

Evaluation of a Multi-Queue Web Caching Scheme that Differentiates the Content-Type of Documents¹

F.J. González-Cañete, E. Casilari, and A. Triviño-Cabrera

*Departamento de Tecnología Electrónica, Universidad de Málaga, E.T.S.I. Telecomunicación,
Campus de Teatinos, 29071 Málaga, Spain
{fgc, ecasilari, atc}@uma.es*

Abstract

This work presents a caching scheme that differentiates the content-type of the documents and uses a different queue for each of them. In that way, the replacement policy that better performance is achieved for each content-type and cache size can be applied. We evaluate by means of simulation the performance of the LRU, LFU-DA, GDSF and GD replacement policies in the classical caching scheme with the results obtained with this new scheme using the same replacement policy for each content-type and using a different replacement policy for each content-type.*

1. Introduction

Web caching is a technique that consists in storing the documents that are requested by the Internet users in a cache that is near the user so when the documents are requested again, they are served directly by the cache. In this way, Web caching reduces the redundant data transfer across the Internet, reduces the traffic on the original servers and decreases the delays perceived by the users. But Web caching suffers some defects such as the difficulty to guarantee the consistency of documents stored, the increment of dynamically generated documents that can not be cached and decrease the cache performance, and the reticence of the Web Masters to lose the control over the users that access the sites because the documents are served from a cache and not from the original server [1]. In spite of these drawbacks, Web caching is a technique that is widely used and is in constant development. The Web pages, and hence the Internet traffic, are composed of documents that are requested by the users. These

documents can be grouped by their content-type in the following five categories: Applications, Audio, Images, Text and Video. The Application content-type corresponds to documents that can be executed (exe, cab, dll,...), some text formats (MS-Word, pdf, rtf,...) and compressed files (zip, tar). The Audio documents contain sound information such as music or voice (wav, wma, realaudio,...). The Image content-type includes graphic objects or photography (gif, jpg, bmp,...). The Text content-type includes plain text documents (html, xml,...). Finally, the Video documents are composed of image sequences with audio (mpg, avi, wmv,...).

Some previous works have studied or developed replacement policies that consider dividing the documents according to their content-type. Yu et al. [2] proposed the MCW algorithm that consists in dividing the cache space in four sections: Text, Images, Audio and Video. The Video section in turn is divided into three sections for B, P and I frames. This replacement policy was specifically developed for streaming although it can be utilized for continuous documents. Each document is weighted using a function that takes into account the frequency, the tendency of references and the priority assigned to each content-type. On the other hand, Khayari et al. [3] proposed only to cache the Video (mpeg), Images (jpg and gif), flash Applications and Text (plain and html) documents in a shared cache space. They evaluated the performance of this policy using some replacement policy algorithms and they concluded that there is not an important performance improvement compared to the performance obtained caching every document type. Lindemann et al. [4] evaluate the performance of the LRU, LFU-DA, GD-Size and GD* replacement

¹ This work was partially supported by the public Project TEL2003-07953-C02-01

policies applied for only one of the HTML, Images, Applications and Multimedia content-types at the same time. They conclude that the GD* is the best option for every document type.

The rest of this paper is organized as follows. In section 2 we expose a caching scheme that uses a different queue for each document content-type and we describe the replacement policies used. Section 3 contains the characterization of the workload utilised to evaluate the new scheme. Section 4 shows the simulation results comparing the new scheme to the classical ones. Finally, section 5 enumerates the conclusion of this work.

2. Experimental design

In this work we propose to divide the cache size in five sections, one for each of the main Web document content-types (Applications, Audio, Images, Text and Video). The size assigned to each content-type is a key decision for the performance of this cache. Two main options can be proposed, a fixed size or a dynamically adjusted size. In this work the fixed cache size for each content-type is considered for its simplicity. The percentage of the traffic generated by the documents of each content-type will determine the percentage of the cache size assigned to each content-type.

In a first experiment we will compare the performance of this architecture applying the same replacement policy to each section with the performance obtained in a classical cache that does not difference between content-types. In a second place, we will assign a different replacement policy to each content-type and evaluate the performance obtained.

The replacement policies considered in this study are explained below:

The LRU (Least Recently Used) replaces the document that was least recently referenced. It has the advantage that it is a very simple algorithm which can be implemented to work very efficiently. On the contrary it has the handicap that it does not take into account the frequency and size of documents.

