UTM UNIVERSITI TEKNOLOGI MALAYSIA
RESEARCH UNIVERSITY

International Journal
of
Innovative Computing

Journal Homepage: http://se.cs.utm.my/ijic

A Survey of Web Caching Architectures or Deployment Schemes

Sarina Sulaiman[1]
Soft Computing Research Group (SCRG)
Faculty of Computing
Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
sarina@utm.my

Siti Mariyam Shamsuddin[2]
Soft Computing Research Group (SCRG)
Faculty of Computing
Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
mariyam@utm.my

Ajith Abraham[3]
Machine Intelligence Research Labs (MIR Labs)
Washington, 98092, USA
http://www.mirlabs.org
abraham@ieee.org

***Abstract*—Web caching (WC) and Web pre-fetching (WP) are ubiquitous techniques used to increase the speed of Web loading processes. This paper commences with a background review on WC scheme with some issues in mobile WC and integration of WC and WP. It also provides a review on the WC architectures or deployment schemes to find the research gap for this area. At the end of this paper is highlighting the future works on WC.**

Keywords — Web Caching, Web Pre-fetching, Architecture or Deployment Scheme

I. Introduction

An interesting fact is that many people tend to access the same piece of information repeatedly [1,2] in World Wide Web (WWW). This could be related to data, news, course notes, technical papers and so on. If too many users attempt to access a website simultaneously, then they may experience problems in getting connected, especially for people who use a mobile device to access the m-services. This is due to slow responses from the server as well as incapability of website in coping with the load. In addition, mobile context has limited resources like speed, memory and screen size [3,4,5,6,7]. Consequently, a better technique for Web loading process is vital.

An alternative in tackling these problems is by implementing Web caching (WC) [8,2] and pre-fetching (WP) [9,8,10,2] in enhancing Web access. WC is beneficial to broad users including those who rely on slow dial-up links as well as on faster broadband connections. The word caching refers to the process of saving the data for future use. In other words, WC is the process of saving copies of content from the Web closer to the end user for quicker access. It uses algorithms to predict user's needs to specific documents and stores important documents. According to Curran and Duffy [11], caching can occur anywhere within a network, on the user's computer or mobile devices, at a server, or at an Internet Service Provider (ISP). Many companies employ Web proxy caches to display frequently accessed pages to their employees, as such to reduce the bandwidth with lower costs [12].

At the same time, WP is another well-known technique in reducing user Web latency by preloading the Web object that is not yet requested by the user [9,8,2]. In other words, WP is a technique that downloads the probabilistic pages that are not requested by the user but could be requested again by the same user. Conventionally, there is some elapse time between two repeated requests by the same user. Pre-fetching usually performs the preloading operation within an elapse time and puts Web objects into the local browser or proxy cache to satisfy the next user's requests from its local cache.

Nevertheless, the WC and WP technologies are the most popular software based solutions [9,10,13,14]. Caching and pre-fetching can work individually or combined. The blending of caching and pre-fetching (so-called pre-caching) doubles the performance compared to single caching [15,16,17,14]. These two techniques are very useful tools to reduce congestion, delays and latency problems. Besides, combination of Web pre-caching and XML technologies will give a momentous enhancement of XML management [18] and Web infrastructure mainly in a mobile environment [5,19,20]. Several benefits of XML implementation are effective and efficient delivery [18], advantageous human language usage, easy to read and understand [21], easy and efficient data searching and easy processing by computers. This solution can also be exploited to solve the problem of poor performance to access websites or download files that resulted from limited resources in mobile devices including storage, processing power, display and communication capacity of the mobile devices [22,23]. Besides, a cache memory management between mobile clients and Web server [24] as another existing issue should be tackled to enhance the performance of these technologies.

In addition, the hidden and meaningless information from Web log mining can be generated through significant rules [17,25,76, 77]. The original Web log data is contaminated or polluted by numerous types of null, irrelevant and redundant information or meaningless noisy information such as proxy caches, local caches, corporate firewalls and others [25]. Consequently, a proxy cache will record a Web logging of all clients that use the same server. However, a decision on location for architecture of WP engine will affect the prediction of Web objects [17]. In this case, to reduce usage of a mobile device memory and to reduce the latency, the best solution is to put this engine near to the client-side, which means in the WP between Web clients and proxy cache [26,27].

The rest of this paper is organised to describe the research background on WC and WP. In addition, the next two sections convey the seven deployment schemes or architectures of WC on proxy cache and mobile environment, followed by summary and comparison of all of the schemes. The last section concludes this paper with the future works in this area at the end.

II. WEB CACHING AND PRE-FETCHING

Many studies attempt to prove that the WC performance with an integration of WP technology is capable to increase the Web performance[23,9,10,28,29,30,31,32,33]. Some studies in this area have also been done for mobile environment [5,20,33,34,23,7, 35,36,37,29,38,39,40]. However, there are only a few investigations implemented to real cases or scenarios for communication between a mobile device, and its server [8] specifically in the aspects of application design, user interaction, mentality model [33] and usability [41].

Consequently, this research area combines all aspects of WC [8] as shown in Figure 1 that involves architecture, coordination requirement, network traffic, complementary technique, practical and performance.

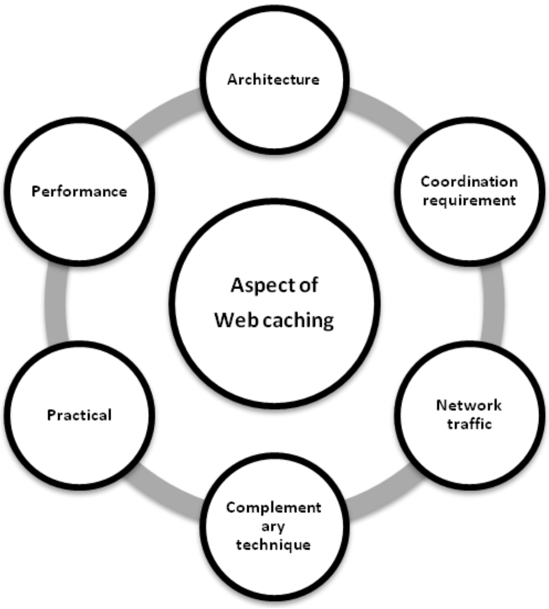

Fig. 1. Aspects of Web caching

In addition, Figure 2 depicts the detail of sub aspects for each WC aspect. For the first aspect, this paper will cover all sub aspects. The second aspect is requiring coordination among cache replacement algorithms to place Web documents within a cache and to minimise different parameters such as the hit ratio, the byte hit ratio, the latency and many others [8,42,43]. Nevertheless, many previous works only involves cache replacement algorithms as a main consideration [40,42,8,31,44,45] to measure the performance of Web caching compared to other sub aspects. The third aspect is crucial in WC research including the popularity characteristics of Web documents, the sizes of files, the time taken either to access a page or to suspend a session and others [46]. Besides, pre-fetching as a sub aspect of complementary technique is the most suitable combination for WC [8,10,39]. Furthermore, a practical aspect in WC is building a cache friendly website to be implemented in mobile devices. The last aspect involves measuring performance as benchmarks to evaluate the WC solution for the problem to be tackled.

