Performability Modeling a Client-Server Communication System with Randomly Changing Parameters using MOSEL

B. Almási, * G. Bolch, J. Sztrik
University of Debrecen, Debrecen, Hungary
* University of Erlangen, Erlangen, Germany
{almasi|jsztrik}@math.klte.hu, bolch@informatik.uni-erlangen.de

1 Introduction.

Performance modeling of recent computer and communication system development has become more complicated as the size and complexity of the system has increased, (see [7] and [4]). Finite source queueing models are efficiently used for performance evaluation of computer systems (see [1], [3], [4] and [9]). Realistic consideration of certain stochastic systems, however, often requires the introduction of a random environment, sometimes referred as to Markov-modulation, where system parameters are subjected to randomly occuring fluctuations or bursts. This situation may be attributed to certain changes in the physical environment such as personal changes and work load alterations. Gaver *et al.* [6] proposed an efficient computational approach for the analysis of a generalized structure involving finite state space birth-and-death processes in a Markovian environment.
This paper deals with a First-Come, First-Served (FCFS) queueing model to analyze the behaviour of heterogeneous finite-source system with a single server. The clients (request sources) and the server are supposed to operate in independent random environments, respectively, allowing the arrival and service processes to be Markov-modulated ones. Each request of the clients is characterized by its own exponentially distributed source and service time with parameter depending on the state of the corresponding environment, that is, the request generation and service rates are subject to random fluctuations. Our aim is to get the usual stationary performance measures of the system, such as utilizations, mean queue lengths, average response times. The main problem is that the state space of the underlying continuous-time Markov-chain will be very large, so we have the state space explosion problem.

2 The Queueing Model.

Consider a finite-source queueing system with N heterogeneous clients (sources) and a single server. The sources and the server operation is influenced by random environments.

[1]Research is partially supported by German-Hungarian Bilateral Intergovernmental Scientific Cooperation, OMFB-DLR No. 21-2000, and Hungarian Scientific Research Found OTKA T0-34280/2000 and FKFP grant 0191/2001.

The server and the clients are collected into M groups ($1 \leq M \leq N+1$). The members of a group may operate in a common random environment. The environmental changes are reflected in the values of the access and service rates that prevail at any point of time. The main objective is to adapt these parameters to respond to random changes effectively and thus maintain derived level of system performance.

The members of group p are assumed to operate in a random environment governed by an ergodic Markov chain $(\xi_p(t), t \geq 0)$ with state space $(1, \ldots, r_p)$ and with transition density matrix

$$\left(a^{(p)}_{i_p j_p}, i_p, j_p = 1, \ldots, r_p, a^{(p)}_{i_p i_p} = \sum_{k \neq i_p} a^{(p)}_{i_p k} \right).$$

Whenever the environmental process $\xi_p(t)$ is in state i_p the probability that client c (a member of group p) generates a request in the time interval $(t, t+h)$ is $\lambda_c(i_p)h + o(h)$, $p = 1, \ldots, M$. Each request is transmitted to a server where the service immediately starts if it is idle, otherwise a queueing line is formed. The service discipline is First-Come, First-Served (FCFS). Assuming, that the server belongs to group 1 and the environmental process $\xi_1(t)$ is in state i_1 the probability that the service of the request originating from client c is completed in time interval $(t, t+h)$ is $\mu_c(i_1)h + o(h)$.

If a given source has sent a request it stays idle and it can not generate another one. After being serviced each request immediately returns to its source and the whole procedure starts again. All random variables involved here and the random environments are supposed to be independent of each other.

Similar models were stuied earlier by different authors (see [6], [10], [11]), but usually simulation tools were used to calculate numerical results. In the followings we shortly introduce a software tool for analytical investigations.

3 The MARKMOD Software Tool.

Figure 1.

The software tool introduced here can produce analytical results for the the described model. The system consists of three parts: MARKMOD, MOSEL, SPNP (see Figure 1.) and can be used in any unix-like environment.

The program MARKMOD implements the mentioned Markov-modulated queueing model. MARKMOD's input is a very simple and well commented parameter file (text file), from which the program generates a MOSEL source file. This MOSEL source file is the input for the MOSEL system. The MOSEL system uses a macro-like language (see [8] and [4]) tuned especially to describe stochastic petri nets. The output of the MOSEL system is a C program, which uses the SPNP library system to solve the Markov chains for the model. SPNP is a powerful mathematical library (see [5] and [12]), written in C and containing well-tuned Markov chain solvers. Using SPNP we can generate an executable program from the MOSEL output with a C compiler. This executable program will produce numerical results for the investigated model.

