A CLIENT-ORIENTATED DYNAMIC WEB SERVER

Cristina Hava Muntean, Jennifer McManis, John Murphy[1] and Liam Murphy[2]

Abstract

The cost of computer systems has decreased continuously in recent years, leading to an exponential growth in the number of computer users. In such an environment, more and more Web servers have been created offering many types of information. As a result Internet traffic has grown significantly, affecting the quality of the services offered by the Web servers. We propose a new approach for designing Web servers, which takes into account client requirements and constraints, and whose implementation is based on Java servlet and applet technology. This client-orientated Web server classifies each client into one of a number of pre-defined categories. The Web page generated for a client then depends on the client's current category. A Web page generated in this way may differ from one generated for another client in its content, number of images, graphic design and structure.

1. Introduction

The greatest proportion of traffic in the World Wide Web is generated by HTTP transactions between clients and Web servers. Clients interested in the data access the servers and request information from them. The servers deliver the data to the clients through the Internet. As the amount of information delivery through the Internet has rapidly grown in the last years the immediate effects were increased delays in accessing the data and the overload of the network. During peak periods Web servers may have to service many requests per second and sometimes reject some of the requests in order to satisfy the others. Some current commercial Web servers don't offer good performance when they are overloaded and they do not have mechanisms to adapt to load conditions in order to prevent the overload. Some Web servers take as much processing power to reject a connection as it would to server it. Considerable research has been done on improving server performance (see Section 2).

In this paper we propose a Web server, which provides an adaptive mechanism for generating the Web pages. The mechanism is based on the Web server's performance as seen by the client. It attempts to ascertain the client's requirements and capabilities, and based on their needs, it generates different Web pages. The client is provided with a customised Web page and this may reduce the load on the server, since the pages provided might be less resource intensive than one designed to satisfy all clients.

The layout of this paper is that we present some previous approaches to Web server performance enhancement in Section 2. Our proposed mechanism for a client-orientated Web server, and the criteria, which could be used, are described in Section 3. To prove the feasibility and the usefulness of this idea we implemented a test system and preliminary results are presented in Section 4. We conclude with possible future enhancements in Section 5.

[1] Performance Engineering Laboratory **http://www.eeng.dcu.ie/~pel**
School of Electronic Engineering, Dublin City University, Dublin 9, Ireland
havac@eeng.dcu.ie, mcmanisj@eeng.dcu.ie, murphyj@eeng.dcu.ie

[2] Department of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland
liam.murphy@ucd.ie

2. Previous Work

Because the Web server's performance has an important impact on the popularity of the Web site, several projects have studied different ways to improve it. Thus, some adaptive Web servers, which optimize the performance, have been proposed. Some of the projects are based on the needs of individual clients and allow the users to customize the site for themselves by describing their interest. The Microsoft Network (MSN) offers us the possibility to create a personalised home page with customized information. Thus, every time we visit the home page the latest news in our interested domain is displayed. Other proposed solutions are to try to guess the user's next step, what link from the current page will be accessed, or even the client's goal, what kind of information is being sought. The Web Watcher [1] project tries to predict what links will be followed from the current page starting from the client's specified interests. The predicted links will be highlighted and displayed on the top of the Web page. At the end of the navigation the clients are asked if they found what they were looking for and the answers are stored. The Avanti [2] project customizes dynamically the content of the Web page based on user's needs and tastes and predicts both the user's next step and their eventual goal. The previous projects adapt the site's presentation for each individual user according to their preferences. Perkowitz and Etzioni [3] proposed another solution for adaptive Web servers. The system allows the server to learn what all the users that access the site want and to automatically adapt the site's content. Thus, it makes the Web site easier to use for everyone, including the clients who are accessing the site for the first time.

All those projects try to learn from clients' behavior and to adapt the content of the Web pages, but they don't improve the Web server performance during the peak period or when the server or network is overloaded. Thus, to avoid the rejection of some requests when the server is overloaded, a multicast solution was proposed [4]. The main idea is that to use a multicast mechanism for distributing commonly requested pages, thus reducing the bandwidth consumption on the server output links. Also, to extend the basic scheme of Asynchronous Multicast Push (AMP) and to accommodate service differentiation among clients with respect to the maximum holding time experienced by their requests. Abdelzaher proposed a new approach to reducing overload using "content adaptation" [5]. The idea is to use multiple copies (pre-processed and pre-stored) of the Web pages that differ in quality and size. Based on a measure of the current degree of server utilization, the appropriate version of the Web page is sent. However user preferences or requirements are not considered.

