УДК 519.872 **DOI** 10.15622/sp.49.3

A.D. KHOMONENKO, M.M. KHALIL, D.T. KASSYMOVA

PROBABILISTIC MODELS FOR EVALUATING THE PERFORMANCE OF CLOUD COMPUTING SYSTEMS WITH WEB INTERFACE

Khomonenko A.D., Khalil M.M., Kassymova D.T., **Probabilistic Models for Evaluating the Performance of Cloud Computing Systems with Web Interface.**

Abstract. For cloud computing systems with web interface a set of probabilistic models is proposed. At the same time a model of Java applications with a web interface based on servlet and filters is considered. These models are based on queuing theory and extend its applications by studying the multichannel systems with "warm-up", "cooling" and phase-type approximation of Markovian and non-Markovian processes. Transition diagrams and matrices for the microstates of queuing systems being models of applications with a web interface are described, and a scheme for computing the stationary probability distributions for requests number, waiting time and expected time in system is being developed. The paper discusses the received computation results of the proposed modeling approach and their application to assessing the performance of the cloud systems using applications based on servlet and filters.

Keywords: cloud computing, web interface, servlet and filters, performance evaluation, queuing system, warm-up and cooling, waiting time in queue.

1. Introduction. An important aspect of software products is their performance. It has become critical for the cloud computing services generally accessed from the web, where the nature of service depends heavily on the performance level. Measuring the performance starts from early design phase with the preliminary efficiency evaluation. This allows estimating in advance the planned performance level and preventing service unavailability under loading, helping plan the equipment costs for the given parameters of the system configuration and performance metrics. With effective planning of computing resources during the development phase, the development time frames could be reduced and, therefore, the costs.

Mathematical modelling plays an important role in analysis the performance parameters of modern cloud computing systems. As they grow bigger and become more complex, the mathematical models respectively take steps from simple ones, when the solution is obtained analytically, towards models, where it is only possible to compute the solution during simulation process. Performance estimates are based on the queuing theory, a discipline within the mathematical theory of probability, which studies waiting lines, or queues. In queuing theory a model of queuing system (QS) is constructed and its life cycle is scrutinized as a stochastic process to predict the probability characteristics of efficiency such as queue lengths and waiting times.

This paper examines multichannel Markovian and non-Markovian queues with warm-up and/or with cooling, which take account of the features

of the functioning of Java applications with web interface. Transition diagrams and matrices for the microstates of a multichannel queues modeling a sample applications with web interface on the basis of servlet and filters are described and a scheme for computing the stationary probability distributions for requests number and waiting time is developed. The paper discusses the received computation results of the proposed modeling approach and their application to assessing the performance of the cloud systems using applications with web interface on the basis of servlet and filters.

2. Common Characteristics of Cloud Computing with Web Interface. Cloud computing is a number of computing resources that are delivered as a service over a network connection (usually the Internet). Therefore, cloud computing relies on sharing a pool of physical and/or virtual resources, rather than deploying local or personal hardware and software. A user is able to connect into a supply of computing resources rather than managing, operating, maintaining, securing and servicing the equipment needed to generate it themselves; just as the same way as a consumer tapping into the national electricity supply, instead of running their own electric generator. Thereby the service is billed likewise where you pay for the amount of service that you consume.

Also likewise cloud computing is more reliable, flexible, scalable and most importantly more economical. You can choose between different types of clouds depending on the amount of security that you demand for your business, these types are: public, private and hybrid cloud. You can also choose the type of service that you need for your business, the major types of services are: SaaS (Software as a service); PaaS (Platform as a service) and IaaS (Infrastructure as a service).

SaaS, or Software as a Service, describes any cloud service where consumers are able to access software applications over the Internet. The applications are hosted in “the cloud” and can be used for a wide range of tasks for both individuals and organisations. Google, Twitter, Facebook and Flickr are all examples of SaaS, with users able to access the services via any Internet enabled device.

In order to connect with devices that are different in size, shape, capacity, power, user language and operating system, a very simple and global interface is needed, that can connect to devices using very simple protocols. Today almost any device can be connected to the Internet and use different protocols to communicate, that’s why if you need to move your applications and data to the cloud, it would be easier and better if the applications had a web interface, as the whole idea of moving to the cloud is so that you can have access anytime from any device with maximum reliability and unlimited computing power.

Internet connections use protocols to connect between different devices, so that after every handshake every device knows what it should do

and how to do it. A web service, in very broad terms, is a method of communication between two applications or electronic devices over the WWW. Web services are of two kinds: Simple Object Access Protocol (SOAP) and Representational State Transfer (REST).

For development applications with web interface the Java platform is widely used.

3. Web Interface on Basis of Java Platform. Applications with web interface on Java platform used servlets and applets. *Servlet* is a Java-program running on the server side and expand server functionality. The servlet communicates with clients by processing HTTP requests. *Filters* allow performing actions on the request before it is processed by the servlet or on the response after its formation. Filters implement model "interceptor request", and for each resource in the web application has its own filter chain can be arranged.

