OPERATIONS RESEARCH AND DECISIONS

No. 3 2017
DOI: 10.5277/ord170305

Iuliia V. IGNATOVA[1]
Anatoliy V. BIEHUN[2]

ESTIMATING THE RELIABILITY OF THE ELEMENTS OF CLOUD SERVICES

Cloud technologies are a very considerable area that influence IT infrastructure, network services and applications. Research has highlighted difficulties in the functioning of cloud infrastructure. For instance, if a server is subjected to malicious attacks or a force majeure causes a failure in the cloud's service, it is required to determine the time that it takes the system to return to being fully functional after the crash. This will determine the technological and financial risks faced by the owner and end users of cloud services. Therefore, to solve the problem of determining the expected time before service is resumed after a failure, we propose to apply Markovian queuing systems, specifically a model of a multi-channel queuing system with Poisson input flow and denial-of-service (breakdown).

Keywords: *cloud technologies, performance evaluation, Markovian queuing systems, Poisson input flow*

1. Introduction

The global integration of information systems which is based on the implementation of cloud technologies and networks via the Internet, has led to the need to solve two main tasks: 1) diagnosing the functioning of cloud servers, workstations and software, 2) protection of information resources against unauthorized influence on a system from different sources. The central issue in solving these problems is to obtain timely information on the following: the state of a cloud, the possible complete or partial loss of working capacity as a result of a failure due to accident or intent, as well as unauthorized access to a cloud's resources.

[1] Department of Economic and Mathematical Modelling, Kyiv National Economic University named after Vadym Hetman, 54/1 Prospect Peremogy, 03057 Kyiv, Ukraine, e-mail address: yuliia.ihnatova@kneu.ua

[2] Department of Information Management, Kyiv National Economic University named after Vadym Hetman, 54/1 Prospect Peremogy, 03057 Kyiv, Ukraine, e-mail address: anatolii.biehun@kneu.ua

In business terms, the main characteristic of cloud technologies is the pay-per-use principle. The end user pays only for the time he uses cloud services. This characteristic distinguishes a cloud technology's business model from, for example, hosted services. This leads to the second characteristic of cloud technologies: the use of a service only on-demand. Another main characteristic is the provision of cloud services to multiple users by the same infrastructure. In the event of a breakdown or occurrence of a force majeure, the server intends that the end user switches to other resources (this load is transferred to other servers) which provides the user with a stable technical resource. In turn, the user remains connected to a particular cloud service provider. The solution to this problem is the standardization of services, which, to some extent, eliminates the fear that any service critical to the company will be unavailable for some time.

On the other hand, it is very important for the provider of cloud technologies to ensure an uninterrupted power supply to all its servers, since every minute of their use generates income. Unfortunately, public and private clouds are the subject of malicious attacks and disruptions to the infrastructure, such as power outages. Such incidents may affect the operation of domain name servers, make cloud servers open to third parties or directly disrupt their functioning. For example, *an attack on Akamai Technologies, that took place in 2004, caused problems with the resolution of domain names and a major failure affecting the Google Inc., Yahoo! Inc. and many other sites. In 2009, Google was the subject of DoS-attacks (denial-of-service) which put Google News and Gmail out of service for a few days* [15].

In cases where a server is subjected to malicious attacks or a force majeure causes a cloud service to fail, it is required to determine the time it takes the system to return to being fully functional after the crash. This will determine the level of technological and financial risks for both the owner and end users of cloud services.

According to [1, 2, 16–21] the operating process of cloud technologies is often described using the theory of queuing systems. The simplest approach to modelling the functioning of a computer server, which is proposed in [10, 14, 17], is based on the use of Markovian queuing systems of type $M/M/k$ (systems with Poisson input flow, exponential output flow, k channels of service). A more complex approach is the use of queueing systems of type $M/G/k$ (which assume Poisson arrivals, but model the output process using a generic distribution) and $G/G/k$. In our research, we will consider the first approach more precisely, which will allow us to explore the details of the functioning of any server in general. After such an investigation, it will be appropriate to increase the complexity of the model and explore queueing systems with other distributions of input and output flow.

Therefore, to solve the problem of determining the time to restore a fixed server to full functionality after a failure, we propose to apply Markovian queuing systems, namely a multi-channel queuing system with Poisson input flow and denial-of-service (breakdown).

2. Mathematical model of a server's performance

There are three models of cloud computing services:

- Software as a Service (SaaS). The end user is provided with software in the form of a provider's applications running on a cloud infrastructure.
- Platform as a Service (PaaS). The end user obtains cloud infrastructure tools to distribute applications created or purchased by the end user via the programming languages and tools supported by the provider.
- Infrastructure as a Service (IaaS). The end user obtains data processing, storage, networking and other basic computing resources, including operating systems and applications, where unspecified software can be deployed and run.

These models of cloud computing services are based on the operation of the main element of cloud technologies – a server. Investigating the mode of server operation (Fig. 1) and determining system characteristics will allow us at any time to calculate the required parameters of the reliability of a cloud service element, which is the main purpose of this article.