3. The Mechanism of the Client-Orientated Web Server

In this section we discuss a new approach for developing a dynamic Web server. It takes into account performance as the clients see it and serves a different Web page to them depending on their needs. This client-orientated Web server classifies each client into one of a number of pre-defined categories according to some criteria (see Section 3.3). The Web page generated for each category may differ in content, number of images, graphic, design and structure, as shown in Figure 1.

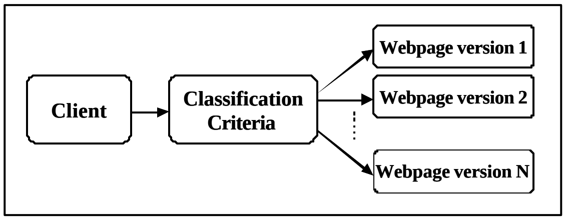


Figure 1. The mechanism of the Web server

The server proposed in this paper is implemented using servlet and applet technology [6]. The servlets and applets allow the creation of Web pages "on the fly". They have the possibility to check both the IP address of the client machine and to set cookies remotely. Using this new technology there are two possible approaches to store the classification information for each client:

- Centralized or Distributed

3.1. Centralized Database

Using this mechanism the classification information for each client is stored and updated in a centralized database at the server side. Thus, at every access made by the client to the site it will be classified according to some criteria in one of the pre-defined categories and the classification information will be added to the database situated on the server or on another computer behind the server as shown in Figure 2. The main advantage for this solution is the security of the stored information with firewalls available to protect them.

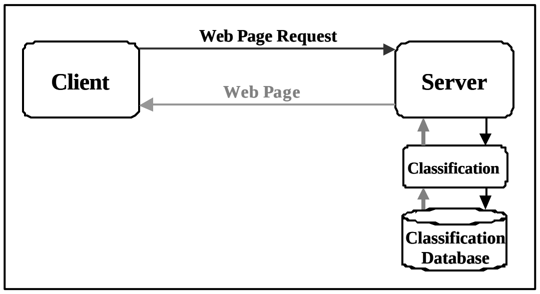


Figure 2. The Web server mechanism with centralized database

Unfortunately, when the number of different clients that access the site is very large the space and the resources consumed for updating the database can be large and might affect the server performance. If this is the case then a better structure of the database is required or a better algorithm for searching in the database has to be implemented or it is necessary to host the database separately in a new machine, thus making the task at the Web server easier to handle.

Another issue is that in many instance there will be multiple users behind the same proxy. In this situation the server sees all the users as having the same IP address. Thus the IP address of the client machine cannot be used as the main factor to make differences between the users and also to identify each of them in our database. One possible solution would be to ask the client to enter a username before accessing the site, however this has an impact on the user friendliness of the site. However one shortcoming of this work is that for the client identity to be known the client must access the Web server from the same machine.

3.2. Distributed Database

The main idea behind this approach is the cookie mechanism. Cookies are small bits of textual information sent by the Web server to a Web browser. The browser stores the message in a text file called "cookie.txt" and returns it unchanged when later visiting the same Web site. Only the Web server that originally sent the cookie can read the information stored within a cookie. Web servers cannot read cookies sent by other Web servers. The main purpose of cookies is to identify users and possibly prepare customized Web pages for them.

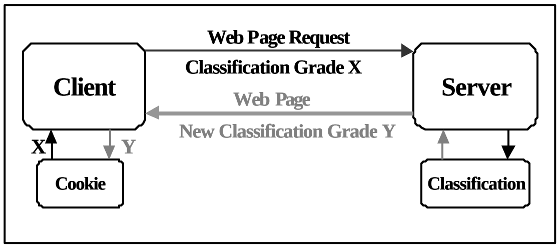


Figure 3. Distributed database using cookies mechanism

Making use of the cookies mechanism, the classification information used by our system could be stored and updated remotely at each client. Thus, the values of the parameters used for the classification algorithm can be saved on the client machine as shown in Figure 3.

