

Article

High-Accuracy Analytical Model for Heterogeneous Cloud Systems with Limited Availability of Physical Machine Resources Based on Markov Chain

Slawomir Hanczewski , Maciej Stasiak  and Michal Weissenberg * 

Faculty of Computing and Telecommunications, Poznan University of Technology, 60-965 Poznan, Poland; slawomir.hanczewski@put.poznan.pl (S.H.); maciej.stasiak@put.poznan.pl (M.S.)

* Correspondence: michal.weissenberg@put.poznan.pl

Abstract: The article presents the results of a study on modeling cloud systems. In this research, the authors developed both analytical and simulation models. System analysis was conducted at the level of virtual machine support, corresponding to Infrastructure as a Service (IaaS). The models assumed that virtual machines of different sizes are offered as part of IaaS, reflecting the heterogeneous nature of modern systems. Additionally, it was assumed that due to limitations in access to physical server resources, only a portion of these resources could be used to create virtual machines. The model is based on Markov chain analysis for state-dependent systems. The system was divided into an external structure, represented by a collection of physical machines, and an internal structure, represented by a single physical machine. The authors developed a novel approach to determine the equivalent traffic, approximating the real traffic appearing at the input of a single physical machine under the assumptions of request distribution. As a result, it was possible to determine the actual request loss probability in the entire system. The results obtained from both models (simulation and analytical) were summarized in common graphs. The studies were related to the actual parameters of commercially offered physical and virtual machines. The conducted research confirmed the high accuracy of the analytical model and its independence from the number of different instances of virtual machines and the number of physical machines. Thus, the model can be used to dimension cloud systems.

Keywords: cloud computing; IaaS; analytical model; Markov chains; system simulation; loss probabilities



Citation: Hanczewski, S.; Stasiak, M.; Weissenberg, M. High-Accuracy Analytical Model for Heterogeneous Cloud Systems with Limited Availability of Physical Machine Resources Based on Markov Chain. *Electronics* **2024**, *13*, 2161. <https://doi.org/10.3390/electronics13112161>

Academic Editor: Mehdi Sookhak

Received: 9 May 2024

Revised: 27 May 2024

Accepted: 30 May 2024

Published: 1 June 2024

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Currently, cloud systems represent one of the key areas within the ICT market. As shown in a report published in September 2023 by Grand View Research [1], the global size of the cloud services market alone could reach more than USD 1550 billion by 2030, which translates into a Compound Annual Growth Rate (CAGR) of 14.1%. This transition is driven by multiple factors, including the move of native applications to the cloud, the demand for storage space, and the migration of physical machines from corporate premises to virtual machines offered by providers as part of an Infrastructure as a Service (IaaS) [2]. IaaS is a solution in which service providers allocate computing power to end users through virtual machines (VMs) hosted on physical machines (PMs) owned by the service provider. There are currently several operators providing such services on the market. Global leaders in this area include Amazon with its AWS service and Microsoft with its Azure service. As of the fourth quarter of 2023, the market share of these players in the cloud services market was more than 50% [3]. Within the solutions of the aforementioned companies, users can purchase access to virtual machines characterized by three main parameters, such as Central Processing Unit (CPU), Random Access Memory (RAM), and Hard Disk Drive (HDD) [4,5].

The recent trend of moving services to the cloud poses several challenges for providers. These include assessing network performance and the ability to provide services at the appropriate quality level (Quality of Service (QoS)) to comply with the service level agreement (SLA) signed with users. Such an assessment can be carried out by utilizing a system model: analytical, simulation, or measurement-based. Analytical models are frequently regarded as the quickest means to assess how the approval of a new request will influence system performance. Many models of such systems working according to the IaaS model can be found in the literature. The authors of the models proposed in [6–8] conducted their tests for homogeneous virtual machines described by a single resource request parameter from the system. Homogeneity of machines means that all requests to create a virtual machine that appear on the system's input require an equal number of system resources. In contrast, the authors of [9,10] proposed an approach in which virtual machines are also described by a single parameter, but their request is heterogeneous, i.e., virtual machines differ in size. Consequently, the input requests to create virtual machines may require different numbers of resources from the system. Multiparametric resource is more often used when trying to determine the utilization of individual system resources [11–19]. Despite the practice adopted by service providers to describe machines with three parameters (CPUs, RAM, HDD), describing virtual machines for creating analytical models using a single resource requested from the system has its justification. In the process of careful analysis of typical VMs offered in different service characteristics offered by both Amazon and Microsoft, it can be seen that successive VM parameters in each range are their multiples. For example, the parameters of virtual machines offered by Microsoft Azure type **D2ads–D8ads v5** [4] are summarized in Table 1.

Table 1. Parameters of virtual machines type **D2ads–D64ads v5** offered by Microsoft [4].

Instance	vCPU(s)	vCPU(s) [AU]	RAM [GB]	RAM [AU]	Storage [GB]	Storage [AU]	VM [AU]
D2ads v5	2	1	8	1	75	1	1
D4ads v5	4	2	16	2	150	2	2
D8ads v5	8	4	32	4	300	4	4

For the purpose of performing research using an analytical model, a process is required to discretize and express requests in dimensionless units of resource allocation [AU—Allocation Unit]. These are determined as the smallest common divisor of individual requests. In the case of the **D2ads–D8ads v5** VMs offered by Microsoft, for vCPU(s), a single allocation unit can correspond to 2 CPU cores, as well as 8 GB of RAM and 75 GB of disk space. The requests expressed in AUs for each type of VM are also shown in Table 1. It is apparent that after the discretization process, the individual requests for VMs are the same. Therefore, they can be reduced to a single parameter, and for the purpose of calculation, a VM can be represented as a single value (VM column, Table 1). Therefore, the primary research objective was to develop an approximate analytical model with high accuracy for a heterogeneous cloud system. In this system, requests to create a virtual machine occurring at the input require certain resources of a single physical machine, which results in limited availability in both the external and internal structure.

However, if scaling was not possible in the used system, it would be necessary to consider a multiparametric system. A model for such a system, where VMs can be described by four independent parameters, has been developed and published in [18,19].

1.1. Related Works

In the models proposed in this article, in contrast to the publications [6–10,20,21], an additional phenomenon related to resource constraints on individual physical machines has been taken into account. The authors considered the case in which a request to create a virtual machine appearing at the input to the system does not have access to all of

the physical machine's resources, but only a limited part of them. The limited internal accessibility may be due to several factors, such as offering certain services only on a part of the system, requiring access to specific memory segments, varying transmission speeds on different disks, or the internal architecture of the physical machine. So far, the authors have proposed two models for cloud systems with limited accessibility to physical resources, published in [22,23].

To summarize the approaches of other authors in this area, it is worth mentioning that in prior work, the authors in [20] introduced an analytical model utilizing Stochastic Reward Nets (SRNs) to characterize and assess an IaaS cloud system. Initially, an SRN was employed to represent a cluster of physical machines managed by MGM, later extending to analyze a group of machines governed by a GM. This two-stage approach facilitated the creation of a comprehensive monolithic model depicting the entire IaaS cloud. Recognizing scalability challenges with the monolithic model, the authors proposed two approximate SRN models leveraging folding and fixed-point iteration techniques. These models enabled the evaluation of performance, availability, and power consumption of the IaaS cloud. Concurrently, in [21], a model was devised to predict the quality of service parameters such as average queue length and throughput, utilizing an open queuing model and the exact spectral expansion method for parameter derivation. Furthermore, ref. [9] presented an analytical model of an IaaS infrastructure incorporating variable activation requirements for new virtual machines based on processor count. This heterogeneous model utilized hierarchical analysis of birth and death processes under the assumption of server homogeneity. Ref. [10] introduced a hierarchical architecture with subsystem decomposition for loss probability determination, employing fixed-point methodology to ascertain total system losses. Additionally, ref. [8] employed the $M/G/m/m + K$ model to evaluate active VM performance in IaaS clouds, providing insight into task distribution and quality parameters such as average response time. Notably, ref. [7] also proposed a scalable SRN-based model for large-scale cloud systems, facilitating the determination of utilization, availability, waiting time, and response speed parameters. Lastly, refs. [22,23] proposed two models for single-parametric heterogeneous cloud systems, relying on established analytical models of under-resourced systems and macro-level Markov chain analysis. However, these models demonstrated lower accuracy in comparison with the approach presented in this article.