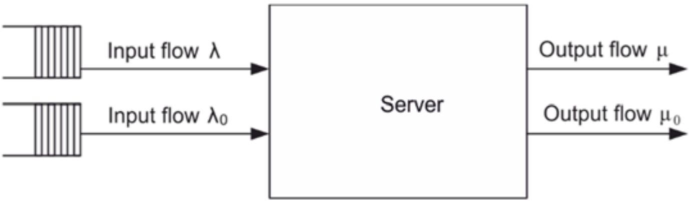

Fig. 1. Description of the server's operation

We consider the operation of a server functioning as a multiple queuing system. A k-channel queuing system receives a Poisson inflow of requests with an intensity λ. The service time is a random variable distributed according to an exponential law with parameter μ. Suppose that each of the service channels may fail and failures occur as a Poisson process with the parameter λ_0. The time spent on repairing the service channel is a random variable which has an exponential distribution with parameter μ_0. A description of the server's operation is shown in Fig. 1.

Let $P_{m(k)}(t)$ be the steady state probability that at time t, k requests are being serviced in the system, the channel is operational, and m requests are in the queue; $R_{m(k)}(t)$ is the probability that k requests are being serviced by the system, the channel is out of order, m requests are in the queue, and new requests continue to enter the system. The number of requests in the system is limited to $N = m + k$, $k = 0, 1, 2, 3, ..., k_1$, $m = 0, 1, 2, 3, ..., m_1$.

Let us define the number of channels to be k_1 and the maximum number of spaces in the queue to be m_1. Firstly, suppose the system is functioning. Considering how the system may leave a given state (negative expressions) and from what states the system may enter the considered state directly (positive terms), we obtain five equations defining the dynamics of a system of this form describing the following situations: 1) the system is empty, 2) not all of the channels are active, 3) all of the channels are active but there are no requests waiting, 4) there are requests waiting, but the system is not full, 5) the system is full. In a similar way, we can define analogous equations defining the dynamics when the system is not functioning. Thus, the stochastic model of the server's dynamics is presented by the following homogeneous system of linear differential equations:

$$P'_{0(0)}(t) = -(\lambda + \lambda_0)P_{0(0)}(t) + \mu P_{0(1)}(t) + \mu_0 R_{0(0)}(t)$$

$$\begin{aligned}\mathrm{P}'_{0(k)}(t) &= -(\lambda + \lambda_0 + k\mu)P_{0(k)}(t) + (k+1)\mu P_{0(k+1)}(t)\\ &\quad + k\mu_0 R_{0(k)}(t) + \lambda P_{0(k-1)}(t), \;\; 0 < k < k_1\end{aligned}$$

$$P'_{0(k_1)}(t) = -(\lambda + \lambda_0 + k_1\mu)P_{0(k_1)}(t) + k_1\mu P_{1(k_1)}(t) + k_1\mu_0 R_{0(k_1)}(t) + \lambda P_{0(k_1-1)}(t)$$

$$\begin{aligned}P'_{m(k_1)}(t) &= -(\lambda + \lambda_0 + k_1\mu)P_{m(k_1)}(t)\\ &\quad + k_1\mu P_{m+1(k_1)}(t) + k_1\mu_0 R_{m(k_1)}(t) + \lambda P_{m-1(k_1)}(t), \;\; 0 < m < m_1\end{aligned} \tag{1}$$

$$P'_{m_1(k_1)}(t) = -(\lambda_0 + k_1\mu)P_{m_1(k_1)}(t) + k_1\mu_0 R_{m(_1k_1)}(t) + \lambda P_{m_1-1(k_1)}(t)$$

$$R'_{0(0)}(t) = -(\lambda + \mu_0)R_{0(0)}(t) + \lambda_0 P_{0(0)}(t)$$

$$R'_{0(k)}(t) = -(\lambda + k\mu_0)R_{0(k)}(t) + \lambda_0 P_{0(k)}(t) + \lambda R_{0(k-1)}(t), \;\; 0 < k \le k_1$$

$$R'_{m(k_1)}(t) = -(\lambda + k_1\mu_0)R_{m(k_1)}(t) + \lambda_0 P_{m(k_1)}(t) + \lambda R_{m-1(k_1)}(t), \;\; 0 < m < m_1$$

$$R'_{m_1(k_1)}(t) = -k_1\mu_0 R_{m(_1k_1)}(t) + \lambda_0 P_{m_1(k_1)}(t) + \lambda R_{m_1-1(k_1)}(t)$$

The system (1) is a Markovian queuing system of type $M/M/k_1/m_1$ with Poisson input flow and denial of service. We can illustrate the server's performance using $N = m_1 = k_1 = 3$ as an example. In this case, the system (1) takes the following form:

$$
\begin{aligned}
&P'_{0(0)}(t) = -(\lambda+\lambda_0)P_{0(0)}(t)+\mu P_{0(1)}(t)+\mu_0 R_{0(0)}(t)\\
&P'_{0(1)}(t) = -(\lambda+\lambda_0+\mu)P_{0(1)}(t)+2\mu P_{0(2)}(t)+\mu_0 R_{0(1)}(t)+\lambda P_{0(0)}(t)\\
&P'_{0(2)}(t) = -(\lambda+\lambda_0+2\mu)P_{0(2)}(t)+3\mu P_{0(3)}(t)+2\mu_0 R_{0(2)}(t)+\lambda P_{0(1)}(t)\\
&P'_{0(3)}(t) = -(\lambda+\lambda_0+3\mu)P_{0(3)}(t)+3\mu P_{1(3)}(t)+3\mu_0 R_{0(3)}(t)+\lambda P_{0(2)}(t)\\
&P'_{1(3)}(t) = -(\lambda+\lambda_0+3\mu)P_{1(3)}(t)+3\mu P_{2(3)}(t)+3\mu_0 R_{1(3)}(t)+\lambda P_{0(3)}(t)\\
&P'_{2(3)}(t) = -(\lambda+\lambda_0+3\mu)P_{2(3)}(t)+3\mu P_{3(3)}(t)+3\mu_0 R_{2(3)}(t)+\lambda P_{1(3)}(t)\\
&P'_{3(3)}(t) = -(\lambda_0+3\mu)P_{3(3)}(t)+3\mu_0 R_{3(3)}(t)+\lambda P_{2(3)}(t)\\
&R'_{0(0)}(t) = -(\lambda+\mu_0)R_{0(0)}(t)+\lambda_0 P_{0(0)}(t)\\
&R'_{0(1)}(t) = -(\lambda+\mu_0)R_{0(1)}(t)+\lambda_0 P_{0(1)}(t)+\lambda R_{0(0)}(t)\\
&R'_{0(2)}(t) = -(\lambda+2\mu_0)R_{0(2)}(t)+\lambda_0 P_{0(2)}(t)+\lambda R_{0(1)}(t)\\
&R'_{0(3)}(t) = -(\lambda+3\mu_0)R_{0(3)}(t)+\lambda_0 P_{0(3)}(t)+\lambda R_{0(2)}(t)\\
&R'_{1(3)}(t) = -(\lambda+3\mu_0)R_{1(3)}(t)+\lambda_0 P_{1(3)}(t)+\lambda R_{0(3)}(t)\\
&R'_{2(3)}(t) = -(\lambda+3\mu_0)R_{2(3)}(t)+\lambda_0 P_{2(3)}(t)+\lambda R_{1(3)}(t)\\
&R'_{3(3)}(t) = -3\mu_0 R_{3(3)}(t)+\lambda_0 P_{3(3)}(t)+\lambda R_{2(3)}(t)
\end{aligned}
\tag{2}
$$

The stochastic model described by (1), which represents the server's operation in real time, can be presented in the following vector-matrix form:

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{AP}(t) \tag{3}$$

where **A** is a square matrix whose elements have constant values a_{ij} that define the system's parameters. The elements on the matrix's leading diagonal, linear combinations of $\lambda, \lambda_0, \mu, \mu_0$, are generally greater in absolute value, compared to the off-diagonal elements, and have a negative sign. The off-diagonal elements of the matrix a_{ij} can be a single element (one of $\lambda, \lambda_0, \mu, \mu_0$), or zero. The zero elements constitute the majority of each row of the matrix. We also note that the non-zero elements are arranged symmetrically with respect to the matrix's leading diagonal.

In this case, the condition of rationing states that:

$$\sum_{k=0}^{k_1}\sum_{m=0}^{m_1} P_{m(k)}(t) + \sum_{k=0}^{k_1}\sum_{m=0}^{m_1} R_{m(k)}(t) = 1 \tag{4}$$

since at any particular time the system must be in exactly one of the states described above. For a correctly built model, $\det(A) = 0$. Hence, it is obvious that satisfying the local balance conditions in all the states of the network is a sufficient condition for the existence of a steady state in the network.

3. Methodology

After performing an analysis of a number of studies [3–14] to address work on such dynamics, it is possible to conclude that the problem has not been studied in depth. It is obvious that most of the literature on queuing theory is dedicated to systems in steady mode operation. Therefore, we try to apply the solution method based on diagonalizing the matrix of the system's coefficients given in the set of equations (1). This method has not previously been applied to a queuing system.

The matrix form of the dynamics, given by equation (3), can be presented as follows. Let

$$\mathbf{P}(0) = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

be the vector describing the distribution of the initial state of the system. We write the vector **P**(t) as $\mathbf{T}\cdot\mathbf{y}(t)$, where

$$\mathbf{y}(t) = \begin{pmatrix} y_1(t) \\ y_{21}(t) \\ \vdots \\ y_n(t) \end{pmatrix}$$

T is matrix whose columns are the eigenvectors of the matrix **A**. Then,

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{T}\frac{d\mathbf{y}(t)}{dt} \to \mathbf{AP}(t) = \mathbf{T}\frac{d\mathbf{y}(t)}{dt} \to \mathbf{AT}\cdot\mathbf{y}(t) = \mathbf{T}\frac{d\mathbf{y}(t)}{dt}$$

$$\to \mathbf{T}^{-1}\mathbf{AT}\cdot\mathbf{y}(t) = \mathbf{T}^{-1}\mathbf{T}\frac{d\mathbf{y}(t)}{dt} \to \frac{d\mathbf{y}(t)}{dt} = \mathbf{T}^{-1}\mathbf{ATy}(t) = \mathbf{D}\cdot\mathbf{y}(t) \qquad (5)$$

Therefore,

$$\frac{d\mathbf{y}}{dt} = \mathbf{T}^{-1}\cdot\mathbf{A}\cdot\mathbf{T} = \mathbf{D}\cdot\mathbf{y}(t) \qquad (6)$$

The matrix **D** is diagonal and the elements on the leading diagonal are the eigenvalues β_i of the matrix **A**. Hence, **P**(t) = **T**·**y**(t) and thus we obtain **y**(t) = $\mathbf{T}^{-1}$·**P**(t).

