An Adaptive Web Server Application

Cristina Hava Muntean

Performance Engineering Laboratory,
School of Electronic Engineering, Dublin City University
Glasnevin, Dublin-9, IRELAND
havac@eeng.dcu.ie

Jennifer McManis

Performance Engineering Laboratory,
School of Electronic Engineering, Dublin City University
Glasnevin, Dublin-9, IRELAND
mcmanisj@eeng.dcu.ie

John Murphy

Performance Engineering Laboratory,
School of Electronic Engineering, Dublin City University
Glasnevin, Dublin-9, IRELAND
murphyj@eeng.dcu.ie

Liam Murphy

Department of Computer Science,
University College Dublin,
Belfield, Dublin-4, IRELAND
Liam.Murphy@ucd.ie

ABSTRACT

The Web is continually increasing in size and importance and so does the interest of the Web server administrators to maximize their users' satisfaction, while efficiently using the existing resources. The paper proposes an adaptive Web server application that tries to increase the performance of the Web server that hosts a Web site, as seen from the clients' point of view. The adaptiveness is based on the customization of the Web site in a manner that emphasizes the interests of the clients. Our tests show that both the user satisfaction and the server performance are improved.

KEYWORDS:

Web server performance, customer satisfaction, adaptive Web server, and performance parameters

1. INTRODUCTION

The usage of Web sites can vary widely during a day making difficult to predict the exact number of requests during any given time. It is therefore important to analyze how the Web sites perform under larger than anticipated loads. Web server performance has a large impact on the popularity of a given Web site and therefore extensive effort has been done to improve it, especially for the sites that service a high number of requests.

Our work focuses on the improvements that bring both a better Web server performance and a greater increase in customer satisfaction with the visited Web site by introducing an adaptive Web server application which makes use of a customizable Web site, according to the traffic-related parameters measured at the client side. By simulating a variable number of clients while monitoring the server activity, we show that the adaptive Web server performs better than a non-adaptive one under different loading conditions.

Two main directions are taken by the research that focuses on improving the Web server's performance and therefore increasing customer satisfaction with a site.

Firstly, some adaptive solutions allow the customization of the Web sites according to the clients' interests. Microsoft Network (MSN) and Yahoo! display the latest news from user-chosen domains of interest. Web Watcher **[1]** tries to predict what links will be followed from the current page, while Avanti **[2]** dynamically customizes the content of the Web page based on the user's needs and tastes, and predicts the user's next step and the eventual goal. Perkowitz and Etzioni's **[3]** adaptive Web server uses feedback from all the users to modify the site’s content.

The second direction seeks to improve the client satisfaction with the Web server by increasing its performance. A multicast solution **[4]** for distributing commonly requested pages, which also accommodates service differentiation among clients and the "content adaptation" **[5]** that serves one out-of-multiple pre-existing different size versions of the requested Web pages reduce the bandwidth consumption on the server output links.

2. ANALYSING USER SATISFACTION

We propose a methodology that identifies different users classes. A class is represented by groups of users with similar characteristics and values for the Web site performance parameters. The structure of classes may dynamically change depending on current user population.

Starting from a Web site structure we try to identify those Web site factors related to features of the site, which might affect user satisfaction.

