. However, their technique has weakness because it also makes network traffic larger by two times [1]. That is why a lot of people try to find latency and network reduction at the same time.

Bestavros et al. [6] have presented a model for the speculative dissemination of World Wide Web data. His work illustrates that reference patterns from a Web server can be used as a key information source for prefetching. They also investigate that latency reduction increases until 50%, though it still increases the bandwidth utilization. See the Figure 1.

Type	NONE	HTML	GIF	DATA	CLASS	JPEG	MPEG	OTHER
New latency	1.3	2.4	1.6	0.3	1.6	3.9	110.8	12.9
Original latency	2.5	4.3	3.3	1.5	3.8	7.2	151.7	17.0
Percentage by count	29.5%	12.1%	42.4%	0.2%	0.3%	11.1%	0.0%	5.4%
Percentage by time	17.8%	12.5%	34.3%	0.1%	0.3%	19.4%	0.4%	15.2%
Passive caching	26.7%	21.8%	34.5%	73.0%	40.4%	27.1%	4.5%	8.8%
Local prefetching	48.3%	43.7%	52.7%	79.9%	58.2%	45.8%	26.9%	24.3%
Server-hints without caching	49.3%	61.6%	62.5%	66.8%	64.4%	62.7%	27.5%	27.9%
Server-hints with caching	51.1%	61.5%	60.1%	79.9%	65.0%	58.4%	26.4%	27.6%

Figure 1. Latency reduction for prefetching and caching by type of data requested (times in seconds) [6]

On the other hand, Pallis et al. in [7] has given some solution on prefetching by implementing clustering method. Web prefetching is an interesting technique that can be used to reduce network resource demand. Furthermore, the combination of Web prefetching and caching will give a momentous enhancement of Web infrastructure.

Pallis et al. in [7] proposed a different technique by creating cluster for prefetching into graph-based clustering which consists of lots of directed clusters of “correlated”. This technique can be adapted into a Web proxy server and is able to rise up the performance. According to the simulation that has been conducted by real data set, the recommended framework is believed to be able to increase performance of the Web caching environment efficiently.

Kroeger [3] observed a local proxy caching which is able to decrease latency until 26%, while prefetching could decrease latency at 57%. The combination of both of them will give better latency reduction until 60%. Furthermore, He also describe found that algorithm on prefetching has contribution on reducing

latency. From his work, it also explained that prefetching can provide double improvement on caching. However, it is only on decreasing the latency, see Figure 2.

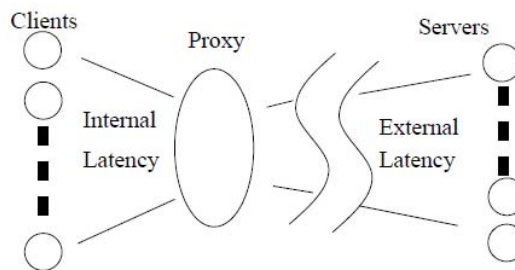


Figure 2. Proxy Configuration [3]

3. Prefetching

Web prefetching is used as a tool for predicting user request by grouping the favorite request into cache. It is different with Web data caching that only utilize temporal locality. Temporal locality is referring to repeated users’ accesses for same object during short time periods. Hence, the bottlenecks and traffic jams on the Web can be solved and object transfer much faster.

3.1. Prefetching approach

The Web prefetching approaches can be characterized according to its short-term and long-term benefits. Pallis et.al in [7,9] classified the existing prefetching policies as follows:

Short-term prefetching policies: Future requests are predicted to the cache’s recent access history. Based on these predictions, clusters of Web objects are prefetched. In this scheme, the short-term prefetching schemes use Dependency Graph (DG) where the patterns of accesses are held by a graph and Prediction by Partial Matching (PPM) where a scheme is used and adopted from the text compression domain. Several short-term prefetching policies are created using Markov models, which are used for modeling and predicting user’s browsing behavior over the Web.

Long-term prefetching policies: Global object access pattern statistics such as objects’ popularity and objects consistency are utilized to identify valuable (clusters of) objects for prefetching. In this type of scheme, objects with higher access frequencies and no longer update time intervals are more likely to be prefetched [7].

