The Analysis of Cloud Computing System as a Queueing System with Several Servers and a Single Buffer

Ivan Zaryadov[1,2(✉)], Andrey Kradenyh[1], and Anastasiya Gorbunova[1]

[1] Department of Applied Probability and Informatics,
Peoples Friendship University of Russia (RUDN University),
6 Miklukho-Maklaya Str., Moscow 117198, Russia
{zaryadov_is,kradenyh_aa,gorbunova_av}@rudn.university
[2] Institute of Informatics Problems of the Federal Research Center
"Computer Science and Control" of the Russian Academy of Sciences,
44-2 Vavilova Str., Moscow 119333, Russia

Abstract. The mathematical model of cloud computing system based on the queuing system with the splitting of the incoming queries and synchronization of services is considered. The queuing system consists of a single buffer and N servers ($N > 2$), service times are independent and exponentially distributed. The incoming query enters the system as a whole and only before service is divided into subqueries, each subquery is served by its device. The servers with parts of the same query are considered to be employed as long as the query is not serviced as a whole: the query is handled only when the last of it is out and a new query may be served only when there are enough free servers (the response time is the maximum of service times of all parts of this query). Expressions for the stationary performance characteristics of the system are presented.

Keywords: Cloud computing system · Splitting of incoming queries
Queueing system · Response time
Stationary probability-time characteristics · Inhomogeneous servers
Homogeneous servers

1 Introduction

This paper is devoted to the problem of cloud computing modeling [1]. There exist several approaches to the cloud computing systems modeling. One approach (see [3–6]) implies that the cloud computing system is modeled via a queuing system with K subqueues and each subqueue consists of a buffer with one or several servers. The incoming query is divided into exactly K subqueries, one for each of the subqueues. On this approach the Fork–Join [3,4] and Split–Merge [7–9] models are based. The Split–Merge model uses the idea of synchronization of servers (only when the service of all subqueries belonging to the same query has been finished, the service of the new query in commenced). The second

V. V. Rykov et al. (Eds.): ACMPT 2017, LNCS 10684, pp. 11–22, 2017.
https://doi.org/10.1007/978-3-319-71504-9_2

approach models the cloud computing system as queuing system with unlimited number of homogeneous servers (see [10,11]), in which the incoming query is split into several subqueries and each is served by one of the free servers.

The mathematical model of cloud computing system presented here may be considered as general case within the second approach: the queuing system consists of N servers and the buffer; each incoming query is split only before the start of the service. We use the idea of synchronization of services as in [7–9]. Our goal is to derive analytical expressions for the main performance characteristics of the model.

The paper is structured as follows. The Introduction section is followed by the section with general system (inhomogeneous servers) description. In the next section some results for the case of the considered system (homogeneous servers) are presented. In Conclusion the directions for further research are given.

2 The General Case of Inhomogeneous Servers

2.1 The System Description

The queuing system consists of N non-homogeneous servers ($N > 2$) labeled with numbers from 1 to N without repetitions and the buffer of size $r \leq \infty$. Queries enter the system according to Poisson flow with rate λ. Before the start of the service the query is divided into N subqueries, the service time of a subquery on server i has exponential distribution with rate μ_i, $i = \overline{1,N}$. The mechanism of synchronization is used — the servers with parts of the same query are considered to be busy as long as the query is not serviced as a whole: the query is handled only when the last part of it is out and a new query may be served only when all servers are free.

Denote the response time of a query by η. It is one of the main characteristics of cloud computing systems (see [2]). It may be defined as $\eta = \max(\eta_1, \ldots, \eta_N)$ (see [3–5]) or as $\eta = \min(\eta_1, \ldots, \eta_N)$ (see [12–15]), where η_i are the service times of the subqueries.

In [16] it is shown that the analysis of queuing models with the response time defined as minimum is equal to the analysis of well-studied multiserver queuing systems [17]. So we will consider only the case of maximum.

The probability distribution function (PDF) of $\eta = \max(\eta_1, \ldots, \eta_N)$ has the form [6,16,18]:

$$P\{\max(\eta_1, \ldots, \eta_N) < x\} = \prod_{i=1}^{N} \left(1 - e^{-\mu_i x}\right). \tag{1}$$

For homogeneous servers ($\mu_i = \mu$, $\forall i = \overline{1,N}$) (1) is reduced to

$$P\{\max(\eta_1, \eta_2, \ldots, \eta_N) < x\} = \left(1 - e^{-\mu x}\right)^N. \tag{2}$$

We will consider the random process $\nu(t)$ defined by

$$\nu(t) = \{\xi(t), \boldsymbol{\delta}(t)\}, \tag{3}$$

where $\xi(t)$ is the number of queries in the buffer at time t and the vector $\boldsymbol{\delta}(t) = (\delta_1(t), \ldots, \delta_N(t))$ describes the servers occupancy ($\delta_i(t) = 1$ if the i-th server is occupied by the i-th part of a query and $\delta_i(t) = 0$ otherwise). It is supposed, that each subquery may enter only its server so the situation when two or more subqueries are directed to the same server is impossible.