1.2. Research Contribution

The main achievements of this article are as follows:

- Development of a heterogeneous analytical model for cloud systems with a single-parameter description of virtual machines and limited internal availability of resources in physical machines;
- The proposal of a novel approach to determining the probability of loss in a complex system, significantly distinct from the methodologies outlined in articles [3–12];
- Development of the simulation model of the cloud system;
- Conducting extensive simulation and analytical studies to confirm the accuracy of the proposed analytical model.

The article is organized in the following manner: Section 2 presents the characteristics of heterogeneous cloud systems with limited availability. In turn, Section 3 discusses the analytical model of the cloud system. Section 4 presents the simulation model of the considered system. Subsequently, Section 5 is devoted to a discussion of example results, including results for a system with parameters corresponding to real systems (System I). Section 6 presents the article's conclusions.

2. Characteristics of Heterogeneous Cloud Systems with Limited Availability

The cloud system model is based on the idea shown in Figure 1. In this model, the first step in initiating a new virtual machine by request involves directing the request to a device known as the Group Manager (GM). This device is responsible for controlling a set of physical machines and activating a virtual machine on one of them in accordance with

the adopted distribution algorithm. In their study, the authors adopted an even distribution of virtual machines as the distribution algorithm throughout the system, which results in loss minimization. Therefore, the case under consideration aims to provide optimization from the point of view of traffic efficiency in the cloud system.

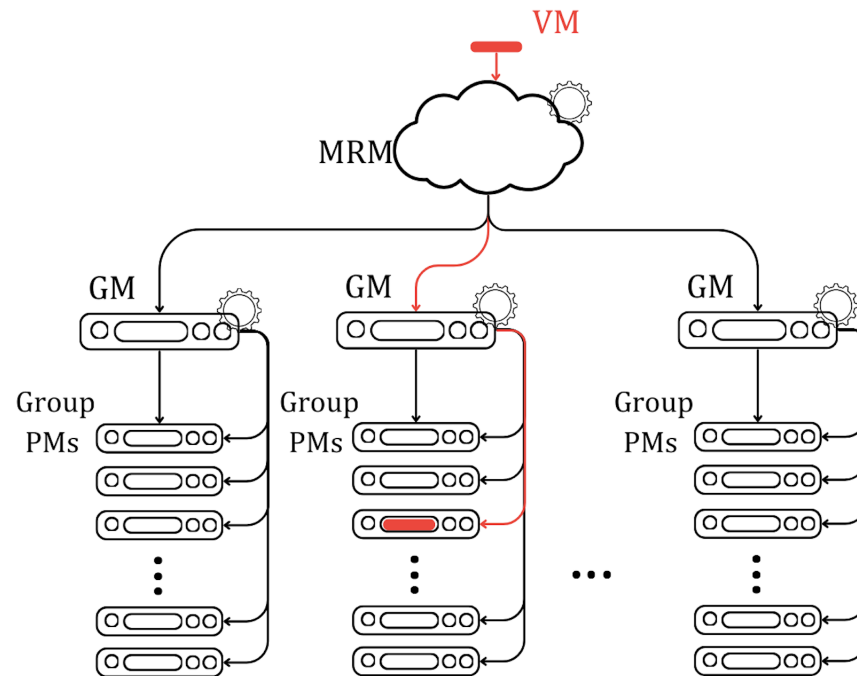


Figure 1. The idea behind the working mechanism of the system.

Due to the assumption that both the demand for creating a VM and the capacity of physical machines can be described by a single parameter, the capacity of the system is referred to as C_V . At the same time, the demand for the creation of a new VM of class i can be described as c_i , understood as a group of machines requiring an identical number of physical machine resources for their creation. Consequently, the different types of instances shown in Table 1 can be treated as a class of requests. The number of VM classes that can be run on the system is equal to m . When the GM receives a new request to create a VM, it attempts to select a PM to activate the machine. The number of machines that can be activated on a single PM depends on the physical resources of the PM and the request describing the VM. It is assumed that a new request to create a VM of class i can only be accepted for service if at least one PM at any given time has free resources that meet the c_i requirements of the VM. However, these resources must be limited to a specifically defined part, rather than the entirety of resources available on the machine. The number of resources accessible by a VM can be determined by the availability parameter d_i . This parameter specifies the number of physical machine resources that the virtual machine can access. The requirement of specific resources (i.e., available resources) from the physical machine may have to do with the internal design of the system, for example, the requirement of a specific area in the disk space or offering the service only in a given part of the system (the speed of one of the disks, the speed of a specific processor, etc.). The model also assumes that the influx of requests to the system is Poisson in nature, with the number of physical machines determined by the parameter k , and that all physical machines controlled by the GM have the same capacity $\frac{C_V}{k}$. If there are no free resources on any of the k physical machines in the system in the required resource group, the request is rejected and considered lost.

3. Analytical Model

The analytical model proposed in the article makes it possible to determine the loss (the ratio of the number of rejected requests to create a VM in the cloud system to all requests of

this type that occur) for different classes of requests in the cloud system. These requests differ in the number of system resources requested and the intensity of the influx, as well as the number of resources available in each physical machine. To determine the probability of loss, the authors used models known from the literature, i.e., a limited-availability system and an Erlang's ideal system.

3.1. Limited-Availability System

A limited-availability system (LAS) is a system built of k identical subsystems with full availability. A diagram of such a system is shown in Figure 2. The capacity of each subsystem is equal to $C_S = \frac{C_{LAS}}{k}$, where C_{LAS} is the capacity of the entire system. The system is offered m classes of requests, each of which requires c_i AUs ($0 \leq i \leq m$) to be handled.

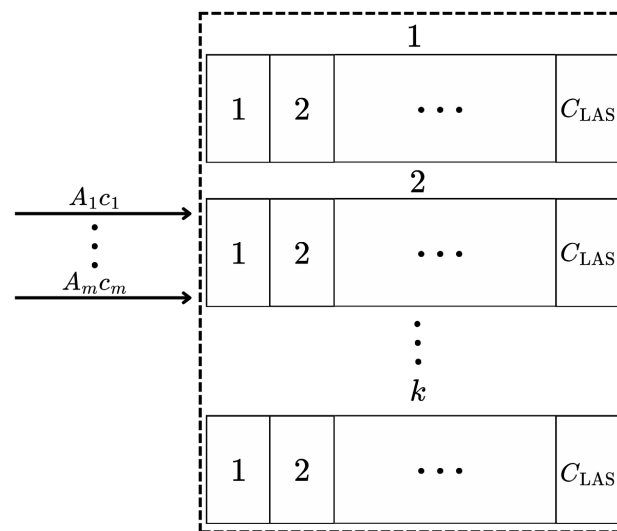


Figure 2. Illustration of LAS model structure.

Request class i appearing at the input to the system can be handled in the system if and only if at least one of the subsystems has enough free resources to handle it, i.e., c_i . Consequently, the LAS excludes the possibility of splitting the c_i AUS request required by the class i request among different subsystems. Analytical models for such systems have been proposed in [24]. According to these models, the distribution of occupancy in LAS can be determined as follows:

$$n[P_{LAS}(n)]_{C_{LAS}} = \sum_{i=1}^m k A_{LAS,i} c_i \sigma_{LAS,i}(n - c_i) [P(n - c_i)]_{C_{LAS}}, \quad (1)$$

where

- $A_{LAS,i}$ is the offered traffic intensity of class i ;
- $[P_{LAS}(n)]_{C_{LAS}}$ is the occupancy probability of n AUs in LAS with a total capacity of C_{LAS} allocation units;
- $\sigma_{LAS,i}(n)$ is the so-called conditional transition coefficient between neighboring occupancy states in LAS:

$$\sigma_{LAS,i}(n) = 1 - \frac{F(C_{LAS} - n, k, c_i - 1)}{F(C_{LAS} - n, k, C_S)}, \quad (2)$$

where $F(x, k, c)$ is the number of possible distributions of x free (unoccupied) AUs in k subsystems, where each of the subsystems has a capacity of $C_S = \frac{C_{LAS}}{k}$ AUs:

$$F(x, k, C_S) = \sum_{i=0}^{\lfloor \frac{x}{C_S+1} \rfloor} (-1)^i \binom{k}{i} \binom{x+k-1-i(C_S+1)}{k-1}. \quad (3)$$

By knowing the occupancy distribution, it becomes feasible to determine the loss probability for each request class serviced by the system under the formula

$$E_i = \sum_{n=0}^{kC_S} [1 - \sigma_{\text{LAS},i}(n)] [P_{\text{LAS}}(n)]_{kC_S}. \quad (4)$$

In a simplified manner, the results of LAS modeling can be symbolically represented as follows:

$$\{\mathbf{P}, \mathbf{E}\} = \text{LAS}(\mathbf{A}, \mathbf{c}, k, C_S), \quad (5)$$

where $\mathbf{P}$ represents the occupancy distribution in LAS, obtained based on (1), and $\mathbf{E}$ is a set of loss probabilities (4), while $\mathbf{A}$ and $\mathbf{c}$ are sets of offered traffic and its demands (Formulas (6) and (7)).

$$\mathbf{A}_{\text{LAS}} = \{A_{\text{LAS},i} : 1 \leq i \leq m\}, \quad (6)$$

$$\mathbf{c} = \{c_i : 1 \leq i \leq m\}. \quad (7)$$

The LAS model approximates the external structure of a cloud system, consisting of k physical machines with equal capacities. These are offered a set of virtual machines with different requirements, expressed in a dedicated class number of AUs.

