Future Generation Computer Systems 102 (2020) 746–761

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Analytical modeling of performance indices under epistemic uncertainty applied to cloud computing systems

Fabio Antonelli [a], Vittorio Cortellessa [a], Marco Gribaudo [b], Riccardo Pinciroli [c], Kishor S. Trivedi [d], Catia Trubiani [e,*]

[a] University of L'Aquila, L'Aquila, Italy
[b] Politecnico di Milano, Milano, Italy
[c] College of William and Mary, Williamsburg, VA, USA
[d] Duke University, Durham, NC, USA
[e] Gran Sasso Science Institute, L'Aquila, Italy

ARTICLE INFO

Article history:
Received 29 March 2019
Received in revised form 12 July 2019
Accepted 6 September 2019
Available online 18 September 2019

Keywords:
Uncertainty modeling
Epistemic uncertainty propagation
Confidence interval
M/M/1 queue
Performance modeling
Cloud computing
VM migration
CloudSim

ABSTRACT

The extent of epistemic uncertainty in modeling and analysis of complex systems is ever growing, mainly due to increasing levels of the openness, heterogeneity and versatility in cloud-based applications that are being adopted in critical sectors, like banking and finance. State-of-the-art approaches for model-based performance assessment do not embed such uncertainty in analytic models, hence the predicted results do not account for the parametric uncertainty. In this paper, we develop a method for incorporating epistemic uncertainty of the input parameters (i.e., the arrival rate λ and the service rate μ) to the M/M/1 queueing models, that are commonly used to analyze system performance. We consider two steady state and average output measures: the number of entities in the system and the response time. We start with closed-form solutions for these measures that enable us to study the propagation of epistemic uncertainty in input parameters to these output measures. We demonstrate the suitability of our method for the performance analysis of a cloud-based system, where the epistemic uncertainty comes from continuous re-deployment of applications across servers of different computational capabilities. System simulation results validate the ability of our models to produce satisfactorily accurate predictions of system performance indices under epistemic uncertainty.

1. Introduction

The ever growing complexity of computer-based critical systems is directly attributable to modern application domains (like banking, finance, and e-health) that are required to dynamically adapt due to changing workloads, scenarios and objectives. This puts the onus on the underlying software and hardware that need to fulfill end users' requirements [1]. However, due to the openness, heterogeneity and versatility of such systems, very often system developers and analysts have to make decisions in presence of a type of uncertainty, frequently called "epistemic".

In this paper, we revisit the well-known M/M/1 queue commonly used to model system performance. Uncertainty in the time between arrivals and service times that are inherent in the system dynamics are captured by the homogeneous continuous time Markov chain that is used to solve for the performance indices of the M/M/1 queue. This type of uncertainty is known as aleatory uncertainty. Two input parameters of this model, namely the arrival rate λ and the service rate μ, are generally considered as fixed values while solving the queueing model. However, the two input parameter values are not known (or accurately estimated) at the analysis time. Variability or uncertainty in the values of these input parameters is of concern in this paper.

There are three types of variability in parameters depending on time scale. If the parametric variation is dynamic with the system operation and fits some well-known stochastic pattern, then such queues have been studied under the aegis of queues in a random environment [2–4]. Examples include queues with server failure and repair [5], and queues with vacations [6]. If the parametric variation is semi-dynamic then the arrival rate and/or the service rate can be seen as a time varying function. For example, arrival rate to a server can be dependent on the time of the day [7], or the service rate of the server may gradually decrease due to onset of the software aging phenomena [8]. The underlying stochastic model can be considered a non-homogeneous continuous time Markov chain (or Non-Homogeneous Poisson Process

* Corresponding author.
E-mail addresses: fabio.antonelli@univaq.it (F. Antonelli), vittorio.cortellessa@univaq.it (V. Cortellessa), marco.gribaudo@polimi.it (M. Gribaudo), rpinciroli@wm.edu (R. Pinciroli), ktrivedi@duke.edu (K.S. Trivedi), catia.trubiani@gssi.it (C. Trubiani).

https://doi.org/10.1016/j.future.2019.09.006
0167-739X/

— NHPP) in this case [7–9]. In the two cases, dynamic and semi-dynamic, the aleatory model is thus modified to account for the variation in parameter values. The third case is the parametric variation that is not dynamic, i.e., parameter values are fixed but unknown a priori. This type of parametric variation is known as epistemic uncertainty and it represents the focus of the paper. Note that modeling techniques such as NHPP, Markov Modulated Poisson Processes (MMPP) or Markov Arrival Processes (MAP) would not be suitable in this context, since they inherently account for the cases in which parameters change is somehow known and can be explicitly modeled.

In practice, the input parameters to aleatory models are estimated from measurements, and thus can be considered as random variables. It behooves the analyst to consider the effects that this randomness of the input parameters have on the output results of the queueing model. The objective of this paper is to show how to carry out such parametric uncertainty propagation to derive confidence intervals on the steady state outputs of the M/M/1 queue. Two examples of such outputs are the average number of entities in the system, and the average response time both in the steady state. Parametric uncertainty propagation has been studied for reliability models [10–13] but, to the best of our knowledge, its application to queueing systems is relatively unexplored [14]. In fact, very few papers have considered parametric uncertainty that arises from the finite number of samples used to estimate the input parameters. Assuming a consistent estimator [15], this uncertainty can be reduced by increasing the sample size or eliminated altogether by an infinite sample size. Hence this parametric uncertainty is called reducible.

In carrying out such uncertainty propagation, the arrival and the service rates for the M/M/1 queue are now considered to be random variables. As they vary over their respective support, not only the stability condition needs to be maintained, but also computational instability arises when the values of the two parameters are nearly equal. An important contribution of this paper is to show how to deal with this problem.

In this paper, we consider epistemic uncertainty and, to the best of our knowledge, the present work represents the first systematic effort to develop a method accounting for parametric uncertainty in a queueing system. Our contributions include developing basic equations for dealing with parameters uncertainty as a double integral, and solving the double integral in closed form in some cases while using numerical integration otherwise. One key difficulty in dealing with such queues is that, as the input parameters vary their values, not only the stability condition ought to be maintained but also need to avoid near instability. This makes the problem of integration somewhat delicate. We investigate and show two different approaches to solve this problem. We are thus able to provide confidence intervals on the two considered outputs of an M/M/1 queue on the basis of epistemic variations in the two input parameters.

This theoretical result can help system designers to tackle many interesting issues of system design and analysis. Below we present three motivating examples that include the one we use in this paper for sake of validation of our approach.

First, assume that we are designing a system that still does not exist. We are in the early life-cycle phases, where we deal with specifications, but not all the system characteristics have been defined. There is a margin of uncertainty that is intrinsically due to the lack of knowledge (i) either about the future decisions that will be taken about the system, (ii) or about the environment where the system will be working in (e.g., adaptive systems are designed to run within different environments without degrading "too much" their quality attributes). If the system is modeled as an M/M/1 queue – as done here –, then it may be the case that we can only establish lower and upper bounds on the service rate (an endogenous system variable, case *i* above), and on the arrival rate (an environmental variable, case *ii* above).

Another issue relates to early decisions about the system design. For example, assume that we have a requirement on the response time. With the techniques that we discuss in this paper, we can study what is the risk of violating that requirement under the current uncertainty in the system and environment parameters. In fact, we can calculate the width of the (resulting) response time interval that fits with the corresponding requirement. This evaluation can lead, for example, to two types of considerations about the system design: first, do not place the system in an environment where the arrival rate is larger than a threshold $\bar{\lambda}$; second, make design decisions that lead the service rate to be higher than another threshold $\bar{\mu}$.

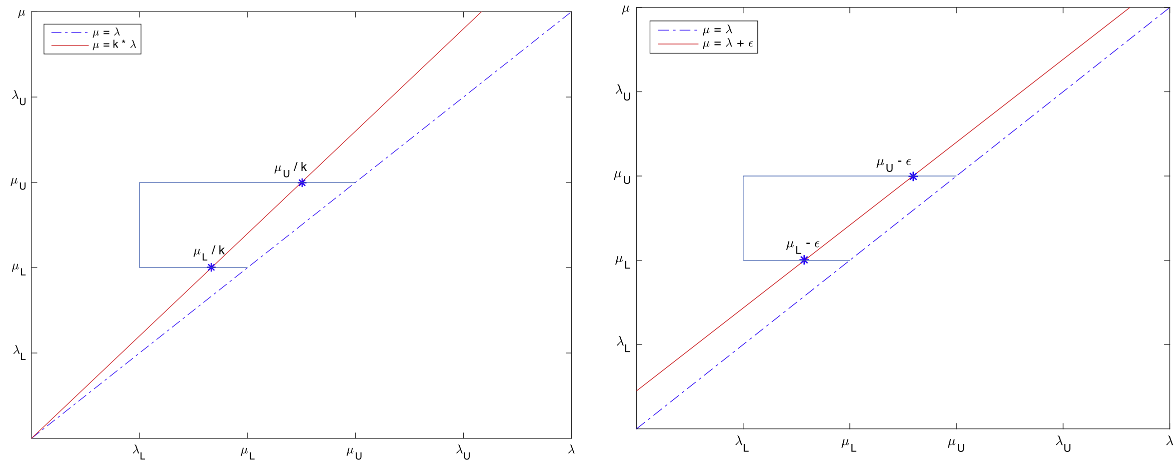
We demonstrate the suitability of our approach within the cloud computing domain. In particular, we consider the uncertainty related to the characteristics of the platform where a cloud-based application runs. In fact, these characteristics derive from decisions taken by the cloud manager that, for the sake of global system optimization, may decide to (continuously) re-deploy an application across the cloud (i.e., epistemic uncertainty), thus leading to change multiple times the running context during the application lifetime. This aspect induces variability in the performance parameters of the application (i.e., the mean arrival rate λ and the mean service rate μ mentioned above), which are difficult to be precisely defined in advance, and therefore they contribute to the system epistemic uncertainty to be taken into account. To be more specific, a cloud application can be affected by two types of variability: one determined by the specific type of server assigned to host the virtual machine, and another due to random environment inherently represented by the data center. For example, in [16] the distribution of the speed of the Virtual Machines (VMs) assigned on the Amazon EC2 platform is studied, and it is shown that users might experience very different performance (depending on the physical architecture of the host on which the VM is executed). If a VM is migrated on another host during its execution, then the performance expected from the application may vary. While the variability induced on the VMs from the data center can be addressed using classical techniques to support random environment such as MMPP, in this paper we focus on the variability determined by the host selected to run the VM. This cannot be anticipated with prior information, but it can be efficiently handled with the techniques proposed in this work.

The paper is structured as follows. Section 2 reviews related work. Section 3 states the problem, and the equations for uniformly distributed input parameters are given. In Section 4 the analytical results are shown and discussed. In Section 5 we apply our approach to a cloud-based application, and we validate our results against simulation. Finally, Section 6 concludes the paper and provides future research directions.

2. Related work

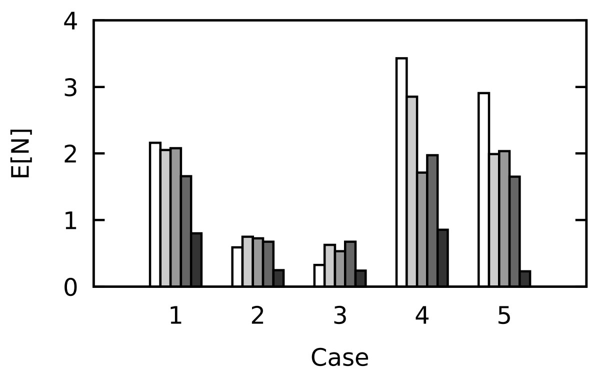
Uncertainty has been classified in two main categories, i.e., aleatory and epistemic [17,18]. The former is also known as irreducible uncertainty, it represents randomness inherent in system dynamics; the latter may be reduced [19] and is due to a limited sample size of measurements needed to estimate the system parameters. In literature, the aleatory uncertainty only has been accounted [2–4]. This paper instead focuses on epistemic uncertainty and most relevant related work is discussed in the following.

In [20] a numerical method for M/G/1/K (finite state) queues with vacation is proposed, and the vacation parameter only is affected by epistemic uncertainty. In [21] parametric uncertainty

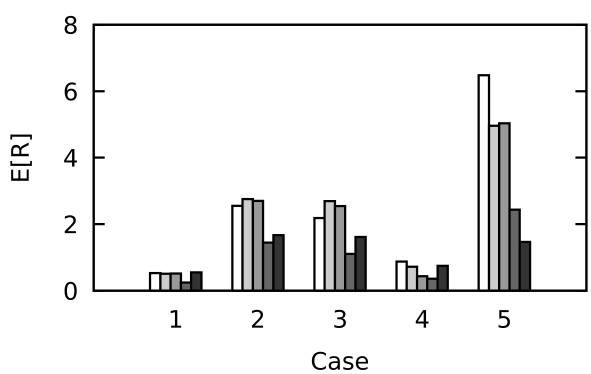


(a) Case (3)-prod: $\mu = k * \lambda$ (b) Case (3)-sum: $\mu = \lambda + \epsilon$

Fig. 1. Case (3): $\lambda_L < \mu_L < \mu_U < \lambda_U$.



(a) Average number of entities



(b) Average response time

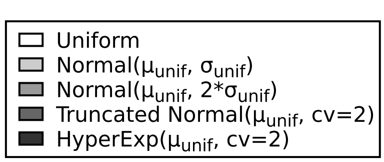


Fig. 2. Effect of different distributions for input parameters on output measures.

is studied targeting Geo/Geo/1/K queues, and input parameter values are regulated by pre-defined bounds. Our work differs from [20] and [21] for several reasons: (i) we consider the case where two input parameters show epistemic uncertainty; (ii) since arrival and service rates are the uncertain input parameters and we are dealing with an infinite state system (i.e., M/M/1 queue), we need also to take into account its stability condition; (iii) we use both closed form expressions and a numerical approach to propagate the uncertainty through the queueing model.