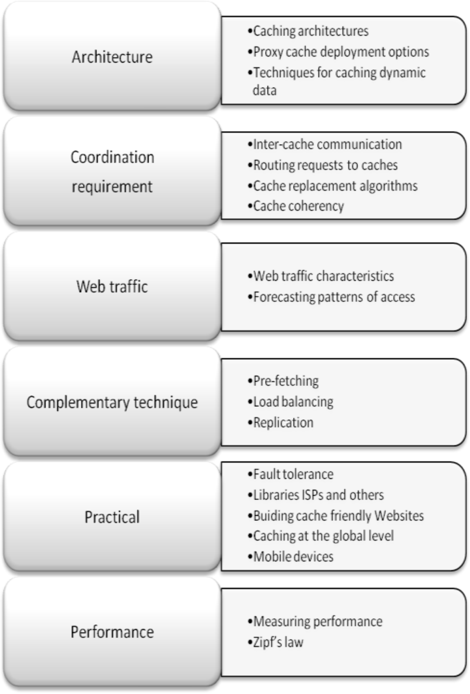
Architecture
•Caching architectures
•Proxy cache deployment options
•Techniques for caching dynamic data
Coordination requirement
•Inter-cache communication
•Routing requests to caches
•Cache replacement algorithms
•Cache coherency
Web traffic
•Web traffic characteristics
•Forecasting patterns of access
Complementary technique
•Pre-fetching
•Load balancing
•Replication
Practical
•Fault tolerance
•Libraries ISPs and others
•Buiding cache friendly Websites
•Caching at the global level
•Mobile devices
Performance
•Measuring performance
•Zipf's law

Fig. 2. Sub aspects of Web caching aspect

III. WEB CACHING ARCHITECTURES OR DEPLOYMENT SCHEMES

Caching operation can be executed at the client application, and generally it is embedded in most Web browsers [44,47,48,49,50]. There are a number of products that extend or replace the embedded caches with systems that contain larger storage, more features, or better performance. In any cases, these systems only cache net objects from many servers for a single user [2,51]. Caching can also be operated between the client and the server as a part of proxy cache [52,53,54], which is often located close to network gateways to decrease the bandwidth connections. These systems can serve many users (clients) with cached objects from many servers. In fact, according to Davison [51] the usefulness of WC had been reported to be up to 80% for some installations based on Web cache objects requested by one client for later retrieval by another client. Even for better performance, many proxy caches are part of cache hierarchies [55]; a cache can appeal to neighbouring caches for a requested document to lessen the need for direct fetching.

Caching can also occur at other levels for example the user's hard disk, and servers located in the institution in which the user is employed, the institution's Internet Service Provider (ISP), the regional Internet hub, the national Internet hub or at the global level. Web browsers accomplish caching by specialised caches known as proxy caches and by Web servers (see Figure 3). Many popular Web browsers cache the Web pages browsed by the user. Very often such browsers enable the users to view the content downloaded earlier, by pressing the back button. In this case, the Web page is fetched from the browser's cache instead of fetching it again from the original source on the Web, thereby avoiding unnecessary downloads.

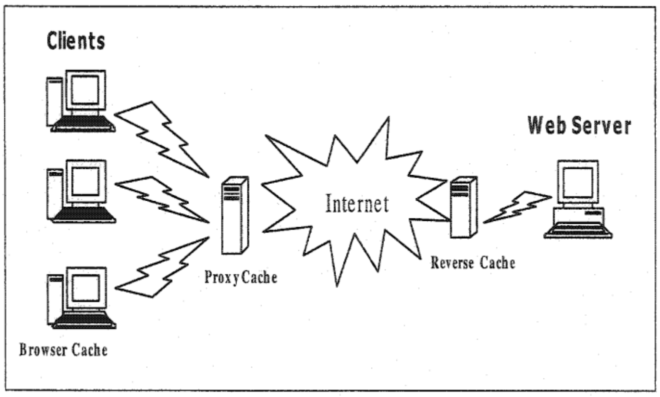

Fig. 3.Web caching and pre-fetching can be implemented at three cache levels; on the client side, at the proxy server and the website [56]

Features of these three kinds of Web cache can be generalised as the followings [16]:

In the client: Web caches can be built into most Web browsers. The first level cache for Web users is generally implemented within a browser. Because this cache only uses some of the main memory, or a small disk space for storage, the size of a browser's cache is small. However, since the browser cache is the closest cache to the end user, a short response time is provided to the user if the requested objects are cached. Since this kind of cache is embedded into the browser, the benefits of having cache objects cannot be shared by other users.

Between the client and the server: A second level cache is usually provided within proxy cache using a local hard disk on the gateway server for storage and its position is always near network gateways for bandwidth reduction on dedicated and expensive Internet connection. This kind of cache can be used to store a significant number of files from many Web servers and, as an additional benefit, permits many users to share the resource from the nearest proxy servers or their neighbouring caches. As a result, this type of cache leads to wide area bandwidth savings, improved latency, and increases the availability of the static Web documents.

Near the servers: A reverse cache or inverse cache is directly positioned in front of a specific Web server. In contrast to a general proxy cache, the reverse cache only handles Web documents from one Web server. This is an attractive solution to reduce the workload of a busy Web server by caching its static documents so that the original server can be dedicated to providing service through generating dynamic pages.

Browser and proxy caching, active Web caching, adaptive Web caching and push caching are the common caching types. However, other current types of intelligent caching are smart intelligent Web caching [57], mobile environment for intelligent genetic search and proxy caching [37], hybrid cache-index forwarding for mobile WWW [35] and adaptive Web caching in peer-to-peer network [58].

A. Browser and Proxy Caching

Each contemporary browser, such as Firefox and Explorer, provides the preferences dialog and one of the features is to set a '*cache*'. If users use this feature, it means that they allow storing browsed objects in their computer hard disk. This browser cache is executed based on basic rules to check once per session either the objects are fresh or not [59,60]. This utility is valuable when the client clicks the '*back*' button to browse to a previously browsed page. In this case, the same navigation objects for example images will be served directly from the browser cache.

