IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 6, 2021, accepted July 6, 2021, date of publication July 14, 2021, date of current version July 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097157

A Multiparameter Analytical Model of the Physical Infrastructure of a Cloud-Based System

SŁAWOMIR HANCZEWSKI, (Member, IEEE), MACIEJ STASIAK, (Member, IEEE), AND MICHAL WEISSENBERG

Faculty of Computing and Telecommunications, Poznan University of Technology, 60-965 Poznań, Poland

Corresponding author: Sławomir Hanczewski (slawomir.hanczewski@put.poznan.pl)

This work was supported by the Polish National Science Centre through the Research Project entitled ‘‘Modeling and Service Quality Evaluation of Internet-Based Services’’ under Grant 2016/23/B/ST7/03925.

ABSTRACT The large popularity of services offered by cloud computing (CC) requires constant expansion of its physical infrastructure. At the same time, CC operators apply various mechanisms that enable the existing physical resources to be optimally used. Useful tools in the design and optimization process are analytical and simulation models. They allow information about the operation of CC to be obtained without the need to make changes to the physical infrastructure. This article presents and discusses a multiservice, multiparameter model of a CC physical infrastructure that provides services of the infrastructure-as-a-service (IaaS) type. In the proposed model, the following four elements are used to specify network settings and describe the demands necessary to activate a virtual machine: the number of processors, the capacity of RAM, the HDD capacity, and the required network bitrate. Such an approach gives the opportunity to describe accurately the use of resources in real servers. To verify and validate the proposed model, we developed and implemented a simulator of the physical infrastructure of CC. The results of the simulations confirm the validity of all the theoretical assumptions adopted in the model. The proposed model can also serve as a tool, in the form of an appropriate application, for determining the resources that are necessary to service calls at the required loss level.

INDEX TERMS Cloud computing, IaaS, modeling.

I. INTRODUCTION

The present-day telecommunications market is usually viewed in terms of requirements approaches and types of requirements specific for users of mobile devices. Indeed, it is the area of wireless access networks in which we observe enormous progress. 4G and 5G mobile networks are fully automated broadband networks that provide voice transmission and transmission of any possible data to their users. The ever-decreasing prices of devices and data transmission make it easier and cheaper for users of mobile networks to transmit large amounts of data. Currently, IP traffic accounts for the most significant part of Internet traffic and is about 38 exabytes per month, and it is expected that in 2025 this traffic will increase to as much as 160 exabytes (10^{18}) per month [1]. However, access networks constitute only a fragment of the whole system of user support. This service would not be possible without appropriate efficient backbone networks and adequate purpose-made high-performance servers that can handle demands. To simplify the management of service processes, servers are typically grouped into large data centers. The largest content delivery networks can be composed of thousands of servers [2], [3]. They make it possible to provide services to end users of any type of network, from instant messaging to social media, and entertainment (online gaming, VoD), to online banking. To simplify the access to physical resources of data centers, the resources are made available in the form of cloud computing instances [4]. The emergence of cloud computing has revolutionized the speed of access to servers. Initially, web-based services were associated exclusively with data storage and access to data from any place in the world. Now, it is a source of physical

The associate editor coordinating the review of this manuscript and approving it for publication was Kezhi Wang.

resources that makes the execution of any task possible. Thanks to a wide range of offered services, cloud-based services are available for both large service operators and individual users interested in their own servers or in memory capacity for storing their own data.

Access to resources of cloud computing can be executed according to three definitions of service [4]:

- Infrastructure as a service (IaaS). This service is accomplished in the form of virtual machines (VMs) that are fully controllable by the user (e.g., it is the user who decides on the installed software).
- Platform as a service (PaaS). Within this group of services, the user obtains access to selected groups of software that constitute a programming platform. Such a platform allows the user to execute and perform any work to achieve the user's objectives. A good example of this type of service is access to programming environments that allow applications to be created, developed, and tested. Most frequently, this type of service is executed (and handled) via a WWW interface.
- Software as a service (SaaS). These services are based on the distribution of software packages. A manufacturer of software makes its programs available without the necessity to download and install them on the user's device. As a result, all maintenance issues, upgrading, or technical assistance for a given application are left to the service provider.

Effective and efficient management of cloud computing requires solutions to a great number of problems related to the maintenance and efficiency of the hardware infrastructure. Access to resources is one of these problems. Operators should make sure that the physical servers that they have at their disposal can effectively manage successive demands, e.g., requests for the creation of VMs on a dedicated system within the scope of IaaS services. Another problem is power consumption and heat abstraction of the heat produced during the time the servers are on. The significance of the problem is proved by available data: In 2018, the electrical energy consumed by data centers was about 1% of the world's total electrical energy consumption. This estimated number is the equivalent of the energy consumption of 17 million households in the United States [5]. To reach the optimum use of physical resources (which, consequently, may lead to a significant curtailment of energy consumption), several algorithms for the occupancy of physical resources have been developed. These algorithms often lead to the equalization of server loads. Good examples of such algorithms are the following: OLB (*opportunistic load balancing*), RR (*round robin*), and CLBDM (*central load balancing decision model)*, randomized, compare and balance, and based random sampling [6]–[8].

A. RELATED WORKS

To analyze cloud systems, analytical or simulation models are used. An important source of information is also provided by measurements in existing CC systems. Measurements make it possible to monitor the performance of systems, while the data obtained as a result of the measurements allow appropriate and relevant analytical and simulation models to be verified. The unquestionable advantage of analytical and simulation models is the short execution time and relatively low costs of obtaining data. The literature on the subject doesn't abound with numerous publications that deal with the modeling of the infrastructure of cloud computing. Analytical models that deal with IaaS services are in preponderance. In [9], the authors present an analytical model of an IaaS infrastructure in which demands for the activation of new VMs differ in the number of processors. This model is based on a hierarchical analysis of birth-death processes. The accompanying assumption is that all servers are identical. [10] proposes a model that takes advantage of the stochastic reward net (SRN) concept [11]. Since the base model was not scalable, it was extended to include two approximations that used computational methods: folding [12] and fixed-point methodology [13]. The model offers the possibility of evaluating and determining the use of resources and the energy consumption. [16] proposes a method for a decomposition of the system into subsystems (a hierarchical architecture is adopted), for which loss probabilities are determined. Then, on the basis of the fixed-point methodology, the final result is determined, i.e., the total losses in the system. In turn, [14], [15] propose models of a mobile edge cloud. Both models are based on an analysis of the birth-death processes that occur in the system. It should be noted that the models using process analysis of the system are associated with the solution of a system of linear equations. The number of these equations depends, among other things, on the system capacity. Hence, the use of this type of models without certain simplifications is ineffective in the case of systems with a very large capacity. In [17], the concept of an analytical model of a cloud infrastructure based on Erlang's ideal grading [18], [19] was proposed. The problems of energy consumption in cloud computing were addressed, among others, in [20], [21], and [22]. The literature also includes simulation models of cloud computing, e.g., CLoudSim [23] and GreenCloud [24].