Since $\mathbf{P}(0) = \mathbf{c}$, where

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

we obtain $\mathbf{y}(0) = \mathbf{T}^{-1}\cdot\mathbf{P}(0) = \mathbf{T}^{-1}\cdot\mathbf{c}$, where

$$\mathbf{y}(0) = \begin{pmatrix} y_1(0) \\ y_2(0) \\ \cdots \\ y_n(0) \end{pmatrix}, \quad \mathbf{T}^{-1}\cdot\mathbf{C} = \begin{pmatrix} c_1^* \\ c_2^* \\ \vdots \\ c_n^* \end{pmatrix}$$

Equation (6) thus becomes:

$$\begin{pmatrix} \dfrac{dy_1(t)}{dt} \\ \dfrac{dy_2(t)}{dt} \\ \vdots \\ \dfrac{dy_n(t)}{dt} \end{pmatrix} = \begin{pmatrix} \beta_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \beta_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \beta_3 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \beta_n \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ \dots \\ y_n(t) \end{pmatrix} \to \begin{pmatrix} \dfrac{dy_1(t)}{dt} \\ \dfrac{dy_2(t)}{dt} \\ \vdots \\ \dfrac{dy_n(t)}{dt} \end{pmatrix} = \begin{pmatrix} \beta_1 y_1(t) \\ \beta_2 y_2(t) \\ \beta_3 y_3(t) \\ \dots \\ \beta_n y_n(t) \end{pmatrix} \qquad (7)$$

Thus, the initial system of linear differential equations given by (3) has been transformed into the form (7), convenient for further analysis, by diagonalization of the matrix **A**. Hence, for the *i*-th equation of (7) we have

$$\frac{dy_1(t)}{dt} = \beta_i dt \to \int_0^t \frac{dy_i(t)}{y_i(t)} = \beta_i \int_0^t dt \to \ln y_i(t) \Big|_0^t = \beta_i t \Big|_0^t$$

$$\to \ln y_i(t) - \ln y_i(0) = \beta_i t \to \ln \frac{y_1(t)}{y_i(0)} = \beta_i t \to \frac{y_1(t)}{y_i(0)} = \mathrm{e}^{\beta_i t}$$

Since $y_i(0) = c_i^*$, the particular solution will be:

$$y_i(t) = c_i^* \mathrm{e}^{\beta_i t} \qquad (8)$$

Using (8), we obtain the general solution of the dynamic system (3) in real-time mode:

$$\begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ \dots \\ y_n(t) \end{pmatrix} = \begin{pmatrix} c_1^* \mathrm{e}^{\beta_1 t} \\ c_2^* \mathrm{e}^{\beta_2 t} \\ c_3^* \mathrm{e}^{\beta_3 t} \\ \dots \\ c_n^* \mathrm{e}^{\beta_n t} \end{pmatrix} \to \mathbf{P}(t) = \mathbf{T} \begin{pmatrix} c_1^* \mathrm{e}^{\beta_1 t} \\ c_2^* \mathrm{e}^{\beta_2 t} \\ c_3^* \mathrm{e}^{\beta_3 t} \\ \dots \\ c_n^* \mathrm{e}^{\beta_n t} \end{pmatrix} \qquad (9)$$

Thus, formula (9) represents the solution of the dynamic system (3) in real-time mode that have been found by the method of diagonalizing matrix **A**.

It should also be noted that, according to the well-known Lyapunov theorem [20, p. 19], the solution of the linear homogeneous system of differential equations with constant coefficients given by (1) will be stable in the case where the characteristic values of the system's matrix coefficients have the following form: $\beta_k = a_k + b_k i,$ where $a_k < 0.$

The steady state solution of (1) can be found using the method of substitution. To do this, from equation (1)

$$(\lambda + \lambda_0)P_{0(0)} = \mu P_{0(1)} + \mu_0 R_{0(0)} \tag{10}$$

using the normalization equation, we obtain:

$$P_{0(0)} = 1 - \sum_{k=1}^{k_1} \sum_{m=0}^{m_1} P_{m(k)} + \sum_{k=0}^{k_1} \sum_{m=0}^{m_1} R_{m(k)} \tag{11}$$

Substituting expression (11) into (10), we obtain:

$$(\lambda + \lambda_0) = (\lambda + \lambda_0)\left(\sum_{k=1}^{k_1} \sum_{m=0}^{m_1} P_{m(k)} + \sum_{k=0}^{k_1} \sum_{m=0}^{m_1} R_{m(k)} \right) + \mu P_{0(1)} + \mu_0 R_{0(0)} \tag{12}$$

Replacing the first equation in system (1) by (12), we obtain a system of the form

$$\mathbf{A}^{*}\mathbf{P} = \mathbf{B} \tag{13}$$

where **P** is the vector describing the stationary distribution of the states of the server (to be determined), $\mathbf{A}^*$ is the matrix of coefficients of these unknown probabilities; **B** is a vector of constants $\mathbf{B}^T = (\lambda + \lambda_0, 0, 0, 0, ..., 0)$.