In [22] uncertainty is analyzed by introducing a new class of models that capture the parameters' variability over time. This approach is based on [23], where DTMCs have been extended with transition probabilities that are regulated by lower and upper bounds. Time Interval Petri Nets have been introduced in [24] to express temporal uncertainty in inputs (i.e., system states) and outputs (i.e., simulation results). However, all these approaches [22–24] deal with uncertain parameters that are not regulated by distribution functions, on the contrary in this paper we exploit the distributions to derive closed form formulas for the performance metrics (e.g., average number of entities in the system and response time).

In [10] parametric epistemic uncertainty propagation through analytic dependability models is presented. Closed-form expressions for the distribution function, expected value and variance of model outputs are derived for calculating the system reliability. As opposite, in this paper we are interested in the system performance. In [25] inverse uncertainty propagation on input parameters is used to find those uncertainties that satisfy a given bound on the simulation output uncertainty. Deducing the exact values of input parameters from observations of the system's results has been tackled in [26]. Our approach, instead, starts from a range of values for uncertain input parameters and determines how such uncertainty propagates to the output results.

In [27,28] multidimensional integration is adopted to analyze the uncertainty propagation, but this is unfeasible when dealing with complex and large systems. Alternative techniques have been proposed in literature: (i) Monte Carlo sampling [12], (ii) parametric sensitivity [29] to identify the input parameters that propagate more uncertainty on the output metrics. The Monte Carlo sampling method is also used for calculating: *(i)* reliability in [30], where a set of parameter range distributions self-regulate the number of architectural evaluations to the desired significance level; *(ii)* performance in [14], where parameters uncertainties are sampled from probability distribution functions and propagated in multiple software architectural models.

More recently, few approaches [31,32] take into account the uncertainty related to the availability of resources. This is mainly due to the adoption of preemptible cloud instances [33] that exhibit lower prices but do not guarantee continuity in the computation. However, to the best of our knowledge, none of the

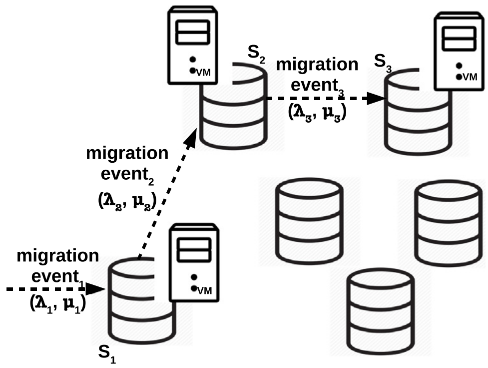


Fig. 3. Migration mechanism for a generic VM.

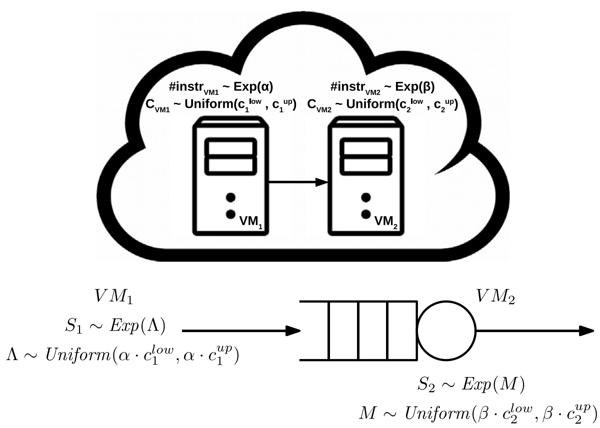


Fig. 4. Logical representation (top) and queueing model (bottom) of our cloud-based application.

existing approaches exploit this uncertainty as mean to drive the automated resource provisioning. When dealing with uncertainty, many approaches are based on sensitivity analysis techniques that aim to identify the parameter ranges affecting the software non-functional properties, e.g., [34]. In software performance, one of the seminal works in this direction is [35], where the concepts of uncertainty, performance conditions and implications first appeared.

3. Embedding epistemic uncertainty in performance models

In this section, we provide the mathematical calculations for analyzing an M/M/1 queueing system, with a certain degree of uncertainty on the arrival and service rates. In particular, we know by M/M/1 definition that interarrival and service times are exponentially distributed, but we assume uncertainty on their respective rates.

M/M/1 queues are usually adopted to analyze queueing models [36]. Some examples of current applications include data centers and cloud resources [37–39].

Hence, we introduce a probability space $(\Omega, \mathcal{F}, P)$, where Ω is the set of all possible outcomes, $\mathcal{F}$ is a set of events and each event contains zero or more outcomes, P is the assignment of probabilities to the events. In this probability space, Λ and M are two random variables respectively defined as the arrival and the service rates of an M/M/1 system. When we assume that these two random variables are independent, their joint probability density can be written as a product of their marginal densities: $f_{\Lambda,M}(\lambda,\mu) = f_\Lambda(\lambda) f_M(\mu)$.

In this setting, if we denote by N the average number of entities in the system in the steady-state, this is itself a random variable, depending on the pair (Λ, M). That is to say that we know N only conditionally on the values of the pair of random variables (Λ, M)

$$N|_{\Lambda=\lambda,M=\mu} = \frac{\lambda}{\mu-\lambda}.$$

In a similar way, we denote by R the average steady-state response time of the system, that is again a random variable depending on the values of the same pair (Λ, M):

$$R|_{\Lambda=\lambda,M=\mu} = \frac{1}{\mu-\lambda}.$$

We wish to study the system under stability condition, namely when the arrival rate is lower than the service rate, so that the average number of entities and the average response time do not tend to infinity in the long run. Hence, in order to limit the divergence of N and R on the random variables joint region (when λ and μ are very close), we introduce two different limiting assumptions, namely multiplicative and additive, respectively. These assumptions lead to two regions of (λ, μ) parameter values that are defined as follows:

$$B_k = \{(\lambda,\mu) : \mu > k\lambda\}, \qquad \text{for some } k > 1.$$

or

$$B_\epsilon = \{(\lambda,\mu) : \mu > \lambda + \epsilon\}, \qquad \text{for some } \epsilon > 0.$$

We remark that the condition $\mu > k\lambda$ is equivalent to the condition $N|_{\Lambda=\lambda,M=\mu} < \frac{1}{k-1}$. Similarly, the condition $\mu > \lambda + \epsilon$ is equivalent to the condition $R|_{\Lambda=\lambda,M=\mu} < \frac{1}{\epsilon}$. Furthermore, both regions are independent of the values of Λ and M and we denote the two events as:

$$A_k = \{(\Lambda, M) \in B_k\} = \left\{N < \frac{1}{k-1}\right\}, \quad \text{and}$$

$$A_\epsilon = \{(\Lambda, M) \in B_\epsilon\} = \left\{R < \frac{1}{\epsilon}\right\}.$$

Taking this into account, we compute

$$E[N|A_k] = \frac{1}{P(A_k)} E[N\mathbf{1}_{A_k}], \quad \text{or} \quad E[N|A_\epsilon] = \frac{1}{P(A_\epsilon)} E[N\mathbf{1}_{A_\epsilon}],$$

and

$$E[R|A_k] = \frac{1}{P(A_k)} E[R\mathbf{1}_{A_k}], \quad \text{or} \quad E[R|A_\epsilon] = \frac{1}{P(A_\epsilon)} E[R\mathbf{1}_{A_\epsilon}],$$

which reduces to computing the two numerators and the two denominators separately. For the denominators, we have:

$$P(A_k) = P((\Lambda, M) \in B_k) = \iint_{B_k} f_{\Lambda,M}(\lambda,\mu) d\lambda d\mu,$$

$$P(A_\epsilon) = P((\Lambda, M) \in B_\epsilon) = \iint_{B_\epsilon} f_{\Lambda,M}(\lambda,\mu) d\lambda d\mu$$

To simplify the presentation in the following we focus on the average number of entities in the system only. For the numerators we use the theorem of total expectation [15] by which we get:

$$\begin{aligned} E[N\mathbf{1}_{A_k}] &= E[\frac{\Lambda}{M-\Lambda}\mathbf{1}_{A_k}] = E[\frac{\Lambda}{M-\Lambda}\mathbf{1}_{B_k}(\Lambda, M)] \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{\lambda}{\mu-\lambda}\mathbf{1}_{B_k}(\lambda,\mu) f_{\Lambda,M}(\lambda,\mu) d\lambda d\mu \\ &= \int\int_{B_k} \frac{\lambda}{\mu-\lambda} f_{\Lambda,M}(\lambda,\mu) d\lambda d\mu, \end{aligned}$$

Table 1
Simulations input parameters and Analytic model accuracy for the five cases identified.

	Case 1	Case 2	Case 3	Case 4	Case 5
c_1^{low} [r/s]	3.6	0.198	0.036	2.7	0.198
c_1^{up} [r/s]	4.5	0.72	0.765	5.4	0.603
c_2^{low} [r/s]	5.49	0.09	0.549	0.9	0.549
c_2^{up} [r/s]	6.57	0.657	0.657	6.57	0.657
ϵ [r/s]	0.36	0.36	0.36	0.36	0.036
$1/\alpha = 1/\beta$ [i/r]	$4 \cdot 10^5$	$4 \cdot 10^5$	$4 \cdot 10^5$	$4 \cdot 10^5$	$4 \cdot 10^5$
T [h]	12	123	123	12	246
`migration_time` [min]	30	30	30	30	60
$N_{analytic}$ [r]	2.161	0.592	0.327	3.431	2.909
$R_{analytic}$ [s]	0.529	2.553	2.187	0.877	6.480
$MAPE_N$ [%]	2.40	**2.13**	**4.86**	3.20	4.43
$MAPE_R$ [%]	1.20	**0.70**	4.82	0.81	**12.26**

where the second equality is justified by the random variable $\mathbf{1}_{A_k}$ that is measurable with respect to the $\sigma-$algebra generated by (Λ, M).

Also we can compute the variance of the same random variable:

$$\mathrm{Var}[N|A_k] = E[N^2|A_k] - [E[N|A_k]]^2 = \frac{1}{P(A_k)} E[N^2 \mathbf{1}_{A_k}] - [E[N|A_k]]^2,$$

therefore reducing the problem to computing:

$$E[N^2 \mathbf{1}_{A_k}] = E[E[N^2 \mathbf{1}_{A_k} | \Lambda, M]] = E[\frac{\Lambda^2}{(M-\Lambda)^2} \mathbf{1}_{\{M > k\Lambda\}}].$$

We may also formalize the expectation and the variance of this random variable, when we restrict to the second type of region:

$$\begin{aligned} E[N\mathbf{1}_{A_\epsilon}] &= E[\frac{\Lambda}{M-\Lambda}\mathbf{1}_{A_\epsilon}] \\ &= E[\frac{\Lambda}{M-\Lambda}\mathbf{1}_{B_\epsilon}(\Lambda, M)] = \int\int_{B_\epsilon} \frac{\lambda}{\mu-\lambda} f_{\Lambda,M}(\lambda,\mu) d\lambda d\mu, \end{aligned}$$

and the variance expression is written accordingly. Analogous expressions can be written also for the response time random variable R.

Several choices can be made for the marginal densities of Λ and M: Uniform, Gaussian, Erlang, Lognormal. In the following we present how to apply the above formulation while considering Uniform and Gaussian densities. The former distribution represents the natural choice to account for the epistemic uncertainty in the M/M/1 input parameters (i.e., the only available information on rates is their respective ranges of possible values). The latter distribution corresponds to having a possible estimate of the area of the most likely values, assuming to be exponentially unlikely to be outside this area.

In the first case, the double integrals can be carried out to obtain explicit closed-form expressions for $E[N\mathbf{1}_{A_k}]$, $E[N\mathbf{1}_{A_\epsilon}]$, $E[R\mathbf{1}_{A_k}]$, and $E[R\mathbf{1}_{A_\epsilon}]$ (see Section 3.1). In the second case, we can only obtain semi-explicit expressions that require numerical evaluation. To get more accurate calculations, we modify and truncate the Gaussian density to limit its values to the positive quadrant ($\lambda > 0$, $\mu > 0$), more details are reported in Section 3.2.

3.1. The uniform distribution

We assume that Λ and M are both uniformly distributed on some known lower (L) and upper (U) bound intervals, that we denote respectively as $[\lambda_L, \lambda_U]$ and $[\mu_L, \mu_U]$. In this case, the two random variables are automatically independent and their joint probability density is given by

$$\begin{aligned} f_{\Lambda,M}(\lambda,\mu) &= f_\Lambda(\lambda) f_M(\mu) \\ &= \frac{1}{(\mu_U-\mu_L)(\lambda_U-\lambda_L)} \mathbf{1}_{[\lambda_L,\lambda_U]\times[\mu_L,\mu_U]}(\lambda,\mu) \\ &=: \frac{1}{C}\mathbf{1}_B(\lambda,\mu), \end{aligned}$$

where $C = (\lambda_U - \lambda_L)\cdot(\mu_U - \mu_L)$ denotes the area of the rectangle $B = [\lambda_L, \lambda_U] \times [\mu_L, \mu_U]$, where the joint density is concentrated.

Thus

$$P(A_k) = \frac{C_k}{C}, \quad P(A_\epsilon) = \frac{C_\epsilon}{C},$$

where by C_k and C_ϵ respectively denote the areas of the regions $B_k \cap B$ and $B_\epsilon \cap B$.

As a consequence, we have

$$\begin{aligned} E[N\mathbf{1}_{A_k}] &= \frac{1}{C}\int\int_{B_k\cap B} \frac{\lambda}{\mu-\lambda} d\lambda d\mu \\ E[N|A_k] &= \frac{1}{C_k}\int\int_{B_k\cap B} \frac{\lambda}{\mu-\lambda} d\lambda d\mu \end{aligned}$$

and

$$\begin{aligned} E[N^2\mathbf{1}_{A_k}] &= \frac{1}{C}\int\int_{B_k\cap B} \frac{\lambda^2}{(\mu-\lambda)^2} d\lambda d\mu \\ E[N^2|A_k] &= \frac{1}{C_k}\int\int_{B_k\cap B} \frac{\lambda^2}{(\mu-\lambda)^2} d\lambda d\mu \end{aligned}$$

Both integrals can be computed, and the results will depend on the relative positions of the endpoints of the $[\lambda_L, \lambda_U]$ and $[\mu_L, \mu_U]$ intervals. We have derived explicit expressions in all the cases, but we show only one case in Section 4, for which we stepwise evaluate the integrals.

