ISBN: 978-972-8924-30-0 © 2007 IADIS

A CONTENT-TYPE BASED EVALUATION OF WEB CACHE REPLACEMENT POLICIES

F.J. González-Cañete, E. Casilari, A. Triviño-Cabrera
Department of Electronic Technology, University of Málaga, Spain
University of Málaga, E.T.S.I. Telecomunicación, Campus de Teatinos, 29071, Málaga, Spain
{fgc,ecasilari,atc}@uma.es

ABSTRACT

In this paper, a study of the performance of six replacement policies taking into account only one content-type of documents each time (Application, Audio, Images, Text and Video) has been developed in order to implement a proxy cache that differences the type of traffic. The classical caching algorithms LRU, LFU and LFU-DA and the caching schemes specifically developed for Web documents GD-SIZE, GDSF and GD* have been studied. Using a trace log of a real proxy cache, a characterization of the main properties of the documents of each content-type has been performed. Finally, a trace driven simulation study of the performance of the six replacement policies has been developed for the traffic generated by each content-type considered. In that way we can conclude which are the replacement policies that better perform for each content-type and cache size.

KEYWORDS

Web caching, replacement policies, document content-types.

1. INTRODUCTION

Internet and the World Wide Web (the Web) are in a continuous evolution and growth, therefore many efforts to optimize them have been developed. One of the most important optimization techniques is the Web proxy caching that store the documents requested by the users close to them. Since it was proposed in (Luotonen, 1997), Web proxy caching has been utilized to reduce the latency that the users perceive, the HTTP traffic as well as the servers load. After this original proposal of a Web proxy cache, many research activities have aimed to study and develop replacement policies (Poplipnig, 2003) (Balamash, 2004), algorithms for cache coherence (Krishmamurthy, 1999) and cache architectures (Busari, 2000) in order to improve the performance of the caching system.

One of the main research lines is based on differencing the types of documents that are present in the Web (Images, Text, Video,…). Khayari proposed to store in the cache only the most frequently demanded document types (mpeg, gif, jpg, flash, html and plain) although his proposal did not outperform the cache performance (Khayari, 2005). In this paper we analyze the best replacement policy for each document type by means of simulations.

This paper is organized as follows. Section 2 summarizes the trace processing and the statistical characterization of the workload based on the content-type and section 3 lists and evaluates the performance of a proxy cache that takes into account only one content-type of the downloaded document. Finally, Section 4 presents the main conclusions of this paper.

2. TRACE PROCESSING AND CHARACTERIZATION

To evaluate the performance of a cache that only considers one type of document content type at a time, a workload trace that contains HTTP requests from a proxy of the IRCache project has been utilised (IRCache). This proxy is located in the Research Triangle Park (North Carolina, USA). The traces include requests from the 7th to the 11th of June 2004 generated by the Squid Web proxy cache software (Squid

Cache). The traces include information for each HTTP request processed by the proxy, such as the time the request was initiated, the documents size, the URL, the document type (content-type), the request method (GET, POST, …) and the response code from the server.

This trace has been preprocessed to purge those requests that have been generated dynamically by CGI (Common Gateway Interface) because the documents returned by these kind of requests are unique for each request and therefore they should not be cached (Zhang, 2000). Because of this fact, the requests that contain the strings 'cgi', 'cgi-bin' or '?' have been discarded. Those requests that contain the string ':3128' have been filtered as this is the port that IRCache utilises to interchange information between collaborating caches. As cacheable response codes, 200 (OK), 203 (Partial), 206 (Partial Content), 300 (Multiple Choices), 301 (Moved) and 302 (Redirects), have been taken into account. For 304 (Not Modified) response code, the size shown in the traces corresponds to the size of the response and not to the real document size (it informs that the document has not been modified since it was requested the last time), consequently these documents have been requested again to the original server to obtain the real size. On the other hand, the documents whose content-type was unknown have been requested again. Table 1 summarises the basic characteristics of the trace after its process.

Table 1. Main trace characteristics

Number of Requests	4,040,036
Size (GB)	40.4
Distinct documents (%)	42.4
One-timers (%)	32.1

The next step is to divide the requests by the content-type of each document according to those defined in (IANA). These content-types are: Applications, Audio, Images, Text and Video. Table 2 shows the characteristics of the documents according to its type.

The Image and Text documents account for the 91 % of requests and the 59 % of bytes transferred, so they are the most influent content-types. There are major differences between the document size of the different types as it can be observed if we compare the mean and median of each type.