B. RESEARCH CONTRIBUTION

The main achievements of this article are as follows:

- An approximate multi-service model of the physical infrastructure of cloud computing is proposed. In the proposed model, the resources of cloud computing and the parameters of VMs are described by the following four parameters: the number of processors, the capacity of RAM, the capacity of the hard disk, and the bitrate. The assumption in the model is that activated VMs can be divided into different classes depending on the demanded resources. To the best of our knowledge, this is the first multi-service analytical model of cloud computing that concurrently takes into consideration the demands of a call for the following resources: a suitable number of processors, an appropriate size of RAM, suitable capacity of the hard disk, and an appropriate bitrate.

- For the construction of the proposed cloud model, the following multi-service models were used: the model of multi-service full-availability resources, the model of multi-service resources with limited availability, and the fixed-point methodology.
- For the verification and validation of the model, a simulator of cloud computing, developed and implemented by the authors, was used. The simulator uses the event-scheduling simulation methodology.
- With the help of the proposed model, it is possible to determine the blocking probability, i.e., the probability of not being able to activate new VMs. The accuracy of the model does not depend on the number of physical machines (servers).

The remaining part of the article is structured as follows. Section II presents the considered structure of cloud computing. Section III discusses the basic multiservice models of resources in telecommunications and computer systems. Then, in Section IV, an analytical model of the physical infrastructure of cloud computing is proposed. Section V presents sample results of the analytical modeling of several selected cloud systems, which are then compared with the relevant results of digital simulation. Finally, Section VI briefly summarizes the article.

II. PHYSICAL INFRASTRUCTURE OF CLOUD COMPUTING

The idea of servicing new VM requests is briefly presented in the diagram shown in Fig. 1. New VM requests are received by controlling devices: the Main Resources Manager (MRM). To simplify the resources management, physical machines (PMs, servers) are grouped into sets with specific functionality in a required number k. Each set is controlled by a dedicated device (the Group Manager, or GM) that is responsible for the activation of VMs. PMs can be grouped depending on the physical location of devices or their particular specifications (a group is composed of devices with identical or similar specifications). The MRM sends a new request to a selected GM to activate a new VM. The GM forwards information about the loads of PMs to the MRM. Each PM is described by the following resources that are used to activate new VMs [17]:

- C_{P} – the number of processors,
- C_{R} – the total capacity of RAM,
- C_{D} – the total capacity of the hard disk,
- C_{bps} – the total bitrate of a network link.

A demand for the creation of a new VM of class i can be described by the four-element set $\mathbf{VM}_i = \{c_{i,\mathrm{P}}, c_{i,\mathrm{R}}, c_{i,\mathrm{D}}, c_{i,\mathrm{bps}}\}$, where

- $c_{i,\mathrm{P}}$ – the number of demanded processors (cores),
- $c_{i,\mathrm{R}}$ – the demands for capacity of RAM,
- $c_{i,\mathrm{D}}$ – the demanded capacity of the hard disk,
- $c_{i,\mathrm{bps}}$ – the demanded speed of a network link,

where i denotes the class of a VM, understood as a group of machines that require identical values of the parameters of set $\mathbf{VM}_i$. Typically, it is assumed that the number of classes of VM is equal to m.

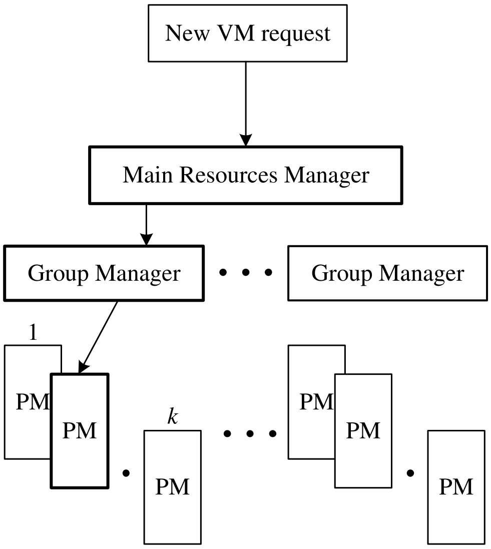

FIGURE 1. Service of a new VM request.

TABLE 1. Parameters of the VMs of the AV2-series.

Instance	vCPU	RAM [GB]	Storage [GB]
A1 v2	1	2	10
A2 v2	2	4	20
A2m v2	2	16	20
A4 v2	4	8	40
A4m v2	4	32	40
A8 v2	8	16	80

TABLE 2. Parameters of the VMs of the D2d-D48d-series.

Instance	vCPU	RAM [GB]	Storage [GB]
D2d v4	2	8	75
D4d v4	4	16	150
D8d v4	8	32	300
D16d v4	16	64	600
D32d v4	32	128	1200
D48d v4	48	192	1800

After a new request is received, the MRM attempts to locate the place where a new machine can be activated. This process is executed according to the physical resources allocation algorithm implemented in the system. After a PM is chosen, information about the new VM is forwarded to the GM that is responsible for the activation of the machine. The number of machines that can be activated on one PM depends on the physical resources of the PM and on the set of parameters that describe VMs. Tables 1 and 2 show the parameters of the VMs that are offered by the Microsoft Azure cloud computing platform [25].

The parameters of VMs can differ significantly (Table 1) or can be changed in a linear way (Table 2). Table 3 shows selected specifications (max parameters) for server that can be used to construct a cloud [26].

The assumption adopted in this article is that a new call of class i for the activation of a VM will be serviced only when at least one PM at a given moment has free resources

TABLE 3. Parameters of the DELL PowerEdge R740xd rack server.

RAM	**max**
RDIMM LRDIMM NVDIMM DCPMM	1.53TB 3TB 192GB DCPMM 6.14TB
Processor	
2nd Generation Intel Xeon Scalable processors	2 x 28 cores
Disk	
Front Bays	24 x 2.5” SAS/SSD/NVMe, max 184TB 12 x 3.5” SAS, max 192TB
Mid Bays	4 x 3.5” SAS, max 64TB 4 x 2.5” SAS/SSD, max 30.72TB
Rear Bays	4 x 2.5” SAS/SSD, max 30.72TB 2 x 3.5” SAS, max 32TB
Network card	
1GbE 10GbE +1GbE 10GbE 25GbE	4 2 4 2

that satisfy the demands $\mathbf{VM}_i$ of this call. Another assumption is that the algorithm for server choice provides equalization of average loads of all servers. Because the lifetime of a VM exceeds the waiting time for its creation (activation), yet another assumption is that the waiting time for a machine to be activated will not be taken into consideration, whereas each call will be admitted for service only if the cloud has free resources. If, in a given moment of receiving a demand, there are no free resources, this call will be rejected.

