Modeling Operation of Web Service

Krzysztof Zatwarnicki and Anna Zatwarnicka

Department of Electrical, Control and Computer Engineering,
Opole University of Technology, Opole, Poland
{k.zatwarnicki,anna.zatwarnicka}@gmail.com

Abstract. This paper presents studies of modeling operation of Web service containing Web and database servers. The first part of the paper describes the method to obtain parameters of real servers. Additionally, detailed description of experiments and the results are presented. In the second part of the paper the Web and database servers models are introduced. In both of the presented models it is taken into account that the server processors are multi-core. Presented models can be used to evaluate proposed Web systems via simulation experiments.

Keywords: simulation model, web server modeling, database server modeling, queuing networks.

1 Introduction

Over the last few years the Internet has become the most innovative source of information and data. It has evolved from a medium for only privileged users into the medium we can't imagine living without. The rapid development of systems using WWW technology gives rise to the need for research on the effectiveness of the whole system delivering the required contents to the user.

The main semantic components forming the service of the Web system are: Hyper Text Markup Language (HTML), Hypertext Transfer Protocol (HTTP) and the Uniform Resource Identyfier (URI). The sources of HTML pages (Web pages) downloaded by clients (Web browsers) are Web servers. Web pages are transferred from Web servers to clients with use of the HTTP protocol. In the interaction between the client and the Web service the client sends the HTTP request to the Web server and the server sends in the response HTTP object [1].

There are a few ways of efficiency evaluation of a prospective service planned in use. One of them is the method of theoretical analysis using a proper mathematical modeling. This method is popular, however it requires simplifications of modeled system [2]. The second method enabling evaluation of proposed solutions is the simulation approach, which allows modeling of complex web services based on analysis of actual systems [3, 4].Queuing Networks usually are used in computer system simulations programs, but other approaches like Petri Nets [5] or propagation models [6] are also considered. The third method consists in building a prototype service containing real web servers [7]. The major disadvantage of this method is a high cost of carrying out experiments compared to the two previous methods.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2013, CCIS 370, pp. 384–393, 2013.

In this paper, methods of building a simulation model of web service are discussed. Firstly, technique enabling obtaining parameters of real Web and database servers is presented. Then the way of conducting experiments and the results of experiments are discussed. At the end, simulation models based on results of experiments are introduced.

The issue of creating the simulation model of the Web service is the topic of several publications. It is worth to start with WWW client's behavior models. The early works in this matter cover the issue of the HTTP object size modeling [8, 9]. Further, the complex client's behavior model [7] is presented as well as the client's behavior model in business service [10]. Other works concern the modeling of whole WWW services that cover the part related to client's modeling and also WWW servers, database servers, web dispatchers and wide area network [11–15]. However, there are not many works related to detailed WWW server simulation modeling [16, 17]. Authors of the paper already proposed such a model [18], however this time the presented model is designed for modern Web and database servers, the model contains much more detail and takes into account the fact that server processors are multi-core.

The paper is divided into five sections. Section 2 presents the method proposed to obtain Web and database servers parameters. Section 3 describes the conducted experiments and the results. Section 4 contains a description of simulation models. The final Section 5 presents concluding remarks.

2 Method to Obtain Parameters of Simulation Model of Web and Database Server

When creating a model of a Web system, it is important to take into account only elements of the system which have a significant impact on the request response time or can become potential bottlenecks [19]. Also, the way of operation of the system is important.

When the HTTP request is serviced, several actions have to be proceeded. At first the client opens the TCP connection with the web server. Next, it sends the HTTP request though the opened TCP connection. After that the web server prepares the response, than it looks for the requested file (in case of request for static object) or it precedes program/script preparing data for the client (in case of request for dynamic object). In the next step, the web server starts to send the response. After finishing service of the request the web server is ready to receive a subsequent HTTP request within the same TCP connection. In the end, after servicing numerous of HTTP requests send by the same client, the web server closes the TCP connection. The web server is able to service numerous HTTP requests concurrently, with the use of many different TCP connections.