Table 2. Trace characteristics by content-type of documents

	Applications	**Audio**	**Images**	**Text**	**Video**
Mean (bytes)	41,480	123,402	4,812	14,687	260,098
Median (bytes)	3,379	7,851	1,406	3,263	573
Standard deviation (bytes)	761,429	948,026	21,399	104,608	974,697
Requests (%)	8.08	0.28	75.54	15.77	0.12
Bytes (%)	33.51	3.49	36.33	23.15	3.19
Distinct documents (%)	26.90	38.00	45.98	33.92	75.02

3. EVALUATION OF REPLACEMENT POLICIES

The main function of the replacement policy is to decide which document or documents to evict from the cache when the storage space is full and a new document has to be inserted. The documents evicted must be the ones that have less probability to be requested again in the near future.

The replacement policies utilized in this study are LRU (Least Recently Used) that evicts the documents that were referenced more time ago, LFU (Least Frequently Used) that evicts the documents with the lower number of reference count, LFU-DA (Least Frequently Used with Dynamic Aging) (Arlit, 1997) (Robinson, 1990) is an evolution of the LFU replacement policy that starts the reference count of the new inserted document with the reference count of the least frequently used document in order to favour the new inserted document, GD-Size (Greedy-Dual Size) (Cao, 1997) uses the function of Ec. 1 to value each document and evicts the documents with the lower value where *Cost(p)* is the cost of retrieving the document *p* from the Web server and *Size(p)* is the size in bytes of *p*, GDSF (Greedy-Dual Size with Frequency) (Cherkasova, 1998) uses a value function similar to GD-Size but also taking into account the frequency (*Freq(p)*) of

requests (Ec. 2) and finally the GD* (Greedy-Dual*) (Jin, 2001) uses a value function that takes into account the temporal correlation using the parameter β (Ec. 3). Table 3 shows the β parameter calculated for the GD* algorithm. The cost functions used in this study are the constant cost function shown in Ec. 4, that is, the cost to retrieve the documents is always the same and the packet cost function shown in Ec. 5 where the cost of retrieving each document is the number of packets needed to retrieve it.

$$V(p) = \frac{Cost(p)}{Size(p)} \quad (1)$$

$$V(p) = Freq(p)\frac{Cost(p)}{Size(p)} \quad (2)$$

$$V(p) = \left(Freq(p)\frac{Cost(p)}{Size(p)} \right)^{\frac{1}{\beta}} \quad (3)$$

$$Cost(p) = 1 \quad (4)$$

$$Cost(p) = 2 + \frac{Size(p)}{1460} \quad (5)$$

Table 3. β parameter calculated to model the temporal locality by content-type for the GD* algorithm

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

To evaluate and compare the efficiency of those policies for each content-type a proxy simulator has been developed. This simulator implements the policies explained in section 3 and can be configured to simulate different sizes of cache. The simulator processes the trace files explained in section 2 returning a result file that contains parameters such as the HR (Hit Ratio) and BHR (Byte Hit Ratio), total size of the cache, number of documents evicted, 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.

We first simulated a cache with infinite size to determine the total size filled in the cache for each content-type. The next simulations were performed using the 50%, 30%, 10%, 5% and 2% of the maximum sizes obtained for each type.

Figure 1 shows the evaluation of the replacement policies for the Application content-type. If we use the constant cost model, the cost based replacement policies clearly outperform the classical algorithms for the HR metric, with a performance close to 60% for a small cache size. This difference is more evident as the cache size decreases. The LRU is the worst option to maximize the HR. For the BHR metric the behavior is similar, although the difference between policies is only about 4%. Under the packet cost model GD* and GDSF are the best options to maximize the HR, but they are less efficient than using the constant cost model. GD* and GDSF outperform the other policies for the BHR too, obtaining slightly better results than with the constant cost model. In summary, for Application content-types GD* and GDSF are the best choices, obtaining high HR but low BHR, as could be expected due to the relationship between size and number of references presented in the trace. Small documents are more accessed than bigger ones and there are a lot of documents referenced many times; therefore a huge HR is obtained even for a small cache. On the other hand, the size distribution shows that the Application documents contains many big documents, therefore these documents are responsible of a great percentage of the traffic, but they are not referenced many times. This is the reason why little BHR is obtained. The simulation results for the Audio content-type are shown in Fig 2. GD*, GDSF and GD-Size obtain the same performance for all cache sizes in the constant cost model

