

# WINDOWS WEB PROXY CACHING SIMULATION: A TOOL FOR SIMULATING WEB PROXY CACHING UNDER WINDOWS OPERATING SYSTEMS

Waheed Yasin, Hamidah Ibrahim,  
Nur Izura Udzir and Nor Asilah Wati Abdul Hamid

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Malaysia

Received 2013-11-22; Revised 2013-12-12; Accepted 2014-03-07

## ABSTRACT

Web caching plays a key role in improving the performance of the web by reducing the latency of delivering web objects to the users. On the other hand, a simulation tool plays a key role in studying the behavior of any network such as studying the effects of web caching on the performance of the network. The aim of this work is to present a tool for simulating a web proxy caching for windows operating systems since there is no existing well-compatible simulation tool for windows operating systems that can simulate Hit Ratio (HR) and Byte Hit Ratio (BHR) for traditional caching policies. The proposed simulation tool is called Windows Web Proxy Caching Simulation (WWPCS). The results show the performance of traditional web caching policies for different cache's sizes. Moreover, in order to show the efficiency of WWPCS, the results of running WWPCS are compared to the results of running a Unix-based tool.

**Keywords:** Simulation, Caching Policies, Web Proxy

## 1. INTRODUCTION

Web caching improves the web performance by storing web objects close to the clients, which reduces the latency of delivering web objects to the end-user. Moreover, it utilizes the bandwidth of the network which is considered as an important aim for network administrators. Furthermore, it reduces the loads on the origin servers (Jelenković and Radovanovic, 2009).

Web caching might be performed in three different stages that are browser stage, proxy stage and web server stage (Kumar and Norris, 2008) as illustrated in **Fig. 1**.

In our previous work (Yasin *et al.*, 2013), we have presented the affects of caching in browser stage on the performance of web objects delivery.

The aim of this study is to introduce a web proxy cache simulation tool called WWPCS which can be used in performance analysis of different web caching policies. Furthermore, since there is no existing of a well-compatible web cache simulation tool for windows

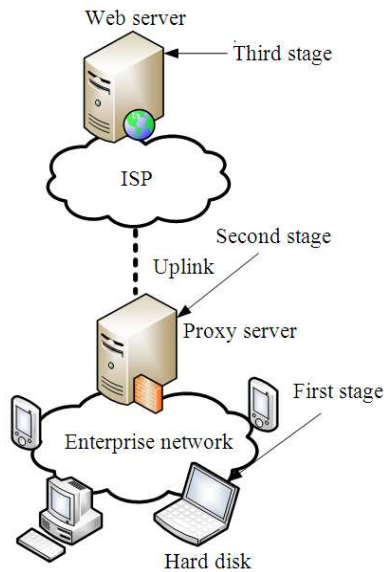
operating systems that can simulate the Hit Ratio (HR) and the Byte Hit Ratio (BHR). Thus, Microsoft tools are used to build WWPCS that make it more compatible since windows operating systems are Microsoft productions. Furthermore, in (Markatchev and Williamson, 2002) it has been recommended developing a caching simulation tool for windows operating systems.

This study is organized as follows: Section 2 presents the materials and methods. Section 3 presents the The results and discussion. Section 4 the limitations and implimintations. The related works are presented in setion 5. Section 6 presents the conclusions of this study. The acknowledgements are presented in section 7. Section 8 presents the references.

### 1.1. WEB Caching

Web caching is the process of storing data in an intermediate media which is called a cache and it is employed for responding to future queries rather thanfetching data from origin sources.

**Corresponding Author:** Waheed Yasin, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Malaysia



**Fig. 1.** Caching stages in computer networks

A cache hit shows that the system responds to the query from the cache; while a cache miss shows that the cache does not have the required data.

Many benefits can be achieved from applying caching such as reducing the load of the system and the latency. Moreover, it utilizes the network bandwidth.

In this section, cache replacement factors are presented in addition to some web caching policies.

### 1.2. Cache Replacement Factors

A web caching policy uses a replacement mechanism which refers to the process that occurs when the cache becomes full and there is not enough space for new items, which leads to the task of removing old items to make spaces for new ones. Thus, replacement mechanism has to decide which items are needed to be stored and which items have to be removed.