The main elements of the web server which take part in the process of servicing HTTP request and have a significant impact on the response time are: processor, hard drive and the main memory, which acts as the cache memory of the file system. In the Fig. 1 the process of service of the request is presented.

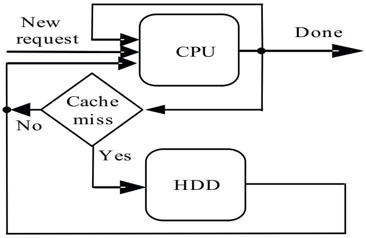

Fig. 1. Schema of process of service of HTTP request

The request service time on the processor is connected with the time to make the TCP connection $D^{\mathrm{CPU,TCP}}$, prepare the data to be sent and transfer the data through the computer network $D^{\mathrm{CPU,P+T}}$. The request service time on the hard drive involves a seek time (time to move the head of the hard drive), a rotational time (time to rotate the plate) and a transfer time, all together denoted as D^{HDD}. There are several methods to obtain service times for different elements of computer system. The first method is to modify the web server software to make the measurement of service time during the system operations [17, 13]. However, this method has some drawbacks. First of all, the access to software source codes of the Web server is required. Moreover, the changes of the software can change the way the examined system operates. These sorts of modifications can increase the load of the processor and prolong the request service time. The other method proposed by authors and presented previously in [18] is to make experiments during which the load and the throughput of the tested element of system are measured. The service demand time is then calculated on the base of service Demand Law [20]. This law specifies the dependence of service demand time D^{Resource} from the resource utilization U^{Resource} and the throughput X of the entire system according to Formula 1:

$$D^{\mathrm{Resource}} = \frac{U^{\mathrm{Resource}}}{X} \ . \tag{1}$$

In order to calculate service demand times several experiments and measurements had to be performed. During the experiments we intended to measure service times for processor and hard drive of web server and separately for processor and hard drive of database server supporting service of dynamic HTTP requests. We decided to obtain following values describing operations of web and database servers:

1. $D^{\mathrm{WWW,CPU,TCP}}$, the web server processor service demand time to open TCP connection;
2. $D_{\mathrm{S}}^{\mathrm{WWW,CPU,P+T}}(z)$, web server processor service demand time to prepare the static data (files of the file system) and transfer them through the computer network for different sizes z of web objects;
3. $D_{\mathrm{D}}^{\mathrm{WWW,CPU,P+T}}(z)$, service demand time of the web server processor to prepare the dynamic data and transferred them through the computer network for different sizes z of web objects;

4. $D_{\mathrm{S}}^{\mathrm{WWW,HDD}}(z)$, web server hard drive service demand time to prepare and transfer data of different sizes z to the web server processor;
5. $D_{\mathrm{Q1}}^{\mathrm{DB,CPU}}$, $D_{\mathrm{Q2}}^{\mathrm{DB,CPU}}$, $D_{\mathrm{Q3}}^{\mathrm{DB,CPU}}$, $D_{\mathrm{Q1}}^{\mathrm{DB,HDD}}$, $D_{\mathrm{Q2}}^{\mathrm{DB,HDD}}$, $D_{\mathrm{Q3}}^{\mathrm{DB,HDD}}$, data base server processor and hard drive service demand times to prepare the requested data for three different SQL queries;
6. $D_{\mathrm{JM}}^{\mathrm{WWW,CPU}}$, $D_{\mathrm{JR}}^{\mathrm{WWW,CPU}}$, $D_{\mathrm{JM}}^{\mathrm{WWW,HDD}}$, $D_{\mathrm{JR}}^{\mathrm{WWW,HDD}}$, $D_{\mathrm{JM}}^{\mathrm{DB,CPU}}$, $D_{\mathrm{JR}}^{\mathrm{DB,CPU}}$, $D_{\mathrm{JM}}^{\mathrm{DB,HDD}}$, $D_{\mathrm{JR}}^{\mathrm{DB,HDD}}$, web and data base server processor and hard drive service demand times to prepare data from a Joomla service main page (denoted as JM) and random page (denoted as JM).

