A Survey on Web Pre-Fetching and Web Caching Techniques for Latency Optimization in Mobile Environment

P. Amudha Bhomini[1], Dr. Jayasudha J.S[2]

[1]N.M.Christian College Marthandam
[2]SCT College of Engineering, Trivandrum
amudhabhomini[at]yahoo.co.in
jayasudhajs[at]gmail.com

Abstract: ***Increasing popularity of the World Wide Web over the past few years has imposed a significant traffic burden upon the internet. The continued increase in demand for objects on the internet causes severe overloading in many sites and network links. Web traffic reduction techniques are mandatory for accessing the websites efficiently. Data caching and prefetching techniques play a key role in improving the web performance in mobile information systems. Generally the mobile devices have less amount of memory storage than computers. Therefore the key challenge in prefetching is to determine 'what to prefetch' so that the improvement in performance will be enhanced. In a mobile environment the mobility brings the new challenges to the prefetch problem.***

Keywords: Prefetching, caching, latency

1. Introduction

Web traffic is a major component of internet traffic, which corresponds to the unstable growth and causes bandwidth consumption that this medium is experiencing. Intention of many researches towards this area is to reduce users perceived latency when they are navigating through websites. The prefetching and caching techniques help a lot in reducing latency. An important advantage of the WWW is that many web servers keep a server access log of its users. These logs are used to prepare a prediction model for future document accesses. Based on these models, one can obtain frequent access patterns in web logs and mine association rules for path prediction. Web prefetching can be applied in web environment between clients and web server, proxy servers and web server and between clients and proxy servers [1]. Applying perfecting between web server and clients reduces latency but increases network traffic and between proxy server and web server decreases bandwidth usage. Web prefetching is also used to fetch web pages by proxy server/client before the page is requested by a client/proxy server. The proxy server intercepts the requests from clients, serves with the requested object if it is present in proxies (pre-fetch area), or retrieves new objects from the appropriate web server, then caches the new objects or updates the existing objects if it has been modified since the last reference. The key idea forecast takes improvement of the idle periods to fetch the files that will likely be requested in the future, so that the user's average waiting time can be decreased. The major factor is to select an efficient web prefetching algorithm; the selection is done by considering its ability to predict the web object to be pre-fetched in order to reduce the latency, without consuming more bandwidth.

Web pre-fetching techniques are broadly categorized into three. They are probability based, clustering based and using weight-functions [2].Probability based method uses history access and assumes that the request follows the same pattern. Clustering based prefetching technique assumes that the pages that are close to those previously fetched pages are more likely to be requested soon. In the weight functions, a weight is given to an accessed page and assumes that the one with the highest weight will be referred in the near future. Although web browsing is an important application for prefetching, generally prefetching scheme may be applied to any network application in which prefetching is possible.

Caching is one of the key techniques which promise to overcome some of the portal performance issues. Web caching techniques can be characterized as back-end or frontend [3]. Back-end approaches are deployed inside the site infrastructure at the original server side. They can reduce content generation delays but they do not address network related delays and bandwidth consumption issues. Front-end approaches concern caching outside the site infrastructure at a proxy side or a cache that resides at the edge of a Content Delivery Network (CDN). Front-end web caching techniques are serving at client side and they have difficulty in handling dynamic web pages due to their strong dependency on the back-end site infrastructure.

One important performance factor of web caches is the replacement strategy. In the simple replacement process, on a cache miss, the cache acquires and stores the requested object. The cache evicts a certain number of objects if the size of the all cached objects exceeds the given cache size. LRU, Pithow/Recker's strategy, SIZE, LFU, LFU-MIN, Greedy Dual Size, Taylor Series, Prediction and RAND are commonly used replacement strategies [46]. Among all strategies LRU has been applied successfully in many different areas.

The rest of the paper is organized as follows. In section 2 various prefetching techniques are discussed. Section 3 deals with different caching techniques. Conclusion remarks are provided in section 4.

