Arabian Journal for Science and Engineering (2018) 43:7789–7802
https://doi.org/10.1007/s13369-018-3196-0

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Modeling and Analysis of Performance and Energy Consumption in Cloud Data Centers

Said El Kafhali[1] · Khaled Salah[2]

Received: 30 September 2017 / Accepted: 20 March 2018 / Published online: 27 March 2018

Abstract
Recently, the deployment of cloud data centers (CDCs) and the adoption of cloud technologies have transformed the way we do computation, storage and networking. Typically in a CDC, virtual machines (VMs) are allocated to physical machines. Estimating correctly the number of needed VMs to meet a given workload and QoS parameters is important for cost and resource efficiency. In this paper, we develop a queuing model to aid in studying and analyzing performance in CDC. We model the CDC platforms with an open queuing system that can be used to estimate the expected quality of service (QoS) parameters such as the throughput, the drop rate, the CPU utilization and the response time. In addition, we present an energy consumption model to study and estimate the energy consumption in the CDC. We give numerical examples to show how the proposed model estimates the number of needed VMs to meet a given level of QoS parameters. The results obtained from our analysis as well as the simulation models show that the proposed model is able to correctly and effectively estimate the number of VM instances required to achieve QoS targets under different workload conditions.

Keywords Cloud data center · Virtualization · Quality of service · Queuing theory · Performance analysis · Energy consumption

1 Introduction

Cloud computing technology is a computing paradigm in which data can be accessed by clients worldwide via a web browser. This technology has four main components [1]: clients, resources allocation, VMs and PMs. The client submits a service request that contains detailed resource requests (for CPU, physical memory, storage, etc.). Then, these resources are provisioned as VM instances in real- time subject to service-level agreement (SLA) document. Finally, the VMs are allocated to a set of PMs to run the application or service to be hosted. This process also takes into account economic factors. This mainly concerns the operator whose objective is to maximize revenue and minimize operational cost. Computing resources (servers, storage, networks, platforms, and software) are offered to clients by cloud providers either dynamically or elastically, according to the client's request and form of payment [2]. Cloud computing has been often used with three computing services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [3]. In IaaS model, the client can exploit many virtualized computing resources in the form of VM instances in order to deploy applications which may include operating systems and other applications over the Internet [4]. In PaaS model, the client has the possibility to use the programming languages and tools supported by the cloud provider [5]. For the SaaS model, the consumer has the capability to use the applications and services of the provider which are running on a cloud infrastructure. However, many client devices have access to applications by means of a thin client interface like a web browser. Furthermore, the customer is not responsible for controlling or monitoring the infrastructure of the cloud data center (e.g., network, operating systems, storage and servers) on which the applications run [6].

In an IaaS CDC, such as Amazon EC2 [7], when a client demands a VM, its instance is first created in accordance with

✉ Said El Kafhali
said.elkafhali@uhp.ac.ma

Khaled Salah
khaled.salah@kustar.ac.ae

[1] Computer, Networks, Mobility and Modeling Laboratory, Faculty of Sciences and Technologies, Hassan 1st University, Settat, Morocco

[2] Electrical and Computer Engineering Department, Khalifa University of Science and Technology, Abu Dhabi, UAE

the client's resource request and the cloud provider determines a PM to host the VM instance. A single PM can be habitually shared by many VM instances according to the CPU, memory and disk capacities requested by each VM. When the VM instance is running, this client can remotely connect and use it. The VM instance is released when the client terminates the work [8]. With pay-as-you-go charging model, clients are capable of renting their VMs with performance separation on CPU, memory and disk resources [9]. The required QoS is a major factor of the SLA contract established between cloud providers and clients. Ensuring the desired QoS is a key business element for a cloud computing provider [10]. Thus, the client makes service requests for processing at the CDC hosted by the cloud provider according to the negotiated SLA at a given price. A typical desired QoS defines a collection of performance parameters: blocking probability, waiting time, response time, queue length and throughput.

There are several methods to model and evaluate the performance of cloud computing systems, such as measurement, simulation and analytical modeling. On the one hand, evaluating with measurement can be costly, requiring extensive experimentations with different workloads and system configurations, which may not be feasible or practical in cloud computing systems due to the great number of key parameters and the dynamic traffic generated by clients. On the other hand, using simulation to evaluate cloud computing systems is flexible, but it requires many and independent runs to get an average result, which might be time-consuming to gain dependable results. Furthermore, multiple and independent runs of the simulation are required to evaluate the impact of different input parameters of simulation model, which makes the running times more rigorous. Using analytical modeling to evaluate such system can be highly costly and efficient when compared to experimental approach. It can also facilitate quick performance results to be obtained under different conditions and architecture [11]. Also, the queuing modeling has been used as an important method to evaluate the effects of different cloud resources and to predict the proper corresponding costs and benefits [3]. To this end, we propose, in this paper, an analytical queuing modeling to evaluate the performance in terms of throughput, drop rate, CPU utilization and the response time in CDC. Besides, an energy consumption model is proposed to study and estimate the energy consumption in the CDC.

Energy efficiency is one of the top priorities in a CDC. That is why ways to reduce energy consumption without sacrificing performance are among the most important research topics [12]. For a virtualized CDC optimizing dynamically, the placement and the migration of VMs are one of the best effective methods for power savings [13]. The energy consumption of CDC can be divided into physical resources and computing resources [14]. Virtualized CDC providers seek to maximize QoS and minimize infrastructure costs. However, an important challenge to save power is, therefore, to explore the trade-offs that can result in an optimal balance between QoS performances and energy consumption and the relations among CDC components. In the literature, the performance of CDC has been the subject of several research papers. The majority of these works have been mostly interested in evaluating the performance of data center in terms of energy consumption [15–17] or QoS parameters [18–21]. However, the objective of this article is to provide the performances in terms of energy consumption and QoS parameters.

Our contributions in this article are summarized as follows:

- A stochastic queuing model to aid in studying and analyzing performance in CDC is proposed and discussed. At any given workload, the model can estimate the number of VMs that are required to satisfy given SLA or QoS parameters.
- An analytical model is presented, and mathematical equations are derived for key performance measures for the proposed queuing model.
- An energy consumption model is presented based on continuous- time Markov chain (CTMC) to study and estimate the energy consumption in the CDC.
- Numerical examples are given to illustrate how this model can be utilized to satisfy key QoS parameters and to estimate the needed number of VMs needed under diversity of conditions for the incoming workload.
- The proposed analytical models are cross-validated with a simulation model based on Java Modeling Tools (JMT) simulator.

This article is a major extension of our preliminary short version [22] which primary focused on studying the performance of services hosted in cloud data centers. In this extended version, we provided greater details for our performance model. Also we give more results and analysis for the performance of two use cases: cloud-based database services and web services. More importantly, we have included a new and a detailed mathematical model for energy consumption and reviewed existing works for modeling energy consumption in CDC. Furthermore, we provided numerical and simulation results with new figures and substantial discussion on energy consumption in CDC.

The rest of this article is organized as follows: Sect. 2 summarizes the related existing works. The proposed CDC model is presented in Sect. 3. Section 4 presents mathematical model for the proposed CDC model. In Sect. 5, we briefly introduce the energy consumption issues and a detailed mathematical model for energy consumption in CDC. Section 6 presents numerical analysis and discusses the limitations of the proposed approach as well as suggesting further research. Finally, Sect. 7 is devoted to the conclusion and future works.

2 Related Work

Cloud computing has earned a lot of attraction, and there are many papers regarding the performance evaluation of CDC environment. In [18], a mathematical performance model of VM live migration is evaluated. This model shows that an effective VM live migration can reduce service rejection probability and the mean total delay. The authors in [19] obtained a distribution of response time using a classic open queuing network, assuming that inter-arrival and service times are exponentially distributed and fixed. Using the obtained distribution, they demonstrated that the analytical model could be used to identify the relationship between the number of required computing servers, the maximum number of requests and the highest level of service. The work in [20] studied the performance of M/M/m queuing model to describe a synthetic method to optimize the service performances in cloud computing. The simulation result shows that the proposed queuing model can allow less waiting time, less queue length and less number of customers using the synthetic optimization function when the number of servers increases. The work [23] proposed an interacting model to get better performance and quantification of a large-scale virtualized IaaS CDC. The analytical results show the impact of the system characteristics and customers workload on two performance measures: mean response time and job-rejection probability. About the service driver [24], they developed an optimization and cost analysis framework by using stochastic availability performance models of an IaaS cloud. Their model takes service requirement variables like downtime and job-rejection rates and then calculates the type and number of nodes you would need to meet those requirements with the least amount of operating cost.