3.2. Erlang Ideal System

An Erlang's ideal system (EIS) is a certain abstract system in which requests of class i ($1 \leq i \leq m$) arriving at its input do not have access to all of the C_{EIS} of the system, but only a limited portion of it, called d_i availability (Figure 3). The concept and analytical model for single-service traffic was proposed by Erlang back in the early 20th century [25]. This model for multi-service traffic was generalized in the works [26,27]. In [26], it was assumed that all classes of traffic have identical availability. In [27], on the other hand, the assumption was that requests from different classes have access to resources with different values, i.e., they have different accessibility.

Requests that access the same system resources are called a load group. The number of load groups in an EIS system (g_i), for request class i ($1 \leq i \leq m$), can be calculated using the following formula:

$$g_i = \binom{C_{\text{EIS}}}{d_i} = \frac{C_{\text{EIS}}!}{(C_{\text{EIS}} - d_i)! d_i!}. \quad (8)$$

The availability parameter is defined for each class of requests in the system. Its value depends on the structure of the system, the mechanism for controlling access to resources, or specific requirements for the resources themselves. It is worth noting that within a single class of requests, understood as requests of the same size (i.e., the required number of c_i AUs ($1 \leq i \leq m$) and value of availability parameter d_i), not all requests necessarily receive access to the same system AUs. Consequently, the demands of a class can be offered in different load groups. Figure 3 illustrates a very simple example of such a situation—a single-service system (occurring in a system of 1 class of requests and requiring a single resource for service) with a capacity of 3 AUs and an availability of $d = 2$ AUs. In the situation shown, according to the Formula (8), 3 load groups can be distinguished, having access to the 1st and the 2nd resource in the system, the 2nd and the 3rd resource in the system, and the 1st and the 3rd resource in the system, respectively. The figure shows the blocked state for the third load group, because even though there is one resource free in

the system (the 2nd place in the system), this request does not have access to it and must be rejected.

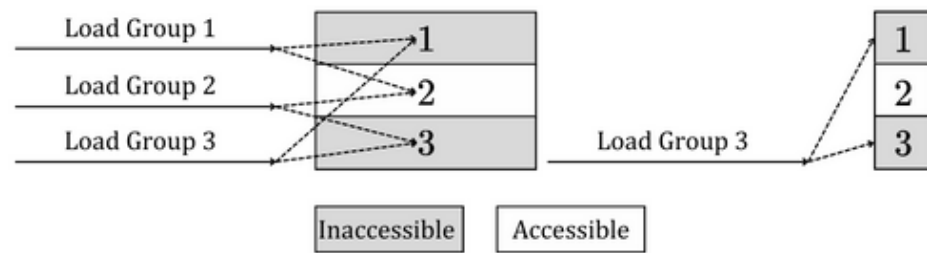


Figure 3. Illustration of the design of a single-service EIS model with a capacity of 3 and an availability of 2.

Analogically to LAS, sufficient resources throughout the system do not guarantee that a request will be accepted for service. An incoming request of class i will be accepted for service if and only if a sufficient number of resources c_i are free within the load group specified for that request. Given this assumption, the conditional transition coefficient in the EIS model can be determined as follows:

$$\sigma_{\text{EIS},i}(n) = 1 - \sum_{d_i - c_i + 1}^z \frac{\binom{d_i}{x} \binom{C_{\text{EIS}} - d_i}{n - x}}{\binom{C_{\text{EIS}}}{n}}, \quad (9)$$

where

- $z = n - c_i$, if $(d_i - c_i + 1) \leq (n - c_i) \leq d_i$,
- $z = d_i$, if $(n - c_i) \geq d_i$.

After determining the conditional transition coefficient between states, the occupancy distribution can be determined using the equation

$$n[P_{\text{EIS}}(n)]_{C_{\text{EIS}}} = \sum_{i=1}^m A_{\text{EIS},i} \sigma_{\text{EIS},i}(n - c_i) [P_{\text{EIS}}(n - c_i)]_{C_{\text{EIS}}}. \quad (10)$$

Once the conditional probability of transition between adjacent states is determined, the occupancy distribution can be determined by the formula

$$E_{\text{EIS},i} = \sum_{n=0}^{C_{\text{EIS}}} [1 - \sigma_{\text{EIS},i}(n)] [P_{\text{EIS}}(n)]_{C_{\text{EIS}}}. \quad (11)$$

In this article, the EIS model was used to approximate the internal structure of the cloud system, in which the upcoming requests (to create MVs of a given class) at the input of a single physical machine have access to a uniquely defined part of the resources of this machine.

To simplify the notation, the results of the EIS model can symbolically be represented as follows:

$$\{\mathbf{P}_{\text{EIS}}, \mathbf{E}_{\text{EIS}}\} = \text{EIS}(\mathbf{A}_{\text{EIS}}, \mathbf{c}, C_{\text{EIS}}), \quad (12)$$

where $\mathbf{P}$ represents the occupancy distribution in the EIS, obtained from an equation analogous to that obtained with Equation (10), while $\mathbf{E}$ is the set of loss probabilities obtained using Equation (11). While $\mathbf{A}_{\text{EIS}}$, $\mathbf{c}$ and $\mathbf{d}$ are sets of offered traffic and its requests and availability:

$$\mathbf{A}_{\text{EIS}} = \{A_{\text{EIS},i} : 1 \leq i \leq m\}, \quad (13)$$

$$\mathbf{c} = \{c_i : 1 \leq i \leq m\}, \quad (14)$$

$$\mathbf{d} = \{d_i, 1 \leq i \leq m\}. \quad (15)$$

The analytical models presented in Sections 3.1 and 3.2 are well known in the literature of the subject and have been successfully used for years for the analysis, design and optimization of ICT systems. These models are used to analyze switching fields [28], physical cloud computing infrastructure [19] or prioritized systems [29]. Regardless of the current work on the systems under consideration, the books [30,31] provide a coherent summary of knowledge on these models and their applications.

3.3. Model L-E-CBP

In articles [22,23], two analytical models of the cloud system were proposed, based on a product-based approach to determine the conditional transition coefficient in systems in which state dependence is influenced by several independent factors [32]. In the model, the cloud system was treated as a unified system (i.e., a system without division into external and internal structures) with a productively determined conditional transition probability. The disadvantage of this approach is the necessity to scale the occupancy state in a single physical machine to the occupancy state in the entire system, which consequently negatively affects the accuracy of the adopted solution. In order to improve the accuracy of analytical modeling of cloud systems, the paper proposes a new model L-E-CBP (LAS-EIS-Cloud Blocking Probability). The authors describe model using the symbols and acronyms listed in Table 2.

Table 2. Symbols and corresponding descriptions.