The three systems work together in the backround. The only neccessary user interaction is to edit the configuration file (see the numerical example below) and to issue the MARKMOD command. Then we will get the numerical results automatically.

4 Numerical Examples.

4.1 Random environment.

In this case we consider the model introduced by [6] in section 6. Gaver *et al.* used the terminology of machine interference problem to describe the same mathematical model (i.e. the word "repairmen" was used instead of server and "machine" instead of client). The system contains $N = 5$ clients connected to a server. All the clients and the server are collected into one group affected by a random environment. The random environment has two states denoted by 1 and 2. The clients are homogeneous with job generation rate 0.12 if the environment is in state 1, and 0.06 if it is in state 2. The service rate of the jobs is 1.0 in both environments.
Using the notations of [6] the rate of changes in the environment is $\alpha_1 = 0.5$ and $\alpha_2 = 1.0$
The MARKMOD configuration file of this model is the following:

```
5            # No. of client stations
1            # No. of background processes
2            # No. of states of the bg. processes
05           # No. of stations influenced by the bg. proc.-s, (0 = server)
0.12  0.06   # Job gen. intensities for station 1
0.12  0.06   # Job gen. intensities for station 2
0.12  0.06   # Job gen. intensities for station 3
0.12  0.06   # Job gen. intensities for station 4
0.12  0.06   # Job gen. intensities for station 5
1.00  1.00   # Job run. intensities for station 1
1.00  1.00   # Job run. intensities for station 2
1.00  1.00   # Job run. intensities for station 3
1.00  1.00   # Job run. intensities for station 4
1.00  1.00   # Job run. intensities for station 5
0.5          # Gen. Matrix for bg.pr. 1 (without main diagonal) - line1
1.0          # Gen. Matrix for bg.pr. 1 (without main diagonal) - line2
```

Performance Measures

Length of the server's queue	0.6418
Utilization of the server	0.4342
Utilization of the clients	0.8716
Response Time	1.4782

Comparing these numerical results to the ones derived from [6] we can see, that the results are the same. The running time of our software tool is about 5 seconds on a 600MHz Pentium PC, that means our software is suitable for such a problems.

4.2 Terminal system with server breakdowns.

In this case we investigate a client-server system with server's breakdowns. We consider a queueing system consisting of n clients connected with a server. Client i has exponentially distributed request generation and processing times with rate λ_i and μ_i respectively. Let us assume that the server is subject to random breakdowns stopping the server's service and the client's work too. The failure-free operation times and repair times are exponentially

distributed random variables with parameters α and β. We investigate here how the server's breakdown parameter (α) influences the utilizations and the response times.
The detailed model description can be found in [1] and [2].
To implement this system in MARKMOD we collect the clients and the server into one group influenced by a background process. The background process has two states: 0 means that the server is operational, 1 means that the system is down.

Input parameters

$n = 4$	$\alpha = 0.02 - 0.25$	$\beta = 0.45$

i	1	2	3	4
$\lambda_i(0)$	0.500	0.400	0.300	0.200
$\mu_i(0)$	0.900	0.800	0.600	0.500
$\lambda_i(1)$	0.001	0.001	0.001	0.001
$\mu_i(1)$	0.001	0.001	0.001	0.001

Performance measures (utilizations, response times)