The state space $\mathcal{X}$ of $\{\nu(t), t \geq 0\}$ is

$$\mathcal{X} = \{(0)\} \cup \{(I, (\delta_1, \ldots, \delta_N))\}, \tag{4}$$

where $I = \overline{0; r}$, $\delta_1, \ldots, \delta_N$ take values 0 or 1. Denote

$$P\{\xi(t) = I, \boldsymbol{\delta}(t) = (\delta_1, \ldots, \delta_N)\} = p_{I,\boldsymbol{\delta}}(t), \quad P\{\nu(t) = 0\} = p_0(t).$$

Assuming that the steady-state exists the stationary probabilities are henceforth denoted by $p_{I,\boldsymbol{\delta}}$ and p_0.

2.2 The System of Equations

In order to derive the system of equilibrium equations for the considered system the following notation is needed:

- $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)$ — the service rate row-vector (size N);
- $\boldsymbol{\mu}_{-(i_1, i_2, \ldots, i_k)}$ — the row-vector of service intensities (size $N - k$) from which the elements with the specified numbers $(\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_k})$ have been deleted $(i_1 \neq i_2 \neq \ldots \neq i_k,\ i_1, i_2, \ldots, i_k = \overline{1, N})$;
- $\Lambda = diag(\lambda)$ — the diagonal arrival rate matrix (of variable size which is defined by the corresponding steady-state probability vector).
- $\boldsymbol{p}_{I,\boldsymbol{k}}$, $I = \overline{0; r}$, $k = \overline{1, N}$, — the vector of steady-state probabilities that there are I queries in the buffer and k occupied servers;
- $\boldsymbol{p}_{I,\boldsymbol{k}_{i_1, i_2, \ldots, i_{k-1}}}$, $I = \overline{0; r}$, $k = \overline{1, N}$, $i_1 \neq i_2 \neq \ldots \neq i_{k-1}$, $i_1, i_2, \ldots, i_{k-1} = \overline{1, N}$, — the vector of steady-state probabilities that there are I queries in the buffer, k servers are occupied and the severs with numbers $i_1, i_2, \ldots, i_{k-1}$ are necessarily under service.

For the probability of the empty system we have the following equation:

$$\lambda p_0 = \mu_1 p_{0,(1,0,\ldots,0)} + \mu_2 p_{0,(0,1,\ldots,0)} + \ldots + \mu_N p_{0,(0,0,\ldots,1)}. \tag{5}$$

We leave the state (0) when the first query enters the system, and we enter this state when the last part of the previous query will finish its service. In matrix form the Eq. (5):

$$\lambda p_0 = \tilde{M}_1 \boldsymbol{p}_{0,\mathbf{1}}, \tag{6}$$

where $\tilde{M}_1 = \boldsymbol{\mu}$.

Now we will consider the set $\{(0, \mathbf{1})\}$ of states $\{(0, (1, 0, \ldots, 0)), (0, (0, 1, 0, \ldots, 0)), \ldots, (0, (0, 0, \ldots, 0, 1))\}$ — the buffer is empty and one server is occupied:

$$\begin{cases} (\lambda+\mu_1)p_{0,(1,0,\ldots,0,0)} = \mu_2 p_{0,(1,1,0,\ldots,0,0)} + \mu_3 p_{0,(1,0,1,\ldots,0,0)} + \ldots + \\ \qquad\qquad + \mu_N p_{0,(1,0,0,\ldots,0,1)}, \\ (\lambda+\mu_2)p_{0,(0,1,\ldots,0)} = \mu_1 p_{0,(1,1,0,\ldots,0)} + \mu_3 p_{0,(0,1,1,\ldots,0,0)} + \ldots + \\ \qquad\qquad + \mu_N p_{0,(0,1,0,\ldots,0,1)}, \\ \qquad\qquad \vdots \\ (\lambda+\mu_N)p_{0,(0,0,\ldots,0,1)} = \mu_1 p_{0,(1,0,0,\ldots,0,1)} + \mu_2 p_{0,(0,1,0,\ldots,0,1)} + \ldots + \\ \qquad\qquad + \mu_{N-1} p_{0,(0,0,0,\ldots,1,1)}, \end{cases}$$

or according to our notation:

$$\begin{cases} (\lambda+\mu_1)p_{0,(1,0,\ldots,0)} = \boldsymbol{\mu}_{-(1)}\boldsymbol{p}_{0,\mathbf{2}_1}, \\ (\lambda+\mu_2)p_{0,(0,1,\ldots,0)} = \boldsymbol{\mu}_{-(2)}\boldsymbol{p}_{0,\mathbf{2}_2}, \\ \qquad\qquad \vdots \\ (\lambda+\mu_N)p_{0,(0,0,\ldots,1)} = \boldsymbol{\mu}_{-(N)}\boldsymbol{p}_{0,\mathbf{2}_N}. \end{cases} \tag{7}$$