3 Experiments Description and Results

A testbed containing three similar computers acting as the web server, the database server and a client were used in all described further experiments. The computers were connected to Gigabit Ethernet Repotec RP-G32224E switch (Fig. 2a).

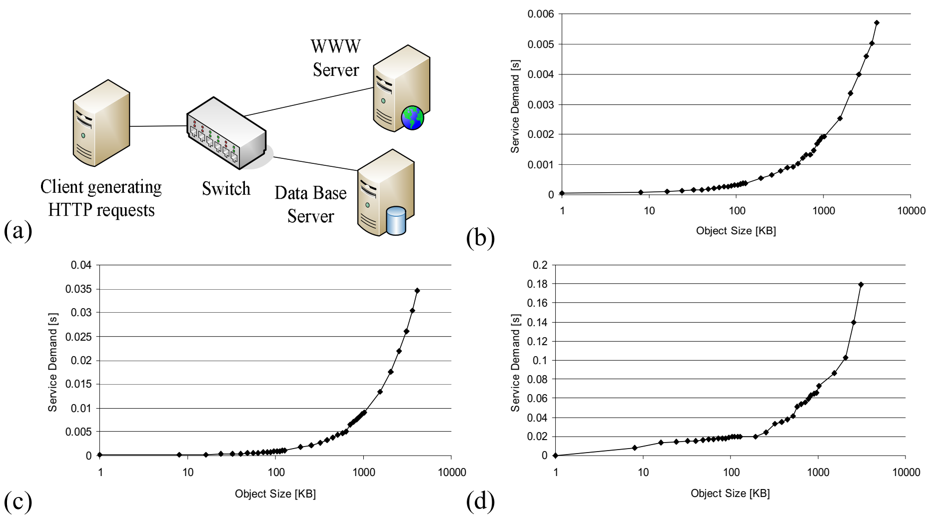

Fig. 2. Experiments and results: (a) testbed, (b) Web server processor service demand time to prepare and transfer static data in HTTP object size function, (c) Web server processor service demand time to prepare and transfer dynamic data in HTTP object size function, (d) Web server hard drive service demand time in HTTP object size function

Computers used in experiments had the following configuration: Intel Core2 Quad CPU Q6700 2.66 GHz processor, 2 x 2 GB PC2-6400 RAM memory, Barracuda ST3250410AS 250 GB 7200 RPM hard drive, Gigabyte GA-G33-DS3R FSB 1333 motherboard. Each of the computer was operating under Linux Debian 6.0.1a operating system. A computer acting as web server had installed

Apache 2.2.16 with PHP 5.3.3 while a computer acting as a database server had installed MySQL 5.6. On the third computer application acting as HTTP client was installed. The application was prepared by authors with use of curl libtaries [21], and enabled to strictly control the process of servicing HTTP request with use of several concurrent TCP connections.

Sar tool well known from unix systems enable us to observe the load of the web server resources. In order to obtain earlier mentioned servers parameters, six groups of experiments had been conducted.

Experiment 1. In the first experiment, the value of the web server processor service demand time to open TCP connection $D^{\mathrm{WWW,CPU,TCP}}$ was obtained. The experiment was divided into many separate stages. During the first stage, an HTTP object of size of 1 KB was downloaded with use of persistent TCP connections. The web client opened several concurrent TCP connections with the web server, and with the use of each the connection was receiving the same requested object many times. During the experiment a mean value of utilization of the processor $U_{\mathrm{E1.1}}^{\mathrm{WWW,CPU}}$ and a mean value of system throughput $X_{\mathrm{E1.1}}^{\mathrm{WWW}}$ was measured. Then the mean service demant time was calculated according to $D_{\mathrm{E1.1}}^{\mathrm{WWW,CPU}} = U_{\mathrm{E1.1}}^{\mathrm{WWW,CPU}} / X_{\mathrm{E1.1}}^{\mathrm{WWW}}$.