2. Prefetching Techniques

2.1 Interactive perfetching

In interactive perfetching technique [4] the retrieved pages are fetched along with hyper links and inline images. Since this method prefetches all hyperlinks in web pages, it requires a lot of space to store the pre-fetched web pages and it increases web traffic for perfetching the hyperlinks.

2.2 Adaptive prefetching

Jiang and Leonard [5] proposed an Adaptive prefetch scheme, in which prefetching is based on user access history and network conditions. It has two modules, a prediction module and a threshold module. Files to be prefetched are the one whose access probabilities greater than or equal to the prefetch threshold.

2.3 Markov Models

Markov models and Hidden Markov Models predict web pages based on the probabilities of web access patterns [6, 7].Traditional Markov models predict the next web page, by matching the user's current access sequence with the user's historical web access sequence. The 0-order, 1-order … k-order Markov models are available for predicting the page to be prefetched. Accuracy of prediction is improved in higher order Markov models, but the cost and complexity is increased.

2.4 Prediction by Partial Match (PPM)

PPM makes pre-fetching decisions by reviewing the URLs that clients on a particular server have accessed over some period [23]. PPM algorithm has three parameters; order, depth and threshold. Order is the length of the history substring that the algorithm uses to find a match. Depth is the number of accesses into the future the algorithm attempts to predict. Thershold is the minimum probability and access must have in order to be considered a pre-fetch candidate. The standard PPM model uses multiple Markov models to store historical URL paths.

2.5 Longest Repeat Sequence Model (LRS)

LRS is a variation of PPM model which keeps the longest repeating subsequences and stores only long branches with frequently accessed URL predictors [8].

2.6 Dynamic Prefetching

Dynamic Prefetching [9] technique stores the preference list of sites of a user in a data base of proxy server. The list of accessed URLs and its weight is stored in a hash table. Depends on weight and bandwidth usage, the agent decides the number of URLs to be prefetched.

2.7 Top – 10 approach

In Top-10 approach the server periodically calculates the list of most popular documents [10]. These documents are periodically pushed from web servers to web proxies then to clients. It is simple and easy to implement in client server architecture.

2.8 Domain top approach

The proxy's active knowledge of their most popular domains and documents with client access profiles are combined in the domain-top approach (DT). Proxy side is responsible for periodically calculating popular domains and popular documents in each domain. Based on the proxy's active knowledge of popular domains and documents, a rank list is prepared for anticipating the client's future requests [11]. In this approach prefetching occurs only in proxy server.

2.9 Link prefetching

It is a browser mechanism, utilizes browser idle time to download documents that the user might visit in the near future. Fisher et.al devised an approach for link prefetching [12]. The browser follows special directives from the web server or proxy server that instructs it to prefetch specific documents. It allows the servers to control the contents to be prefetched by the browser.

2.10 Non-interfering prefetching

This scheme avoids interference by effectively utilizing spare resources on the servers and the network [13]. Prediction and resource management are the important tasks. Predictor predicts prioritized lists of high valued documents for prefetching. Resource manager limits the number of documents to prefetch and schedules the prefetch request to avoid interference.

2.11 Semantic prefetching

In this scheme prediction of future request is based on preferences of past retrieved documents in semantics [14, 15]. Semantic prefetching techniques tend to capture the client surfing interest from users past access patterns and predict future preferences from a list of objects when a new page is visited. Semantic knowledge of web documents are automatically extracted and adaptive semantic nets are constructed between web documents.

2.12 Prefetching based on web usage mining

The prediction scheme for prefetching [16] is based on a learning algorithm called Fuzzy-LZ which mines the history of user access and identifies patterns of recurring accesses. The prediction algorithm mines access logs collected at or near the user. This focus on the access behavior of individual users rather than on the structure of content at any one web server.