There are several steps that must be conducted to create Web prefetching scheme:

Step 1: A Web user requests an object in a particular time.
 Step 2: The proxy recognizes the Web user based on its IP address, and assigns it to one of the client groups. The proxy will search the existing clusters in order to find the requested existing Web object.
 Step 3: The proxy prefetches all objects that exist in the cluster from the origin servers. Next, the objects are managed by the proxy's cache replacement policy.
 Step 4: The proxy will send the requested object to user.

The proposed prefetching concept is based on context alteration. The design of the following prefetching strategy adapts to the changing contexts [9]:

If C_1, C_2, \dots, C_n , and their weights on prefetching decisions is W_i ($W_i > 0, \sum W_i=1, i=1..n$). Jin adapted Multi-Attribute Decision Making method to compute W_i by comparing the contexts in duo. He also put more consideration such as the grade of screen size, device status (remaining power) and wireless network connection status. Given W_1, W_2, W_3 are the weight on power, network delay and device display size for GUI. Let initialise X_{ij} ($i \leq j$) in matrix X be the pairwise comparison value of context i and context j . For example, context i 's importance is divided by context j 's while X_{ji} in X is a set of the reciprocal of X_{ij} . In general, number 5, 1 and 0.1 respectively stand for moderately more, equally and very less important, so Jin in [9] designate X as follows:

$$\begin{matrix}
 \text{power} & \begin{pmatrix} 1 & 5/3 & 5/1 \\ 3/5 & 1 & 3/1 \\ 1/5 & 1/3 & 1 \end{pmatrix} \\
 \text{network delay} & \\
 \text{device display size} &
 \end{matrix}$$

By calculating eigenvector of X , we get $W_1=0.56, W_2=0.33$ and $W_3=0.11$. Next, we calculate a normalized ranking:

$$C = \sum w_i \times \frac{CUR_i - MIN_i}{BEST_i - MIN_i}, i = 1..n$$

3.2. Prefetching approach on mobile environment

The prefetching problem can be solved in two steps. First, we need to find out the favorite request in which each file will be accessed. In this article, we estimate this locality with high access probability that will be requested by user in the near future. When the access probability of object is achieved, the following is how to determine the efficient way to fetch the file. Prefetching on mobile environment can be created by using technology such as .NET compact framework, J2ME and etc.

4. Web caching

Web caching is able to reduce bandwidth utilization, network jamming, and network traffic because it stores the repeatedly requested content nearer to users [16].

Xu et al. in [10] proposed solutions by creating proxy management. The Caching dynamic content is obtained by generating data and personalize data that contributes up to 30-40% of the total traffic. These types of data are normally identified as "uncachable". To further improve Web performance, reverse caching has also been suggested to make more dynamic content cachable and manageable [7]. However, the challenge still stuck on how to efficiently maintain consistency between the cached content and the data source that frequently change. Another important issue is the analysis of query semantics to evaluate a complex query over the cached content.

Caching streaming objects: It is predicted that streaming media such as music or video clips will symbolize a significant portion of Web traffic over the Internet. Due to the distinct features of streaming objects like big size, long duration, intensive use of bandwidth, and interactivity, conventional proxy caching techniques are not able to solve this problem. To solve these problems, many partial caching algorithms have been proposed in recent years [11,12]. The proposed algorithms expressed that even if small size of video is stock up on the proxy, the consumption of network will be reduced significantly.

Xu in [10] generally, collaboration among proxies is based on the premise that it would be faster and cheaper to fetch an object from another close proxy rather than the origin server. See Figure 3 for the cooperative cache organization.

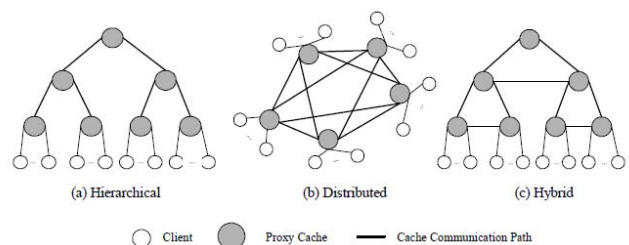


Figure 3. Examples of different cooperative cache organizations [10]

Teng et.al, in [13] proposed a combination between Web caching and Web prefetching. These two techniques can go together since the Web caching technique use the temporal locality while Web prefetching technique utilizes the spatial locality of Web objects. The proposed technique is obtained by evaluating the prefetching rules.