Clearly, all integrals result in closed-form expressions thanks to the choice of uniform densities for our random variables.

3.2. The Gaussian distribution

Let us denote by ϕ the distribution function of a standard normal random variable and let us consider two normal random variables, i.e., $\mathcal{N}(\mu_0, \sigma^2)$ and $\mathcal{N}(\lambda_0, \eta^2)$, for some given positive parameters $\mu_0, \lambda_0, \sigma, \eta$. Taking into account that the rates should take only positive values, we assume that the random variables Λ and M have the following modified Gaussian joint probability density

$$\begin{aligned} f_{\Lambda,M}(\lambda,\mu) &= \frac{1}{C(\mu_0,\sigma,\lambda_0,\eta)} \\ &\times \frac{1}{2\pi\sigma\eta\sqrt{1-\rho^2}} \\ &\times \mathrm{e}^{-\frac{1}{2(1-\rho^2)}\left[\frac{(\mu-\mu_0)^2}{\sigma^2} + \frac{(\lambda-\lambda_0)^2}{\eta^2} - \frac{2\rho(\mu-\mu_0)(\lambda-\lambda_0)}{\sigma\eta}\right]} \mathbf{1}_{\{\mu>0,\lambda>0\}}, \end{aligned}$$

where $\rho \in [-1, 1]$ is the correlation coefficient between M and Λ, the covariance matrix is given by $\Sigma = \begin{pmatrix} \sigma^2 & \rho\sigma\eta \\ \rho\sigma\eta & \eta^2 \end{pmatrix}$ and the normalizing constant $C(\mu_0, \sigma, \lambda_0, \eta)$ by

$$\int_{-\frac{\mu_0}{\sigma}}^{+\infty} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{z^2}{2}} \Phi\Big(\frac{\mu_0}{\sigma\sqrt{1-\rho^2}} + \frac{\rho}{\sqrt{1-\rho^2}} z\Big) dz,$$

with Φ denoting the cumulative distribution function of a standard Normal. Considering dependent random variables makes sense, for example, in a cloud environment such us the one motivating the results proposed in this paper. Specifically, the

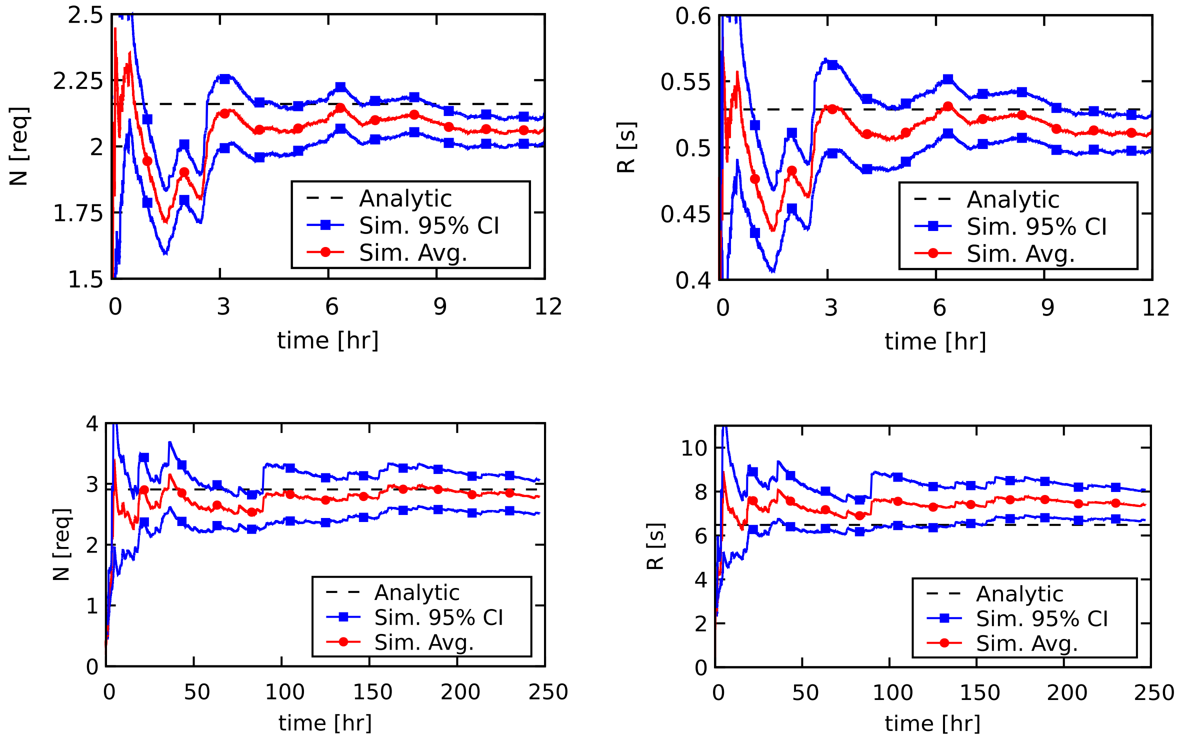


Fig. 5. Number of customers (left), and system response time (right), for case 1 (top) and case 5 (bottom).

correlation among service time can occur due for example to autoscaling (that induce dependency) or to failures that can affect several servers at the same time in a data center. Nevertheless, for ease of exposition and to simplify computations, in what follows we consider the independent case, hence

$$f_{\Lambda,M}(\lambda,\mu)=f_{\Lambda}(\lambda)f_M(\mu)=\frac{1}{\Phi(\frac{\mu_0}{\sigma})}\frac{1}{\Phi(\frac{\lambda_0}{\eta})}\times\frac{1}{2\pi\sigma\eta}e^{-\frac{(\mu-\mu_0)^2}{2\sigma^2}}e^{-\frac{(\lambda-\lambda_0)^2}{2\eta^2}}\mathbf{1}_{\{\mu>0,\lambda>0\}}.$$

In this case may compute

$$P(A_k)=\int_0^{+\infty}\int_{k\lambda}^{+\infty}\frac{1}{\Phi(\frac{\mu_0}{\sigma})}\frac{1}{\Phi(\frac{\lambda_0}{\eta})}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\mu-\mu_0)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi}\eta}\times e^{-\frac{(\lambda-\lambda_0)^2}{2\eta^2}}d\mu d\lambda$$

$$=\frac{1}{\Phi(\frac{\mu_0}{\sigma})\Phi(\frac{\lambda_0}{\eta})}\int_0^{+\infty}\Phi\left(\frac{\mu_0-k\lambda}{\sigma}\right)\frac{1}{\sqrt{2\pi}\eta}e^{-\frac{(\lambda-\lambda_0)^2}{2\eta^2}}d\lambda,$$

$$P(A_\epsilon)=\frac{1}{\Phi(\frac{\mu_0}{\sigma})\Phi(\frac{\lambda_0}{\eta})}\int_0^{+\infty}\Phi\left(\frac{\mu_0-\lambda-\epsilon}{\sigma}\right)\frac{1}{\sqrt{2\pi}\eta}e^{-\frac{(\lambda-\lambda_0)^2}{2\eta^2}}d\lambda$$

and as a consequence, we have

$$E[N\mathbf{1}_{A_k}]=\frac{1}{\Phi(\frac{\mu_0}{\sigma})}\frac{1}{\Phi(\frac{\lambda_0}{\eta})}\int_0^{+\infty}\int_{k\lambda}^{+\infty}\frac{\lambda}{\mu-\lambda}\times\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\mu-\mu_0)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi}\eta}e^{-\frac{(\lambda-\lambda_0)^2}{2\eta^2}}d\mu d\lambda,$$

$$E[N|A_k]=\frac{1}{P(A_k)}\frac{1}{\Phi(\frac{\mu_0}{\sigma})}\frac{1}{\Phi(\frac{\lambda_0}{\eta})}\int_0^{+\infty}\int_{k\lambda}^{+\infty}\frac{\lambda}{\mu-\lambda}\times\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\mu-\mu_0)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi}\eta}e^{-\frac{(\lambda-\lambda_0)^2}{2\eta^2}}d\mu d\lambda$$

and

$$E[N^2\mathbf{1}_{A_k}]=\frac{1}{\Phi(\frac{\mu_0}{\sigma})}\frac{1}{\Phi(\frac{\lambda_0}{\eta})}\int_0^{+\infty}\int_{k\lambda}^{+\infty}\frac{\lambda^2}{(\mu-\lambda)^2}\times\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\mu-\mu_0)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi}\eta}e^{-\frac{(\lambda-\lambda_0)^2}{2\eta^2}}d\mu d\lambda,$$

$$E[N^2|A_k]=\frac{1}{P(A_k)}\frac{1}{\Phi(\frac{\mu_0}{\sigma})}\frac{1}{\Phi(\frac{\lambda_0}{\eta})}\int_0^{+\infty}\int_{k\lambda}^{+\infty}\frac{\lambda^2}{(\mu-\lambda)^2}\times\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\mu-\mu_0)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi}\eta}e^{-\frac{(\lambda-\lambda_0)^2}{2\eta^2}}d\lambda d\mu.$$

All the integrals in this subsection require numerical integration, that we carry out with the aid of Mathematica ([1]). Similar calculations can be run for the additive bound case.

4. Theoretical issues around specific modeling cases

We provide here details of the derivations of $E[N|A_{k,\epsilon}]$, $E[R|A_{k,\epsilon}]$, $Var[N|A_{k,\epsilon}]$, and $Var[R|A_{k,\epsilon}]$, while considering different relative positions of λ's and μ's lower and upper bounds (i.e., λ_L, λ_U, μ_L, μ_U). From now on to simplify notation we will indicate those as $E[N]$, $E[R]$, $Var[N]$, and $Var[R]$. In order to cope with all possibilities, five different cases have to be considered, i.e.:

(1) $\lambda_L<\lambda_U<\mu_L<\mu_U$;
(2) $\mu_L<\lambda_L<\mu_U<\lambda_U$;
(3) $\lambda_L<\mu_L<\mu_U<\lambda_U$;
(4) $\mu_L<\lambda_L<\lambda_U<\mu_U$;
(5) $\lambda_L<\mu_L<\lambda_U<\mu_U$.

For the sake of order, their graphic representations are given in this section only for case (3), and in Appendix A for all other cases.

[1] https://www.wolfram.com/mathematica/.

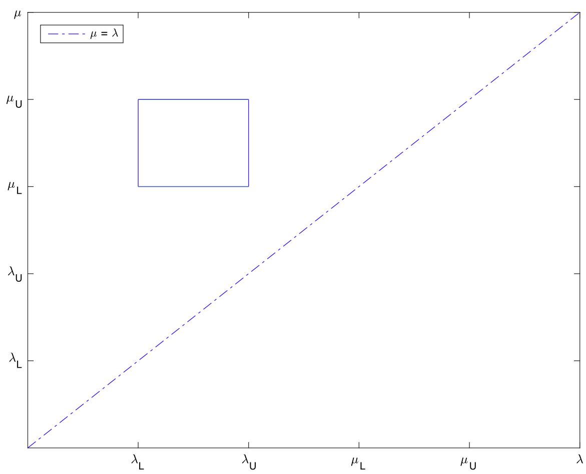
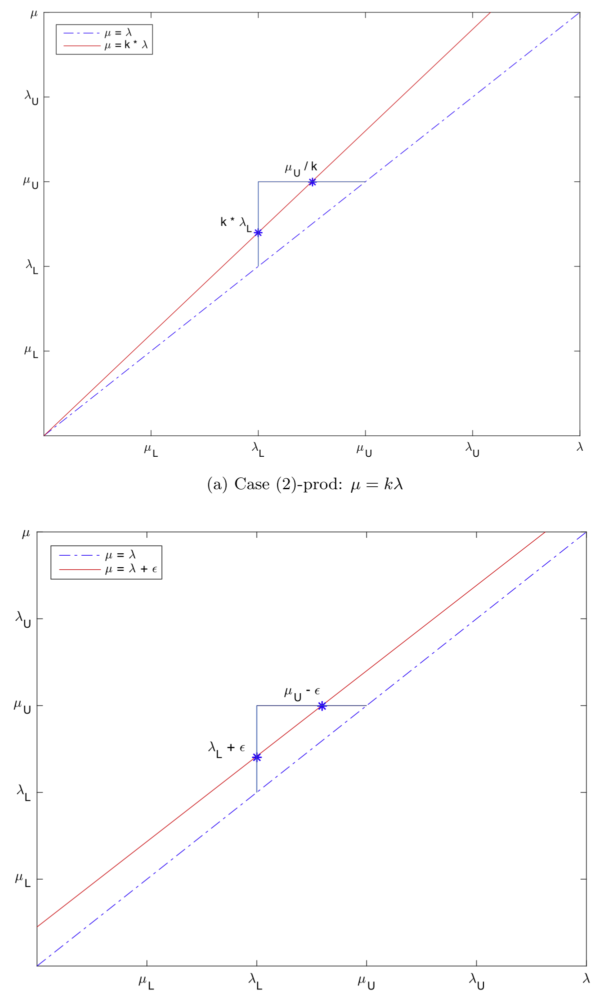


Fig. A.6. Case (1): $\lambda_L < \lambda_U < \mu_L < \mu_U$.



(a) Case (2)-prod: $\mu = k\lambda$

(b) Case (2)-sum: $\mu = \lambda + \epsilon$

Fig. A.7. Case (2): $\mu_L < \lambda_L < \mu_U < \lambda_U$.