Many factors should be considered when the cache has to take a replacement decision (ElAarag, 2013). Some of these factors are listed below:

- Recency: Time since the last using of a web object
- Frequency: Total number of requests of a web object
- Cost of fetching a web object: Cost to fetch a web object from the original source, including processing, bandwidth and other resources
- Modification time: Time since the last modification.
- Expiration time: Time as soon as the web object becomes useless and it is time to be replaced

### 1.3. Web Caching Policies

In this work, some of the web caching policies are simulated that are listed below:

- First In First Out (FIFO). A simple mechanism to perform data replacement might be based on replacing the first coming web item by the new one
- Least Recently Used (LRU). The basic approach of LRU is to first remove the web objects that are least recently used from the storage of the cache. Thus, LRU has to track these web objects by implementing a recency index for each object, which indicates the time since the last use of the item. The least used one will be replaced with a new coming object. LRU is considered as one of the common mechanisms that are used in data replacement. Many variants of LRU have been proposed in the literature such as SVM-LRU (Podlipnig and Boszormenyi, 2003) and NBLRU (Ali *et al.*, 2012)
- Least Frequently Used (LFU). The basic approach of LFU is to first remove the web objects that are least frequently used from the cache. Thus, LFU has to track these web items by implementing a frequency index for each object, which indicates the total number of requests. A web object with the least index value will be replaced with a new coming item. Many variants of LFU have been proposed in the literature such as LFU-DA (ElAarag, 2013) and LFU-Aging (Podlipnig and Boszormenyi, 2003)
- Random (RAND). The basic approach of RAND is to employ a random function in order to choose victim objects in the cache that have to be replaced with new coming ones. Thus, there is no need to implement any indices to objects
- Greedy Dual Size (GD-Size). It has been proposed to consider the differences of objects' sizes that have different costs to fetch them from origin sources. GD-size uses an index which is defined as the cost of fetching the object to the size of that object in bytes. The object with the minimum index value is replaced (Cao and Irani, 1997)

## 2. MATERIALS AND METHODS

Markatchev and Williamson (2002) a unix-based tool called WebTraff has been presented for simulating web proxy caching. On the other hand, the authors have recommended developing the tool under windows operating systems to enable wider usage. In this study, WebTraff is called Unix-WebTraff-WPCS.

Unix-WebTraff-WPCS consists of three parts that are: Web Workload Generation, Web Workload Analysis and Web Proxy Cache Simulation.

In this study, the Web Proxy Cache Simulation which is the last part of Unix-WebTraff-WPCS is developed under windows operating system.

A laptop with the characteristics that are shown in **Table 1** has been used in this study. Furthermore, the following tools are used:

- Microsoft Visual Studio 2010 Express (Version 10.0.30319.1 RTMRel)
- Microsoft .NET Framework (Version 4.0.30319 RTMRel)
- Microsoft Office Access 2007 (Version 12.0.4518.1014 MSO 12.0.4518.1014)
- WinZip version 16 Pro(9691)-32-bit
- Notepad (Version 6.1)
- GNU PLOT (Version 4.6)

There are several steps that have to be performed before the simulation is launched. The steps are presented in this section.

### 2.1. Raw Data Collection

First, data for the proxy logs files and traces have been obtained from several proxy servers of the IRCache network for the requested web objects. More details about IRCache network are available on <http://www.ircache.net>. Each entry of a proxy log file contains ten fields that are listed below:

- Timestamp which is the time when the client’s socket is closed
- Elapsed time or Cost which is the time between the accepting and closing of the client socket. For HTTP streams, it is defined as the time between reading the request’s first byte and writing the reply’s last byte
- Client IP address
- Log tag and HTTP code that describe the treatment of the request in addition to the HTTP status code in the first line of the HTTP header
- Size which refers to the total bytes sent to the client.
- HTTP request method
- The requested URL
- User identification which is ‘-’ for all logs
- Hierarchy data and hostname that describe how and where the requested object has been fetched
- Content type

**Table 1.** Laptop characteristics

Description	Item
TOSHIBA	System manufacturer
Windows 7 ultimate	Operating system
Intel(R) Core(TM) Duo CPU T850 @ 2.16 GHz 2.17 GHz	Processor
2.00 GB	Memory (RAM)
32-bit operating system	System type

**Table 2** shows an example of the IRCache proxy logs file.

### 2.2. Data Pre-Processing

The second step is called pre-processing where the inappropriate and invalid logs such as un-cacheable requests and entries with unsuccessful HTTP status codes are removed from the proxy log files. Also, the unnecessary fields are ignored.