The main advantage of the cookies solution is that it saves space at the server, since the database is distributed among the clients. This solution also solves the problem of differentiating between users behind the same proxy. The server can set a unique number on each computer in the cookies file at the first visit of the client. At subsequent visits this information will be sent automatically to the server by the Web browser. Thus, the users are not obliged to enter a username every time when they visit the site. Unfortunately from security point of view the cookies, stored as text files on the clients' machines, allow the users to modify and even remove their content.

3.3. Classification Criteria

The client-orientated Web server classifies each client into one of a number of pre-defined categories, each time they access the site. The generated Web page depends on the client's current category. To classify a client we propose some criteria that take into account client requirements and server performance as the client sees it. Possible classification criteria include:

- Number of the accesses to the Web server made by the client
- Download time of the initial Web page
- Load on the Web server during the time the client accesses the server
- Type of connection used by the client
- IP address or domain name of the client's machine
- Enabled multimedia options (e.g. animation, sounds, images) on the client
- Type and version of browser client uses to access the Web site
- Existence of Java Virtual Machine on the browser

All the classification criteria can be recomputed at every access. Thus, when the client accesses a link from the current Web page or revisits the site, the classification factors will be recomputed and stored in the database or cookies. In this way the client can be dynamically transferred into a different category. A weighted algorithm for classification has to be implemented to determine the category the client belongs to according to the measured classified factors.

4. Preliminary Results

To demonstrate the feasibility and the usefulness of this idea an application has been built which sends Web pages to the users with different structures depending on the client classification, as shown in Figure 4. The classification criterion used is the number of accesses made by the client to the Web server. The current test setup used in the laboratory was with a Pentium III 800MHz as the Web server with an operating system of Windows NT together with Java WebServer version 2.0 application. Three categories were defined for these testes with a different Web page structure for each.

The Web server application was implemented using servlet and applets technology. The servlets

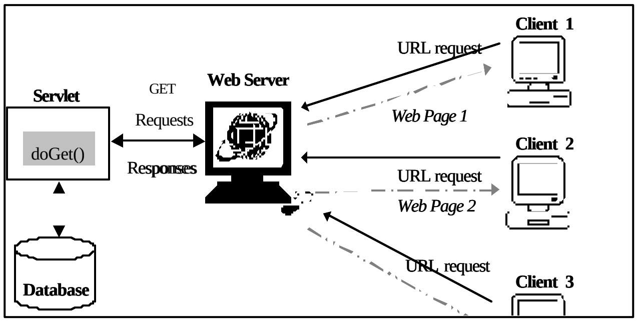
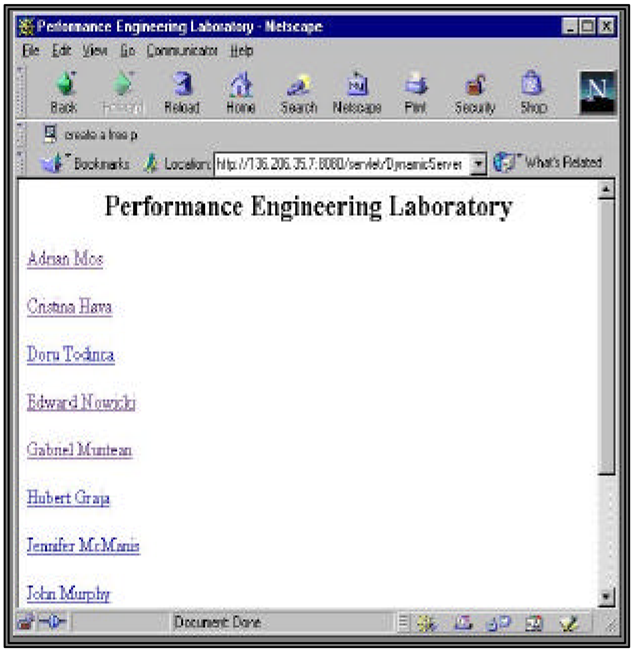


Figure 4. Web Server Application Structure

represent the latest idea in Java programming which expands the Java language from developing client-side Web content, represented by applets, to server-side development. They allow the creation and modeling of the structure and content of the Web page. The main advantages of servlet technology include portability, safety, flexibility and integration with the server. They are supported on all major platforms and work with all the mainstream Web servers.


