Method for Quality of Network Services Analysis Using Queuing Modelling of Information Systems and Computer Simulation Techniques

Łukasz Bagrij
Wroclaw University of Technology, Poland
lukasz.bagrij@pwr.wroc.pl

Katarzyna Nowak
Wroclaw University of Technology, Poland
katarzyna.nowak@pwr.wroc.pl

ABSTRACT

From the moment when computer networks were no longer domain restricted only to military purposes or an academic “experiment” and became widespread mean for work and entertainment, we observe still rapidly growing number of its end-users. Furthermore computer systems started to be more and more complicated and thus capable of delivering sophisticated services. Hence nowadays we cope with complex networked Information Systems (IS) providing multifarious multipart electronic business services. From the very beginning of this improvement, systems administrators were extremely interested in sustaining and even increasing dependability level of infrastructures they managed. This subject becomes to be extremely important considering rising amount of threats affecting networks, both these malicious as well as these “normal” caused by development. For this reason lots of tools have been build to enhance management process and in consequence dependability level. However in spite of well studied Dependability Theory researchers still didn’t find appropriate method to analyse modern IS and evaluate its functional-dependability parameters effectively. This paper introduces method based on Queuing Theory for modelling and computer simulation for estimation of work efficiency of IS, which allows to design and configure these systems more accurately.

Key Words: Information System, dependability, efficiency estimation, Queuing Theory, queuing functional-dependability model, Monte Carlo simulation.

1. Introduction

Due to notable and still growing complexity of technical solutions being created, it occurred to be urgent to truly analyse work of modern systems, which often act as a one whole, but actually are compounded of many autonomous elements that in turn are under strong influence of a random factors. Above rational implies the strong need for considering studied system twofold, i.e. as a set of independent components connected with one another somehow (each of which possesses some functionality and dependability), and as one coherent system. Each type of a system may be always characterized using a predefined set of attributes describing its abilities. Especially regarding to work effectiveness evaluation, some parameters making possible to depict functional and dependability properties of intended system, might be very useful.

Although both classes of these properties are not exactly equivalent, they are tightly related. Classically Dependability Theory defines six basic measures [1], which in some sense embody also functional aspects of a system. These are [11]: **availability** – readiness for delivering a correct service, **reliability** meaning continuation of delivering a correct service, **safety** i.e. possibility of appearance of a catastrophic consequences for user and environment, so-called **confidentiality**, that is risk of compromising unauthorized information, **integrity** – chance that system will step into erroneous state and **maintainability** which is capability of a system to recover (submit for repairs and modifications).

One of the most significant treatments utilized for efficiency analysis of some specific group of complex systems is Queuing Theory. This discipline aims to build mathematic models, which are being used to rational manage whichever servicing systems, also named Mass Service Systems [10], in order to optimize parameters of their work, in particular: queue length, number of service channels, service intensity, etc. Lack of appropriate configuration of Mass Service System may result in incorrect work of a whole system and thus be a direct cause of profits decrease, or even serious loss caused by appearance, in some time frames during system work, of resources (e.g. personnel and machines) overload and below-load effects.

Marked here the problem of configuration optimization of queuing systems has been adopted in this paper to the reality of modern network services – in this situation – the WWW net. The aims of the undertaken work were to create a realistic model of a concrete real Mass Service System, and further to build an appropriate software simulator which would allow to do a research on activity of analysed system working under defined circumstances regarding its performance.

2. Computer systems as Mass Service Systems

Currently receptive field of applications for this method, as well as other that addresses the same target, are new electronic systems (mainly computer systems), which year on year become more complicated to satisfy clients and their escalating needs.

Exclusive example of computer systems extending very dynamically are Communication and Information Systems, based mostly on telecommunication networks. Complexity of modern networks, i.e. their physical structure, methods of functionality, and chiefly – software, brings the serious need for applying special models within design process, which would let to forecast consequences of decisions made by system operators and administrators at the early stage. For example for better network usage it is possible to manipulate a different system parameters, especially its resources, e.g. amount of memory allocated to application/service, links throughput, processor usage, etc. Therefore computer networks, and computer systems in general, are objects which performance and quality of service is determined by many parameters. This explains why there is necessity for tools supporting numerical work estimation of these systems.