From (14) we obtain the steady state solution of the system (1):

$$\mathbf{P} = (\mathbf{A}^{*})^{-1} \cdot \mathbf{B} \tag{14}$$

4. Numerical illustration

We first set the parameters of the server. As an example, consider the productivity of the green zone and red zone of SharePoint Server 2013 [21]. The relative load on the green zone server is less than 60%, while the relative load on server resources in the red zone is almost 100%. According to [21], the intensity of requests in the first stream of queries is 2 queries/s. The server declines 30% of these requests, i.e., 0.60 queries/s, the server response time (95th percentile) is 412 ms. Thus we obtain λ = 2 queries/s, $\lambda_0 = 0.60$ queries/s, $\mu = 1/0.412 = 2.43$ queries/s, $\mu_0 = 1/0.412 = 2.43$ queries/s.

We consider system (2), which assumes that the system can hold no more than 6 queries simultaneously and serve no more than 3, i.e., $N = m_1 + k_1 = 3 + 3 = 6$. The steady state solution of the system (2) is illustrated in Fig. 2.

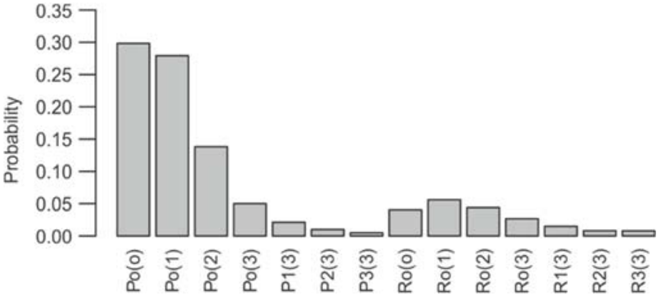

Fig. 2. The probability histogram of the stationary distribution of the server's state*.
Source: the authors' calculations for the system (2) based on (14) using *R*

Next, we describe the system's evolution from an initial state. First, we prove that the system tends to the steady state distribution. The real parts of the eigenvalues of the coefficient matrix, listed in Table 1, are all negative, which indicates that the system (2) always tends towards its unique stable point.

Table 1. The eigenvalues β_i of the system of homogeneous linear differential equations (2)

Eigenvalue	Value
β_1	–16.41 + 0*i*
β_2	–13.189 + 0*i*
β_3	–9.588 + 0*i*
β_4	–6.589 + 0.151*i*
β_5	–6.589 – 0.151*i*
β_6	–5.248 + 1.29*i*
β_7	–5.248 – 1.29*i*
β_8	–4.101 + 1.398*i*
β_9	–4.101 – 1.398*i*
β_{10}	–3.087 + 0.794*i*
β_{11}	–3.087 – 0.794*i*
β_{12}	–3.027 + 0*i*
β_{13}	–1.335 + 0*i*
β_{14}	0 + 0*i*

Source: the authors' calculations for (2) using *R*

Thus, by assuming that initially there are no requests in the system, i.e., $\mathbf{P}(0)^T = (1, 0, 0, 0, ..., 0)$, using (9) we obtain the solution to the system (2) for any given moment of time *t*. This solution is described in Table 2.

Table 2. The probabilities of the server being in a given state depending on the time *t*, s

State	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$	$t = 7$	$t = 8$	$t \to \infty$
$P_{0(0)}$	0.3627	0.31225	0.30197	0.29937	0.29868	0.2985	0.29845	0.29844	0.29844
$P_{0(1)}$	0.28811	0.2832	0.28038	0.27954	0.27932	0.27926	0.27924	0.27924	0.27924
$P_{0(2)}$	0.11674	0.13433	0.13725	0.13793	0.1381	0.13815	0.13816	0.13816	0.13817
$P_{0(3)}$	0.03221	0.04559	0.04889	0.04975	0.04997	0.05003	0.05005	0.05005	0.05006
$P_{1(3)}$	0.00874	0.01711	0.01999	0.02079	0.021	0.02106	0.02108	0.02108	0.02108
$P_{2(3)}$	0.00229	0.00686	0.00901	0.00965	0.00983	0.00987	0.00988	0.00989	0.00989
$P_{3(3)}$	0.00061	0.00291	0.00432	0.00477	0.00489	0.00492	0.00493	0.00493	0.00493
$R_{0(0)}$	0.05443	0.04326	0.04114	0.04063	0.04049	0.04046	0.04045	0.04045	0.04045
$R_{0(1)}$	0.06435	0.05874	0.05679	0.05629	0.05616	0.05613	0.05612	0.05612	0.05612
$R_{0(2)}$	0.04102	0.04488	0.04433	0.04414	0.04409	0.04408	0.04408	0.04408	0.04408
$R_{0(3)}$	0.01873	0.02611	0.02663	0.02668	0.02669	0.02669	0.02669	0.0267	0.0267
$R_{1(3)}$	0.00701	0.01356	0.01465	0.01485	0.0149	0.01491	0.01492	0.01492	0.01492
$R_{2(3)}$	0.00225	0.00653	0.00773	0.00799	0.00806	0.00807	0.00808	0.00808	0.00808
$R_{3(3)}$	0.00082	0.00467	0.00692	0.00762	0.00781	0.00786	0.00787	0.00788	0.00788

Source: the authors' calculations for the system (2) based on (9) using *R*.