Relations (7) is the matrix form can be written as

$$\left(\tilde{\Lambda}_1 + M_1\right)\boldsymbol{p}_{0,\mathbf{1}} = \tilde{M}_2\boldsymbol{p}_{0,\mathbf{2}}. \tag{8}$$

Here $\tilde{\Lambda}_1 = diag(\lambda)$ and $M_1 = diag(\mu_i)_{i=\overline{1,N}}$ — the diagonal matrices with the same size as the vector $\boldsymbol{p}_{0,\mathbf{1}}$, $\tilde{M}_2 = diag(\boldsymbol{\mu}_i)_{i=\overline{1,N}}$ — the diagonal matrix with the same size as the vector $\boldsymbol{p}_{0,\mathbf{2}}$, column-vector $\boldsymbol{p}_{0,\mathbf{2}} = \left(\boldsymbol{p}_{0,\mathbf{2}_1}, \ldots, \boldsymbol{p}_{0,\mathbf{2}_N}\right)$.

For the set $\{(0,\mathbf{2})\}$ of states, when the buffer is empty and two servers are occupied (C_N^2 states), the following system may be presented:

$$\begin{cases} (\Lambda + M_{2_1})\boldsymbol{p}_{0,\mathbf{2}_1} = M_{-(1,\cdot)}\boldsymbol{p}_{0,\mathbf{3}_{1,\cdot}}, \\ (\Lambda + M_{2_2})\boldsymbol{p}_{0,\mathbf{2}_2} = M_{-(2,\cdot)}\boldsymbol{p}_{0,\mathbf{3}_{2,\cdot}}, \\ \qquad\qquad \vdots \\ (\Lambda + M_{2_N})\boldsymbol{p}_{0,\mathbf{2}_N} = M_{-(N,\cdot)}\boldsymbol{p}_{0,\mathbf{3}_{N,\cdot}}, \end{cases}, \tag{9}$$

where $\boldsymbol{p}_{0,\mathbf{3}_{k,\cdot}}$ is a column-vector with elements $\boldsymbol{p}_{0,\mathbf{3}_{k,i}}$, $i \neq k$, $k, i = \overline{1,N}$, matrices Λ and $M_{2_k} = diag(\mu_k + mu_i)_{i=\overline{1,N}, i\neq k}$ are $(N-1)$-by-$(N-1)$ diagonal matrices, $M_{-(k,\cdot)} = diag(\boldsymbol{\mu}_{-(k,i)})_{i=\overline{1,N}, i\neq k}$, $k = \overline{1,N}$.

The matrix form of (9) is:

$$\left(\tilde{\Lambda}_2 + M_2\right)\boldsymbol{p}_{0,\mathbf{2}} = \tilde{M}_3\boldsymbol{p}_{0,\mathbf{3}}, \tag{10}$$

where

$$\tilde{\Lambda}_2 = diag(\Lambda), \quad M_2 = diag(M_{2_i})_{i=\overline{1,N}},$$

$$\tilde{M}_3 = diag(M_{-(i,\cdot)})_{i=\overline{1,N}},$$

and a column-vector

$$\boldsymbol{p}_{0,\mathbf{3}} = \left(\boldsymbol{p}_{0,\mathbf{3}_{1,\cdot}}, \ldots, \boldsymbol{p}_{0,\mathbf{3}_{N,\cdot}}\right).$$

With the set $\{(0, \boldsymbol{k})\}$ of states, when the buffer is empty and k servers are occupied (C_N^k states) the following system of equations is connected:

$$\left(\Lambda + M_{k_{i_1,i_2,\ldots,i_{k-1}}}\right) \boldsymbol{p}_{0,\boldsymbol{k}_{i_1,i_2,\ldots,i_{k-1}}} = M_{-(i_1,i_2,\ldots,i_{k-1},\cdot)} \boldsymbol{p}_{0,\boldsymbol{k}+\mathbf{1}_{i_1,i_2,\ldots,i_{k-1},\cdot}}, \quad k = \overline{3, N-1}, \tag{11}$$

here

$$M_{k_{i_1,i_2,\ldots,i_{k-1}}} = diag(\mu_{i_1} + \ldots + \mu_{i_{k-1}} + \mu_i)_{i_1,\ldots,i_{k-1},i=\overline{1,N},i_1\neq\ldots\neq i_{k-1}\neq i},$$

