A New Dynamic Web Server

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

[1] Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Glasnevin, Dublin 9, Ireland

[2] Department of Computer Science, University College Dublin, Dublin 4, Ireland

{havac, mcmanisj, murphyj}@eeng.dcu.ie, liam.murphy@ucd.ie

ABSTRACT

The growth of traffic on the Internet and the explosion in the number of Web sites created in recent years make Web server performance an important issue for Web site designers. To improve the server's performance, it is necessary to determine the main factors affecting it before proposing new solutions for Web server design. Here we first present some experimental results on factors which influence Web server performance. We show how the number of concurrent clients accessing the server and the overall network traffic dynamics affect the performance. The details of a Web page's composition are also studied to determine their effect on performance. Then we describe a new approach for developing a Web server, in which the server takes its performance as the clients see it into account and dynamically generates the requested Web pages. Their content depends on the traffic conditions and client capabilities. Some results are described to show the feasibility of our proposed design.

I. INTRODUCTION

The Internet world consists of millions of nodes, each of them a computer. Depending on their role, they can be servers or clients. Servers store the information, listen for requests, and deliver data. Clients interested in the data access the servers and request information from them. The data is delivered to the clients through the Internet.

The above is a very simple scenario of an Internet exchange of information. To allow such an exchange both the servers and the clients have to “understand” each other. Thus special sets of rules, or *protocols*, have been established in order to permit different types of information exchange. Some examples are FTP (File Transfer Protocol), which permits file transfer over the Internet; HTTP (Hypertext Transfer Protocol), which allows hypertext transmission of files; POP (Post Office Protocol); and SMTP (Simple Mail Transport Protocol), which is used for sending and receiving e-mails.

HTTP is one of the most widespread protocols, because it allows both easy access and navigation from/to documents. A special document format is needed to facilitate the reading of Web documents. HTML was the first popular document format, but it has proved insufficient for the increasing variety of document types users expect. Java and Visual Basic Scripts embedded into documents have already expanded the limits of the HTML language. Using Java applets as part of a document was a step forward. They run on the client computer after the download of the page they are part of. Unfortunately, running on the client, they cannot make use of the information stored at the server location. Java servlets address this problem. The servlets run on the Web server machine and the applets are sent to the client and run by the browser (using Java Virtual Machine). The servlet and the applet can communicate with each other using standard methods.

Recently a new technology (Java Server Pages) appeared offering the possibility of generating Web pages dynamically using servlets and applets. JSP technology allows the mixing of regular static HTML pages with dynamically generated content from the servlets or applets [1].

The increase in complexity of the techniques for designing Web sites, combined with the addition of more and more types of components to Web pages, has had an impact on both network and server performance. The immediate effects are increased delays in accessing the documents and overloading of the network. A lot of researchers are trying to find out the main factors which influence performance, and are proposing new solutions for a better structure of Web sites and increased performance for Web servers, especially during peak periods when the servers have to service more requests per second.

This paper analyses some of the factors which influence Web server performance in Section II. It also proposes a new approach for developing a Web server in Section III, which takes its performance as the clients see it into account. The server dynamically generates different content for the visited Web pages, adapting to the traffic conditions and client capabilities. Previous approaches along these lines are briefly described in Section IV, and some Conclusions are presented in Section V.

II. SOME FACTORS WHICH AFFECT WEB SERVER PERFORMANCE

Web pages used to be text-only with sizes on the order of hundreds of kilobytes. Nowadays, Web pages have become much more complex. A lot of new types of components have been added to the basic Web page, such as static and animated pictures, sounds, dynamically generated pages, and multimedia components. These new components have increased the total size of some Web pages to megabytes. Complex Web pages may be more attractive to clients, but are also more resource-intensive to send and retrieve. At the same time, both the number of Web sites and the number of users continue to increase dramatically. The immediate effects are increased delays in accessing documents due to both server and network overloading. As a result fewer users are able to access Web site information in a given time period and these users experience increasing delays.

