

Research Article

An Energy-Efficient Task Scheduling Mechanism with Switching On/Sleep Mode of Servers in Virtualized Cloud Data Centers

Chunxia Yin ^{1,2,3} Jian Liu,⁴ and Shunfu Jin ^{1,2}

¹College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China

²Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Yanshan University, Qinhuangdao 066004, China

³Ocean College, Hebei Agricultural University, Qinhuangdao 066003, China

⁴Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100090, China

Correspondence should be addressed to Shunfu Jin; jsf@ysu.edu.cn

Received 13 June 2019; Revised 9 January 2020; Accepted 16 January 2020; Published 18 February 2020

Guest Editor: Filipe J. Marques

Copyright © 2020 Chunxia Yin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the energy consumption of cloud data centers has continued to increase. A large number of servers run at a low utilization rate, which results in a great waste of power. To save more energy in a cloud data center, we propose an energy-efficient task-scheduling mechanism with switching on/sleep mode of servers in the virtualized cloud data center. The key idea is that when the number of idle VMs reaches a specified threshold, the server with the most idle VMs will be switched to sleep mode after migrating all the running tasks to other servers. From the perspective of the total number of tasks and the number of servers in sleep mode in the system, we establish a two-dimensional Markov chain to analyse the proposed energy-efficient mechanism. By using the method of the matrix-geometric solution, we mathematically estimate the energy consumption and the response performance. Both numerical and simulated experiments show that our proposed energy-efficient mechanism can effectively reduce the energy consumption and guarantee the response performance. Finally, by constructing a cost function, the number of VMs hosted on each server is optimized.

1. Introduction

Cloud computing is a paradigm that provides computing, communication and storage resources as a service on demand [1]. The cloud computing paradigm has experienced rapid growth, which has led to increases in the size and number of cloud data centers [2]. Data centers have experienced rapid growth in the number of hosted servers. Some large international business companies have more than 1 million servers in their data centers, including Google and Microsoft [3]. In 2017, the total number of cloud data centers deployed worldwide was estimated to have reached 8.6 million by the IDC (International Data Corporation) [4].

Consequently, the worldwide energy consumption of these massive data centers has increased dramatically. In 2010, electricity usage in global data centers accounted for approximately 1.3% of the total electricity usage worldwide

[5]. The total energy usage of data centers has increased dramatically each year. Recently, according to Forbes Technology Council's report in 2017, electricity usage in global data centers accounted for approximately 3% of the total electricity usage worldwide [6]. In addition to the enormous electricity costs, the energy usage of cloud data centers creates a number of environmental problems. Therefore, energy efficiency is now considered a chief concern for data center operators, even more so than the traditional considerations of availability and security [7]. Green cloud computing has become a trend in the development of current data centers. It is very important to solve the energy-efficient resource scheduling problems in cloud data centers [8].

One of the major reasons for energy inefficiency is the waste of idle power when data center servers are providing computing and storage capabilities at low utilization rates

[9]. The task scheduling algorithms have a significant impact on the energy consumption of large data centers [3]. Therefore, how to improve server utilization to save energy has become a difficult problem in cloud data centers that urgently needs to be resolved. Moreover, the rapid development of high-performance computing techniques requires increased response performance in cloud data centers. The trade-off between energy consumption and response performance has also become a hot research topic [10].

Currently, various approaches have been proposed to optimize and manage energy consumption in cloud data centers. Some approaches have investigated techniques such as processor speed controls [11], voltage adjustments [12], service migration [13], switching on servers, and hibernation or sleep modes [14]. Among these techniques, the popular scheme of switching idle servers to sleep mode or powering them down was studied by many researchers and helped provide good improvements in the energy efficiency and power consumption within data centers.

According to the previous work [15], most studies only consider switching the state of the physical server to improve energy efficiency. Few researchers have taken into account the server virtualization of the cloud data center. Meanwhile, cloud data center virtualization has become more popular. Other studies consider switching servers to a sleep mode after VMs migration. However, few of them model the energy-efficient scheme as a queueing system and the overhead of VM migration is not negligible. The goal of this paper is to study the energy-efficient mechanism in virtualized data centers based on multiserver vacation queueing theory. In this paper, we model an energy-efficient task scheduling mechanism with switching on/sleep mode of servers as a multiserver vacation queueing system and then analyse the system's performance. First, we introduce an energy-efficient task scheduling mechanism with the switching on/sleep mode of servers in a virtualized data center. Then, we establish a two-dimensional Markov chain and estimate the energy consumption and the response performance. We also present numerical results with analyses to show the validity of the novel energy-efficient task scheduling mechanism. Finally, by constructing a cost function, the number of VMs that can be hosted on each server is optimized.

We make three contributions to solve the research problem of the energy-efficient task scheduling of virtualized cloud data centers.

- (1) We propose an energy-efficient task scheduling mechanism with switching servers on/sleep mode for virtualized cloud data centers. When the number of idle VMs reaches a specified threshold, the server with the maximum idle VMs will be switched to sleep mode after migrating all the running tasks to other servers. In this way, the idle energy consumption in the virtualized cloud data center can be reduced.
- (2) We present a method to model the proposed mechanism and evaluate the system performance mathematically. We model the virtualized cloud data center with the proposed mechanism as an $M/M/n$

vacation queueing system. By constructing a two-dimensional Markov chain, we evaluate the system performance in terms of the response performance and the energy consumption.

- (3) We construct a cost function by considering the system performance metrics for virtualized cloud data centers. The optimal number of VMs hosted on each server is obtained using the cost function.

The remainder of this paper is organized as follows. In Section 2, we summarize the related work for the technique of switching idle servers off or to sleep in cloud data centers. In Section 3, we propose an energy-efficient task scheduling mechanism with switching on/sleep mode of servers and introduce the features of the mechanism and then construct the system model and assess the stability of the system. In Section 4, we analyse the queueing model by the matrix-geometric solution method. In Section 5, we derive the formulas for the performance measures and then demonstrate the influences of the system parameters on the system performance with numerical results. In Section 6, by constructing a cost function, we optimize the number of VMs that are hosted on each server. Finally, we draw the conclusions in Section 7.

2. Related Work

2.1. Energy-Efficient Strategies of Physical Servers. Some researchers employ the technique of switching idle servers off or to sleep mode to conserve energy.

In [16], Cheng et al. applied vacation queueing theory to model a cloud computing system and analysed the expectations of the task sojourn time and the energy consumption of a cloud computing system. Based on their analysis, they proposed a task scheduling algorithm based on similar tasks to optimize the energy consumption and evaluate the performance through simulations.

In [17], Phung-Duc and Kawanishi considered an $M/M/c/c$ retrial queue where a server in the queueing system corresponds to one physical server in data centers. They analysed the multiserver retrial queue with setup time by a level dependent QBD process and established a sufficient condition of the stability for the level dependent QBD process. Finally, they proposed a heuristic technique to determine the truncation point, which has been shown to be practically useful.