Performance Engineering Laboratory - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
create a free p
Bookmarks Location: http://136.206.35.7:8080/servlet/DynamicServer
What's Related
Document: Done

Performance Engineering Laboratory

Adrian Mos

Cristina Hava

Doru Todinca

Edward Nowicki

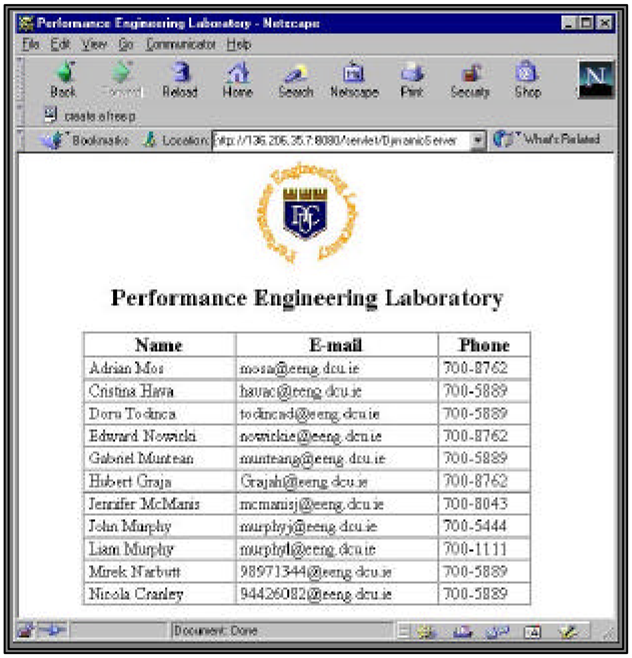
Gabriel Muntean

Hubert Graja

Jennifer McManis

John Murphy

Figure 5. Web page for Category 1

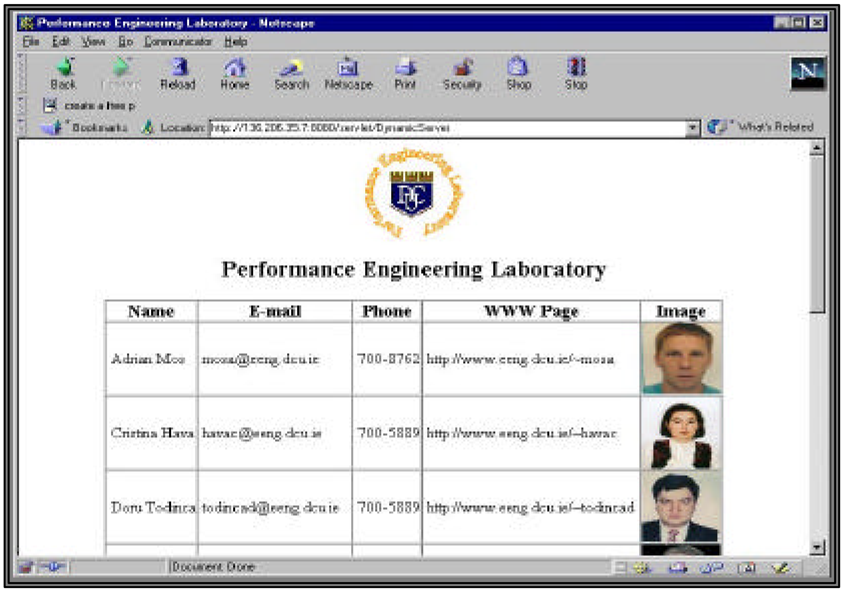

Performance Engineering Laboratory - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
create a free p
Bookmarks Location: http://136.206.35.7:8080/servlet/DynamicServer
What's Related
Performance Engineering Laboratory
Document: Done