Much research effort has gone into alleviating network and server loading. Both network traffic and server performance has been considered. The latter can be considered from either the client perspective [2] or the server perspective [3,4]. Different performance indices have been defined and measured, such as download time, number of simultaneous clients accepted, minimum/average/maximum transaction response time, and service availability over time.

In our opinion, the most important perspective on server performance is client perception. Thus we have implemented a tool for monitoring Web servers, which simulates various types and numbers of accesses and computes and evaluates some client-orientated performance indices [5]. In particular, we concentrate on the download time seen by the client, since this is closely correlated with client satisfaction. Our monitoring tool can analyse both a single connection to the Web server at a time, and multiple clients accessing the same server at the same time. Detailed experimental results are presented in [6].

We now summarize some of the results from [5,6,7]. It is demonstrated that server and network loadings affect download time. It is also shown that the number of images, as well as their overall size, has an impact on download time. Finally, other potential influences on download time are discussed, as well as some suggested improvements.

A. The Web Server Load

Our experimental results suggest that download times increase rapidly with the increasing number of the clients accessing the server in parallel (Figure 1). For some servers, such as Server D, a significant increase can be observed when there are 100 clients accessing the same page concurrently. Other servers show a much smaller reaction to 100 parallel clients, presumably due to server power. However, it is reasonable to assume that for some loading these sites will also show a degradation in performance.

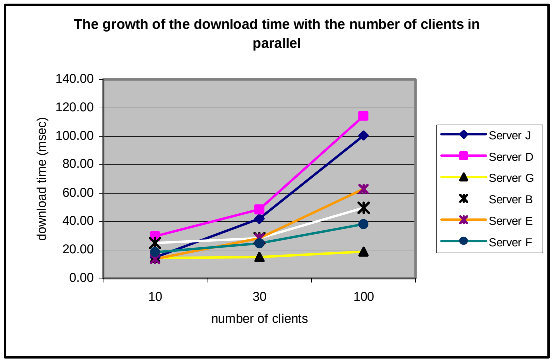

Figure 1. The growth of the download time

Ultimately, if the number of users' requests per second increases too much, the Web server will have to reject some of them, leading not only to a slowdown in response, but a denial of service to some clients.

B. Performance Affected by the Time of Day

Another factor, which influences Web server performance as the client sees it, is the time of day when the connections are established. Using our monitoring tool, the response times of different Web sites were monitored during a day from 9am until midnight (Figure 2). To observe the Web servers'

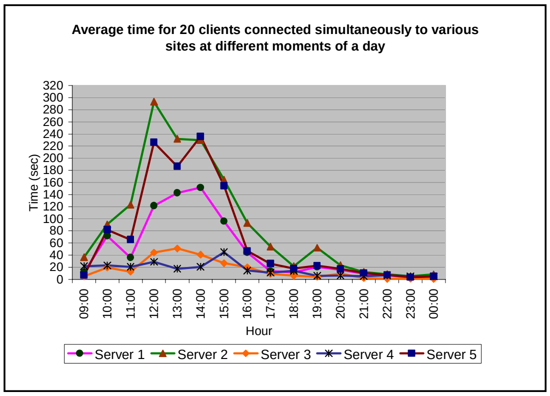

Figure 2. Twenty Parallel Connections

reaction, different numbers of parallel clients were used for the tests.

Large variations in the response times were noticed for download of the same WWW pages, although some servers were more affected than others. Also, during the peak hours the download times increase quite a lot compared with the off-peak periods of the day. For this period of time, the HTTP connections were established, but to download all of the data from the Web server took a disproportionately long time.

C. Sensitivity of Performance to Web Page Composition

A study of Web page composition and which components have an important influence on access time is presented in [7]. The composition with respect to file size and type of different commercial sites was analysed (Figure 3).