In [18], Meisner and Wenisch use both DPM (Dynamic Power Management) with the sleep state and dynamic voltage scaling to reduce the power consumption and meet response time constraints. This work has been done considering a single physical server. One server is modelled as an $M/G/1/PS$ queue. The goal of this work is to minimize the mean power consumption of a server under a given response time.

In [19], Vakili et al. proposed an energy-efficient strategy to turn off a server pool to save power if the servers are not currently serving any jobs. They developed a mathematical model using queueing theory to analyse the performance of the proposed strategy for a data center with

N server pools. Moreover, they considered four schemes, i.e., always on, reactive, proactive, and optimal prediction, and compared their performances.

In [20], Sakata and Kasahara first modelled a data center as a multiserver queueing system. The system alternates independently between two modes, i.e., the normal-operating mode and the power-saving mode, according to the number of jobs in the system. When the system works in the power-saving mode, part of the servers in each cluster are powered off after completing the existing job. Second, the authors constructed a continuous-time Markov chain for the system and derived the steady-state probability vector by the matrix-geometric method. Finally, they investigated how the performance measures are affected by the system parameters, including the number of clusters and the energy efficiency grade.

In [21], Nguyen et al. proposed a novel three-state model for cloud servers. The model was deployed in both single and multiple finite capacity queues. They also proposed several strategies to control multiple queues simultaneously. This approach reduces the service waiting time for jobs and elastically manages the service capability for the system. The power consumption was also evaluated to provide insights into the system performance with respect to the energy-performance trade-off.

In [22], Wang et al. modelled a data center as a heavy traffic approximate large-scale queueing system, and then they designed a load routing mechanism between multiple data centers to reduce the energy costs. Based on multiserver vacation queueing theory, the authors presented a dual-threshold to dynamically adjust the various states (On/Off/Idle) of a single server to reduce the energy costs in the cloud data center.

In [23], Liao et al. minimized the power consumption by switching on/off a certain group of servers according to whether the current queue size reached the threshold. They introduced two activation thresholds and modelled the data center as an $M/M/n + m1 + m2$ queueing system. Finally, they developed a mathematical model using queueing theory to determine the activation threshold.

In [24], Cheng et al. introduced an $M/M/n$ queueing model with impatient tasks to model the task scheduling in the cloud data center. Based on the $M/M/n$ queueing system model, they analysed the average number of task betrayals and the power consumption costs. To save energy, they proposed a strategy of dynamically adjusting the threshold of the task number. A server is chosen to be awakened according to the cold area where the server located and the length of its shutdown time.

2.2. Energy-Efficient Strategies of Virtual Systems. The above studies are focused on switching the state of the physical servers to improve the energy efficiency; however, the server virtualization is neglected. Fewer studies propose the strategy of shutting down a subset of servers or switching some servers to sleep mode in virtual systems.

In [25], Shen et al. investigated the problem of “right-sizing” a cloud data center for energy efficiency through

virtualization, which allows for the consolidation of workloads into smaller number of servers while dramatically powering off the idle ones.

In [26], Deng et al. proposed a VM consolidation based size decision approach, which migrates VMs from underutilized servers to more active ones according to the monitored resource utilization in real time and then turns off the underutilized servers.

In cloud data centers, the arrival of tasks is batch and random. The capability of the servers decreases as the number of processing tasks increases. Large-scale task is generally coordinated by a cluster of multiple servers. Therefore, the task scheduling process of the cloud data center is very consistent with the classic $M/M/n$ multiserver queueing model.

3. Energy-Efficient Mechanism and Model Description

In this section, we propose an energy-efficient task scheduling mechanism with switching servers on/sleep for virtualized cloud data centers and introduce the features of the mechanism. Then, we model the data center with the energy-efficient mechanism as an $M/M/n$ queueing system with a threshold vacation policy.

3.1. An Energy-Efficient Mechanism. To ensure responsiveness, many servers in conventional data centers are always on, even though there are not any tasks to be processed. Many idle VMs lead to a waste of energy. To reduce the extra energy consumption of idle VMs, we propose an energy-efficient mechanism with switching servers on/sleep in a virtualized cloud data center with some homogeneous physical servers. In the cloud data center, each physical server hosts a fixed number of VMs with a fixed number of CPU cores and random access memory. The fixed number of VMs hosted on each server is a threshold, say d . VMs are created, managed, and assigned by the cloud management system (CMS). When users submit tasks to the cloud center, the CMS allocates VMs to process the tasks. Each task is assigned to one VM.

In our proposed energy-efficient mechanism, the server will be switched between two modes, namely, on mode and sleep mode. When a server is in on mode, all the VMs hosted on this server are activated and can run incoming tasks one by one. When a server is in sleep mode, it means that all VMs hosted on the server are closed and the server is deactivated. The proposed energy-efficient mechanism is detailed as follows:

- (1) The proposed energy-efficient task scheduling mechanism assumes that all the tasks can be scheduled by the CMS to continue running on other VMs during the running process.
- (2) The CMS monitors the total number of idle VMs in real time. The idle VMs will be distributed among different servers. If the number of idle VMs reaches d , the CMS will schedule all the running tasks on the

server with the most idle VMs, migrating these tasks from the busy VMs to other idle VMs on other servers.

- (3) The scheduling process of a running task mainly includes the following two steps. First step: the CMS sends messages to the VMs on the selected server to terminate the execution of the running task, and then the VMs stop executing the running task. Second step: the CMS sends messages to the idle VMs on the other server to perform the running task with the relevant parameters, and then these idle VMs start to continue the running task.
- (4) The CMS switches the physical server without busy VMs to sleep mode.
- (5) Once a physical server switches to sleep mode, a sleep timer will be started, and the sleep period is set by this sleep timer. If there are no tasks waiting to run in the system, the server will start another sleep period once the sleep timer is over. Conversely, if there are tasks waiting to run in the system, the server will be switched to on mode and d hosted VMs will run the waiting tasks immediately.

We note that the energy consumption in sleep mode is lower than that in on mode. The energy consumption will be reduced in the data center with the proposed energy-efficient mechanism.

3.2. Features of the Energy-Efficient Task Scheduling Mechanism. In current cloud data centers, Torque [27], Slurm [28], Google Cloud Scheduler [29], Cloud Foundry [30], and other resource management systems of the server clusters provide many functions, and related APIs, including system status monitoring, task scheduling, server operating mode switching, and so on. These functions and APIs provide support for the efficient operation and energy saving of server cluster systems in cloud data centers.