The authors in [25] developed analytical models for IaaS cloud where user's requests are processed by VMs. The effectiveness of VM replication in terms of request completion time performance and energy consumption is also analyzed. They introduced a power consumption modeling approach which discusses the measurement of power consumption at different modes of operation (cold, warm and hot) and the associated energy consumption. The proposed model has been validated with CloudSim simulator. The numerical examples show the effectiveness of VM replication schemes on energy consumption and on job completion time. The work in [21] constructed a complex queuing model to measure the performance of heterogeneous CDC. They analyzed different QoS parameters. The obtained results demonstrate the accuracy of the proposed model and its convenience to analyze heterogeneous data centers. The work in [26] developed a new technique based on queuing models for efficient live migration of multiple VMs. They modeled the request arrival using the M/M/C/C queue in order to evaluate the blocking rate of client requests. Similarly, they modeled the arrival request using the M/M/C queue to evaluate the average waiting queue length, the average queue length, the average waiting time and the average sojourn time. The proposed model is evaluated by conducting mathematical analysis. The numerical results show that the proposed approach outperforms the existing literature approach.

The authors in [27] proposed simple performance approaches for evaluating the response times of applications executed in a SaaS cloud. They deployed the proposed models on a real virtualized platform. The experiment results show that the models allow predicting response times of SaaS applications accurately. The work in [28] presented a comprehensive modeling of a disaster tolerant data center (DTDC) using stochastic reward nets (SRN). The obtained results show the availability progress of DTDC and featured system responses corresponding to the selected variables (downtime analysis cost and sensitivity analysis). In [29], the authors proposed a queuing model that can be used to warrant proper elasticity for cloud clients. The proposed model predicts the needed number of VMs to satisfy a predefined service SLA performance requirement. A test-bed setup on the AWS platform is used to conduct experimental measurements that validated the analytical model.

There are several works that consider power management in virtualized CDC by means of hardware- and software-based solutions [15–17,30]. In [15], the authors considered Containers- as-a-Service approach and presented a method for finding efficient VM sizes for hosting containers. They analyzed clouds trace logs from Google and taking into consideration the cloud workload variances which is crucial for testing and validating the proposed models. The experimental results showed that the proposed model up to 7.55% of improvement in the mean energy consumption compared to the existing scenarios where the VM sizes are fixed. Furthermore, the number of instantiated VMs for hosting the containers is also improved by 68% on average. The work in [16] developed a dynamic VM placement algorithm based on the evolutionary game theory. A model of energy consumption for computing the amount of energy consumed during the process of dynamic adjustment of the VMs placement is built. By adjusting VMs placement dynamically, the obtained experimental results demonstrate that the energy consumption can be reduced. In [17], the authors proposed a power management problem for parallel services computations in CDC. They adopt an optimal energy/time allocation among levels of tasks and equal power supply to tasks at the same level and show significant performance improvement. The authors in [30] introduced a unique replication solution which considers both energy efficiency and bandwidth consumption in geographically distributed CDC. The proposed model improves network bandwidth and communication delay between geographically dispersed CDC as well as inside each CDC.

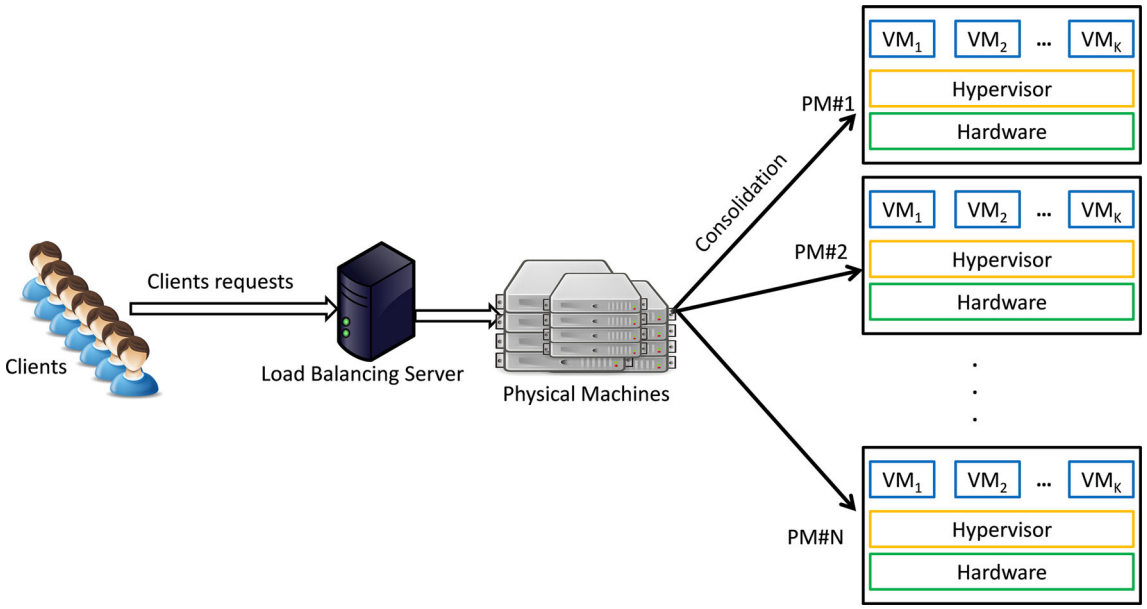

Fig. 1 A typical architecture of the cloud data center system

In contrast to prior work reported in the literature, in our proposed model, we optimize cloud resources allocation and workload scheduling to minimize the overall resources cost by allocating the smallest number of VM nodes needed to meet the required performance parameters. The model can also be used in scalability solutions and dynamic scaling in which resources are allocated as the incoming workload changes. So if the incoming workload goes up, more VM nodes have to be allocated to meet such increase. Conversely, if the workload goes down, the number of provisioned VM nodes has to decrease. In addition, none of the prior works have focused on determining the number of VM nodes required to meet SLA performance parameters for cloud services with an estimation of the overall energy consumption. Also, some prior works reported in the literature ignore the role of the load balancer which is a key in dispatching, monitoring and tracking the availability of compute instances at the CDC.

3 Modeling Cloud Data Centers

We consider a large CDC hosting multiple physical machines (or PMs), whereby the cloud compute unit or virtual machines (i.e., VMs) run on the top of PM according to the VMs requests. Typically, a CDC as those seen in AWS Cloud, Microsoft Azure, Google, Facebook, Yahoo, etc., contains tens of thousands of PMs [31]. Each VM is allocated to one PM, whereas a PM can be allocated multiple VMs through a hypervisor. We propose a multi-server queuing model [32] to model the CDC depicted in Fig. 1.

The load balancing (LB) server retains the scheduled queue to receive all incoming requests from clients. A request from a client is transmitted to the LB server [33], associated with a SLA document. Client requests are received by LB queue and then processed on the first-in first-out (FIFO) basis. The inter-arrival durations between successive arriving requests are typically independent and exponentially distributed with rate $1/\lambda$. Queued requests are distributed to different PMs. We assume that the service time of the LB server is exponentially distributed with mean service time $1/\mu$ and the LB server is modeled as an M/M/1/C queuing model [34]. The LB works as follows: when a new client request arrives, if the number of client requests in the queue is more than the threshold C, then the new client request will be blocked; otherwise, it will be accepted.