The *client* can use any device he likes, as long as that device has a connection to the Internet and a web browser. Almost no or very little amount of data need to be stored or computed in the client side, all the data is stored and processed in the cloud's side.

The request then reaches the servlet container in the web application to be processed, the servlet must process the request and generate as much of the response as the application requires. A servlet is a small program that runs on a *server*. The term usually refers to a Java applet that runs within a web server environment.

The HttpServlet class reads the HTTP request, and determines if the request is an HTTP GET, POST, PUT, DELETE, HEAD etc. and calls one the corresponding method.

But, there are many cases where some pre-processing of the request for servlets would be useful. In addition, it is sometimes useful to modify the response from a class of servlets. One example is encryption. A servlet, or a group of servlets in an application, might generate response data that is sensitive and should not go out over the network in clear-text form, especially when the connection has been made using a non-secure protocol such as HTTP.

A filter can encrypt the responses. A filter is where you want to apply pre-processing or post-processing to requests or responses for a group of servlets, not just a single servlet, so when your application is working in a cloud environment filters are very useful tools. Filters are designed to be able to manipulate a request or response (or both) that is sent to a web application, yet provide this functionality in a method that won't affect servlets and JSPs being used by the web application unless that is the desired effect.

A good way to use filters is as a chain of steps that a request and response must go through before reaching a servlet, JSP, or static resource such

as an HTML page in a web application. Figure 1 shows the commonly used illustration of this concept.

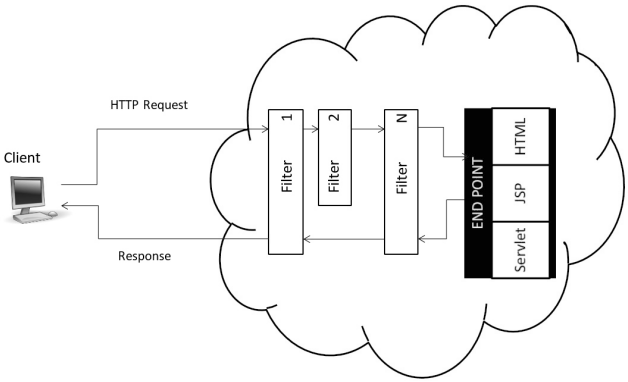


Fig. 1. Using filters with servlets

Possible main usage of filter include following:

– recording all incoming requests;

– logs the IP addresses of the computers from which the requests originate;

– conversion;

– data compression;

– encryption and decryption;

– input validation etc.

In listing 1 an example of a simple log filter is shown:

```
public class LogFilter implements Filter  {
  public void  doFilter(ServletRequest request,
            ServletResponse response,
            FilterChain chain)
            throws java.io.IOException, ServletException {
    // Get the IP address of client machine.
    String ipAddress = request.getRemoteAddr();
    // Log the IP address and current timestamp.
    System.out.println("IP "+ ipAddress + ", Time "
                        + new Date().toString());
    // Pass request back down the filter chain
    chain.doFilter(request,response);
  }
}
```

Listing 1. Example of a simple log filter

As can be seen from Figure 1 the filter or filter chain can run before call of servlet (or JSP, or HTML) and/or after finishing servlet execution. In first case (at the entry of servlet) filter can make any preparatory work (for example, decompression of data), that named by *"warming-up"* or *"warm-up"*. In second case (after returning from servlet) filter can make any final work (for example, data compression), that named by *"cooling"*.

Let us consider how possible modeling peculiar properties of the process functioning of application with web interface using models of queuing theory.

4. Existing Models. In queuing theory, the Kendall's notation is the standard system to describe and classify a queuing node. It uses three factors written $A/S/n$, where A denotes the time between arrivals to the queue, S — the service time distribution and n denotes the number of servers at the node. It been extended to $A/S/n/K/N/D$ where K and D mean the capacity of the queue and queuing discipline and N denotes the size of the population of jobs to be served. Best studied QS M/M/n — with n channels, service time distribution and the distribution for the time between arrivals follow an exponential distribution. Because of the assumptions made, such models known as Markov models, have limited appliance and do not fit for most practical systems.

The most examined are the relatively simple M/M/n models. The simplest example is M/M/1 queue, for which textbooks pertaining to performance evaluation usually present the results to compute the steady state distribution of number of requests. Well studied class of one-channel models with specific flow characteristics, discussed e.g. by Ryzhikov in [22] or Eremin [9], which analyse the behaviour QS with determined delay in starting the service.

The biggest interest was recently been focused on the investigations in multichannel non-Markovian queues where flows are approximated by phase-type distributions. For example, Bubnov et al. [4] or Danilov et el. [8] models to forecast software reliability characteristics, such as number of corrected errors, required debugging time, etc. Brandwajn and Begin in [2] propose a semi-numerical approach to compute the steady-state probability distribution for the number of requests at arbitrary and at arrival time instants in Ph/M/c-like systems.