In the proposed energy-efficient mechanism, VMs switch the operating mode in units of VM groups, avoiding thrashing of the VM clusters, which is caused by frequent switching on/sleep mode of a small number of VMs in a short period of time. In this paper, the proposed energy-efficient mechanism is based on the sleep mode (low power mode) of the physical servers. In the sleep mode, the physical server is not shut down, and the runtime resources of the VMs hosted on the physical server are still stored in the memory of the physical server. It takes a short time for physical servers to switch operating modes between on and sleep. Once the operating mode of a physical server is switched from sleep to on, all of the VMs hosted on the physical server can immediately perform tasks. The proposed energy-efficient mechanism does not involve the process of switching operating mode of the VMs from sleep to on. Therefore, the time for the VMs to switch from nonserviceable to serviceable in the proposed energy-efficient mechanism is very short.

In virtualized cloud data centers, the proposed energy-efficient mechanism avoids thrashing of the VM clusters in

the process of implementing energy-efficient task scheduling. And in the proposed energy-efficient mechanism, the method of switching operating modes of physical servers is applied to improve system response performance.

3.3. Model Description. To analyse the performance metrics of the cloud data center, many studies assume that the service times of tasks are exponentially distributed and model the cloud data centers as $M/M/n$ queueing systems [21, 22, 24]. In our paper, we also assume that the service times of tasks are exponentially distributed. The assumptions are as follows.

We assume that the cloud data center consists of M ($M \geq 1$), homogeneous physical servers, and each server hosts d ($d \geq 1$) VMs. The total number of VMs in the cloud center can be denoted as $N = Md$. We assume users' tasks arrive at the data center according to a Poisson process with the parameter λ ($\lambda > 0$), and the service rate of each VM is exponentially distributed with parameter μ ($\mu > 0$). The sleep period time of the sleep timer for each sleeping server follows an exponential distribution with parameter θ ($\theta > 0$). Tasks are independent and processed on a First-come First-served (FCFS) basis. Random scheduling algorithm is selected as the VM scheduling strategy. If there are idle VMs, the tasks will be processed immediately, and otherwise they wait in the queue. We assume that the migration time of the task is so fast that we can ignore it. Based on these assumptions above, we model the cloud center with the proposed energy-efficient mechanism as an $M/M/n$ queueing system with a threshold vacation policy, as shown in Figure 1.

From the perspective of the number of tasks and the number of servers in sleep mode in the system, we construct a two-dimensional Markov chain to estimate the system's performance.

Let $L(t) = i$ be the number of tasks to be sent to the data center at time t and $J(t) = j$ be the number of physical servers in sleep mode at time t . Because all random variables are exponentially distributed, $\{L(t), J(t)\}$ constitutes a two-dimensional continuous-time stochastic process with the state space Ω as

$$\Omega = \{(i, j) \mid i \in \{1, 2, \dots\}, j \in \{M, (M-1), \dots, 0\}\}. \quad (1)$$

We define $\pi_{i,j}$ as the steady-state probability when the number of tasks is i and the number of physical servers in sleep mode is j in the system. $\pi_{i,j}$ is then as

$$\pi_{i,j} = \lim_{t \rightarrow \infty} P\{L(t) = i, J(t) = j\}. \quad (2)$$

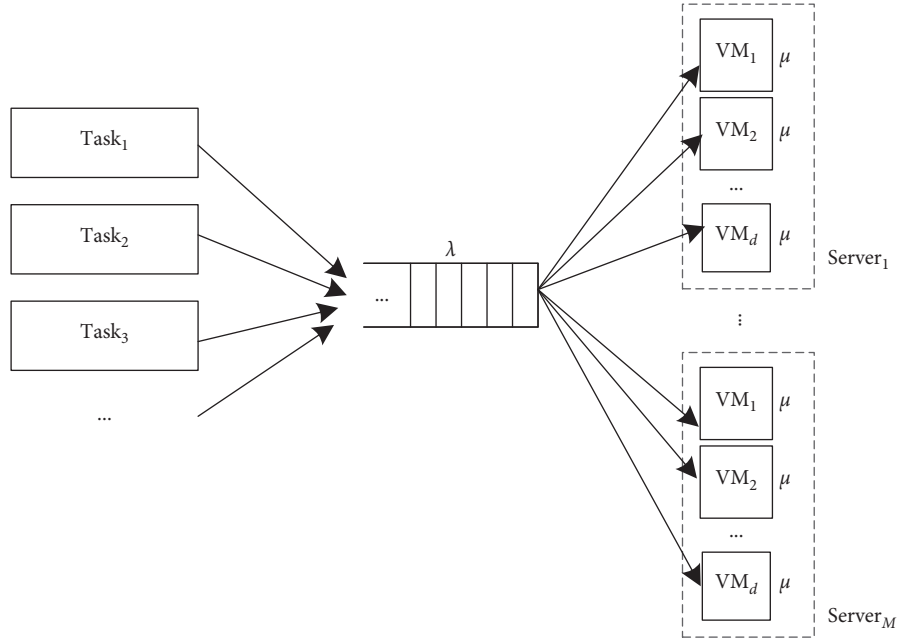
We define π_i as the steady-state probability vector where the number of tasks is i . Given that $i = 0$, π_0 is given as

$$\pi_0 = \pi_{0,M}. \quad (3)$$

Given that $kd + 1 \leq i \leq (k+1)d$ ($k = 0, 1, \dots, M-2$), π_i is given as

$$\pi_i = (\pi_{i,M}, \pi_{i,(M-1)}, \dots, \pi_{i,(M-k-1)}). \quad (4)$$

Given that $i > (M-1)d$, π_i is given as

FIGURE 1: The $M/M/n$ queueing system for a virtualized cloud data center.

$$\pi_i = (\pi_{i,M}, \pi_{i,(M-1)}, \dots, \pi_{i,1}, \pi_{i,0}). \quad (5)$$

We define Π as the steady-state probability distribution of the two-dimensional continuous-time Markov chain, and Π is composed of π_i . Π is given as

$$\Pi = (\pi_0, \pi_1, \pi_2, \dots). \quad (6)$$

4. Model Analysis

4.1. Transition Rate Matrix. We define \mathbf{Q} as the one-step transition rate matrix of the two-dimensional continuous-time Markov chain. Considering the number of tasks in the system, we give the transition rate matrix \mathbf{Q} in a block structure. The matrix \mathbf{Q} is composed by several submatrices. The submatrix \mathbf{B}_i represents the one-step transition rate for the number of tasks changing from i to $i-1$. The submatrix \mathbf{C}_i represents the one-step transition rate for the number of tasks changing from i to $i+1$. The submatrix \mathbf{A}_i represents the one-step transition rate for the number of tasks fixed at i . According to different numbers of tasks in the system, we address \mathbf{B}_i , \mathbf{A}_i , and \mathbf{C}_i with the following conditions.