2.13 Online PPM prediction model

This model is based on a non compact suffic tree. It keeps most recent web object using a sliding window. This model combines entropy, prediction accuracy rate and the longest match rule. Online Prediction by Partial Match (PPM) [17] keeps the most recent web requests using a sliding window and deletes the request as they drop out of the sliding window. The log entries record user request sequence. Three modules of online PPM are insertion, deletion and prediction. Prediction module determines the page to be prefetched.

2.14 Rule Assisted Prefetching (RAP)

Qiang Yang and Zhen Zhang [18] have focused contribution on the design of a novel perfecting strategy called Rule Assisted Prefetching (RAP) [19]. It identifies a set of association rules from the Web server's access log and finds the first rule whose X is not in the cache. Read the X in to cache. The performance shows that RAP coupled with the LFU-MIN replacement policy performs the best overall.

2.15 Predictions from HTML content

Davidson [20] suggested a technique that predicts the user’s next request by analyzing the recently requested page content by the user. It prefetches the top 5 URLs predicted by this system.

2.16 Proxy based ad prefetching

A mobile advertising system has five parties: mobile clients, advertisers, ad servers, ad exchanges, ad networks [21]. This approach uses a proxy between the ad server and the mobile client. A client with available ad slots contacts the proxy that prefetches a batch of ads from the ad exchange (through the ad server) and sends the batch to the client. After the client has displayed all ads of the batch, it contacts the proxy again and get the next batch of ads.

2.17 Informed mobile prefetching

This technique uses available networks while ensuring that prefetches do not degrade network performance for foreground activity [22]. It decides when and how much data to prefetch using a cost/ benefit analysis inspired by Transparent informed prefetching. It estimates the potential cost and benefit of prefetching a data item for each of its three metrics: performance, energy use and data consumption.

2.18 Intelligent mobile prefetching

Siti Mariyam Shamsuddin et al. proposes a new technique known as an Intelligent Moble Web Pre-fetching (IMWeP) that creates pre-fetching in a mobile environment using Extensible Markup Language (XML) technology. This technique proves to be faster in searching or accessing process in mobile applications. A case study on Facebook Mobile investigates the loading process by creating priority ranking on XML files using the proposed IMWeP technique [47]. This approach is mainly used for social network site Facebook. It calculates the cumulative number of hits for each Facebook mobile feature (inbox, home, profile and friends) when the user visits in future.

3. Web Caching Techniques

3.1 Co-operative caching

Cooperative proxy caching is the sharing and coordination of caches among multiple caching proxies [24, 25, 26]. Co-operation among caches can be performed in horizontal, vertical and orthogonal dimensions. Horizontal cooperation is performed by geographically clustered caches that have equal distant to the web servers. Vertical cooperation is performed by geographically distributed caches that have unequal distant to the web servers.

3.2 Distributed caching

Distributed cache is a set of cooperative caches placed at the same level of the network, a missing of resources at one proxy causes a search in all cooperating cache servers for caches hit. In distributed caching no intermediate caches are set up and there are only institution caches at the edge of the network which cooperate to serve each other’s misses [27, 28, 29].

3.3 Hierarchical caching

Proxies or cache servers are arranged in a tree like structure either logically or physically. In hierarchical caching architecture, caches are placed at multiple levels of the network [30]. Individual caches can be interconnected hierarchically to mirror an inter network’s topology. Each bottom level cache is associated with a set of clients. A client request is first sent to the bottom level cache and then iteratively forwarded up the hierarchy such as institutional cache, regional cache, national cache, until the request is satisfied. If the root cache does not have target object, the request is finally directed to the origin server [31].

3.4 Hybrid caching

Hybrid caching architecture combines hierarchical caching with distributed caching at even level of a caching hierarchy [32]. In a hybrid scheme, cache cooperates with other caches at same level or at a higher level using distributed caching. The document is fetched from parent/neighbor cache that has the lowest round trip time.

3.5 Transparent web caching

Transparent web caching uses network devices to redirect HTTP traffic to caching servers. This technique is called transparent because web browsers do not have to be explicitly configured to point to a cache server [33]. There are two ways to deploy transparent proxy caching: at the switch level and at the router level. Router based transparent caching uses policy-based routing to direct request to the appropriate caches. In switch based transparent caching, the switch acts as a dedicated load balancer [34].