As can be seen in Fig. 2 and Table 2, the probabilities of the states $P_{0(0)}(t)$ and $P_{0(1)}(t)$ are relatively large, while the other probabilities are close to zero and exhibit small, damped oscillations. The convergence of the probabilities $P_{0(0)}(t)$ and $P_{0(1)}(t)$ to their steady state values is shown in Fig. 3.

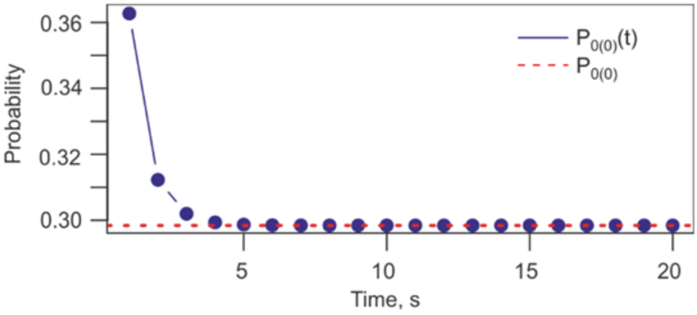

Fig. 3. Convergence of the probability $P_{0(0)}(t)$ to its steady state value.
Source: authors' calculations using *R*

We now change the distribution of the initial state to $\mathbf{P}(0)^T = (0.1\ 0.2\ 0\ 0\ 0\ 0.7\ 0\ 0\ 0\ ...\ 0)$, in order to study the effect of the initial state on the general solution given by (9) for the system (2). Obviously, the steady state solution of the system (2) is unchanged and, as in the previous case, 6 s the distribution of the states is very close to the stationary distribution. Figures 4, 5 show the dynamics of $P_{0(0)}(t)$, which is the most significant component of the vector $\mathbf{P}(t)$.

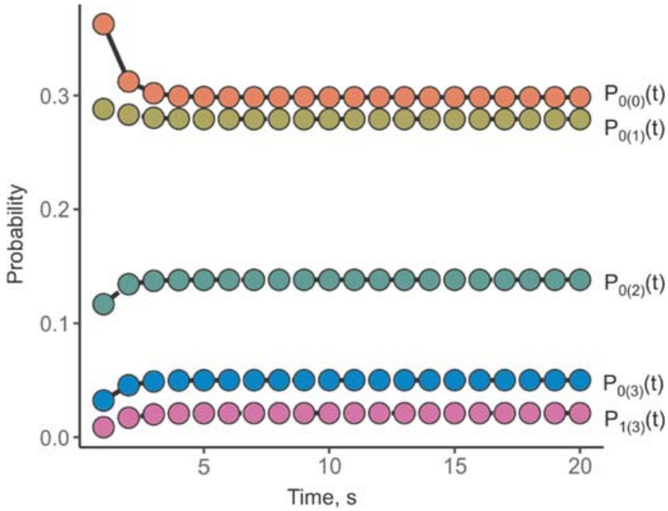

Fig. 4. Dynamics of the distribution of the states when the initial distribution is given by $\mathbf{P}(0)^T = (0.1\ 0.2\ 0\ 0\ 0\ 0.7\ 0\ 0\ 0 \ldots 0$.
Source: authors' calculations using *R*

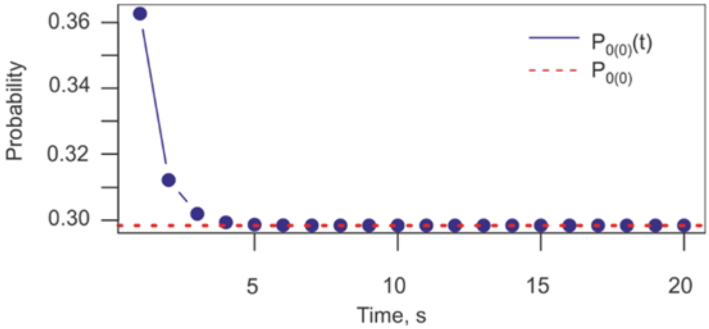

Fig. 5. Dynamics of the distribution of the states when the initial distribution is given by $\mathbf{P}(0)^T = (0.1\ 0.2\ 0\ 0\ 0\ 0.7\ 0\ 0\ 0 \ldots 0)$.
Source: authors' calculations using *R*

We now investigate how increasing the queuing capacity of the system (1) affects its functioning. It is assumed that the system can now hold 20 requests (3 being served and 17 in the queue), i.e. $N = m_1 + k_1 = 17 + 3 = 20$. This dramatically increases the number of equations describing the system up to 42. The effect of these changes on the dynamics of the system are illustrated in Figs. 5–7.