III. BASICS OF THE MODELING OF MULTISERVICE SYSTEMS

The models of multiservice systems known from the theory of telecommunications and computer traffic provide the basis of the proposed cloud model: the FAR (full available resources) model, also known as the full-availability group model [27], [28], and the LAR (limited available resources) model, also known as the limited-availability group model [29], [30]. This section also contains basic information about the fixed-point method.

A. FAR MODEL

FAR is a model of full-availability resources. The assumption is that resources with a capacity of C allocation units (AUs) are offered m classes of Erlang calls, whereas each call of class i ($i\epsilon\{1,\ldots,m\}$) requires c_i AUs to be serviced. A new call of class i will be serviced only if the FAR model has c_i free units. The occupancy distribution for FAR is described by the following formula:

$$n[P(n)]_C = \sum_{i=1}^{m} a_i c_i [P(n-c_i)]_C, \tag{1}$$

where a_i is the intensity of traffic of class i and $[P(n)]_C$ is the occupancy probability of n AUs in FAR with a capacity of C AUs. The blocking probability for calls of class i is determined by the following formula:

$$e_i = \sum_{n=C-c_i+1}^{C} [P(n)]_C. \tag{2}$$

To simplify the presentation, we use the following notation:

$$\mathbf{e} = \mathrm{FAR}(\mathbf{a}, \mathbf{c}, C), \tag{3}$$

where $\mathbf{a}$ is a set of intensities of offered traffic:

$$\mathbf{a} = \{a_1, a_2, \ldots, a_m\}; \tag{4}$$

$\mathbf{c}$ is a set of demands of individual classes, expressed in number of AUs:

$$\mathbf{c} = \{c_1, c_2, \ldots, c_m\}; \tag{5}$$

and $\mathbf{e}$ is a set of blocking probabilities of individual classes:

$$\mathbf{e} = \{e_1, e_2, \ldots, e_m\}. \tag{6}$$

B. LAR MODEL

LAR is a model of resources that are composed of k identical separate resources. A call of class i that requires c_i AUs can be admitted for service only when it can be entirely serviced by one of the separate resources. This means that all c_i units must be serviced only by one, from k, separate resources, so there is no possibility of dividing c_i units between a number of separate resources.

The occupancy distribution in LAR can be determined by:

$$n[P(n)]_{kC} = \sum_{i=1}^{m} A_i c_i \sigma_i(n-c_i)[P(n-c_i)]_{kC}, \tag{7}$$

where

- A_i – the traffic intensity of traffic class i offered to LAR;
- $[P(n)]_{kC}$ – the occupancy probability of n AUs in LAR with a total capacity of kC units, where C is the capacity of single resources;
- $\sigma_i(n)$ – the so-called conditional transition probability for transitions between neighboring occupancy states in LAR:

$$\sigma_i(n) = 1 - \frac{F(kC-n, k, c_i-1)}{F(kC-n, k, C)}, \tag{8}$$

where $F(x, k, c)$ is the number of possible distributions of x free (unoccupied) AUs in k separate resources, where each of the resources has a capacity of C units:

$$F(x,k,f)=\sum_{i=0}^{\left\lfloor \frac{x}{f+1} \right\rfloor} (-1)^i \binom{k}{i} \binom{x+k-1-i(f+1)}{k-1}. \tag{9}$$

The blocking probability for calls of class i can be determined by the following formula:

$$E_i = \sum_{n=0}^{kC} [P(n)]_{kC}(1-\sigma_i(n)). \tag{10}$$

To simplify the presentation, we adopt the following notation:

$$\mathbf{E} = \mathrm{LAR}(\mathbf{A}, \mathbf{c}, kC), \tag{11}$$

where $\mathbf{A}$ is a set of intensities of offered traffic:

$$\mathbf{A} = \{A_1, A_2, \ldots, A_m\}; \tag{12}$$

$\mathbf{c}$ is a set of demands of individual classes, expressed in number of AUs:

$$\mathbf{c} = \{c_1, c_2, \ldots, c_m\}; \tag{13}$$

and $\mathbf{E}$ is the set of call blocking probabilities for calls of individual classes in LAR:

$$\mathbf{E} = \{E_1, E_2, \ldots, E_m\}. \tag{14}$$

C. FIXED-POINT METHOD

In the fixed-point (FP) method, the problem of determining the blocking probability for calls demanding access to several resources (subsystems) at the same time is solved based on the following scheme. It is assumed that a given resource is offered only the part of total traffic that is not lost in other resources demanded by the given call. This assumption determines how the blocking probability is determined.

Let us assume that a call class i ($1 \leq i \leq m$) demands access to s resources at the same time. The offered traffic by class i calls to resources j ($1 \leq j \leq s$) is defined as follows:

$$A_i(j) = A_i \prod_{l=1, l\neq j}^{s} [1 - E_i(l)], \tag{15}$$

where $E_i(l)$ is the blocking probability for calls of class i in resource l.

Note that for determining the offered traffic $A_i(j)$, it is necessary to know the value of blocking probability $E_i(l)$ in other resources, i.e., all $l \neq j$. Therefore, the FP method is an iterative method.

FP Method

1) Initialization of the iteration step: $z = 0$.
2) Determining the initial approximations for the blocking probabilities of all call classes in all resources:

$$\bigwedge_i \bigwedge_l E_i^{(0)}(l) = 0. \tag{16}$$

3) Increasing the iteration step:

$$z = z + 1. \tag{17}$$

4) Determination of the value of the offered traffic $A_i^k(l)$:

$$\bigwedge_i \bigwedge_l A_i^{(z)}(j) = A_i \prod_{l=1, l\neq j}^{s} [1 - E_i^{(z-1)}(l)]. \tag{18}$$

5) Determination of the blocking probability for all call classes in all resources:

$$\bigwedge_i \bigwedge_l E_i^{(z)}(j) = \mathrm{FUN}(A^{(z)}, \mathbf{c}_j, C_j), \tag{19}$$

where FUN is a function determining the blocking probability of individual call classes in individual resources, e.g., FAR or LAR, and $\mathbf{c}_j$ is a set of demands of individual classes in resource j.

6) Determination of the total blocking probability for particular call classes:

$$\bigwedge_i E_i^{(z)} = \prod_{l=1, l\neq j}^{s} [1 - E_i^{(z)}(l)]. \tag{20}$$

7) Checking the accuracy of the calculations:

$$\bigwedge_i \left| \frac{E_i^{(z)} - E_i^{(z-1)}}{E_i^{(z)}} \right| \leq \epsilon. \tag{21}$$

If for all i the condition is not met, go to Step 3; otherwise, $E_i = E_i^{(z)}$ and the calculations end.

In the presented algorithm, it is assumed that $X^{(z)}$ is the value of parameter X in the z-th iteration step. The ϵ parameter is the absolute error of the calculations, which specifies the accuracy of the iteration process.

