Cost-Benefit Analysis of Web Prefetching Algorithms from the User's Point of View

Josep Domènech, Ana Pont, Julio Sahuquillo, José A. Gil

Department of Computing Engineering (DISCA)
Universitat Politècnica de València. Spain
jodode@doctor.upv.es;{apont,jsahuqui,jagil}@disca.upv.es

Abstract. Since web prefetching techniques were proposed in the second half of the 90s as mechanisms to reduce final users' perceived latency, few attempts to evaluate their performance have been done in the research literature. Even more, to the knowledge of the authors this is the first study that evaluates different proposals from the user's point of view, i.e., considering the latency perceived by the user as the key metric. This gap between the proposals and their correct performance comparison is due to the difficulty to use a homogeneous framework and workload. This paper is aimed at reducing this gap by proposing a cost-benefit analysis methodology to fairly compare prefetching algorithms from the user's point of view. The proposed methodology has been used to compare three of the most used algorithms in the bibliography, considering current workloads.

1 Introduction

Several ways of prefetching user's requests have been proposed in the literature: the preprocessing of a request by the server [1], the transference of the object requested in advance [2], and the pre-establishment of connections that are predicted to be made [3]. Despite the large amount of research works focusing on this topic, comparative and evaluation studies from the user's point of view are rare. On the one hand, the underlying baseline system where prefetching is applied differs widely among the studies. On the other hand, different performance key metrics were used to evaluate their benefits [4]. In addition, the used workloads are in most cases rather old, which significantly affects the prefetching performance [5], making the conclusions not valid for current workloads.

Researchers usually compare the proposed prefetching system with a non-prefetching one [6, 2], under heterogeneous conditions making it impossible to compare the goodness and benefits of each proposal.

Some papers comparing the performance of prefetching algorithms have been published [7, 8, 9, 10, 11] but they mainly concentrate on predictive performance [7, 8, 9, 10].

[1] This work has been partially supported by Spanish Ministry of Education and Science and the European Investment Fund for Regional Development (FEDER) under grant TSI 2005-07876-C03-01

In addition, performance comparisons are rarely made using a useful cost-benefit analysis, i.e., latency reduction as a function of the traffic increase. As examples of some timid attempts, Dongshan and Junyi [7] compare the accuracy, the model-building time, and the prediction time in three versions of a predictor based in Markov chains. Another current work by Chen and Zhang [8] implements three variants of the PPM predictor by measuring the hit ratio and traffic under different assumptions.

Nanopoulos *et al.* [9] show a cost-benefit analysis of the performance of four prediction algorithms by comparing the precision and the recall to the traffic increase. Nevertheless, they ignore how the prediction performance affects the final user. Bouras *et al.* in [10] show the performance achieved by two configurations of the PPM algorithm and three of the n-most popular algorithm. They quantify the usefulness (recall), the hit ratio (precision) and the traffic increase but they present a low number of experiments, which make it difficult to obtain conclusions. In a more recent work [11] they also show an estimated upper bound of the latency reduction for the same experiments.

In this paper we propose and implement a cost-benefit methodology to perform fair comparisons of web prefetching algorithms from the user's point of view. Some experiments were performed to illustrate how we can evaluate the benefits of the prefetching.

The remainder of this paper is organized as follows. Section 2 describes the experimental environment used to run the experiments. Section 3 proposes a methodology to evaluate prefetching algorithms. Section 4 analyzes the experimental results of an example of application of the proposed methodology. Finally, Section 5 presents some concluding remarks.

2 Experimental Environment

2.1 Framework

In [12] we proposed an experimental framework for testing web prefetching techniques. In this section we summarize the main features of such environment and the configuration used to carry out the experiments presented in this paper.

The architecture consists of two main parts: the back end (server and surrogate), and the front end (client). The framework implementation combines both real and simulated parts in order to provide flexibility and accuracy.

The back end part includes the web server and the surrogate server. The framework emulates a real surrogate, which is used to access a real web server. We use the surrogate as a predictor. To this end, it adds new HTTP headers to the server response with the result of the prediction algorithms, as implemented in Mozilla. The server is an Apache web server set up to act as the original one. For this purpose, it has been developed a CGI program that returns objects with the same size and MIME type than those recorded in the traces.

The front end, or client part, represents the users' behavior exploring the Web with a prefetching enabled browser. To model the set of users that access

concurrently to a given server, the simulator is fed by using real traces. The simulator collects basic information for each request performed to the web server, then writes it to a log file. By analyzing this log at post-simulation time, all performance metrics can be calculated.