Cox showed in [7] that an arbitrary distribution of length of a random variable can be represented by a compound of exponential stages or phase-type distribution. The advantage of such a representation is that it ensures convenience of approximation of the random process to a Markov process and gives the power of creating and solving the system of equations describing the behaviour of the corresponding model.

Described here multichannel non-Markovian QS with warm-up require more complex mathematical description, compared to the Markov mod-

els, e.g. the request flow can be recurrent or represented by an arbitrary stochastic function. Examples of previous works addressing QS with warm-up are by Kolahi in [18] or by Kreinin in [19] for the characteristics of single channel QS, or by Bin Sun, A. N. Dudin in [1] studying the MAP/PH/n multichannel QS with warm-up and broadcasting service discipline. Mao and Humphrey in [20] examine the influence of the warm-up during virtual machine start-up in the cloud system.

Examples of early works on cloud performance subject can be found in [9] and [23]. In a fairly recent work of several authors investigate questions of evaluation of performance of cloud and other systems on the basis of models of multi-channel queuing systems with heating [15, 16] and cooling [17, 21].

In cloud computing application with web interface the usage of filter adds additional costs:

– for the "warm-up" (for example, for decompression of data) when filter is used at the entry of servlet;

– for the "cooling" (for example, data compression) when filter act after returning from servlet.

– To study the cloud systems with described ***warm-up*** it is useful to introduce an enhanced notation A/W/S/n, which compared to original Kendall's notation contains additional *W* denoting the warm-up time distribution.

5. Cloud Computing Using Web Interface Models with "Warm-up". Principally, in real systems all three properties might be non-Markovian. The systems where A — the incoming flow — is approximated by two-phase Hyper exponential distribution (H_2) have been studied in [7] and [8]. Below a model with two-phase H_2-distribution of warm-up process (W) is examined.

Let's describe the parameters of a model which is set up. For modelling simplicity let's denote status of a multichannel non-Markov QS in the form of a set of microstates. Microstates are all sorts of states which the system might be in while in operation.

The QS has the classification $M/H_2/M/n$ and has the microstate diagram, as shown in Figure 2.

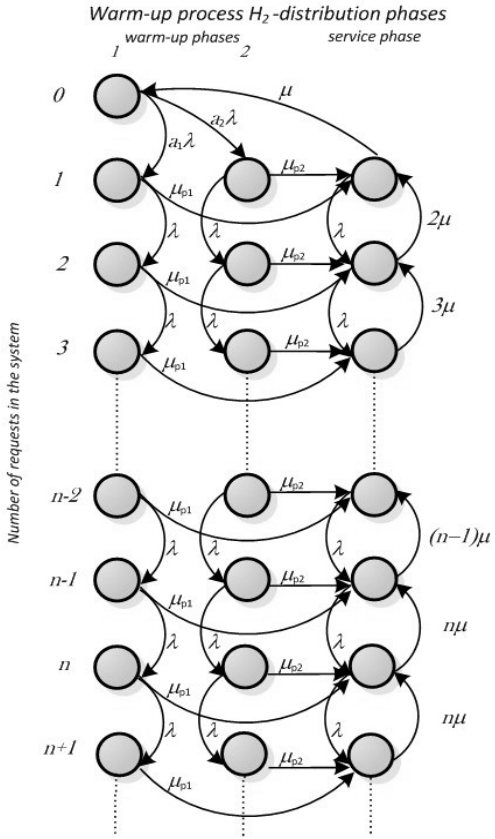


Fig. 2. Diagram for $\boldsymbol{M/H_2/M/n}$ system with "warm-up"

The warm-up process has Hyper exponential distribution H_2 with parameters: μ_{p_1} and μ_{p_2} — intensities of transition in corresponding phases, a_1 — probability of choosing the first phase, $a_2 = 1 - a_1$ — probability of choosing the second phase; parameters λ and μ describe the intensity of the input and service flows in the system.

The column on the left shows the number of requests in the system. An n-channel system after reaching the full load (on the diagrams — layers with numbers greater than *n*) eventually stabilizes.

Let us consider the QS $M/H_2/M/n$ for $n = 3$. For computation a numerical method for the steady-state probabilities of a GI/G/n QS in a general

class, introduced by Takahashi and Takami [22], and widely used for different multi-channel not Markov QS (for example [14]) can be used.

Denote as S*j* the set of system microstates when exactly *j* requests are served, and by σ*j* — number of elements in S*j*. Then from the microstates diagram, analytically the following matrices describing the system are defined:

- $A_j[\sigma_j \times \sigma_{j+1`}]$ — in S_{j-1} (request arrival);
- $B_j[\sigma_j \times \sigma_{j-1`}]$ — in S_{j-1} (request service completion);
- $C_j[\sigma_j \times \sigma_j]$ — in S_j (request service in progress);
- $D_j[\sigma_j \times \sigma_j]$ — leaving microstates of tier *j* (a diagonal matrix).