Performance Engineering Laboratory

Name	E-mail	Phone
Adrian Mos	mosa@eeng.dcu.ie	700-8762
Cristina Hava	havac@eeng.dcu.ie	700-5889
Doru Todinca	todincad@eeng.dcu.ie	700-5889
Edward Nowicki	nowickie@eeng.dcu.ie	700-8762
Gabriel Muntean	munteang@eeng.dcu.ie	700-5889
Hubert Graja	Grajah@eeng.dcu.ie	700-8762
Jennifer McManis	mcmanisj@eeng.dcu.ie	700-8043
John Murphy	murphyj@eeng.dcu.ie	700-5444
Liam Murphy	murphyl@eeng.dcu.ie	700-1111
Mirek Narbutt	98971344@eeng.dcu.ie	700-5889
Nicola Cranley	94426082@eeng.dcu.ie	700-5889

Figure 6. Web page for Category 2

The application in this project generates three different versions of the main page for DCU's Performance Engineering Laboratory. The simplest structure is a list of the persons involved in this lab and a link to their Web page as shown in Figure 5. As can be seen no pictures were added as in general images represent the biggest part of the download time of the whole Web page [7]. This simple structure is generated for the clients from the first category. For the second category, a more complex Web page is generated where the page does not have many images but some extra text and a different organization of the information from the Web page is added, as shown in Figure 6. Finally for the third category, all the structure and all the components of the Web page are generated as shown in Figure 7. The page has a lot of images, which will


Performance Engineering Laboratory - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
create a free p
Bookmarks Location: http://136.206.35.7:8080/servlet/DynamicServer
What's Related
Performance Engineering Laboratory
Document: Done

Performance Engineering Laboratory

Name	E-mail	Phone	WWW Page	Image
Adrian Mos	mosa@eeng.dcu.ie	700-8762	http://www.eeng.dcu.ie/~mosa	
Cristina Hava	havac@eeng.dcu.ie	700-5889	http://www.eeng.dcu.ie/~havac	
Doru Todinca	todincad@eeng.dcu.ie	700-5889	http://www.eeng.dcu.ie/~todincad	

Figure 7. Web Page for Category 3

affect the waiting time of the client.

In this test example the assumption is that there is already a database with the clients' classification by having the database stored locally on the server machine. Every time a client makes an access, the client's category is checked in the database and the Web page corresponding to that category is sent to the user. The classification factor is updated and there is a check to see if the client's category should be changed. In this simple test it is found that the correct Web page is returned for the client's current category, and the category is correctly updated when the threshold number of visits is reached.

5. Conclusions and Future Work

A lot of research has been done in order to improve Web server performance and there are many solutions possible to create adaptive Web servers, some of which have been implemented. Each of them has been tried with different points of view in mind and the aim is to try and improve the quality of the services offered by the servers.

This paper has proposed a new approach for developing a Web server, which dynamically adjusts the composition of the site it hosts, according to the performance of the server as the clients see it. The clients who access the site are classified into categories according to some criteria. This classification is dynamically updated at every access. The criterion currently implemented is the number of accesses made by the client to the Web page.

In the future there is an intention to extend this work to measure the download time of the Web page as a classification criterion, as well as other criteria. It is possible to consider the use of a weighted algorithm for classification as well. In this paper a centralized database was used to store the classification information, however in the future a distributed database will be implemented and a comparison between the two methods will be carried out.

References

1. T. Joachims, D. Freitag, T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web", in Proceeding of IJCAI'97, Nagoya, Japan 1997, pp. 770-775.
2. J. Fink, A. Kobsa, A. Nill, "User-oriented adaptivity and adaptability in the AVANTI Project", in Designing for the Web: Empirical Studies, Microsoft Usability Group, Redmond, 1996
3. M. Perkowitz, O. Etzioni, "Towards Adaptive Web Sites: Conceptual Framework and Case Study", Artificial Intelligence 118 (2000), pg. 245-275.
4. P. P. White, J. Crowcroft, "WWW Multicast Delivery with Classes of Service", in Proceeding of HIPPARCH'98, University College, London, June 1998.
5. T. F. Abdelzaher, N. Bhatti, "Web Server QoS Management by Adaptive Content Delivery", HP Labs 1999 Technical Reports, HPL-1999-161.
6. J. Hunter, W. Crawford, "Java Servlet Programming", O'Reilly, 1998.
7. C. Hava Muntean, J. McManis, J. Murphy, "The Influence of Web Page Images on the Performance of Web Servers", 8th IEEE International Conference on Networking, ICN'2001, Colmar, France, July 2001.