LFU-DA (LFU-Dynamic Aging) was proposed in [5] and solves the “cache pollution” problem of the LFU algorithm using a variable that contains the “age” of the cache, i.e. the number of references to the least frequently used document. When a new document is inserted or the referenced document is already in cache, the reference count of the document is added to the aging variable.

The GDSF (Greedy-Dual Size with Frequency) policy was proposed in [6]. It is a modification of the Greedy-

Dual-Size algorithm which considers the reference count of documents. The function which weights each document is shown in Ec. 1, where $F(p)$ is the number of references to document p , $C(p)$ is the cost and $S(p)$ is the size of p . The documents evicted when a new document comes in, are the ones with the least weight value. As functions costs Eq 2 and Eq 3 have been proposed among others. Eq. 2 considers that every document has the same transmission cost, while Eq. 3 takes into account the number of packets needed to transmit the document as the transmission cost. They are called ‘constant’ and ‘packets’ cost models respectively. To distinguish the cost model utilised, GDSF(1) is used for the constant cost model and GDSF(p) is used for the packets cost model.

$$V(p) = F(p) \frac{C(p)}{S(p)} \quad (1)$$

$$C(p) = 1 \quad (2)$$

$$C(p) = 2 + \frac{S(p)}{1460} \quad (3)$$

The GD* (Greedy-Dual*) [7] algorithm is a modification of the Greedy-Dual-Size algorithm taking into account the frequency count of documents and the temporal correlation of references using a parameter β in the value function (Eq. 4), where the only difference to the GDSF replacement policy is the parameter β . This parameter is a number between zero and one that models the temporal correlation between two successive accesses to the same document and it is a characteristic of each workload.

$$V(p) = \left(F(p) \frac{C(p)}{S(p)} \right)^{\frac{1}{\beta}} \quad (4)$$

The LRU and LFU-DA replacement policies have been selected because they are classical algorithms that have been widely used as basic algorithms. On the other hand, the GDSF and GD* replacement policies have been considered because it has been demonstrated that they outperforms the other algorithms for Web caching because they consider the size of the documents.

3. Workload characterization

The workload utilised in this work corresponds to a Squid proxy cache located in the Research Triangle Park (North Carolina, USA) from the 7th to the 11th of June 2004.

This trace has been processed to purge those requests that have been generated dynamically by CGI (Common Gateway Interface) discarding the requests that contain the strings ‘cgi’, ‘cgi-bin’ or ‘?’. As cacheable response codes, 200 (OK), 203 (Partial), 206 (Partial Content), 300 (Multiple Choices), 301 (Moved) and 302 (Redirects), have been considered [8]. For the requests with a 304 (Not Modified) response code as well as the requests whose content-type was unknown, the document has been requested again to the original server to obtain the real size or the content-type. Table I summarises the basic characteristics of the trace after its process.

Table II shows the distribution of the number of requests and traffic generated by each content-type. The Images account for a 75% of the request although the percentage of traffic generated is 36%. On the other hand, the Applications that only account the 8% of the request are responsible of the 33% of the traffic generated. The size assigned to each content-type for the cache evaluation is shown as ‘Selected’ in this table. The percentage of bytes requested have been truncated for every content-type except for the Images that is increased with the percentage of documents of unknown content-type and the rounded sizes of the other content-types.

In order to compute the β parameter of the GD* replacement policy, it is necessary to model the temporal correlation using the method proposed in [9]. This method consists of drawing in a log/log scale the probability distribution of distances between references to documents with the same frequency of access k in the whole workload considered and calculating the slope β of the distribution. The slope has been calculated using a minimum square regression approximating the Eq. 5.

A value of $k = 8$ has been selected for this analysis, although similar values of β are obtained for other values of k . Table III summarizes the values of β calculated for each content type.

TABLE I
Main workload characteristics

Number of Requests	4,040,036
Size (GB)	40.4
Distinct documents	1,713,903
One timers	1,299,217

TABLE II
Distribution of content-types

Content-Type	Requests (%)	Bytes (%)	Selected
Applications	8.08	33.51	33
Audio	0.28	3.49	3
Images	75.54	36.33	38
Text	15.77	23.15	23
Video	0.12	3.19	3
Unknown	0.21	0.33	0

$$T \propto \frac{1}{t^\beta} \quad \text{for documents with frequency } k \quad (5)$$

The distance between consecutive references to the same document is an important topic in Web caching because if these distances are small, the probability of being evicted from the cache decreases. As we separate the documents according to their content-type, a reduction of the distances is expected. Fig. 2 represents the histogram of the distances between consecutive accesses for all documents and for each content-type calculated using the LRU Stack model [10].