Symbol	Description
PM	Physical Machine
VM	Virtual Machine
AU	allocation unit
LAS	limited-availability system
EIS	Erlang's ideal system
CBP	cloud blocking probability
k	number of PMs in system
$A_{LAS,i}$	the offered traffic intensity of class i in LAS model (on the input of the entire cloud system)
$[P_{LAS}(n)]_{kC_{EIS}}$	the occupancy probability of n AUs in LAS with a total capacity of kC_{EIS} AUs
$\sigma_{LAS,i}(n)$	conditional transition coefficient between neighboring occupancy states in LAS for class i in occupancy state n
C_{EIS}	size of a single PM (capacity of the EIS system)
C_{LAS}	the entire capacity of the cloud system built of k PMs (capacity of the LAS system equal to kC_{EIS})
$E_{LAS,i}$	loss probability of class i in LAS
$[P_{EIS}(n)]_{C_{EIS}}$	the occupancy probability of n AUs in EIS with a total capacity of C_{EIS} AUs
$\sigma_{EIS,i}(n)$	conditional transition coefficient between neighboring occupancy states in EIS for class i in occupancy state n
$E_{EIS,i}$	loss probability of class i in EIS
d_i	availability parameter of class i in EIS
$A_{EIS,i}^*$	equivalent traffic of request of class i at the input of the single PM
$E_{CBP,i}$	total loss probability of class i requests in the cloud system

The model assumes that the external structure of the cloud system (a system of physical machines) is modeled based on LAS, while the internal structure (a single virtual machine) is modeled based on EIS. First, the L-E-CBP model determines the loss probability resulting from the external structure of the cloud system, i.e., a system consisting of k independent physical machines with identical parameters. In these physical machines, a request for a virtual machine creation can be serviced only if there are free resources in at least one physical machine of the system to support the virtual machine completely, i.e., without participating in the support of other physical machines. Then, assuming that the algorithms for allocating resources to virtual machines strive to load all physical machines equally, it can be assumed that the traffic offered to individual physical machines is identical and results from the division of traffic flowing down from the external structure (i.e., traffic that has not been lost in the external structure) into k traffic streams of identical intensity. In the article, such traffic is called equivalent traffic and is the basis for modeling the internal structure of the cloud system (a single physical machine) based on the EIS model. In the next step, the actual loss probability of the cloud system is determined, taking into account both the availability of a single physical machine and the number of physical machines in the system. Figure 4 shows the concept of traffic distribution in a cloud system, according to the proposed L-E-CBP model.

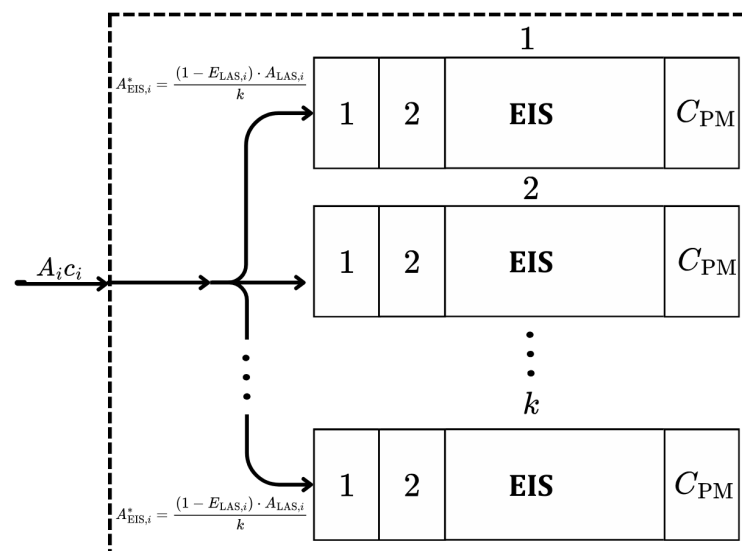


Figure 4. Illustration of the L-E-CBP model concept.

In the proposed L-E-CBP model, the probability of loss of class i , resulting from the external structure of the cloud system, is approximated by a LAS model consisting of k subsystems, which in turn are EIS systems with a capacity of $C_S = C_{EIS} = \frac{C_{LAS}}{k}$ AUs. Thus, the probability of loss in the external structure of the system, expressed by Equation (4), can be rewritten in the adopted notation as follows:

$$E_{LAS,i} = \sum_{n=0}^{kC_{EIS}} [1 - \sigma_{LAS,i}(n)] [P_{LAS}(n)]_{kC_{EIS}}. \quad (16)$$

Simultaneously, the probability of loss due to the internal configuration of the system is approximated by the EIS model, consisting of C_{EIS} AUs, where requests of class i have access to a certain portion of the physical machine's resources determined by the value of the availability parameter d_i . This probability of loss is given by the Equation (11).

Let us consider a cloud system with a total capacity of C_{LAS} providing access to k PMs. Note that the external structure's probability of loss $E_{LAS} = LAS(\mathbf{A}_{LAS}, \mathbf{c}, k, C_{EIS})$ is determined by the traffic offered to the entire cloud system, i.e., the A_{LAS} traffic. On the other hand, the probability of internal structure loss $E_{EIS} = EIS(\mathbf{A}_{EIS}^*, \mathbf{c}, \mathbf{d}, C_{EIS})$ is determined from the equivalent traffic sets $\mathbf{A}_{EIS}^*$, where each element of $A_{EIS,i}^*$ is defined.

$$A_{EIS,i}^* = \frac{(1 - E_{LAS,i}) \cdot A_{LAS,i}}{k}. \quad (17)$$

Thus defined equivalent traffic—offered to a single internal structure—is traffic that is not lost in the external structure. The total probability of loss in the cloud system E_{CBP} , taking into account losses in the external structure and k internal structures in the L-E-CBP model, is approximated by the probability of traffic loss. The probability of traffic loss is defined as the ratio of traffic lost to traffic offered to a given system. Traffic lost of class i in an external structure has a value of $A_{LAS,i}E_{LAS,i}$. Traffic lost in a single internal structure (physical machine) is equal to $A_{EIS,i}^*E_{EIS,i}$. Thus, the total class and lost traffic in an external structure and k internal structures is equal to $A_{LAS,i}E_{LAS,i} + kA_{EIS,i}^*E_{EIS,i}$. Now, based on the definition of traffic loss probability, the total loss probability of class i requests in the cloud system can be approximated as follows:

$$E_{CBP,i} = \frac{[A_{LAS,i}E_{LAS,i} + kA_{EIS,i}^*(E_{EIS,i})^k]}{A_{LAS,i}}. \quad (18)$$

The values of $E_{CBP,i}$ are the elements of the sets $\mathbf{E}_{CBP}$ covering loss probabilities for all classes of traffic offered to the cloud system.

The determination of cloud system loss probabilities based on the L-E-CBP model can be written in the form of the L-E-CBP METHOD:

L-E-CBP METHOD:

1. Determine, according to the LAS model, the occupancy distribution and the loss probability in the external structure of the system according to the formula $\{\mathbf{P}_{LAS}, \mathbf{E}_{LAS}\} = LAS(\mathbf{A}_{LAS}, \mathbf{c}, k, C_{EIS})$.
2. Determine the equivalent traffic—based on Formula (17)—appearing at the input of a single physical machine:

$$\mathbf{A}_{EIS}^* = \{A_{EIS,i}^* : A_{EIS,i}^* = \frac{(1 - E_{LAS,i})A_{LAS,i}}{k} \wedge 1 \leq i \leq m\}.$$

3. Determine the occupancy distribution and the loss probability in a single physical machine of capacity $C_{EIS} = \frac{C_{LAS}}{k}$ at the input of which the equivalent traffic offered determined in Step 2 appears. Based on the EIS model, it can be written as follows: $\{\mathbf{P}_{EIS}, \mathbf{E}_{EIS}\} = EIS(\mathbf{A}_{EIS}^*, \mathbf{c}, \mathbf{d}, C_{EIS})$.
4. Determine the total losses in the cloud system based on the Formula (18):

$$\mathbf{E}_{CBP} = \{E_{CBP,i} : E_{CBP,i} = \frac{[A_{LAS,i}E_{LAS,i} + kA_{EIS,i}^*(E_{EIS,i})^k]}{A_{LAS,i}} \wedge 1 \leq i \leq m\}.$$

4. Simulation Model

As part of the research on complex cloud systems, a simulation model of such a system was developed and implemented [22,23], which was appropriately modified for the research presented in this article. The simulator, developed in C++, allows for the analysis of the probability of rejection of requests to create a new virtual machine for particular classes of virtual machines. VM classes represent types of virtual machines of different sizes and are interpreted in the simulator as classes of traffic offered to the cloud system with appropriate parameters. The program was divided into modules responsible for carrying out the various tasks of the simulator, and their interrelationships are shown in Figure 5.