In the second stage of the experiment, the web client again was downloading the same 1 KB HTTP object however this time, for each download object a separate TCP connection was opened and immediately closed just after service was completed. Again the mean service demant time was calculated on the base of conducted measurements according to $D_{\mathrm{E1.2}}^{\mathrm{WWW,CPU}} = U_{\mathrm{E1.2}}^{\mathrm{WWW,CPU}} / X_{\mathrm{E1.2}}^{\mathrm{WWW}}$.

The mean service demand time to open TCP connection was calculated according to $D^{\mathrm{WWW,CPU,TCP}} = D_{\mathrm{E1.2}}^{\mathrm{WWW,CPU}} - D_{\mathrm{E1.1}}^{\mathrm{WWW,CPU}}$.

The values obtained during the experiment are as follow: $U_{\mathrm{E1.1}}^{\mathrm{WWW,CPU}} = 0.9946$, $X_{\mathrm{E1.1}}^{\mathrm{WWW}} = 320301\frac{1}{\mathrm{s}}$, $D_{\mathrm{E1.1}}^{\mathrm{WWW,CPU}} = 31.05\,\mu\mathrm{s}$, $U_{\mathrm{E1.2}}^{\mathrm{WWW,CPU}} = 0.9953$, $X_{\mathrm{E1.2}}^{\mathrm{WWW}} = 19707.62\frac{1}{\mathrm{s}}$, $D_{\mathrm{E1.2}}^{\mathrm{WWW,CPU}} = 50.5\,\mu\mathrm{s}$.

In the end the mean obtained service demand time to open TCP connection is $D^{\mathrm{WWW,CPU,TCP}} = 19.45\,\mu\mathrm{s}$ [22].

Experiment 2. During the second experiment the values of web server processor service demand times to prepare the static data and transfer them through the computer network for different sizes of web objects $D_{\mathrm{S}}^{\mathrm{WWW,CPU,P+T}}(z)$ were obtained. The second experiment was divided into several stages, each stage was dedicated to different size of downloaded web object. During each stage the same HTTP object with use of several persistent TCP connections was downloaded millions times. The sizes of downloaded objects were: 1, 8, 16, ..., 128, 256, 384, ..., 1024, 1536, 2048, ..., 4096 [KB].

The values of the web server processor service demand were calculated according to: $D_{\mathrm{S}}^{\mathrm{WWW,CPU,P+T}}(z) = U_{\mathrm{E2}}^{\mathrm{WWW,CPU}}(z) / X_{\mathrm{E2}}^{\mathrm{WWW}}(z)$, where z is the size in KB of downloaded HTTP object.

Results of conducted experiments are presented on Fig. 2b as a diagram of mean values of web server processor service demand time in the HTTP object size function.

A linear regression algorithm was used to prepare formula letting estimate the service demand time. The formula is as follow:

$$D_{\mathrm{S}}^{\mathrm{WWW,CPU,P+T}}(z) = \begin{cases} z \cdot 268.18 \cdot 10^{-6} + 6.0043 \cdot 10^{-5}, & z \in \langle 1, 128 \rangle \\ z \cdot 1.707 \cdot 10^{-6} + 2.1128 \cdot 10^{-4}, & z \in \langle 192, 1024 \rangle \\ z \cdot 1.119 \cdot 10^{-6} + 8.2385 \cdot 10^{-4}, & z \in \langle 1536, 4096 \rangle \end{cases} \quad (2)$$

where $D_{\mathrm{S}}^{\mathrm{WWW,CPU,P+T}}(z)$ is in seconds. Obtained model well fitt to results achieved in experiments and the coefficient of determination is $R^2 = 0.9989$ [22].