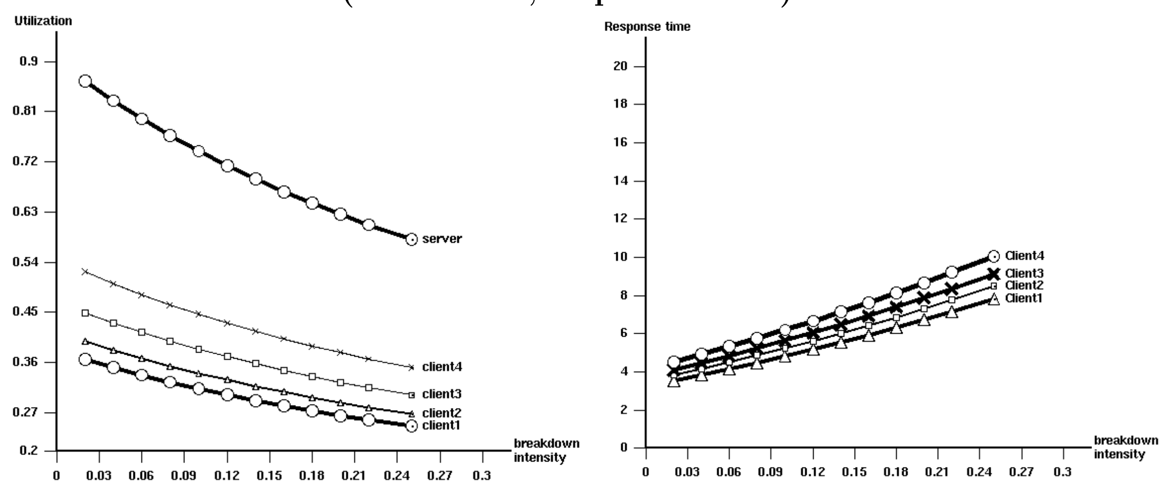

Figure 2.

We made experiment series in this case to illustrate how the server's breakdown influences the utilizations and response times (see Figure 2). It can be seen, that the utilizations are linearly decreasing (for the server and for the clients too), and the response times are linearly increasing with the server's breakdown parameter. It can be seen also, that the server's utilization is the most sensible to the parameter variation.
Also the correct implementation of the model can be seen with this example, because the performance measures for $\alpha = 0.02$ is nearly the same as in case 1 (failure free system) in [1], similarly for $\alpha = 0.25$ the outputs are the same to the ones in case 3.
The running time of the experiment series (12 experiments!) was about 1 minute, which confirms, that the software tool is really appropriate for such problems.

5 Conclusion.

In this paper a markov-modulated finite source non-homogeneous queueing model has been treated to analyse a client-server communication system. Also a software tool is introduced (based on MOSEL and SPNP) which can be used to calculate analytical results for the model. Furthermore some numerical example illustrate the problem in question and confirms, that the tool is useful for performance analysis of such systems.

References

[1] **Almási, B** A Queuing Model for a Non-Homogeneous Terminal System Subject to Breakdowns, *Computers and Mathematics with Applications* Vol. 25, No. 4, 105-111, (1993)

[2] **Almási, B** Response Time for Finite Heterogeneous Nonreliable Queueing Systems, *Computers and Mathematics with Applications* Vol. 31, No. 11, 55-59, (1996).

[3] **Almási, B. and Sztrik, J.** Optimization problems on the performance of a non-reliable terminal system, *Computers and Mathematics with Applications* Vol. 38, No. 3, (1999).

[4] **Begain, K.; Bolch, G.; Herold, H.** Practical Performance Modeling Kluwer, (2001).

[5] **Ciardo, G.; Muppala, J.K.** Manual for the SPNP Package Version 3.1, Duke University, Durham, NC, USA, (1991).

[6] **Gaver D.P., Jacobs P.A., Latouche G.** Finite birth-and-death models in randomly changing environments, *Advances in Applied Probability* 16, pp. 715-731, (1984).

[7] **Haverkort B.** *Performance of Computer Communication Systems,* John Wiley & Sons, (1998).

[8] **Herold, H.** MOSEL An Universal Language for Modeling Computer, Communication and Manufacturing Systems. *Phd Dissertation,* University Erlangen, (2000).

[9] **Kameda, H.** A finite-source queue with different customers. J. ACM 29, 478-491, (1982).

[10] **Sztrik J., Kouvatsos D.D.** Asymptotic Analysis of a Heterogeneous Multiprocessor System in a Randomly Changing Environment, *IEEE Transactions on Software Engineering* 17, pp. 1069-1075, (1991).

[11] **Sztrik J., Moeller O.** A tool for simulation of Markov-modulated finite-source queueing systems, *Proceedings of Messung Modellirung und Berwertung (MMB99), Trier, Germany* pp. 109-114, (1999).

[12] **Trivedi, K.S.; Ciardo, G.** A Decomposition Approach for Stochastic Reward Net Models, Duke University, Durham, NC, USA, (1991).