- Input data are used to enter basic parameters into the system, such as the type and number of request classes along with their size and availability, system parameters

(external and internal structure) along with capacities, and the definition of the queue in the system along with the management algorithm;

- Requests are used to initialize the various classes of submissions, along with their basic handling implemented in the distinguished methods, depending on the adopted parameters (type of traffic (BPP), type of traffic elastic, adaptive, streaming);
- System is used to initialize the structure of the system (number of physical machines, availability in their area) and methods for handling servers;
- Simulation is used to initiate and control the process of the simulation being carried out according to the adopted scenario;
- Events are used to describe and execute timed and conditional events occurring in the system;
- Events_List is used to store events occurring in the system and their handling according to the order of occurrence.
- Generators are used to generate distributions and random numbers;
- Statistics is used to determine statistical data related to quality parameters of the system;
- Output_Data are used to save the statistical data resulting from the carried out digital simulation process in output files.

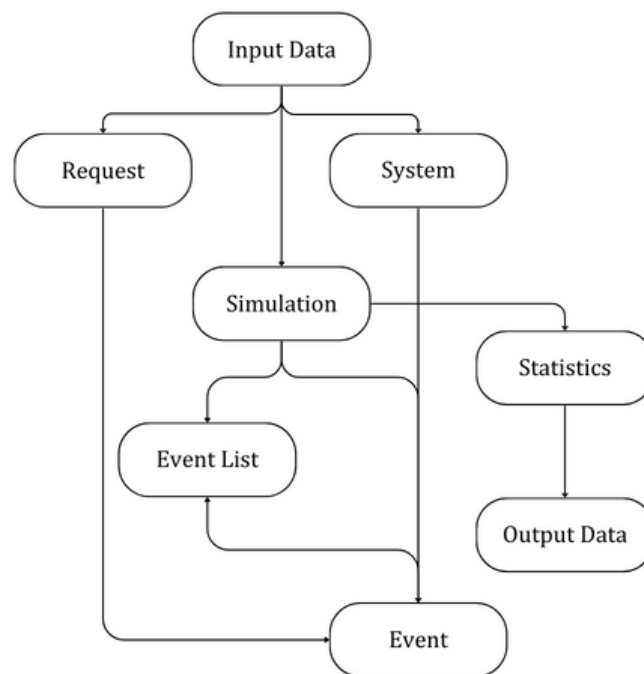


Figure 5. The block diagram of the simulation model operation.

The simulation process assumed a uniform identical capacity for all k physical machines, expressed in adopted AUs and equal to C_{PM} . Thus, the total capacity of the cloud system is $C = kC_{PM}$. Each traffic class is characterized by the following parameters:

- λ_i —average intensity of class i requests;
- μ_i —average intensity of handling a class i request;
- c_i —the average number of AUs required to serve a request of class i ;
- d_i —availability for class i requests, i.e., the number of resources, expressed in AUs, in which a given request can be fulfilled.

In the simulator, the input parameter for traffic is the average traffic offered to a single AU of the cloud system a_{AU} , which is defined as follows:

$$a_{AU} = \frac{\sum_{i=1}^m \lambda_i c_i}{C}. \quad (19)$$

The proportion of offered traffic $x_1 : x_2 : \dots : x_m$, determining their share in the aggregate traffic stream:

$$x_1 : x_2 : \dots : x_m = A_1 c_1 : A_2 c_2 : \dots : A_m c_m. \quad (20)$$

Based on (19) and (20), the simulator determines the values of the offered traffic of each class:

$$A_i = \frac{a_{AU} C x_i}{\sum_{n=1}^m x_n c_n}. \quad (21)$$

Then, this allows you to determine the request intensity of each class of λ_i requests. Assuming that the offered traffic is Erlang traffic, the stream of requests is a Poisson stream of intensities:

$$\lambda_i = \frac{A_i}{\mu_i}. \quad (22)$$

The parameters λ_i and μ_i allow for the use of random number generators with an exponential distribution, necessary to simulate the process of appearing and handling new requests. The authors implemented pseudo-random number generators in the simulation model. The primary generator implemented was a uniform distribution generator, which formed the basis for the other generators. For this purpose, a linear multiplicative generator, proposed in the work of [33], was developed, operating according to the formula:

$$X_{n+1} = (aX_n) \% (z), \quad (23)$$

where

- $a = 16,807$;
- $z = 2^{31} - 1$.

The developed generator became the basis of an exponential number generator essential in analyzing systems with Erlang traffic. The exponential distribution was obtained according to the method of inverting the distribution (inverse method). Since an exponential distribution is described by the formula:

$$\begin{aligned} F(X) &= 0 \quad \forall \quad x < 0, \\ F(X) &= 1 - e^{-\lambda x} \quad \forall \quad (\lambda > 0 \wedge x \geq 0), \end{aligned} \quad (24)$$

it is possible to determine the inverse distribution of the distribution for $x \geq 0$ underlying the exponential distribution:

$$X = -\frac{1}{\lambda} \ln(1 - u), \quad (25)$$

where

- X represents the next designated random number with an exponential distribution;
- u represents the next value determined by the exponential generator.

4.1. General Simulation Algorithm

The developed discrete-time continuous simulation model was implemented in C++ using the event scheduling method [34]. The general algorithm according to which the simulation program works can be presented in the form of the following steps [34]:

1. Initial configuration of the simulation model—creating all sources generating requests of individual traffic classes, setting input parameters.
2. Initializing the model by activating traffic sources and placing events (appearance of requests) in the event list.
3. Checking the end condition of the simulation. If the end condition is met, the simulation is terminated and the results are saved.
4. Assigning the system time to the time of the first event in the list.
5. Executing the first event in the event list.
6. Removing the first event from the list and returning to Step 3.

4.2. Handling Events in the System

In the simulation model of a cloud system with a complex limited availability structure, two time-related events are defined: *the arrival of a new request* and *the completion of the request handling*. According to the event scheduling method, each event is represented by a separate child class of the Events class and executed by methods implemented in them. The described approach allows the system to define many different Erlang-type traffic classes.

In the conducted research, m traffic classes of the Erlang type were defined. The function that implements events related to the arrival of a new request can be described as follows:

1. Schedule the appearance of a new class i request based on a random number generator with an exponential distribution whose parameter is the intensity of λ_i . Placing the event in the event list.
2. Determine d system resources to which the new request has access, according to the d_i accessibility adopted for the class.
3. Determine the ability to handle a new request, i.e., whether the system has sufficient resources to handle a new request.
 - (a) Check whether any of the k physical machines have a minimum of c_i free AUs within the designated d_i available AUs. If not, the request is lost due to the cloud system's external structure blockage.
 - (b) If there is more than one physical machine with a sufficient number of AUs, the physical machine that will handle the new request is selected based on a random number generator with a uniform distribution.
4. Seizure of resources required by the i class request.
5. Schedule the termination of the handling of a new request based on a random number generator with an exponential distribution whose parameter is the intensity of the handling according to the exponential distribution μ_i . Place the event in the event list.

The function that implements the request service termination event can be described as follows:

1. Checking which physical machine was used to handle the request.
2. Checking which physical machine resources were used to handle the request.
3. Releasing the resources of the physical machine and removing the object representing the request from the system.

The simulator that was used to verify the results obtained with the analytical model has a high degree of scalability and the ability to take into account various scenarios for cloud systems, which makes it possible to test analytical models for very extreme cases. The simulation environment allows for determining the number and size of physical machines, as well as the distribution of available resources into those required by each request, taking into account different disks, different processors, etc. In addition, it is possible to use different types of distribution of incoming traffic and its division into different classes of requests understood as different virtual machines. Since the model was developed by the article's authors, it remains a closed solution and is not as widely used by the scientific community as well-known simulators like CloudSim. in Ref. [35] or GreenCloud [36]. This provides some constraints on the addition of new libraries by the research community. At the same time, it provides full control and reliability from the authors' point of view and an understanding of how the system works according to the assumptions made, which translates into a more complete and reliable ability to test the analytical model.

4.3. End of Simulation Condition

In determining the probability of loss, the condition for the end of the simulation experiment is counting the appropriate number of generated requests of the least active class (usually the class with the highest number of requested AUs). The average result is calculated based on 10 series. In practice, to obtain a 95% confidence interval of no more

than 5% of the average value of the results obtained from the simulation experiments, about 750,000 notifications of the least active class should be generated.