3.6 Cascaded caching

Cascaded caching is also known as Multi-layer caching; Multiple copies of same object may be available in many caches placed at different locations. There can be cache servers both at the client location and server location. This allows a cache server at client location to get a requested document from the cache server. The overall performance of cascaded caching depends on how the cache contents are managed, including object placement and replacement algorithms [35].

3.7 Adaptive caching

An adaptive web caching has a scalable, robust, adaptive, and fully distributed protocol for sell-organizing cache servers into overlapping multicast groups [36]. Web caches maintain a URL routing table. The primary keys of URL routing table are URL prefixes, associated with one or more identifiers to the next hop caches or cache groups. The routing table is used for deciding whether to forward a request to another cache in the web caching infrastructure. Adaptive algorithms are used for exchanging of information among caches in a cache mesh. Cache group management protocol is used for making the entire cache topology group-wise connected and management protocol is used for minimizing the number of hops a URL request must travel upon cache fault.

3.8 Server side caching

Even if client side caches could not satisfy the user requests, a cache at sever side is useful in reducing the delay in accessing the documents from the server's hard disk. Temporal and geographical localities of reference are exploited on a much large scale at server side [37]. Web server accelerator contains a cache and load balancer. It resides in front of a web server for delivering cached responses and leaving the role of content generation to the web server. Cached objects are directly sent from the accelerator to the clients.

3.9 Caching of Dynamic contents

Dynamic web pages are created on request by application programs stored in the back-end site infrastructure, caching of dynamic web pages are essential for improving the performance of web sites containing significant dynamic content and information personalized to individual users [38]. Dynamic content has three forms of locality: identical requests, equivalent requests and partially equivalent requests. Identical requests have identical URLs which result in the generation of the same content. The URLs of equivalent requests are syntactically different but result in generation of identical content. Partially equivalent requests result in generation of content which can be used as temporary place-holder for each other.

3.10 Proxy caching

Proxy servers intercept HTTP requests from clients and if the requested objects are in its cache, it returns the object to the client. If not, it forwards the object to the server, gets the object, stores it in cache and returns the object to the user. The drawback of this design is that the cache represents single point failure in the network [39]. This can be avoided by sharing the caches of proxy servers.

3.11 Scalable Asynchronous Cache Consistency Scheme

A novel scheme called scalable asynchronous cache consistency scheme (SACCS) was proposed by Wang et al. [43, 44] to maintain data consistency for mobile computing systems. SACCS relies on three key features: (1) use of flag bits as the server and mobile device's cache to maintain cache consistency; (2) use of an identifier for each entry in mobile device's cache after its invalidation in order to maximize the broadcast bandwidth efficiency; and (3) rendering of all valid entries of a mobile device's cache to uncertain state upon wake up. These features make SACCS a highly scalable algorithm with minimum data management overhead in both single and multi – cell environments [45].

3.12 Greedy Dual Least Utility Caching

Novel energy and bandwidth efficient data caching mechanism, called Greedy Dual Least Utility (GD-LU) that enhances dynamic data availability while maintaining consistency [40]. The utility-based caching mechanism considers several characteristics of mobile distributed systems, such as connection-disconnection, mobility handoff, data update and user request patterns to achieve significant energy savings in mobile devices. GD-LU has two main components: GD-LU cache replacement algorithm and GD-LU passive prefetch algorithm. The GD-LU cache replacement algorithm selects data items with the most utility to cache in local memory. In passive prefetch, mobile devices acquire the data items from broadcast channels for the user's requests based on data items relative utility values. Based on priority queue management, the GD-LU cache replacement and passive prefetching algorithms achieve a time complexity of O(log N) where N is the number of data items in the cache. Experiments demonstrate that the proposed caching mechanism achieves more than 10% energy saving and near-optimal performance tradeoff between access latency and energy consumption.