We suppose there are N PMs in the CDC. The requests are uniformly distributed by the LB server to each PM with the same probability $1/N$. Accordingly, the arrivals of client requests at each PM follow Poisson process with arrival rate λ/N. We assume the PMs in a CDC are homogeneous service and the server time of each PM has an exponential distribution with a mean service time of $1/\mu_1$. We model each PM in the CDC as an M/M/K/m queue [34]. Each PM may support up to m VMs. We assume that the service times and the inter-arrival time of client requests are exponentially distributed. If the queue obtains its maximum limit, the extra requests are dropped. If the resource is available, then request is accepted and routed by the LB to the corresponding VM. It is assumed that all VMs allow the same web services with the same functionality to clients via Internet. Each VM necessitates a set of computing resources including network bandwidth, CPU, memory and storage space. As the maximum number

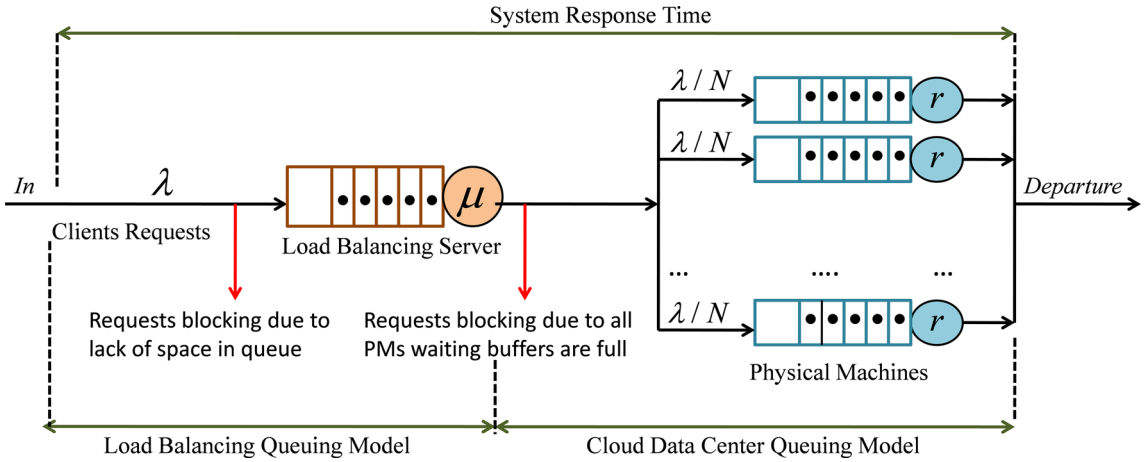

Fig. 2 Queuing model for workload distribution

of requests in the system is C, we assume that C is equal to the number of PM times and to the number of VMs that can be allocated to a single PM. So, C is calculated by using the following formula

$$C = N \times K \tag{1}$$

The proposed queuing model of the CDC is shown in Fig. 2. An arriving request will be dropped when it finds the LB queue full. Otherwise, it must wait in the LB queue until the LB processes it on a FIFO basis. We assume we operate in a homogenous cloud environment whereby all PMs are equal in processing capacity and sizes. Each PM queue has a buffer which can contain at most $K-1$ requests. The incoming request is put into a PM waiting queue with free buffer space if it exists. Where all PMs waiting buffers are overflow, the arrival client request is dropped. Consequently, a client request may be queued in a PM waiting buffer, dropped because all waiting buffers are full, or because of the insufficient LB buffer space.

It is worth noting that, in our model, we focus on the main queue related to load balancing (scheduling queue) and the queues (local's queues) allocated to PM nodes in the cloud data center. The load balancer receives requests from end users and then distributes them evenly among the VM nodes in CDC. End users in the model are represented by the generated requests, whereas scheduling queue and local queue are the processing stations for these requests. In addition, we focus on deriving the QoS-related formulas and equations that capture the behavior and performance of the entire system that contains two layers (scheduling and cloud data center).

For our analysis, we assumed that the arrival rate of client requests follows a Poisson arrival and the service times are exponentially distributed. These assumptions may not necessarily always hold. In fact, the incoming request behavior in such system must rely on more complex traffic models than a simple Poisson process (such as semi-Markov, self-similar behavior, Markov modulated Poisson processes, business behavior and long-range dependency). Also, the distribution of services times may not always be exponential. It is worth noting that an analytical solution becomes intractable when considering non-Poisson arrivals, and when also considering general service times. On the other hand, assuming Poisson arrivals and exponential service time has been used in the literature and can provide adequate approximation of real systems [8,20,21,23,29,32,33,35].

4 Analytical Queuing Model

4.1 Load Balancing Queuing Model

Request LB is an increasingly available feature of cloud computing offerings. A LB dispatches received requests from clients to servers in CDC according to a load dispatching policy. Dispatching policies differ for the decision approach and for the amount of information they use. In this paper, the LB is modeled as a M/M/1/C queue [35]. The queuing system has a maximum capacity of C, i.e., the maximum queue length is $C-1$. The arrival request waits in queue if it finds less than C requests in the system; otherwise, the request is dropped as the request cannot be accommodated in the queue. From the balanced equations and by applying the normalization condition, we can express the steady-state probability of k requests in the LB system as follows

$$\pi_k = \frac{1-\rho_1}{1-\rho_1^{C+1}}\rho_1^k \tag{2}$$

where $\rho_1 = \lambda/\mu$ represents the incoming workload at LB server.

We can derive important performances parameters in the LB system. First, the mean throughput is

$$X = \lambda \frac{1-\rho_1^C}{1-\rho_1^{C+1}} \tag{3}$$

The average number of requests in the LB system is

$$E(k) = \sum_{k=1}^{C} k\pi_k = \frac{\rho_1}{1-\rho_1} \frac{1-(C+1)\rho_1^C + C\rho_1^{C+1}}{1-\rho_1^{C+1}} \tag{4}$$

The loss probability due to lack of space in LB queue is given by

$$P_{\text{loss}} = \pi_C = \frac{1-\rho_1}{1-\rho_1^{C+1}} \rho_1^C \tag{5}$$

Finally, by using the Little's law formula [36] we obtain the average response time at the LB system as

$$E(rb) = \frac{E(k)}{X} \tag{6}$$

4.2 Cloud Data Center Queuing Model

We model each PM as an M/M/m/K queue. Each PM can have up to m VMs, and K is the maximum number of the client requests in the PM. VM refers to the software implementation of computer that runs its own operating system and applications as if it is a PM. Figure 3 shows the transition diagram for M/M/m/K queuing modeling for a single PM.

Let $\pi_i(n)$ denote the steady-state probability of having n requests in $PM_i (i = 1, 2, \ldots, N)$. Using the balanced equations and we note $\alpha = \lambda/N$, we find that

$$\pi_i(n) = \begin{cases} \frac{\pi_0(\alpha)^n}{n!\mu_1^n}, & \forall n < m \\ \frac{\pi_0(\alpha)^n}{m!m^{n-m}\mu_1^n}, & \forall n \geq m \end{cases} \tag{7}$$

where π_0 is given by

$$\pi_0 = \left[1 + \frac{(m\rho)^m(1-\rho^{k+1-m})}{m!(1-\rho)} + \sum_{i=1}^{m-1} \frac{(m\rho)^i}{i!} \right]^{-1} \tag{8}$$

where ρ denotes the incoming workload and is expressed as

$$\rho = \frac{\alpha}{m\mu_1} \tag{9}$$

We deduce the performances parameters as follows. Firstly, the rate of drop or loss can be expressed as

$$v = \alpha\pi_i(K) \tag{10}$$

The effective rate of arrivals, λ_{eff}, to the service can be given as follows

$$\lambda_{\text{eff}} = \alpha(1-\pi_i(K)) \tag{11}$$

The central processing unit utilization of each VM is given by

$$U_{VM} = \frac{\lambda_{\text{eff}}}{m\mu_1} = \rho(1-\pi_i(K)) \tag{12}$$

The average number of requests in the PM_i can be expressed as

$$E(n) = \sum_{j=1}^{K} j\pi_i(j) \tag{13}$$

The average number of requests waiting in the PM_i can be obtained from

$$E(n_w) = \sum_{j=m+1}^{K} (j-m)\pi_i(j) \tag{14}$$

Finally, we can express the mean waiting time and the mean response time in the PM_i as follows

$$E(w) = \frac{E(n_w)}{\alpha(1-\pi_i(K))}, \quad E(r_c) = \frac{E(n)}{\alpha(1-\pi_i(K))} \tag{15}$$

The response time for a request in CDC is defined as the time (in seconds or milliseconds) from the moment when the client request arrives at the CDC to the time when execution is complete. Based on Fig. 2, we compute the response time T for a request on a system, the probability that a request is served below a specific time t is given by

$$T = E(rb) + E(rc) \tag{16}$$

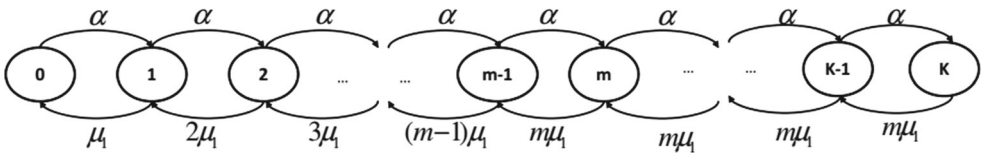

Fig. 3 Continuous time Markov chain for new request in a single PM

5 Modeling Energy Consumption

5.1 Energy Consumption in Cloud Data Center

The continued and rapid growth in the demand for IT resources by different cloud services expressed by telecommunication operators, banks, etc., has led to a fast multiplication of large CDC with millions of PMs [37]. Thus, CDC being large-scale computing resources needs a huge energy budget which leads to various energy efficiency issues. Several methods can be used to achieve energy efficiency, such as energy efficient hardware, improvement of applications algorithms, dynamic voltage and frequency scaling (DVFS) [38] and virtualization of computer resources [39]. Using virtualization method to improve the energy efficiency in CDC is one of the ways that many researches are focusing on [37]. Virtualization technology refers to the act of creating multiple VMs on a single PM. By this technology, data centers can improve the resource utilization and reduce energy consumption by switching PMs off or turning to the idle mode. An important quantity of power is consumed even when the PM is idle, thus opening a possibility for VMs migration in CDC for reducing energy cost [40]. Therefore, many studies [41–45] demonstrate that in average an idle PM consumes about 70% of the power consumed by the PM running at full CPU utilization. This fact justifies the technique of switching idle PMs off to reduce total energy consumption.