- (1) Given that $i = 0$, it means that there are no tasks in the system. In this case, all the servers in the system are in sleep mode and the total number of sleeping physical servers is M . The submatrices \mathbf{A}_0 and \mathbf{C}_0 are given as follows:

$$\begin{aligned} \mathbf{A}_0 &= -\lambda, \\ \mathbf{C}_0 &= \lambda. \end{aligned} \quad (7)$$

- (2) Given that $i = kd + 1$ ($k = 0, 1, \dots, M-1$), the possible values for the j sleeping physical servers are

$j = (M, M-1, \dots, M-k-1)$. The transition matrices \mathbf{B}_i , \mathbf{A}_i , and \mathbf{C}_i are given as follows:

$$\begin{aligned} \mathbf{B}_i &= \begin{bmatrix} 0 & & & & \\ & d\mu & & & \\ & & 2d\mu & & \\ & & & \ddots & \\ & & & & kd\mu \\ & & & & & i\mu \end{bmatrix}, \\ \mathbf{A}_i &= \begin{bmatrix} \gamma_0 & M\theta & & & \\ & \gamma_1 & (M-1)\theta & & \\ & & \gamma_2 & (M-2)\theta & \\ & & & \ddots & \ddots \\ & & & & \gamma_{k+1} & (M-k-1)\theta \\ & & & & & -(i\mu + \lambda) \end{bmatrix}, \end{aligned} \quad (8)$$

where $\gamma_i = -((M-i)\theta + id\mu + \lambda)$, $0 \leq i \leq k$.

$$\mathbf{C}_i = \lambda \mathbf{I}_{(k+2) \times (k+2)}, \quad (9)$$

where $\mathbf{I}_{(k+2) \times (k+2)}$ is an identity matrix of order $(k+2)$.

- (3) Given that $kd + 1 < i \leq (k+1)d$ ($k = 0, 1, \dots, M-2$) and $kd + 1 < i < (k+1)d$ ($k = M-1$), the possible values for the j sleeping physical servers are $j = \{M, M-1, \dots, M-k-1\}$. The transition matrices \mathbf{B}_i , \mathbf{A}_i , and \mathbf{C}_i are given as follows:

$$\mathbf{B}_i = \begin{bmatrix} 0 & & & & \\ & d\mu & & & \\ & 2d\mu & & & \\ & & \ddots & & \\ & & & kd\mu & \\ & & & & i\mu \end{bmatrix},$$

$$\mathbf{A}_i = \begin{bmatrix} \gamma_0 & M\theta & & & \\ & \gamma_1 & (M-1)\theta & & \\ & & \gamma_2 & (M-2)\theta & \\ & & & \ddots & \ddots \\ & & & & \gamma_{k+1} & (M-k-1)\theta \\ & & & & & -(i\mu + \lambda) \end{bmatrix}, \quad (10)$$

where $\gamma_i = -((M-i)\theta + i d\mu + \lambda)$, $0 \leq i \leq k$.

$$\mathbf{C}_i = \lambda \mathbf{I}_{(k+2) \times (k+2)}, \quad (11)$$

where $\mathbf{I}_{(k+2) \times (k+2)}$ is an identity matrix of order $(k+2)$.

- (4) Given that $i \geq N = Md$, the possible values for the j sleeping physical servers are $j = \{M, M-1, \dots, 1, 0\}$. When the number of tasks in the system reaches N , the submatrices in \mathbf{Q} are repeated forever. The repeated submatrices are \mathbf{B} , \mathbf{A} , and \mathbf{C} , which are given as follows.

$$\mathbf{B} = \begin{bmatrix} 0 & & & & \\ & d\mu & & & \\ & 2d\mu & & & \\ & & \ddots & & \\ & & & (M-1)d\mu & \\ & & & & Md\mu \end{bmatrix}, \quad (12)$$

$$\mathbf{A} = \begin{bmatrix} \gamma_0 & M\theta & & & \\ & \gamma_1 & (M-1)\theta & & \\ & & \gamma_2 & (M-2)\theta & \\ & & & \ddots & \ddots \\ & & & & \gamma_{N-1} & \theta \\ & & & & & \gamma_N \end{bmatrix},$$

where $\gamma_i = -((M-i)\theta + i d\mu + \lambda)$, $0 \leq i \leq N$.

$$\mathbf{C} = \lambda \mathbf{I}_{(N+1) \times (N+1)}, \quad (13)$$

where $\mathbf{I}_{(N+1) \times (N+1)}$ is an identity matrix of order $(N+1)$.

Based on the discussions above, all the submatrices in the \mathbf{Q} have been addressed. The \mathbf{Q} is a partitioned matrix

TABLE 1: Iterative algorithm to obtain the rate matrix \mathbf{R} .

Step	Operation
	Initialize the error precision ε (for example, $\varepsilon = 10^{-8}$)
Step1	Initialize M, d, λ, μ , and θ as needed Initialize the rate matrix $\mathbf{R} = 0$
Step2	Input \mathbf{B}, \mathbf{A} , and \mathbf{C}
Step3	Calculate \mathbf{R}^* by $\mathbf{R}^* = \mathbf{R}^2 \times \mathbf{B} + \mathbf{R} \times (\mathbf{A} + \mathbf{I}) + \mathbf{C}$ % \mathbf{I} is an identity matrix % if $(\ \mathbf{R}^* - \mathbf{R}\ _{\infty} > \varepsilon)$ { $\mathbf{R} = \mathbf{R}^*$
Step4	$\mathbf{R}^* = \mathbf{R}^2 \times \mathbf{B} + \mathbf{R} \times (\mathbf{A} + \mathbf{I}) + \mathbf{C}$ Go to Step 4} else {Go to Step 5}
Step5	$\mathbf{R} = \mathbf{R}^*$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{C}_0 & & & \\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{C}_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{B}_{N-1} & \mathbf{A}_{N-1} & \mathbf{C}_{N-1} \\ & & & \mathbf{B} & \mathbf{A} & \mathbf{C} \\ & & & & \ddots & \ddots & \ddots \end{bmatrix}. \quad (14)$$

4.2. Matrix-Geometric Solution. To analyse the two-dimensional continuous-time Markov chain $\{L(t), J(t)\}$ with the one-step state transition rate matrix \mathbf{Q} , we need to solve the minimal nonnegative solution of the matrix quadratic equation $\mathbf{R}^2 \mathbf{B} + \mathbf{R} \mathbf{A} + \mathbf{C} = 0$. The solution \mathbf{R} is called the rate matrix. The rate matrix \mathbf{R} plays a crucial role in the analysis of the proposed mathematical queueing model. Under the condition that $\rho = \lambda(N\mu) - 1 < 1$, the matrix quadratic equation $\mathbf{R}^2 \mathbf{B} + \mathbf{R} \mathbf{A} + \mathbf{C} = 0$ has a minimal nonnegative solution \mathbf{R} and the spectral radius $\mathbf{SP}(\mathbf{R}) < 1$ [31].