To simplify the presentation, we adopt the following notation:

$$\mathbf{E} = \mathrm{FP}(\mathbf{A}, \mathbf{c}_{\mathrm{FP}}, \mathbf{C}_{\mathrm{FP}}), \tag{22}$$

where

- $\mathbf{A}$ is the set of intensities of offered traffic (it should be stressed that this set $\mathbf{A}$ is the same for all demanded resources):

$$\mathbf{A} = \{A_1, A_2, \ldots, A_m\}; \tag{23}$$

- $\mathbf{c}_{\mathrm{FP}}$ is the set of demands of individual classes in particular resources:

$$\mathbf{c}_{\mathrm{FP}} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s\}, \tag{24}$$

where

$$\mathbf{c}_l = \{c_1, c_2, \ldots, c_m\}, \quad 1 \leq l \leq s; \tag{25}$$

- $\mathbf{C}_{\mathrm{FP}}$ is the set of capacities of particular resources:

$$\mathbf{C}_{\mathrm{FP}} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_s\}. \tag{26}$$

IV. PROPOSED MODEL

The proposed analytical model that makes it possible to determine the blocking probabilities in a system offering IaaS services takes into consideration the following factors: the physical architecture of cloud computing, the parameters that describe VMs, and the fact that a VM is activated on one PM only (i.e., all the resources required by a new VM are in one server).

Assume that the physical infrastructure of cloud computing (presented in Fig. 1) is composed of k identical PMs. A call to set up a new machine is described by the relevant sets of parameters $\mathbf{VM}_i$, where $1 \leq i \leq m$. New calls are received by the Cloud Manager and are forwarded according to the implemented resources allocation algorithm to the appropriate server.

Activation of a new VM on a PM can be effected only if this PM has enough free resources defined by a call $\mathbf{VM}_i$. A decomposition of PMs into four independent system components is carried out in the model: processors (P), RAM (R), the hard disk (D), and bitrate (bps) (Fig. 2). The assumption is that each system component services demands that are proper for it; i.e., P will service demands that are related to the number of processors, R will service demands that are related to RAM, and so on.

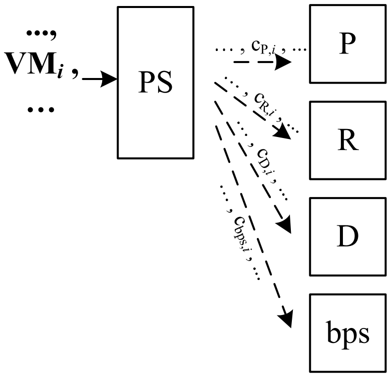

FIGURE 2. **Decomposition of a PM into subsystems.**

A call of class i can be serviced in a given server only when the server has free AUs necessary for this call to be serviced in each of its subsystems. Then, the possibility for this call to be serviced in the cloud is based on the identification of at least one such server from among the k servers that create the cloud system.

To determine the total blocking probability in a web-based system (cloud), the LAR model cannot be used directly. This model describes a system composed of k separate resources. Therefore, the model allows us to determine the blocking probability for calls of a given class in k separate resources (PMs), each with the capacity of a given subsystem in the individual servers. A demand for access to all the subsystems requires an appropriate analytical method that takes into consideration the simultaneity of service to a given call in the relevant subsystems. An application of this method to a decomposed group of servers (consisting of LAR processors, LAR RAM, etc.) leads to misleading and erroneous results, since the obtained solution may also take into consideration those states (treating them as non-blocking states) in which the available resources in one subsystem, e.g., P, are located on one server while the available resources of another subsystem, e.g., R, are located on another server.

The model proposed here can be written (in a simplified form) in the form of the following computational method.

Method Cloud 1. 1. Determination of the blocking probability in each of the subsystems of a single PM:

$$\mathbf{e}_x = \mathrm{FAR}(\mathbf{a}, \mathbf{c}_X, C_X), \tag{27}$$

where the set $\mathbf{c}_x = \{c_{1,x}, c_{2,x}, \ldots, c_{m,x}\}$ is the set of demands for AUs by individual call classes in a given subsystem $\mathbf{X} = \{P, R, D, bps\}$.

The set C_X determines the capacity of subsystem X in a given PM.

2. Determination of the blocking probability in a group of k subsystems (separate resources) X, where $\mathbf{X} = \{P, R, D, bps\}$:

$$\mathbf{E}_X = \mathrm{LAR}(k\mathbf{a}, \mathbf{c}_X, kC_X). \tag{28}$$

3. Determination of the set ρ_X of the relations between the blocking probability in subsystem $\mathbf{e}_X$ and a group of subsystems $\mathbf{E}_X$:

$$\rho_X = \{\rho_{1,X}, \rho_{2,X}, \ldots, \rho_{m,X}\}, \tag{29}$$

where

$$\bigwedge_{i=\{1,2,\ldots,m\}} \rho_{i,X} = \frac{E_{i,X}}{e_{i,X}}. \tag{30}$$

4. Determination of the blocking probability in a single PM (in each of the subsystems) on the basis of the FP methodology:

$$\mathbf{e}^* = \mathrm{FP}(\mathbf{a}, \mathbf{c}_{\mathrm{CC}}, \mathbf{C}_{\mathrm{CC}}), \tag{31}$$

$$\bigwedge_{X=\{P,R,D,bps\}} \mathbf{e}^* = \{e^*_{1,X}, e^*_{2,X}, \ldots, e^*_{m,X}\}, \tag{32}$$

where

- $\mathbf{C}_{\mathrm{CC}}$ is the set of capacities of all the subsystems in a given PM:

$$\mathbf{C}_{\mathrm{CC}} = \{C_{\mathrm{P}}, C_{\mathrm{R}}, C_{\mathrm{D}}, C_{\mathrm{bps}}\}; \tag{33}$$

- $\mathbf{c}_{\mathrm{CC}}$ is the set of demands of VM classes:

$$\mathbf{c}_{\mathrm{CC}} = \{\mathbf{VM}_1, \mathbf{VM}_2, \ldots, \mathbf{VM}_m\}. \tag{34}$$

5. Determination of the blocking probability $\mathbf{E}$ in the cloud:

$$\mathbf{E} = \{E_1, E_2, \ldots, E_m\},$$

$$\bigwedge_{i=\{1,2,\ldots,m\}} E_i = 1 - \Big[(1 - E^*_{i,P})(1 - E^*_{i,R}) \tag{35}$$

$$(1 - E^*_{i,D})(1 - E^*_{i,bps})\Big], \tag{36}$$

$$\bigwedge_{i=\{1,2,\ldots,m\}} \bigwedge_{X=\{P,R,D,bps\}} E^*_{i,X} = \rho_{i,X} e^*_{i,X}. \tag{37}$$

The proposed method solves the problem of servicing a call that demands simultaneity of free resources in each subsystem of a single server.