5. Numerical Results

In this section, the results obtained from the proposed analytical model are compared with the results of the corresponding simulation experiments. To determine the parameters of the tested systems, a comprehensive analysis of both physical and virtual machine resources available in the market was conducted, including an analysis of the available virtual machines offered by vendors such as Microsoft and Amazon. It was decided to choose machines of types **D2ads–D8ads v5** offered within the Azure platform. The parameters of typically offered machines are delineated in Table 1. As discernible, the subsequent parameters of the virtual machines within each range are multiples thereof. This trend is prevalent across most available machine types within both AWS and Azure solutions. SuperMicro AMD EPY A+ Server 4124GS-TNR servers were considered for this purpose. For the sake of deliberation, the hardware configuration depicted in Table 3 was adopted.

Table 3. Hardware configuration of servers.

Server	CPU(s)	RAM	Storage
Supermicro	2 × Milan 7453	2 × 128 GB	1 × Samsung PM883
AMD EPY A+	DP/UP 28C/56T	DDR4-3200 MHz	1.92 TB SATA 6 Gb/s
Server 4124GS-TNR	2.75 G 64 M 225 W	4R × 4 ECC RDIMM	V4 TLC 2.5'' 7 mm

It is also assumed that to create virtual machines for users, 50 CPU cores, and 200 GB of RAM are used on each physical machine, and 1875 GB of disk space is reserved. The remaining server resources are reserved for device management and control operations. The system assumes the use of three servers simultaneously, whereby a virtual machine can be created if and only if a sufficient number of resources are available on a single server in the area reserved for it, based on availability. Taking into account the adopted AU values for individual resources, the capacity of servers can be expressed in AUs according to Table 4.

Table 4. Hardware configuration of servers in AUs.

Server	CPU(s) [AUs]	RAM [AUs]	Storage [AUs]
Supermicro			
AMD EPY A+	25	25	25
Server 4124GS-TNR			

A VM of a given class is treated as a multiplication of the VM with the smallest requests, which in turn defines one allocation unit ($c_1 = 1$ AU). It is associated with a requirement of two processor cores, 8 GB of RAM, and 75 GB of disk space (Table 1). In the analyzed system, named System I, the following classes of requests (VM allocations) are still supported: $c_2 = 2$ AUs, $c_3 = 4$ AUs, with the capacity of a single server being $C = 25$ AUs. The final parameters of System I are presented in Table 5.

The authors also conducted a number of other studies, taking into account scenarios which demand that the requested AUs are not their consecutive multiples, as well as situations in which the availability for different classes of requests varies. In order to present these scenarios, the authors have included in the article two additional systems (System II and System III), whose parameters are shown in Table 6.

Table 5. Parameters of a System 1.

System 1		
No. of PMs	PM Capacity	
$k = 3$	$C_{PM} = 25$ AU	
VMs demands		
VM class 1	$c_1 = 1$ AUs	$d_1 = 15$ AUs
VM class 2	$c_2 = 2$ AUs	$d_2 = 15$ AUs
VM class 3	$c_3 = 4$ AUs	$d_3 = 15$ AUs

Table 6. Parameters of a Systems 2, 3, and 4.

System 2		
No. of PMs	PM Capacity	
$k = 3$	$C_{\text{PM}} = 25$ AUs	
VMs demands		
VM class 1	$c_1 = 1$ AUs	$d_1 = 15$ AUs
VM class 2	$c_2 = 2$ AUs	$d_2 = 15$ AUs
VM class 3	$c_3 = 3$ AUs	$d_3 = 15$ AUs
VM class 4	$c_4 = 4$ AUs	$d_4 = 15$ AUs
System 3		
No. of PMs	PM Capacity	
$k = 3$	$C_{\text{PM}} = 18$ AUs	
VMs demands		
VM class 1	$c_1 = 1$ AUs	$d_1 = 5$ AUs
VM class 2	$c_2 = 2$ AUs	$d_2 = 8$ AUs
VM class 3	$c_3 = 4$ AUs	$d_3 = 12$ AUs

In all of the considered scenarios, it was assumed that the total offered traffic was divided among the different classes of requests in the following proportions: $A_1c_1 : A_2c_2 : \dots : A_m c_m = 1 : 1 : \dots : 1$. In each simulation experiment, to determine a single measurement, 10 series of simulations were run, each with 800,000 requests of the class that required the largest number of AUs to handle. The results are plotted as a function of the traffic offered to a single AU available in the cloud system. Figures 6–8 show the loss probabilities obtained for three different systems (see Tables 5 and 6).

The proposed analytical model constitutes an approximation. Yet, it demonstrates commendable accuracy in capturing the dynamics of single-parameter cloud systems, whereby incoming requests for physical machine allocation are confined to limited resource subsets rather than availing access to entire physical machines. Notably, the model exhibits heightened precision when a fixed availability parameter is upheld across all request classes, representing a consistent fraction of the overall system capacity. The model's fidelity remains robust irrespective of request distribution. However, under scenarios of subdued traffic and heterogeneous availability across request classes, marginal diminutions in accuracy are observed. To further illustrate the accuracy of the model, sample results obtained using the analytical and simulation model for System 1 are summarized in Table 7. The table presents the loss probabilities for each class of requests, with selected input traffic per single AU of the entire system. The authors decided to select a_{AU} traffic at 0.7 Erl and 1 Erl.

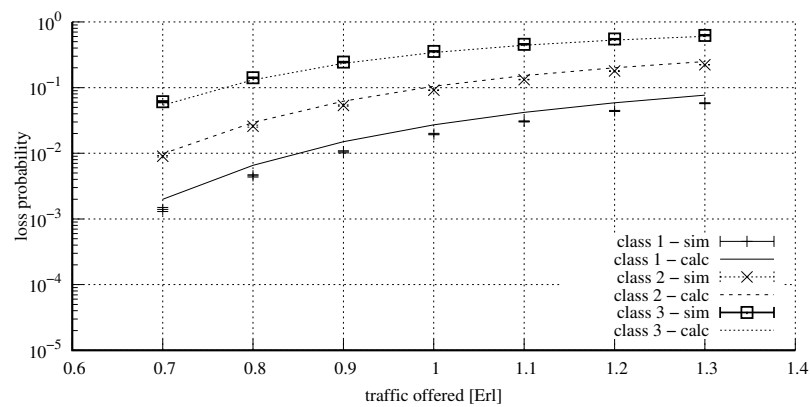


Figure 6. Loss probability in System 1.

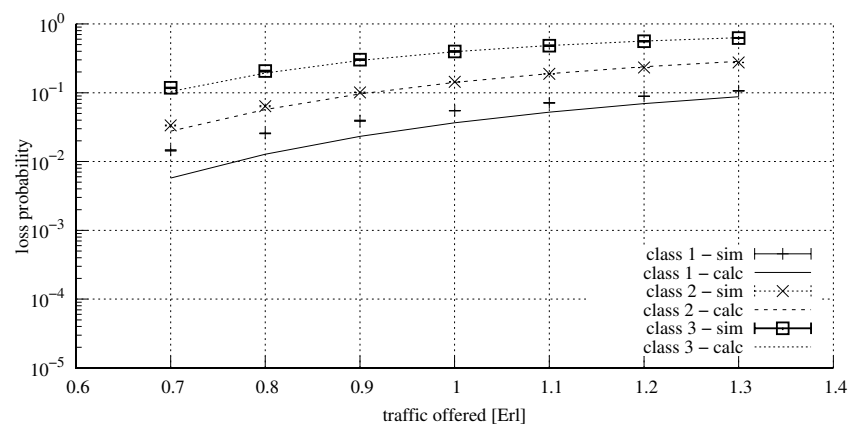


Figure 7. Loss probability in System 2.