On the other hand, in order to reduce the simulation time, each URL is replaced with an integer identifier that is called “URL ID”.

**Figure 2** illustrates a snapshot of the WWPCS tool; while **Fig. 3** illustrates a flowchart of the WWPCS tool.

The WWPCS considers only two fields that are URL ID and size. The log file was collected from bo2 proxy server on the 18th March 2013. An example of log entries after data pre-processing step is shown in **Table 3**.

### 2.3. Simulation

As we mentioned earlier WWPCS is a trace-driven simulator which is build using Microsoft Visual C++ 2010 Express for evaluating web caching policies that are FIFO, LRU, LFU, RAND and GD-Size. WWPCS uses the revised proxy log file as an input and generates files that contain the HR and BHR as outputs. In this section, an explanation of the WWPCS tool and how it works are presented.

First, the raw data is imported by a database management system which is Microsoft Access 2007. Then, the raw data is pre-processed as mentioned earlier. After pre-processing the generated trace file is exported to a text file.

WWPCS uses the generated trace text file as an input. On the other hand, in order to run the simulation, there are some parameters have to be set before the simulation run that are the maximum cache size (Infinite Cache) in GB and a warm up (Warmup) parameter in MB. Also, the web cache policy has to be determined. WWPCS starts with 1 MB and increases according to a base 2 logarithmic scale.

WWPCS has a button called “Simulate” which is used for simulating the selected caching policy. It generates output files which are used to calculate the HR and the BHR for the caching policy.

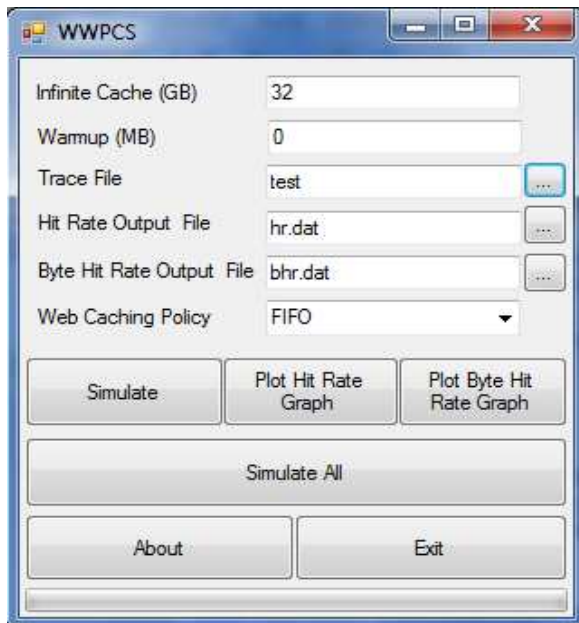
Furthermore, the “Plot Hit Rate Graph” and “Plot Byte Hit Rate Graph” buttons are used to plot the HR and the BHR graphs for the selected caching policy respectively. GNU PLOT (Version 4.6) is used by WWPCS to plot the graphs showing the performance metrics of web caching policies.