In Step 1, the blocking probability in each of the subsystems in a single PM is determined on the basis of the FAR model. The accompanying assumption is that, as a result of the operation of load equalization algorithms, the traffic offered to a single PM $\mathbf{a}$ is k times lower than the traffic $k\mathbf{a}$ offered to the cloud.

In Step 2, the blocking probability E_X in each group of subsystems in the cloud is determined on the basis of the LAR model. The assumption is that LAR is composed of k separate resources with capacities C_X that correspond to the capacities of the subsystems of a single PM.

In Step 3, the relations ρ_X between the blocking probability e_X in single resources and the blocking probability E_X in a group of subsystems are determined. Direct application of the LAR model to determine the loss probabilities in the whole system is not possible, since, according to the adopted definition of a VM, the latter is described by four parameters and they have to be available in one physical server. The application of the FP methodology to LAR, understood as a group of separate resources, would eventually lead to a situation in which the elements of a VM would reside in different servers. Hence, the relation ρ for each of the system components must be determined. This relation makes it possible to take into consideration the multiserver structure in the calculations of a single server.

The relation ρ is a key element in the model, for the assumption is that the same relation will be effected between the total blocking probability in a single server and the total blocking probability in a group of servers.

In Step 4, the real blocking probability e_X^* in the subsystems of a single server, which corresponds to the total blocking probability e_i^* of the server determined on the basis of the FP methodology, is determined. The method allows us to determine the loss probability in a subsystem with the assumption that a call that has been rejected in one subsystem is not offered to another subsystems.

In Step 5, the total blocking probability in the cloud computing system is determined on the basis of (35). The accompanying assumption is that the real blocking probabilities E_X^* in each group of subsystems, which satisfy the condition of simultaneous availability for all subsystems for a given call in a single server, are subject to the relation ρ to the real probability e_X^* in a single subsystem that meets the condition of the simultaneous availability of all subsystems in this server.

In Fig. 3, the idea of the Cloud 1 method is presented. The figure shows the resources of physical servers (in the example of available processors) that are used for calculations in the individual steps of the proposed method.

V. EXPERIMENTAL RESULTS

To validate (and verify) the operation of the proposed multiparameter analytical model, the results of the model were compared with the results of a digital simulation. For this purpose, a simulation model of a cloud system composed of k servers on which VMs are to be created was developed and implemented in the C++ language. Each server had an identical number of processors, an identical capacity of RAM and disks, and an identical bitrate for the available subsystems. An event-scheduling method was used in the simulator. To determine a single measurement, 10 series of simulations were performed, each with 1,000,000 calls of the class that demanded the highest number of AUs for service. The obtained results are plotted as a function of traffic offered to a single AU available in the subsystem that corresponds to RAM

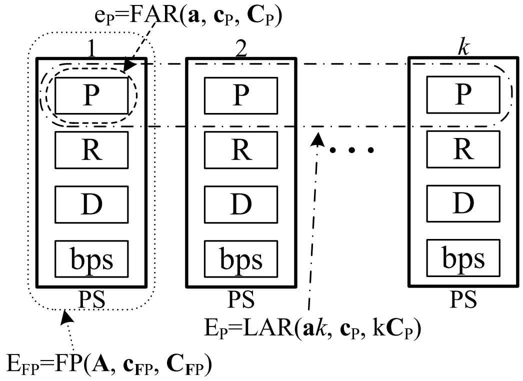

FIGURE 3. Idea of the cloud 1 method.

TABLE 4. AU definitions.

RAM	1AU = 8GB
Processor	1AU = 2 cores
Disk	1AU = 75GB
Bandwidth	1AU = 100Mbps

$$a = \frac{\sum_{i=1}^{m} a_i c_{i,\mathrm{R}}}{C_\mathrm{R}}. \tag{38}$$

Another assumption was that the total offered traffic was divided between individual call classes in the following proportions: $a_1c_{1,\mathrm{R}} : a_2c_{2,\mathrm{R}} : \ldots : a_mc_{m,\mathrm{R}}=1 : 1 : \ldots : 1$.

Figs. 4–7 show the blocking probabilities obtained for 4 different systems (Table 5). The capacity of each considered system and VMs demands are expressed in AUs.

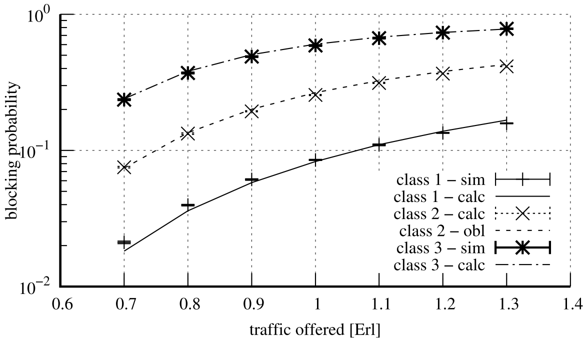

FIGURE 4. Blocking probability in system 1.

Fig. 4 shows the blocking probability for System 1, which was composed of three identical PMs ($k = 3$) with a RAM capacity of 256 GB, 48 processor cores, a disk size of 2100 GB, and an available bandwidth of 4 GB. On the basis of the AUs definitions (Table 4), the capacity of each PM was recalculated and expressed in AUs. A similar operation was carried out for the other considered systems. However, for the simplicity of this operation, the original PM parameters were omitted. They can be determined on the basis of the value of the corresponding parameter expressed in AUs and the definition of AU. VMs requests, too, are expressed in AUs.

TABLE 5. Parameters of the considered systems.