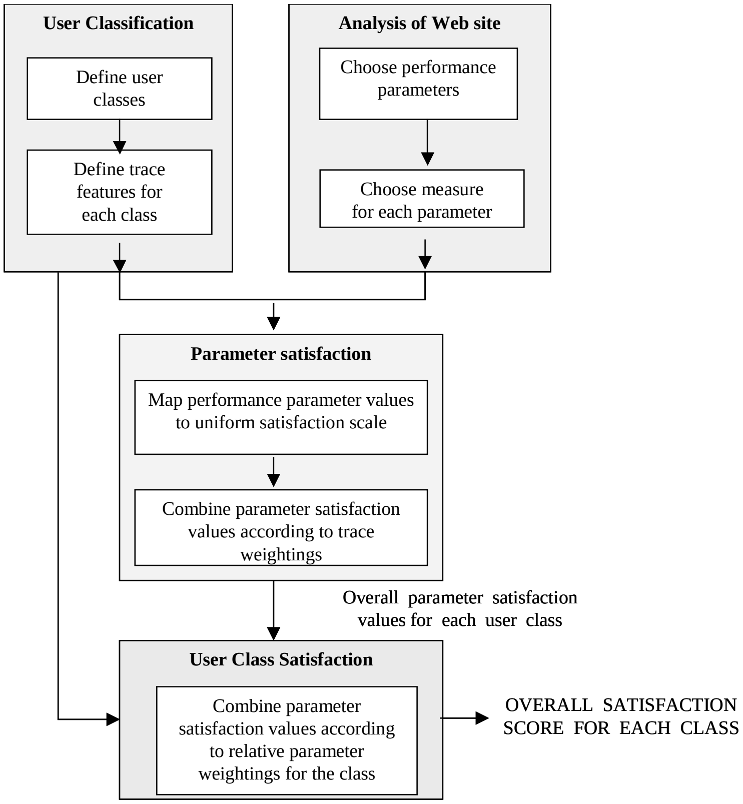

User Classification
Define user classes
Define trace features for each class
Analysis of Web site
Choose performance parameters
Choose measure for each parameter
Parameter satisfaction
Map performance parameter values to uniform satisfaction scale
Combine parameter satisfaction values according to trace weightings
Overall parameter satisfaction values for each user class
User Class Satisfaction
Combine parameter satisfaction values according to relative parameter weightings for the class
OVERALL SATISFACTION SCORE FOR EACH CLASS

Figure 1. Proposed methodology for analysis of user satisfaction

Having identified our user classes and the characteristics of the Web site that will affect the user satisfaction, we seek to measure user satisfaction in a consistent and quantifiable way. In figure 2 we summarize the steps needed to measure user satisfaction.

3. THE ARHITECTURE OF THE WEB SERVER APPLICATION

To customize a Web site means to offer different versions of the site to different users. We divide users into classes, each class receiving a version of the site optimized to their requirements. The classification of the users is based on a wide variety of parameters including both clients' capabilities and server performance measures, as the clients perceive them. By serving smaller versions of a site to users instead of a bulky 'one-size-fits-all' version, we can increase overall user satisfaction with a Web site and, at the same time, the server performance.

Some of the parameters used for the classification are: download time of Web pages, client's bandwidth, the client's connection type, Web server's load, the type of the browser's plug-in components. The parameters' values are stored in a centralized database situated on the server side. The classification information is computed at every access done by the user and therefore the class the user belongs to can be dynamically changed.

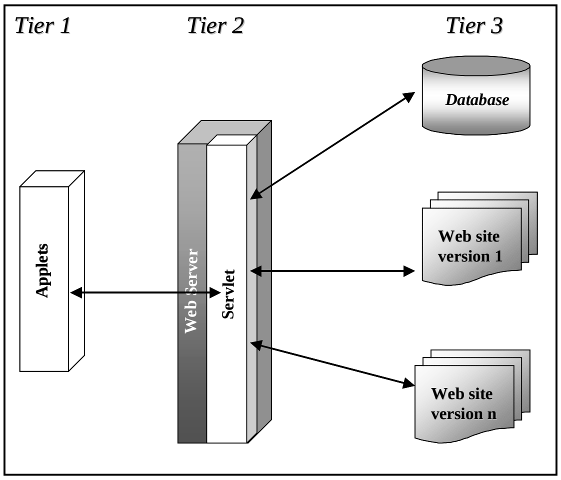


Figure 2. The Three-Tier Architecture for our Web application

Three -Tier Architecture

Our Web server application is based on applets-servlets technology. The application has a Three-Tier Architecture shown in Figure 2. The servlets act as gateway for the back-end system and control the access to the system resources. The back-end system consists of all the versions of the Web site and the database that saves the client-related performance parameters. These can be situated on another machine, behind a firewall if desired.