Proper choice of a system configuration, i.e. values of particular configuration parameters, has not trifling impact on its efficiency and quality of work. It may also considerably influence on system’s inoperabilities, particularly so-called information deformations, i.e. temporal failures which source in most cases is software. As shown in [12] in practice their stimulus is crucial for dependability of a whole system.

3. Queuing model of client-server system

Method for computer system work effectiveness estimation will be shown basing on the example of analysis of really simple client-server system, where server plays WWW service provider’s role founded on the standard HTTP/1.1 protocol [7]. However simplicity of the test-bed system does not compromise at all the proposed approach, since it might be easily generalized and adopted to other systems which could be more difficult to resolve.

In order to execute any experiments it is indispensable to introduce models describing examined objects behaviour, i.e. functionalities implemented by them. From computer simulation's point of view functional model of a system is defined similar to automata (finite state machine) as it depicts his reactions for all possible events on the basis of its internal states. State itself is transformed to a new value as a result of immediate answer to some precise event. Hence model changes its internal states to a new values and generates further events. Moreover system state is constant between sequential events.

Analysed model of Mass Service System consists of two autonomous functional models, i.e. behavioural model of a client and WWW server functional model. The following figure (Fig.1) presents the general model of studied client-server system.

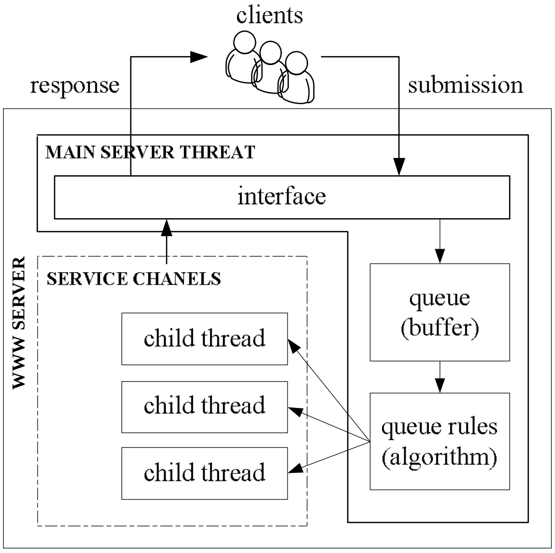

Fig. 1 Queuing model of WWW Server

4. Client behavioural model

The task of these objects' (a group of objects more precisely) model of activity is first of all to provide input events stream (requests) for server functional model. This client model contains the set of parameters describing particular features and detailed aspects, which impress the client's role during interaction with server model.

It is assumed that each client generates requests for accessing server's resources in a random time moments. Number of generated tasks by clients is shaped by exponential distribution. The PDF (Probability Density Function) of this distribution is defined as follows:

$$f\left(t;l\right)=\begin{cases} l\,e^{-l\,t} & for\ \ t\geq 0 \\ 0 & for\ \ t<0 \end{cases} \qquad (1)$$

where:
t – time
λ – mean time between two next client requests

Client on the basis of suitable parameters executes the following actions:

- With specified probability uses a graphic or textual browser for communication with WWW server;
- Requests access to random resource (chosen from predefined set of available ones);
- Requests execution of random method addressing particular resource (one of the set instantiated by HTTP/1.1 standard protocol);
- Waits some fixed time for service within server. After violating this restriction event is treated as exceeded and removed from queue. In reality software by client's side (browser) should inform the user about the error often displaying, not necessarily true, the standard information site – error 404 (not fund).