System 1				
No. of PMs	Server capacity [AU]			
$k = 3$	$C_{\mathrm{R}} = 32$	$C_{\mathrm{P}} = 24$	$C_{\mathrm{D}} = 28$	$C_{\mathrm{bps}} = 40$
VM demands [AU]				
VM class 1	$c_{1,\mathrm{R}} = 1$	$c_{1,\mathrm{P}} = 1$	$c_{1,\mathrm{D}} = 1$	$c_{1,\mathrm{bps}} = 2$
VM class 2	$c_{2,\mathrm{R}} = 2$	$c_{2,\mathrm{P}} = 2$	$c_{2,\mathrm{D}} = 2$	$c_{2,\mathrm{bps}} = 2$
VM class 3	$c_{3,\mathrm{R}} = 4$	$c_{3,\mathrm{P}} = 4$	$c_{3,\mathrm{D}} = 4$	$c_{3,\mathrm{bps}} = 2$
System 2				
No. of PMs	Server capacity [AU]			
$k = 3$	$C_{\mathrm{R}} = 32$	$C_{\mathrm{P}} = 40$	$C_{\mathrm{D}} = 40$	$C_{\mathrm{bps}} = 40$
VM demands [AU]				
VM class 1	$c_{1,\mathrm{R}} = 1$	$c_{1,\mathrm{P}} = 2$	$c_{1,\mathrm{D}} = 1$	$c_{1,\mathrm{bps}} = 2$
VM class 2	$c_{2,\mathrm{R}} = 2$	$c_{2,\mathrm{P}} = 2$	$c_{2,\mathrm{D}} = 2$	$c_{2,\mathrm{bps}} = 2$
VM class 3	$c_{3,\mathrm{R}} = 4$	$c_{3,\mathrm{P}} = 3$	$c_{3,\mathrm{D}} = 2$	$c_{3,\mathrm{bps}} = 2$
System 3				
No. of PMs	PM capacity [AU]			
$k = 3$	$C_{\mathrm{R}} = 36$	$C_{\mathrm{P}} = 30$	$C_{\mathrm{D}} = 40$	$C_{\mathrm{bps}} = 40$
VM demands [AU]				
VM class 1	$c_{1,\mathrm{R}} = 1$	$c_{1,\mathrm{P}} = 1$	$c_{1,\mathrm{D}} = 3$	$c_{1,\mathrm{bps}} = 1$
VM class 2	$c_{2,\mathrm{R}} = 2$	$c_{2,\mathrm{P}} = 2$	$c_{2,\mathrm{D}} = 3$	$c_{2,\mathrm{bps}} = 2$
VM class 3	$c_{3,\mathrm{R}} = 3$	$c_{3,\mathrm{P}} = 4$	$c_{3,\mathrm{D}} = 2$	$c_{3,\mathrm{bps}} = 5$
System 4				
No. of PMs	PM capacity [AU]			
$k = 5$	$C_{\mathrm{R}} = 24$	$C_{\mathrm{P}} = 24$	$C_{\mathrm{D}} = 32$	$C_{\mathrm{bps}} = 28$
VM demands [AU]				
VM class 1	$c_{1,\mathrm{R}} = 1$	$c_{1,\mathrm{P}} = 1$	$c_{1,\mathrm{D}} = 1$	$c_{1,\mathrm{bps}} = 1$
VM class 2	$c_{2,\mathrm{R}} = 2$	$c_{2,\mathrm{P}} = 2$	$c_{2,\mathrm{D}} = 3$	$c_{2,\mathrm{bps}} = 1$
VM class 3	$c_{3,\mathrm{R}} = 4$	$c_{3,\mathrm{P}} = 3$	$c_{3,\mathrm{D}} = 3$	$c_{3,\mathrm{bps}} = 2$
System 5				
Offered traffic	Server capacity [AU]			
$a = 1$	$C_{\mathrm{R}} = 16$	$C_{\mathrm{P}} = 16$	$C_{\mathrm{D}} = 14$	$C_{\mathrm{bps}} = 12$
VM demands [AU]				
VM class 1	$c_{1,\mathrm{R}} = 1$	$c_{1,\mathrm{P}} = 1$	$c_{1,\mathrm{D}} = 1$	$c_{1,\mathrm{bps}} = 2$
VM class 2	$c_{2,\mathrm{R}} = 2$	$c_{2,\mathrm{P}} = 1$	$c_{2,\mathrm{D}} = 1$	$c_{2,\mathrm{bps}} = 1$
VM class 3	$c_{3,\mathrm{R}} = 3$	$c_{3,\mathrm{P}} = 2$	$c_{3,\mathrm{D}} = 1$	$c_{3,\mathrm{bps}} = 2$
System 6				
Offered traffic	PM capacity [AU]			
$a = 1$	$C_{\mathrm{R}} = 16$	$C_{\mathrm{P}} = 16$	$C_{\mathrm{D}} = 14$	$C_{\mathrm{bps}} = 12$
VM demands [AU]				
VM class 1	$c_{1,\mathrm{R}} = 1$	$c_{1,\mathrm{P}} = 2$	$c_{1,\mathrm{D}} = 1$	$c_{1,\mathrm{bps}} = 1$
VM class 2	$c_{2,\mathrm{R}} = 2$	$c_{2,\mathrm{P}} = 2$	$c_{2,\mathrm{D}} = 2$	$c_{2,\mathrm{bps}} = 1$
VM class 3	$c_{3,\mathrm{R}} = 4$	$c_{3,\mathrm{P}} = 2$	$c_{3,\mathrm{D}} = 3$	$c_{3,\mathrm{bps}} = 3$

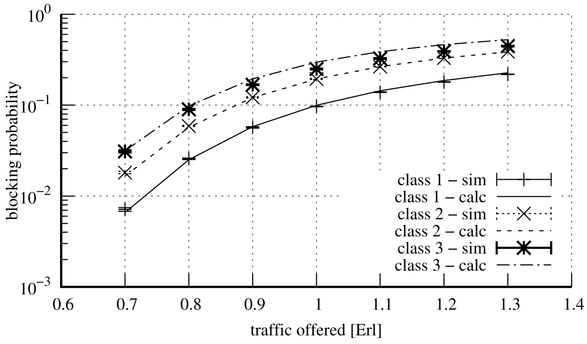

FIGURE 5. Blocking probability in system 2.

Figs. 8–9 show the results obtained for two different systems in which constant traffic was offered ($a = 1$ Erl.) as a function of the number of PMs. The number of PMs was in the range of 2–10.

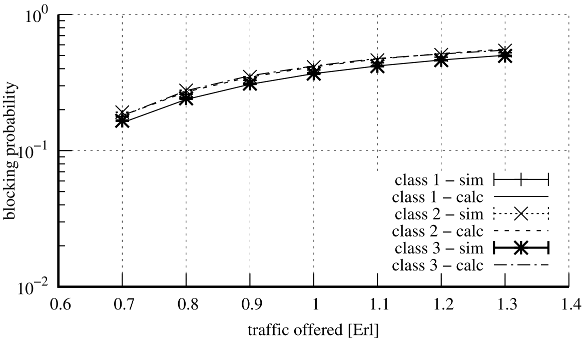

FIGURE 6. Blocking probability in system 3.

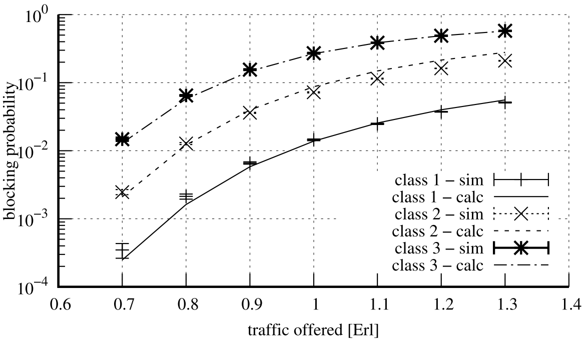

FIGURE 7. Blocking probability in system 4.