**Table 2.** An example of the IRCache proxy logs file

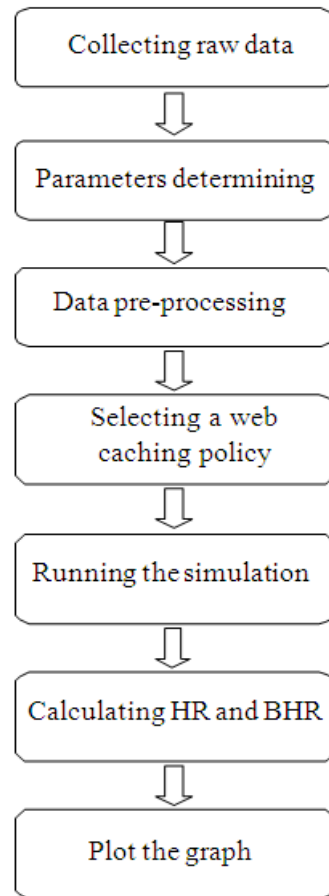
Timestamp	Hierarchy elapsed time (ms)	Client address	Log tag and HTTP code	Size (byte)	Method request	URL	User ID	data and hostname	Content type
1.363E+09	85	118.205.12.90	TCP_CLIENT_HIT/200	426	GET	http://88.208.61.79/key=cJi.LBVa9tE,end=1362774016/refrag=5412166/buffer=1M/speed=123657/2/sp/4/441749.flv?[3Ee3bA7hNXZYxJ17mVGXKL]	-	DIRECT/199.117.103.80	text/plain
1.363E+09	49	118.205.12.90	TCP_HIT/504	4574	GET	http://88.208.61.79/key=cJi.LBVa9tE,end=1362774016/refrag=5412166/buffer=1M/speed=123657/2/sp/4/441749.flv?[3Ee3bA7hNXZYxJ17mVGXKL]	-	DIRECT/200.147.68.8	application/zip
1.363E+09	75	118.205.12.90	TCP_CLIENT_REFRESH_MISS/200	7329	GET	http://88.208.61.79/key=cJi.LBVa9tE,end=1362774016/refrag=5412166/buffer=1M/speed=123657/2/sp/4/441749.flv?[3Ee3bA7hNXZYxJ17mVGXKL]	-	DIRECT/199.117.103.80	application/zip
1.363E+09	354	118.205.12.90	TCP_CLIENT_REFRESH_MISS/200	2614	GET	http://ads.globo.com/RealMedia/ads/adstream_lx.ads/maisvoce/oprograma/materias/L27/16439651643965407/Frame1/globocom/prod_prod_001_navegg_120801/	-	DIRECT/200.147.68.8	image/jpeg

**Table 3.** An example of log entries after data pre-processing

Timestamp	URL ID	Size (Byte)
1362705584.099	1	426
1362705584.915	2	4574
1362705584.998	3	7329
1362705585.16	4	3960
1362705585.16	6	2614
1362705585.208	5	3600
1362705585.247	4	3960
1362706447.423	6	2614
1362706447.438	1	426
1362706448.057	3	4574



**Fig. 2.** A snapshot of WWPCS tool



**Fig. 3.** A flowchart of the WWPCS tool

The “simulate all” button is used for running the simulation for all the web caching policies that are listed

in the combo box. Also, it is used to plot the HR and the BHR graphs for all web caching policies.

The “about” and “exit” buttons are used to show information about WWPCS and close the simulation tool respectively.

On the other hand, there are some parameters have to be set before the simulation run that are the maximum cache size (infinite cache) in GB and a warm up (warmup) parameter in MB. Also, the web cache policy has to be determined. WWPCS starts with 1 MB and increases according to a base 2 logarithmic scale. Moreover, WWPCS is able to plot graphs showing the performance metrics of web caching policies.

Before running the simulation, the maximum cache size (infinite cache) is set randomly to 32 GB, also the Warmup is set to 100 MB. The simulation is carried out five times for web caching policies that are FIFO, LRU, LFU, RAND and GD-Size.

## 2.4. Performance Evaluation

The performance of web caching policy might be measured using many metrics. In this simulation, two main performance metrics are used which are the HR and the BHR because they are the most widely used metrics for evaluating the performance of Web proxy caching policies.

Equation 1 defines the HR which is the total number of requests satisfied by the cache divided by the total number of requests.

Equation 2 defines the BHR which is the total number of bytes found in the cache divided by the total number of bytes requested within an observation period. BHR measures how much bandwidth the cache has saved:

$$HR = \frac{\text{Total No.of hits}}{\text{Total No.of requests}} \quad (1)$$

$$BHR = \frac{\text{Total No.of bytes in the cache}}{\text{Total No.of requested bytes in an observation period}} \quad (2)$$

A comparison between running the WWPCS and WebTraff-WPCS is conducted in order to evaluate the performance of the proposed tool.

A Linux operating system which is Fedora 16 is installed on the laptop. Furthermore, the following tools are used:

- gcc (Version 4.6.3-2.fc16.i686)
- Tcl (Version 8.5.11-1.fc16.i686)
- Tk (Version 8.5.11-1.fc16.i686)

- Perl (Version 5.14.3-205.fc16.i686)
- Gnuplot (Version 4.4.3-4.fc16.i686)
- GPL Ghostscript (Version 9.05)

WebTraff-WPCS has a default test file which is used in the comparison. Furthermore, the infinite cache size parameter is set to 32 GB and warmup parameter is set to 0 MB. The simulation is carried out five times for web caching policies that are FIFO, LRU, LFU, RAND and GD-Size.