Besides that for sake of simplicity of the model we make assumption due to loss-free transmission. In practice this means that client always replies with TCP ACK header packet to a received TCP packet sent by the server. This confirms reception of data. In addition in order to introduce further elements of the reality to the model, we presume that time when client sends confirmation is random and its value depends on uniform distribution with the interval (0,1). The PDF of this distribution is shown below:

$$f(t)=\begin{cases}\dfrac{1}{b-a} & \mathit{for}\ \ a<t<b \\ 0 & \mathit{for}\ \ t<a \vee t>b\end{cases} \qquad (2)$$

where:
t – time
a – left interval limit
b – right interval limit

5. Functional model of the WWW server

As in considered case the WWW server is actually our studied Mass Service System, naturally its model is much more complicated than above talked about client's. In such coincidence several techniques helpful in modelling which simultaneously makes these descriptions more readable may be applied [12]. Very popular are UML diagrams and promising but still being in development phase – the orchestration and choreography of network business service description languages. Its worth to mention here about the most important, like WSDL[1], WS-CDL[2], WS-BPEL[3], WSFL[4], BPML[5], and others.

5.1 Algorithm

In this work the functional model used in functional-dependability analysis will be presented in form of addressed server generic work algorithm, which supplied by model parameters creates unified and clear functional picture of WWW server. Designed algorithm will be demonstrated for the purpose of this paper as wording instead of chart. According to it – WWW server (more specifically its main thread) waits in never-ending loop for client's requests occurrence, that is try to establish connection with server on port tcp/80 through HTTP protocol. When such event happens the server may find out that unfortunately main thread is busy by other connection. In this situation the incoming one is rejected (server simply do not reply for request). However if in the moment of the event arrival main thread was not occupied then server will response (for TCP transport protocol this is the three-way handshake sequence) and establish the connection with client.

Nevertheless accepted connection can be qualified as queued (waiting for service) only if one critical condition is fulfilled: server must have enough number of resources to process a request. To check whether this constraint is fulfilled or not a simple test is performed, i.e. server tries the actual number of busy child threads if it reaches the maximal threshold. If it is so, client receives the answer with the code 503 – service unavailable. Additionally server places in the header of the reply packet an information pointing the period after which client may retry. When this test is accomplished with success, the main thread decides on assignment of a free existing child thread or enforces creation of a new one. After this is finished finally the role of the main thread ends and it returns to listening on port tcp/80.

The marker that a request is being processed by one particular service station (child thread) is acceptance of a packet containing a method's name (the one that other side requests to execute) by threat, e.g. HTTP GET. At this point server is obligated to perform another test. It has to check correctness of received client's request. To make the model not too complicated this verification is restricted only to two major issues:

- Rightness of URI pointer (Unified Resource Identifier) – if server states that this id is incorrect or none of server's resources is referred to by this pointer, than server sends status code 404 (not found) and relevant connection is closed automatically by server.

[1] Web Services Description Langage; http://www.w3.org/TR/wsdl20/
[2] Web Services Choreography Description Language; http://www.w3.org/TR/ws-cdl-10/
[3] Web Services Business Process Execution Language; http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.html
[4] Web Services Flow Language; www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf
[5] Business Process Modelling Language; http://xml.coverpages.org/BPML-2002.pdf

- Permission to execute requested method regarding requested resource – when resource was properly pointed but client requires to take a forbidden action on particular resource, than status code number 405 – method not allowed is sent and similar to before relevant connection is closed automatically by the server.

After request has passed previous stages client receives awaited reply. Described model provides simplified processing – it provides only two of the HTTP methods, i.e. HEAD and GET for website and GET for graphics. The standard server response for them is data packet signed with code 200 (OK). What is more, often server meets the situation when size of a resource goes beyond volume of a single packet. Then the server sends as many times as it takes a rest of requested data, up to the moment when all data is sent. This type of packets is tagged with code 100 (continue).

At the end what is more interesting the server may find that a resource requested by client is functionally associated with other resources. In this particular case one such situation is considered, case when a graphic browser needs many other resources for successful and complete download and display of a webpage. Most frequently these are pictures and other graphic elements. Thus in case of an event telling the server to "get the main page", just after finishing this transfer, accordingly to the idea of persistent connections introduced with version 1.1 of HTTP protocol, server will automatically send all the related resources. Not before this happens (complex client service) handled request can be removed from the queue, connection closed, and child thread may fall into idle state.