Experiment 3. The third experiment was similar to the second one. The service demand times of the web server processor to prepare the dynamic data and transfer them through the computer network $D_{\mathrm{D}}^{\mathrm{WWW,CPU,P+T}}(z)$ for different sizes z of web objects were obtained. Each stage of the experiment was conducted for objects of the same size. The objects were created dynamically with use of PHP script after HTTP requests arrival. The PHP script generated random text of size entered as a parameter of the script. The sizes of downloaded objects were the same as in Experiment 2. Again the service demand times were calculated on the base of following formula: $D_{\mathrm{D}}^{\mathrm{WWW,CPU,P+T}}(z) = U_{\mathrm{E3}}^{\mathrm{WWW,CPU}}(z) / X_{\mathrm{E3}}^{\mathrm{WWW}}(z)$, where $U_{\mathrm{E3}}^{\mathrm{WWW,CPU}}(z)$ was the utilization of the web server processor during the experiment, and $X_{\mathrm{E3}}^{\mathrm{WWW}}(z)$ was the throughput of the web server.

Figure 2c presents results as a diagram of mean values of web server processor service demand time for dynamic objects in the HTTP object size function. The relation between service demand time and the HTTP object size can be also modeled in following way:

$$D_{\mathrm{D}}^{\mathrm{WWW,CPU,P+T}}(z) = \begin{cases} z \cdot 7.8986 \cdot 10^{-6} + 9.4416 \cdot 10^{-5}, & z \in \langle 1, 128 \rangle \\ z \cdot 9.1403 \cdot 10^{-6} + 2.0679 \cdot 10^{-4}, & z \in \langle 192, 1024 \rangle \\ z \cdot 8.2572 \cdot 10^{-6} + 6.9411 \cdot 10^{-4}, & z \in \langle 1536, 4096 \rangle \end{cases} \quad (3)$$

where $D_{\mathrm{D}}^{\mathrm{WWW,CPU,P+T}}(z)$ is in seconds, and the coefficient of determination is $R^2 = 0.9998$ [22].

Experiment 4. In the experiment the web server hard drive service demand times to prepare and transfer data of different sizes $D_{\mathrm{S}}^{\mathrm{WWW,HDD}}(z)$, were obtained. During the experiment the hard drive of the web server and its utilization $U_{\mathrm{E4}}^{\mathrm{WWW,HDD}}(z)$ was observed. Similarly to two of the previous experiments, the tests had been divided in to many stages. During each stage different files of the same size were downloaded by the web client. The total size of files of the same stage was larger then 2 GB, thanks to this the requested objects were always fetched from the hard drive, and not from the cache memory. The sizes of the files were the same as in Experiment 2. The values of the web server hard drive service demand times were calculated similarly to the way presented in Experiment 2 and 1 according to: $D_{\mathrm{S}}^{\mathrm{WWW,HDD}}(z) = U_{\mathrm{E4}}^{\mathrm{WWW,HDD}}(z)/X_{\mathrm{E4}}^{\mathrm{WWW}}(z)$, where $X_{\mathrm{E4}}^{\mathrm{WWW}}(z)$ is the throughput of the web server.

Figure 2d presents diagram of mean values of web server hard drive service demand time for dynamic objects in the HTTP object size function. A chosen function to model the hard drive service demand time is following:

$$D_{\mathrm{S}}^{\mathrm{WWW,HDD}}(z) = \begin{cases} z \cdot 8.559 \cdot 10^{-4} - 3.0556 \cdot 10^{-4}, & z \in \langle 1, 8 \rangle \\ z \cdot 5.4206 \cdot 10^{-5} + 1.3261 \cdot 10^{-2}, & z \in \langle 16, 128 \rangle \\ z \cdot 6.1816 \cdot 10^{-5} + 1.09 \cdot 10^{-2}, & z \in \langle 192, 1024 \rangle \\ z \cdot 6.1482 \cdot 10^{-5} + 1.473 \cdot 10^{-2}, & z \in \langle 1536, 3076 \rangle \end{cases} \quad (4)$$

where $D_{\mathrm{S}}^{\mathrm{WWW,HDD}}(z)$ is in seconds, and the coefficient of determination is $R^2 = 0.9955$ [22].