On the other hand, we run the simulation on data collected from BO2 proxy server, where the maximum cache size (Infinite Cache) is set randomly to 32 GB, also the warmup is set to 100 MB. The simulation is carried out five times ND and GD-Size.

## 3. RESULTS AND DISCUSSION

### 3.1 RESULTS

In this section, the results gathered from the comparison between the WWPCS and WebTraff-WPCS simulations are presented. Moreover, the results gathered from running the WWPCS on the data collected from BO2 proxy server are presented.

### 3.2. DISCUSSION

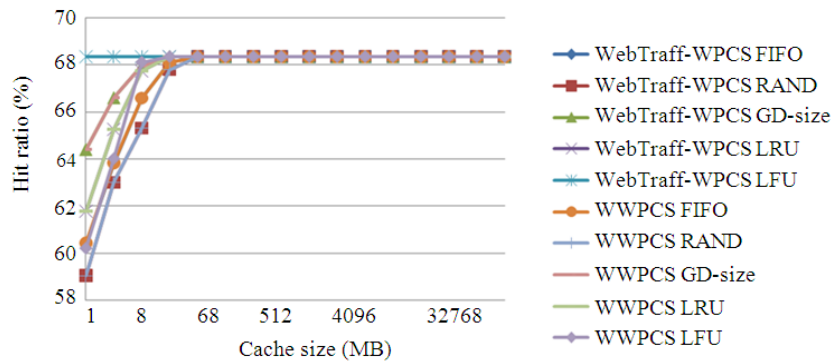
**Figure 4** illustrates the comparison between the HR of WWPCS and the HR of WebTraff-WPCS, while **Fig. 5** illustrates the comparison between the BHR of WWPCS and the BHR of the WebTraff-WPCS.

Referring to **Fig. 5**, it can be observed that the BHR of WWPCS and WebTraff-WPCS have quite similar performance for RAND, FIFO, GD-Size and LRU caching policies; however, the BHR of WebTraff-WPCS has a better performance than WWPCS for LFU caching policy.

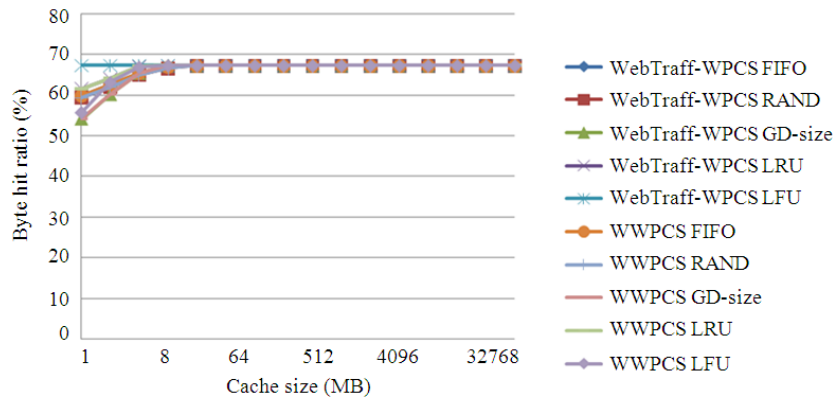
**Figure 6** shows the HR of simulated web caching policies, while **Fig. 7** shows the BHR of the simulated web caching policies.