In addition, Web proxy cache or so-called forward proxy caching applies the similar procedure even for a huge scale consisting of thousands users. Normally, large corporations and ISP's set up their proxies on their firewalls to reduce latency and traffic. This is because popular objects are usually requested only once [59,60,61] and are provided to a bulky number of clients. Most of the time, these large companies or ISPs employ proxy caches to decrease Internet bandwidth usage. This is because a large number of users share the cache objects from the same proxy caches, which are requested from different clients named as '*shared hits*'. Usually, a common hit rates for efficiency is 50% and above [62,63,8] for any proxy caches. Moreover, the deployment of caches include forward proxy caching, reverse proxy caching and transparent caching [64]. Figure 4 shows a sample of proxy caching architecture that involves video server, proxy and client.

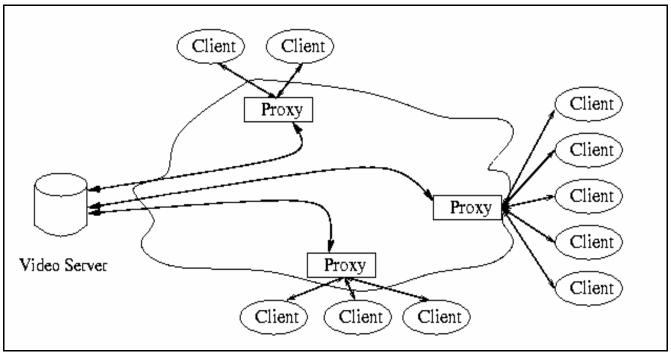

Fig. 4. Proxy caching

B. Active Web Caching

Generally, a proposed active Web caching is to concern on caching dynamic Web objects [65]. This type of caching is practical compared to traditional caching strategies because this strategy can cache the dynamic data. Moreover, users prefer to use placing cookies to personalise Web pages in their workstations. The personalisation of Web document using cookies through HTTP header elements to signal the personalised content services becomes a trend and need for the users. Besides, a large ISP trace revealed that client requests carrying cookies are more than 30% [66].

Cao *et al.* [65] suggested Java applets usage when a user requests a dynamic Web page to allow servers to transfer a part of its effort to a proxy cache. These applets will make possible customisation of uncacheable Web objects and it is practical while a user hits to a Web object with no more interaction to the server but only to the proxy cache [67]. Consequently, the server gives the requested Web objects and relevant cache applets for the first time of request personalisation. In the next requests for the similar Web document, the cache applets process the task without referring to the origin server. Therefore, active cache can be a realistic scheme, which can help bandwidth usage reduction [65,67].

Besides, a proposed active query caching for database Web servers is needed to solve a problem on uncacheable data from dynamic content providers too [68]. The proposed solution can be viewed in Figure 5. This scheme could be applied at query level that covers an individual user request. According to Luo *et al.* [68] the query level caching is more efficient and feasible than table level caching.

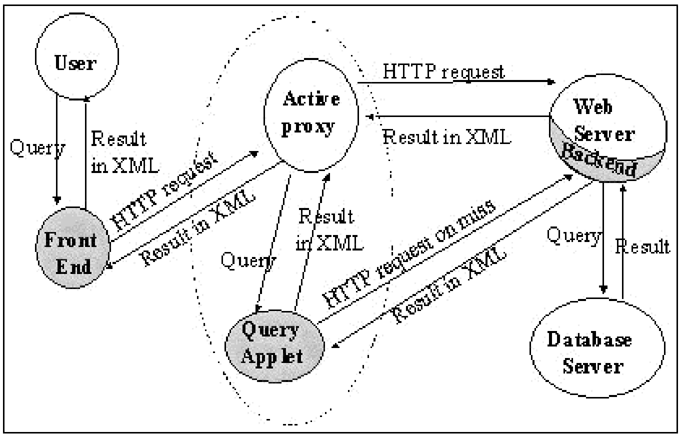

Fig. 5.System architecture of an active query caching [68]

C. Adaptive Web Caching

Adaptive caching [69,70] views the caching problems as a way of optimising global data dissemination. The key problem, which adaptive caching is targeted on is the "hot spot" phenomenon, where various, short-lived Internet content could get a sudden surge of requests and become massively needed and highly popular.

Adaptive caching consists of multiple, distributed caches, which dynamically join and leave cache groups (referred to as *cache meshes*) based on content demand [58]. Adaptability and the self-organising property of meshes is a response to scenarios, including in cases where demand for objects gradually evolves and demands are unpredictably high or low (spike). Cache Group Management Protocol (CGMP) and Content Routing Protocol (CRP) are employed in adaptive caching [71]. CGMP specifies how meshes are formed and how individual cache joins and leaves those meshes. CRP is used to locate cache content from within the existing meshes. It is assumed that in this caching approach, the deployment of cache clusters across administrative boundaries is not an issue. Figures 6 and 7 depict two examples of adaptive WC architecture.

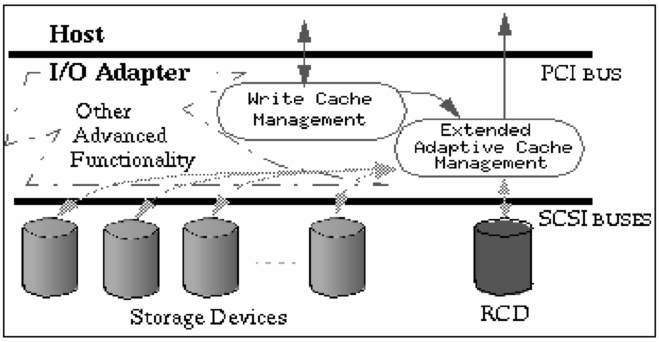

Fig. 6. Adaptive Web caching [69]