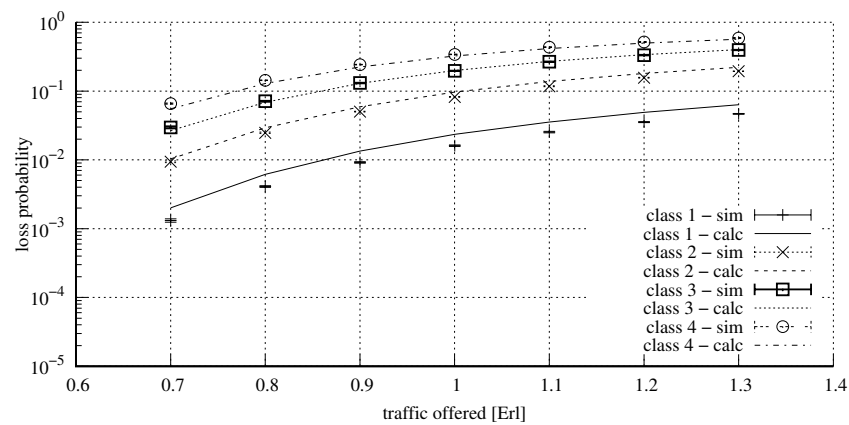


Figure 8. Loss probability in System 3.

Table 7. Simulation and calculation loss probability results for System 1 with 0.7 Erl and 1 Erl traffic.

$a_{AU} = 0.7$ Erl								
Class 1			Class 2			Class 3		
calc.	sim.		calc.	sim.		calc.	sim.	
0.0020	0.0014	± 0.0001	0.0098	0.0090	± 0.0002	0.0537	0.0610	± 0.0016
$a_{AU} = 1$ Erl								
Class 1			Class 2			Class 3		
calc.	sim.		calc.	sim.		calc.	sim.	
0.0271	0.0196	± 0.0005	0.1045	0.0912	± 0.0012	0.3419	0.3552	± 0.0017

It is imperative to underscore that the devised methodology is fully scalable for servers featuring identical specifications. This scalability stems from the inherent scalability of all constituent components of the proposed method, ensuring fidelity independent of the number of physical machines. The foundational models underpinning the LE-CBP methodology are widely embraced for modeling contemporary ICT systems and networks. Consequently, the proposed model holds utility in preemptively assessing requisite parameters for physical machines, including quality metrics such as virtual machine creation request rejections. Moreover, the model's user-friendliness and superior accuracy vis-à-vis previously posited models by the authors [22,23] further accentuate its practical appeal.

In conclusion, the proposed model has sufficient accuracy to be used for preliminary analysis of cloud systems to model them for selected quality parameters. The developed method is fast and allows us to determine parameters in a virtual machine environment of variable size, which is a great advantage compared to many other models known in the literature. In addition, it allows analysis for systems in which access to virtual machines in physical machines is limited. These constraints can arise from the structure of physical machines, differences in parameters such as memory or processor speed, or access to specific areas in memory.

The ability to accurately assess the loss probability in the system makes it possible to adjust the physical size of the machines at the design stage to best suit the end user's requirements. In addition, the model allows for verifying the feasibility of using fewer machines and assessing the impact on potential losses in the system, so that the number of machines will be optimally adjusted, consequently reducing energy consumption. At the same time, by determining the occupancy distribution of each resource, it is possible to determine the probability of the occupied state. This illustrates the probability with which a given physical machine (its resources) will be loaded to a certain percentage. This distribution also makes it possible to determine the average utilization of system resources [18].

6. Conclusions

As part of the article, the authors' objective was to develop an approximate analytical model based on the Markov chain for a heterogeneous cloud system to determine loss probability. The model assumed a system composed of multiple physical machines on which virtual machines with varying requirements were created. These requirements relate both to the number of physical resources (variation in the size of virtual machines) and access to specific resources of the physical machine necessary to perform the service. A dedicated simulation model was developed to determine the accuracy of the analytical model. The authors, by distinguishing the external structure of the system consisting of multiple physical machines and the internal structure consisting of a single server, determined the equivalent traffic. This traffic approximated the actual traffic entering each physical machine, which made it possible to determine the actual loss probability in the entire system with limited availability. The conducted tests confirm the high accuracy of the proposed L-E-CBP model, the consequence of which is its applicability in the process of designing and sizing cloud systems.

The paper assumes a uniform distribution of virtual machines between physical machines, as well as an equal number of resources within the available physical machines. It also assumes that requests to create a virtual machine can be scaled from four parameters (RAM, CPU, disk, and bandwidth) to a single parameter. This is a certain limitation in its applicability since it is not universal for every system type. However, as demonstrated in the article, such scaling is typical of solutions available on the market.

In the future, the authors plan to consider other ways of distributing VMs and control mechanisms known from the literature [37–41], as well as the possibility of moving VMs between physical machines to minimize the loss factor. In addition, they assume the use of queuing, compression in the system, and the use of traffic distributions other than

Erlang, which are described in the articles [27,42]. Moreover, the authors will attempt to develop a multi-parametric model under the accessibility limitations adopted in this article. Another aspect important to consider in the future will be the use of cloud system resources and the application of cloud theory to optimize energy use [11,16,43–45]. Thus, the authors plan to initiate research on multidimensional Markov chains considering the use of SD-FIFO input queue alongside Erlang, Engset, and Pascal type adaptive traffic [42]. The next important stage of research will be the application of overflows considering the MIM-NSD-BPP method [46] and applying the methodology for the multi-parametric model presented in [18,19].

Author Contributions: Conceptualization, M.S. and M.W.; Methodology, S.H., M.S. and M.W.; Software, M.W.; Validation, M.W.; Formal analysis, M.W.; Writing—original draft, M.W.; Visualization, M.W.; Writing—review & editing, S.H., M.S. and M.W.; Supervision, S.H. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Ministry of Education and Science, Grant 0313/SBAD/1311.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The data are not publicly available due to the project's limitations.

Conflicts of Interest: The authors declare no conflicts of interest. The sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Cloud Computing Market to Reach \$1554.94 Bn by 2030. Available online: <https://www.grandviewresearch.com/press-release/global-cloud-computing-market> (accessed on 1 May 2024).
2. Mell, P.; Grance, T. *The NIST Definition of Cloud Computing*; Special Publication (NIST SP), National Institute of Standards and Technology: Gaithersburg, MD, USA, 2011.
3. Infographic: Amazon Maintains Cloud Lead as Microsoft Edges Closer. 2024. Available online: <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers> (accessed on 1 May 2024).
4. Windows Virtual Machines Pricing. Available online: <https://azure.microsoft.com/en-gb/pricing/details/virtual-machines/windows/#m-series> (accessed on 1 May 2024).
5. EC2 on-Demand Instance Pricing—Amazon Web Services. Available online: <https://aws.amazon.com/ec2/pricing/on-demand/> (accessed on 1 May 2024).
6. Ghosh, R.; Longo, F.; Naik, V.K.; Trivedi, K.S. Modeling and performance analysis of large scale IaaS Clouds. *Future Gener. Comput. Syst.* **2013**, *29*, 1216–1234. [CrossRef]
7. Bruneo, D. A Stochastic Model to Investigate Data Center Performance and QoS in IaaS Cloud Computing Systems. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 560–569. [CrossRef]
8. Chang, X.; Wang, B.; Muppala, J.K.; Liu, J. Modeling Active Virtual Machines on IaaS Clouds Using an M/G/m/m+K Queue. *IEEE Trans. Serv. Comput.* **2016**, *9*, 408–420. [CrossRef]
9. Chang, X.; Xia, R.; Muppala, J.K.; Trivedi, K.S.; Liu, J. Effective Modeling Approach for IaaS Data Center Performance Analysis under Heterogeneous Workload. *IEEE Trans. Cloud Comput.* **2018**, *6*, 991–1003. [CrossRef]
10. Khazaei, H.; Mišić, J.; Misic, V.; Beigi, N. Modeling the Performance of Heterogeneous IaaS Cloud Centers. In Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, Philadelphia, PA, USA, 8–11 July 2013; pp. 232–237. [CrossRef]
11. Mehmood, T.; Latif, S.; Malik, S. Prediction Of Cloud Computing Resource Utilization. In Proceedings of the 2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), Islamabad, Pakistan, 8–10 October 2018; pp. 38–42. [CrossRef]
12. Borkowski, M.; Schulte, S.; Hochreiner, C. Predicting cloud resource utilization. In Proceedings of the 9th International Conference on Utility and Cloud Computing, Shanghai China, 6–9 December 2016; pp. 37–42. [CrossRef]
13. Zia Ullah, Q.; Hassan, S.; Khan, G.M. Adaptive Resource Utilization Prediction System for Infrastructure as a Service Cloud. *Comput. Intell. Neurosci.* **2017**, *2017*, e4873459. [CrossRef] [PubMed]
14. Islam, S.; Keung, J.; Lee, K.; Liu, A. Empirical prediction models for adaptive resource provisioning in the cloud. *Future Gener. Comp. Syst.* **2012**, *28*, 155–162. [CrossRef]
15. Bey, K.B.; Benhammadi, F.; Mokhtari, A.; Guessoum, Z. CPU Load Prediction Model for Distributed Computing. In Proceedings of the 2009 Eighth International Symposium on Parallel and Distributed Computing, Lisbon, Portugal, 30 June–4 July 2009; pp. 39–45. [CrossRef]