## 4. LIMITATIONS AND IMPLICATIONS

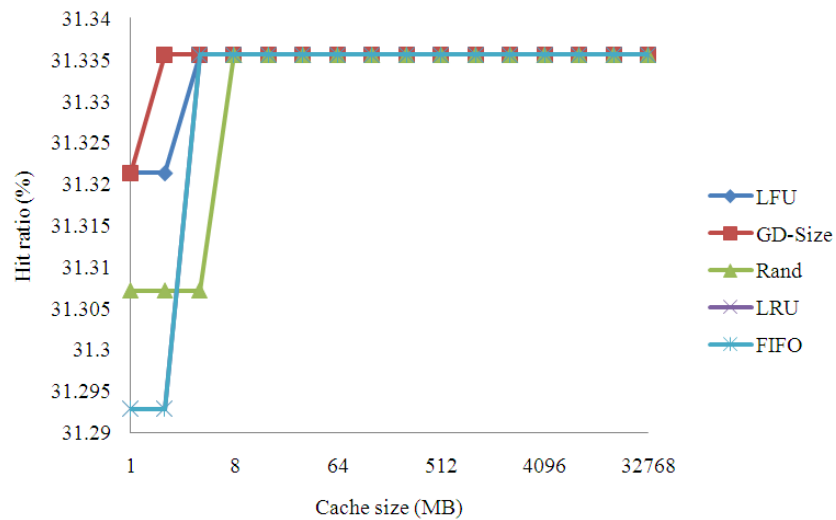
However WWPCS is tool for simulating traditional web proxy caching policies, it cannot be used for simulating web caching policies that are integrated with machine learning techniques. Moreover, log files that have a huge size such as 1 GB cannot be manipulated by WWPCS, which is considered as one of its limitations.



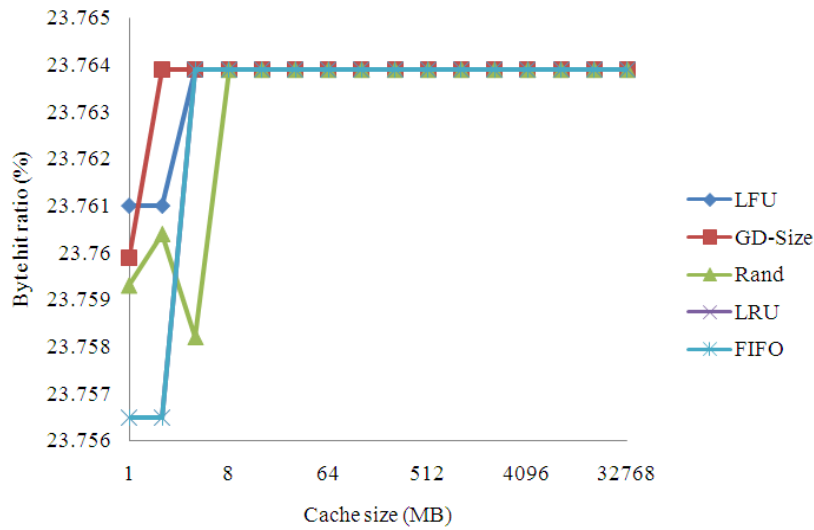
**Fig. 4.** A comparison between the HR of WWPCS and the HR of WebTraff-WPCS for simulated web caching policies, where infinite cache = 32 GB and warmup = 0 MB



**Fig. 5.** A comparison between the BHR of WWPCS and the HR of webtraff-WPCS for simulated web caching policies, where infinite cache = 32 GB and warmup = 0 MB



**Fig. 6.** The HR of web caching policies for BO2 proxy server, where infinite cache = 32 GB and warmup = 100 MB



**Fig. 7.** The BHR of web caching policies for BO2 proxy server, where infinite cache = 32 GB and Warmup = 100 MB

Unix-WebTraff-WPCS consists of three parts that are listed below:

- Web workload generation which is used for synthetic generation of Web proxy traces with controllable workload characteristics
- Web workload analysis which used for evaluating the sensitivity of web proxy cache performance to certain workload characteristics
- Web proxy cache simulation, which used for simulating traditional we proxy caching policies

On the other hand, WWPCS only consists of one part that related to web proxy caching. Thus, it is considered a limitation of WWPCS.

A limitation of a WWPCS is that it is only used in the second stage of the web caching stages which is called the Proxy stage.

On the other hand, WWPCS has many implications because it is a windows-based simulation tool which would make it widely used.

### 5. RELATED WORKS

Many researches have been conducted in the area of web proxy caching. In this section, recent related works are reviewed.

Cardenas *et al.* (2005) a framework for comparing the performance of web cache simulations has been presented. The proposed framework has been compared

to a commercial web proxy cache system. Only LRU web caching policy has been considered in this comparison which is considered as a limitation of the proposed framework.