2.2 Workload Description

The behavior pattern of users was taken from two different logs. Traces A and B were collected during May 12^{th} 2003. They were obtained by filtering their accesses in the log of a Squid proxy of the Polytechnic University of Valencia. The trace A contains accesses to a news web server, whereas the trace B has the accesses to a student information web server. The main characteristics of the traces are shown in Table 1. The training length of each trace has been adjusted to optimize the perceived latency reduction of the prefetching.

2.3 Prefetching Algorithms

The experiments were run using three of the most widely used prediction algorithms in the literature: two main variants of the *Prediction by Partial Match* (PPM) algorithm [13, 7, 8, 9] and the *Dependency Graph* (DG) based algorithm [2, 9].

The PPM prediction algorithm uses Markov models of m orders to store previous contexts. Predictions are obtained from the comparison of the current context to each Markov model. PPM algorithm has been proposed to be applied either to each object access [13] or to each page (i.e., to each container object) accessed by the user [7, 8]. In this paper we implement the object-based version of the algorithm.

The DG prediction algorithm constructs a weighted dependency graph that depicts the pattern of accesses to the objects. The prefetching aggressiveness is controlled by a cutoff threshold parameter applied to the arcs weight.

Table 1. Traces characteristics

Characteristics	**Trace**	
	A	**B**
Year	2003	2003
Users	300	132
Page Accesses	2,263	1,646
Objects Accesses	65,569	36,837
Training length (accesses)	35,000	5,000
Bytes Transferred (MB)	218.09	142.49

2.4 Performance Indexes

The performance of the algorithms has been evaluated using the two main user related metrics [4], each one representing the cost and the benefit of the web prefetching. Both indexes are better as lower their value is.

- Latency per page ratio (L_p): The latency per page ratio is the ratio of the latency that prefetching achieves to the latency with no prefetching. The latency per page is calculated by comparing the time between the browser initiation of an HTML page GET and the browser reception of the last byte of the last embedded image or object for that page.
- Traffic Increase (ΔTr): The bytes transferred through the network when prefetching is employed divided by the bytes transferred in the non-prefetching case. Notice that this metric includes both the extra bytes wasted by prefetched objects that the user will never use, and the network overhead caused by the transference of the prefetch hints.

3 Methodology

The comparison of prefetching algorithms should be made from the user's point of view and using a cost-benefit analysis. Despite the fact that prefetching has been also used to reduce the peaks of bandwidth demand [14], its primary goal; i.e., the benefit, is usually the reduction of the user's perceived latency.

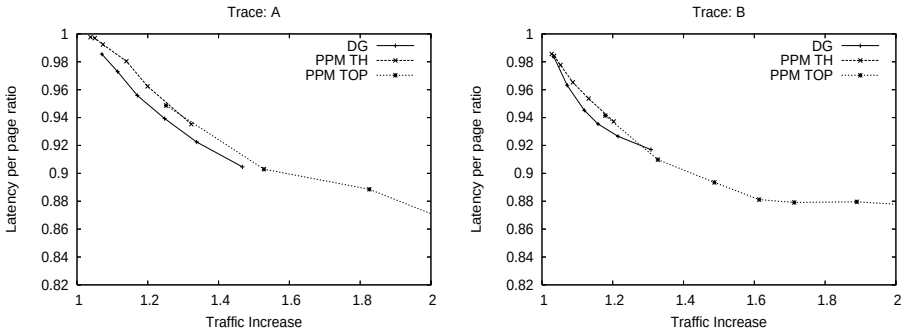
When predictions fail, prefetched objects waste user and/or server resources. Since in most proposals the client downloads the predicted objects in advance, the main cost of the latency reduction in prefetching systems is the network traffic increase. As a consequence, the performance analysis should consider the benefit of reducing the user's perceived latency at the cost of increasing the network traffic.

For comparison purposes, we have simulated systems implementing the above described algorithms. Each simulation experiment on a prefetching system takes as input the user behaviour and the prefetching parameters. The main results obtained are the traffic increase and the latency per page ratio values.

Comparisons of two different algorithms only can be fairly done if either the benefit or the cost have the same or close value. For instance, when two algorithms present the same or very close values of traffic increase, the best proposal is the one that presents less user perceived latency, and vice versa.

For this reason, in the examples shown in this paper the performance comparisons are made through curves that include different pairs of traffic increase and latency per page ratio for each algorithm. In order to obtain each point in the curve we have varied the aggressiveness of the algorithm, i.e., how much an algorithm will predict. This aggressiveness is controlled by a threshold parameter in those algorithms that support it (i.e., DG and PPM-TH) and by the number of returned predictions in the PPM-TOP.