16. Li, R.; Wang, X.; Xiao, D.; Huang, C. Cloud Instance Resources Prediction Based on Hidden Markov Model. In Proceedings of the 2023 IEEE 9th International Conference on Cloud Computing and Intelligent Systems (CCIS), Dali, China, 12–13 August 2023; pp. 516–520. [\[CrossRef\]](#)
17. Farahnakian, F.; Liljeberg, P.; Plosila, J. LiRCUP: Linear Regression based CPU Usage Prediction Algorithm for Live Migration of Virtual Machines in Data Centers. In Proceedings of the 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, Spain, 4–6 September 2013; pp. 358–364. [\[CrossRef\]](#)
18. Hanczewski, S.; Stasiak, M.; Weissenberg, M. An Analytical Model of IaaS Architecture for Determining Resource Utilization. *Sensors* **2024**, *24*, 2758. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Hanczewski, S.; Stasiak, M.; Weissenberg, M. A Multiparameter Analytical Model of the Physical Infrastructure of a Cloud-Based System. *IEEE Access* **2021**, *9*, 100981–100990. [\[CrossRef\]](#)
20. Ataie, E.; Entezari-Maleki, R.; Rashidi, L.; Trivedi, K.; Movaghar, A. Hierarchical Stochastic Models for Performance, Availability, and Power Consumption Analysis of IaaS Clouds. *IEEE Trans. Cloud Comput.* **2017**, *7*, 1039–1056. [\[CrossRef\]](#)
21. Kirsal, Y. Performance Modelling and Analysis of Clustered Servers in Mobile Edge Computing. In Proceedings of the 2020 28th Signal Processing and Communications Applications Conference (SIU), Gaziantep, Turkey, 5–7 October 2020; pp. 1–4. [\[CrossRef\]](#)
22. Hanczewski, S.; Stasiak, M.; Weissenberg, M. The Analytical Model of Complex Non-Full-Availability System. In *Image Processing and Communications: Techniques, Algorithms and Applications; Advances in Intelligent Systems and Computing*, AISC; Springer: Berlin/Heidelberg, Germany, 2020; pp. 279–286. [\[CrossRef\]](#)
23. Hanczewski, S.; Stasiak, M.; Weissenberg, M. Modeling of Complex Non-Full-Availability Systems. *J. Telecommun. Inf. Technol.* **2020**, *1062*, 70–77. [\[CrossRef\]](#)
24. Stasiak, M. Blocking probability in a limited-availability group carrying mixture of different multichannel traffic streams. In *Annales des Télécommunications*; Springer Science and Business Media LLC.: Berlin/Heidelberg, Germany, 1993. [\[CrossRef\]](#)
25. Erlang, K.A. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Post Off. Electr. Eng. J.* **1917**, *10*, 189–197.
26. Stasiak, M. An approximate model of a switching network carrying mixture of different multichannel traffic streams. *IEEE Trans. Commun.* **1993**, *41*, 836–840. [\[CrossRef\]](#)
27. Glabowski, M.; Hanczewski, S.; Stasiak, M.; Weissenberg, J. Modeling Erlang’s Ideal Grading with Multirate BPP Traffic. *Math. Probl. Eng.* **2012**, *2012*, e456910. [\[CrossRef\]](#)
28. Stasiak, M.; Sobieraj, M.; Zwierzykowski, P. Modeling of Multi-Service Switching Networks With Multicast Connections. *IEEE Access* **2022**, *10*, 5359–5377. [\[CrossRef\]](#)
29. Nowak, B.; Zwierzykowski, P. A New Model of the Limited Availability Group with Priorities for Multi-Service Networks. *Electronics* **2024**, *13*, 1174. [\[CrossRef\]](#)
30. Stasiak, M.; Glabowski, M.; Wiśniewski, A.; Zwierzkowski, P. *Modeling and Dimensioning of Mobile Wireless Networks: From GSM to LTE*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2010.
31. Logothetis, M.; Moscholios, I.D. *Efficient Multirate Teletraffic Loss Models Beyond Erlang*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2019.
32. Glabowski, M.; Kaliszan, A.; Stasiak, M. Modeling product-form state-dependent systems with BPP traffic. *Perform. Eval.* **2010**, *67*, 174–197. [\[CrossRef\]](#)
33. Lehmer, D.H. Mathematical Methods in Large-scale Computing Units. In Proceedings of the Second Symposium on Large Scale Digital Computing Machinery, Cambridge, UK, 10–12 December 1951; pp. 141–146.
34. Tyszer, J. *Object-Oriented Computer Simulation of Discrete-Event Systems: 10*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1999.
35. Neves Calheiros, R.; Ranjan, R.; Beloglazov, A.; De Rose, C.A.; Buyya, R. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exp.* **2011**, *41*, 23–50. [\[CrossRef\]](#)
36. Kliazovich, D.; Bouvry, P.; Audzevich, Y.; Khan, S.U. GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers. *J. Supercomput.* **2010**, *62*, 1263–1283. [\[CrossRef\]](#)
37. Nuaimi, K.A.; Mohamed, N.; Nuaimi, M.A.; Al-Jaroodi, J. A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms. In Proceedings of the 2012 Second Symposium on Network Cloud Computing and Applications, London, UK, 3–4 December 2012; pp. 137–142. [\[CrossRef\]](#)
38. Rai, H.; Ojha, S.; Nazarov, A. Cloud Load Balancing Algorithm. In Proceedings of the 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 18–19 December 2020; pp. 861–865. [\[CrossRef\]](#)
39. Lohumi, Y.; Gangodkar, D.; Srivastava, P.; Khan, M.Z.; Alahmadi, A.; Alahmadi, A.H. Load Balancing in Cloud Environment: A State-of-the-Art Review. *IEEE Access* **2023**, *11*, 134517–134530. [\[CrossRef\]](#)
40. Zhao, J.; Lv, Y.; Zeng, Q.; Wan, L. Online Policy Learning-Based Output-Feedback Optimal Control of Continuous-Time Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 652–656. [\[CrossRef\]](#)
41. Sakr, S.; Liu, A.; Batista, D.M.; Alomari, M. A Survey of Large Scale Data Management Approaches in Cloud Environments. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 311–336. [\[CrossRef\]](#)
42. Weissenberg, J.; Weissenberg, M. Model of a Queuing System With BPP Elastic and Adaptive Traffic. *IEEE Access* **2022**, *10*, 130771–130783. [\[CrossRef\]](#)

43. Tian, G.; Zhang, C.; Fathollahi-Fard, A.M.; Li, Z.; Zhang, C.; Jiang, Z. An Enhanced Social Engineering Optimizer for Solving an Energy-Efficient Disassembly Line Balancing Problem Based on Bucket Brigades and Cloud Theory. *IEEE Trans. Ind. Inform.* **2023**, *19*, 7148–7159. [[CrossRef](#)]
44. Yeh, W.C.; Zhu, W.; Yin, Y.; Huang, C.L. Cloud Computing Considering Both Energy and Time Solved by Two-Objective Simplified Swarm Optimization. *Appl. Sci.* **2023**, *13*, 2077. [[CrossRef](#)]
45. Zhang, L.; Zhang, C.; Tian, G.; Chen, Z.; Fathollahi-Fard, A.M.; Zhao, X.; Wong, K.Y. A multi-criteria group-based decision-making method considering linguistic neutrosophic clouds. *Expert Syst. Appl.* **2023**, *226*, 119936. [[CrossRef](#)]
46. Głabowski, M.; Kmiecik, D.; Stasiak, M. Modelling Overflow Systems with Queuing in Primary Resources. In Proceedings of the Quality, Reliability, Security and Robustness in Heterogeneous Systems: 14th EAI International Conference, Qshine 2018, Ho Chi Minh City, Vietnam, 3–4 December 2018; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, LNICST; Springer: Berlin/Heidelberg, Germany, 2019; pp. 148–157. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.