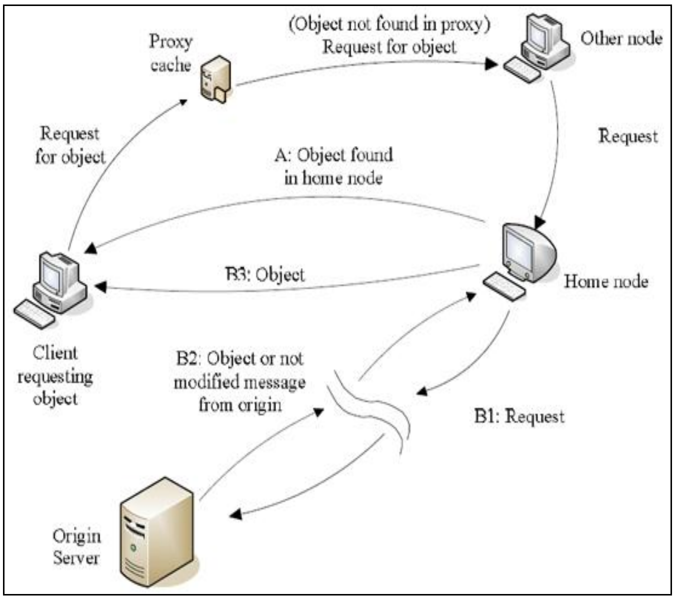

Fig. 7. Adaptive Web caching in Peer-to-Peer Network [58]

Figure 7 shows "*a typical home-node approach in the Squirrel caching technique: When the requested object is not found in the local cache, the request is passed to the home node. If the object is there, it is dispatched to the requesting client (A). Otherwise, the request is passed to the origin server (B1), and the origin server replies with a not modified message, or the object to the home-node (B2). Then the home-node sends the object to the requesting client*" [58].

D. Push Caching

As described by Gwertzman and Seltzer [72], the key idea behind push caching is to keep cached data close to those clients requesting that information. Cached data is dynamically mirrored as the originating server identifies where the requests came from. For instance, if traffic to a Kuala Lumpur based site started to rise because of increasing requests from the Johor based, the Kuala Lumpur site would response initiating the Johor based cache.

As with adaptive caching, one of the main assumptions of push caching is the ability to launch caches, which may cross administrate boundaries. However, push caching is targeted mostly at content providers, which will most likely control the potential sites to which the caches could be deployed [56,73]. Unlike adaptive caching, it does not attempt to provide a general solution to improve content access for all types of content, from all providers [8].

E. Intelligent Web Caching

Mohamed [57] proposed an intelligent WC 3-layer general design in his research. Figure 8 illustrates the architecture design of intelligent WC. In the intelligent WC 3-layer design, the ASP pages reside within the presentation layer where the clients will be interacting directly to the ASP pages without noticing the object engine, as they will only view the pages created from the ASP pages in the IIS server. All the data from the client will be submitted via the ASP pages and passed on to the object engine.

In the application layer, the object engine will be interacting with the ASP pages and the WC module. The Intelligent WC module will handle the data caching from the database server and will receive data request from the object engine. The data layer is where the database of the application resides. By using this structure, any modification within the database, application or Web presentation are possible, time independent from each other as long as the interface between layers is maintained [57]. Viewing the system as a 3-layer application design enables observation of the data communication, network connection, and separation in the application technology used between the layers in the system. On the other hand, the system is also viewable from the data flow perspective, which enables the development of the system prototype to be done by modules implementation.

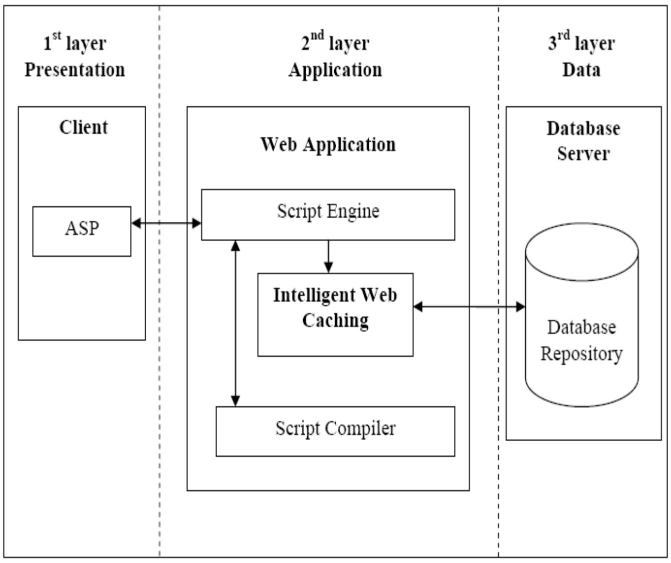

Fig. 8. Intelligent Web caching 3-layer design [57]

F. Mobile Environment for Intelligent Genetic Search and Proxy Caching

Cvetkovic *et al.* [37] suggested an implementation of Java distributed object application in experimenting with genetic search and proxy caching algorithms for Internet. With the intention to reduce the needed transfer overhead and to accelerate the search, in this project the existing static agent system is upgraded by turning it into a system based on mobile agents. Instead of fetching all the documents over the net, mobile agents are sent to the locations where these documents are, and perform evaluations over there. Another improvement is in the parallelism of execution. Mobile agents are sent to more than one location, to make evaluations in parallel.

Gain in time is enormous and corresponds to the lowering of the network traffic because significantly fewer amounts of data are transferred in the case of mobile implementation. Block diagram of their package in the mobile implementation is shown in Figure 9 (Continuous lines show the flow of data, and dashed lines show the control flow. Rectangles represent the applications and ovals the input and output data structures).

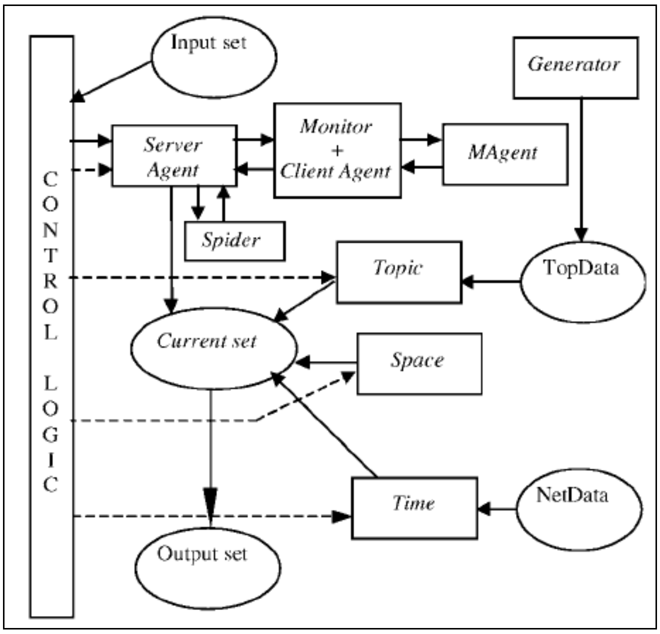

Fig. 9. Block diagram of the mobile implementation of the project [37]

G. Hybrid Cache-index Forwarding for Mobile WWW

Ahn and Han [35] recommended a cache-index forwarding that can be used to let the base station know the caching information on the mobile hosts. They proposed a hybrid cache-index forwarding scheme (integration between MowgliWWW and CINDEX architectures), which sends the caching data information on mobile hosts per document to the base station. Their scheme also transfers all of the cache-index data from the old base station to the new one during the handover phase.