It is difficult to give the expression of the rate matrix \mathbf{R} in a closed form with the higher-order matrix equation. Here, we give an iterative algorithm to obtain the rate matrix \mathbf{R} numerically. The main steps of the algorithm are shown in Table 1.

To ensure the convergence of the algorithm, the accuracy ε is given in the algorithm. The algorithm will repeatedly determine whether the value of $\|\mathbf{R}^* - \mathbf{R}\|_{\infty}$ is greater than ε . When the value of $\mathbf{R}^* - \mathbf{R}$ is not greater than ε , the \mathbf{R} is the approximate solution to meet the accuracy ε . With the rate matrix \mathbf{R} obtained by the iterative algorithm in Table 1, we construct a square matrix $\mathbf{B}[\mathbf{R}]$ as follows:

$$\mathbf{B}[\mathbf{R}] = \begin{bmatrix} \mathbf{A}_0 & \mathbf{C}_0 & & & \\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{C}_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{B}_{N-1} & \mathbf{A}_{N-1} & \mathbf{C}_{N-1} \\ & & & \mathbf{B} & \mathbf{R}\mathbf{B} + \mathbf{A} \end{bmatrix}. \quad (15)$$

By using the balance equation and the normalized condition, we give the system of equations as follows, where \mathbf{e} is a column vector with ones.

$$\begin{cases} \mathbf{\Pi Q} = 0, \\ \mathbf{\Pi e} = 1. \end{cases} \quad (16)$$

We conclude that \mathbf{Q} is the one-step transition matrix of a Quasi Birth-and-Death Process. By using the geometric-matrix method solution [32], we can give the system of equations as follows:

$$\mathbf{\Pi} = (\pi_0, \pi_1, \pi_2, \dots) \begin{Bmatrix} \pi_{0,M} & \pi_{1,M} & \cdots & \pi_{(M-1)d,M} & \pi_{Md,M} & \cdots \\ 0 & \pi_{1,(M-1)} & \cdots & \pi_{(M-1)d,(M-1)} & \pi_{Md,(M-1)} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots \\ 0 & 0 & \cdots & \pi_{(M-1)d,1} & \pi_{Md,1} & \cdots \\ 0 & 0 & \cdots & 0 & \pi_{Md,0} & \cdots \end{Bmatrix}. \quad (18)$$

In equation (18), $\mathbf{\Pi}$ is composed of the vectors of π_i , and π_i is the steady-state probability vector where the number of tasks is i . π_i is composed of the values of $\pi_{i,j}$, and $\pi_{i,j}$ is the steady-state probability when the number of tasks is i and the number of physical servers in sleep mode is j in the system. When i goes to infinity, the sum of $\pi_{i,j}$ is 1.

5. Performance Analysis and Numerical Results

5.1. Performance Analysis

- (1) Let L_s be the average number of tasks in the system. According to Little's law, we can obtain the mean sojourn time of tasks as W_s . Therefore, we have

$$\begin{aligned} L_s &= \sum_{i=0}^{\infty} i \cdot \pi_{i,M} + \sum_{i=1}^{\infty} i \cdot \pi_{i,(M-1)} + \sum_{i=d+1}^{\infty} i \cdot \pi_{i,(M-2)} \\ &\quad + \cdots + \sum_{i=(M-1)d+1}^{\infty} i \cdot \pi_{i,0}, \\ W_s &= \frac{L_s}{\lambda}. \end{aligned} \quad (19)$$

- (2) Let L_{busy} be the average number of busy VMs in the system, L_{idle} be the average number of idle VMs in the system, and L_{sleep} be the average number of physical servers in sleep mode. The average number of physical servers in on mode is $M - L_{\text{sleep}}$. The power consumed by a server in on mode is determined by the power of the physical server in on mode and the power of the VMs that are hosted on the physical server. We define the power consumed by a physical server in on mode as P_{on} and the power consumed by a physical server in sleep mode as P_{sleep} . We define the power consumed by a busy VM as P_{busy} and the

$$\begin{cases} (\pi_0, \pi_1, \dots, \pi_N) \mathbf{B}[\mathbf{R}] = 0, \\ \sum_{i=0}^{N-1} \pi_i \mathbf{e} + \pi_N (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e} = 1, \\ \pi_i = \pi_N \mathbf{R}^{i-N}, i \geq N. \end{cases} \quad (17)$$

By applying the Gauss-Seidel method [33] to solve equation (17), $\pi_0, \pi_1, \dots, \pi_i \dots$ can be obtained. Then, the steady-state probability distribution $\mathbf{\Pi} = (\pi_0, \pi_1, \pi_2, \dots)$ of the system can be given numerically.

power of consumed by an idle VM as P_{idle} . Let P_{total} be the total energy consumption. We have

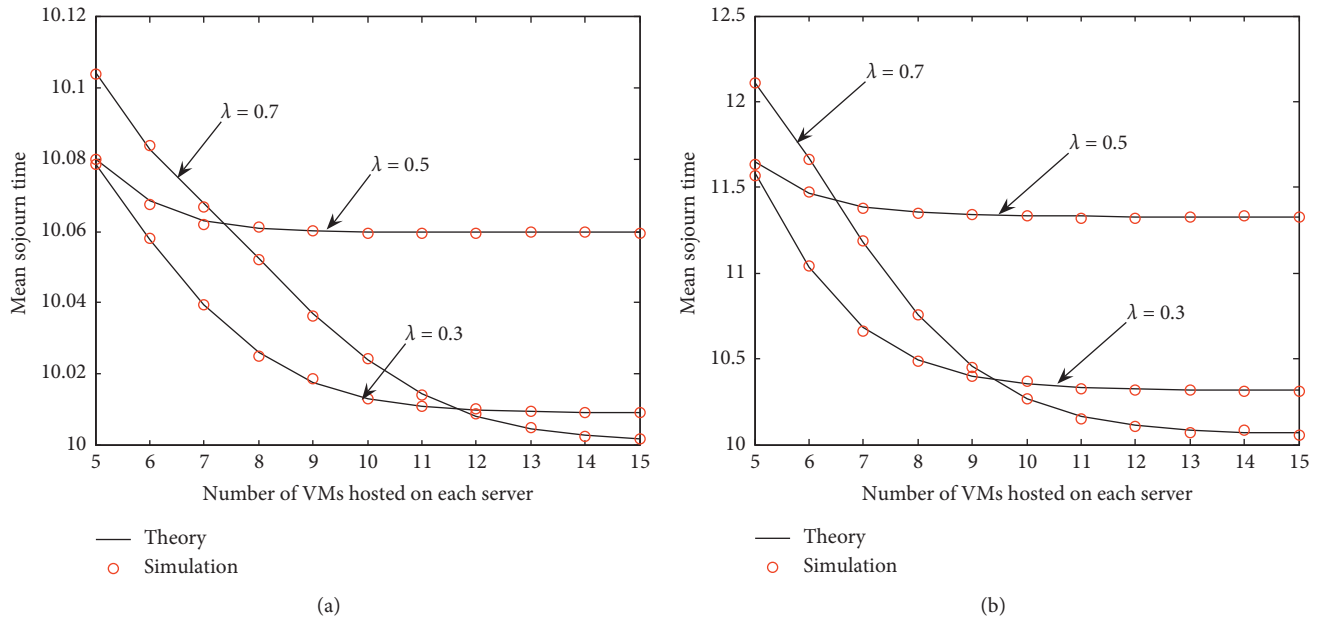
$$\begin{aligned} P_{\text{total}} &= P_{\text{busy}} L_{\text{busy}} + P_{\text{idle}} L_{\text{idle}} + P_{\text{on}} (M - L_{\text{sleep}}) \\ &\quad + P_{\text{sleep}} L_{\text{sleep}}, \\ L_{\text{sleep}} &= \sum_{j=1}^M \sum_{i=0}^{\infty} j \pi_{i,j}, \\ L_{\text{idle}} &= \sum_{j=1}^M \sum_{(j-1)d+1}^{jd} ((M-j)d - i) \pi_{i,j} \\ &\quad + \sum_{(M-1)d+1}^{Md} (Md - i) \pi_{i,0}, \\ L_{\text{busy}} &= \sum_{j=1}^M \sum_{(j-1)d+1}^{jd} i \pi_{i,j} + \sum_{j=1}^M \sum_{jd+1}^{\infty} (M-j)d \pi_{i,j} \\ &\quad + \sum_{(M-1)d+1}^{Md} i \pi_{i,0} + \sum_{Md+1}^{\infty} Md \pi_{i,0}. \end{aligned} \quad (20)$$