For each tier j denote by vectors $\gamma_j = \{\gamma_{j,1}, \gamma_{j,1}, ..., \gamma_{j,\sigma_j}\}$ the probability that a QS is in microstate *(j, i), j* = 0,1,… Then it is possible to write the system of vector-matrix balance equations describing transitions between microstates:

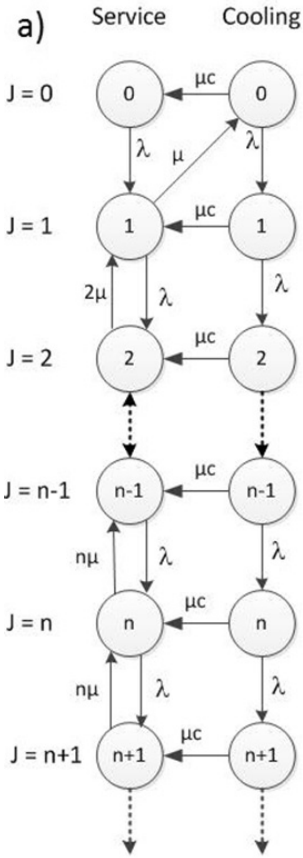
$$\gamma_0 D_0 = \gamma_0 C_0 + \gamma_1 B_1,$$
$$\gamma_j D_j = \gamma_{j-1} A_{j-1} + \gamma_j C_j + \gamma_{j+1} B_{j+1}, \quad j = 1,2,...$$

An iterative numerical method introduced by Takahashi and Takami in [24] is then used to solve the system and find the steady-state microstates probability distribution. This classical iterative method chosen for it well known convergence properties [6]. To the authors' is known about global convergence theorem and speed of convergence for competitive iterative algorithms except the direct successive substitution method.

After that the mean parameters of requests servicing process are obtained using Little's law and the waiting time distributions for waiting time in the queue, time in the system, the distribution of the number of requests in the queue and in the system are obtained using the results of [13] to compute the Laplace-Stieltjes transform.

6. Models with "Cooling". To study the cloud systems with described ***"cooling"*** it is useful to introduce an enhanced notation *A/C/S/n*, which compared to original Kendall's notation contains additional C denoting the cooling time distribution.

Let's describe the parameters of a model which is set up. For example note that QS *M/M/M/n* with cooling has the microstate diagram and matrixes, shown in the Figure 3 a) and b).



b)

$$A_0 = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix};$$

$$A_j = A_0 \ , \quad j \geq 1.$$

$$B_j = \begin{bmatrix} 0 & j\mu \\ 0 & 0 \end{bmatrix}, \quad 1 < j \leq n;$$

$$B_j = B_n \ , \quad j > n.$$

$$C_j = \begin{bmatrix} 0 & 0 \\ \mu_C & 0 \end{bmatrix} , \quad j \geq 0.$$

$$D_0 = \begin{bmatrix} \lambda & 0 \\ 0 & \mu_C + \lambda \end{bmatrix};$$

$$D_j = \begin{bmatrix} j\mu + \lambda & 0 \\ 0 & \mu_C + \lambda \end{bmatrix}, \quad j \leq n;$$

$$D_j = D_n \ , \quad j > n.$$

Fig. 3. Diagram and matrix for M/M/M/n QS with "cooling"

The cooling process has only one parameter μ_C — intensity of "cooling".

7. Practical Results. A Java program has been developed to implement the described numeric method. It has been designed to perform actions on matrices in general, and therefore it allows obtaining results for both presented models as well as for models of other systems within the comparable classes of phase-type distributions.

The initial data has been produced from test statistics gathered for a sample cloud system as shown on Figure 1. To collect it a simulation performance testing in an Apache JMeter tool was run with 300 threads in parallel and independently performing queries of various kinds.

The following computational resources were used to perform the testing:

Application servers have the following characteristics:

– CPU Intel Xeon X7560 (2.26 GHz), 8 cores.
– Hyperthreading disabled.
– 512Gb RAM.
– Memory allocated in the range of 6-8 Gb per node.
– Average CPU load in the range 0.3-0.5 per core.

Network segments have the following characteristics:

– Nodes reside in the same logical network segment.
– 100/1000 Mbps Ethernet ports are used.

It is useful to experiment with the model parameters under different scenarios and determine the conditions for receiving the desired QoS level.

In a simulation for a system corresponding to the model shown on Figure 4 with static parameters: mean warm-up time changing between 0.1 and 0.9 seconds the following results were received for the expected waiting time in the queue at different values, as described in Figure 4.

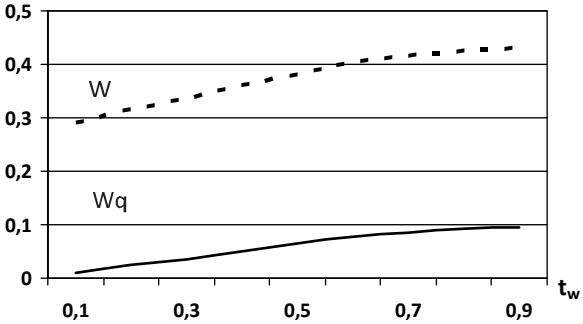


Fig. 4. Expected waiting and sojourn time in H2/M/M/3 QS with “warm-up”

The corresponding experiment results shown in Figure 4 show waiting time in the queue — Wq, and expected time in system, including service time — W at different values of the mean warm-up time t_w between 0,1 and 0,9 seconds.