The energy consumed by a CDC can be classified into two categories [46]: (1) the energy consumed by infrastructure facilities (e.g., cooling systems and power conditioning systems) and (2) the energy consumed by IT equipments (e.g., PMs, storage, network bandwidth). Nonetheless, modeling the energy consumption behavior of a CDC, either at the individual component level or at the whole system level, is not easy. Indeed, the energy consumption in a CDC depends on different factors such as workload, hardware specifications, applications types and cooling requirements, which cannot be estimated facilely. Since running PMs consume electricity, the cloud provider decides dynamically how many PMs to run by intermediary of a resources allocation policy. The principal objective is to get the optimal number of PMs, which should be switched on in order to optimize the cloud provider's profit [47]. The number of running PMs should change in response to the change in the workload conditions; the issue is how to estimate the best number of PMs that switched on in order to achieve QoS targets. In order to achieve this objective, we proposed in this section a CTMC model to estimate the energy consumption in CDC.

5.2 Energy Consumption Model

Besides the advantages offered by the virtualization in cloud computing, the use of this technology can lead to a high level of power consumption due to resource utilization and to the expansion of the number of VMs needed to satisfy the clients' requests. This leads to look at the behavior of CDC in terms of energy consumption related to the resources virtualization. Previous works in the literature that studied energy consumption in the data center consider that each PM contains a single VM [37,41,47–52]. To this end, we developed a model of energy consumption where each PM supports more than one VM. Due to the dynamic change of resource requests of VMs, the changes in energy consumption form a stochastic process [53]. For evaluating the energy consumption, we define additional states for PM (*Off* and *Idle* states). With this, our energy consumption model is modeled with a two-dimensional CTMC shown in Fig. 4. This CTMC captures the details of queuing at the PM input queue in order to estimate the energy consumption.

The state index of the CTMC is denoted by (j, x), where j indicates the number of requests in the queue and x denotes the state of the PM where the VM request is undergoing provisioning. The states O, F and I mean that the PM is *On*, the PM is *Off*, and the PM is Idle, respectively. The *Idle* PMs without serving requests will be turned *Off* to save power. States of the form $(n, F)(n = 1, 2, \ldots, K)$ indicate the PM is empty which means there are no service requests and there are n requests in the PM. States of the form $(n, O)(n = 1, 2, \ldots, K)$ signify states in which the VM serves a request and there are in total n requests in the PM. The state $(0, I)$ signifies that the PM is idle and that there are no requests in the queue and the client arrival request will be accepted. At the arrival of first request, the system moves to state $(1, O)$ which means that the request will be treated immediately in

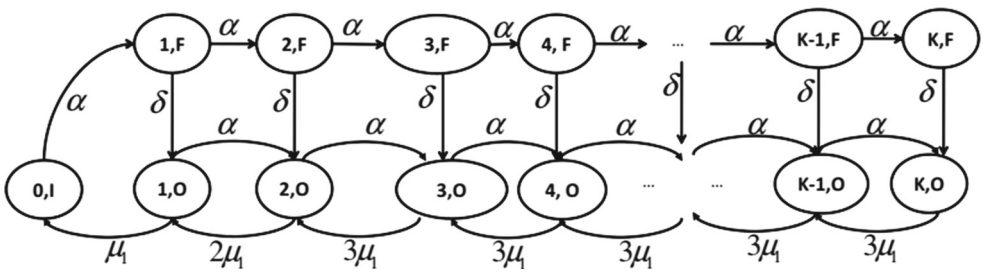

Fig. 4 Energy consumption Markov chain for a single PM

the PM. Another request has arrived, and the system transits to state $(2, O)$ with rate α. A PM accepts the request and processes it. So the system moves back to state $(1, O)$ with rate μ_1. Once a PM in the states F and I, we do not have any requests to process. Afterward, the PM moves from *Off* state to *On* state with rateδ. The PM moves from $(0, I)$ to $(1, F)$ with rate α. Finally, the PM moves from $(1, O)$ to $(0, I)$ state with rate μ_1. After obtaining the On PM, the request runs in the cloud and releases the PM when it finishes. We assume that homogenous requests (one PM for each request, the same mean service time for all requests). The waiting queue size is K.

In Fig. 4, we remark that starting from any state $(i, O)(i = 1, 2, \ldots, K)$ it is not possible to do transitions to states $(j, F)(i = 1, 2, \ldots, K)$ without visiting firstly state $(0, I)$. From the proposed CTMC, it is possible to compute the stationary probability and performance parameters under various request strategies. Let $\pi_{j,x}$ be the equilibrium steady-state probabilities at state (j, x). From CTMC depicted in Fig. 4, the balance equations can be written as follows.

$$\pi_{0,I}\alpha = \pi_{1,O}\mu_1 \tag{17}$$

$$\pi_{n,F}(\alpha+\delta) = \pi_{n-1,F}\alpha,\ n = 2, \ldots, K-1 \tag{18}$$

$$\pi_{K,F}\delta = \pi_{K-1,F}\alpha \tag{19}$$

$$\pi_{1,O}(\alpha+\mu_1) = \pi_{1,F}\delta + \pi_{2,O}2\mu_1 \tag{20}$$

$$\pi_{2,O}(\alpha+2\mu_1) = \pi_{1,O}\alpha + \pi_{2,F}\delta + \pi_{3,O}3\mu_1 \tag{21}$$

$$\pi_{3,O}(\alpha+3\mu_1) = \pi_{2,O}\alpha + \pi_{3,F}\delta + \pi_{4,O}3\mu_1 \tag{22}$$

$$\pi_{i,O}(\alpha+3\mu_1) = \pi_{i-1,O}\alpha + \pi_{i,F}\delta + \pi_{i+1,O}3\mu_1,\quad i = 3, \ldots, K-1 \tag{23}$$

$$\pi_{K,O}3\mu_1 = \pi_{K-1,O}\alpha + \pi_{K,F}\delta \tag{24}$$

By iterating Eqs. (18) and (19), taking into account Eq. (17), we obtain

$$\pi_{n,F} = \frac{\mu_1}{\alpha}\sigma^n\pi_{1,O},\ n = 1, \ldots, K-1 \quad \text{with}\quad \sigma = \frac{\alpha}{\alpha+\delta} \tag{25}$$

$$\pi_{K,F} = \frac{\mu_1}{\alpha}\sigma^{K-1}\pi_{1,O} \tag{26}$$

From Eqs. (21) and (25), we have

$$\pi_{2,O} = \frac{\alpha(\mu_1+\alpha+\delta)}{2\mu_1(\alpha+\delta)}\pi_{1,O} \tag{27}$$

From Eq. (23) it follows that $(\pi_{n,O} : n = 3, \ldots, K)$ is a solution of the non-homogeneous linear difference equation with constant coefficients

$$3\mu_1 x_{n+1} - (\alpha+3\mu_1)x_n + \alpha x_{n-1} = \frac{-\delta\mu_1}{\alpha}\sigma^n\pi_{1,O},\quad n = 2, \ldots, m \tag{28}$$

where the last equation is due to Eq. (25). Using the same approach used in [54] for solving such equations, we consider the corresponding characteristic equation

$$3\mu_1 x^2 - (\alpha+3\mu_1)x + \alpha = 0 \tag{29}$$

which has two roots $\alpha/3\mu_1$ and 1. The general solution of the homogeneous version of Eq. (28) is given from [49] as $x_n^{\text{hom}} = A1^n + B\,(\alpha/3\mu_1)^n$. The general solution x_n^{gen} of Eq. (28) is given as $x_n^{gen} = x_n^{\text{hom}} + x_n^{\text{spec}}$, where x_n^{spec} is a specific solution of Eq. (28). Because the non-homogeneous part $-\frac{\delta r}{\alpha}\sigma^n\pi_{1,O}$ of Eq. (28) is geometric with parameter σ, we can also use the approach used [54] in to get a specific solution. Therefore, we propose specific solutions of the form $C\sigma^n$ [49]. Substituting $x_n = C\sigma^n$in Eq. (28), we have

$$C = \frac{3\mu_1(\alpha+\delta)}{\alpha(3\mu_1-\alpha-\delta)}\pi_{2,O} \tag{30}$$