3.13 Rough Neuro – PSO Web Caching

Sarina Sulaiman et al. proposed a new hybrid technique based on combination of ANN and Particle Swarm Optimization (PSO) for classification of web object either to cache or not and generate rules from log data by using Rough Set technique on proxy server (Rough Neuro-PSO) [42]. This approach is needed because mobile context has limited resources like speed and memory. XML file is used for prefetching which is saved into moble memory. Prefetching that uses XML file is much faster to be searched or accessed. In Web caching side, enhance the speed by using Rough Neuro-PSO in order to choose the log. PSO is used to adjust weights on ANN in order to find optimal weights. ANN is designed to train Web caching until it is able to find which object that are very popular to be accessed. Then, Rough Set is used to get the Web caching rules. This framework was designed to rise up the access of social network using mobile devices.

3.14 Universal Mobile Caching

Universal Mobile Caching (UMC), which is suitable for managing object caches in structurally varying environments and which is self-optimizing for changing workloads [41]. UMC is based on a simple set of basic criteria which reflects a spectrum of possible caching policies. UMC has demonstrated the ability to provide caching benefits in the on-demand retrieval of web documents for the mobile web, wherein multiple levels of intervening caches can create adverse workloads for other general caching schemes. UMC policy offers a self-optimizing replacement algorithm that is usable for general object caching. Universal Mobile Caching uses a very simple set of object properties to select which objects (of varying size) will be removed from the cache.

4. Conclusion

In this paper an exhaustive survey of prefetching and caching strategies are proposed. These techniques are used for reducing user's perceived latency. Nowadays, many researches are going in the mobile environment for integrating caching and prefetching techniques to reduce the latency as well as the network traffic.

References

[1] Seung Won Shin, Byeong Hag Seong & Daeyeon Park, "Improving World-Wide- Web Performance Using Domain-Top Approach to Prefetching", Fourth International Conference on High-Performance Computing in the Asia-Pacific Region, vol. 2, pp. 738-746, 2000.

[2] Yingyin Jiang, Min-You Wu & Wei Shu, "Web Prefetching: Costs, Benefits and Performance", Proceedings of 7th International Workshop on Web Caching and Distribution, 2002.

[3] Manolis Veliskakis, John Roussos, Panos Georgantas & Timos Sellis, "DOM Proxy : Enabling Dynamic-Content Frontend Web Caching", Proceedings of 10th International Workshop on Web content Caching and Distribution, IEEE Computer Society September, pp. 56-61, 2004.

[4] Ken-ichi chinen & Sugunu Yamaguchi "An Interactive Prefetching Proxy Server for the Improvement of WWW Latency", Proceedings of the 7th Annual Conference of the Internet Society, Kuala Lumpur , June 1997.

[5] Zhimei Jiang & Leonard Kleinrock, "An Adaptive Network Prefetch Scheme", IEEE Journal on Selected Areas in Communications, Vol. 16, No. 3, pp. 358-368,1998.

[6] Doug Joseph, Dirk Gurunwaid, "Prefetching using Markov Predictors", Proceedings of the international symposium on Computer Architecture, June 1997.

[7] Xing Doushan, Shen Junyi, "A Markov model for Web Acees Predictor", Proceedings of IEEE conference on Computing in Science and Engineering, pp. 34-39, 2002.

[8] Xin Chen, Xiaodong, Zhang, "A Popularity Based Prediction Model for Web Prefetching", IEEE Computer, Vol. 36 No. 3, pp. 63-70, 2003.

[9] Achuthsankar S.Nair & J.S. Jayasudha, "Dynamic Web Pre-fetching Technique for Latency Reduction", International Conference on Computational Intelligence and Multimedia Applications, 2007.

[10] Evangelos P Markatos, Catherine E Chronaki, "A top 10 approach to prefetching the web", Proceedings of the Internet Sumit, 1998.