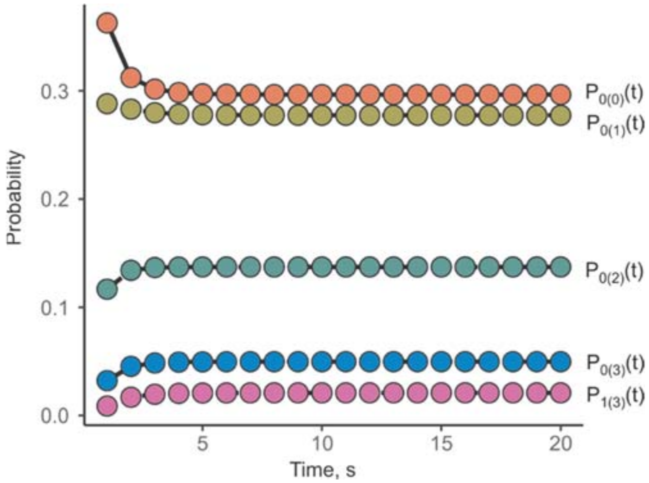

Fig. 6. Effect of increasing the queuing capacity of the system on its dynamics *N* = 20. Source: authors' calculations using *R*

From Figures 6, 7, it is obvious that an increase in the queuing capacity of the system given by (1) from 6 to 20 does not significantly affect the steady state functioning of the system (the probability of there being more than six requests in the system is very small). As in the previous cases, the system reaches its steady state distribution in ca. 8 s.

Recall that the results shown in Figs. 2–7 and Tables 1, 2 are obtained based on the fact that the input parameters are set based on the green state of the servers. Thus the relative load on the servers is 67%.

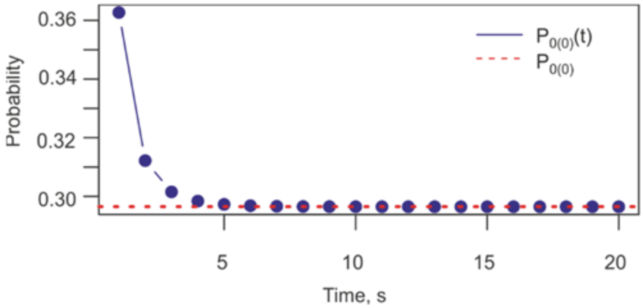

Fig. 7. Dynamics of $P_{0(0)}(t)$ after increasing the capacity of the queue *N* = 20. Source: authors' calculations using *R*

We now change the server's settings to the red state. According to [3], the intensity of admissions is 3 queries/s. It is assumed that the server declines 30% of these queries, i.e., 0.90 queries/s and the server response time per query is (95th percentile) 635 ms.

Thus $\lambda = 3$ queries/s, $\lambda_0 = 0.90$ queries/s, $\mu = 1/0.635 = 1.57$ queries/s, μ_0 1/0.635 = 1.57 queries/s.

Equations (9) and (14) are used to determine the dynamics of the system (2). A histogram of the steady state probabilities of the server's performance is shown in Fig. 8. As we can see, $P_{0(0)} = 0.0437$. Therefore, according to [13], utilization – fraction of time the server is busy equals to $\rho = 1 - P_{0(0)}$, or the utilization is 100 – 4.37 = 95.63% of system capacity. Moreover, the greatest changes occur in the probabilities $P_{0(0)}$ and $R_{3(3)}$. We need to take into consideration how to change the server's output time in the steady state mode with the output parameters corresponding to the red state. Figures 9, 10 indicate that the system reaches steady state mode in ca. 10 s.

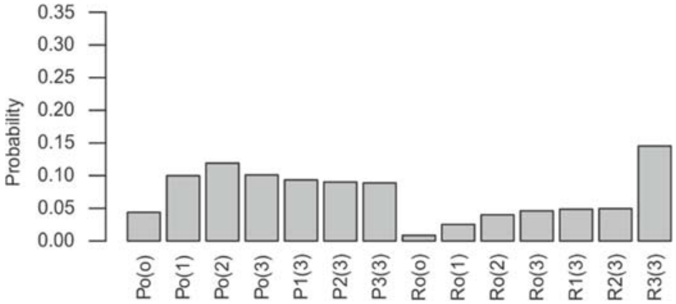

Fig. 8. The steady state distribution of the state of the system in the red state.
Source: the authors' calculations using *R* for the system (2) based on (14)

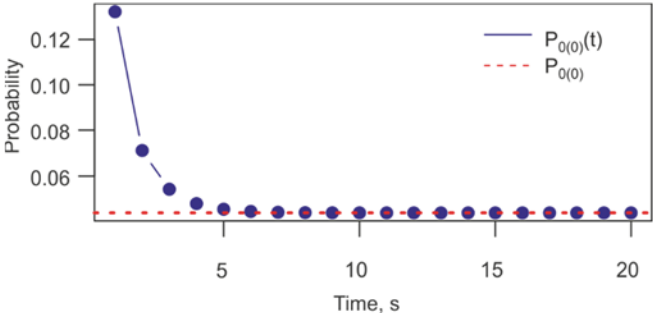

Fig. 9. Dynamics of $P_{0(0)}(t)$ in the red state.
Source: authors' calculations using *R*

After careful analysis of the results of the simulations, it can be argued that the system evolves relatively quickly to essentially the same steady state regardless of the initial distribution, and the queue capacity (for $m_1 \geq 3$). The relative load on the system in the steady state is only really affected by the service time (given a fixed arrival rate).