Thus, the general solution of Eq. (28) is calculated by

$$x_n^{\text{gen}} = A1^n + B\left(\frac{\alpha}{3\mu_1}\right)^n + C\sigma^n, n = 2, 3, \ldots, K-1 \tag{31}$$

where the constant C is given by Eq. (30) and the coefficients A, B are to be calculated. Using the same method in [49], we obtain $A = 0$ and $B = -\frac{3\mu_1(\alpha+\delta)}{\alpha(3\mu_1-\alpha-\delta)}\pi_{2,O}$, thus, from Eq. (31)

$$\pi_{n,O} = \frac{3\mu_1(\alpha+\delta)}{\alpha(3\mu_1-\alpha+\delta)}\left(\sigma^n - \left(\frac{\alpha}{3\mu_1}\right)^n\right)\pi_{2,O},\quad n = 3, 4, \ldots, K-1 \tag{32}$$

$$\pi_{K,O} = \frac{\alpha}{3\mu_1}\pi_{K-1,O} + \frac{\delta}{3\mu_1}\pi_{K,F} \tag{33}$$

$$\Rightarrow \pi_{K,O} = \left(\frac{(\alpha+\delta)}{(3\mu_1-\alpha+\delta)}\left(\sigma^K - \left(\frac{\alpha}{3\mu_1}\right)^K\right) + \frac{\delta}{3\alpha}\right) \tag{34}$$

from which together with the normalization equation

$$\pi_{0,I} + \sum_{i=1}^{K}(\pi_{i,O}+\pi_{i,F}) = 1 \tag{35}$$

We obtain steady-state probabilities $\pi_{j,x}$, and the performance measures such as probability of blocking due to insufficient resources in the PM

$$P_{bloc} = \pi_{K,O} + \pi_{K,F} \tag{36}$$

The energy consumption by a single PM is the total quantity of work performed by a PM over a duration period t (measured in seconds) can be expressed as [37]

$$E_{PM} = P_{PM} * t \tag{37}$$

where P_{PM} is the power of PM and is the rate at which the work is performed by the PM.
We assume that the time spent in *On* and *Idle* states is t_{On} and t_{Idle}, respectively. From CTMC depicted in Fig. 4, we have

$$t_{On} = \sum_{i=1}^{K} \frac{1}{1 - \pi_{i,O}} \quad \text{and} \quad t_{Idle} = \frac{1}{1 - \pi_{0,I}} \tag{38}$$

When a PM is processing requests, it is in the *On* state and we denote its energy consumption by E_{On}. If there are no requests to process, the PM can remain *Idle* and be turned *Off*. Moreover, the energy consumption E_{Idle} of a PM remained *Idle* is not null, and $E_{Idle} < E_{On}$. If the PM is turned *Off*, it consumes zero energy. Then $0 = E_{\text{Off}} < E_{\text{Idle}} < E_{\text{On}}$. Therefore, E_{On} and E_{Idle} are expressed as follows

$$E_{\text{On}} = P_{\text{On}} * t_{\text{On}} \text{ and } E_{\text{Idle}} = P_{\text{Idle}} * t_{\text{Idle}} \tag{39}$$

where P_{On} and P_{Idle} are the power of PM in P_{On} and $idle$ states, respectively. The mean energy consumption in PM is as follows

$$E_{\text{consumption}} = E_{\text{On}} + E_{\text{Idle}} \tag{40}$$

6 Simulation and Numerical Results

There are several publicly and commercially available network simulators. A number of these simulation tools are designed particularly for cloud networks (e.g., iCanCloud, CloudSim, MDCSim and EMUSIM) [55,56], and some other simulators are generic in natures (e.g., J-Sim, OPNET, OMNeT, NS) [57]. All of these simulators did not have the capabilities to capture precisely the internal behavior and dynamics of the CDC. For that, we choose the JMT tool [58,59] to implement the performance of the proposed model. The simulation is carried out on a high-end workstation with Intel Core i5 CPU @ 2.40 GHz, 4 GB of memory and a 250 GB permanent storage.

Simulation Environment We consider a scenario of a small-scale CDC where the CDC has 10 PMs, with a mean arrival rate of 1000 client requests by second and a mean client request service time by the LB server of 0.0001 s. Two types of common requests are considered, namely (1) web requests and (2) database requests. The average service time at the PMs of a web requests is 10 milliseconds, while of a database requests require on average 15 milliseconds. Note that we vary some of these parameters depending on the simulation scenarios, whereas the others remain fixed. The parameters in Table 1 are used in our simulation.

Table 1 Input parameters and their values

Parameters	Description	Values
λ	Request arrival rate	[1000 to 3000] (Req/s)
$1/\mu$	Mean request LB service time	0.0001 (s)
$1/r$	Mean web request VM service time	0.01 (s)
$1/r$	Mean database request VM service time	0.015 (s)
C	Maximum number of client requests in the system	300
K	Maximum number of the client requests in the PM	30
N	Number of PMs in the data center	10

Performance Analysis The simulation results presented in this section and the respective figures are the mean of five simulation runs. Simulation results are represented by the black circles, whereas the analysis results are represented by lines. The numerical results are shown in Figs. 5 and 6. We plotted the web and database performance curves using multiple VM instances as function of requests arrival rate. In each figure, the performance values are computed from analysis and simulations models by varying arrival rates. The performance measures we considered are those of the response time, drop rate throughput and CPU utilization are plotted in the figures. We validate the proposed model by comparing the numerical results with the results obtained from simulation. Figure 5 shows performance results obtained in a simulation and analysis for web requests, whereas Fig. 6 exhibits those results for database requests. All of these figures depict clearly the impact of the number of VMs on key performance metrics. Specifically, in Fig. 5a, b, it is clear that when the requests arrival rate reaches the 2200 requests per second, we can discuss the impact of the number of the VM instances on the throughput and the response time measures. It is obvious that as the number of VMs increases, the throughput increases. However in the data performance case, when the arrival rate reaches the 1400 requests per second, we see that, as the requests arrival rate and the number of VMs increases, the throughput increases and the response time decreases. The drop rate variation versus requests arrival rate depicted in Fig. 5c demonstrates that from 1800 requests per second, as the number of VM instances increases the drop rate decreases and reaches 700 requests per second. Considering the CPU utilization parameter depicted in Fig. 5d, we observe that for