Fig. 1 illustrates case (3), where on the (x, y)-axes we report λ's and μ's lower and upper bounds, knowing that $\lambda_L < \mu_L < \mu_U < \lambda_U$. Dashed lines denote the bisector ($\mu = \lambda$), whereas solid lines represent our two limiting assumptions (i.e., multiplicative and additive). In Fig. 1 we can notice that solid lines intersect the integration area, thus further elaborations are needed to evaluate the integral. Specifically, this case is analyzed by considering the limitations defined in Section 3, i.e.: *(i)* $\mu > k\lambda$, that will be referred as *prod*, and *(ii)* $\mu > \lambda + \epsilon$, that will be referred as *sum*. In Fig. 1(a) we report the *prod* case, where we can notice that the intersection with the line $\mu = k\lambda$ occurs in points $(\mu_L/k, \mu_L)$ and $(\mu_U/k, \mu_U)$. In Fig. 1(b) we report the *sum* case, where we can notice that the intersection with the line $\mu = \lambda + \epsilon$ occurs in points $(\mu_L - \epsilon, \mu_L)$ and $(\mu_U - \epsilon, \mu_U)$. These points are then used in the formulas for $E[N]$, $E[R]$, $Var[N]$, and $Var[R]$ since they delimit the area of interest. We remark that, in both cases, the approximation parameter (i.e., λ or μ) has to be limited so that the cutting line does not overcome the closest vertex of the integration region. For example, in Fig. 1(b) we must ensure that $\epsilon < \mu_L/\lambda_L$, otherwise the shape of integration area changes.

In the following we run explicit computation for case (3). In this case, the area of the integration region is $C_k = \frac{1}{2}(\mu_U - \mu_L)(\mu_U/k - \lambda_L + \mu_L/k - \lambda_L)$, and the integral introduced in Section 3 becomes:

$$\begin{aligned}
\int\int_{B_k \cap B} \frac{\lambda}{\mu - \lambda} d\lambda d\mu &= \int\int_{B_k \cap B} (\frac{\mu}{\mu - \lambda} - 1) d\lambda d\mu \\
&= \int_{\mu_L}^{\mu_U} \int_{\lambda_L}^{\frac{\mu}{k}} \frac{\mu}{\mu - \lambda} d\lambda d\mu - C_k \\
&= -\int_{\mu_L}^{\mu_U} \mu \ln(\mu - \lambda)\Big|_{\lambda_L}^{\frac{\mu}{k}} d\mu - C_k \\
&= -\int_{\mu_L}^{\mu_U} \mu \ln(\frac{k-1}{k}\mu) d\mu \\
&\quad + \int_{\mu_L}^{\mu_U} \mu \ln(\mu - \lambda_L) d\mu - C_k.
\end{aligned}$$

All the above integrals are easily computable in an explicit manner, leading to

$$\begin{aligned}
\int\int_{B_k \cap B} \frac{\lambda}{\mu - \lambda} d\lambda d\mu &= \frac{1}{2}\ln(\frac{k}{k-1})[\mu_U^2 - \mu_L^2] \\
&\quad - \Big[\frac{\mu^2}{2}\ln\mu\Big|_{\mu_L}^{\mu_U} - \frac{\mu^2}{4}\Big|_{\mu_L}^{\mu_U}\Big] \\
&\quad + [\frac{\mu^2}{2}\ln(\mu - \lambda_L)\Big|_{\mu_L}^{\mu_U} \\
&\quad - \frac{1}{2}\int_{\mu_L}^{\mu_U} \frac{\mu^2}{(\mu - \lambda_L)} d\mu - C_k] \\
&= \Big[\frac{1}{2}\ln(\frac{k}{k-1}) + \frac{1}{4}\Big][\mu_U^2 - \mu_L^2] \\
&\quad - \Big[\frac{\mu_U^2}{2}\ln\left(\frac{\mu_U}{\mu_U - \lambda_L}\right) - \frac{\mu_L^2}{2}\ln\left(\frac{\mu_L}{\mu_L - \lambda_L}\right)\Big] \\
&\quad - \frac{1}{2}\int_{\mu_L}^{\mu_U} \frac{(\mu - \lambda_L)^2 + 2\lambda_L(\mu - \lambda_L) + \lambda_L^2}{(\mu - \lambda_L)} d\mu \\
&\quad - C_k \\
&= \Big[\frac{1}{2}\ln(\frac{k}{k-1}) + \frac{1}{4}\Big][\mu_U^2 - \mu_L^2] \\
&\quad - \Big[\frac{\mu_U^2}{2}\ln\left(\frac{\mu_U}{\mu_U - \lambda_L}\right) - \frac{\mu_L^2}{2}\ln\left(\frac{\mu_L}{\mu_L - \lambda_L}\right)\Big] \\
&\quad - \frac{1}{2}\Big[\frac{1}{2}[\mu_U^2 - \mu_L^2] + \lambda_L[\mu_U - \mu_L] \\
&\quad + \lambda_L^2 \ln\left(\frac{\mu_U - \lambda_L}{\mu_L - \lambda_L}\right)\Big] - C_k \\
&= \frac{1}{2}\ln(\frac{k}{k-1})[\mu_U^2 - \mu_L^2]
\end{aligned}$$

Table B.2
Summary of closed formulas obtained for E[N].

	E[N]
1	$\frac{-(a-b)(c-d)+c^2\text{Log}\left[\frac{b-c}{a-c}\right]-b^2\text{Log}[-b+c]+a^2\text{Log}\left[\frac{a-c}{a-d}\right]+d^2\text{Log}[-a+d]+(b-d)(b+d)\text{Log}[-b+d]}{2(a-b)(c-d)}$
2-prod	$\frac{d(-d+ak)+d^2k\text{Log}\left[\frac{(-a+d)k}{d(-1+k)}\right]+a^2k\text{Log}\left[\frac{a-ak}{a-d}\right]}{(d-ak)^2}$
2-sum	$(\frac{(a+3d-\epsilon)(a-d+\epsilon)+2(a-d)(a+d)\text{Log}\left[-\frac{\epsilon}{a-d}\right]}{2(a-d+\epsilon)^2})$
3-prod	$\frac{-(c-d)(c+d-ak)+a^2k\text{Log}\left[\frac{a-d}{a-c}\right]-c^2k\text{Log}\left[\frac{c(-1+k)}{(-a+c)k}\right]+d^2k\text{Log}\left[\frac{d(-1+k)}{(-a+d)k}\right]}{(c-d)(c+d-2ak)}$
3-sum	$(-\frac{2\left(a^2-c^2\right)\text{Log}[-a+c]-2\left(a^2-d^2\right)\text{Log}[-a+d]+(c-d)(-2a+3c+3d-4\epsilon+2(c+d)\text{Log}[\epsilon])}{2(c-d)(-2a+c+d-2\epsilon)})$
4-prod	$-\frac{-ad+bd+d^2\text{Log}\left[\frac{b-d}{a-d}\right]-a^2\text{Log}\left[\frac{a-ak}{a-d}\right]+b^2\text{Log}\left[\frac{b-bk}{b-d}\right]}{(a-b)(-2d+(a+b)k)}$
4-sum	$(-\frac{-(a-b)(a+b+2d)+2d^2\text{Log}\left[\frac{b-d}{a-d}\right]-2a^2\text{Log}\left[-\frac{\epsilon}{a-d}\right]+2b^2\text{Log}\left[-\frac{\epsilon}{b-d}\right]}{2(a-b)(a+b-2d+2\epsilon)})$
5-prod	$((2k(\frac{1}{2k^2}(cdk-bdk^2+(b^2-d^2)k^2\text{Log}[-b+d]+(-c^2+d^2k^2)\text{Log}[d-\frac{c}{k}]-b^2k^2\text{Log}[b(-1+k)]+$ $c^2\text{Log}[\frac{c(-1+k)}{k}])+\frac{1}{2k^2}(c^2k-cdk-ack^2+adk^2+(a^2-c^2)k^2\text{Log}[-a+c]+\left(-a^2+d^2\right)k^2\text{Log}[-a+d]-$ $d^2k^2\text{Log}[d-\frac{c}{k}]-c^2\text{Log}[c(-1+k)]+c^2k^2\text{Log}[\frac{c(-1+k)}{k}]+c^2\text{Log}[-c+dk])))/(2(-c+d)(c-ak)+$ $(-c+2d-bk)(-c+bk)))$
5-sum	$((2(\frac{1}{4}(-(b+c+2d-\epsilon)(b-c+\epsilon)+2(b-d)(b+d)\text{Log}[\frac{-b+d}{\epsilon}]+2(c-d-\epsilon)(c+d-\epsilon)$ $\text{Log}[\frac{\epsilon}{-c+d+\epsilon}])+\frac{1}{2}(-(c-d)(a-c+\epsilon)-c^2\text{Log}[-a+c]+a^2\text{Log}[\frac{a-c}{a-d}]+$ $d^2\text{Log}[-a+d]+(2c-\epsilon)\epsilon\text{Log}[\epsilon]+(c-d-\epsilon)(c+d-\epsilon)$ $\text{Log}[-c+d+\epsilon])))/(2(-c+d)(-a+c-\epsilon)+(-b-c+2d-\epsilon)(b-c+\epsilon)))$

$$
\begin{aligned}
&-\left[\frac{\mu_U^2}{2}\ln\left(\frac{\mu_U}{\mu_U-\lambda_L}\right)-\frac{\mu_L^2}{2}\ln\left(\frac{\mu_L}{\mu_L-\lambda_L}\right)\right]\\
&-\frac{1}{2}\left[\lambda_L[\mu_U-\mu_L]+\lambda_L^2\ln\left(\frac{\mu_U-\lambda_L}{\mu_L-\lambda_L}\right)\right]-C_k.
\end{aligned}
$$

Alternatively written

$$
\begin{aligned}
\int\int_{B_k\cap B}\frac{\lambda}{\mu-\lambda}d\lambda d\mu =\;& \frac{\mu_U^2}{2}\ln\left(\frac{k(\mu_U-\lambda_L)}{(k-1)\mu_U}\right)\\
&-\frac{\mu_L^2}{2}\ln\left(\frac{k(\mu_L-\lambda_L)}{(k-1)\mu_L}\right)\\
&-\frac{\lambda_L}{2}(\mu_u-\mu_L)-\frac{\lambda_L^2}{2}\ln\left(\frac{\mu_U-\lambda_L}{\mu_L-\lambda_L}\right)-C_k.
\end{aligned}
$$

With similar calculations we obtain

$$
\begin{aligned}
\int\int_{B_k\cap B}\frac{\lambda^2}{(\mu-\lambda)^2}d\lambda d\mu =\;& C_k+\mu_U^2\ln\left(\frac{(k-1)\mu_U}{k(\mu_U-\lambda_L)}\right)\\
&-\mu_L^2\ln\left(\frac{(k-1)\mu_L}{k(\mu_L-\lambda_L)}\right)-\frac{k-2}{k-1}(\mu_U^2-\mu_L^2)
\end{aligned}
$$

We obtain an explicit expression for the expectation and the variance of the steady state average number of entities in the system.

Computations run along the same lines also when we wish to compute

$$
\begin{aligned}
&\int_{\mathbb{R}}\int_{\mathbb{R}}\frac{\lambda}{\mu-\lambda}\mathbf{1}_{\{\mu>\lambda+\epsilon\}}f_{\Lambda,M}(\lambda,\mu)d\lambda d\mu=\int_{\lambda_L}^{\lambda_U}\int_{\lambda+\epsilon}^{\mu_U}\frac{\lambda}{\mu-\lambda}d\mu d\lambda\\
&=\ldots\\
&=\frac{(\mu_U^2-\lambda_L^2)}{2}\ln(\mu_U-\lambda_L)-\frac{(\mu_U^2-\lambda_U^2)}{2}\ln(\mu_U-\lambda_U)\\
&-(\lambda_U-\lambda_L)\left[\frac{1}{4}(\lambda_U+\lambda_L)+\frac{\mu_U}{2}+\ln\epsilon\right].
\end{aligned}
$$

We refer the reader to Appendix B for $E[N]$, $E[R]$, $Var[N]$, and $Var[R]$ formulas of all cases that have been derived using Mathematica ([2]). Different approximations can be introduced for sake of stability guarantee, depending whether *prod* or *sum* case is adopted, and the consequent settings of their respective parameters k or ϵ. The choice is in the hands of performance analyzers, and it is typically application-based. In fact, for critical systems that require to be analyzed in proximity of their limit conditions, smaller k's or ϵ's shall be set. It is out of this paper scope to analyze applications on such basis, however in Table B.8 we report closed formulas for the errors incurred by *prod* and *sum* in our cases.

Although some cases have more complex formulas than others, it is worth noting that while considering Λ and M as uniformly distributed in the ranges (λ_L, λ_U) and (μ_L, μ_U), we are able to obtain the average number of entities and the average response time along with their variances as closed-form formulas in all cases, as claimed in Section 3. For other distributions (e.g., Normal, Hyperexponential), it is not possible to derive closed-form formulas and the double integral presented in Section 3 is solved via numerical integration.

Fig. 2 illustrates the effect of different distributions for input parameters when the five cases defined in this section are analyzed. The values of the input parameters are given in Table 1 (see Section 5). Both the input parameters follow the same distribution. The mean of all the distributions is always the same (i.e., $\bar{\lambda} = (\lambda_L + \lambda_U)/2$ and $\bar{\mu} = (\mu_L + \mu_U)/2$). However,

[2] Formulas are available for download at the following url: http://tiny.cc/c1kk8y.

Table B.3
Summary of closed formulas obtained for Var[N].