5.2. Numerical Results. To evaluate the system performance with the proposed energy-efficient mechanism, we provide numerical experiments and simulation experiments with analyses. The results are calculated using Matlab 2011a. The simulation program is written in Java language. The simulation results are obtained by averaging the results from 100 independent runs.

We assume there are 10 physical servers hosted in the data center, and each server can host d VMs. Each VM is configured with a fixed number of CPU cores and random access memory. Each physical server can host at least 5 and at most 20 VMs. The number of VMs that can be hosted on each server can be dynamically adjusted according to the

TABLE 2: The parameter set.

Parameter	Value	Explanation
μ	0.1	The service rate of a VM
M	10	The number of physical servers in a data center
P_{on}	10 W	Power use of a physical server in on mode
P_{busy}	1 W	Power use of a busy VM
P_{idle}	0.1 W	Power use of an idle VM
P_{sleep}	0.5 W	Power use of a physical server in sleep mode

FIGURE 2: The mean sojourn times of tasks. (a) $\theta = 0.2$. (b) $\theta = 0.02$

workload. The parameters of the numerical experiments are set in Table 2.

Figure 2 examines how the mean sojourn time of tasks changes in relation to the number of VMs hosted on each server with different task arrival rates. From Figures 2(a) and 2(b), we find that, for the three different arrival rates, when the number of VMs hosted on each server increases, the mean sojourn time decreases. The reason is that as the number of VMs hosted on each physical server increases, the total number of VMs will increase. This means that there are more servers to execute the tasks in the queueing system, and consequently the mean sojourn time of tasks will decrease.

By comparing the results of Figures 2(a) and 2(b), we see that the mean sojourn times of tasks increase as the sleeping parameter θ decreases. The reason is that $1/\theta$ is the average length of the sleep period time of the sleep timer. As the sleeping parameter θ decreases, the sleep time of the server in sleep mode will be longer. Then, the tasks will wait for a longer time in the buffer and the mean sojourn time of tasks will increase.

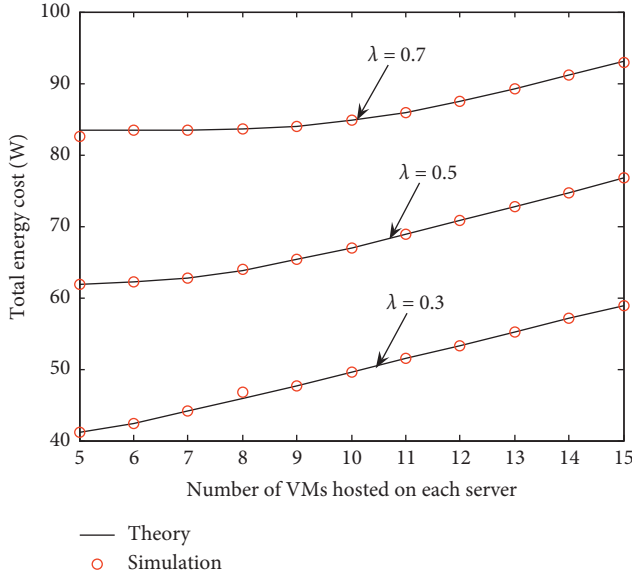
According to the classic multiserver vacation queueing theory, the sleeping parameter θ reflects the idleness of the VMs, and the mean sojourn time reflects the task execution performance. The smaller the value of θ , the longer the sleep time of the VMs, and the greater the probability that the

VMs are busy. The longer the mean sojourn time, the worse the QoS.

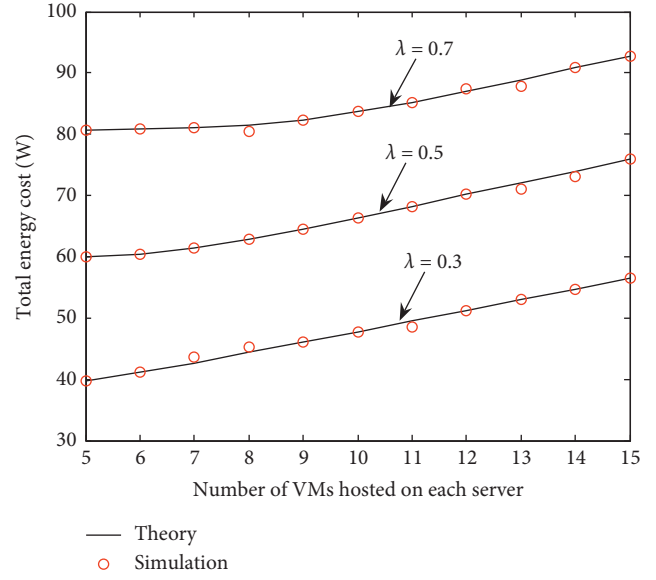
As shown in Figure 2, the value of θ of Figure 2(b) is smaller than the value of θ of Figure 2(a), indicating that the VMs in Figure 2(b) are more busy. Under the condition that the number of VMs hosted on each server is the same and the value of θ is the same, the value of the mean sojourn time in Figure 2(b) is smaller than the value of the mean sojourn time in Figure 2(a). This shows that when the number of busy VMs hosted on each server is large, the QoS of the system deteriorates, and the service level is reduced.