$$M_{-(i_1,i_2,\ldots,i_{k-1},\cdot)} = diag(\boldsymbol{\mu}_{i_1,i_2,\ldots,i_{k-1},i})_{i_1,\ldots,i_{k-1},i=\overline{1,N},i_1\neq\ldots\neq i_{k-1}\neq i},$$

and column-vector

$$\boldsymbol{p}_{0,\boldsymbol{k}+\mathbf{1}_{i_1,i_2,\ldots,i_{k-1},\cdot}} = \left(\boldsymbol{p}_{0,\boldsymbol{k}+\mathbf{1}_{i_1,i_2,\ldots,i_{k-1},\cdot}}\right)_{i_1,\ldots,i_{k-1},i=\overline{1,N},i_1\neq\ldots\neq i_{k-1}\neq i}.$$

The matrix form of (11) is

$$\left(\tilde{\Lambda}_k + M_k\right)\boldsymbol{p}_{0,\boldsymbol{k}} = \tilde{M}_{k+1}\boldsymbol{p}_{0,\boldsymbol{k}+\mathbf{1}}, \tag{12}$$

where

$$\tilde{\Lambda}_k = diag(\Lambda),$$

$$M_k = diag(M_{k_{i_1,i_2,\ldots,i_{k-1}}})_{i_1,\ldots,i_{k-1}=\overline{1,N},i_1\neq\ldots\neq i_{k-1}},$$

$$\tilde{M}_{k+1} = diag(M_{-(i_1,i_2,\ldots,i_{k-1},\cdot)})_{i_1,\ldots,i_{k-1}=\overline{1,N},i_1\neq\ldots\neq i_{k-1}},$$

and a column-vector

$$\boldsymbol{p}_{0,\boldsymbol{k}+\mathbf{1}} = \left(\boldsymbol{p}_{0,\boldsymbol{k}+\mathbf{1}_{i_1,i_2,\ldots,i_{k-1},\cdot}}\right), i_1,\ldots,i_{k-1} = \overline{1,N}, i_1 \neq \ldots \neq i_{k-1}.$$

Now we will consider the case, when the buffer is empty and all servers are occupied — $\{(0, \boldsymbol{N})\} = \{(0, (1, 1, \ldots, 1))\}$:

$$(\lambda + \mu_1 + \mu_2 + \ldots + \mu_N)\, p_{0,(1,1,1,\ldots,1,1)} = \lambda p_0 + \mu_1 p_{1,(1,0,0,\ldots,0,0)} + \\ + \mu_2 p_{1,(0,1,0,\ldots,0,0)} + \mu_N p_{1,(0,0,0,\ldots,0,1)}, \tag{13}$$

and the matrix form of (13)

$$\left(\tilde{\Lambda}_N + M_N\right)\boldsymbol{p}_{0,\boldsymbol{N}} = \lambda p_0 + \tilde{M}_1 \boldsymbol{p}_{1,\mathbf{1}}, \tag{14}$$

with

$$\tilde{\Lambda}_N = \lambda, \quad M_N = \mu_1 + \mu_2 + \ldots + \mu_N,$$

and the matrix $\tilde{M}_1$ was defined in (6).

For the case, when the buffer is not empty, we will not derive the systems of equations in detail, but immediately represent them in the matrix form:

$$\left(\tilde{\Lambda}_k + M_k\right) \boldsymbol{p}_{I,\boldsymbol{k}} = \lambda \boldsymbol{p}_{I-1,\boldsymbol{k}} + \tilde{M}_{k+1} \boldsymbol{p}_{I,\boldsymbol{k+1}}, \quad I \geq 1, \quad 1 \leq k \leq N-1, \tag{15}$$

and for $k = N$

$$\left(\tilde{\Lambda}_N + M_N\right) \boldsymbol{p}_{I,\boldsymbol{N}} = \lambda \boldsymbol{p}_{I-1,\boldsymbol{N}} + \tilde{M}_1 \boldsymbol{p}_{I+1,\boldsymbol{1}}, \quad I \geq 1, \tag{16}$$

matrices $\tilde{\Lambda}_k$, M_k and $\tilde{M}_k$, $k = \overline{1,N}$, are defined in (6), (8), (10), (12), (14). The vectors $\boldsymbol{p}_{I,\boldsymbol{k}}$, $I \geq 1$, $1 \leq k \leq N$ have the same structure and size as vectors $\boldsymbol{p}_{0,\boldsymbol{k}}$, $1 \leq k \leq N$, defined in (6), (8), (10), (12), (14).