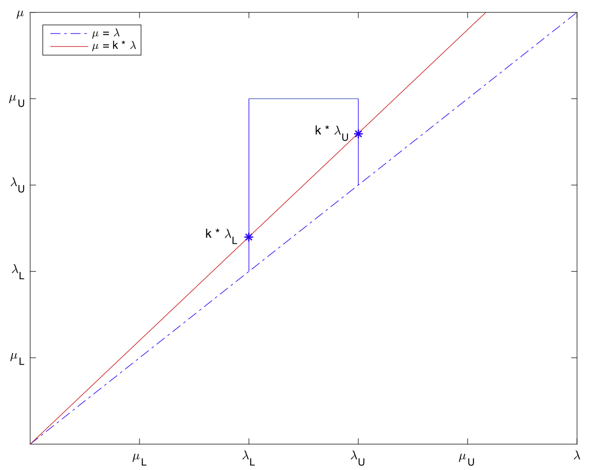
	Var[N]
1	$(\frac{(a-b)(c-d)-c^2\mathrm{Log}\left[\frac{b-c}{a-c}\right]+d^2\mathrm{Log}\left[\frac{b-d}{a-d}\right]}{(-a+b)(-c+d)} - \frac{1}{4(-a+b)^2(-c+d)^2}(-(a-b)(c-d)+c^2\mathrm{Log}[\frac{b-c}{a-c}]- b^2\mathrm{Log}[-b+c]+a^2\mathrm{Log}[\frac{a-c}{a-d}]+d^2\mathrm{Log}[-a+d]+(b-d)(b+d)\mathrm{Log}[-b+d])^2)$
2-prod	$(-\frac{\left(\left(-a^2+d^2\right)k\mathrm{Log}[-a+d]+a^2k\mathrm{Log}[a(-1+k)]-d\left(d-ak+dk\mathrm{Log}\left[\frac{d(-1+k)}{k}\right]\right)\right)^2}{(d-ak)^4}+ \frac{2k\left(\frac{(d-ak)(ak+d(-1+2k))}{2(-1+k)k}+d^2\mathrm{Log}\left[\frac{d(-1+k)}{(-a+d)k}\right]\right)}{(d-ak)^2})$
2-sum	$(-\frac{\left((a+3d-\epsilon)(a-d+\epsilon)+2(a-d)(a+d)\mathrm{Log}\left[-\frac{\epsilon}{a-d}\right]\right)^2}{4(-a+d-\epsilon)^4}+ \frac{(-a+d-\epsilon)\left(2\left(a^2+ad+d^2\right)+(a+5d)\epsilon-\epsilon^2\right)+6d^2\epsilon\mathrm{Log}\left[-\frac{\epsilon}{a-d}\right]}{3(-a+d-\epsilon)^2\epsilon})$
3-prod	$(-\frac{1}{(-c+d)^2(c+d-2ak)^2}(c^2-d^2-ack+adk+c^2k\mathrm{Log}[c]+(a^2-c^2)k\mathrm{Log}[-a+c]- d^2k\mathrm{Log}[d]-a^2k\mathrm{Log}[-a+d]+d^2k\mathrm{Log}[-a+d]+c^2k\mathrm{Log}\left[\frac{-1+k}{k}\right]-d^2k\mathrm{Log}\left[\frac{-1+k}{k}\right])^2+ \frac{c^2-d^2-2(c-d)(a+c+d)k+2a(c-d)k^2+2(-1+k)k(-c^2\mathrm{Log}[\frac{c(-1+k)}{(-a+c)k}]+d^2\mathrm{Log}[\frac{d(-1+k)}{(-a+d)k}])}{(-c+d)(-1+k)(c+d-2ak)})$
3-sum	$(-\frac{\left(2\left(a^2-c^2\right)\mathrm{Log}[-a+c]-2\left(a^2-d^2\right)\mathrm{Log}[-a+d]+(c-d)(-2a+3c+3d-4\epsilon+2(c+d)\mathrm{Log}[\epsilon])\right)^2}{4(-c+d)^2(-2a+c+d-2\epsilon)^2}+ \frac{-2c^3+2d^3-3(c-d)(-2a+c+d)\epsilon+6(c-d)\epsilon^2-6c^2\epsilon\mathrm{Log}\left[-\frac{\epsilon}{a-c}\right]+6d^2\epsilon\mathrm{Log}\left[-\frac{\epsilon}{a-d}\right]}{3(-c+d)(-2a+c+d-2\epsilon)\epsilon})$
4-prod	$(\frac{2\left(-\frac{(a-b)(2d(-1+k)+(a+b)k)}{2(-1+k)}-d^2\mathrm{Log}\left[\frac{a-d}{b-d}\right]\right)}{(-a+b)(2d-ak-bk)}- \frac{\left(ad-bd+d^2\mathrm{Log}\left[\frac{a-d}{b-d}\right]+a^2\mathrm{Log}\left[\frac{a-ak}{a-d}\right]+b^2\mathrm{Log}\left[\frac{b-d}{b-bk}\right]\right)^2}{(-a+b)^2(2d-ak-bk)^2})$
4-sum	$(\frac{-3(a-b)(a+b+2d)-\frac{2a^3}{\epsilon}+\frac{2b^3}{\epsilon}+6d^2\mathrm{Log}\left[\frac{b-d}{a-d}\right]}{3(-a+b)(-a-b+2d-2\epsilon)}- \frac{\left((a-b)(a+b+2d)+2\left(-a^2+d^2\right)\mathrm{Log}\left[\frac{-a+d}{\epsilon}\right]+2(b-d)(b+d)\mathrm{Log}\left[\frac{-b+d}{\epsilon}\right]\right)^2}{4(-a+b)^2(-a-b+2d-2\epsilon)^2})$
5-prod	$(\frac{2k\left(-\frac{(c-bk)(c+2d(-1+k)+bk)}{2(-1+k)k}+\frac{(-c+d)(c-ak)+c^2k\mathrm{Log}\left[\frac{(-a+c)k}{c(-1+k)}\right]+d^2k\mathrm{Log}\left[\frac{c-dk}{ak-dk}\right]}{k}-d^2\mathrm{Log}\left[\frac{c-dk}{bk-dk}\right]\right)}{2(-c+d)(c-ak)+(-c+2d-bk)(-c+bk)}- (4k^2(\frac{cdk-bdk^2+\left(b^2-d^2\right)k^2\mathrm{Log}[-b+d]+(-c^2+d^2k^2)\mathrm{Log}[d-\frac{c}{k}]-b^2k^2\mathrm{Log}[b(-1+k)]+c^2\mathrm{Log}[\frac{c(-1+k)}{k}]}{2k^2}+ \frac{1}{2k^2}(c^2k-cdk-ack^2+adk^2+\left(a^2-c^2\right)k^2\mathrm{Log}[-a+c]+\left(-a^2+d^2\right)k^2\mathrm{Log}[-a+d]-d^2k^2\mathrm{Log}\left[d-\frac{c}{k}\right]- c^2\mathrm{Log}[c(-1+k)]+c^2k^2\mathrm{Log}\left[\frac{c(-1+k)}{k}\right]+c^2\mathrm{Log}[-c+dk]))^2)/(2(-c+d)(c-ak)+(-c+2d-bk)(-c+bk))^2)$
5-sum	$((2(ac-c^2-ad+cd+\frac{b^3-(c-\epsilon)^3}{3\epsilon}+c\epsilon-d\epsilon+\frac{1}{2}(b+c+2d-\epsilon)(b-c+\epsilon)+c^2\mathrm{Log}[-a+c]- d^2\mathrm{Log}[-a+d]-c^2\mathrm{Log}[\epsilon]+d^2\mathrm{Log}\left[\frac{-b+d}{-c+d+\epsilon}\right]+d^2\mathrm{Log}[-c+d+\epsilon]))/(2(-c+d)(-a+c-\epsilon)+ (-b-c+2d-\epsilon)(b-c+\epsilon))-(4(\frac{1}{4}((b-c+\epsilon)(-b-c-2d+\epsilon)+2(b-d)(b+d)\mathrm{Log}[\frac{-b+d}{\epsilon}]+ 2(c-d-\epsilon)(c+d-\epsilon)\mathrm{Log}[\frac{\epsilon}{-c+d+\epsilon}])+\frac{1}{2}((-c+d)(a-c+\epsilon)-c^2\mathrm{Log}[-a+c]+a^2\mathrm{Log}[\frac{a-c}{a-d}]+d^2\mathrm{Log}[-a+d]+ (2c-\epsilon)\epsilon\mathrm{Log}[\epsilon]+(c-d-\epsilon)(c+d-\epsilon)\mathrm{Log}[-c+d+\epsilon]))^2)/(2(-c+d)(-a+c-\epsilon)+(-b-c+2d-\epsilon)(b-c+\epsilon))^2)$

each distribution has a different variance. Fig. 2(a) depicts $E[N]$ for each case and for each input parameter distribution. $E[N]$ is affected by multiple parameters: the analyzed case with the same distribution, e.g., see cases (1) and (2) with Uniform distribution; the distribution type, e.g., see case (1) when Uniform and Hyperexponential distributions are considered; the variance of the distribution, e.g., see case (4) when the input parameters follow $Normal(\mu_{unif}, \sigma_{unif})$ and $Normal(\mu_{unif}, 2{\cdot}\sigma_{unif})$ distributions. Similar considerations can be drawn for $E[R]$ depicted in Fig. 2(b).

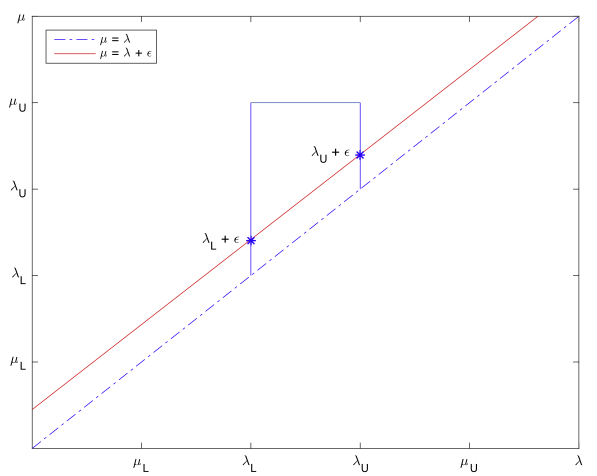
5. Application to cloud computing

Cloud applications may need to be re-deployed or migrated multiple times on different VMs to continuously comply with Service Level Agreements (SLAs). Performance of a VM may deteriorate, during its execution, due to many different factors. For example, Amazon Web Services ([3]) has recently introduced burstable instances, a new type of on-demand and low-cost instances that can burst their performance above the baseline for a limited period. In order to burst the VM performance some *CPU credits* are required. Each instance is instantiated with a given amount of initial *CPU credits*. Moreover, *CPU credits* are also generated when the VM is idle. Since the initial amount of credit is free, researchers have tried to make them last longer with different techniques [40,41] (e.g., injecting delay, throttling CPU utilization). A framework [42] has also been proposed to autonomically adapt the CPU credits consumption to the user's SLAs, and determine the time the application must be migrated to minimize SLAs violations. Another reason to re-deploy an application on a new VM is software rejuvenation [43]. Aging-related

3 aws.amazon.com.

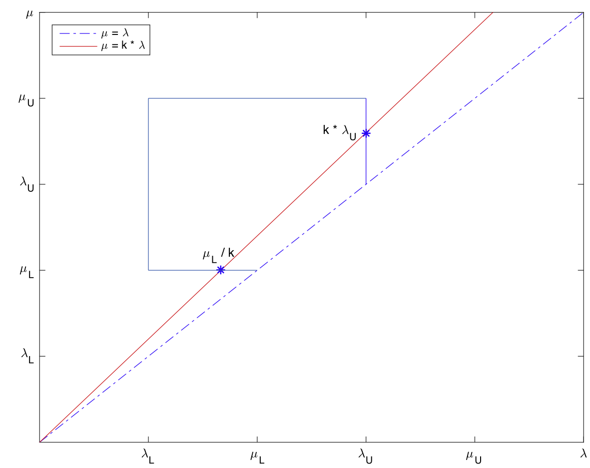


(a) Case (4)-prod: $\mu = k * \lambda$

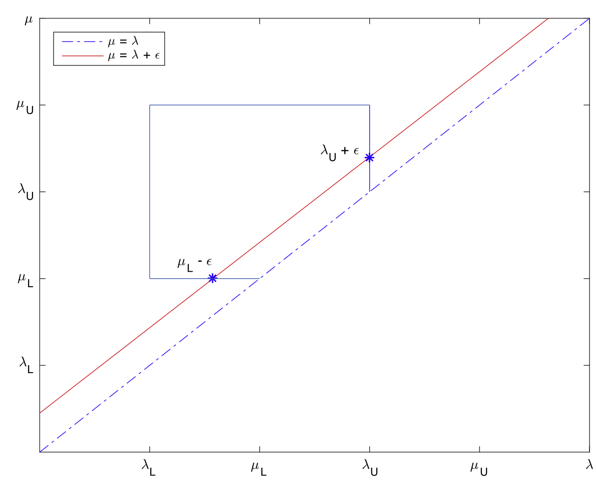


(b) Case (4)-sum: $\mu = \lambda + \epsilon$

Fig. A.8. Case (4): $\mu_L < \lambda_L < \lambda_U < \mu_U$.



(a) Case (5)-prod: $y = k * \lambda$



(b) Case (5)-sum: $y = \lambda + \epsilon$

Fig. A.9. Case (5): $\lambda_L < \mu_L < \lambda_U < \mu_U$.

bugs have been observed in many open-source cloud computing systems [44]. In [45] shutting down, rebooting and migrating a VM are investigated as three different rejuvenation techniques to reduce the impact of aging-related issues.

In order to experiment with the epistemic uncertainty propagation technique proposed in Sections 3 and 4, we use CloudSim [46], i.e., a framework that models and tests cloud computing infrastructures and services. For this purpose, given a request's interarrival and service times, CloudSim schedules the request on one of the available VM in the simulated data center. Then, it returns performance indexes (e.g., response time). We patch CloudSim to periodically change arrival and service rates of the incoming requests. In particular, these two rates follows a Uniform distribution, hence we can simulate an $M/M/1$ queue with uniform distributed arrival and service rates. The simulated cloud environment is presented in Fig. 3. Each VM is created with 1 vCPU, 512 MB memory, and 10 GB image size. Periodically, the application deployed in the cloud must be moved on new VMs to restore its original performance. When the application is migrated on new VMs, its arrival and service rates may change due to different characteristics of the hosting physical machine.