Results show that the “warm-up” time change may significantly affect the waiting time and service time, especially when the “warm-up” and service times are of the same order of magnitude. On the right side of the Figure 4,

when the “warm-up” time is similar or even bigger to the service time, reducing the “warm-up” would result in a substance speed up of the service. But in the left side of Figure 4, the performance curve gets steeper as the “warm-up” time starts to be similar or less than the service time.

A simulation in [17] has been held for a *M/M/M/n* queuing system with “*cooling*” with a stable input flow to examine the behaviour changes of the expected number of customers in the system — L, expected queue length — Lq, waiting time in the queue — Wq, and expected time in system — W, while changing the cooling frequency. The parameters for that run were: λ=4.0, μ=1.8, μ_c=ranges from 0.5 to 3.5 seconds. The following results were received, as described in Figure 5.

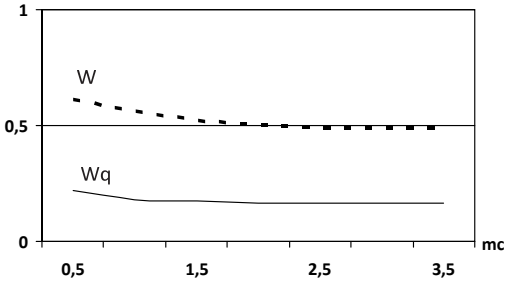


Fig. 5. Average waiting and sojourn time in M/M/M/n with “cooling”

The study demonstrates the influence of the cooling patterns on the performance and shows the need to collect data and examine cooling patterns to assure that the system capabilities are appropriate for significantly different levels and patterns of demand that might be relevant during a given time period.

8. Conclusion. Our paper proposed uses of models of multi-channel queuing systems with "warm-up" and "cooling", which for the user interface of the web application allows to take into account the additional costs of employment of the filter, respectively: when the filter is used in the servlet's input; when the filter is used after the return from the servlet.

The further study is to focus on:

– Expanding the characteristic distribution function of the described models on to the networks of QS with multiple nodes;

– Expanding the research on Hyper exponential distribution case with complex coefficients;

– Expanding the modelling class on multichannel systems with both the warm-up time distribution and the distribution for the time between arrivals approximated by phase-type distributions.

Further investigation advisable to continue in the direction of the development of models of multi-channel QS with phase-type distributions (Erlang, Hyper exponential et al.). For example, Figure 6 proposed transition diagram for the QS type $M / E_2 / E_2 / M / n$ — with a Poisson input flow with intensity λ, generalized Erlang distribution of order 2 the duration of the "warm-up" and "cooling" with parameters $\{\mu_{W1}, \mu_{W2}\}$ and $\{\mu_{C1}, \mu_{C2}\}$, exponentially distributed service time with intensity μ.

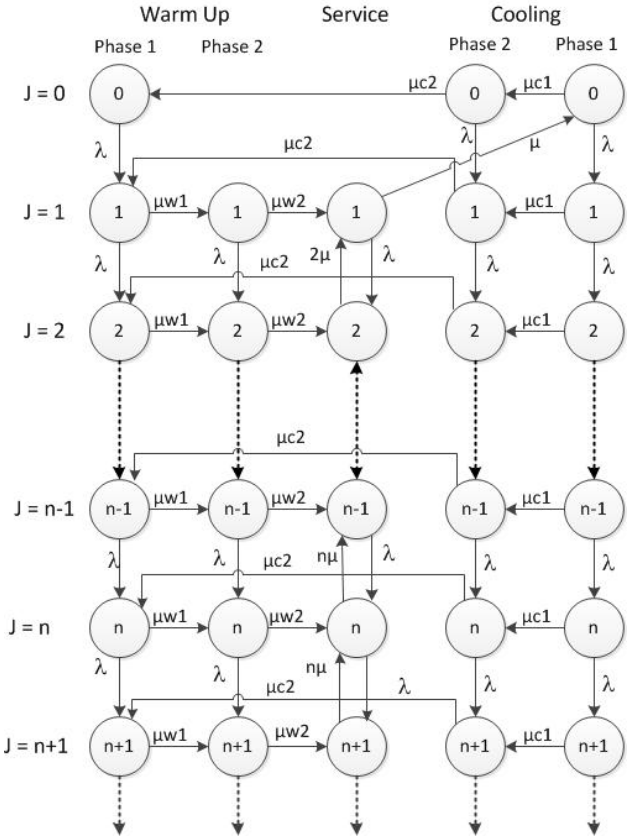


Fig. 6. Diagram for $M / E_2 / E_2 / M / n$ QS with "warm-up" and "cooling"

For the proposed model QS it is required to check, as far as the transition between the microstates correspond to the behavior of the application with a web

interface. In addition, it should develop a matrix of transitions between states of the QS and implement a calculation of the basic output characteristics.