The normalization condition for the system with unlimited buffer size is

$$p_0 + \sum_{I=0}^{\infty} \sum_{k=1}^{N} \mathbf{1}_k \boldsymbol{p}_{I,\boldsymbol{k}} = 1, \tag{17}$$

where $\mathbf{1}_k = (1, 1, \ldots, 1, 1)$ is a vector which size is equal to the size of the vector $\boldsymbol{p}_{I,\boldsymbol{k}}$, $1 \leq k \leq N$.

For the system with finite-capacity buffer the following equations hold:

$$M_k \boldsymbol{p}_{r,\boldsymbol{k}} = \lambda \boldsymbol{p}_{r-1,\boldsymbol{k}} + \tilde{M}_{k+1} \boldsymbol{p}_{r,\boldsymbol{k+1}}, \quad k = \overline{1,N-1}, \tag{18}$$

and

$$M_N \boldsymbol{p}_{r,\boldsymbol{N}} = \lambda \boldsymbol{p}_{r-1,\boldsymbol{N}}, \tag{19}$$

where matrices M_k and $\tilde{M}_k$, $1 \leq k \leq N$, are defined in (6), (8), (10), (12), (14) and vectors $\boldsymbol{p}_{r,\boldsymbol{k}}$, $1 \leq k \leq N$, have the same structure and the same size as vectors $\boldsymbol{p}_{0,\boldsymbol{k}}$, $1 \leq k \leq N$, defined in (6), (8), (10), (12), (14).

The normalization condition (17) for the finite buffer system takes the form:

$$p_0 + \sum_{I=0}^{r} \sum_{k=1}^{N} \mathbf{1}_k \boldsymbol{p}_{I,\boldsymbol{k}} = 1. \tag{20}$$

For the infinite system (6), (8), (10), (12), (14), (15), (16), as well for the finite system (6), (8), (10), (12), (14), (15), (16), (18), (19) the solution may be found by using matrix-analytical methods [19–26].

2.3 Marginal Probability Distributions

By $\tilde{p}_{I,k} = \mathbf{1}\boldsymbol{p}_{I,\boldsymbol{k}}$, $I \geq 0$, $k = \overline{1,N}$ we will denote the probability that there are I, $I \geq 0$, queries in the buffer and k, $k = \overline{1,N}$, servers are occupied. Then the system (6), (8), (10), (12), (14), (15), (16) takes form:

$$\begin{cases} \lambda p_0 = \tilde{M}_1 \boldsymbol{p}_{0,\boldsymbol{1}}, \\ \lambda \sum\limits_{k=1}^{N} \tilde{p}_{I,k} = \tilde{M}_1 \boldsymbol{p}_{I+1,\boldsymbol{1}}, \quad I \geq 0. \end{cases} \tag{21}$$

The steady-state probability $\tilde{\pi}_I$, that there are I, $I \geq 0$, queries in the buffer and at least one server is busy, is

$$\tilde{\pi}_I = \sum_{k=1}^{N} \tilde{p}_{I,k}, \quad I \geq 0, \tag{22}$$

and from (21) with (22) the following relations are obtained:

$$\tilde{\pi}_I = \frac{1}{\lambda} \tilde{M}_1 \boldsymbol{p}_{I+1,\mathbf{1}}, \quad I \geq 0. \tag{23}$$

The steady-state probabilities π_k that there are k, $k = \overline{0,N}$, servers are occupied regardless of the number of queries in the buffer are follows:

$$\begin{cases} \pi_0 = p_0, \\ \pi_k = \sum\limits_{I=0}^{\infty} \mathbf{1} \boldsymbol{p}_{I,\boldsymbol{k}}, \quad k = \overline{1,K}. \end{cases} \tag{24}$$

We will use the probability distributions (23) and (24) for the system with homogeneous servers presented in the next section.

By substituting $\min \boldsymbol{\mu}$ and $\max \boldsymbol{\mu}$ instead of μ_j, $j = \overline{1,N}$ in (21) and making the same computations as for the system with homogeneous server (which are presented in the next section) one may obtain the following inequality for the probability p_0:

$$1 - \frac{\lambda}{\min(\mu_1, \ldots, \mu_N)} \sum_{j=1}^{N} \frac{1}{jC_N^j} \leq p_0 \leq 1 - \frac{\lambda}{\max(\mu_1, \ldots, \mu_N)} \sum_{j=1}^{N} \frac{1}{jC_N^j}. \tag{25}$$

3 The Case of Homogeneous Servers. Stationary Performance Characteristics

3.1 The System Description and the System of Equations

Let's assume that all servers are homogeneous ($\mu_1 = \mu_2 = \ldots = \mu_N = \mu$), then we may redefine the random process $\{\nu(t), tge0\}$ (3) as $\nu(t) = \{\xi(t), \delta(t)\}$, where $\delta(t)$ — not the vector, but the scalar – the number of occupied servers. The set of states $\mathcal{X}$ (4) can be redefined as $\mathcal{X} = \{(0)\} \cup \{(i;j)\}$, where $i = \overline{0,r}$ ($r \leq \infty$) is the number of queries in the buffer and $j = \overline{1,N}$ is the number of occupied servers.