The analysis of Web page composition showed that images represent the biggest percentage of Web page size, and hence account for a considerable proportion of the download time for the page. Apart from images, the Web pages studied included other components such as JavaScript, ASP, and MicroMedia's Shockwave files.

Site Address (URL)	Total size (KB)	Html file size (%)	Imgs size (%)	Others size (%)	No of imgs
Server A	368.5	1.06	98.94	0	2
Server B	331.6	4.70	88.53	6.77	90
Server C	136.9	8.66	73.90	17.44	26
Server D	71.3	3.53	76.27	20.22	13
Server E	72.0	9.30	90.70	0	8
Server F	113.2	0.75	51.01	48.24	59
Server G	57.1	17.27	68.63	14.11	14
Server H	78.9	0.23	86.33	13.44	28
Server I	117.6	7.28	92.71	0	6
Server J	86.6	13.69	85.83	0.48	43
Server K	51.5	13.66	33.39	52.95	88

Figure 3. Statistics about the composition of the Web pages studied

Some of the pages studied have a very large number of images, which can affect both the network and the server, especially in peak hours when there are a lot of clients visiting the page. Also the total size of the Web page has an important contribution to the Web server's performance.

To retrieve all the components of the Web page a lot of requests will be sent to the server by the client. Along with all the requests sent by other clients, these can easily overwhelm the server. The immediate effect will be a slower response from the server to the requests, and an increase in download time.

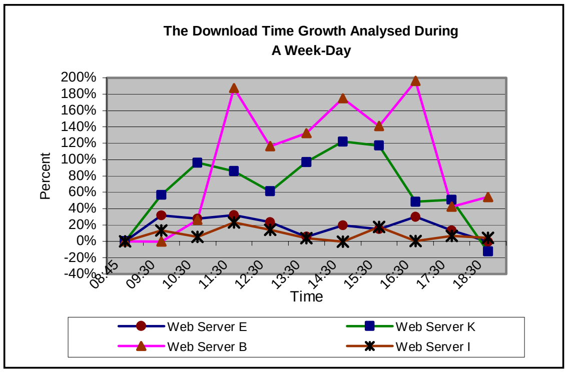

Figure 4. Download time growth for different Web pages during a day

Our experimental results suggest that the number of images has a great influence on download time. As can be seen in Figure 4, the pages with a large number of images had a much larger growth of download time than the pages with a small number of images. This indicates that a large number of images can seriously degrade Web server performance.

D. Other Factors Which Affect Web Server Performance

Several projects have studied different factors that influence Web server performance as perceived by the clients. Krishnamurthy [8] did an extensive study on the influence of the following factors on the end-to-end performance: the version of the HTTP protocol [9], caching, multi-server content, and byte range requests. The results show that the HTTP/1.1 protocol, particularly with pipelining, improves Web server performance. But if the Web page has a small number of embedded objects or the server closes the persistent connection without any notification, the improvement is reduced or lost. Using caching and multi-server content distribution is shown to improve performance.

Apart from the previously mentioned factors there are others which could be taken into account, such as network delay, redirection to another URL, DNS lookup time, the total number of objects making up the Web page, and CPU power. Varying any of these factors is likely to affect the performance delivered to the clients, though to what extent is not clear.

Different solutions have been proposed to improve end-to-end performance, such as a faster Internet connection, a faster server, a smaller number of images, and smaller Web pages size using different compression algorithms for their components. Apart from these static solutions, some dynamic ones have been suggested (e.g. DHTML, JavaScript, Java Web Pages, Shockwave/Flash). In the next Section we propose a new Web server design as another possible solution.

III. A SMART WEB SERVER

Large Web pages with many images can be very attractive to a client. However, this attraction fades rapidly if the pages take too long to download. We propose a new approach for building a smart dynamic Web server. It takes into account performance as the clients see it and serves different Web page content depending on the measured performance. This server may be implemented using servlet and applet technology [10]. The servlets and the applets allow the creation of Web pages on the fly. They have the possibility both to check the IP address of the client machine and to set cookies remotely. Using servlet capabilities, the server classifies the clients when they access any page according to some criteria, and stores this classification information.