5.2 Model parameters

The description of the system structure, which in this case roughly speaking defines two network nodes, one of which is simple element playing role of an events generator, and the other is build from three components: management module (main thread), workers (child threads) and buffer (queue), along with its functionality definition (algorithm) is insufficient and must be filled out with information on model characteristics, called generally – configuration.

System configuration [3] is defined by some strictly outlined key parameters. In this matter the distinguished as follows attributes are differentiated into several domains of:

Server global queue

- Max queue length – modelled system is Mass Service System with limited queue length. This model implements one global event queue.

Server's threads – this attribute allows defining closed interval, which specifies the permitted number of child treads created during server's (main thread) work. In considered case threads match service stations of Mass Service System.

- Max number of child threads – upper threshold of resources available to be used by server while processing upcoming requests.
- Min number of child threads – lower threshold of resource that should always be ready to use immediately (allocated threads).
- Main thread work time – time needed for processing received single request and for making up mind whether to establish connection or drop it. Might be estimated arbitrary, random or calculated on a basis of empirical observations.
- Child thread basic work time – time that each child thread needs for completing its basic operations, e.g. request correctness, sending single data packet, etc. About its value may decide analogous factors as for previous one parameter.

WWW server resources – these are one of the most significant issues affecting this type of system. Indeed – they are the reason why clients send their requests to the system.

- Number of server resources – self explaining.
- Resources size – matrix defining size of each available resource. Values within it should be randomized for each new experiment on the foundation of discrete uniform distribution with predefined interval.
- Relations between resources – alike above – table; sometimes for successful retrieving a resource, download of other connected resources is demanded, e.g. getting webpage with graphics.
- Max size of data within single packet – fixed parameter referring to the fact that in TCP/IP networks acceptable size of one packet is restricted.

HTTP methods

- Allowed methods – describes which HTTP methods server carries out (not every method is allowed to be executed on whatever resource).

6. Functional-dependability model of client-server system

Depending on the technique how anticipated object is willing to be studied and what sort of results are expected to be obtained, topic of research should be described in diverse way. Moreover each system has its own specificity, hence in each instance individually for concerned system there is need to define suitable set of parameters. Mentioned in the chapter 1 – Introduction generic set of dependability attributes may serve here only as the groundwork for qualification of these characteristics of the system, which analysed would give satisfying answers to stated questions.

Functional-dependability model is understood in this paper as definition of exact set of observed parameters of the system, regarding their expected values (or ranges of this values) and states the foundation for numerically or concerning quality work estimation.

For the purpose of this experimentation the four planes (criterions) have been recognized. These are: **clients' requests** linked with events queue, **system's task**s reflecting system load, **server's threads** expressing system resources, and **time** – determining overall system efficiency. According to presented classification taken tests were made to assess the following statistical traits:

- **Mean number of all events in the system** – i.e. clients' requests that came to the Mass Service System during whole simulation (one experiment);
- **Mean number of rejected connections** – i.e. refused clients' requests
  (HTTP 503 – service unavailable) during whole simulation;
- **Mean number of exceeded connections** – i.e. clients' requests that went above time limit (server did not response in expected time) during whole simulation;
- **Mean number of tasks** (clients' requests) **in the system** – defined as average current sum of tasks waiting in the queue and being processed in service stations;
- **Mean queue length** – average current number of clients' requests (tasks) residing in the buffer waiting for service;
- **Mean time of created service stations** – defined as average current number of allocated child threads of the WWW server;
- **Mean number of busy service stations** – defined as average current number of child threads of the WWW server engaged in requests processing;
- **Mean time** spent **in queue** – that is average time wasted for waiting for processing by child threads during whole simulation;
- **Mean time of request service** – described as average time needed for complex processing of client's request, i.e. by both threads (main and child) during whole simulation.

Please note that single experiment equals to one simulation, or more specifically to one set of simulations, that means the same experiment taken, for exactly the same model and simulation parameters, repeated many times.