The steady-state probabilities p_0 (the system is empty) and $p_{i,j}$, $i = \overline{0,r}$, $r \leq \infty$, $j = \overline{1,N}$ (i queries in the buffer and j servers are occupied by subqueries) satisfy the following systems of equations (when $r = \infty$):

$$\begin{cases} \lambda p_0 = f_1(\mu) p_{0,1}, \\ (\lambda + f_j(\mu))\, p_{0,j} = f_{j+1}(\mu) p_{0,j+1}, \quad j = \overline{1,N-1}, \\ (\lambda + f_N(\mu))\, p_{0,N} = \lambda p_0 + f_1(\mu) p_{1,1}, \\ (\lambda + f_j(\mu))\, p_{i,j} = \lambda p_{i-1,j} + f_{j+1}(\mu) p_{i,j+1}, \quad i \geq 1, j = \overline{1,N-1}, \\ (\lambda + f_N(\mu))\, p_{i,N} = \lambda p_{i-1,N} + f_1(\mu) p_{i+1,N}, \end{cases} \tag{26}$$

with

$$f_j(\mu) = jC_N^j\mu, \quad j = \overline{1,N} \tag{27}$$

for the system with the unlimited-capacity buffer.

The normalization condition is:

$$p_0 + \sum_{i=0}^{\infty}\sum_{j=1}^{N} p_{i,j} = 1. \tag{28}$$

For the system with the finite-capacity buffer (26) takes form:

$$\begin{cases} \lambda p_0 = f_1(\mu)p_{0,1}, \\ (\lambda + f_j(\mu))\, p_{0,j} = f_{j+1}(\mu)p_{0,j+1}, \quad j = \overline{1,N-1}, \\ (\lambda + f_N(\mu))\, p_{0,N} = \lambda p_0 + f_1(\mu)p_{1,1}, \\ (\lambda + f_j(\mu))\, p_{i,j} = \lambda p_{i-1,j} + f_{j+1}(\mu)p_{i,j+1}, \quad i = \overline{1,r-1}, j = \overline{1,N-1}, \\ (\lambda + f_N(\mu))\, p_{i,N} = \lambda p_{i-1,N} + f_1(\mu)p_{i+1,N}, \quad i = \overline{1,r-1}, \\ f_j(\mu)p_{r,j} = \lambda p_{r-1,j} + f_{j+1}(\mu)p_{r,j+1}, \quad j = \overline{1,N-1}, \\ f_N(\mu)p_{r,N} = \lambda p_{r-1,N}, \end{cases} \tag{29}$$

with normalization condition

$$p_0 + \sum_{i=0}^{r}\sum_{j=1}^{N} p_{i,j} = 1. \tag{30}$$

3.2 Marginal Probability Distributions

If we denote by π_j, $j = \overline{0,N}$, the marginal probability distribution of the number of occupied servers and by $\tilde{\pi}_i$, $i \geq 0$, the probability distribution of the number of queries in the buffer, defined in (24) and (23) correspondingly, then from (26) and (28) we obtain:

$$\begin{cases} \lambda\tilde{\pi}_0 = f_1(\mu)(p_{0,1} + p_{1,1}), \\ \lambda\tilde{\pi}_i = f_1(\mu)p_{i+1,1}, \quad i \geq 1. \end{cases} \tag{31}$$

and

$$\begin{cases} \pi_0 = p_0, \\ \pi_j = \dfrac{\lambda}{f_j(\mu)}, \quad j = \overline{1,N}, \end{cases} \tag{32}$$

where $f_j(\mu)$ is defined in (27).

From (32) and normalization condition (28) for π_j, $j = \overline{0,N}$,

$$p_0 + \sum_{j=1}^{N} pi_j = 1,$$

the probability p_0 of the system being empty is obtained:

$$p_0 = 1 - \frac{\lambda}{\mu} \sum_{j=1}^{N} \frac{1}{jC_N^j}. \tag{33}$$

3.3 The Laplace-Stieltjes Transformation of Waiting Time Probability Distribution Function

If we define as $\omega(s)$ the Laplace-Stieltjes transformation (LST) of waiting time PDF for an arbitrary query, $\omega_{i,j}(s)$ — the LST of waiting time PDF for the incoming query when there are i, $i \geq 0$, other queries in the buffer and j, $j = \overline{1,N}$, servers are occupied, then:

$$\omega(s) = p_0 + \sum_{i=0}^{\infty} \sum_{j=1}^{N} \omega_{i,j}(s) p_{i,j} = p_0 + \sum_{i=0}^{\infty} \omega_N^i(s) \sum_{j=1}^{N} \omega_j(s) p_{i,j}, \tag{34}$$

where $\omega_j(s)$ is the LST of PDF (2) for $\eta = \max(\eta_1, \ldots, \eta_j)$, $j = \overline{1,N}$ (see [18]):

$$\omega_j(s) = \sum_{i=1}^{j} (-1)^{i-1} \sum_{k=1}^{C_j^i} k \frac{i\mu}{s + i\mu}. \tag{35}$$