One possible approach is that the classification information for each client is stored in a centralized database at the server side. Another idea is to make use of the cookies mechanism and to store and update the classification information remotely, at each client. Both solutions have advantages and disadvantages.

The main advantage of the cookies solution is that it saves space at the server, the database being distributed among the clients. Also it is useful when some clients are behind the same firewall. In this case all the computers behind the firewall are seen as having the same IP address. Thus, using cookies mechanism, a unique number can be set on each computer and it will differentiate between them when they access the site. Unfortunately from the security point of view the cookies, stored as text files in the clients computers, allow the user to modify and even remove the content.

The major advantage for the centralized database solution is that it is safely stored at the server, which can take some special security measures (e.g. firewall) to protect it. Another advantage is that the database can be hosted separately by a different machine, reducing the server loading. The resources consumed for database updating can be important and is a negative issue for this approach. A solution to differentiate between computers behind the same proxy has to be found.

The idea behind the smart Web Server consists of the classification of the clients at every access. According to the latest classification, the server will generate a different Web page (Figure 5).

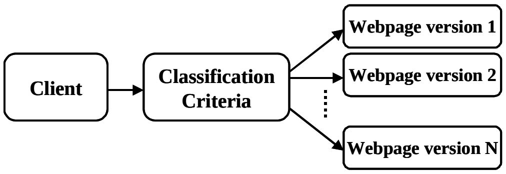

Figure 5. Smart Web server mechanism

The generated Web page may differ from another one generated for another client in content, the number of images, graphic design, structure, etc. This will affect its size and thus will influence some of the servers' performance indices.

JSP could be used to create different structures of the Web page, which will consist of static HTML content and dynamic content generated by servlets. Those dynamic parts will be generated according to the client's category. Another solution to generate the Web page is using a pure servlet, which will generate all the structure of the Web page. However this approach doesn't have the same flexibility as using JSP.

A possible classification criterion used to categorize the clients could be the download time of the Web page. Using an applet–servlet couple and a communication mechanism between them like Remote Method Invocation (RMI), Socket or HTTP download time can be computed.

Other classification criteria which could be taken into consideration are:

- the number of the accesses to the Web page made by the client up to that time;
- the load of the Web server for the period of time when the client access the server;
- the type of connection used by the client;
- if the client has a certain Plug-ins installed on its computer;
- what type and version of browser it uses to access the Web site;
- if the client has enabled multimedia option (for example: animation, sounds, images);
- the existence of the Java Virtual Machine browser.

To demonstrate the feasibility of the proposed server, we built an application, which sends Web pages with different structures according to which category a client belongs to. Three categories with different Web pages were defined (Figure 6); we used number of previous visits to define the client categories in this example.

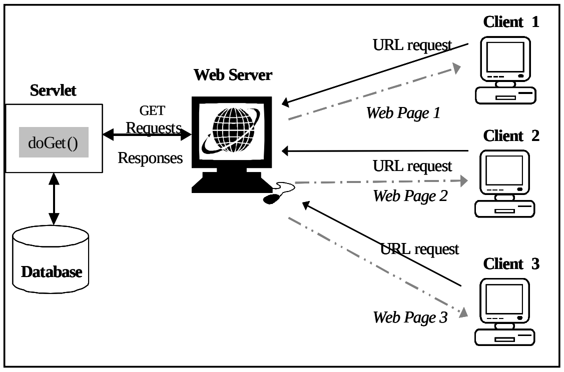

Figure 6. Web Server Application Structure

When a client wants to access the site, a HTTP GET request is sent to the Web server. Each time the server gets a request it dispatches it to the servlet. The servlet has two methods (doGet() and doPost()) to handle GET and POST requests which will be invoked by the server.