Markatchev and Williamson (2002) a synthetic workload generator called ProWGen has been presented for simulation evaluation of web proxy caches. ProWGen tool considers three characteristics of web workload that are document popularity distribution, document temporal locality and the correlation between the size and popularity of the document. Furthermore, ProWGen tool simulates three web caching polices that are LRU, LFU-A and GD-Size; however, it is a Unix-based tool.

Jin *et al.* (2000) a trace-driven simulator for evaluating the performance of web proxy caching policies has been presented. The proposed tool has studied the effects of workload characteristics such as object size, recency and frequency on the performance of web proxy caches and their replacement policies.

On the other hand, five months have been spent for collecting proxy logs that are around 117 million logs.

Marquez *et al.* (2008) a windows-based simulator for caching and prefetching has been presented. The proposed tool takes Squid traces as inputs and simulates the functionality of different web caching policies. The outputs of the simulator are saved in text files. The proposed tool has been built using the Borland C++ Builder 2006 IDE; however, WWPCS is built using Microsoft Visual Studio 2010 Express which makes it more compatible for windows operating systems since they are Microsoft productions.

Houtzager and Williamson (2003) a packet-level simulation for studying the optimal web proxy cache placement has been performed based on NS-2 network simulations. The study has been carried out considering the network-level effects on the user-level web performance.

Liyanaarachchi and Weerawarana (2012), an end-to-end caching algorithm for web services has been proposed. It consists of two caches that are located in client and server sides.

Jayasooriya *et al.* (2013), A web data caching algorithm for Mobile Ad Hoc Network (MANET) that is called iCache has been proposed. iCache can be implemented for Android and Linux platforms. Also, it reduces the response delay, improves data availability and preserves bandwidth usage.

Tiwari and Kumar (2012a), a hybrid web caching algorithm has been proposed to overcome the scaling and reliability problems of the World Wide Web (WWW) such as congested proxy servers. The algorithm uses hierarchical web caching approaches along with a dynamic mechanism of proxy servers. This mechanism utilizes the bandwidth because most of requests are satisfied from local sites.

In (Sathiyamoorthi and Murali Bhaskaran, 2012), a web caching replacement policy has proposed which is based on recency, frequency and popularity factors with the help of web usage mining.

In (Tiwari and Kumar, 2012b), a dynamic web caching has been proposed to overcome the delays frequent disconnections of incurred by proxy servers.

## 6. CONCLUSION

A simulation tool is defined as a piece of software or hardware which can predict the behavior of a certain model that represents the characteristics of a system.

In computer science, simulations play key roles in studying and analyzing the behavior of computer networks under different conditions.

In this study, a simulation tool called WWPCS for analyzing the performance of traditional web proxy caching policies has been presented.

However WebTraff is a unix-based simulation tool, WWPCS runs under windows operating systems. Furthermore, WWPCS takes into account the HR and the BHR as performance metrics of web caching policies.

WWPCS is focusing on web proxy caching because web caching improves the web performance by storing web objects close to the clients, which reduces the latency of delivering web objects to the end-user. Moreover, it utilizes the bandwidth of the network.

WWPCS simulates five web caching policies that are FIFO, RAND, GD-Size, LRU and LFU. WWPCS takes the modified proxy log files as input; while it calculates web caching policy's HR and BHR as outputs.

The results show that the performance of a web caching policy depends on the replacement factor which is considered by the policy.

The computational overhead of running the simulation is considered as one of the limitations of WWPCS. Thus, reducing the simulation time might be considered as future work. Also, simulating other web caching policies would be considered as another improvement of WWPCS.

## 7. ACKNOWLEDGEMENT

The traces and proxy log files have been provided by the National Science Foundation (grants NCR-9616602 and NCR-9521745) and the National Laboratory for Applied Network Research (NLANR).