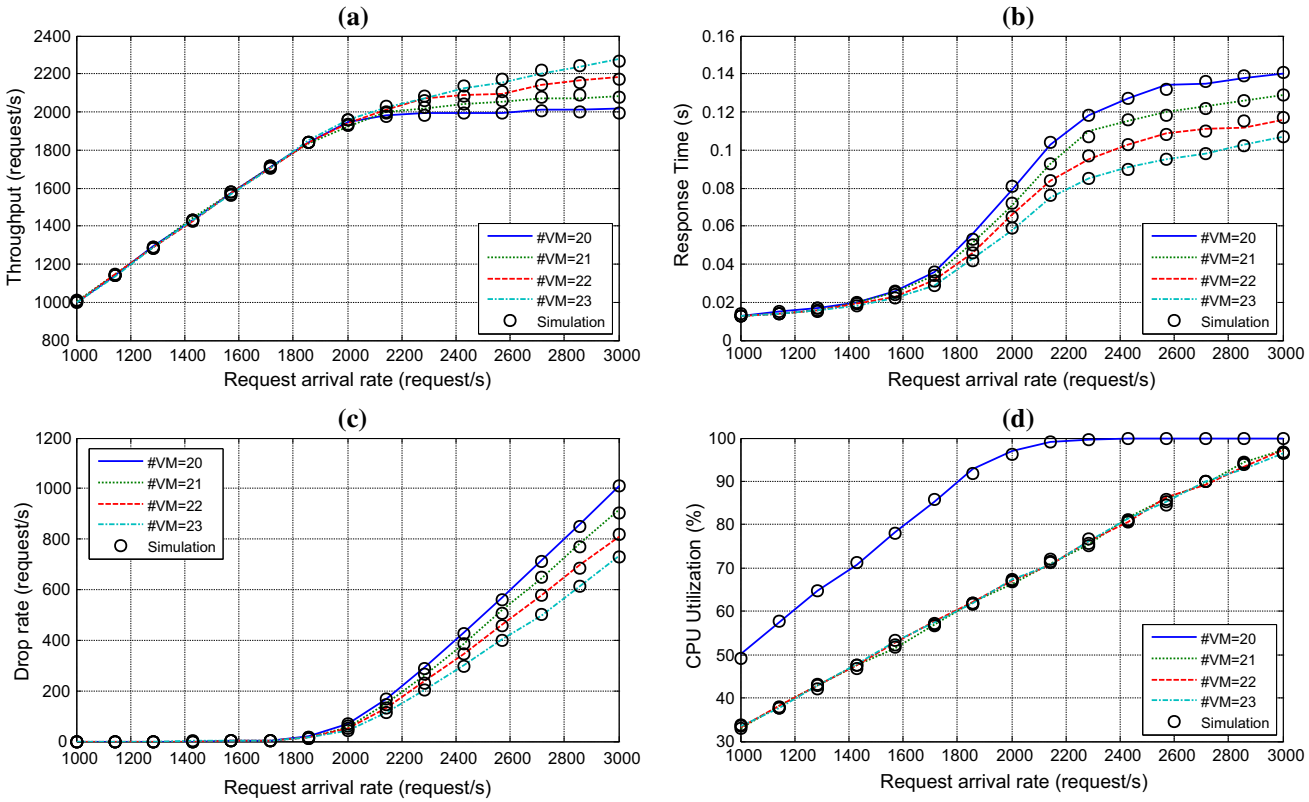

Fig. 5 Web performance curves using multiple VM instances as a function of arrival rate

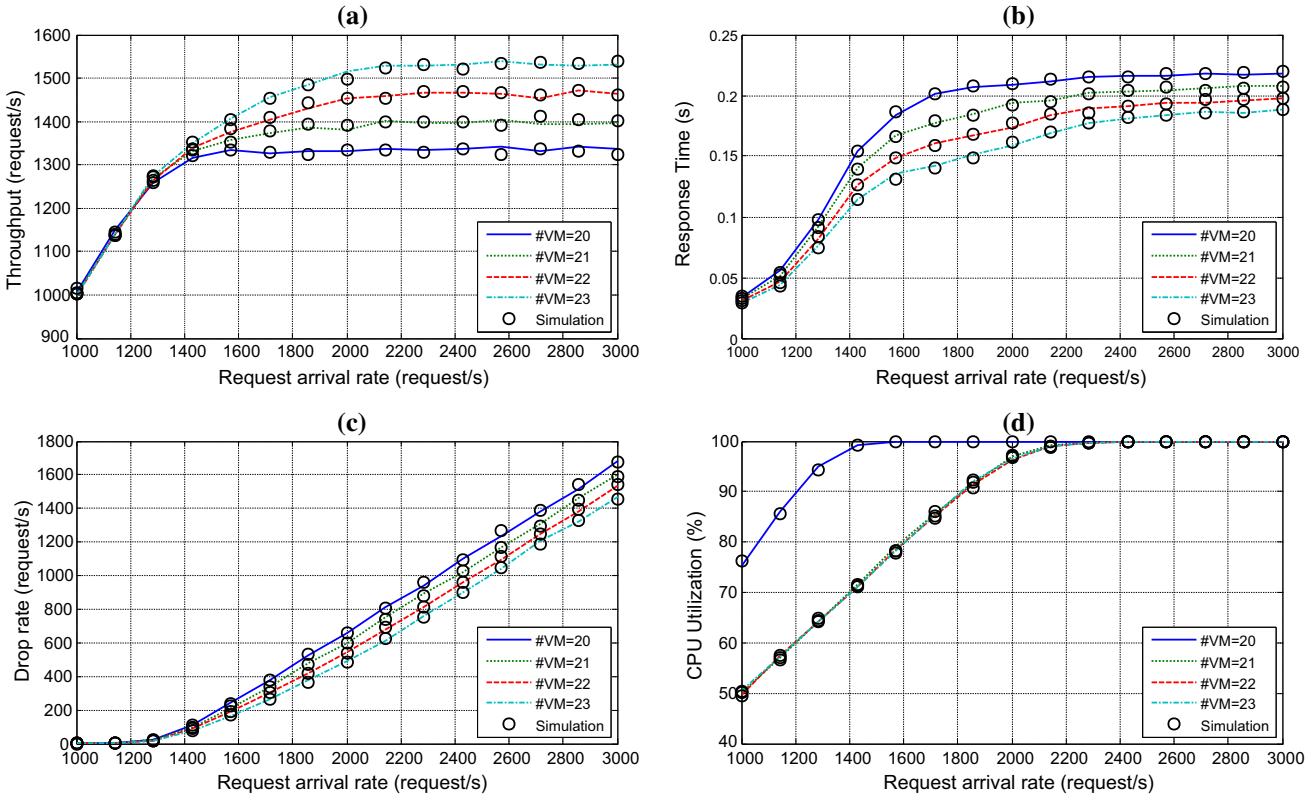

Fig. 6 Database performance curves using multiple VM instances as a function of arrival rate

Table 2 Comparison of simulation results with analysis for web service

	Response time(s)				Drop rate(Req/s)				CPU utilization(%)			
	Analysis	Simulation			Analysis	Simulation			Analysis	Simulation		
	Avg	Avg	Min	Max	Avg	Avg	Min	Max	Avg	Avg	Min	Max
10 VMs at a rate of 1000 Req/s	0.150	0.155	0.148	0.161	33.517	32.463	29.528	36.047	97.7	96.6	94.8	98.3
20 VMs at a rate of 2000 Req/s	0.078	0.080	0.078	0.082	64.997	66.350	60.489	73.469	96.3	96.6	95.1	98.1
30 VMs at a rate of 3000 Req/s	0.054	0.053	0.052	0.055	102.938	98.890	87.949	112.941	96.7	97.0	95.5	98.5

Table 3 Comparison of simulation results with analysis for database service

	Response time(s)				Drop rate(Req/s)				CPU utilization(%)			
	Analysis	Simulation			Analysis	Simulation			Analysis	Simulation		
	Avg	Avg	Min	Max	Avg	Avg	Min	Max	Avg	Avg	Min	Max
10 VMs at a rate of 1000 Req/s	0.418	0.423	0.415	0.431	332.527	336.149	326.998	345.827	99.3	98.9	97.15	100
20 VMs at a rate of 2000 Req/s	0.212	0.209	0.204	0.213	670.517	665.074	650.657	680.145	98.3	98.6	97.1	100
30 VMs at a rate of 3000 Req/s	0.140	0.141	0.138	0.144	1008.219	1010.704	982.794	1040.245	99.1	98.8	97.1	100

the first three values of arrival rate, the curve is approximately linear but when using 23 VM instances, we observe that this metric reaches the 100% when we have 2200 requests per second and so on.

Figure 6a, c exhibits that when we increase the number of VM instances and the requests arrival rate, the response time and drop rate metrics decrease. Figure 6d shows the CPU utilization for multiple numbers of VMs. The curves of CPU utilization reach 100% when the rate of arrival request approaches the 2200 requests by second. As a result, when the number of VMs decreases, the CPU utilization metric becomes higher.

Our analytical model is validated by Tables 2 and 3 which compare analytical results with simulations results for some performance parameters: CPU utilization, response time and drop rate. Specifically, we considered 10 VMs, 20 VMs and 30 VMs, with 1000, 2000 and 3000 requests/s, respectively. From the tables, it was observed that the results from the analytical model are conforming to the simulation measurements which validate our analytical model. The examination of the minimum and maximum values for the recorded measurements shows considerable fluctuation and variability. This can be explained by the virtualization technology implemented at the CDC. Besides, other actions carried out by other cloud applications and services, which might be running at the same time in the CDC, can explain this significant fluctuation. The workload and traffic that can be introduced by these cloud-hosted applications and services can significantly impact the total network delay and performance.

It is worth noting that the simulation cannot produce instant results and takes time to be performed. In fact, the simulation requires many independent runs which may take long time to produce accurate results. The input values for the various parameters as well as the actual running of simulation program or software usually have to be done manually and cannot be integrated with elasticity and scalability algorithms, whereas analytical methods can obtain instantaneous results and with mathematical formulas being easily integrated with other algorithms.