[11] Won Shin, Byeong Hag Seong and Dareyeon Park "Improving World – Wide Web performance using Domain Top Approach to Prefetching", 4th International Conference on High Performance Computing in the Asia – Pasific Region Vol. 2, pp. 738-746, 2000.

[12] Darin Fisher,Gagan Saksena, "Link Prefetching in Mozilla: A Server Driven Approach", Proceedings of the 8th International conference on web content caching and distribution, 2003.

[13] Ravi Kokku,Praveen Yalagandula, Arun Venkataramani, Mike Dahlin, "NPS: A Non interfering Deployable Web Prefetching System", Proceedings of USENIX, Symposium in Internet Technologies and Systems, 2003.

[14] Cheng-Zhong Xu & Tamer I.Ibrahim, "Towards Semantic Based Prefetching to reduce Web Access Latency", Proceedings of IEEE symposium on Applications and Internet, 2003.

[15] Cheng-Zhong Xu & Tamer I.Ibrahim, "A Keyword-Based Semantic Prefetching Approach in Internet News Services", IEEE Transactions on Knowledge and data engineering, vol.16, pp. 601-611, 2004.

[16] Daby M. Sow. David P. Olshefski, Mandis Beigi and Guruduth Banavar "Prefetching based on Web Usage Mining", Middleware, pp. 262-281, 2003.

[17] Zhijie Ban, Zhimini Gu and Yu Jin, "An Online PPM Model for Web Prefetching", WIDM'07, November 9, 2007.

[18] Qiang Yang & Zhen Zhang, "Model Based Predictive Perfecting", IEEE International Workshop on Database and Expert Systems Applications, 2001.

[19] Bin Lan, Stephane Bressan, Beng Chin Ooi, Kian-Lee Tan, "Rule-Assisted Prefetching in Web-Server Caching", ACM International Conference on Information and Knowledge Management, 2000.

[20] B. Davidson, "Predicting Web Actions from HTML Content", Proceedings of 13th ACM Conference on Hypertext and Hypermedia, 2002.

[21] Prashanth Mohan,Suman Nath and Oriana Riva, 'Prefetching mobile ads, Can advertising systems afford it?' Eurosys'13, ACM, April 15-17, 2013.

[22] Brett D. Higgins, Jason Flinn, T.J. Giuli" Informed Mobile Prefetching", Mobisys'12, ACM 2012.

[23] Themistokils, Palpanas, Alberto Mendeizon, "Web Prefetching Using Partial Match Prediction", Proceedings of 4th Web Caching Workshop, October 1998.

[24] Jean Marc Menaud, Valerie lssarny, Michel Banatre, "A Scalable and Efficient Cooperative System for

Web Caches", IEEE Concurrency, Vol. 8, No. 3, pp. 56-62, July 2000.

[25] Sandra G. Dykes, Kay A. Robbins, "A Viability Analysis of Cooperative Proxy Caching", Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 1205-1213, 2001.

[26] Alec Wolman, Geoffrey M. Voelker, Nitin Sharma, "On the Scale and Performance of Cooperative Web Proxy Caching", Proceedings of 17th ACM Symposium on Operating Systems Principles, Operating Systems Review, Vol. 34, No.5, pp. 16-31, December 1999.

[27] Pablo Rodriguez, Christian Spanner, Ernst W. Blersack, "Analysis of Web Caching Architectures: Hierarchical and Distributed Caching", IEEE/ACM Transactions on Networking, Vol. 2, pp. 404-418, 2001.

[28] M.N. Wijesundara, T.T. Tay, "Distributed Web Caching", Proceedings of 8th International IEEE Conference on Communication Systems, pp. 1142-1146, 2002.

[29] M.N. Wijesundara, T.T. Tay, "An Object Replacement Strategy for Global Performance in Distributed Web Caching", Proceedings of International Conference on Communication Technology, pp. 1687-1690, 2003.

[30] Mudashiru Busari, Carey Willamson, "Simulation Evaluation of a Heterogeneous Web Proxy Caching Hierarchy", Proceedings of IEEE International Symposium on Modeling Analysis and Simulation of Computer and Telecommunication System, pp. 279-388, 2001.