3.4 Probability Generation Function

In this subsection the probability generating function $P(z_1; z_2)$ for the probability distribution p_0 (the system is empty) and $p_{i,j}$, $i = \overline{0,r}$, $j = \overline{1,N}$, is introduced:

$$P(z_1; z_2) = p_0 + \sum_{i=0}^{\infty} z_1^i \sum_{j=1}^{N} p_{i,j} z_2^j.$$

Multiplying each equation of (26) by z_1 and z_2 raised to the corresponding degrees the following relation is obtained:

$$P(z_1; z_2) = \frac{1}{\lambda(1 - z_1)} \left(\lambda p_0 \frac{1 - z_1}{z_1} \left(z_1 - z_2^N \right) \right.$$
$$\left. - f_1(\mu) P_1(z_1) z_2 \frac{z_1 - z_2^{N-1}}{z_1} + (1 - z_2) \sum_{j=2}^{N} f_j(\mu) P_j(z_1) \right), \tag{36}$$

where p_0 is defined by (33) and $P_j(z_1) = \sum_{i=0}^{\infty} p_{i,j} z_1^i$.

4 Numerical Experiment

Here we present numerical results of computation of the probability p_0 of system being empty, probability $\tilde{\pi}_r$ of the buffer being overfull and the mean number $\tilde{N}$ of queries in the buffer in the system with homogeneous servers and a finite-capacity buffer.

The first table gives results for the case with $N = 5$ servers, $r = 100$ and service rate $\mu = 10$.

Table 1. System with $N = 5$ servers, $r = 100$ and service rate $\mu = 10$

λ	p_0	$\tilde{\pi}_r$	$\tilde{N}$
1	0.946667	1.56534e-018	0.00195716
5	0.733333	1.70661e-017	0.0631629
10	0.466667	1.26398e-016	0.397024
15	0.2	4.71081e-016	2.08437
20	2.75912e-006	0.0625026	90.1282
30	2.54953e-034	0.375	98.8256
40	2.16074e-056	0.53125	99.3479
50	≈ 0	0.625	99.5403

The second table gives results for p_0 (computed from (33)) in the system with unlimited buffer, $N = 5$ servers and service rate $\mu = 10$.

Table 2. System with unlimited buffer, $N = 5$ servers and service rate $\mu = 10$

λ	p_0
1	0.9466667
5	0.7333333
10	0.4666667
15	0.2
18	0.0400000

It is easily seen that the probabilities p_0 in the first and the second tables correspond each other (Tables 1 and 2).

5 Conclusions and Furthest Problems

The brief introduction to the mathematical model of cloud computing system based on the queuing system with the splitting of the incoming queries and synchronization of services was presented.

Our future goals are:

- to evaluate probabilities $\tilde{\pi}_i$(31), $i \geq 0$, and $p_{i,j}$ (26), $i \geq 0, j = \overline{1, N}$;
- to evaluate LST (34) for N homogeneous servers;
- to obtain the condition of the steady-state regime existence for the system with unlimited buffer and homogeneous servers;
- to construct and analyze the mathematical model of the system with $N = \alpha K$ homogeneous servers, where K is the number of subqueries for a incoming query and α is a positive integer;
- to construct and analyze the mathematical model of the system with N homogeneous servers and arbitrary number of subqueries for a incoming query.

Acknowledgments. The publication was financially supported by the Ministry of Education and Science of the Russian Federation (the Agreement number 02.a03.21.0008) and partially supported by RFBR Grants No. 15-07-03007, No. 15-07-03406 and No. 14-07-00090.