Experiment 5. During the fifth experiment data base server processor and hard drive service demand times $D_{\mathrm{Q1}}^{\mathrm{DB,CPU}}$, $D_{\mathrm{Q2}}^{\mathrm{DB,CPU}}$, $D_{\mathrm{Q3}}^{\mathrm{DB,CPU}}$, $D_{\mathrm{Q1}}^{\mathrm{DB,HDD}}$, $D_{\mathrm{Q2}}^{\mathrm{DB,HDD}}$, $D_{\mathrm{Q3}}^{\mathrm{DB,HDD}}$, to prepare the requested data for three different SQL queries were obtained. In order to conduct required tests a database containing tree related tables had been created on the server with MySQL database. Each of the tables contained 5 millions rows consisting of mainly text data. On the web server there had been tree PHP scripts containing SQL queries of different complexity. The first script (denoted as Q1) enables to download 10 randomly chosen rows from one table, the second one (Q2)was fetching 20 randomly chosen rows from two related tables, and in the end third script (Q3) allow to download 100 randomly chosen related rows from three tables. During each experiment client opened many concurrent connection to download data and the mean value of the load of the processor $U_{\mathrm{Qn}}^{\mathrm{DB,CPU}}$, hard drive $U_{\mathrm{Qn}}^{\mathrm{DB,HDD}}$, and the throughput $X_{\mathrm{Qn}}^{\mathrm{DB}}$ had been measured. The service demand times were calculated on the base of Formula 1. The results are as follow: $D_{\mathrm{Q1}}^{\mathrm{DB,CPU}} = 4.49\,\mathrm{ms}$, $D_{\mathrm{Q1}}^{\mathrm{DB,HDD}} = 4.995\,\mathrm{ms}$, $D_{\mathrm{Q2}}^{\mathrm{DB,CPU}} = 55.53\,\mathrm{ms}$, $D_{\mathrm{Q2}}^{\mathrm{DB,HDD}} = 72.68\,\mathrm{ms}$, $D_{\mathrm{Q3}}^{\mathrm{DB,CPU}} = 187.91\,\mathrm{ms}$, $D_{\mathrm{Q3}}^{\mathrm{DB,HDD}} = 250.8\,\mathrm{ms}$ [22].

Experiment 6. In the end, during the last experiment web and data base server processor and hard drive service demand times to prepare data from a CMS Joomla service main page and random pages were obtained. In order to conduct this experiment, very popular CMS Joomla 2.56 system was installed on web and database servers. About 50 articles containing 4 KB of HTML text were added to the CMS system. During tests client opened many concurrent TCP connection to download data. Also the load of the processors of the web and database servers and their hard drives were observed.

The mean web server processor service demand time obtained in the experiment is $D_{\mathrm{JM}}^{\mathrm{WWW,CPU}} = 25.464\,\mathrm{ms}$, while the mean service demand time for the hard drive was negligibly small. The mean database server processor and hard drive demand time is $D_{\mathrm{JM}}^{\mathrm{DB,CPU}} = 5.043\,\mathrm{ms}$ and $D_{\mathrm{JM}}^{\mathrm{DB,HDD}} = 1.333\,\mathrm{ms}$. Results obtained for random aces pages are as follow: $D_{\mathrm{JR}}^{\mathrm{WWW,CPU}} = 28.215\,\mathrm{ms}$, $D_{\mathrm{JR}}^{\mathrm{DB,CPU}} = 5.539\,\mathrm{ms}$ and $D_{\mathrm{JR}}^{\mathrm{DB,HDD}} = 6.851\,\mathrm{ms}$ [22].

In almost all the of conducted experiments the load of the web and database processors was high, and the load ranged from 80 to 99 %. Also the number of concurrently serviced request ranged from 50 to 100 requests. This fact is important due to the way of constructing simulation model of processors.

4 Simulation Model Construction

The most common way to build simulation model of computer system is to use queuing network model [23]. Each resource is then modeled as a separate queue containing a waiting queue and a resource or multiple resources for the same queue.