In our example we assumed that we already have a database with the clients' classification. At every access made by a client, the client’s category was checked in the database and the Web page corresponding to that category was sent. Our example system successfully classified clients and sent different content depending on the classification decision.

IV. PREVIOUS WORK

Different ways to create adaptive Web servers to optimize performance have been proposed recently. Some of them allow the clients to customize the site for themselves by describing their interests. Others attempt to guess the client's goal. The Web Watcher [11] project tries to predict what links a user will follow from the current page and highlight them. Another solution proposed for adaptive Web servers is to allow the server to learn what all the users that access the site want and to automatically adapt the site's content [12].

To avoid the rejection of some requests when the server is overloaded, a multicast solution was proposed [13]. The main idea is to use a multicast mechanism for distributing commonly requested pages, thus reducing the bandwidth consumption on the server output links. Abdelzaher proposed a new approach to reducing overload using "content adaptation" [14]. 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.

V. CONCLUSIONS AND FUTURE WORK

Extensive research has been done in order to test Web performance in different stress conditions. The number of clients which access the site at the same time and in parallel has been varied. The download time was measured at different times of the day and the values at peak and off-peak hours were compared. Also the components of the Web sites were analysed and their influence on the download time was studied.

We have proposed a smart Web server, which dynamically adjust 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. Some other criteria will be added and a weighted algorithm for classification will be developed. A comparison between the two methods of implementing the centralized and distributed database will be carried out.

REFERENCES

[1] M. Hall "Servlets and JavaServer Pages", Sun Microsystems Press, Prentice Hall PTR, 2000.

[2] Y. Nakamura, K. Chinen, H. Sunahara, S.Yamaguchi and Y. Oie, “ENMA: The WWW Server Performance Measurement System via Packet Monitoring”, in Proceeding of INET'99 Conference, San Jose, California, USA, June 1999.

[3] SPEC, “An explanation of the SPECWeb96 benchmark”, December 1996, http://www.spec.org/osg/Web96/.

[4] G. Trant and M. Sake, “WebStone: The First Generation in HTTP Server Benchmarking”, Technical report, MTS, Silicon Graphics Inc., February 1995, http://www.mindcraft.com/ Webstone/paper.html.

[5] C. Hava, S. Holban, L. Murphy, J. Murphy, “Initial Tool for Monitoring Performance of the Web Sites”, in Proceeding of CONTI’2000 Conference, Timisoara, Romania, October 2000.

[6] C. Hava, L. Murphy, “Performance Measurement of WWW Servers”, in Proceedings of 16th IEE UK Teletraffic Symposium, Harlow, UK, May 2000.

[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.

[8] B. Krishnamurthy, C. E. Wills, "Analyzing Factors that Influence End-to-End Web Performance", Computer Networks 33 (2000), pp. 17-32.

[9] H. Frystyk Nielsen, J, Gettys, A. Baird-Smith, E. Prud'hommeaux, H. Lie and C. Lilley, "Network Performance Effects of HTTP1.1, CSS1 and PNG", in Proceeding of ACM SIGCOMM'97, Conference, September, 1997, http://www.acm.org/sigcomm/ sigcomm97/papers/p102.html

[10] J. Hunter, W. Crawford, "Java Servlet Programming", O'Reilly, 1998.

[11] 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.

[12] M. Perkowitz, O. Etzioni, "Towards Adaptive Web Sites: Conceptual Framework and Case Study", Artificial Intelligence 118 (2000), pg. 245-275.

[13] P. P. White, J. Crowcroft, "WWW Multicast Delivery with Classes of Service", in Proceeding of HIPPARCH'98, University College, London, June 1998.

[14] T. F. Abdelzaher, N. Bhatti, "Web Server QoS Management by Adaptive Content Delivery", HP Labs 1999 Technical Reports, HPL-1999-161.