Two different VMs (i.e., VM_1 and VM_2) are instantiated in a cloud system, and they must process requests consisting of an exponentially distributed number of instructions. On the average, there are $1/\alpha$ instructions per request (i/r) in VM_1 and $1/\beta$ i/r in VM_2. The two VMs can process instructions with C_{VM1} and C_{VM2} speeds, respectively. The speed of a VM depends on the physical machines on which they are deployed. In particular, we assume to know the speed range for both the VMs, that is, $C_{\mathrm{VM1}} \sim \mathit{Uniform}(c_1^{low}, c_1^{up})$ and $C_{\mathrm{VM2}} \sim \mathit{Uniform}(c_2^{low}, c_2^{up})$. Thus, the average service time of VM_1 (VM_2) can be derived as the ratio of number of instructions $1/\alpha$ ($1/\beta$) to the VM speed c_{VM1} (c_{VM2}). Such service time follows an exponential distribution with rate $\alpha \cdot c_{\mathrm{VM1}}$ ($\beta \cdot c_{\mathrm{VM2}}$). When all the instructions of a request are executed by VM_1, the request is completed and can go to VM_2. After being processed also by the second VM, the request leaves the system. The logical representation and queueing model of the cloud system are given in Fig. 4.

To simulate the VMs migration (i.e., the change of physical constraints), we specify a `migration_time` parameter, that is the period between two migration events. In fact, after such a time the two VMs are migrated to new physical hosts, and the requests are processed with restored performance. The migration does not affect the average number of instructions each VM must serve to complete a requests. However, the VM speeds change, so their average service times do. In order to avoid VM_2 saturation we assume the cloud system is provided with an *oracle* that checks if VM_2 is faster than VM_1 after each migration. For this purpose, the oracle may adopt either the additive or the multiplicative limiting assumption (i.e., $\lambda < \mu - \epsilon$ or $\lambda < \mu/k$)

Table B.4
Summary of closed formulas obtained for E[R].

	E[R]
1	$\frac{-c\text{Log}\left[\frac{a-c}{b-c}\right]+a\text{Log}[-a+c]-b\text{Log}[-b+c]+d\text{Log}\left[\frac{a-d}{b-d}\right]-a\text{Log}[-a+d]+b\text{Log}[-b+d]}{(a-b)(c-d)}$
2-prod	$\frac{2k\left((-a+d)\text{Log}[-a+d]+a\text{Log}[a(-1+k)]-d\text{Log}\left[\frac{d(-1+k)}{k}\right]\right)}{(d-ak)^2}$
2-sum	$(\frac{2\left(a-d+\epsilon+(a-d)\text{Log}\left[-\frac{\epsilon}{a-d}\right]\right)}{(a-d+\epsilon)^2})$
3-prod	$-\frac{2k\left(c\text{Log}[c]-d\text{Log}[d]+a\text{Log}\left[\frac{(-a+c)k}{-1+k}\right]+c\text{Log}\left[\frac{1-k}{ak-ck}\right]+a\text{Log}\left[\frac{1-k}{ak-dk}\right]-d\text{Log}\left[\frac{1-k}{ak-dk}\right]\right)}{(c-d)(c+d-2ak)}$
3-sum	$(\frac{2\left(-c+d-a\text{Log}[-a+c]+(a-d)\text{Log}[-a+d]+c\text{Log}\left[\frac{-a+c}{\epsilon}\right]+d\text{Log}[\epsilon]\right)}{(c-d)(-2a+c+d-2\epsilon)})$
4-prod	$-\frac{2\left(d\text{Log}\left[\frac{b-d}{a-d}\right]+a\text{Log}\left[\frac{a-d}{a-ak}\right]+b\text{Log}\left[\frac{b-bk}{b-d}\right]\right)}{(a-b)(-2d+(a+b)k)}$
4-sum	$(-\frac{2\left(-a+b+d\text{Log}\left[\frac{b-d}{a-d}\right]+a\text{Log}\left[\frac{-a+d}{\epsilon}\right]+b\text{Log}\left[-\frac{\epsilon}{b-d}\right]\right)}{(a-b)(a+b-2d+2\epsilon)})$
5-prod	$((2k(\frac{(b-d)k\text{Log}[-b+d]+(-c+dk)\text{Log}\left[d-\frac{c}{k}\right]-bk\text{Log}[b(-1+k)]+c\text{Log}[\frac{c(-1+k)}{k}]}{k}+$ $\frac{(a-c)k\text{Log}[-a+c]-ak\text{Log}[-a+d]-c\text{Log}[c(-1+k)]+ck\text{Log}[c(-1+k)]-ck\text{Log}[k]+dk\text{Log}\left[\frac{(-a+d)k}{-c+dk}\right]+c\text{Log}[-c+dk]}{k}))$ $/(2(-c+d)(c-ak)+(-c+2d-bk)(-c+bk)))$
5-sum	$((2(-b+c-\epsilon-c\text{Log}[-a+c]+a\text{Log}[\frac{a-c}{a-d}]-d\text{Log}[-b+d]+b\text{Log}[\frac{-b+d}{\epsilon}]+$ $(c-\epsilon)\text{Log}[\epsilon]+\epsilon\text{Log}[\epsilon]+d\text{Log}[\frac{-a+d}{-c+d+\epsilon}]+(c-\epsilon)\text{Log}[-c+d+\epsilon]+$ $(-c+d+\epsilon)\text{Log}[-c+d+\epsilon]))/(2(-c+d)(-a+c-\epsilon)+(-b-c+2d-\epsilon)(b-c+\epsilon)))$

to add a safety margin to the stability condition. If VM_2 has been deployed on a physical host that is slower than the VM_1's one, then the oracle immediately migrates both the VMs. This way, the system only operates in the five areas described in Section 4.

In order to derive the point estimates of the output measures and their confidence intervals, we use the regenerative simulation [47]. It consists in dividing each simulation run into a series of cycles such that, from a probabilistic point of view, the system evolution is the same in any cycle.

For the sake of clarity, in this section we analyze the performance output measures of only VM_2. Fig. 5 compares simulation and analytic results for two of the five cases identified in Section 4 (i.e., case 1 and case 5). We set the same average number of instructions per request for both cases, that is, $1/\alpha = 1/\beta = 400000$, whereas the period between two migrations of the VMs (i.e., the `migration_time` parameter) is 30 and 60 min for case 1 and case 5, respectively.

In Fig. 5, top graphs depict case 1 – that is, the case in which arrival rate is always smaller than service rate – is considered, and the VM speeds are distributed as follows: $C_{VM1} \sim Uniform(3.6\ r/s, 4.5\ r/s)$ and $C_{VM2} \sim Uniform(5.49\ r/s, 6.57\ r/s)$. Additive limitation is adopted by the oracle of the cloud system, and $\epsilon = 0.36\ r/s$. Using the formulas shown in Appendix B, we can analytically derive $E[N] = 2.161\ r$ and $E[R] = 0.529\ s$ (see Tables B.2 and B.4, respectively). Such values are represented in the two case 1 graphs in Fig. 5 by dashed lines. During a 12-hours long simulation, we collect the average values of the output measures (red line) and their 95% confidence intervals (blue lines). In order to estimate the goodness of our analytic formulas with respect to the results obtained through simulation we compute the Mean Absolute Percentage Error — MAPE ([4]) as:

$$MAPE_N = \frac{|N_{analytic} - N_{simulation}|}{N_{simulation}}$$

where $N_{analytic}$ is the $E[N]$ derived with our methodology and $N_{simulation}$ is the average number of customers in the system estimated via simulation. $MAPE_R$ is defined in a similar way. For case 1, $MAPE_N = 2.40\%$ and $MAPE_R = 1.20\%$.

Bottom graphs of Fig. 5 depict case 5 results. $C_{VM1} \sim Uniform(0.198\ r/s, 0.603\ r/s)$ and $C_{VM2} \sim Uniform(0.549\ r/s, 0.657\ r/s)$ and we simulate 246 h to get more accurate results since the oracle adopts a smaller ϵ (i.e., $\epsilon = 0.036\ r/s$) ([5]). The analytic values obtained with our formulas are $E[N] = 2.91\ r$ and $E[R] = 6.48\ s$, and the percentage errors with respect to the simulation are $MAPE_N = 4.43\%$ and $MAPE_R = 12.26\%$. The comparison is also performed for the other three cases. For the sake of brevity, we report only numerical values in Table 1.

The analysis of the five cases shows that our methodology is providing accurate results with respect to parameters estimation performed via simulation. In particular, the error made by our methodology is never larger than 15%. Moreover, it is faster than any simulation since it works with closed-form equations. Instead, the time required to complete the CloudSim simulations on a private cloud environment ([6]) takes from 12 to 24 h depending on the values of input parameters.

[4] Note that the denominator is intentionally represented by the simulative results, since we are interested to study the goodness of our methodology w.r.t. the results of the simulation.

[5] Note that such a smaller ϵ is required to make the cloud system work in the desired case and to not incur in the problem, illustrated in Section 4, of overcoming the closest vertex of the integration region.

[6] policloud.polimi.it.

Table B.5
Summary of closed formulas obtained for Var[R].

	Var[R]
1	$\left(\frac{\mathrm{Log}\left[\frac{(a-c)(b-d)}{(b-c)(a-d)}\right]}{(-a+b)(-c+d)}-\frac{(-c\mathrm{Log}\left[\frac{a-c}{b-c}\right]+a\mathrm{Log}[-a+c]-b\mathrm{Log}[-b+c]+d\mathrm{Log}\left[\frac{a-d}{b-d}\right]-a\mathrm{Log}[-a+d]+b\mathrm{Log}[-b+d])^2}{(-a+b)^2(-c+d)^2}\right)$
2-prod	$\left(-\frac{4k^2\left((-a+d)\mathrm{Log}[-a+d]+a\mathrm{Log}[a(-1+k)]-d\mathrm{Log}\left[\frac{d(-1+k)}{k}\right]\right)^2}{(d-ak)^4}+\frac{2k(-\mathrm{Log}[-a+d]+\mathrm{Log}[a(-1+k)]+\frac{k(\mathrm{Log}[d]-\mathrm{Log}[ak])}{-1+k})}{(d-ak)^2}\right)$
2-sum	$\left(-\frac{4\left(a-d+\epsilon-a\mathrm{Log}[-a+d]+d\mathrm{Log}\left[\frac{-a+d}{\epsilon}\right]+a\mathrm{Log}[\epsilon]\right)^2}{(-a+d-\epsilon)^4}-\frac{2(a-d+\epsilon+\epsilon\mathrm{Log}[-a+d]-\epsilon\mathrm{Log}[\epsilon])}{(-a+d-\epsilon)^2\epsilon}\right)$
3-prod	$\left(\frac{2k\left(\mathrm{Log}\left[\frac{a-c}{a-d}\right]+\frac{k\mathrm{Log}\left[\frac{d}{c}\right]}{-1+k}\right)}{(-c+d)(c+d-2ak)}-\frac{1}{(-c+d)^2(c+d-2ak)^2}4k^2(-c+d+c\mathrm{Log}[c]+(a-c)\mathrm{Log}[-a+c]-d\mathrm{Log}[d]+(-a+d)\mathrm{Log}[-a+d]+(c-d)(1+\mathrm{Log}[-1+k]-\mathrm{Log}[k]))^2\right)$
3-sum	$\left(\frac{2\left(-c+d+\epsilon\mathrm{Log}\left[\frac{a-c}{a-d}\right]\right)}{(-c+d)(-2a+c+d-2\epsilon)\epsilon}-\frac{4\left(c-d+(a-c)\mathrm{Log}[-a+c]-a\mathrm{Log}[-a+d]+d\mathrm{Log}\left[\frac{-a+d}{\epsilon}\right]+c\mathrm{Log}[\epsilon]\right)^2}{(-c+d)^2(-2a+c+d-2\epsilon)^2}\right)$
4-prod	$\left(\frac{2\left(\frac{\mathrm{Log}\left[\frac{b}{a}\right]}{-1+k}+\mathrm{Log}\left[\frac{b-d}{a-d}\right]\right)}{(-a+b)(2d-ak-bk)}-\frac{4(-a+b+a\mathrm{Log}[a]-b\mathrm{Log}[b]+(-a+d)\mathrm{Log}[-a+d]+(b-d)\mathrm{Log}[-b+d]+(a-b)(1+\mathrm{Log}[-1+k]))^2}{(-a+b)^2(2d-ak-bk)^2}\right)$
4-sum	$\left(\frac{2\left(-a+b+\epsilon\mathrm{Log}\left[\frac{b-d}{a-d}\right]\right)}{(-a+b)(-a-b+2d-2\epsilon)\epsilon}-\frac{4\left(a-b+(-a+d)\mathrm{Log}[-a+d]-d\mathrm{Log}[-b+d]+b\mathrm{Log}\left[\frac{-b+d}{\epsilon}\right]+a\mathrm{Log}[\epsilon]\right)^2}{(-a+b)^2(-a-b+2d-2\epsilon)^2}\right)$
5-prod	$\left(\frac{2k\left(\mathrm{Log}\left[\frac{a-c}{a-d}\right]+\frac{\mathrm{Log}\left[\frac{bk}{c}\right]}{-1+k}+\mathrm{Log}\left[\frac{c-dk}{c-ck}\right]+\mathrm{Log}\left[\frac{(-b+d)k}{-c+dk}\right]\right)}{2(-c+d)(c-ak)+(-c+2d-bk)(-c+bk)}-(4k^2(\frac{(b-d)k\mathrm{Log}[-b+d]+(-c+dk)\mathrm{Log}[d-\frac{c}{k}]-bk\mathrm{Log}[b(-1+k)]+c\mathrm{Log}[\frac{c(-1+k)}{k}]}{k}+\frac{(a-c)k\mathrm{Log}[-a+c]-ak\mathrm{Log}[-a+d]-c\mathrm{Log}[c(-1+k)]+ck\mathrm{Log}[c(-1+k)]-ck\mathrm{Log}[k]+dk\mathrm{Log}[\frac{(-a+d)k}{-c+dk}]+c\mathrm{Log}[-c+dk]}{k})^2)/(2(-c+d)(c-ak)+(-c+2d-bk)(-c+bk))^2\right)$
5-sum	$(-((4(-b+c-\epsilon-c\mathrm{Log}[-a+c]+a\mathrm{Log}\left[\frac{a-c}{a-d}\right]-d\mathrm{Log}[-b+d]+b\mathrm{Log}\left[\frac{-b+d}{\epsilon}\right]+(c-\epsilon)\mathrm{Log}[\epsilon]+\epsilon\mathrm{Log}[\epsilon]+d\mathrm{Log}\left[\frac{-a+d}{-c+d+\epsilon}\right]+(c-\epsilon)\mathrm{Log}[-c+d+\epsilon]+(-c+d+\epsilon)\mathrm{Log}[-c+d+\epsilon])^2)/(2(-c+d)(-a+c-\epsilon)+(-b-c+2d-\epsilon)(b-c+\epsilon))^2)+\frac{2\left(\mathrm{Log}\left[\frac{a-c}{a-d}\right]+\frac{b-c+\epsilon+\epsilon\mathrm{Log}\left[\frac{-b+d}{-c+d+\epsilon}\right]}{\epsilon}+\mathrm{Log}\left[\frac{-c+d+\epsilon}{\epsilon}\right]\right)}{2(-c+d)(-a+c-\epsilon)+(-b-c+2d-\epsilon)(b-c+\epsilon)})$