Figure 3 examines how the total energy costs change in relation to the number of VMs hosted on each server with different task arrival rates.

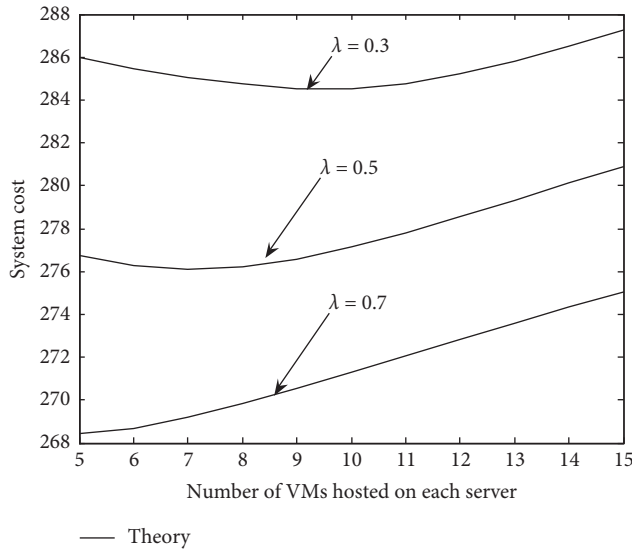
From Figures 3(a) and 3(b), we find that, for the three different arrival rates, when the number of VMs hosted on each server increases, the total energy costs increase. The reason is that as the number of VMs hosted on each physical server increases, the total number of VMs will increase. This means that there are more servers to execute the tasks in the queueing systems, and consequently the total energy costs will increase. By comparing the results of Figures 3(a) and 3(b), we see that the total energy costs of tasks decrease as the sleeping parameter θ decreases. The reason is that $1/\theta$ is the average length of the sleep period time of the sleep timer. As



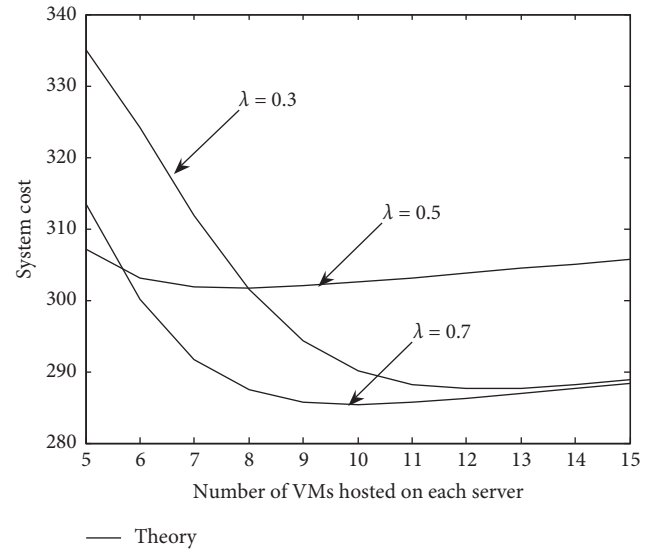
(a)



(b)

FIGURE 3: Total energy consumption. (a) $\theta = 0.2$. (b) $\theta = 0.02$ 

(a)



(b)

FIGURE 4: The system cost. (a) $\theta = 0.2$. (b) $\theta = 0.02$

the sleeping parameter θ decreases, the sleep time of the server in sleep mode will be longer. A server in sleep mode consumes less energy, and so the total energy consumption of systems will decrease.

6. System Optimization

To ensure the response performance, one factor affecting the system costs is the mean sojourn time of tasks. As the mean sojourn time of tasks increases, the total numbers of tasks processed in the system per unit of time will decrease. From the perspective of reducing traffic in the throughput-sensitive system, this factor should be set higher. Another

factor affecting the system costs is the total energy consumption. As the total energy consumption increases, the electricity costs for the data center increase too. From the perspective of improving energy efficiency in the energy-efficient system, this factor should be set higher.

To obtain the optimal amount of VMs hosted on one server, we construct a system cost function as follows:

$$C(d) = f_1 W_s + f_2 P_{\text{total}}, \quad (21)$$

where f_1 is the impact factor of the system costs introduced by the mean sojourn time of tasks and f_2 is the impact factor of the system costs introduced by the energy consumption.

TABLE 3: The optimal parameter d .

λ	$\theta = 0.2$	$\theta = 0.02$
0.3	10	12
0.5	7	7
0.7	5	10

The values of f_1 and f_2 are set as needed in practice. In this paper, as an example, we set f_1 set to 25 and f_2 to 0.4. The change trends for the system costs are shown in Figure 4.

Figure 4 examines how the system costs change in relation to the number of VMs hosted on each server with different task arrival rates. From Figures 4(a) and 4(b), we can see that when the number of VMs hosted on each server increases, the system costs decrease. The reason is that the main factor affecting the system costs is the mean sojourn time of tasks, and when the number of VMs hosted on each server increases, the mean sojourn time of tasks decreases. However, when the number of VMs hosted on each server is greater, the system costs do not decrease but rather increase gradually. The reason is that the energy consumption also affects the system costs to a certain degree. Consequently, there is minimum system cost when the parameters d are set optimistically. According to the results shown in Figures 4(a) and 4(b), the optimal parameters d can be given as shown in Table 3.

7. Conclusions

In this paper, we propose an energy-efficient sleep mechanism in a virtualized cloud data center with homogeneous servers. Each server hosts d VMs, and when the number of idle VMs reaches the threshold d , the server with the maximum number of idle VMs will be switched to sleep mode after migrating all the running tasks to other servers. The advantage of the proposed mechanism is that it can dynamically switch servers to sleep mode according to the workload. To analyse the proposed mechanism, we construct a two-dimensional continuous-time Markov chain to estimate the system's performance and energy consumption. The experimental results are valuable for studying the power-performance trade-off in cloud data centers. By considering the influence of different performance metrics, a cost function for the cloud data center is constructed. We obtain the optimal number of VMs hosted on one server by the cost function.

VM migration technology is very important in virtualized cloud data centers. In the future, we plan to study the energy-efficient task scheduling mechanism with VM migration in virtualized cloud data centers.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by National Natural Science Foundation (no. 61472342), China, and the Strategic Priority Research Program of the Chinese Academy of Sciences (no. XDA06010401).