The MowgliWWW and CINDEX are designed to increase the usability, reliability and efficiency of client-server communication between mobile hosts and fixed hosts. The basic idea is to split a channel with end-to-end control into two parts using a store-and forward type interceptor. This interceptor allows them to replace the client server paradigm with the client-mediator-server paradigm. As shown in Figure 10, this common architecture is composed of a Web browser, HTTP Agent, HTTP Proxy and Web server.

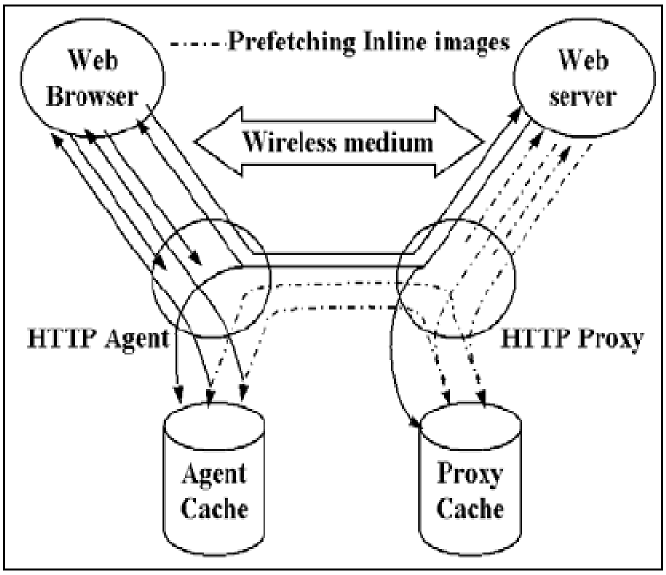

Fig. 10. Common architecture [35]

The basic operation of these components is as follows: When a WWW browser requests a particular document, the agent first checks the document's Uniform Resource Locator (URL) against the cache index. If the document is present, the agent simply returns it to the client. If the document is missing, the agent forwards a request to the proxy and creates a new entry in the cache index. Eventually, the proxy will respond to the request with either an error response or a document response. As soon as the agent receives the response header, it starts sending an arriving response to the waiting client. In addition, a document response is stored in the agent cache. From this moment on, any new HTTP request for this document object is served from the cache.

While MowgliWWW is simple and easy to implement and provides a higher cache hit ratio, it may cause unexpected and serious wireless network delays due to the overhead of cache-index transmission and management. While CINDEX obtains efficient cache-index transmission and management, it may produce high latency due to its lower cache hit ratio. Considering these advantages and disadvantages, they proposed a hybrid cache-index forwarding method for better performance, which supports for high mobility of mobile hosts and provides a high cache hit ratio. In their cache-index forwarding scheme, cache index forwarding is conducted in both the mobile host and old base station. A mobile host forwards only a cache-index per requested document such as CINDEX. Figure 11 shows a graphical representation of the proposed scheme.

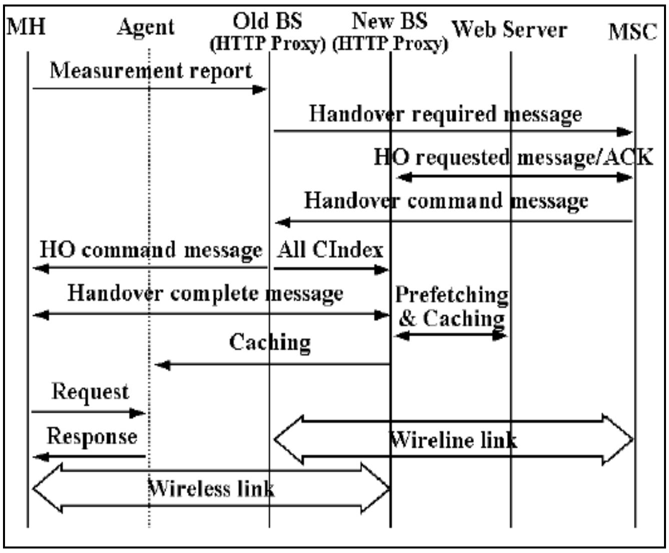

Fig. 11. Message flows in a hybrid cache-index forwarding [35]

IV. SUMMARY AND CONCLUSION

Table 1 describes the advantages and disadvantages of different WC architectures. This summary reports the existing architectures of Web caching are affected either in the client, between the client and the server or near the servers. This is because we have to choose the right architecture or deployment scheme of WC for the right implementation to ensure the proposed architecture will enhance the performance of Web documents.

TABLE 1. SUMMARY OF WEB CACHING ARCHITECTURES

Architecture	Description	Advantage	Disadvantage
Proxy (known as forward proxy caching) [59,60,61]	Deployed at the edge of the network	Easy to deploy	Single point of failure
- Reverse proxy caching	Deployed near origin	Server farm management	Single point of failure
- Transparent proxy caching	Intercepting HTTP request	Eliminate single point of failure	Violates end-to-end statement
Active Web caching [65,67,68]	Applets; Caching for dynamic documents	Caching dynamic documents and personalized cache	Issues of privacy
Adaptive Web caching [58,69,70]	Optimising global data distribution. Consists of multiple, distributed caches which dynamically join and leave cache groups; CGMP,CRP	Tackling “hotspot” phenomenon	Assumption: Deployment of cache clusters across administrative boundaries is not an issue
Push caching [72]	To keep cached data close to those clients requesting that information (concept of mirror site)	Targeted providers	Assumption: Ability to launch caches which may cross administrative boundaries
Smart intelligent Web caching [57]	Applying neural network and network analysis	Adaptable to environment	High computational complexity and applied on 3-tier design
Mobile environment for intelligent genetic search and proxy caching [37]	Implementation of mobile agents with genetic search and proxy caching algorithms	Efficient search for group of people sharing interests in some subject	Spend a lot of time for fetching documents from the Internet onto the local disk
Hybrid cache-index forwarding for mobile WWW [35]	Hybrid MowgliWWW and CINDEX schemes to send the caching data information on mobile hosts per document to the base station and transfers all of the cache-index data from the old base station to the new one during the handover phase	Supports for high mobility of mobile hosts and provides a high cache hit ratio	Does not suffer from wireless network delays, because the cache-index transmission is performed not by the mobile host but by the base station

V. FUTURE WORKS

Due to the limitations of this review paper, possible future works are recommended for further exploration as follows:

(i) Involving other mobile events classification

This research can be extended with mobility and event environment classification study [2]. The mobility event will monitor the current position, speed, and direction as three key parameters to represent the mobile host’s current mobility status. After that, the event environment will change accordingly to the operating environment by either disconnection or handoff to monitor for the mobile client.