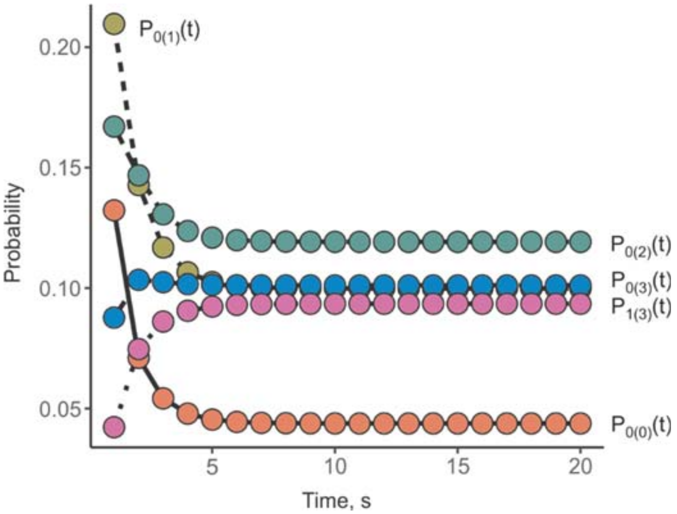

Fig. 10. Dynamics of the distribution of the state of the system in the red zone.
Source: authors' calculations using *R*

5. Conclusions and follow-up research

In the modern world, the use of cloud computing is a part of everyday life, covering all the vital spheres of business and related infrastructure, the functioning of the state and e-government. It is becoming an activity of daily life for ordinary people in every country of the world.

The model of a server subject to breakdown presented above gives us the opportunity to determine the time required to return to the steady state mode of operation after the system fails. This, in turn, will help the owner of the cloud server to calculate the risks of financial losses due to technological failures. One subject for follow-up research is to construct mathematical models to describe the entire infrastructure of a cloud service. Besides this, it would be logical to consider various types of distributions describing the input and output flows of the queuing system. Therefore, the authors plan to use *M*/*G*/*k* and then *G*/*G*/*k* systems to create mathematical models to analyse the entire infrastructure of cloud services.

References

[1] Altman E., Avrachenkov K., Barakat C., *A stochastic model of TCP/IP with stationary random losses*, Proc. ACM SIGCOMM Computer Communication Review, 2000, 30 (4), 231–242.

[2] Baccelli F., Hong D., *AIMD, fairness and fractal scaling of TCP traffic*, Proc. IEEE INFOCOM, New York 2000, 1, 229–238.

[3] BINI D., LATOUCHE G., MEINI B., *Numerical Methods for Structured Markov Chains*, Oxford University Press, New York 2005.

[4] BRANDT A., BRANDT M., *On the M(n)/M(n)/s queue with impatient calls*, Perf. Eval., 1999, 35 (1–2), 1–18.

[5] BRUNEEL H., KIM B., *Discrete-Time Models for Communication Systems Including ATM*, Kluwer Academic, Boston 1993.

[6] BUCHHOLZ P., *A class of hierarchical queueing networks and their analysis*, Queuing Syst., 1994, 15 (1–4), 59–80.

[7] BUCKLEW J., *Large Deviation Techniques in Decision, Simulation and Estimation*, Wiley, New York 1990.

[8] BUNKE H., CAELLI T., *Hidden Markov models. Applications in computer vision*, World Scientific, Singapore 2001, 244.

[9] BUZACOTT J., SHANTHIKUMAR J., *Stochastic Models of Manufacturing Systems*, Prentice-Hall, New Jersey 1993.

[10] CHANG C., *Performance Guarantees in Communication Networks*, Springer-Verlag, London 2000.

[11] CHEN H., YAO D., *Fundamentals of Queueing Networks. Performance, Asymptotics and Optimization*, Springer-Verlag, New York 2001.

[12] GNEDENKO B., KOVALENKO I., *Introduction to Queueing Theory*, Birkhauser Boston, Inc., Cambridge 1968.

[13] KLEINROCK L., *Queueing Systems*. Vol. 1. *Computer Systems Modelling Fundamentals*, 2nd Ed., Wiley, 2009, 576.

[14] LE BOUDEC J., THIRAN P., *Network Calculus.: A Theory of Deterministic Queueing Systems for the Internet*, Springer-Verlag, Berlin 2001.

[15] MARINESCU D., *Cloud Computing: Cloud vulnerabilities*, TechNet Magazine, July 2013. Available at: https://technet.microsoft.com/en-au/library/dn271884.aspx

[16] RIORDAN J., *Stochastic Service Systems*, Wiley, New York 1962.

[17] SMITH J., TAN B., *Handbook of Stochastic Models and Analysis of Manufacturing System Operations*, Springer-Verlag, New York 2013, 373.

[18] SRIDHAR T., *Cloud Computing – a primer. Part 1. Models and Technologies*, Int. Protocol J., 2009, 12 (3). Available at: https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-45/123-cloud1.html

[19] SRIDHAR T., *Cloud Computing – a primer. Part 2. Infrastructure and Implementation Topics*, Int. Protocol J., 2009, 12 (3). Available at: ttps://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-46/124-cloud2.html

[20] SRIKANT R., YING L., *Communication Networks. An Optimization, Control and Stochastic Networks Perspective*, Cambridge University Press, Cambridge 2013, 363.

[21] *Estimate capacity and performance for Web Content Management (SharePoint Server 2013)*, TechNet Magazine, December 2016. Available at: https://technet.microsoft.com/en-gb/library/gg398060.aspx

[22] *A language and environment for statistical computing*, R Core Team R, R Foundation for Statistical Computing, Vienna 2015. Available at: https://www.R-project.org/

Received 2 February 2017
Accepted 16 October 2017