References

1. Buyya, R., Broberg, J., Goscinski, A.M.: Introduction to Cloud Computing. Cloud Computing: Principles and Paradigms. Wiley, Hoboken (2011)
2. Khazaei, H., Misic, J., Misic, V.B.: A fine-grained performance model of cloud computing centers. IEEE Trans. Parallel Distrib. Syst. **24**(11), 2138–2147 (2012)
3. Flatto, L., Hahn, S.: Two parallel queues created by arrivals with two demands. SIAM J. Appl. Math. **44**(5), 1041–1053 (1984)
4. Nelson, R., Tantawi, A.N.: Approximate analysis of fork/join synchronization in parallel queues. IEEE Trans. Comput. **37**(6), 739–743 (1988)
5. Thomasian, A.: Analysis of fork/join and related queueing systems. ACM Comput. Surv. (CSUR) **47**(17), 17.1–17.71 (2014)
6. Gorbunova, A., Zaryadov, I., Matyushenko, S., Sopin, E.: The estimation of probability characteristics of cloud computing systems with splitting of requests. In: Vishnevskiy, V.M., Samouylov, K.E., Kozyrev, D.V. (eds.) DCCN 2016. CCIS, vol. 678, pp. 418–429. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-51917-3_37
7. Duda, A., Czachórski, T.: Performance evaluation of fork and join synchronization primitives. Acta Informatica **24**(5), 525–533 (1987)
8. Kim, M.Y., Tantawi, A.N.: Asynchronous disk interleaving: approximating access delays. IEEE Trans. Comput. **40**(7), 801–810 (1991)
9. Fiorini, P.M.: Exact analysis of some split merge queues. SIGMETRICS Perform. Eval. Rev. **43**(2), 51–53 (2015)
10. Moiseeva, S., Sinyakova, I.: Investigation of Queueing System GI(2)—M2—∞. In: Proceedings of the Internattional Conference on Modern Probabilistic Methods for Analysis and Optimization of Information and Telecommunication Networks, pp. 219–225 (2011)
11. Moiseeva, S., Sinyakova, I.: Investigation of output flows in the system with parallel service of multiple requests. In: Problems of Cybernetics and Informatics (PCI-2012) : IV International Conference (IEEE), pp. 180–181, Baku, Azerbaijan (2012)

12. Tsimashenka, I., Knottenbelt, W.J.: Reduction of subtask dispersion in fork-join systems. In: Balsamo, M.S., Knottenbelt, W.J., Marin, A. (eds.) EPEW 2013. LNCS, vol. 8168, pp. 325–336. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40725-3_25
13. Dean, J., Barroso, L.: The tail at scale. Commun. ACM **56**(2), 74–80 (2013)
14. Joshi, G., Soljanin, E., Wornell, G.: Efficient redundancy techniques for latency reduction in cloud systems. arXiv preprint. arXiv:1508.03599 (2015)
15. Gardner, K., Harchol-Balter, M., Scheller-Wolf, A.: A better model for job redundancy: decoupling server slowdown and job size. In: Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 1–10. IEEE (2016)
16. Gorbunova, A.V., Kradenyh, A.A., Zaryadov, I.S.: The mathematical model of a cloud computing system. In: Proceedings of the Nineteenth International Scientific Conference: Distributed Computer and Communication Networks: Control, Computation, Communications (DCCN-2016), Youth School-Seminar, vol. 3, pp. 169–175 (2016)
17. Bocharov, P.P., D'Apice, C., Pechinkin, A.V., Salerno, S.: Queueing Theory. VSP, Utrecht, Boston (2004)
18. Harrison, P., Zertal, S.: Queueing models with maxima of service times. In: Kemper, P., Sanders, W.H. (eds.) TOOLS 2003. LNCS, vol. 2794, pp. 152–168. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45232-4_10
19. Neuts, M.F.: Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach. Johns Hopkins University Press, Baltimore (1981)
20. Neuts, M.F.: Matrix-analytic methods in queuing theory. Eur. J. Oper. Res. **15**(1), 2–12 (1984)
21. Neuts, M.F.: Structured Stochastic Matrices of M/G/1 Type and Their Applications. Marcel Dekker Inc., New York (1989)
22. Chakravarthy, S., Alfa, A.S., Attahiru, S.: Matrix-Analytic Methods in Stochastic Models. Taylor & Francis Group, Routledge (1996)
23. Breuer, L., Baum, D.: An Introduction to Queueing Theory and Matrix-Analytic Methods. Springer, Dordrecht (2005)
24. Ibe, O.: Markov Processes for Stochastic Modeling. Elsevier Science, Amsterdam (2013)
25. He, Q.-M.: Fundamentals of Matrix-Analytic Methods. Springer, New-York (2014). https://doi.org/10.1007/978-1-4614-7330-5
26. Trivedi, K.S.: Probability and Statistics with Reliability, Queuing, and Computer Science Applications. Wiley, Hoboken (2016)

http://www.springer.com/978-3-319-71503-2

Analytical and Computational Methods in Probability Theory
First International Conference, ACMPT 2017, Moscow, Russia, October 23-27, 2017, Proceedings
Rykov, V.; Singpurwalla, N.D.; Zubkov, A.M. (Eds.)
2017, XVI, 540 p. 64 illus., Softcover
ISBN: 978-3-319-71503-2