(ii) Proposing Tolerant Rough Set Approximation (TRSA)

This proposed algorithm addresses the delays associated with dynamic page generation issues by capturing the behaviour of the data that does not have to be trained, but will be considered as a condition attribute in resolving WC performance especially for caching streaming multimedia over the Internet. The TRSA condition attributes are knowledge representation for script identifications and with the existence of many condition attributes, better decisions for script caching identification can be optimised.

(iii) Merging Social Networking Concept in Web Pre-caching with Cloud Computing

These proposed two solutions above will increase the capability of Web pre-caching to merge business and education technologies with this combined solution. Instead of finding similarity characteristics for Web objects’ visualisation, companies and higher learning institutions are developing interactive communities that connect individuals based on shared business and education needs or experiences in the same proxy cache. This proposed solution might reduce the budget and increase the productivity of companies and higher learning institution.

(iv) Database Summarisation for Mobile Web Pre-fetching

Mobile database summarisation is a process of reduction of the size and information capacity of a database while maximising the usability of the resultant (summarised) dataset. Besides, the data model for the database is constructed in order to generate a decision table for classification and decision rules for prediction with soft computing approaches [76,77] and then to summarise a mobile database that will be used to pre-fetch Web contents.

(v) Pre-fetching based Cooperative Caching in Mobile Adhoc Networks (MANETs)

MANETs with pre-fetching as the integrated part of data caching will sense the future needs of mobile nodes (MNs)[75]. This proposed solution can reduce the query latency and improve the data availability. Moreover, other areas such as m-forensic, m-commerce and other domains related to cache management in mobile computing environments could implement MANETs.

ACKNOWLEDGMENT

This research is supported by The Ministry of Higher Education (MOHE) and collaboration with Research Management Centre (RMC), Universiti Teknologi Malaysia. This paper is financial supported by University Grant (GUP) Program, Tier 1 (vote: Q.J130000.2528.03H00). The main author would like to express her deepest gratitude to Assoc. Prof. Dr. Shahida and Dr. Mazura who have contributed in the writing phase.

REFERENCES

[1] Lu, J., Ruan D. and Zhang G. (2007). E-Service Intelligence: An Introduction. Studies in Computational Intelligence (SCI) 37, Heidelberg, Berlin: Springer-Verlag, 1-33.

[2] Davison, B. D. (2002). The Design and Evaluation of Web Prefetching and Caching Techniques. Doctor of Philosophy thesis, Graduate School of New Brunswick Rutgers, The State University of New Jersey, United State.

[3] Wu, S., Chang, C., Ho, S. and Chao, H.(2008). Rule-based intelligent adaptation in mobile information systems. Expert Syst. Appl. 34(2), 1078-1092.

[4] Zhang, D. (2007). Web content adaptation for mobile handheld devices. Journal Commun. ACM. 50(2), 75-79.

[5] Said, E.G, Omar, E.B. and Robert,R. (2009). Data Prefetching Algorithm in Mobile Environments. European Journal of Scientific Research. 28(3), 478-491.

[6] Lim, B. and Kim, J. (2010). Page flip contents caching method using user request on digital media server. 2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). 18-20 October. Moscow, Russia: IEEE, 115-121.

[7] Dzolkhifli, Z., Ibrahim H., and Affendey, L.S. and Madiraju P. (2009). A Framework for Caching Relevant Data Items for Checking Integrity Constraints of Mobile Database. International Journal of Interactive Mobile Technologies, Technical Basics. 3(2), 18-23.

[8] Nagaraj, S. V. (2004). Web Caching and Its Applications. Boston/Dordrecht/London: Kluwer Academic Publishers.

[9] Acharjee, U. (2006). Personalized and Intelligence Web Caching and Prefetching. Master thesis, Faculty of Graduate and Postdoctoral Studies, University of Ottawa, Canada.

[10] Garg, A. (2003). Reduction of Latency in the Web Using Prefetching and Caching. Doctor of Philosophy thesis, University of California, Los Angeles, United State.

[11] Curran, K. and Duffy, C. (2005). Understanding and Reducing Web Delays. Int. J. Network Mgmt. 15, 89-102.

[12] Saiedian, M. and Naeem, M. (2001). Understanding and Reducing Web Delays. IEEE Computer Journal. 34(12).

[13] Kazi, T.H., Feng, W. and Hu, G. (2010). Web Object Prefetching: Approaches and a New Algorithm. 2010 11th ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD). 9-11 June. 115-120.

[14] Nigam, B. and Jain, S. (2010). Analysis of Markov model on different web Prefetching and caching schemes. 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC).28-29 December. Coimbatore, India: IEEE, 1-6.

[15] Kroeger,T. M., Long, D. E. and Mogul, J. C. (1997). Exploring The Bounds of Web Latency Reduction from Caching and Prefetching. Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems (USITS'97). 8-11 December. USENIX Association, Berkeley, CA, USA. 13-22.

[16] Wang, Y. (2003). A Hybrid Markov Prediction Model for Web Prefetching. Master thesis, Department of Electrical and Computer Engineering, University of Calgary, Alberta.

[17] Ahmad, N., Malik, O, ul Hassan, M, Qureshi, M. S. and Munir, A. (2011). Reducing user latency in web prefetching using integrated techniques. 2011 International Conference on Computer Networks and Information Technology (ICCNIT). 11-13 July. Abbottabad, Pakistan: IEEE, 175-178.

[18] Hua, C. (2007). Frequent Query Patterns Guided XML Caching and Materialization, International Conference on Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007, 21-25 September. Shanghai, China: IEEE, 3673 – 3676.

[19] Song, T.-S., Lee, J.-S., Choy, Y.-C. and Lim, S.-B. (2006). Mynews: Personalization of Web Contents Transcoding for Mobile Device Users. Advances in Artificial Reality and Tele-Existence. Z. Pan, A. Cheok, M. Halleret al, SpringerBerlin / Heidelberg. 4282: 512-521.