6. Conclusions and future directions

In this paper, we have derived basic equations to deal with the epistemic uncertainty propagation on the steady state mean number of entities in the system and the steady state mean response time. The problem is cast as a two-dimensional integration after multiplication by the joint density of the two parameters, namely, the arrival rate Λ and the service rate M. With the assumption the uniform distribution of the two parameters we obtain the results in a closed-form. We have also shown that with the assumption normal densities, the integration can be carried out numerically. Even in the seemingly simple case of uniform densities, difficulty in the integration arises from the values of the integrand that can become very large leading to numerical instability. We proposed two different limiting assumptions to avoid the numerical instability. The equations developed here can be directly applied to a tandem network of M/M/1 queues since the average number of entities in the system and the average response time are additive in this case [15,48]. To demonstrate the applicability of our approach, we presented the performance analysis of a cloud-based application that is continuously migrated onto different VMs. This migration implies that the speed of the VM changes leading to epistemic uncertainty in the performance characteristics of the application. Results assess that our analytical formulas accurately capture the propagation of uncertainty from input parameters to output measures. To further assess the validity of our formulation, we are interested to investigate preemptible cloud resources [33] in the near future. This implies that we set the lower bound of service rates M to zero, thus to model the risk of having resources that are claimed back by the cloud provider. To consider any open product-form network, we also plan to compare these results while considering further types of the densities, e.g., Erlang. Moreover, we aim to generalize our equations to the cases of M/M/m, M/G/1, and G/M/1 queues, for which closed form analytical solutions exist. Then we also plan to handle the uncertainty in the branching probabilities, as branches are allowed in Jackson networks [15, 48].

Table B.6
Summary of closed formulas obtained for confidence intervals on N.

	$Prob(N > n)$	Range of n
All cases	1	$n \leq \frac{a}{d-a}$
1	$1 - \frac{[(d-a)n-a]^2}{2n(n+1)(d-c)(b-a)}$	$\frac{a}{d-a} < n \leq \frac{a}{c-a}$
$\left[\frac{d}{c} \leq \frac{b}{a}\right]$	$\frac{b}{b-a} - \frac{(d+c)n}{2(b-a)(n+1)}$	$\frac{a}{c-a} < n \leq \frac{b}{d-b}$
	$\frac{[(c-b)n-b]^2}{2n(n+1)(d-c)(b-a)}$	$\frac{b}{d-b} < n \leq \frac{b}{c-b}$
	0	$n > \frac{b}{c-b}$
1	$1 - \frac{[(d-a)n-a]^2}{2n(n+1)(d-c)(b-a)}$	$\frac{a}{d-a} < n \leq \frac{b}{d-b}$
$\left[\frac{d}{c} > \frac{b}{a}\right]$	$\frac{(a+b)(n+1)}{2(d-c)n} - \frac{c}{d-c}$	$\frac{b}{d-b} < n \leq \frac{a}{c-a}$
	$\frac{[(c-b)n-b]^2}{2n(n+1)(d-c)(b-a)}$	$\frac{a}{c-a} < n \leq \frac{b}{c-b}$
	0	$n > \frac{b}{c-b}$
2	$\frac{nd^2-(n+1)a^2}{n(n+1)(d-a)^2}$	$n > \frac{a}{d-a}$
3	$\frac{n(d^2-a^2)-n(n+1)(c-a)^2-a^2}{n(n+1)(d-c)(d+c-2a)}$	$\frac{a}{d-a} < n \leq \frac{a}{c-a}$
	$\frac{c+d}{(n+1)(d+c-2a)}$	$n > \frac{a}{c-a}$
4	$\frac{n(2bd-a^2-b^2)-n^2(d-b)^2-a^2}{n(n+1)(b-a)(2d-b-a)}$	$\frac{a}{d-a} < n \leq \frac{b}{d-b}$
	$\frac{b+a}{n(2d-b-a)}$	$n > \frac{b}{d-b}$
5	$x = 2\left(bd - da + ac - \frac{b^2}{2} - \frac{c^2}{2}\right)$	
$\left[\frac{d}{c} \leq \frac{b}{a}\right]$	$\frac{2(ac+bd)-(b^2+c^2)}{x} - \frac{[(n+1)a^2]}{nx} - \frac{nd^2}{(n+1)x}$	$\frac{a}{d-a} < n \leq \frac{a}{c-a}$
	$\frac{1}{x}\frac{(2db-b^2-c^2)-n(b-d)^2}{(n+1)}$	$\frac{a}{c-a} < n \leq \frac{b}{d-b}$
	$\frac{1}{x}\frac{n(b^2-c^2)+b^2}{n(n+1)}$	$n > \frac{b}{d-b}$
5	$x = 2\left(bd - da + ac - \frac{b^2}{2} - \frac{c^2}{2}\right)$	
$\left[\frac{d}{c} > \frac{b}{a}\right]$	$\frac{2(ac+bd)-(b^2+c^2)}{x} - \frac{[(n+1)a^2]}{nx} - \frac{nd^2}{(n+1)x}$	$\frac{a}{d-a} < n \leq \frac{b}{d-b}$
	$\frac{1}{x}\frac{(b^2-a^2)(n+1)-n[2c(b-a)+(b-c)^2]}{n}$	$\frac{b}{d-b} < n \leq \frac{a}{c-a}$
	$\frac{1}{x}\frac{n(b^2-c^2)+b^2}{n(n+1)}$	$n > \frac{a}{c-a}$

Declaration of competing interest

The authors declare that there is no conflict of interest for this research.

Appendix A. Representation of the five possible cases

The graphical representations of the cases (1), (2), (4) and (5) identified in Section 4 are shown in Figs. A.6, A.7, A.8 and A.9, respectively.

As already said, case (1) is the only one where the integration area does not intersect the line $\lambda = \mu$, thus the integral on the area is calculated without further assumption. Instead, in all the other cases a limitation assumption must be adopted (i.e., k-limitation referred to as *prod* and ϵ-limitation referred to as *sum*).

As for case (3) – discussed in Section 4 and depicted in Fig. 1 – the points of intersection between the integration areas and the line $\lambda = \mu$ are then used to compute $E[N]$, $E[R]$, $var[N]$ and $Var[R]$.

Appendix B. List of formulas for the uniform case

For the sake of simplicity we adopt here the following variable renaming: $\lambda_L \mapsto a$, $\lambda_U \mapsto b$, $\mu_L \mapsto c$, $\mu_U \mapsto d$.

Table B.2 reports the formulas obtained for $E[N]$, Table B.3 the ones for $Var[N]$ and, similarly, Tables B.4 and B.5 for $E[R]$ and $Var[R]$, respectively.

Table B.2 is structured as follows. The first column lists all the cases presented above and, except for case (1), all cases include the *prod* and *sum* scenarios that determine different limits in the calculation of the integral. The second column of the Table reports the outputs obtained while calculating $E[N]$. For example, case (1) is obtained by calculating the expectation of $\frac{\Lambda}{M-\Lambda}$ and assuming that Λ and M are both uniformly distributed on the interval $[a, b]$ and $[c, d]$ and their joint probability density is given by $\frac{1}{b-a}\frac{1}{d-c}$.

Table B.7
Summary of closed formulas obtained for confidence intervals on R.

	$Prob(R > r)$	Range of r
All cases	1	$r \leq \frac{1}{d-a}$
1	$1 - \frac{\left(d-a-\frac{1}{r}\right)^2}{2(d-c)(b-a)}$	$\frac{1}{d-a} < r \leq \frac{1}{c-a}$
$[d-c \leq b-a]$	$\frac{\left[2\left(b+\frac{1}{r}\right)-(d+c)\right]}{2(b-a)}$	$\frac{1}{c-a} < r \leq \frac{1}{d-b}$
	$\frac{\left(b+\frac{1}{r}-c\right)^2}{2(d-c)(b-a)}$	$\frac{1}{d-b} < r \leq \frac{1}{c-b}$
	0	$r > \frac{1}{c-b}$
1	$1 - \frac{\left(d-a-\frac{1}{r}\right)^2}{2(d-c)(b-a)}$	$\frac{1}{d-a} < r \leq \frac{1}{d-b}$
$[d-c > b-a]$	$\frac{\left[2\left(\frac{1}{r}-c\right)+(a+b)\right]}{2(d-c)}$	$\frac{1}{d-b} < r \leq \frac{1}{c-a}$
	$\frac{\left(b+\frac{1}{r}-c\right)^2}{2(d-c)(b-a)}$	$\frac{1}{c-a} < r \leq \frac{1}{c-b}$
	0	$r > \frac{1}{c-b}$
2	$\frac{1}{r}\frac{\left(2d-2a-\frac{1}{r}\right)}{(d-a)^2}$	$r > \frac{1}{d-a}$
3	$\frac{\left[2\frac{(d-a)}{r}-(c-a)^2-\frac{1}{r^2}\right]}{(d-c)(d+c-2a)}$	$\frac{1}{d-a} < r \leq \frac{1}{c-a}$
	$\frac{2}{r(d+c-2a)}$	$r > \frac{1}{c-a}$
4	$\frac{\left(b+d-\frac{1}{r}-2a\right)\left(b-d+\frac{1}{r}\right)+2(d-b)(b-a)}{(b-a)(2d-b-a)}$	$\frac{1}{d-a} < r \leq \frac{1}{d-b}$
	$\frac{2}{r(2d-a-b)}$	$r > \frac{1}{d-b}$
5	$x = 2(b-a)(d-c)-(b-c)^2$	
$[d-c \leq b-a]$	$\frac{1}{x}\left[\frac{2}{r}(d-a)-(d-b)^2-(c-a)^2-\frac{1}{r^2}\right]$	$\frac{1}{d-a} < r \leq \frac{1}{c-a}$
	$\frac{1}{x}\left[\frac{2}{r}(d-c)-(b-d)^2\right]$	$\frac{1}{c-a} < r \leq \frac{1}{d-b}$
	$\frac{1}{x}\frac{1}{r}\left[2(b-c)+\frac{1}{r}\right]$	$r > \frac{1}{d-b}$
5	$x = 2(b-a)(d-c)-(b-c)^2$	
$[d-c > b-a]$	$\frac{1}{x}\left[\frac{2}{r}(d-a)-(d-b)^2-(c-a)^2-\frac{1}{r^2}\right]$	$\frac{1}{d-a} < r \leq \frac{1}{d-b}$
	$1-\frac{1}{x}\left[2(d-\frac{1}{r})-(a+b)\right](b-a)$	$\frac{1}{d-b} < r \leq \frac{1}{c-a}$
	$\frac{1}{x}\frac{1}{r}\left[2(b-c)+\frac{1}{r}\right]$	$r > \frac{1}{c-a}$

$$\frac{-(a-b)(c-d)+c^2\mathrm{Log}\left[\frac{b-c}{a-c}\right]-b^2\mathrm{Log}[-b+c]+a^2\mathrm{Log}\left[\frac{a-c}{a-d}\right]+d^2\mathrm{Log}[-a+d]+(b-d)(b+d)\mathrm{Log}[-b+d]}{2(a-b)(c-d)}$$

Box I.

$$\frac{-c\mathrm{Log}\left[\frac{a-c}{b-c}\right]+a\mathrm{Log}[-a+c]-b\mathrm{Log}[-b+c]+d\mathrm{Log}\left[\frac{a-d}{b-d}\right]-a\mathrm{Log}[-a+d]+b\mathrm{Log}[-b+d]}{(a-b)(c-d)}$$

Box II.

Table B.8
Summary of closed formulas obtained for errors.

	error
2-prod	$\frac{(-1+k)\left(d^2-\mathrm{ka}^2\right)}{(-a+d)^2 k}$
2-sum	$\frac{(-2a+2d-\epsilon)\epsilon}{(-a+d)^2}$
3-prod	$\frac{(c+d)(-1+k)}{(-2a+c+d)k}$
3-sum	$\frac{2\epsilon}{-2a+c+d}$
4-prod	$\frac{(k-1)(b+a)}{(2d-b-a)}$
4-sum	$\frac{2\epsilon}{(2d-a-b)}$
5-prod	$\frac{c^2(1-k)+b^2(-1+k)k}{-(b-c)^2 k+2(-a+b)(-c+d)k}$
5-sum	$\frac{\epsilon(2b-2c+\epsilon)}{2(b-a)(d-c)-(d-c)^2}$

The assumption of case (1) is that the following relationship holds, i.e., $a < b < c < d$, and the outcome of the average number of customers (i.e., $E[N]$) calculated with such bounds results to be given in Box I, as reported in Table B.2.

Similarly Table B.3 reports the outputs obtained while calculating $Var[N]$.