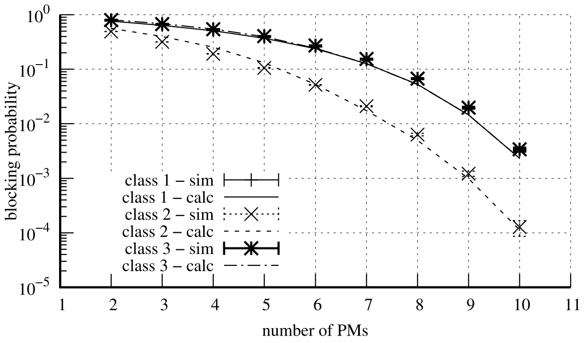

FIGURE 8. Blocking probability in system 5.

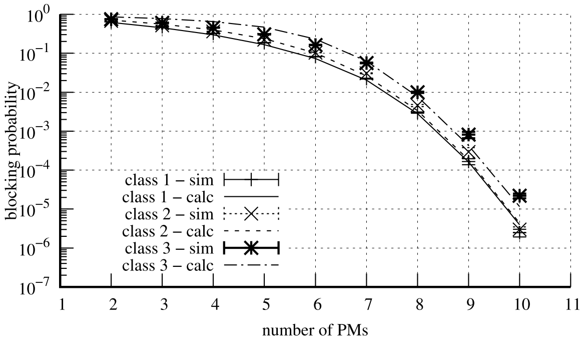

FIGURE 9. Blocking probability in system 6.

The proposed analytical model is an approximate one. However, the presented results clearly show that the model approximates cloud systems well. The accuracy of the model

TABLE 6. Relative error between the results of simulation and calculation for class 1 in System 1.

Traffic	Blocking probability			relative
offered [Erl]	calculation	simulation	±Δ	error [%]
0.7	0.0182001	0.0211088	0.000471517	13.7796
0.8	0.0359843	0.0396389	0.000502649	9.2197
0.9	0.0580409	0.061103	0.000745327	5.0114
1.0	0.0830069	0.0851476	0.000635586	2.5141
1.1	0.110054	0.109928	0.00149726	0.1146
1.2	0.138295	0.134285	0.000976096	2.9862
1.3	0.166743	0.158056	0.00126403	5.4962

TABLE 7. Relative error between the results of simulation and calculation for class 2 in System 1.

Traffic	Blocking probability			relative
offered [Erl]	calculation	simulation	±Δ	error [%]
0.7	0.0741812	0.075295	0.000925046	1.4792
0.8	0.134925	0.132964	0.00167495	1.4748
0.9	0.201024	0.19387	0.00157178	3.6901
1.0	0.265596	0.255171	0.00191014	4.0855
1.1	0.325505	0.313309	0.00271551	3.8926
1.2	0.379795	0.366402	0.00158539	3.6553
1.3	0.428293	0.414395	0.00150072	3.3538

TABLE 8. Relative error between the results of simulation and calculation for class 3 in System 1.

Traffic	Blocking probability			relative
offered [Erl]	calculation	simulation	±Δ	error [%]
0.7	0.239756	0.235851	0.00154874	1.6557
0.8	0.381498	0.370748	0.00226975	2.8995
0.9	0.505444	0.48869	0.00277739	3.4283
1.0	0.60361	0.589649	0.00243425	2.3676
1.1	0.678819	0.66844	0.00357067	1.5527
1.2	0.73628	0.733533	0.00166143	0.3745
1.3	0.780408	0.782785	0.0012316	0.3037

does not depend on the offered traffic and number of physical servers (PMs) in the system. Additionally, to confirm the correctness of the model operation, the obtained results for System 1 are presented in tabular form. Tables 6–8 show the blocking probabilities obtained with the use of the analytical and simulation models for System 1 for each class of offered traffic. In the following columns, the offered traffic is specified, followed by the value of the blocking probability for a given class obtained with the use of the analytical model, and then the blocking probability obtained with the use of the simulation model together with the confidence interval. The last column shows the relative error between the values obtained by means of calculations and simulations. The relative error observed for class 1, at low offered traffic load (0.7 Erl), is at a level of 13%. This is due to from the very low blocking probability. The absolute error for this offered traffic is 0.0029087, with the blocking probability of 2%. This is a typical phenomenon for analytical models in the case of low values of offered traffic.

It should be stressed that the proposed method for servers with identical parameters is fully scalable. This is due to the fact that all components of the proposed method, i.e., the fixed-point methodology, the full-availability resource model, and the model of resources with limited availability, are scalable. All these models are widely used in modeling modern telecommunications systems and networks. This means that the proposed model can be applied to calculations of cloud computing with the parameters that are individually set up by providers. The computational complexity of the proposed model is due to the complexity of its components. Since the calculations are based on the Kaufman-Roberts recursion, the complexity of the model is of the order $O(m\max(C_{\mathrm{P}}, C_{\mathrm{R}}, C_{\mathrm{D}}, C_{\mathrm{bps}}))$ [31]. The proposed model is easy to use. To perform calculations, the parameters of a single PM as well as the number of PMs are required. This will provide a basis for calculations; after a calculated adjustment factor is introduced, it is possible to determine precisely and accurately the loss coefficient for traffic classes offered to a system. The validated accuracy of the model makes it possible to use it to model real cloud-based systems that are based on the IaaS structure. The model can be used in designing a new cloud infrastrucure or in optimizing the existing cloud infrastructure.

VI. CONCLUSION

This article presents an approximate multiparameter analytical model of the physical infrastructure of a cloud-based system that offers IaaS services. The results obtained by the model are compared with the results of a digital simulation, which confirms the validity of all theoretical assumptions of the model. The model is not complex and is easily programmable. The proposed model in the article can be used in designing new cloud computing systems and or in optimizing existing cloud systems. The model takes into consideration the real parameters of VMs. The model allows the loss level (i.e., not being able to activate new VMs with the required parameters) to be determined. The model might also be used to determine the required capacity of the infrastructure, i.e., the number of servers with the required parameters for which the loss level does not exceed the required level.

REFERENCES

[1] *Ericsson Mobility Report*, Ericsson, Stockholm, Sweden, Nov. 2019.

[2] (Jun. 1, 2021). *Azure Global Infrastructure*. [Online]. Available: https://azure.microsoft.com/

[3] (Jun. 1, 2021). *Google Cloud Infrastructure*. [Online]. Available: https://cloud.google.com/

[4] P. Mell and T. Grance, "The NIST definition of cloud computing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-145, 2011.

[5] S. Lohr, "Cloud computing is not the energy hog that had been feared," New York Times, Feb. 2020.

[6] R. Kaur and P. Luthra, "Load balancing in cloud computing," in *Proc. Int. Conf. Recent Trends Inf., Telecommun. Comput. (ITC)*, 2012, pp. 1–8.