[31] Anirban Mahanti, Carey Willamson, Derek Eager, "Traffic Analysis of a Web proxy Caching Hierarchy", IEEE Network, Vol. 14, No. 3, pp. 16-23, May 2000.

[32] Mingkuan Liu, Fei - Yue Wang, Daniel Zeng, Lizhi Yang, "An Overview of World Wide Web Caching", Proceedings of IEEE International Conference on systems, and Cybemetics, Vol. 5, pp. 3045-3059,2001.

[33] Zhengang Liang, Hossam Hassanein, Patrick Martin, "Transpaent Distributed Web Caching", Proceeding of 26th Annual IEEE Conference on Local Computer Networks, pp. 225-231, 2001.

[34] Athena I. Vakali, George E. Pallis, "A Study of Web Caching Architectures and Performance", Proceedings of 5th World Multi - Conference on Systemics, Cybernetics and Informatics, 2001.

[35] Xueyan Tang, Samuel T. Chanson, "Coordinated Management of Cascaded Caches for Efficient Content Distribution", Proceedings of 19th IEEE International Conference on Data Engineering, pp. 37-42, 2003.

[36] Lixia Zhang, Sally Floyd, Van Jacobson, "Adaptive Web Caching", Proceedings of the NLANR Web Cache Workshop, June 1997.

[37] Azer Bestavros, "WWW Traffic Reduction and Load Balancing Through Server Based Caching", IEEE Concurrency, Vol. 5, No. 1, pp. 56-67, January 1997.

[38] James R. Challenger, Paul Dantzig, Arun Iyengar, Mark S. Squillante, Li Zhang, "Efficiently Serving Dynamic Data at Highly Accessed Web Sites", IEEE/ACM Transactions on Networking, Vol. 12, No. 2, pp. 233-23, April 2004.

[39] Athena I. Vakali and George E. Pallis, "A Study of Web Caching Architectures and Performance", Proceedings of 5th World Multi-conference on Systemics, Cybermeics and Informatics, 2011.

[40] Johann Marquez, Josep Domenech, Jose A. Gil and Ana Pont, "Exploring the Benefits of Caching and Prefetching in the Mobile Web", Spanish Ministry of Education and Science and the European Investment Fund for Regional Development (FEDER) under grant TSI 2005.

[41] Ganesh Santhanakrishnan, Ahmed Amer, Panos K. Chrysanthis, "Towards Universal Mobile Caching", MobiDE'05 June 12, 2005.

[42] Sarina Sulaiman, Siti Mariyam Shamsuddin & Ajith Abraham, "Rough Neuro – PSO Web Caching and XML Prefetching for Accessing Facebook from Mobile Environment", World Congress on Nature and Biologically Inspired Computing, 2009.

[43] Z. Wang, S.K. Das, H. Che & M. Kumar, "SACCS: Scalable asynchronous cache consistency scheme", Proceedings of ICDCS International Workshop on Mobile Wireless Networks, pp. 797-802, 2003.

[44] Z. Wang, S.K. Das & H. Shen, "Investigation of cache maintenance strategies for multi-cell Mobile Data Management", pp.29-44,2003.

[45] Huaping Shen, Mohan Kumar, Sajal K. Das and Zhijun Wang, "Energy - Efficient Data Caching and Prefetching for Mobile Devices Based on Utility", Springer Science, Mobile Networks and Applications, pp. 475-486, 2005.

[46] Stefan Podliping and Laszlo Boszormenyi, "A Survey of Web Cache Replacement Strategies", ACM, Computing Surveys, Vol. 36, pp. 374-398, Dec. 2003.

[47] Sarina Sulaiman, Siti Mariyam Shamsuddin & Ajith Abraham, "Intelligent Mobile Web Pre-fetching (IMWeP) Using XML Technology", 6th IEEE International Conference on Next Generation Web Services Practices, pp. 475-480, 2010