References

1. Sun B., Dudin A.N. The MAP/PH/N multi-server queuing system with broadcasting service discipline and server heating. *Automatic Control and Computer Sciences*. 2013. vol. 47. no. 4. pp. 173–182.
2. Brandwajn A., Begin. T. A recurrent solution of Ph/M/c/N-like and Ph/M/c-like queues. *Journal of Applied Probability*. 2012. vol. 49(1). pp. 84–99.
3. Bruneo D. A stochastic model to investigate data center performance and qos in iaas cloud computing systems. *IEEE Transactions on Parallel and Distributed Systems*. 2014. pp. 560–569.
4. Mukhar K. et al. Beginning Java EE 5 From Novice to Professional. Apress. 2006. 641 p.
5. Bubnov V.P., Tyrva A.V., Khomonenko A.D. Model of reliability of the software with Coxian distribution of length of intervals between the moments of detection of errors. In proceedings of 34th Annual IEEE Computer Software and Applications Conference (COMPSAC 2010). 2010. pp. 238–243.
6. *Cao W., Stewart W.T.* Iterative Aggregation/Disaggregation Techniques for Nearly Uncoupled Markov Chains. *Journal of the ACM*. 1985. vol. 32. pp. 702–719.
7. Cox D.R. A use of complex probabilities in the theory of stochastic processes. Proc. Cambr. Phil. Soc. 1955. vol. 51. no. 2. pp. 313–319.
8. Homonenko A.D., Danilov A.I., Danilov A.A., Gerasimenko P.V. [Nonstationary models of debugging of programs with Cox's distribution duration of correction of errors]. *XIX Mezhdunarodnaja konferencija po mjagkim vychislenijam i izmerenijam (SCM-2016)* [XIX International Conference on Soft Computing and Measurements (SCM-2016)]. 2016. Issue 1. pp. 163–166. (In Russ.).
9. Eremin A.S. A Queuing System with Determined Delay in Starting the Service. *Intellektual'nye tehnologii na transporte – Intellectual Technologies on Transport*. 2015. vol. 4. pp. 23–26.
10. Gong C. et al. The characteristics of cloud computing. 39th International Conference on Parallel Processing Workshops (ICPPW). IEEE Press. 2010. pp. 275-279.
11. Grassmann, W.K. Warm-up periods in simulation can be detrimental. *Probab. Engrg. Inform. Sci*. 2008. vol. 22(3). pp. 415–429.
12. van Hoorn M.H., Seelen L.P. Approximations for the GI/G/c Queue. *Journal of Appl. Probability*. 1986. vol. 23. no. 2. pp. 484–494.
13. Khomonenko A.D. [Waiting time distribution in queuing systems of type GIq/Hk/n/R≤∞]. *Avtomatika-i-telemekhanika – Automation and Remote Control*. 1990. no. 8. pp. 91–98. (In Russ.).
14. Khomonenko A.D., Bubnov V.P. A use of Coxian distribution for iterative solution of M/G/N/R≤ ∞ queuing systems. *Problems of Control and Information Theory*. 1985. vol. 14. no. 2. pp. 143–153.
15. Khomonenko A.D., Gindin S.I. Stochastic models for cloud computing performance evaluation. Proceedings of the 10th Central and Eastern European Software Engineering Conference in Russia. ACM. 2014. pp. 20. Available at: http://dl.acm.org/citation.cfm?id=2687233. (accessed 25.11.2016).
16. Khomonenko A., Gindin S. Performance evaluation of cloud computing accounting for expenses on information security. 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT). 2016. pp. 100–105.

vychislenijam i izmerenijam (SCM-2016) [XIX International Conference on Soft Computing and Measurements (SCM-2016)]. 2016. Issue 1. pp. 247–251. (In Russ.).
18. Kolahi S.S. Simulation Model, Warm-up Period, and Simulation Length of Cellular Systems. Second International Conference on Intelligent Systems, Modelling and Simulation. IEEE. 2011. pp. 375–379.
19. Kreinin Ya. Single-channel queuing system with warm up. *Automation and Remote Control*. 1980. vol. 41. no. 6. pp. 771–776.
20. Mao M., Humphrey M. A performance study on the vm startup time in the cloud. IEEE 5th International Conference on Cloud Computing (CLOUD). IEEE Press. 2012. pp. 423–430.
21. Lokhvitskii V.A., Ulanov A.V. [Numerical analysis of queuing systems with Hyper exponential "cooling"]. *Vestnik-tomskogo-gosudarstvennogo-universiteta-upravlenie-vychislitelnaya-tekhnika-i-informatika – Tomsk State University Journal of Control and Computer Science.* 2016. no. 4(37). (In Print) (In Russ.).
22. Ryzhikov Yu.I. [Distribution of the Number of Requests in a Queuing System with warm-up]. *Problemy-peredachi-informacii – Problems of Information Transmission.* 1973. vol. 9(1). pp. 88–97. (In Russ.).
23. Takahashi Y., Takami Y. A numerical method for the steady-state probabilities of a GI/G/c queuing system in a general class. J. *of the Operat. Res. Soc. of Japan*. 1976. vol. 19. no. 2. pp. 147–157.
24. *Xiong K., Perros H.* Service performance and analysis in cloud computing. World Conference on Services-I. IEEE Press. 2009. pp. 693–700.