and outperform the other algorithms for more than 20% for the HR. For the BHR, LFU and LFU-DA are the best choices for a very small cache, although the performance is similar for all policies while the cache size increases. Under the packet cost model, GD* is the best choice to maximize the HR, while all policies, except LRU and GD-Size, provide the same good performance for the BHR. For this content-type, both metrics can not be optimized simultaneously for cache sizes less than 10%. For a bigger cache size the cost replacement policies outperform HR and BHR. The Image content-type performance study is represented in Figure 3. This figure shows that the cost replacement policies maximize the HR in both cost models, although the performance is slightly better for the constant cost model. If we consider the BHR, there are tiny differences between policies, but LFU is the best choice for cache size greater than 10% and GDSF is the best for smaller cache sizes. Figure 4 illustrates the performance evaluation for the Text content-type and it shows similar characteristics to the Image content-type for the HR. The GD* and GDSF algorithms obtain the best performance for all cache sizes and both cost models. The LFU-DA algorithm seems to be the best choice to maximize the BHR, although GDSF(packets) obtains similar results. For the Text content-type, the GD-SIZE algorithm is the worst for both metrics. Finally, Figure 5 depicts the performance evaluation for the Video content-type. Only the constant cost model results have been depicted because both models obtain the same performance. The figure shows that, except for the LFU algorithm, the other policies present the same behaviour for both metrics and cost models, so any of them could be a good choice. There is little differences between replacement policies due to the fact that the percentage of references is only 0.12% and the document sizes versus number of references is very sparse, therefore cost policies can not obtain optimal results.

4. CONCLUSION

In this paper, an exhaustive study of the different content-types of HTTP traffic for Web documents has been developed. To perform this study a workload trace from a real proxy cache has been characterized obtaining its main statistic properties. Six replacement policies have been presented. Three classical schemes (LRU, LFU, LFU-DA) and three size-based schemes specifically developed for Web caches (GD-SIZE,GDSF, GD*) using two cost models (constant cost and packet cost). Then, a simulation driven study of the performance of these replacement policies for each content-type of Web documents has been performed. From this study some conclusions can be derived:

- Applications: The three cost algorithms with constant cost model maximizes the HR, but GDSF(p) and GD*(p) maximizes the BHR.
- Audio: For the HR, GD-SIZE(1), GDSF(1) and GD*(1) are good choices. To maximize the BHR, all policies give the same performance, except LFU that works better for a small cache size.
- Images: GDSF(1) and GD*(1) outperform the other policies for the HR. To maximize the BHR GDSF(p) and GD*(p) are the best choice for small caches (less than 10%) and LFU is better for bigger ones.
- Text: GDSF(p) maximizes both the HR and the BHR.
- Video: All replacement policies considered except LFU obtain the same results for HR and BHR.

As it can be observed in the previous summary, there is no a replacement policy that outperforms the others for all content-types, so to develop a proxy cache that distinguishes the content-types of documents, the best algorithm for each content-type should be applied. Another consideration to take into account is the metric that we need to maximize, since, except for the Text and Video content-types, the best algorithm for each metric differs.

ACKNOWLEDGEMENT

We would like to thank Duane Wessels for the access to the workload traces. This work was partially supported by the public Project N^{er} TEL2003-07953-C02-01

REFERENCES

Arlitt, M. and Williamson, C., 1997, Internet Web Servers: Workload Characterization and Performance Implications, *IEEE/ACM Transactions on Networking*, Vol. 5, N^{er} 5, pp. 631-645

Balamash, A. and Krunz, M., 2004, An Overview of Web Caching Replacement Algorithms, *IEEE Communications Surveys and Tutorials,* Vol. 6, N^{er} 2, pp. 44-56

Busari, M., 2000, Simulation of Web Caching Hierarchies, *Pd. Thesis*

Cao, P., 1997, Cost-Aware WWW Proxy Caching Algorithms, *USENIX Symposium on Internet Technologies and Systems,* Monterey, California

Cherkasova, L., 1998, Improving WWW Proxies Performance with Greedy-Dual-Size Frequency Caching Policy, *Technical Report HP Labs HPL-98-69*

Internet Assigned Numbers Authority (IANA), http://www.iana.org/assignments/media-types

IRCache project home page, http://www.ircache.net

Jin, S. and Bestabros, A., 2001, GreedyDual* Web Caching Algorithm: Exploiting the Two Sources of Temporal Locality in Web Request Streams, *Intl' Journal of Computer Communications*, Vol. 24, N^{er}. 2, pp. 174-183