The examined computer system presented in the paper can be modeled in two ways as is shown in the Fig. 3a and 3b. In the Fig. 3a the processor is presented as a load dependent resource where the service time depends on the number of requests being serviced. The differences in the service time are due to the fact that each of the processors contains four cores acting as separate services. Therefore the service time can be calculated as follow:

$$D^{\mathrm{CPU}}(n) = \begin{cases} D_{\mathrm{measured}}^{\mathrm{CPU}} \cdot (4-n) & \text{for } n < 4 \\ D_{\mathrm{measured}}^{\mathrm{CPU}} & \text{for } n \geqslant 4 \end{cases}, \tag{5}$$

where $D_{\mathrm{measured}}^{\mathrm{CPU}}$ is processor demand time obtained in experiments, and n is number of request being serviced, and waiting in a queue to the processor.

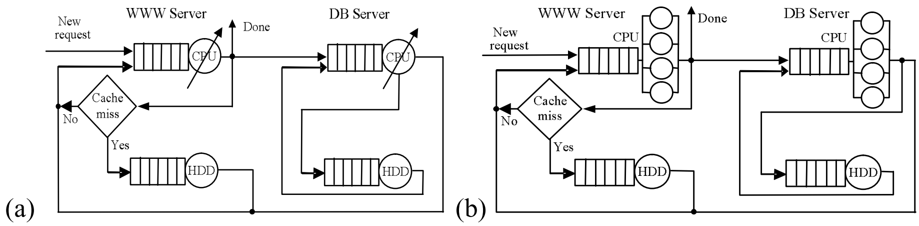

Fig. 3. Simulation models of web systems (a) processors modeled as load dependent resources, (b) processors modeled as load independent resources

In the model presented on Fig. 3b the processor is modeled as load independent resource therefore in this case the processor service time should be calculated in following way $D^{\mathrm{CPU}}(n) = D_{\mathrm{measured}}^{\mathrm{CPU}} \cdot 4$.

The demand times for the hard driver resources in the simulation model can have the same values as obtained during experiments due to the fact that hard drives have single services.

The module of cache memory in the web server simulation model determine if given HTTP object is located in cache memory or should be retrieved from the hard drive. The size of cache memory in simulation can depend on requirements. The algorithm responsible for swapping HTTP objects in the memory can be the Last Recently Used policy according to which the object not used for the longest time first leaves the memory in case of cache memory overflow [16]. The service time to retrieve data from the cache memory and transmit it to the processor is included in processor demand times.

Both of the presented simulation models should make it possible to obtain similar mean request response times, however if not only the mean values are important than the second (Fig. 3b), more precise model, should be used.

The simulation program implementing proposed models can be written with the use of appropriate simulation package enabling management of discrete-event simulations. Example of such environments are CSIM 20 [24] or OMNET++ [25].

5 Summary

In the article, the way of building the simulation model of Web and database servers was presented. A method to obtain parameters of simulation model of Web and database server was introduced also the way of conducting experiments was described. With the use of service demand low the required HTTP request service demand times for the processor and the hard drive ware determined.

On the base of obtained experiments results the Web and database servers models were proposed. In both of presented models it was taken into account that the server processors are multi-core. Presented models can be used in simulation based experiments evaluating propositions of designed Web systems.

References

1. Krishnamurthy, B., Rexford, J.: HTTP/1.1, networking protocols, caching, and traffic measurement. Addison-Wesley, Boston (2001)
2. Czachórski, T.: A diffusion approximation model of web servers. In: Cellary, W., Iyengar, A. (eds.) Proceedings of IFIP TC6/WG6. 4 Workshop on Internet Technologies, Applications and Social Impact (WITASI 2002), Wroclaw, Poland, October 10-11, pp. 83–92. Kluwer Academic Publishers, Boston (2002)
3. Barford, P., Misra, V.: Measurement, Modeling and Analysis of the Internet. In: IMA Workshop on Internet Modeling and Analysis, Minneapolis, MN (2004)
4. Zatwarnicki, K.: Providing Web service of established quality with the use of HTTP requests scheduling methods. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010, Part I. LNCS (LNAI), vol. 6070, pp. 142–151. Springer, Heidelberg (2010)
5. Rzońca, D., Stec, A., Trybus, B.: Data Acquisition Server for Mini Distributed Control System. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2011. CCIS, vol. 160, pp. 398–406. Springer, Heidelberg (2011)
6. Olejnik, R.: A Floor Description Language as a Tool in the Process of Wireless Network Design. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2009. CCIS, vol. 39, pp. 135–142. Springer, Heidelberg (2009)