For example, case (1) is obtained by applying the formula of the variance related to the value $\frac{\lambda}{\mu-\lambda}$. The same assumptions used for calculating $E[N]$ hold here, in fact we assume that λ and μ are both uniformly distributed on the interval $[a, b]$ and $[c, d]$ and their joint probability density is given by $\frac{1}{b-a}$ and $\frac{1}{d-c}$. We recall that for case (1) the lower and bounds have the following ordering, i.e., $a < b < c < d$. The outcome of the variance calculation on the average number of customers (i.e., $Var[N]$) results to be:

$$\frac{(a-b)(c-d)-c^2\mathrm{Log}\left[\frac{b-c}{a-c}\right]+d^2\mathrm{Log}\left[\frac{b-d}{a-d}\right]}{(-a+b)(-c+d)} - \frac{1}{4(-a+b)^2(-c+d)^2}(-(a-b)(c-d)+c^2\mathrm{Log}[\frac{b-c}{a-c}]-$$

$$b^2\mathrm{Log}[-b+c]+a^2\mathrm{Log}[\frac{a-c}{a-d}]+d^2\mathrm{Log}[-a+d]$$
$$+(b-d)(b+d)\mathrm{Log}[-b+d])^2$$

as reported in Table B.3. Also for the calculation of $Var[N]$, there are some cases where the formulas appear to be more elaborated than others, for example both the *5-prod* and *5-sum* scenarios are associated with a quite complex formula. However, it is worth to remark that in all cases we are able to derive a closed formula that allows us to calculate the actual values of $Var[N]$.

Instead, Tables B.4 and B.5 report closed-form formulas for $E[R]$ and $Var[R]$, respectively, for all cases.

Similarly to Table B.2, Table B.4 reports the outputs obtained while calculating $E[R]$. For example, case (1) is obtained by applying the formula of the average response time of the system to the value $\frac{1}{\mu-\lambda}$. The same assumptions used for calculating $E[N]$ hold here, in fact we assume that λ and μ are both uniformly distributed on the interval $[a, b]$ and $[c, d]$ and their joint probability density is given by $\frac{1}{b-a}$ and $\frac{1}{d-c}$. We recall that for case (1) the lower and bounds have the following ordering, i.e., $a < b < c < d$. The outcome of the average system response time calculation (i.e., $E[R]$) results to be given in Box II, as reported in Table B.4.

Also for the calculation of $E[R]$, we found that the formulas for *5-prod* and *5-sum* appear to be more complex than others. However, also for the calculation of $E[R]$ all cases are expressed by means of closed formulas that allow us to calculate the actual values of $E[R]$.

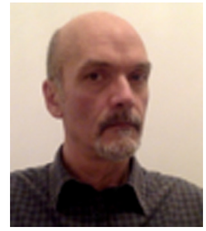
In Tables B.6 and B.7 we reported the formulas of $Prob(X > x)$ where X is N and R, respectively, while varying the range of x. In the first row of each table we reported the condition under which $Prob(X > x)$ is equal to one, because such condition holds in all the other cases.

Finally, the formulas to compute the error made by ϵ and k limitations – when the input parameters have Uniform density – are reported in Table B.8.

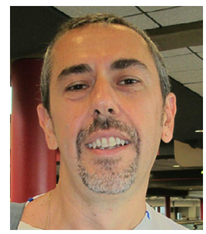
References

[1] E. Incerto, M. Tribastone, C. Trubiani, Software performance self-adaptation through efficient model predictive control, in: International Conference on Automated Software Engineering, ASE, 2017, pp. 485–496.
[2] E. Gelenbe, C. Rosenberg, Queues with slowly varying arrival and service processes, Manage. Sci. 36 (8) (1990) 928–937.
[3] C. Rosenberg, R. Mazumdar, L. Kleinrock, On the analysis of exponential queuing systems with randomly changing arrival rates: stability conditions and finite buffer scheme with a resume level, Perform. Eval. 11 (4) (1990) 283–292.
[4] O.J. Boxma, I. Kurkova, The M/M/1 queue in a heavy-tailed random environment, Stat. Neerl. 54 (2) (2000) 221–236.
[5] I.L. Mitrany, B. Avi-Itzhak, A many-server queue with service interruptions, Oper. Res. 16 (3) (1968) 628–638.
[6] H. Takagi, M/G/1//N Queues with server vacations and exhaustive service, Oper. Res. 42 (5) (1994) 926–939.
[7] A. Rindos, S. Woolet, I. Viniotis, K. Trivedi, Exact methods for the transient analysis of nonhomogeneous continuous time Markov chains, in: Computations with Markov Chains, Springer, 1995, pp. 121–133.
[8] S. Garg, A. Puliafito, M. Telek, K. Trivedi, Analysis of preventive maintenance in transactions based software systems, IEEE Trans. Comput. 47 (1) (1998).
[9] K. Trivedi, A. Bobbio, Reliability and Availability Engineering: Modeling, Analysis, and Applications, Cambridge University Press, 2017.
[10] K. Mishra, K.S. Trivedi, Uncertainty propagation through software dependability models, in: International Symposium on Software Reliability Engineering, ISSRE, 2011, pp. 80–89.
[11] K. Mishra, K.S. Trivedi, Closed-form approach for epistemic uncertainty propagation in analytic models, in: Stochastic Reliability and Maintenance Modeling, Springer, 2013, pp. 315–332.
[12] K. Mishra, K. Trivedi, R. Some, Uncertainty analysis of the remote exploration and experimentation system, J. Spacecraft Rockets 49 (6) (2012) 1032–1042.
[13] R. Pinciroli, A. Bobbio, C. Bolchini, D. Cerotti, M. Gribaudo, A. Miele, K. Trivedi, Epistemic uncertainty propagation in a weibull environment for a two-core system-on-chip, in: International Conference on System Reliability and Safety, ICSRS, 2017, pp. 516–520.
[14] C. Trubiani, I. Meedeniya, V. Cortellessa, A. Aleti, L. Grunske, Model-based performance analysis of software architectures under uncertainty, in: International Conference on Quality of Software Architectures, QoSA, 2013, pp. 69–78.
[15] K. Trivedi, Probability & Statistics with Reliability, Queueing & Computer Science applications, second ed., John Wiley & Sons, 2016.
[16] D. Cerotti, M. Gribaudo, P. Piazzolla, G. Serazzi, Flexible CPU provisioning in clouds: a new source of performance unpredictability, in: International Conference on Quantitative Evaluation of Systems, QEST, 2012, pp. 230–237.
[17] M. Sallak, S. Destercke, W. Schön, F. Vanderhaegen, D. Berdjag, C. Simon, Uncertainty, elicitation of experts' opinion, and human failures: Challenges for RAM analysis of ERTMS SoS, in: International Conference on System of Systems Engineering, SoSE, 2015, pp. 88–93.
[18] M. Paté-Cornell, Uncertainties in risk analysis: Six levels of treatment, Reliab. Eng. Syst. Saf. 54 (2–3) (1996) 95–111.
[19] H.-R. Bae, R.V. Grandhi, R.A. Canfield, Epistemic uncertainty quantification techniques including evidence theory for large-scale structures, Comput. Struct. 82 (13) (2004) 1101–1112.
[20] B. Takhedmit, K. Abbas, A parametric uncertainty analysis method for queues with vacations, J. Comput. Appl. Math. 312 (2017) 143–155.
[21] S. Lopatatzidis, J. De Bock, G. De Cooman, S. De Vuyst, J. Walraevens, Robust queueing theory: an initial study using imprecise probabilities, Queueing Syst. 82 (1–2) (2016) 75.

[22] L. Bortolussi, N. Gast, Mean field approximation of uncertain stochastic models, in: International Conference on Dependable Systems and Networks, DSN, 2016, pp. 287–298.
[23] D. Škulj, Discrete time Markov chains with interval probabilities, Int. J. Approx. Reason. 50 (8) (2009) 1314–1329.
[24] V. Bulitko, D.C. Wilkins, Qualitative simulation of temporal concurrent processes using time interval Petri nets, Artificial Intelligence 144 (1–2) (2003) 95–124.
[25] P. Baumgärtel, G. Endler, A.M. Wahl, R. Lenz, Inverse uncertainty propagation for demand driven data acquisition, in: International Conference on Simulation, 2014, pp. 710–721.
[26] F. Baccelli, B. Kauffmann, D. Veitch, Inverse problems in queueing theory and internet probing, Queueing Syst. 63 (1) (2009) 59–107.
[27] N.D. Singpurwalla, Reliability and Risk: A Bayesian Perspective, John Wiley & Sons, 2006.
[28] M. Gribaudo, R. Pinciroli, K. Trivedi, Epistemic uncertainty propagation in power models, Electron. Notes Theoret. Comput. Sci. 337 (2018) 67–86.
[29] R. Pinciroli, K. Trivedi, A. Bobbio, Parametric sensitivity and uncertainty propagation in dependability models, in: International Conference on Performance Evaluation Methodologies and Tools, 2017, pp. 44–51.
[30] I. Meedeniya, I. Moser, A. Aleti, L. Grunske, Architecture-based reliability evaluation under uncertainty, in: International Conference on the Quality of Software Architectures, QoSA, 2011, pp. 85–94.
[31] D.J. Dubois, G. Casale, Autonomic provisioning and application mapping on spot cloud resources, in: International Conference on Autonomic Computing, ICAC, 2015, pp. 57–68.
[32] D.J. Dubois, C. Trubiani, G. Casale, Model-driven application refactoring to minimize deployment costs in preemptible cloud resources, in: International Conference on Cloud Computing, CLOUD, 2016, pp. 335–342.
[33] J. Li, M. Qiu, Z. Ming, G. Quan, X. Qin, Z. Gu, Online optimization for scheduling preemptable tasks on IaaS cloud systems, J. Parallel Distrib. Comput. 72 (5) (2012) 666–677.
[34] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, John Wiley & Sons, 2008.
[35] J.C. Doyle, J.E. Wall, G. Stein, Performance and robustness analysis for structured uncertainty, in: IEEE Conference on Decision and Control, 1982.
[36] D. Meisner, J. Wu, T.F. Wenisch, Bighouse: A simulation infrastructure for data center systems, in: International Symposium on Performance Analysis of Systems and Software, ISPASS, 2012, pp. 35–45.
[37] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, A.Y. Zomaya, Energy-efficient data replication in cloud computing datacenters, Cluster Comput. 18 (1) (2015) 385–402.
[38] X. Nan, Y. He, L. Guan, Optimal resource allocation for multimedia cloud based on queuing model, in: International Workshop on Multimedia Signal Processing, MMSP, 2011, pp. 1–6.
[39] Y. Xia, M. Zhou, X. Luo, S. Pang, Q. Zhu, A stochastic approach to analysis of energy-aware DVS-enabled cloud datacenters, IEEE Trans. Syst. Man Cybern. Syst. 45 (1) (2015) 73–83.
[40] J. Wen, L. Lu, G. Casale, E. Smirni, Less can be more: Micro-managing VMs in Amazon EC2, in: International Conference on Cloud Computing, CLOUD, 2015, pp. 317–324.
[41] F. Yan, L. Ren, D.J. Dubois, G. Casale, J. Wen, E. Smirni, How to supercharge the amazon T2: observations and suggestions, in: International Conference on Cloud Computing, CLOUD, 2017, pp. 278–285.
[42] A. Ali, R. Pinciroli, F. Yan, E. Smirni, CEDULE: a scheduling framework for burstable performance in cloud computing, in: International Conference on Autonomic Computing, ICAC, 2018, pp. 141–150.
[43] M. Grottke, R. Matias, K.S. Trivedi, The fundamentals of software aging, in: International Conference on Software Reliability Engineering Workshops, ISSRE, 2008, pp. 1–6.
[44] F. Machida, J. Xiang, K. Tadano, Y. Maeno, Aging-related bugs in cloud computing software, in: International Conference on Software Reliability Engineering Workshops, ISSRE, 2012, pp. 287–292.
[45] F. Machida, D.S. Kim, K.S. Trivedi, Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration, Perform. Eval. 70 (3) (2013) 212–230.
[46] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, R. Buyya, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Softw. - Pract. Exp. 41 (1) (2011) 23–50.
[47] F.S. Hillier, G.J. Lieberman, Introduction to Oper. Res., ninth ed., McGraw-Hill, 2010.
[48] G. Bolch, S. Greiner, H. de Meer, K.S. Trivedi, Queueing Networks and Markov Chains - Modeling and Performance Evaluation with Computer Science Applications, second ed., Wiley, 2006.



Fabio Antonelli Graduate in Mathematics from University of Rome "La Sapienza" Ph. D. in Mathematics at Purdue University – W. Lafayette, Indiana (USA) Current Position: associate professor in Probability at the University of L'Aquila Research interests Probability Theory, Stochastic Calculus, Financial Mathematics



Vittorio Cortellessa is a Full Professor at the Department of Information Engineering, Computer Science and Mathematics at University of L'Aquila (Italy). His main research interests are in the area of model-driven analysis of software performance and reliability. He has published more than 100 papers on international journals and conferences, and he has served and serves as member of Program Committees and Editorial Boards for, respectively, conferences and journals in his research areas. More information at http://people.disim.univaq.it/cortelle/

Marco Gribaudo is an Associate Professor at the Politecnico di Milano, Italy. He works in the performance evaluation group. His current research interests are multi-formalism modeling, queueing networks fluid models, mean field analysis and spatial models. The main applications to which the previous methodologies are applied comes from Big Data applications, Cloud Computing, Multi-Core Architectures and Wireless Sensor Networks.

Riccardo Pinciroli received the M.S. and Ph.D. degrees in computer engineering from Politecnico di Milano, in 2014 and 2018, respectively. He is currently a Postdoctoral Research Assistant in Computer Science at College of William and Mary, VA, USA. His research interests include stochastic modeling, performance evaluation, energy efficiency and uncertainty propagation. That is applied to cloud computing, data-centers and IoT environments.

Kishor S. Trivedi holds the Hudson chair in the Department of Electrical and Computer Engineering at Duke University. He is the author of Probability and Statistics with Reliability, Queueing and Computer Science Applications, published by John Wiley. He has published more than 500 articles and has supervised 47 Ph.D. dissertations. He received the IEEE Computer Society Technical Achievement Award for his research on Software Aging and Rejuvenation.

Catia Trubiani is Assistant Professor in Computer Science at the Gran Sasso Science Institute, Italy. She received a Ph.D. from the University of L'Aquila, Italy, with a dissertation on the automated of architectural feedback from software performance analysis results. Her research interests include the quantitative modeling and analysis of interacting heterogeneous distributed systems. More information at: https://cs.gssi.it/catia.trubiani