Lots of future works have been drawn by previous researchers especially on clustering prefetching,

caching on proxy level or even designing cache organization. The list of issues on Web caching and prefetching is drawn in Table 1.

Table 1. The comparison of previous work on prefetching and web caching

Author	Problem, solution and future works
Hung et al. (2008)	Proposed two levels caching to improve the average of cache access time and an adaptive prefetch strategy. Still hard to adapt with different application.
Nair and Jayasudha (2007)	Proposed an intelligent agent that is able to monitor bandwidth usage and helps the prediction engine to decide the number of Web pages to be prefetched. Need an improvement in artificial intelligence of the agent.
Jin et al. (2007)	Proposed an adaptive prefetching concept to be more intelligible in term of mobile context. Need to be paired with Web caching technology in order to produce faster performance.
Pallis et al. (2007)	Proposed clustering method between object that needs to be fetched and user position. Need more study on implementing another clustering algorithm.
Xu et al. (2006)	Proposed cooperative cache organizations between client and proxy cache. Need more study on searching nearby proxy when some objects are requested.
Teng et al. (2005)	Proposed an algorithm that called IWCP (standing for the Integration of Web Caching and Prefetching). In order to create appropriate combination of prefetching and Web caching by utilizing innovative cache replacement. Still hard to be implemented in different platform such as Web and mobile.

5. Prefetching and Web caching on mobile environment

According to Jiang and Kleinrock [17] prefetching can be used together with caching and pushing to improve the Internet performance. Harding et.al in [5] has proposed a concept of prefetching on mobile device by enhancing the feature of mobile device to the optimum utilization. By using the features user only connect to the server when it need to connect and only few amount of data that need to be transferred.

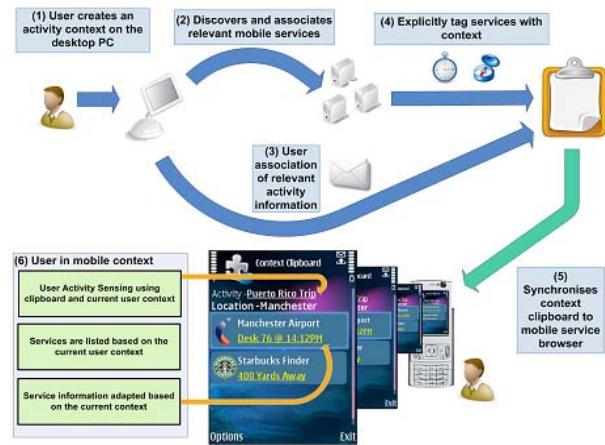


Figure 4. Process for preparing and using mobile services using offsetting [5]

On the diagram shown in Figure 4, almost all activities required more power, lots of tasks and services have been done in desktop application. The mobile environment only needs to be synchronized and customized a little bit when it needs more data. However, Harding in [5] mentioned that the method still faces challenges on application design, user interaction, usability and mentality model.

Ye et al. in [8] created a model for handling Web caching in multi-device that are connected to two kinds of wireless network: MSS (Mobile Support Station), and the neighbor peers to form the P2P network.

Mobile environment has limited screen size, memory and also the speed process. Therefore, lots of methods have been observed to handle those limitations. See Figure 5 for the mobile device system illustration.



Figure 5. Mobile device system (Windows pocket PC)

6. New framework, mobile environment, social network

Based on issues we have discussed in the previous section, we have proposed a new framework by combining Web caching and prefetching on mobile environment. This framework is designed to rise up the access of social network using mobile devices. Firstly, we handled the bandwidth loading by applying Web caching on proxy level. This Web caching is loaded by a combination between Rough Set, ANN and PSO. Secondly, we placed prefetching based on XML on mobile device in order to handle communication between client and server. Therefore, by handling and preparing data for client, we have made the client a smart client. The smart client will then control the timing to access to server. Next, we classified the feature on social network into certain priorities based on the popularity and file size. See Figure 6 for details.