Khayari, R.A. et al., 2005, 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*

Krishnamurthy, B. and Wills, C.E., 1999, Proxy Cache Coherency and Replacement – Towards a More Complete Picture, *IEEE International Conference on Distributed Computing Systems*

Luotonen, A., Altis, K., 1997, World-Wide Web Proxies, *First International Conference on the WWW*

Poplipnig, S. and Böszörmenyi, L., 2003, A Survey of Web Cache Replacement Strategies, *ACM Computing Surveys*, Vol. 35, N^{er} 4, pp. 374-398

Robinson, J.T. and Devarakonda, M.V., 1990, Data Cache Management Using Frequency-Based Replacement, *1990 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp.134-142

Squid Web Proxy Cache home page, http://www.squid-cache.org

Zhang , X., 2000, Cachability of Web Objects, *Technical Report 2000-19*

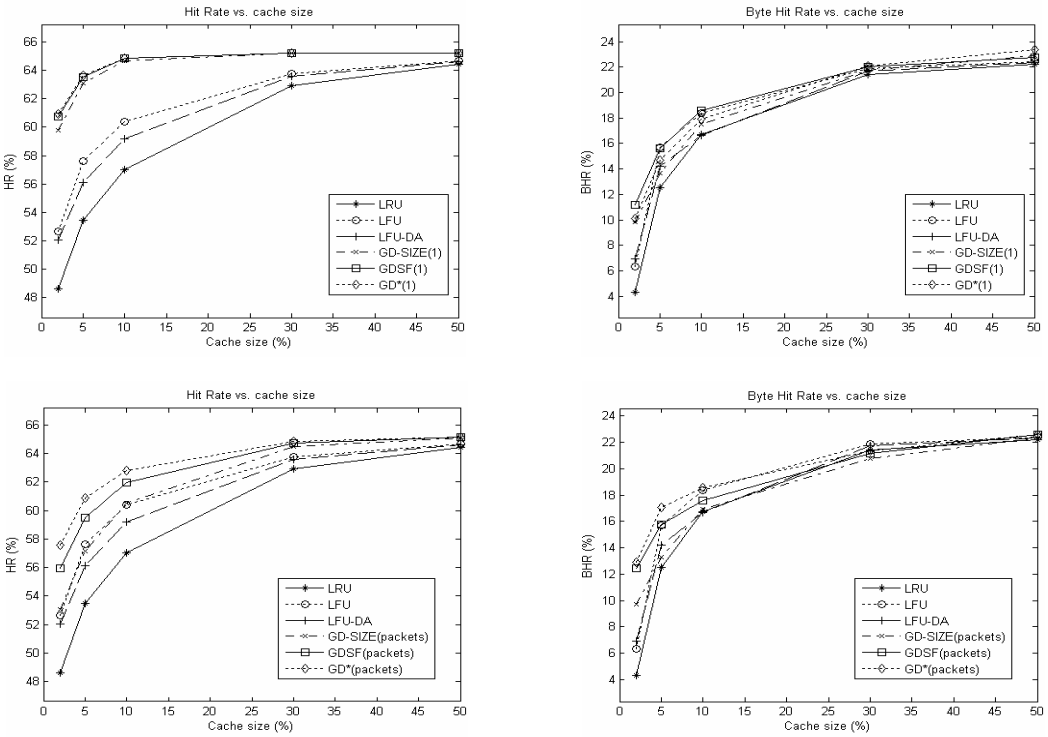

Figure 1. Evaluation of cache replacement policies for Application content-type.