The servlets communicate with back-end resources looking up and updating the information from a back-end database, request/offer information from/to the applet, perform calculations and provide Web pages. The applets are used to measure different performance parameters on the client machine and to send the results back to the servlets.

Tools For Deploying And Testing Our Application

In order to deploy our adaptive application we have developed a simple Web site using the Adobe GoLive tool. The Web site consists of different versions of pages containing the same type of information, but its complexity, the level in the hierarchy where it is located and the sizes of the Web pages may differ.

Sun's Java Web-Server 2.0 was used to capture the HTTP requests and to dispatch them to our application. The database is implemented using a MySQL Server and it is accessed using the Java Database Connectivity (JDBC) technology.

In order to generate the Web users that visit the Web site and to do the performance measurements we used OpenSTA **[6]** (Figure 3). This tool is used to measure both some performance parameters at the client (e.g. download time), and at the server sides (e.g. CPU load, server latency time, rejected requests, throughputs, number of HTTP responses per second sent by the server, etc.).

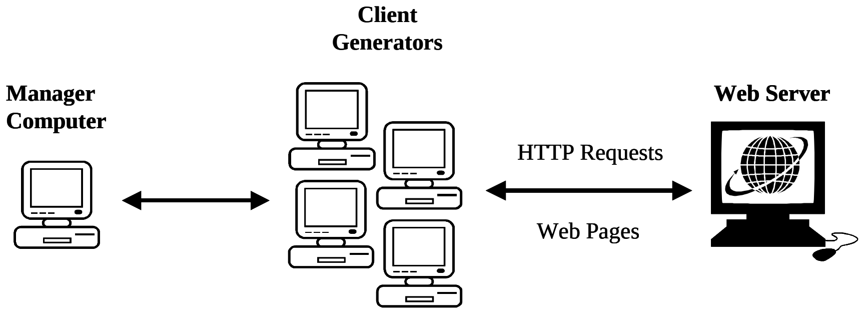
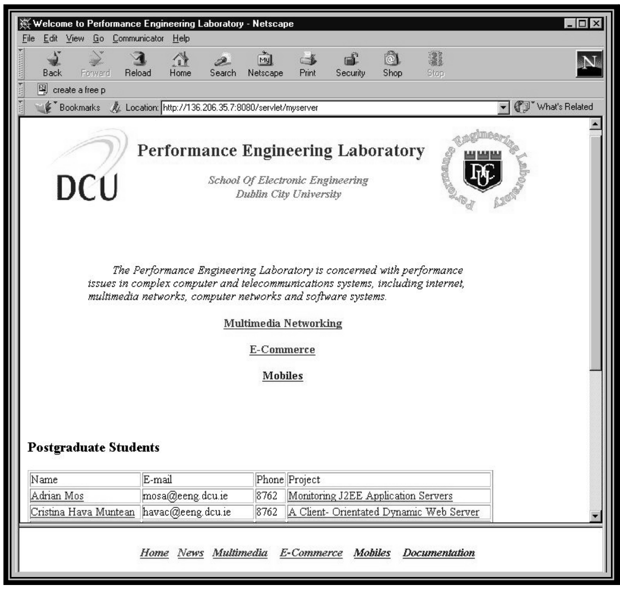


Figure 3. OpenSTA architecture

4. TEST RESULTS FOR OUR SIMPLE ADAPTIVE WEB SERVER APPLICATION

As a test Web site, we built two versions of a simplified Web site of our research group (Figures 4 and 5). One version is suitable for accesses by external users to our research group, while the second version is better designed for internal access (i.e. by the members of our group). The database stores the information about the classified clients.


Welcome to 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/myserver
What's Related
DCU

Performance Engineering Laboratory

School Of Electronic Engineering
Dublin City University

The Performance Engineering Laboratory is concerned with performance issues in complex computer and telecommunications systems, including internet, multimedia networks, computer networks and software systems.