Khomonenko Anatoly Dmitrievich — Ph.D., Dr. Sci., professor, head of information and computing systems department, Emperor Alexander I St. Petersburg State Transport University. Research interests: queuing systems, artificial intelligence, databases. The number of publications — 150. khomon@mail.ru, http://www.pgups.ru; 9, Moskovsky pr., Saint Petersburg, 190031; office phone: 8(812)457-80-23, Fax: 8(812)310-75-25.

Khalil Maad Modher — Ph.D. srudent, Emperor Alexandr I Petersburg State Transport University. Research interests: cloud computing, modelling, queuing systems. The number of publications — 4. maadalomar@gmail.com; 9, Moskovsky pr., Saint Petersburg, 190031; office phone: +79643312131, Fax: +7(812)310-75-25.

Kassymova Dinara Tugelbekovna — Ph.D. student of information and computer technology institute, Kazakh National Research Technical University named after K.I.Satpayev (KazNTU after K.I.Satpaev), senior lecturer, Turan University. Research interests: cloud computing, modelling. The number of publications — 2. dika.cat@mail.ru; 110, Dostyk Ave., Almaty, 050010; office phone: +7(727)2463982, Fax: +7(727)2604023.

DOI 10.15622/sp.49.3

А.Д. Хомоненко, М.М. Халил, Д.Т. Касимова

ВЕРОЯТНОСТНЫЕ МОДЕЛИ ДЛЯ ОЦЕНИВАНИЯ ОПЕРАТИВНОСТИ СИСТЕМ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ С ВЕБ-ИНТЕРФЕЙСОМ

Хомоненко А.Д., Халил М.М., Касимова Д.Т. **Вероятностные модели для оценивания оперативности систем облачных вычислений с веб-интерфейсом.**

Аннотация. Для систем облачных вычислений с веб-интерфейсом предлагается ряд вероятностных моделей. При этом рассматриваются модели Java-приложений с веб-интерфейсом, построенных на основе сервлетов и фильтров. Эти модели основаны на теории массового обслуживания и расширяют ее приложения путем изучения многоканальных систем с «разогревом», «охлаждением» и аппроксимирующими распределениями фазового типа для Марковских и немарковских процессов. Приводятся примеры диаграмм и матриц переходов между микросостояниями систем массового обслуживания, являющихся моделями приложений с веб-интерфейсом, а также разрабатывается схема для вычисления стационарного распределения числа заявок в системе, времени ожидания в очереди и пребывания в системе. В статье обсуждаются результаты численных расчетов, полученные с помощью предлагаемого подхода и их применение для оценки оперативности функционирования облачных систем с приложениями на основе сервлетов и фильтров.

Ключевые слова: облачные вычисления, веб-интерфейс; сервлеты и фильтры, оценка оперативности; системы массового обслуживания; разогрев и охлаждение; время ожидания в очереди.

Хомоненко Анатолий Дмитриевич — д-р техн. наук, профессор, заведующий кафедрой информационных и вычислительных систем, ФГБОУ ВО Петербургский государственный университет путей сообщения Императора Александра I. Область научных интересов: численная теория массового обслуживания, программирование, операционные и информационные системы. Число научных публикаций — 150. khomon@mail.ru, http://www.pgups.ru; Московский пр., 9, Санкт-Петербург, 190031; р.т.: 8(812)457-80-23, Факс: 8(812)310-75-25.

Халил Маад Модер — аспирант, ФГБОУ ВО Петербургский государственный университет путей сообщения Императора Александра I. Область научных интересов: облачные вычисления, моделирование, системы массового обслуживания. Число научных публикаций — 4. maadalomar@gmail.com; Московский пр., 9, Санкт-Петербург, 190031; р.т.: +79643312131, Факс: +7(812)310-75-25.

Касымова Динара Тугелбековна — докторант института информации и компьютерных технологий, Казахский национальный исследовательский технический университет имени К.И. Сатпаева (КазНИТУ им. К.И. Сатпаева), старший преподаватель, Университет "Туран". Область научных интересов: облачные вычисления, моделирование. Число научных публикаций — 2. dika.cat@mail.ru; пр. Достык, 110, Алматы, 050010; р.т.: +7(727)2463982, Факс: +7(727)2604023.