A plot can gather the curves obtained for each algorithm in order to be compared. By drawing a line over the desired latency reduction in this plot,



(a) Trace A (b) Trace B

Fig. 1. Performance comparison between objects based algorithms. Each point in the curves represents a given threshold in PPM-TH and DG, while it represents a given amount of returned hints in PPM-TOP.

one can obtain the traffic increase of each algorithm. The best algorithm for achieving that latency per page is the one having less traffic increase.

4 Algorithms Comparison

Figure 1 shows the results for the algorithms described in Sect. 2.3. Each algorithm is evaluated in two situations each one using one of the two described workloads (i.e., A and B). The curves of each plot in DG and PPM-TH algorithms are obtained by varying the confidence threshold of the algorithms, from 0.2 to 0.7 in steps of 0.1. To make the curves of the PPM-TOP algorithm, the number of returned predictions are ranged from 1 to 9 in steps of 1, except for 6 and 8. Results for traffic increases greater than 2 are not represented in order to keep the plot focused on the area where the algorithms can be compared.

Figure 1(a) illustrates the performance evaluation of the algorithms simulating users who have 1 Mbps of available bandwidth and behave in accordance with the workload A. This plot shows that the DG algorithm achieves better performance than the others in the range in which it is evaluated, since its curve falls always below the ones of the PPM algorithms.

Figure 1(b) shows that the algorithms exhibit minor performance differences when using the trace B. DG algorithm slightly outperforms the others in all its range with the only exception of the most aggressive threshold (i.e., th=0.2), in which the PPM-TOP algorithm achieves a slightly higher latency reduction with the same traffic increase.

5 Conclusions

A large amount of research works has focused on web prefetching. However, comparative studies are rare and usually ignore the user's point of view. In this paper we have described a cost-benefit methodology to evaluate and compare prefetching algorithms from the user's point of view.

Using the proposed methodology, three prediction algorithms have been implemented and compared. Experimental results show that DG algorithm slightly outperforms the PPM-TH and the PPM-TOP algorithms in most of the analyzed cases. However, the aggressiveness (and, consequently, the latency reduction) of the DG is more limited than the PPM-TOP one. For this reason, when prefetching is not desired to be very aggressive, DG achieves the best cost-effectiveness.

References

[1] Schechter, S., Krishnan, M., Smith, M.D.: Using path profiles to predict http requests. In: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia (1998)
[2] Padmanabhan, V.N., Mogul, J.C.: Using predictive prefetching to improve World-Wide Web latency. In: Proceedings of the ACM SIGCOMM '96 Conference, Stanford University, USA (1996)
[3] Cohen, E., Kaplan, H.: Prefetching the means for document transfer: a new approach for reducing web latency. Computer Networks **39** (2002)
[4] Domènech, J., Gil, J.A., Sahuquillo, J., Pont, A.: Web prefetching performance metrics: A survey. Accepted to be published in Performance Evaluation (2006)
[5] Domènech, J., Sahuquillo, J., Pont, A., Gil, J.A.: How current web generation affects prediction algorithms performance. In: Proceedings of SoftCOM Int. Conf. on Software, Telecommunications and Computer Networks, Split, Croatia (2005)
[6] Duchamp, D.: Prefetching hyperlinks. In: Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems, Boulder, USA (1999)
[7] Dongshan, X., Junyi, S.: A new markov model for web access prediction. Computing in Science and Engineering **4** (2002)
[8] Chen, X., Zhang, X.: A popularity-based prediction model for web prefetching. IEEE Computer **36** (2003)
[9] Nanopoulos, A., Katsaros, D., Manolopoulos, Y.: A data mining algorithm for generalized web prefetching. IEEE Trans. Knowl. Data Eng. **15** (2003)
[10] Bouras, C., Konidaris, A., Kostoulas, D.: Efficient reduction of web latency through predictive prefetching on a wan. In: Proceedings of the 4th Int. Conf. on Advances in Web-Age Information Management, Chengdu, China (2003)
[11] Bouras, C., Konidaris, A., Kostoulas, D.: Predictive prefetching on the web and its potential impact in the wide area. World Wide Web **7** (2004)
[12] Domènech, J., Pont, A., Sahuquillo, J., Gil, J.A.: An experimental framework for testing web prefetching techniques. In: Proceedings of the 30th EUROMICRO Conference 2004, Rennes, France (2004)
[13] Sarukkai, R.: Link prediction and path analysis using markov chains. Computer Networks **33** (2000)
[14] Maltzahn, C., Richardson, K.J., Grunwald, D., Martin, J.H.: On bandwidth smoothing. In: Proceedings of the 4th International Web Caching Workshop, San Diego, USA (1999)