Clearly, it can be concluded that from the obtained results which are reported in the figures for response time criteria, the simulation and analytical results show that with a fixed number of VM instances, if the workload increases, the response time increases, and inversely, if the workload decreases, the response time decreases. Consequently, it is important to scale up and down VM instances as function of the incoming workload. This will help the cloud provider minimize the cost by determining the minimum number of VMs needed to handle the offered workload and at the same time guaranteeing QoS by satisfying the SLA requirements. Precisely, from Fig. 5b, 21 number of VM instances are only needed to satisfy a response time of 0.12 s, given a workload of 2600 web requests/second. In the case of databases requests, from Fig. 6b, to satisfy a response time of 0.2 s, 20 VM instances are needed given a workload of 1700 database

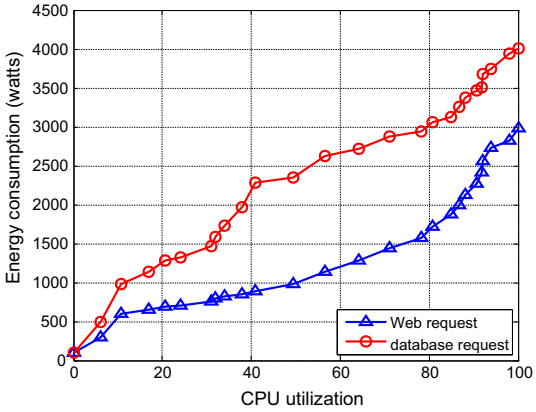

Fig. 7 Energy consumption curves in relation to CPU utilization

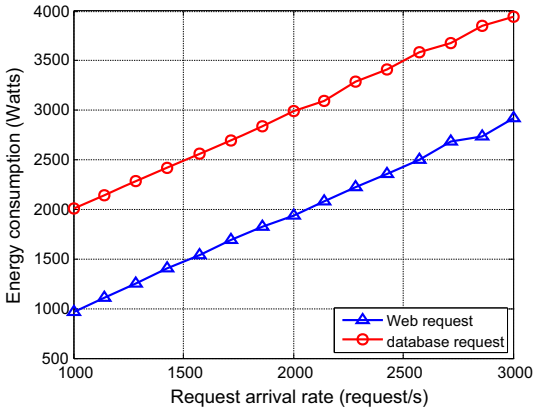

Fig. 8 Energy consumption curves in relation to arrival rate

requests/second. However, when the workloads are under approximately 1600 requests/second, a SLA response time latency of 0.15 s is well satisfied by using 22 VM nodes.

Evaluating Energy Consumption For estimating and evaluating the energy consumption in CDC, we assume that $P_{On} = 100W$ and $P_{Idle} = 0.6P_{On}$. Restarting an off PM necessitates a mean time of $1/\delta = 100s$. The parameter values used in this paper are based on empirical measurements from prior works [37,41,47–52]. Since the JMT tools cannot capture the CTMC energy consumption model, we used SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator) [60], for evaluating the energy consumption model. The SHARPE implements description techniques for Markov chains, multi-chain product-form queuing networks, and others reliability engineering models [61]. The energy consumption metrics as we vary the CPU utilization and request arrival rate are plotted in Figs. 7 and 8.

The change in the energy consumed by a PM in CDC is principally due to changes in the CPU utilization. To quantify these changes, we estimated the amount of energy drawn by a PM in the presence of an increasing requests arrival rate. Figure 7 plots the energy consumption for the cloud web and database requests in relation to CPU utilization. The curves reported in Fig. 7 show a dependency between the energy consumption and CPU utilization. The energy consumption has been evaluated with different values of CPU utilization in both cases of web and databases requests. The obtained results prove that as the CPU utilization increases, the energy consumption increases. We observe that with the web requests according to CPU utilization consume less energy when compared with database requests. This can be explained by the fact that the database request took longer CPU time than web request. When the CPU utilization reaches 100%, the energy consumption tends to 3000 Watts and 4000 Watts, respectively, in web requests and databases requests.

Figure 8 plots the energy consumption versus the requests arrival rate regarding two types of requests: web and databases requests. The figure shows that when the databases request arrival rate increases, the energy consumption increases as well which is natural and expected. Considering the web request arrival rate, it is evident that the consumption of energy is less than for database requests. Figure 8 quantifies the impact of both types of request rates.

7 Conclusion

In this article, we proposed analytical and simulation models that can be immensely useful for capacity engineering for cloud compute resources within a CDC environment. We considered the typical architecture in which a CDC has a collection of PMs and with each PM having multiple VMs. We derived closed-form formulas from analytical models for key performance measures such as throughput, response time, drop rate and CPU utilization. The analytical models were cross-validated by JMT simulator. In addition, we have presented an energy consumption model to estimate the energy consumption for a typical CDC environment. For this, we used SHARPE toolkit to study and estimate the energy consumption by varying the CPU utilization and the incoming request arrival rate for web and database request. We observed that the energy consumption in relation to CPU utilization is less for web requests than those for database requests. The analysis and the simulation results are in good agreement—which implies that our proposed analytical model is correct. As a future work, we plan to extend our model for study cloud-hosted containerized services and applications. We will consider both types of containers: VM hosted and PM hosted.

Acknowledgements The authors thank the anonymous reviewers for their valuable comments, which helped us to considerably improve the content, quality and presentation of this paper.