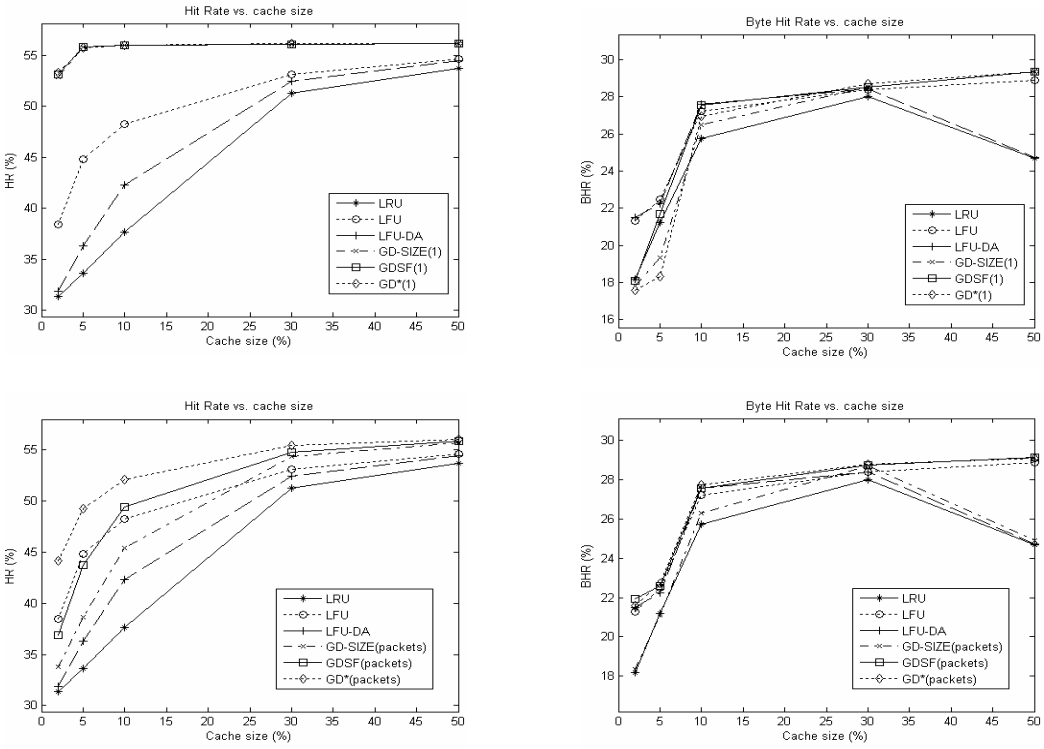

Figure 2. Evaluation of cache replacement policies for Audio content-type

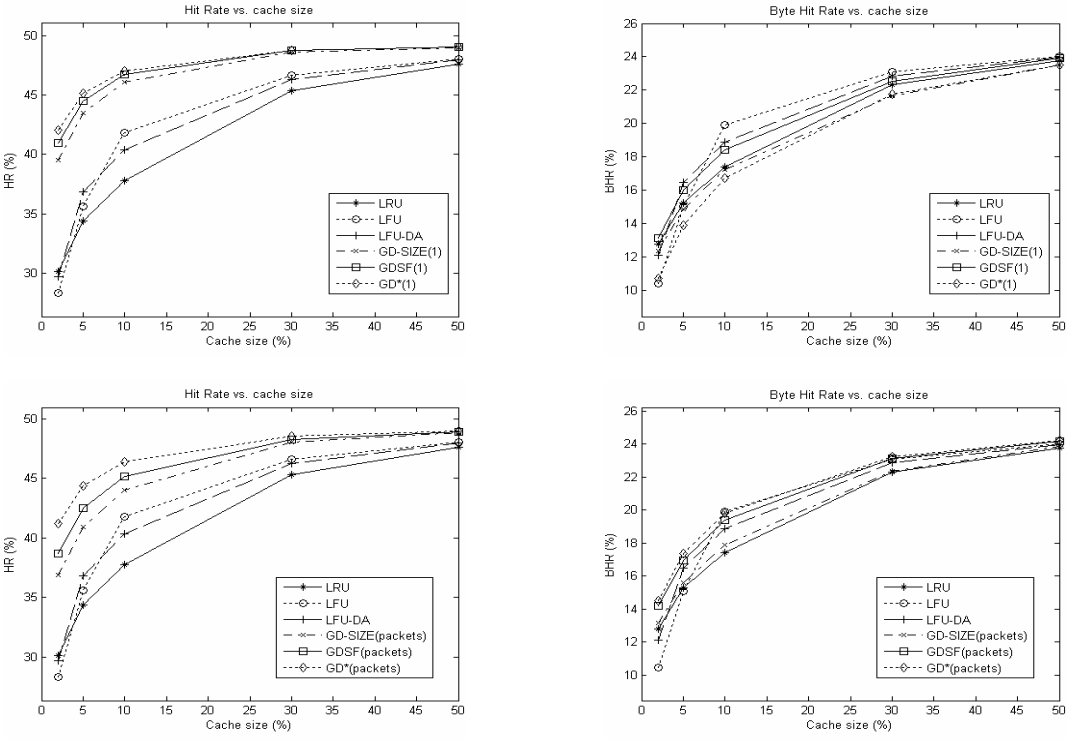

Figure 3. Evaluation of cache replacement policies for Image content-type