Multimedia Networking

E-Commerce

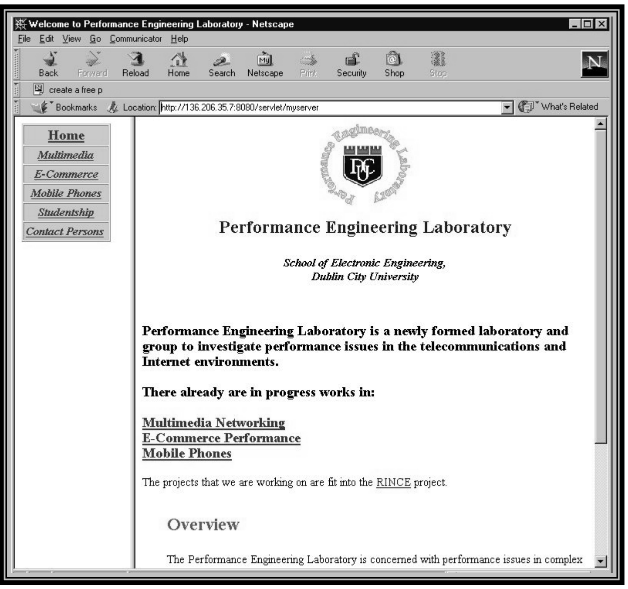
Mobiles

Postgraduate Students

Name	E-mail	Phone	Project
Adrian Mos	mosa@eeng.dcu.ie	8762	Monitoring J2EE Application Servers
Cristina Hava Muntean	havac@eeng.dcu.ie	8762	A Client- Orientated Dynamic Web Server

Home *News* *Multimedia* *E-Commerce* *Mobiles* *Documentation*

Figure 4. First version of our test Web site suitable for internal users


Welcome to 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/myserver
What's Related
Home
Multimedia
E-Commerce
Mobile Phones
Studentship
Contact Persons

Performance Engineering Laboratory

School of Electronic Engineering,
Dublin City University

Performance Engineering Laboratory is a newly formed laboratory and group to investigate performance issues in the telecommunications and Internet environments.

There already are in progress works in:

Multimedia Networking
E-Commerce Performance
Mobile Phones

The projects that we are working on are fit into the RINCE project.

Overview

The Performance Engineering Laboratory is concerned with performance issues in complex

Figure 5. Second version of our test Web site suitable for external users

We tested the adaptive and non-adaptive versions of our Web server using pre-defined navigation tasks, with different numbers of clients executing them in parallel. We measured the *HTTP monitored bytes/second* and *HTTP responses/second* parameters. The first parameter shows the total number of bytes per second outputted by the Web server during the simulation. The second one indicates the total number of responses per second the Web server had to handle (Figures 6 and 7).