References

1. Voorsluys, W.; Broberg, J.; Buyya, R.: Introduction to cloud computing. Cloud computing: principles and paradigms, pp. 1–44 (2011)
2. Furht, B.: Cloud Computing Fundamentals. Handbook of Cloud Computing, pp. 3–19. Springer, US (2010)
3. El Kafhali, S.; Salah, K.: Performance analysis of multi-core VMs hosting cloud SaaS applications. Comput. Stand. Interfaces **55**, 126–135 (2018)
4. Huang, W.; Ganjali, A.; Kim, B.H.; Oh, S.; Lie, D.: The state of public infrastructure-as-a-service cloud security. ACM Comput. Surv. **47**(4), 68 (2015)
5. Alam, A.F.; Soltanian, A.; Yangui, S.; Salahuddin, M.A.; Glitho, R.; Elbiaze, H.: A cloud platform-as-a-service for multimedia conferencing service provisioning. In: Proceedings of the 21st IEEE Symposium on Computers and Communications, IEEE ISCC'16, Messina, Italy (2016)
6. Schafer, J.; Lichter, H.: Changes in requirements engineering after migrating to the software as a service model. In: Full-Scale Software Engineering/Current Trends in Release Engineering, pp. 25–30 (2016)
7. Amazon, E.: Amazon elastic compute cloud. Retrieved Feb, Vol. 10 (2009)
8. Ghosh, R.; Trivedi, K.S.; Naik, V.K.; Kim, D.S.: End-to-end performability analysis for infrastructure-as-a-service cloud: an interacting stochastic models approach. In: Proceedings of the 16th Pacific Rim International Symposium on Dependable Computing, PRDC'10, Tokyo, Japan, pp. 125–132 (2010)
9. Jennings, B.; Stadler, R.: Resource management in clouds: survey and research challenges. J. Netw. Syst. Manag. **23**(3), 567–619 (2015)
10. Kim, W.: Cloud computing: today and tomorrow. J. Object Technol. **8**(1), 65–72 (2009)
11. Chen, H.; Yao, D.D.: Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization, vol. 46. Springer, Berlin (2013)
12. Khojasteh, H.; Misic, J.; Misic, V.B.: Characterizing energy consumption of IaaS clouds in non-saturated operation. In: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), INFOCOM'14, Toronto, Canada, pp. 398–403 (2014)
13. Masdari, M.; Nabavi, S.S.; Ahmadi, V.: An overview of virtual machine placement schemes in cloud computing. J. Netw. Comput. Appl. **66**, 106–127 (2016)
14. Wen, Y.-F.: Energy-aware dynamical hosts and tasks assignment for cloud computing. J. Syst. Softw. **115**, 144–156 (2016)
15. Piraghaj, S.F.; Dastjerdi, A.V.; Calheiros, R.N.; Buyya, R.: Efficient virtual machine sizing for hosting containers as a service (services 2015). In: Proceedings of the IEEE 11th World Congress on Services, SERVICES'15, New York, pp. 31–38 (2015)
16. Xiao, Z.; Jiang, J.; Zhu, Y.; Ming, Z.; Zhong, S.; Cai, S.: A solution of dynamic VMs placement problem for energy consumption optimization based on evolutionary game theory. J. Syst. Softw. **101**, 260–272 (2015)
17. Li, K.: Power and performance management for parallel computations in clouds and data centers. J. Comput. Syst. Sci. **82**(2), 174–190 (2016)
18. Khazaei, H.; Misic, J.; Misic, V.B.: Performance of an IaaS cloud with live migration of virtual machines. In: Proceedings of the IEEE Global Communications Conference (GLOBECOM), GLOBECOM'13, Aalanta, USA, pp. 2289–2293 (2013)
19. Xiong, K.; Perros, H.: Service performance and analysis in cloud computing. In: Proceedings of the 1st IEEE Congress on Services, SERVICES'09, Los Angeles, California, USA, pp. 693–700 (2009)
20. Guo, L.; Yan, T.; Zhao, S.; Jiang, C.: Dynamic performance optimization for cloud computing using m/m/m queueing system. J. Appl. Math. **2014** (2014)
21. Bai, W.-H.; Xi, J.-Q.; Zhu, J.-X.; Huang, S.-W.: Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model. Math. Probl. Eng. **2015** (2015)
22. El Kafhali, S.; Salah, K.: Stochastic modelling and analysis of cloud computing data center. In: 20th ICIN Conference Innovations in Clouds, Internet and Networks, IEEE, Paris, France, pp. 122–126 (2017)
23. Ghosh, R.; Longo, F.; Naik, V.K.; Trivedi, K.S.: Modeling and performance analysis of large scale iaas clouds. Future Gener. Comput. Syst. **29**(5), 1216–1234 (2013)
24. Ghosh, R.; Longo, F.; Xia, R.; Naik, V.K.; Trivedi, K.S.: Stochastic model driven capacity planning for an infrastructure-as-a-service cloud. IEEE Trans. Serv. Comput. **7**(4), 667–680 (2014)
25. Mondal, S.K.; Muppala, J.K.; Machida, F.: Virtual machine replication on achieving energy-efficiency in a cloud. Electronics **5**(3), 37 (2016)
26. Sun, G.; Liao, D.; Anand, V.; Zhao, D.; Yu, H.: A new technique for efficient live migration of multiple virtual machines. Future Gener. Comput. Syst. **55**, 74–86 (2016)
27. Cheikh, H.B.; Doncel, J.; Brun, O.; Prabhu, B.: Predicting response times of applications in virtualized environments. In: Proceedings of the 3rd Symposium on Network Cloud Computing and Applications, NCCA'14, Rome, pp. 83–90 (2014)
28. Nguyen, T.A.; Kim, D.S.; Park, J.S.: Availability modeling and analysis of a data center for disaster tolerance. Future Gener. Comput. Syst. **56**, 27–50 (2016)
29. Salah, K.; Elbadawi, K.; Boutaba, R.: An analytical model for estimating cloud resources of elastic services. J. Netw. Syst. Manag. **24**(2), 285–308 (2016)
30. Boru, D.; Kliazovich, D.; Granelli, F.; Bouvry, P.; Zomaya, A.Y.: Energy-efficient data replication in cloud computing datacenters. Cluster Comput. **18**(1), 385–402 (2015)
31. Katz, R.H.: Tech titans building boom. IEEE Spectr. **46**(2), 40–54 (2009)
32. Salah, K.; El Kafhali, S.: Performance modeling and analysis of hypoexponential network servers. J. Telecommun. Syst. **65**(4), 717–728 (2017)
33. Vilaplana, J.; Solsona, F.; Teixido, I.; Mateo, J.; Abella, F.; Rius, J.: A queuing theory model for cloud computing. J. Supercomput. **69**(1), 492–507 (2014)
34. Bolch, G.; de Greiner, S.; Meer, H.; Trivedi, K.S.: Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications. Wiley, London (2006)
35. El Kafhali, S.; Salah, K.: Efficient and dynamic scaling of fog nodes for IoT devices. J. Supercomput. **73**(12), 5261–5284 (2017)
36. Nelson, R.: Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modeling. Springer, Berlin (2013)
37. Dayarathna, M.; Wen, Y.; Fan, R.: Data center energy consumption modeling: a survey. IEEE Commun. Surv. Tutor. **18**(1), 732–794 (2016)
38. Semeraro, G.; Magklis, G.; Balasubramonian, R.; Albonesi, D.H.; Dwarkadas, S.; Scott, M.L.: Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling. In: Proceedings of the IEEE 8th International Symposium on High Performance Computer Architecture, HPCA'02, Cambridge, pp. 29–40 (2002)
39. Beloglazov, A.; Buyya, R.: Energy efficient resource management in virtualized cloud data centers. In: Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGrid'10, Melbourne, Victoria, Australia, pp. 826–831 (2010)

40. Radhakrishnan, A.; Kavitha, V.: Energy conservation in cloud data centers by minimizing virtual machines migration through artificial neural network. Computing **98**(11), 1185–1202 (2016)
41. Gandhi, A.; Harchol-Balter, M.; Das, R.; Lefurgy, C.: Optimal power allocation in server farms. In: ACM SIGMETRICS Performance Evaluation Review, SIGMETRICS'09, vol. 37, no. 1, pp. 157–168 (2009)
42. Kusic, D.; Kephart, J.O.; Hanson, J.E.; Kandasamy, N.; Jiang, G.: Power and performance management of virtualized computing environments via lookahead control. Clust. Comput. **12**(1), 1–15 (2009)
43. Raghavendra, R.; Ranganathan, P.; Talwar, V.; Wang, Z.; Zhu, X.: No power struggles: coordinated multi-level power management for the data center. In: ACM SIGOPS Operating Systems Review, ASPLOS'08 vol. 42, no. 2, pp. 48–59 (2008)
44. Verma, A.; Ahuja, P.; Neogi, A.: pmapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, Middleware'08, Leuven, Belgium, pp. 243–264 (2008)
45. Awada, U.; Li, K.; Shen, Y.: Energy consumption in cloud computing data centers. Int. J. Cloud Comput. Serv. Sci. **3**(3), 145–162 (2014)
46. Yeo, S.; Hossain, M.M.; Huang, J.-C.; Lee, H.-H.S.: Atac: Ambient temperature-aware capping for power efficient datacenters. In: Proceedings of the ACM Symposium on Cloud Computing, SoCC'14, Seattle, WAACM, pp. 1–14 (2014)
47. Mazzucco, M.; Dyachuk, D.; Dikaiakos, M.: Profit-aware server allocation for green internet services. In: Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication System, MASCOTS'10, Miami, Florida, USA, pp. 277–284 (2010)
48. Gandhi, A.; Harchol-Balter, M.; Adan, I.: Server farms with setup costs. Perform. Eval. **67**(11), 1123–1138 (2010)
49. Burnetas, A.; Economou, A.: Equilibrium customer strategies in a single server markovian queue with setup times. Queueing Syst. **56**(3–4), 213–228 (2007)
50. Gandhi, A.; Harchol-Balter, M.: M/g/k with exponential setup, Tech. Rep. CMU-CS-09-166, School of Computer Science, Carnegie Mellon University (2009)
51. Nguyen, B.M.; Tran, D.; Nguyen, Q.: A strategy for server management to improve cloud service qos. In: Proceedings of the IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications, IEEE/ACM DS-RT'15, Chengdu, China, pp. 120–127 (2015)
52. Mazzucco, M.; Dyachuk, D.: Balancing electricity bill and performance in server farms with setup costs. Future Gener. Comput. Syst. **28**(2), 415–426 (2012)
53. Han, Z.; Tan, H.; Chen, G.; Wang, R.; Chen, Y.; Lau, F.: Dynamic virtual machine management via approximate markov decision process. In: Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM'16, San Francisco, CA, USA, pp. 1–9 (2016)
54. Elaydi, S.: An Introduction to Difference Equations. Springer, Berlin (2005)
55. Bahwaireth, K.; Benkhelifa, E.; Jararweh, Y.; Tawalbeh, M.A.: Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications. EURASIP J. Inf. Secur. **2016**(1), 1–14 (2016)
56. Tian, W.; Xu, M.; Chen, A.; Li, G.; Wang, X.; Chen, Y.: Open-source simulators for cloud computing: comparative study and challenging issues. Simul. Model. Pract. Theory **58**, 239–254 (2015)
57. Fahmy, H.M.A.: Simulators and emulators for WSNs. In: Wireless Sensor Networks. Signals and Communication Technology. Springer, Berlin, pp. 381–491 (2016)
58. Bertoli, M.; Casale, G.; Serazzi, G.: JMT: performance engineering tools for system modeling. ACM SIGMETRICS Perform. Eval. Rev. **36**(4), 10–15 (2009)
59. Sarna, D.E.: Implementing and Developing Cloud Computing Applications. CRC Press, Boca Raton (2010)
60. Trivedi, K.S.; Sahner, R.: Sharpe at the age of twenty two. ACM SIGMETRICS Perform. Eval. Rev. **36**(4), 52–57 (2009)
61. Sharpe. https://sharpe.pratt.duke.edu/