This figure shows that the distances are reduced drastically for every content-type except for the Images where the reduction is small. This behaviour is due to the fact that the 75% of requests are images, so for the rest of the content- types, when they are treated separately, the distances between them are reduced.

TABLE III
The β parameter calculated for each content-type

Content-Type	β
All	0.46
Application	0.51
Audio	0.40
Image	0.46
Text	0.54
Video	0.95

4. Cache simulation results

Two metrics have been utilized to evaluate and compare the efficiency of the replacement policies dividing the cache size between the content-types.

The HR (Hit ratio) is defined as the total number documents that caused a hit in the cache (i.e. the document already was in the cache) divided by the total number of requests; and the BHR (Byte Hit Ratio) is defined as the summation of the document sizes that caused a hit in the cache divided by the size of the documents processed.

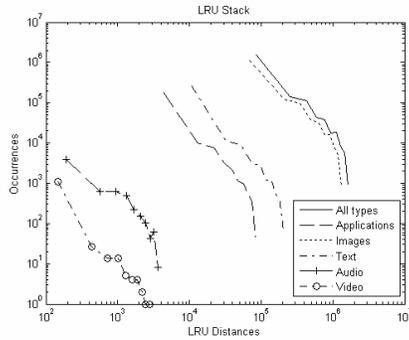


Fig. 2. Distribution of distances between references using the LRU Stack model

To process the workload and calculate the performance of the different schemes, a proxy cache simulator has been developed. This simulator implements the policies explained in section 2 and can be configured to simulate different sizes of cache. The simulator processes the workload files returning a result file that contains parameters such as the HR and BHR, the total size of the cache, the number of documents evicted, the total number of documents at the end of the simulation, etc. 10% of the trace has been used to “warm up” the cache and avoid cold start influences. To distinguish the modification of a document from the interruption of a transfer we compare the difference between sizes of successive requests to the same document. If the difference is less than 5% of the document size, we consider that the document has been modified and it has to be treated as a new document; otherwise a cancel is considered [11]. Documents with an unknown content-type are rejected and they are not cached.

The simulations performed have shown that the performance of the classical schemes, LRU and LFU-DA, is slightly better than the multi-queue scheme for the HR, although the difference is smaller as the cache size is decreased. Similar results are found for the BHR, but for small cache sizes, the new scheme outperforms the classical ones although the performance difference is insignificant. In spite of the reduction in the distances between successive requests of the same document that was demonstrated in the previous section, we did not obtain a better performance in the LRU replacement policy. This behaviour is due to the fact that the size of the documents is a factor to take into account and it is not considered in the LRU Stack model and that the Images, that are the majority of the documents, only reduce their distances slightly.

If we use the GDSF algorithm, the multi-queue scheme outperforms the classical scheme for the HR if the packet cost model is used and the BHR if the constant cost model is utilised for all cache sizes.

The results obtained for the GD* policy are similar to the ones obtained for the GDSF algorithm although the performance is only improved for the BHR and the constant cost model.

As can be observed from the previous results there is not a clear advantage on using a multi-queue scheme dividing the traffic according to the content-type of the documents and applying the same replacement policy to each queue. Only an improvement is obtained for the BHR using the GDSF(1) algorithm and for the HR using GDSF(p). A better performance is obtained for the BHR using the GD*(1) replacement policy although the HR is decreased. The main drawback is that once the documents have been separated according to their content-type, the characteristics of the new workload can vary from the characteristics of the whole workload, so a different replacement policy could work better for each group of documents.

Based on the results obtained in [12], a different replacement policy has been associated to the queues used to store the documents of each content-type. The main objective of this assignment is trying to maximize the HR and the BHR choosing the algorithms that better performance offers for the different cache sizes. To maximize the HR, the GD*(1) replacement policy has been chosen for all content-types except for the Video content-type that uses GD(1). These replacement policies were assigned for all cache sizes because they obtain the best HR for each content-type. On the contrary, to maximize the BHR, the GDSF algorithm with the packet cost model was utilised for the Application and the Audio documents for all cache sizes. The LFU was used for Image caches of 5% and 2% of the cache size and GD*(p) for greatest caches. For the Text and Video content-types, the LFU-DA and the LRU replacement policies were utilised respectively. Fig. 3 represents the performance obtained by this replacement policy assignment compared with the results of the replacement policies that do not difference the documents by their content-type. As it can be observed, the multi-queue scheme that tries to maximize the HR obtains a HR similar to the performance of the best replacement policy, but it has the drawback that the BHR is very low. On the other hand, the scheme that tries to maximize the BHR achieves a similar BHR or even it outperforms the other algorithms, especially for small caches, as well as