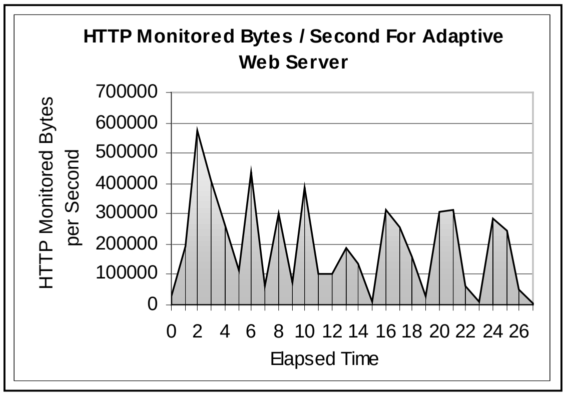


Figure 6. HTTP monitored bytes/second vs. elapsed time

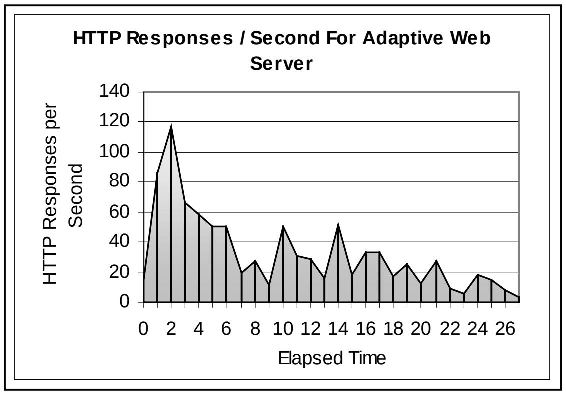


Figure 7. HTTP responses / second vs. elapsed time

In order to compare the adaptive and non-adaptive versions of our sites, we perform the above test on both of them, increasing the number of the Web users simulated in parallel (Figures 8 and 9). We noticed that the total number of *HTTP requests* is smaller when the clients are classified than when all the users were treated alike. Similarly, the number of *HTTP monitored bytes* is bigger when the users are not classified. This is because some users have not only to visit more Web pages to find the desired information, but also the downloaded components were larger and more numerous.

In all cases, the adaptive server is subjected to a smaller workload than the non-adaptive server. However, looking to the growth of *HTTP monitored bytes* for adaptive and non-adaptive Web site with the number of clients, we noticed that the difference between them decreases with the increasing of the number of clients due to the Web server's physical limitation of the maximal number of bytes/second sent.

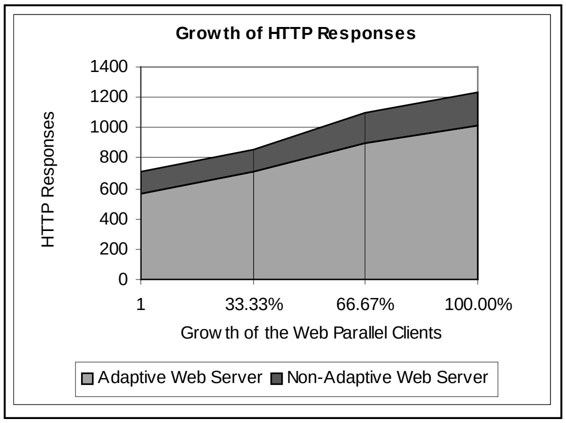


Figure 8. Growth of HTTP responses with the number of parallel clients

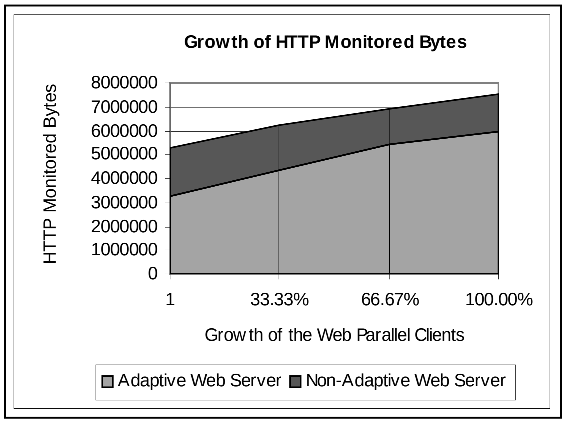


Figure 9. Growth of HTTP monitored bytes with the number of parallel clients

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this article we discussed a new design of a Web Server application, which seeks to improve both the Web user satisfaction and Web server performance. The design of the Web server is based on the server performance, as the Web users perceive it. Improving the server performance, the users' satisfaction can be increased. The test results presented in the paper analyze the improvements of performance on the server side only. We monitored some of the server performance indices that showed some improvements with our adaptive Web server. More investigations have to be done on other performance parameters especially when the Web server is overloaded and requests are rejected. Further work is also needed on quantifying user satisfaction and identifying relevant performance parameters.

REFERENCES

1. T. Joachims, D. Freitag, T. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. *Proceedings 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. *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), pp. 245-275.
4. P. P. White, J. Crowcroft. WWW Multicast Delivery with Classes of Service. *Proceedings of HIPPARCH'98*, University College, London, June 1998
5. T. F. Abdelzaher, N. Bhatti. Web Server QoS Management by Adaptive Content Delivery. *Hewlett-Packard Laboratories Technical Report HPL-1999-161*. http://www.hpl.hp.com/techreports/2000/HPL-1999-161.html.
6. OpenSTA. http://opensta.com/