[20] Hong M., Ryu, D-Y., Sir, J-C., Kim, E-Y. and Lim,Y-H. (2004). Using a Transcode and Prefetch Method for Playing XML Contents Containing Multiple Multimedia Data on Mobile Terminals. EDBT 2004Workshops, LNCS 3268, 309–317.

[21] Saha, G. K. (2005). A Novel 3-tier XML schematic Approach for Web Page Translation. Ubiquity 2005. November (November 2005), New York, USA: ACM, 1-1.

[22] Marquez, J., Domenech, J., Gil, J. and Pont, A. (2008). Exploring the benefits of caching and prefetching in the mobile web. 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, Pretoria, South Africa.

[23] Jin, B., Tian, S., Lin, C., Ren, X. and Huang, Y. (2007). An Integrated Prefetching and Caching Scheme for Mobile Web Caching System. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. 30 July–2 August: IEEE, 522-527.

[24] Tadano, K., Kawato, M., Machida, F. and Maeno, Y. (2010). Resource Information Cache Update Control for Scalable Access Control Management Systems. 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD). 5-10 July. Miami, Florida :IEEE, 538-539.

[25] Jian, L. and Yan-Qing, W. (2011). Web log data mining based on association rule. 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). 26-28 July. Shanghai, China: IEEE, 3, 1855-1859.

[26] Padmanabhan, V. N. and Mogul, J. C. (1996). Using Predictive Prefetching to Improve World Wide Web Latency. SIGCOMM Computer Communications Review, ACM. 26(3), 22-36.

[27] Loon, T.S. and Bharghavan, V. (1997). Alleviating the latency and bandwidth problems in WWW browsing. Proceedings of the 1997 Usenix Symposium on Internet Technologies and Systems (USITS-97). December. Monterey, California: USENIX, 219-230.

[28] Kobayashi, H. and Yu, S-Z (2000). Performance Models of Web Caching and Prefetching for Wireless Internet Access. International Conference on Performance Evaluation: Theory, Techniques and Applications (PerETTA 2000). 21-22 September. University of Aizu, Fukushima, Japan.

[29] Komninos, A. and Dunlop, M.D. (2007). A calendar based Internet content pre-caching agent for small computing devices. Journal of Personal and Ubiquitous Computing, Springer-Verlag, London, UK. 12(7), 495-512.

[30] Mohamed, F. and Shamsuddin, S. M. (2005). Smart Web Cache Prefetching with MLP Network. Proceedings of The 1st IMT-GT Regional Conference on Mathematics, Statistics and their Applications. 555-567.

[31] Teng, W.-G., Chang, C.-Y. and Chen, M.-S. (2005). Integrating Web Caching and Web Prefetching in Client-Side Proxies. IEEE Transaction on Parallel and Distributed Systems. 16(5), 444-455.

[32] Yang, W. and Zhang, H. H. (2002). Integrating Web Prefetching and Caching Using Prediction Models. Journal of World Wide Web, Kluwer Academic Publishers. 4 (4), 299-321.

[33] Harding, M., Storz O., Davies, N. and Friday, A. (2009). Planning ahead: techniques for simplifying mobile service use, Proceedings of the 10th workshop on Mobile Computing Systems and Applications. February 23-24. Santa Cruz, California, 1-6.

[34] Ye, F., Li, Q. and Chen, E. (2008). Adaptive Caching with Heterogeneous Devices in Mobile Peer to Peer Network, SAC’08, 16-20 March. Fortaleza, Ceará, Brazil: ACM, 1897-1901.

[35] Ahn, K. H. and Han, K. J. (2003). A Hybrid Cache-Index Forwarding Scheme for Mobile WWW. Proceedings of CIC'02 Proceedings of the 7th CDMA international conference on Mobile communications. Seoul, Korea: Springer-Verlag, 461–469

[36] Cho, G. (2002). Using Predictive Prefetching to Improve Location Awareness of Mobile Information Service. Lecture Notes in Computer Science. 2331,1128-1136..

[37] Cvetkovic, D., Pesic, M., Petkovic, D., Milutinovic, V., Horvat, D., Kocovic, P. and Kovacevic, V. (2001). Architecture of the Mobile Environment for Intelligent Genetic Search and Proxy Caching. Telecommunication Systems. 18(1–3), 255–270.

[38] Santhanakrishnan, G, Amer, A. and Chrysanthis, P.K.(2005). Towards Universal Mobile Caching. Proceedings of the 4th ACM international workshop on Data engineering for wireless and mobile access. Baltimore, Maryland, USA: ACM, 73-80.

[39] Song, H. and Cao, G. (2005). Cache-Miss-Initiated Prefetch in Mobile Environments. Computer Communications. 28(7), 741-753.

[40] Zabian, A. and Qawasmeh, S. (2006). Web Caching in Mobile Environment. Proceedings of the first Mobile Computing and Wireless Communication International Conference, 2006. MCWC 2006. 17-20 September. Amman, Jordan: IEEE, 184-190.

[41] Gob, A., Schreiber, D., Hamdi, L., Aitenbichler, E. and Muhlhauser, M. (2009). Reducing User Perceived Latency with a Middleware for Mobile SOA Access. IEEE International Conference on Web Services, 2009, ICWS 2009. 6-10 July. Los Angeles, CA : IEEE, 366-373.

[42] Wong, A. K. Y. (2006). Web Cache Replacement Policies: A Pragmatic Approach. IEEE Network Magazine. 20(1), 28–34.

[43] Koskela, T., Heikkonen, J. and Kaski, K. (2003). Web Cache Optimization with Nonlinear Model Using Object Features. Computer Networks Journal, Elsevier. 43(6), 805-817.

[44] Ali, W. and Shamsuddin, S.M. (2011). Neuro-fuzzy system in partitioned client- side Web cache. Expert Systems with Applications. 38, 14715–14725.

[45] Sulaiman, S., Shamsuddin, S.M. and Abraham, A. (2009). Rough Web Caching, Rough Set Theory: A True Landmark in Data Analysis, Studies in Computational Intelligence, Springer Verlag, Germany, 187-211.

[46] Pitkow, J. E. (1999). Summary of WWW characterizations. World Wide Web. 2(1-2): 3-13.

[47] Tan,Y., Ji,Y. and Mookerjee, V.S.(2006). Analyzing Document-Duplication Effects on Policies for Browser and Proxy Caching. INFORMS Journal on Computing. 18(4), 506-522.

[48] Sieminski, A. (2005). Changeability of Web objects - browser perspective. Proceeding of the 5th International Conference on Intelligent Systems Design and Applications. 8-10 September: IEEE, 476- 481.