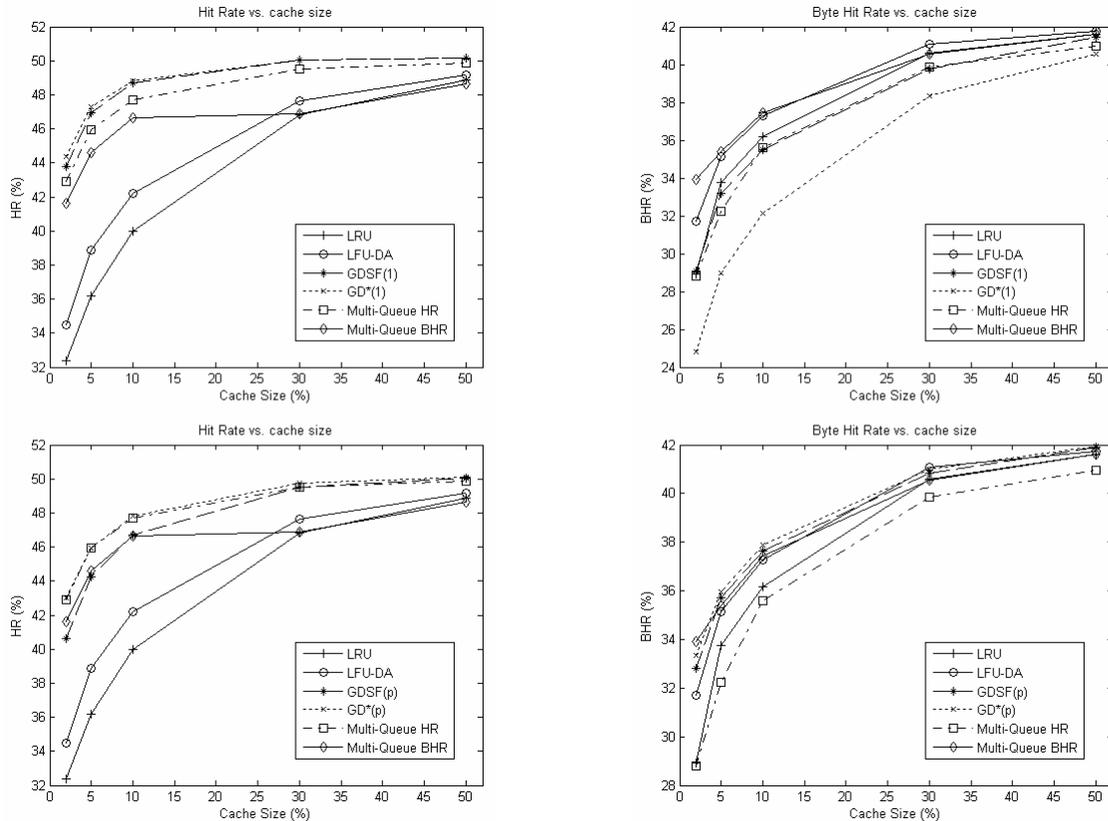


Fig. 3. HR (left) and BHR (right) obtained for each replacement policy compared with the multi-queue schemes using the constant cost model (top) and the packet cost model (bottom)

a good HR. From the previous results, we can conclude that this scheme at least obtains similar performance for the HR and better result for the BHR than the best algorithm with a unique queue.

In spite of the previous results, the drawback of dividing the cache storage space into constant size sections is that the space assigned to each queue can be wasted because the documents that enter the cache evict other documents of the same content-type. In that way, a small document could evict a big document in one of the queues leaving a big storage space free, but if the next document has to enter another queue, some documents could be evicted even though there is free space in the cache. Table IV summarizes the percentage of cache storage space that is wasted at the end of the simulation that tries to maximize the HR in Fig. 3. This table shows that there is a lot of space wasted that could be used and a new document that reaches the cache could take advantage of it. This could be done assigning the free space dynamically between the queues or even reassigning the cache space from content-type queues that are not getting a

good performance to another that is obtaining a better performance. This dynamic distribution should be done considering that each content-type of documents obtains different performance results. Table V shows the mean of HR and BHR obtained by each content-type.

All content-types obtain a high HR except the Video. Applications, Text and Video obtain a high BHR but Audio and Images obtain a poor BHR, even when the Image documents are the most requested.