Литература

1. *Sun B., Dudin A. N.* The MAP/PH/N multi-server queuing system with broadcasting service discipline and server heating // Automatic Control and Computer Sciences. 2013. vol. 47. no. 4. pp. 173–182.

2. *Brandwajn A., Begin. T.* A recurrent solution of Ph/M/c/N-like and Ph/M/c-like queues // Journal of Applied Probability. 2012. vol. 49(1). pp. 84–99.
3. *Bruneo D.* A stochastic model to investigate data center performance and qos in iaas cloud computing systems // IEEE Transactions on Parallel and Distributed Systems. 2014. pp. 560–569.
4. *Mukhar K. et al.* Beginning Java EE 5 From Novice to Professional // Apress. 2006. 641 p.
5. *Bubnov V.P., Tyrva A.V., Khomonenko A.D.* Model of reliability of the software with Coxian distribution of length of intervals between the moments of detection of errors // In proceedings of 34th Annual IEEE Computer Software and Applications Conference (COMPSAC 2010). 2010. pp. 238–243.
6. *Cao W., Stewart, W.T.* Iterative Aggregation/Disaggregation Techniques for Nearly Uncoupled Markov Chains // Journal of the ACM. 1985. vol. 32. pp. 702–719.
7. *Cox D.R.* A use of complex probabilities in the theory of stochastic processes // Proc. Cambr. Phil. Soc. 1955. vol. 51. no. 2. pp. 313–319.
8. *Хомоненко А.Д., Данилов А.И., Данилов А.А., Герасименко П.В.* Нестационарные модели отладки программ с распределением Кокса длительности исправления ошибок // XIX Международная конференция по мягким вычислениям и измерениям (SCM-2016). 2016. Т. 1. С. 163–166.
9. *Eremin A.S.* A Queuing System with Determined Delay in Starting the Service // Интеллектуальные технологии на транспорте. 2015. № 4. С. 23–26.
10. *Gong C. et al.,* The characteristics of cloud computing // 39th International Conference on Parallel Processing Workshops (ICPPW). IEEE Press. 2010. pp. 275–279.
11. *Grassmann W.K.* Warm-up periods in simulation can be detrimental // Probab. Engrg. Inform. Sci. 2008. vol. 22(3). pp. 415–429.
12. *van Hoorn M. H., Seelen L. P.* Approximations for the GI/G/c Queue // Journal of Appl. Probability. 1986. vol. 23. no. 2. pp. 484–494.
13. *Хомоненко А.Д.* Распределение времени ожидания в системах массового обслуживания типа GIq/Hk/n/R≤∞ // Автоматика и телемеханика. 1990. № 8. С. 91–98.
14. *Khomonenko A.D., Bubnov V.P.* A use of Coxian distribution for iterative solution of M/G/N/R≤ ∞ queuing systems // Problems of Control and Information Theory. 1985. vol. 14. no. 2. pp. 143–153.
15. *Khomonenko A. D., Gindin S. I.* Stochastic models for cloud computing performance evaluation // Proceedings of the 10th Central and Eastern European Software Engineering Conference in Russia. ACM. 2014. pp. 20. URL: http://dl.acm.org/citation.cfm?id=2687233. (дата обращения: 25.11.2016).
16. *Khomonenko A. D., Gindin S. I.* Performance Evaluation of Cloud Computing Accounting for Expenses on Information Security // 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT). 2016. pp. 100–105.
17. *Хомоненко А.Д., Халиль М.М., Гиндин С.И.* Моделирование облачных вычислений с использованием многоканальной системы массового обслуживания с «охлаждением» // XIX Международная конференция по мягким вычислениям и измерениям (SCM-2016). 2016. Т. 1. С. 247–251.
18. *Kolahi S.S.* Simulation Model, Warm-up Period, and Simulation Length of Cellular Systems // Second International Conference on Intelligent Systems, Modelling and Simulation. IEEE. 2011. pp. 375–379.
19. *Kreinin Ya.* Single-channel queuing system with warm up // Automation and Remote Control. 1980. vol. 41. no. 6. pp. 771–776.
20. *Mao M., Humphrey M.* A performance study on the vm startup time in the cloud // IEEE 5th International Conference on Cloud Computing (CLOUD). IEEE Press. 2012. pp. 423–430.

21. *Лохвитский В.А., Уланов А.В.* Численный анализ системы массового обслуживания с гиперэкспоненциальным «охлаждением» // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2016. №4(37). (в печати).
22. *Рыжиков Ю.И.* Распределение числа требований в системе массового обслуживания с “разогревом” // Пробл. передачи информ. 1973. № 9(1). С. 88–97.
23. *Takahashi Y., Takami Y.* A numerical method for the steady-state probabilities of a GI/G/c queuing system in a general class // J. of the Operat. Res. Soc. of Japan. 1976. vol. 19. no. 2. pp. 147–157.
24. *Xiong K., Perros H.* Service performance and analysis in cloud computing // World Conference on Services-I. IEEE Press. 2009. pp. 693–700.