[7] K. Nuaimi, N. Mohamed, M. Nuaimi, and J. Al-Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms," in *Proc. 8th IEEE Int. Conf. Cloud Comput.*, Dec. 2012, pp. 137–142.

[8] H. Rai, S. K. Ojha, and A. Nazarov, "Cloud load balancing algorithm," in *Proc. 2nd Int. Conf. Adv. Comput., Commun. Control Netw. (ICACCCN)*, Dec. 2020, pp. 861–865.

[9] X. Chang, R. Xia, J. K. Muppala, K. S. Trivedi, and J. Liu, "Effective modeling approach for IaaS data center performance analysis under heterogeneous workload," *IEEE Trans. Cloud Comput.*, vol. 6, no. 4, pp. 991–1003, Oct. 2018.

[10] E. Ataie, R. Entezari-Maleki, L. Rashidi, K. S. Trivedi, D. Ardagna, and A. Movaghar, "Hierarchical stochastic models for performance, availability, and power consumption analysis of IaaS clouds," *IEEE Trans. Cloud Comput.*, vol. 7, no. 4, pp. 1039–1056, Oct. 2019.

[11] R. Shojaee, A. Latifi, and N. Yazdani, "A stochastic reward net approach to model availability of cloud virtualization," in *Proc. 7th Int. Symp. Telecommun. (IST)*, Sep. 2014, pp. 683–688.

[12] O. C. Ibe, H. Choi, and K. S. Trivedi, "Performance evaluation of client-server systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 11, pp. 1217–1229, Nov. 1993.

[13] M. R. Kelly, *Nielsen Fixed Point Theory on Surfaces*. Dordrecht, The Netherlands: Springer, 2005, pp. 647–658.

[14] H. Xu, L. Luo, X. Qiu, and Y. Xiang, "A performance modeling approach for mobile cloud system," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6, doi: 10.1109/WCNC.2019.8885772.

[15] Y. Kirsal, "Performance modelling and analysis of clustered servers in mobile edge computing," in *Proc. 28th Signal Process. Commun. Appl. Conf. (SIU)*, Oct. 2020, pp. 1–4, doi: 10.1109/SIU49456.2020.9302306.

[16] H. Khazaei, J. Mišić, V. B. Mišić, and S. Rashwand, "Analysis of a pool management scheme for cloud computing centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 5, pp. 849–861, May 2013.

[17] S. Hanczewski and M. Weissenberg, "Concept of an analytical model for cloud computing infrastructure," in *Proc. 11th Int. Symp. Commun. Syst., Netw. Digit. Signal Process. (CSNDSP)*, Jul. 2018, pp. 1–4.

[18] M. Głabowski, S. Hanczewski, M. Stasiak, and J. Weissenberg, "Modeling Erlang's ideal grading with multi-rate BPP traffic," *Math. Problems Eng.*, vol. 2012, Oct. 2012, Art. no. 456910.

[19] S. Hanczewski, M. Stasiak, and M. Weissenberg, "The analytical model of complex non-full-availability system," in *Image Processing and Communications*, M. Choraś and R. S. Choraś, Eds. Cham, Switzerland: Springer, 2020, pp. 279–286.

[20] S. Puhan, D. Panda, and B. K. Mishra, "Energy efficiency for cloud computing applications: A survey on the recent trends and future scopes," in *Proc. Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA)*, Mar. 2020, pp. 1–6.

[21] R. Yadav and W. Zhang, "MeReg: Managing energy-SLA tradeoff for green mobile cloud computing," *Wireless Commun. Mobile Comput.*, vol. 2017, Dec. 2017, Art. no. 6741972.

[22] R. Yadav, W. Zhang, H. Chen, and T. Guo, "MuMs: Energy-aware VM selection scheme for cloud data center," in *Proc. 28th Int. Workshop Database Expert Syst. Appl. (DEXA)*, Aug. 2017, pp. 132–136.

[23] R. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.

[24] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: A packet-level simulator of energy-aware cloud computing data centers," *J. Supercomput.*, vol. 62, no. 3, pp. 1263–1283, 2010.

[25] *Windows Virtual Machines Pricing*. Accessed: Jun. 1, 2021. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/

[26] *PowerEdge R740xd Rack Server*. Accessed: Jun. 1, 2021. [Online]. Available: https://www.dell.com/en-us/work/shop/povw/poweredge-r740xd

[27] J. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. COM-29, no. 10, pp. 1474–1481, Oct. 1981.

[28] J. Roberts, "A service system with heterogeneous user requirements—Application to multi-service telecommunications systems," in *Performance of Data Communications Systems and Their Applications*, G. Pujolle, Ed. Amsterdam, The Netherlands: North Holland, 1981, pp. 423–431.

[29] M. Stasiak, "Blocking probability in a limited-availability group carrying mixture of different multichannel traffic streams," *Annales Télécommun.*, vol. 48, nos. 1–2, pp. 71–76, Jan. 1993.

[30] M. Głąbowski, M. Sobieraj, and M. Stasiak, "Analytical and simulation modeling of limited-availability systems with multi-service sources and bandwidth reservation," *Int. J. Adv. Telecommun.*, vol. 6, nos. 1–2, pp. 1–11, 2013.

[31] T. Bonald and J. Virtamo, "A recursive formula for multirate systems with elastic traffic," *IEEE Commun. Lett.*, vol. 9, no. 8, pp. 753–755, Aug. 2005.

SŁAWOMIR HANCZEWSKI (Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications and the D.Sc. degree in ICT from the Poznan University of Technology, Poland, in 2001, 2006, and 2020, respectively. Since 2001, he has been working with the Poznan University of Technology (currently at the Faculty of Computing and Telecommunications). He is also an Assistant Professor with the Institute of Communications and Computer Networks. He is the author or coauthor of more than 60 scientific articles, related mostly to the analytical modeling of communications systems.

MACIEJ STASIAK (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the Institute of Communications Engineering, Moscow, Russia, in 1979 and 1984, respectively, and the D.Sc. degree in electrical engineering from the Poznan University of Technology. From 1983 to 1992, he worked with the Polish Industry Sector as a Designer of electronic and microprocessor systems. In 2006, he was nominated as a Full Professor. He joined the Poznan University of Technology, where he is currently the Director of the Faculty of Computing and Telecommunications, Institute of Communications and Computer Networks. He is the author or coauthor of more than 300 scientific articles and five books. He is engaged in research and teaching in the area of performance analysis and modeling of queuing systems, multiservice networks, and switching systems. Since 2004, he has been actively carrying out research on the modeling and dimensioning of cellular networks.

MICHAL WEISSENBERG received the M.Sc. degree in electronics and telecommunications from the Faculty of Electronics and Telecommunications, Poznan University of Technology, Poland, in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Computing and Telecommunications, Institute of Communications and Computer Networks, Poznan University of Technology. His research interests include the analytical modeling of communications and cloud systems.