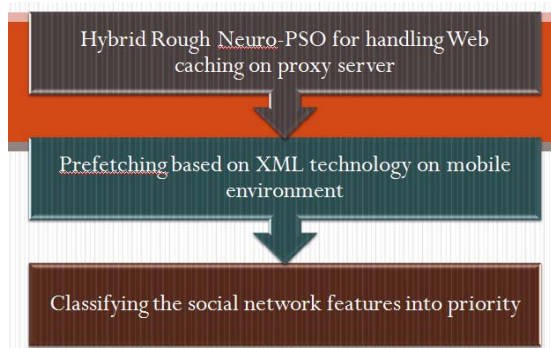


Figure 6. Proposed framework

6.1. Rough Neuro-PSO

We used PSO to adjust weight on ANN in order to find an optimal weight. This ANN is designed to train Web caching until it is able to find which object that very popular to be accessed. Then, Rough Set is used to get the Web caching rules. Consequently, it will be able to control load balancing among server and client. See Figure 7 for the detail.

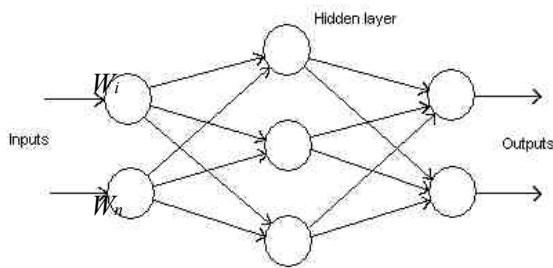


Figure 7. Hybrid Neuro-PSO

6.2. Prefetching based on XML

Said et.al [14] proposes a prefetching concept based on XML database. The client will do “cold data prefetching” on mobile device by creating query generator. Like Said et.al, we have implementing same idea. However, our approach more focuses on providing off setting and saving the data on mobile client to increase the performance of access time server.

XML format is file format that we used to save the user preference object on mobile device. Similar to Harding et.al [5], we have xml as offsetting record in mobile environment. See Figure 8 for illustration.

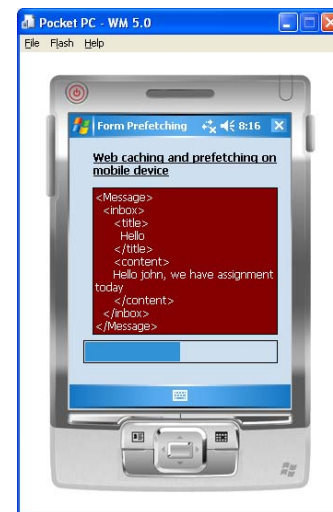


Figure 8. XML management on mobile device

According to Fox Interactive Media survey[15], lots of people on the Internet interact with social network applications such as Facebook, MySpace, Friendster and etc. only to read comments/messages or posting some messages while other people spend more time on checking profiles, settings and the layout or watch online TV episodes. See Figure 9 for details.

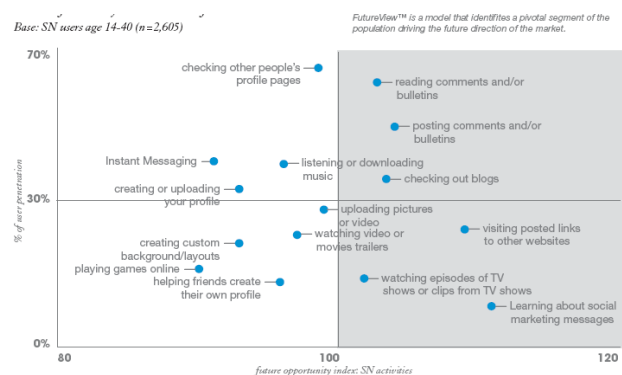


Figure 9. Social network trend [15]

Our classification on social network is to set up priorities among social network features or elements in order to balance out the network traffic. In this case, we take Facebook as our case study due to Facebook being the most popular among other social networking websites. See Table 2 for details.

Table 2. Social network classification

Facebook elements	Priority
Message	High
Forum /blog	Medium
Updated profile notification	Low

Table 2 shows that we have classified the priorities of Facebook elements according to Fox Interactive Media survey and the most popular Facebook feature that often being used by users. Based on the classification above, our system will be able to speed up the communication between the client and server.