References

- [1] L. Gergo, N. Florian, and H. de Meer, "Performance tradeoffs of energy aware virtual machine consolidation," *Cluster Computing*, vol. 16, pp. 37–38, 2013.
- [2] J. Shuja, K. Bilal, S. A. Madani et al., "Survey of techniques and architectures for designing energy-efficient data centers," *IEEE Systems Journal*, vol. 10, no. 2, pp. 507–519, 2016.
- [3] S. K. Uzaman, A. U. R. Khan, J. Shuja, T. Maqsood, F. Rehman, and S. Mustafa, "A systems overview of commercial data centers: initial energy and cost analysis," *International Journal of Information Technology and Web Engineering*, vol. 14, no. 1, pp. 42–65, 2019.
- [4] L. Richard and Villars, *Worldwide Data Center Census and Construction 2014–2018 Forecast: Aging Enterprise Data Centers and the Accelerating Service Provider Buildout*, International Data Corporation (IDC), Framingham, MA, USA, 2014.
- [5] S.-Y. Jing, S. Ali, K. She, and Y. Zhong, "State-of-the-art research study for green cloud computing," *The Journal of Supercomputing*, vol. 65, no. 1, pp. 445–468, 2013.
- [6] R. Danilak, *Why Energy is a Big and Rapidly Growing Problem for Data Centers*, Forbes Media LLC, Forbes Rep., New York, NY, USA, 2017.
- [7] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.
- [8] J. Shuja, R. W. Ahmad, and A. Gani, "Greening emerging IT technologies: techniques and practices," *Journal of Internet Services and Applications*, vol. 8, no. 1, pp. 1–11, 2017.
- [9] A. Hameed, A. Khoshkbarforoushha, R. Ranjan et al., "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, vol. 98, no. 7, pp. 751–774, 2016.
- [10] V. K. M. Raj and R. Shriram, "Power management in virtualized data center—a survey," *Journal of Network and Computer Applications*, vol. 69, pp. 117–133, 2016.
- [11] K. Li, "Improving multicore server performance and reducing energy consumption by workload dependent dynamic power management," *IEEE Transactions on Cloud Computing*, vol. 4, no. 2, pp. 122–137, 2016.
- [12] L. Wang, G. V. Laszewski, J. Dayal, and F. Wang, "Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS," in *Proceeding of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 68–377, Melbourne, Australia, May 2010.
- [13] J. Kang and S. Ranka, "Dynamic slack allocation algorithms for energy minimization on parallel machines," *Journal of Parallel and Distributed Computing*, vol. 70, no. 5, pp. 417–430, 2010.
- [14] J. A. Kumar and S. Vasudevan, "Verifying dynamic power management Schemes using statistical model checking," in *Proceedings of the 17th Asia and South Pacific Design Automation Conference*, pp. 579–584, Bangalore, India, January 2012.
- [15] Y. Chunxia and J. Shunfu, "An energy-saving strategy based on multi-server vacation queuing theory in cloud data center,"

- The Journal of Supercomputing*, vol. 74, no. 12, pp. 6766–6784, 2018.
- [16] C. L. Cheng, L. Jun, and W. Ying, “An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing,” *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 28–39, 2015.
 - [17] T. Phung-Duc and K. Kawanishi, “Impacts of retrials on power-saving policy in data centers,” in *Proceedings of the 11th International Conference on Queueing Theory and Network Applications*, vol. 22, Wellington, New Zealand, December 2016.
 - [18] D. Meisner and T. F. Wenisch, “DreamWeaver: architectural support for deep sleep,” in *Proceeding of the 7th International Conference on Architectural Support for Programming Languages & Operating Systems*, pp. 313–324, London, UK, March 2012.
 - [19] S. Vakiliinia, M. M. Ali, and D. Qiu, “Modeling of the resource allocation in cloud computing centers,” *Computer Networks*, vol. 91, no. 14, pp. 453–470, 2015.
 - [20] T. Sakata and S. Kasahara, “Multi-server queue with job service time depending on a background process,” in *Queueing Theory and Network Applications*, pp. 163–171, Springer, Berlin, Germany, 2016.
 - [21] B. M. Nguyen, D. Tran, and G. Nguyen, “Enhancing service capability with multiple finite capacity server queues in cloud data centers,” *Cluster Computing*, vol. 19, no. 4, pp. 1747–1767, 2016.
 - [22] W. Wang, J. Z. Luo, and A. B. Song, “Dynamic pricing based energy cost optimization in data center environments,” *Chinese Journal of Computers*, vol. 36, no. 3, pp. 599–612, 2013.
 - [23] D. Liao, K. Li, G. Sun, V. Anand, Y. Gong, and Z. Tan, “Energy and performance management in large data centers: a queuing theory perspective,” in *Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC)*, pp. 287–291, Anaheim, CA, USA, February 2015.
 - [24] C. L. Cheng, Y. Wang, and D. Y. Zhang, “Strategy of servers awakening based on dynamic threshold in cloud computing,” *Systems Engineering and Electronics*, vol. 37, no. 6, pp. 1437–1445, 2015.
 - [25] D. Shen, J. Luo, F. Dong et al., “Stochastic modeling of dynamic right-sizing for energy-efficiency in cloud data centers,” *Future Generation Computer Systems*, vol. 48, pp. 82–95, 2015.
 - [26] D. Deng, K. He, and Y. Chen, “Dynamic virtual machine consolidation for improving energy efficiency in cloud data centers,” in *Proceedings of International Conference on Cloud Computing & Intelligence Systems*, pp. 366–370, Beijing, China, August 2016.
 - [27] D. Klusáček, V. Chlumský, and H. Rudová, “Planning and optimization in TORQUE resource manager,” in *Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing*, pp. 203–206, Portland, OR, USA, June 2015.
 - [28] Y. Georgiou, T. Cadeau, D. Glesser, D. Auble, M. Jette, and M. Hautreux, “Energy accounting and control with SLURM resource and job management system,” in *Proceedings of the 12th International Conference on Distributed Computing and Networking*, pp. 96–118, Coimbatore, India, 2014.
 - [29] B. R. Ray and S. Chowdhury, “Reverse engineering technique (RET) to predict resource allocation in a Google cloud system,” in *Proceedings of the 8th International Conference on Cloud Computing*, pp. 688–693, Funchal, Portugal, March 2018.
 - [30] D. Bernstein, “Cloud foundry aims to become the OpenStack of PaaS,” *IEEE Cloud Computing*, vol. 1, no. 2, pp. 57–60, 2014.
 - [31] S. H. Choi, B. Kim, and K. Sohrawy, “On matrix-geometric solution of nested QBD chains,” *Queueing Systems*, vol. 43, no. 1-2, pp. 5–28, 2003.
 - [32] G. Latouche, V. Ramaswami, and V. G. Kulkarni, “Introduction to matrix analytic methods in stochastic modeling,” *Society for Industrial and Applied Mathematics*, vol. 12, no. 4, pp. 435–436, 1999.
 - [33] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.