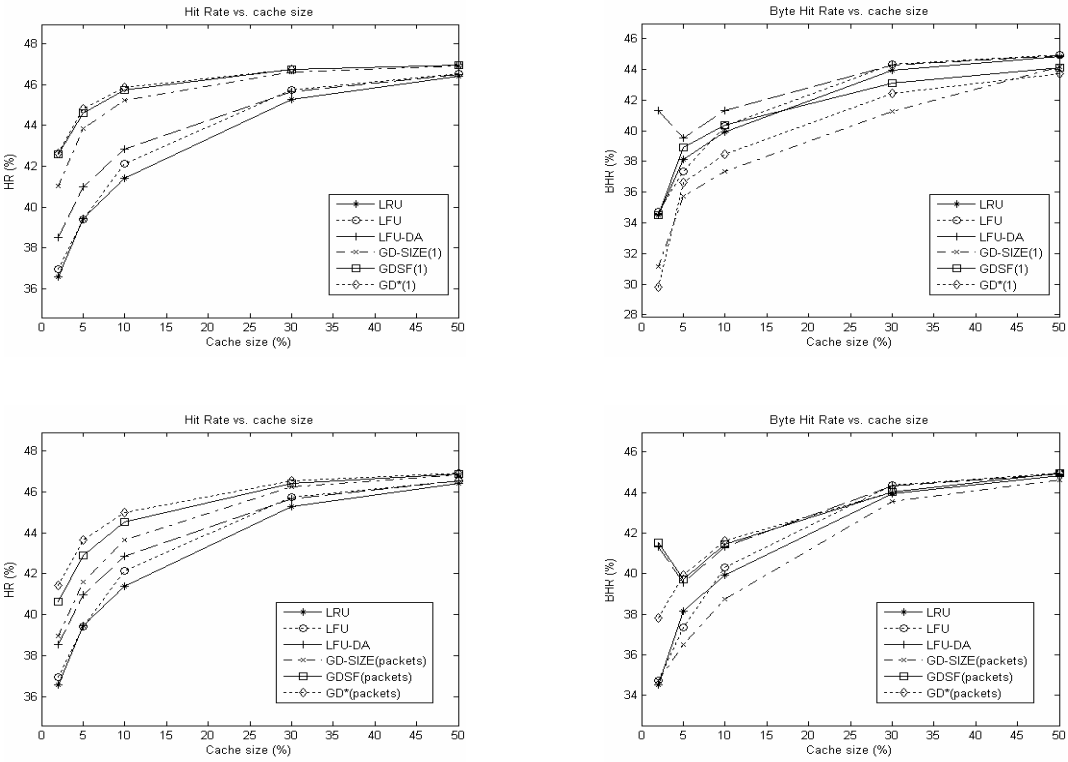

Figure 4. Evaluation of cache replacement policies for Text content-type

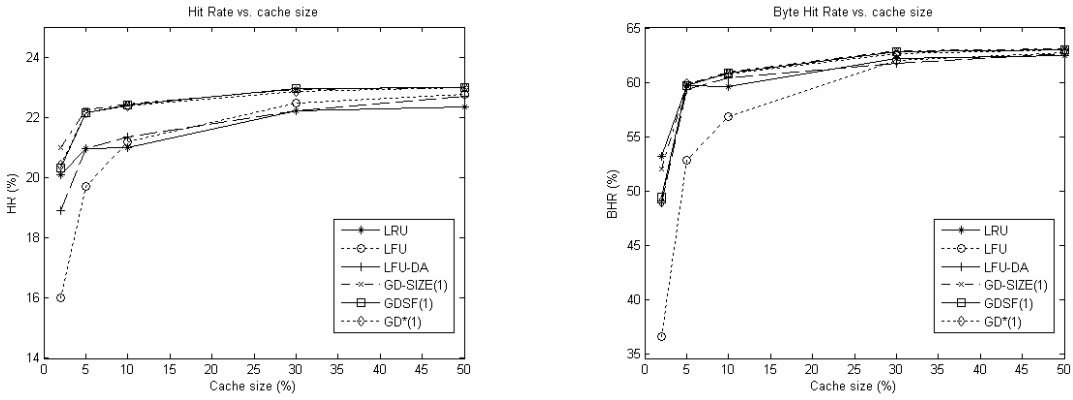

Figure 5. Evaluation of cache replacement policies for Video content-type.