7. Investigation results

The tool used for work efficiency estimation of example Information System was specially designed computer program [8], which function was to simulate the modelled system. Developed and implemented exclusively for this purpose simulator possesses two major characteristics, and therefore is: **nondeterministic** (a probabilistic data are expected on the input and output of the simulator) and **event-driven** (simulation triggered by events, not time). Its worth to notice that investigation was focused mainly on the processes of the application layer (within TCP/IP network model), and thus server load nor network traffic in lower layers (e.g. IP packet level) was not considered.

Depicted chosen simulation method is commonly called the Monte Carlo method. Symptomatic for this technique is requirement to repeat same experiment many times, because in order to acquire reliable data and to be able to make genuine conclusions coming from a research, results of lots of single simulation iterations must be averaged. The basic principle and architecture of built simulator is illustrated on the figure 1 placed below.

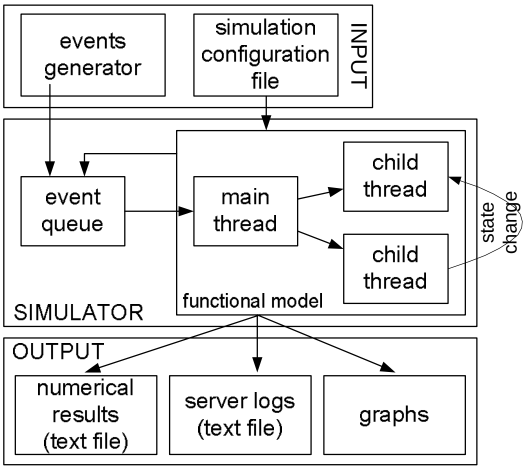

Fig. 2 Architecture of client-server simulator

Following this rule, all interesting from work efficiency analysis point of view four subsets of means was computed and collected. As a result underneath are presented example outcomes of experiments achieved during all investigation process. Shown results are the following chart representing function of one of examined compound parameters with textual interpretation. This part selected from all simulation products shows very clearly the method how in practice complex estimation of system work might be realized. For practical purposes reader should find most helpful the following analysis (figure 3) – average percent of clients' request which came and has been processed in function of diverse number of child threads.

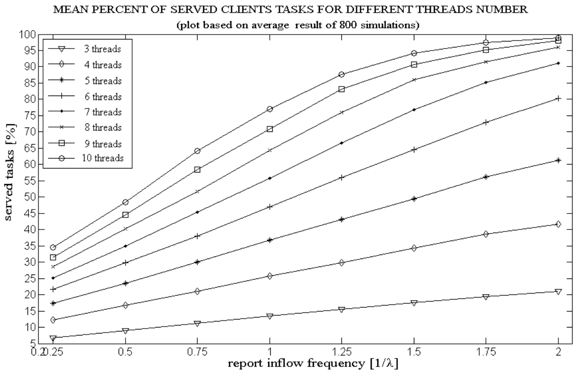

Fig. 3 Average percentage of served demands in function of report inflow frequency

Disregarding the fact whether request was processed successfully in standard way (because was defined correctly) or not because of some mistakes inside, the most precious is possibility to evaluate efficiency of WWW server under particular configuration working under particular environment conditions, i.e. predefined external load. This gives power to estimate the quality of service delivered by this server (Information System).

As appeared, unfortunately in case of large traffic and many very frequent connection attempts (i.e. $l \le 0{,}5$) even the „richest” configuration is not enough, since efficiency decreases below 50%, which for sure is unacceptable in reality. On the other hand figure 3 shows very evidently that while $l = 2$, system might run only 9 service stations in practice and there is no need for feeding with valuable resources more than that.

8. Conclusions

After close investigation of different set of dependencies (in particular number of processed requests – figure 3, and number of server resources being used) a strange situation has been observed. Afterward it became obvious why this was originally seen as paradox. Namely, on the one hand for situation with $l = 1{,}25$, number of utilized resources should indicate 7 as the perfect amount of child threads for good functioning of the system under these circumstances, as more resources would not significantly enhance the availability and thus would be squandered. On the other hand, what can be watched on the previous diagram, for this configuration of the model work efficiency is not really impressing – not even 70%.