[49] Mookerjee, V. S. and Tan,Y.(2002). Analysis of a least recently used cache management policy for Web browsers. Operations Research. 50(2), 345-357.

[50] Reddy, M. and Fletcher, G.P. (1998a). An adaptive mechanism for Web browser cache management. IEEE Internet Computing. 2(1), IEEE Educational Activities Department Piscataway, NJ, USA: IEEE, 78-81.

[51] Davison, B. D. (2008). Web Caching Overview. Available at: http://www.Web-caching.com/welcome.html [Accessed March 15, 2008].

[52] Pallis, G., Vakali, A. and Pokorny, J. (2008). A Clustering-based Approach for Short-term Prefetching on a Web Cache Environment. Computers & Electrical Engineering Journal, Elsevier. 34(4), 309-323.

[53] Nair, A. S. and Jayasudha, J. S (2007). Dynamic Web pre-fetching technique for latency reduction. Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) - Volume 04. 13-15 December. Sivakasi, Tamil Nadu, India: IEEE, 202-206.

[54] Hung, S.-H. , Wu, C.-C. and Tu, C.-H. (2008). Optimizing the embedded caching and prefetching software on a network-attached storage system. Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous

Computing - Volume 01 (EUC '08).Washington, DC, USA: IEEE, 1, 152-161.

[55] Mahanti, A., Williamson, C. and Eager, D. (2000). Traffic analysis of a Web proxy caching hierarchy. IEEE Network. 14(3), May/June, 16-23.

[56] Wang, J. (1999). A Survey of Web Caching Schemes for the Internet. ACM Computer Communication Review. 25(9), 36-46.

[57] Mohamed, F. (2007). Intelligent Web Caching Architecture, Master, Universiti Teknologi Malaysia, Skudai, Malaysia.

[58] Tauhiduzzaman, M., Nizamee, M.R., Osmani, S., Khan, M.M. and Mahmood, A. (2008). Survey on adaptive caching techniques in peer-to-peer network. International Conference on Electrical and Computer Engineering, 2008. ICECE 2008. 20-22 December. Dhaka, India: IEEE, 501-505.

[59] Ali, W. and Shamsuddin, S. M. (2009a). Intelligent Client-side Web Caching Scheme Based on Least recently Used Algorithm and Neuro-Fuzzy System. The sixth International Symposium on Neural Networks (ISNN 2009). Lecture Notes in Computer Science (LNCS). (pp. 70–79). Berlin Heidelberg: Springer-Verlag.

[60] Ali, W. and Shamsuddin, S. M. (2009b). Integration of Least Recently Used Algorithm and Neuro-Fuzzy System into Client-side Web Caching. International Journal of Computer Science and Security. 3(1), 1-15.

[61] Chen, X. and Zhang, X. (2005). Coordinated data prefetching for web contents. Computer Communications. 28(17), 1947–1958.

[62] Bestavros, A. and Cunha, C. (1995). A Prefetching Protocol Using Client Speculation for The WWW. Tech. Rep. TR-95-011, Boston University, Department of Computer Science, Boston, MA 02215, April.

[63] Douglis, F., Feldmann, A., Krishnamurthy, B. and Mogul, J. (1997). Rate of Change and Other Metrics: A Live Study of the World-Wide Web. Proceedings of the 1997 USENIX Symposium on Internet Technologies and Systems (USITS-97): USENIX, 147-158.

[64] Web Caching (2008). Caching Tutorial for Web Authors. Available at: http://www.web-caching.com/mnot_tutorial/intro.html [Accessed April 10, 2008].

[65] Cao, P., Zhang, J. and Beach, K. (1999). Active Cache:Caching Dynamic Contents on The Web. Distributed Systems Engineering. 6(1), 43-50.

[66] Cáceres, R., Douglis, F., Feldmann, A., Glass, G. and Rabinovich, M. (1998). Web Proxy Caching: The Devil is in The Details. SIGMETRICS Perform. Eval. Rev. 26(3), 11-15.

[67] Labrinidis, A., Luo, Q., Xu, J. and Xue, W. (2010). Caching and materialization for Web databases. Foundations and Trends in Databases. 2(3), 169–266.

[68] Luo, Q., Naughton, J. F., Krishnamurthy, R., Cao, P. and Li, Y. (2001). Active Query Caching for Database Web Servers. Selected papers from the Third International Workshop WebDB 2000 on The World Wide Web and Databases: Springer-Verlag, 92-104.

[69] Zhang, G., Patuwo E. and Michael, Y. H. (1998). Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting. Elsevier Science Journal. 14(1), 35-62.

[70] Li, W.-S., Po, O., Hsiung, W.-P., Candan, K.S. and Agrawal, D. (2003). Freshness-driven adaptive caching for dynamic content. Proceedings of Eighth International Conference on Database Systems for Advanced Applications, 2003 (DASFAA 2003). 26-28 March, 203- 212.

[71] Almeida, J., Broder, A. Z., Cao, P. and Fan, L. (1998). A Scalable Wide-Area Web Cache Sharing Protocol. SIGCOMM98. October. Vancouver, British Columbia: ACM, 28(4).

[72] Gwertzman J. and Seltzer M. (1995). The Case for Geographical Push-Caching. Proceedings of the Fifth Annual Workshop on Hot Operating Systems, Hot OS-V: IEEE, 51-67.

[73] Podlipnig, S. and Boszormenyi, L. (2003). A Survey of Web Cache Replacement Strategies. ACM Computing Surveys, 35(4): 374-398.

[74] Yoo, J. A., Choi, I. S. and Lee, D. C. (2005). Prefetching Scheme Considering Mobile User’s Preference in Mobile Networks, Lecture Notes in Computer Science, 2005, Volume 3481, Computational Science and Its Applications – ICCSA 2005, 889-895.

[75] Chauhan, N. and Awasthi, L.K. (2012). Prefetching based Cooperative Caching in Mobile Adhoc Networks, International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE’2012). 24-25 March, Dubai, India: IEEE, 60-64.

[76] Sulaiman S., Shamsuddin S. M. and Abraham, A. (2012). Implementation of Social Network Analysis for Web Cache Content Mining Visualization, Computational Social Networks: Mining and Visualization, Series in Computer Communications and Networks, Springer Verlag, London, 345-376.

[77] Sulaiman S., Shamsuddin S. M., Ahmad, N.B. and Abraham, A. (2012). Meaningless to Meaningful Web Log Data for Generation of Web Pre-caching Decision Rules Using Rough Set , Forth Conference on Data Mining and Optimization (DMO 2012), 9-11 September, Langkawi, Malaysia: IEEE, 107-114.