7. Borzemski, L., Zatwarnicki, K.: A Fuzzy Adaptive Request Distribution Algorithm for Cluster-Based Web Systems. In: Proceedings of 11th PDP Conference, pp. 119–126. IEEE Press, Los Alamitos (2003)
8. Arlitt, M., Friedrich, R., Jin, T.: Workload characterization of a Web proxy in a cable modem environment. ACM Performance Eval. Review 27(2), 25–36 (1999)
9. Williams, A., Arlitt, M., Williamson, C., Barker, K.: Web workload characterization: ten years later. In: Tang, X., Xu, J., Chanson, S.T. (eds.) Publish Info. Web Content, pp. 3–22. Springer, New York (2005)
10. Cardellini, V., Colajanni, M., Yu, P.S.: Impact of workload models in evaluating the performance of distributed Web-server systems. In: Gelenbe, E. (ed.) System Performance Evaluation: Methodologies and Applications, pp. 397–417. CRC Press (2000)
11. Borzemski, L., Zatwarnicka, A., Zatwarnicki, K.: Global Adaptive Request Distribution with Broker. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part II. LNCS (LNAI), vol. 4693, pp. 271–278. Springer, Heidelberg (2007)
12. Borzemski, L., Suchacka, G.: Business-oriented admission control and request scheduling for e-commerce websites. Cybernetics and Systems 41(8), 592–609 (2010)
13. Riska, A., Riedel, E.: Disk Drive Level workload Characterization. In: Proceedings of the USENIX Annual Technical Conference, Boston, pp. 97–103 (2006)
14. Zatwarnicki, K.: Neuro-Fuzzy Models in Global HTTP Request Distribution. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010, Part I. LNCS, vol. 6421, pp. 1–10. Springer, Heidelberg (2010)
15. Zhang, Q., Riska, A., Riedel, E., Mi, M., Smarni, E.: Evaluating performability of systems with background jobs. In: Proceedings of The Symposium on Dependable Systems and Networks (DSN), Philadelphia, pp. 495–505 (2006)
16. Aron, M., Druschel, P., Zwaenepoel, W.: Efficient Support for P-http in Cluster-based Web Servers. In: Proceedings of the USENIX 1999 Annual Technical Conference, Monterrey, CA (1999)
17. Cherkasova, L., Gardner, R.: Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor. In: USENIX Association, Berkeley, CA, Anaheim, USA, pp. 24–27 (2005)
18. Zatwarnicki, K.: Identification of Web server. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2011. CCIS, vol. 160, pp. 45–54. Springer, Heidelberg (2011)
19. Fortier, P.J., Howard, M.E.: Computer Systems Performance Evaluation and Prediction. Digital Press, USA (2003)
20. Denning, P., Buzen, J.: The Operational Analysis of Queueing Network Models. ACM Computer Survey 10(3), 225–261 (1978)
21. CURL library documentation (2012), http://curl.haxx.se/
22. Krawczyk, J.: Determination of parameters of the web and database servers. M.S. thesis, Department of Electroengineering, Automatic Control and Computer Science, Opole University of Technology, Opole, Poland (2012)
23. Menascé, D., Bennani, M.: Analytic performance models for single class and multiple class multithreaded software servers. In: Int. CMG Conference 2006, pp. 475–482 (2006)
24. CSIM20. CSIM20, Mesquite Software, 2012 Development toolkit for simulation and modeling (2012), http://www.mesquite.com
25. OMNET++, Development toolkit for simulation and modeling (2012), http://www.omnetpp.org/