These results suggest that the distribution of the cache space into the different queues should be taken into account if we want to maximize the HR or the BHR to distribute the space according to the content-

TABLE IV
Percentage of cache space wasted

	Cache Size (%)				
	50	30	10	5	2
Space wasted	1 %	2 %	7 %	14 %	3 %

TABLE V
Mean of HR and BHR by content-type

	HR	BHR
Applications	64 %	53 %
Audio	54 %	21 %
Images	45 %	15 %
Text	45 %	38 %
Video	22 %	60 %

types that obtain better results for the metric we want to maximize. We left this study for a future work and development.

4. Conclusions

In this paper, a cache scheme that divides the cache storage space into five sections and each of them only store one content-type of documents (Applications, Audio, Images, Text and Video) has been proposed. The performance of this scheme has been evaluated by means of simulation using the same replacement policy algorithm for each section and assigning a constant size to each section depending on the percentage of traffic generated by the documents of each content-type. The LRU, LFU-DA, GDSF and GD* replacement policies have been utilised for those simulations.

We have compared the performance of the multi-queue scheme to the classical one that uses a unique queue to store all documents, resulting that the classical scheme outperforms the multi-queue in most cases. The new scheme outperforms the classical ones for the BHR using the GDSF(1) and GD(1) replacement policies although the HR is decreased. There is not an improvement using the LRU and LFU-DA algorithms.

In a second evaluation we have assigned to the queue utilised for each content-type of documents, the replacement policy that better performance obtains for the HR and the BHR according to the storage assigned. The scheme that tries to maximize the HR as well as the scheme that tries to maximize the BHR obtains a similar HR performance compared to the algorithms that utilise a unique queue. On the other hand, the first scheme outperforms the BHR for small cache sizes although the second scheme obtains a poor NHR.

We have evaluated the space that is wasted using the scheme proposed due to the constant size assigned to store the documents of each content-type and we have proposed to assign and redistribute the cache space dynamically to take advantage of this space. This allocation of the cache size should be decided

considering that different content-types obtain different performance values, so the space distribution function should consider what content-types should be prioritized to maximize one of the metrics considered.

5. References

- [1] D. Wessels, "Web Caching", O'Reilly & Associated, June 2000.
- [2] F. Yu, Q. Zhang, W. Zhu, Y. Zhang, "Network-Adaptive Cache Management Schemes for Mixed Media", Second IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM), October, 2001.
- [3] R.A. Khayari, M. Best, A. Lehmann, "Impact of Document Types on the Performance of Caching Algorithms in WWW Proxies: A Trace Driven Simulation Study", 19th IEEE International Conference on Advanced Information Networking and Applications, 2005.
- [4] C. Lindemann, O.P. Waldhorst, "Evaluating the Impact of Different Documents Types on the Performance of Web Cache Replacement Schemes", IEEE International Conference on Dependable Systems and Networks (DSN'02), Washington, June, 2002.
- [5] M. Arlitt, C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications", IEEE/ACM Transactions on Networking, 1997.
- [6] L. Cherkasova, "Improving WWW Proxies Performance with Greedy-Dual-Size Frequency Caching Policy", Technical Report HP Labs HPL-98-69, 1998.
- [7] S. Jin, A. Bestavros, "GreedyDual* Web Caching Algorithm: Exploiting the Two Sources of Temporal Locality in Web Request Streams", Intl' Journal of Computer Communications, Vol. 24, No. 2, pp. 174-183, 2001.
- [8] X. Zhang, "Cachability of Web Objects", Technical Report 2000-19, 2000.
- [9] S. Jin, A. Bestavros, "Temporal Locality in Web Request Streams. Sources, Characteristics and Caching Implications", Poster proceedings ACM SIGMETRICS 2000 Conference, 2000.
- [10] R. Mattson, J. Gecsei, D. Slutz, I. Traigner, "Evaluation Techniques for Storage Hierarchies", IBM Systems Journal, 9, pp. 78-117, 1970.
- [11] M. Arlitt, R. Friedrich, T. Jin. "Workload Characterization of a Web Proxy in a Cable Modem Environment", Technical Report HPL-1999-48, Hewlett-Packard Laboratories, 1999.
- [12] F.J. Gonzalez-Cañete, E. Casilari, A. Triviño-Cabrera, "Characterizing Document Types to Evaluate Web Cache Replacement Policies", Pending of acceptance in Content Caching and Distribution Networks WorkShop, Networking 2006, Coimbra, Portugal, 15-19 May 2006.