7. Discussion and conclusions

We have presented a new framework to increase access speed in mobile environment context. Based on our hypothesis, Web caching will be able to reduce server and bandwidth load. At the same time, prefetching is believed to contribute on rising up the communication to server from client side. Our third step is classifying the most favorable Facebook feature accessed by user. Our three approaches are believed to be able to solve the problem on time-consuming access between mobile client and server. In the future, we will present a comparison time based on our full experimental and testing.

8. Acknowledgments

The authors would like to thank Ministry of Science, Technology and Innovation Malaysia (MOSTI) and Universiti Teknologi Malaysia (UTM) for their kind financial support under the e-science fund.

9. References

[1] Padmanabhan, V.N. and Mogul, J.C., *Using Predictive Prefetching to Improve World Wide Web Latency*, Computer Communications Review, vol.26, ACM, SIGCOMM'96, July 1996, pp. 22–36.
 [2] Davison, B.D., *Adaptive Web Prefetching*, Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW, 2001.

[3] Kroeger, T. M., Long, D. E. and Mogul, J. C., *Exploring the Bounds of Web Latency Reduction from Caching and Prefetching*, USENIX Symposium on Internet Technologies and Systems (USITS), Monterey, CA, December 8-11, 1997.
 [4] Maltzahn, C., Richardson, K. J., Grunwald, D. and Martin, J. H., *On Bandwidth Smoothing*, Proceedings of the Fourth International Web Caching Workshop (WCW99), San Diego, CA, March 31-April 2, 1999.
 [5] Harding, M., Storz O., Davies, N. and Frida, A., *Planning Ahead: Techniques for Simplifying Mobile Service Use*, HotMobile 2009, Santa Cruz, CA, USA. ACM, February 23–24, 2009.
 [6] Bestavros, A. and Cunha, C., *A Prefetching Protocol Using Client Speculation for The WWW*, Tech. Rep. TR-95-011, Boston University, Department of Computer Science, Boston, MA 02215, Apr. 1995.
 [7] Pallis, G., Vakali, A. and Pokorny, J., “A Clustering-based Approach for Short-term Prefetching on a Web Cache Environment”, *Computers & Electrical Engineering Journal*, Elsevier, 34(4), 2008, pp. 309-323.
 [8] Ye, F., Li, Q. and Chen, E., *Adaptive Caching with Heterogeneous Devices in Mobile Peer to Peer Network*, SAC'08, ACM, Fortaleza, Ceará, Brazil, March 16-20, 2008.
 [9] Jin, B., Tian, S., Lin, S., Ren, X. and Huang, Y., *An Integrated Prefetching and Caching Scheme for Mobile Web Caching System*, The 8 ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE, 2007.
 [10] Xu, J., Liu, J., Li, B. and Jia, X., *Caching and Prefetching for Web Content Distribution*, IEEE Computing in Science and Engineering Magazine: Special Issue on Web Engineering, 2006.
 [11] Nair, A.S. and Jayasudha J.S., *Dynamic Web Prefetching Technique for Latency Reduction*, International Conference on Computational Intelligence and Multimedia Applications, IEEE, 2007.
 [12] Hung, S.H., Wu, C.C. and Tu, C.H. *Optimizing the Embedded Caching and Prefetching Software on a Network-Attached Storage System*, IFIP International Conference on Embedded and Ubiquitous Computing, IEEE, 2008.
 [13] Teng, W.G., Chang, C.Y. and Chen, M.S., “Integrating Web Caching and Web Prefetching in Client-Side Proxies”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, No. 5, May 2005.
 [14] Said, E.G., Omar, E.B., Robert, R., “Data Prefetching Algorithm in Mobile Environments”, *European Journal of Scientific Research*, ISSN 1450-216X Vol.28 No.3 (2009), pp.478-491.
 [15] Fox Interactive Media, Inc., *Never Ending Friending, - A Journey into Social Networking*, 2007.
 [16] Williams, S., Abrams, M., Standridge, C. R., Abdulla, C. and Fox, E. A., *Removal Policies in Network Caches for World-Wide Web Documents*, Proceedings of the 1996 SIGCOMM, ACM, July 1996, pp. 293–305.
 [17] Jiang, Z. and Kleinrock, L., *Web Prefetching in a Mobile Environment*, IEEE Personal Communications, October 1998.