Here comes out that problem with WWW server efficiency do not lie only in the amount of system resources available for using during tasks completion. This must be realized that equally important to the number of child threads is main thread itself. This means that server's main thread is the Information System's bottleneck. When giving a bit more care to this issue, shortly after it, it easily may be founded that this is the place where the whole service is managed. After all, main thread accepts or refuses clients' requests and eventually passes the process to the appropriate module. If frequency of requests inflow exceeds "some" threshold, than logically main thread do not follow up incoming jobs. Principally it is possible to observe situation when there exist many free to use resources within the system, however due to its bottleneck, which is on the input of the system, upcoming tasks could not even reach service stations. When this happens in most cases this leads to the time exceed error. This is the real reason why percentage of processed requests is relatively small comparing to plenty of offered resources.

The cure for discovered problem is for example creation of redundant WWW server. Approximate calculation says that architecture of Information System with two servers could allow even to double the number of accepted connections. Such solutions still with "poorest" configuration of each server for sure would be more efficient than the standard one. The issue of implementation is not leading, and choice would be determined by many other aspects, like network safety.

9. Summary

At the end summarising and very important conclusion, which was coming on the mind all the time, no matter what type of statistical parameters and their values as well as changes of these values has been studied. In order to sustain dependability level of Information System, especially quality of service delivered by it, while dealing with growing frequency of requests occurrence, the system has to allocate considerable many resources. In other words, amount of resources should rise faster than job frequency to make it possible to keep the level of system efficiency. Unluckily administrator can not increase resources infinitely because in short time the second bottleneck (first is the available resources) will show up – the main thread of WWW server.

10. Acknowledgement

Work reported in this paper was sponsored by a EU grant DESEREC IST-2004-026600, (years 2006-2008) "Dependability and Security by Enhanced Reconfigurability".

References

[1] Avizienis A., Laprie J. C., Randell B., "*Fundamental Concepts of Dependability*", 2000.

[2] Baccelli F., Makowski A.M., "*Queueing models for systems with synchronization constraints*",

Proceedings of the IEEE Vol. 77(1), Jan 1989.

[3] Bagrij Ł., "*Simulator of Complex Queuing System*" (MSc Thesis), Wrocław 2006.

[4] Bollen M.H.J, "*Method for reliability analysis of industrial distribution systems*", Generation, Transmission and Distribution, IEEE Proceedings C Vol. 140 (6), pp. 497-502 November 1993.

[5] Caban D., „*Software influence on reliability*", Inżynieria komputerowa, praca zbiorowa pod red. Wojciecha Zamojskiego; Warszawa 2005 [in polish].

[6] Czachórski T., „*Modele kolejkowe w ocenie efektywności sieci i systemów kolejkowych*", WPK J. Skalmierskiego, Gliwice 1999 [in polish].

[7] Fielding R., Gettys J., Mogul J. C., Frystyk H., Masinter L., Leach P., Berners-Lee T., "*Hypertext Transfer Protocol – HTTP/1.1*", RFC 2616,1999.

[8] Kalinowski B., „*Computer tools for dependability parameters estimation and selection of maintenance strategy for wide hierarchical technical system*" (PhD Thesis), Wrocław 2005.

[9] Lui S., Xue L., Ying Lu, Abdelzaher T., "*Queueing model based network server performance control*", Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE 2002.

[10] Obretenow A., Dimitrow B., „*Teoria masowej obsługi – poradnik*", PWN, Warszawa 1989 [in polish].

[11] Zamojski W., "*Introduction to the dependability modeling of computer systems*". Proceedings of International Conference on Dependability of Computer Systems. DepCoS - RELCOMEX 2006. Eds W. Zamojski [i in.]. Szklarska Poręba, 25-27 May 2006. Los Alamitos [i in.]: IEEE Computer Society [Press], cop. 2006.

[12] Zamojski W. "*Raliability – functional model of computer-human system*", Inżynieria komputerowa. Praca zbiorowa pod red. W. Zamojskiego, Warszawa 2005 [in polish].

[13] Zamojski W., " *Remarks on reliability of future computer systems*", The Proceedings of the International Conference on Information Technology & Natural Science. ICITNS 2003. Ed. by Abdelfatah A. Yahya, Amman 2003.