## 8. REFERENCES

- Ali, W., S.M. Shamsuddin and A.S. Ismail, 2012. Intelligent naïve bayes-based approaches for web proxy caching. *Knowl. Based Syst.*, 31: 162-175. DOI: 10.1016/j.knosys.2012.02.015
- Cao, P. and S. Irani, 1997. Cost-aware WWW proxy caching algorithms. *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, Dec. 8-11, USENIX Association, USA, pp: 193-206.
- Cardenas, L.G., J.A. Gil, J. Domenech, J. Sahuquillo and A. Pont, 2005. Performance comparison of a web cache simulation framework. *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, Mar. 28-30, IEEE Xplore Press, pp: 281-284. DOI: 10.1109/AINA.2005.275
- ElAarag, H., 2013. A Quantitative Study of Web Cache Replacement Strategies Using Simulation. In: *Web Proxy Cache Replacement Strategies*, Cobb J. and S. Romano (Eds.), Springer Briefs in Computer Science, Springer, London, ISBN: 978-1-4471-4892-0, pp: 17-60.
- Houtzager, G. and C. Williamson, 2003. A packet-level simulation study of optimal web proxy cache placement. *Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems*, Oct 12-15, IEEE Xplore Press, pp: 324-333. DOI: 10.1109/MASCOT.2003.1240677



- Jayasooriya, I.U., T.A. Nallaperuma, U.K. Herath, S.R. Ranasinghe and M. Liyanage *et al.*, 2013. Decentralized peer to peer web caching for mobile ad hoc networks (iCache). Proceedings of the 8th International Conference on Computer Science and Education, Apr. 26-28, IEEE Xplore Press, Colombo, pp: 218-223. DOI: 10.1109/ICCSE.2013.6553913
- Jelenković, R.P. and A. Radovanovic, 2009. Asymptotic optimality of the static frequency caching in the presence of correlated requests. *Operat. Res. Lett.*, 37: 307-311. DOI: 10.1016/j.orl.2009.03.011.
- Jin, M., F. Rich and T. Jin, 2000. Performance evaluation of web proxy cache replacement policies. *Perform. Evaluat.*, 39: 149-164. DOI: 10.1016/S0166-5316(99)00062-0
- Kumar, C. and J.B. Norris, 2008. A new approach for a proxy-level web caching mechanism. *Decision Support Syst.*, 46: 52-60 DOI: 10.1016/j.dss.2008.05.001
- Liyanaarachchi, A. and S. Weerawarana, 2012. An end-to-end caching protocol for web services. Proceedings of the International Conference on Advances in ICT for Emerging Regions, Dec. 13-14, IEEE Xplore Press, Colombo, pp: 96-102. DOI: 10.1109/ICTer.2012.6421410
- Markatchev, N. and C. Williamson, 2002. WebTraff: A GUI for web proxy cache workload modeling and analysis. Proceedings of the 10th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, Oct. 16-16, IEEE Xplore Press, pp: 356-373. DOI: 10.1109/MASCOT.2002.1167096
- Marquez, J., J. Domenech, J.A. Gil and A. Pont, 2008. A web caching and prefetching simulator. Proceedings of the 16th International Conference on Software, Telecommunications and Computer Networks, Sep. 25-27, IEEE Xplore Press, Split, pp: 346-350. DOI: 10.1109/SOFTCOM.2008.4669508
- Podlipnig, S. and L. Boszormenyi, 2003. A survey of web cache replacement strategies. *ACM Comput. Surveys*, 35: 374-398. DOI: 10.1145/954339.954341.
- Sathiyamoorthi, V. and V. Murali Bhaskaran, 2012. Web Caching through modified cache replacement algorithm. Proceedings of the International Conference on Recent Trends in Information Technology, Apr. 19-21, IEEE Xplore Press, Chennai, Tamil Nadu, pp: 483-487. DOI: 10.1109/ICRTIT.2012.6206749
- Tiwari, R. and N. Kumar, 2012a. A novel hybrid approach for web caching. Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Jul. 4-6, IEEE Xplore Press, Palermo, pp: 512-517. DOI: 10.1109/IMIS.2012.39
- Tiwari, R. and N. Kumar, 2012b. Dynamic web caching: for robustness, low latency and disconnection handling. Proceedings of the 2nd IEEE International Conference on Parallel Distributed and Grid Computing, Dec. 6-8, IEEE Xplore Press, Solan, pp: 909-914. DOI: 10.1109/PDGC.2012.6449945
- Yasin, W., H. Ibrahim, N.A.W.A. Hamid and N.I. Udzir, 2013. The affects of caching in browser stage on the performance of web items delivery. Proceedings of the 2nd International Conference on Digital Enterprise and Information Systems, Mar. 4-6, SDIWC, Malaysia, pp: 212-219.