

STOCHASTIC MODELING AND PERFORMANCE ANALYSIS OF ENERGY-AWARE CLOUD DATA CENTER BASED ON DYNAMIC SCALABLE STOCHASTIC PETRI NET

Hua HE

*School of Mathematics and Statistics, Shandong University of Technology
266 Xincunxi Road, Zhangdian District, 255000 Zibo City, Shandong, China
e-mail: huahe@sdut.edu.cn*

Yu ZHAO

*Institute of Rural Development, Shandong Academy of Social Sciences
56 Shungeng Road, 250002 Jinan, China
e-mail: yuzhaosdass@foxmail.com*

Shanchen PANG

*College of Computer Science and Technology, China University of Petroleum
66 Changjiangxi Road, Huangdao District, 266580 Qingdao City, Shandong, China
e-mail: shanchenpang@sohu.com*

Abstract. The characteristics of cloud computing, such as large-scale, dynamics, heterogeneity and diversity, present a range of challenges for the study on modeling and performance evaluation on cloud data centers. Performance evaluation not only finds out an appropriate trade-off between cost-benefit and quality of service (QoS) based on service level agreement (SLA), but also investigates the influence of virtualization technology. In this paper, we propose an Energy-Aware Optimization (EAO) algorithm with considering energy consumption, resource diversity and virtual machine migration. In addition, we construct a stochastic model for Energy-Aware Migration-Enabled Cloud (EAMEC) data centers by introducing Dynamic Scalable Stochastic Petri Net (DSSPN). Several performance parameters are defined

to evaluate task backlogs, throughput, reject rate, utilization, and energy consumption under different runtime and machines. Finally, we use a tool called SPNP to simulate analytical solutions of these parameters. The analysis results show that DSSPN is applicable to model and evaluate complex cloud systems, and can help to optimize the performance of EAMEC data centers.

Keywords: Stochastic Petri net, QoS, energy efficiency, performance evaluation, cloud computing

Mathematics Subject Classification 2010: 68M20

1 INTRODUCTION

Cloud computing can provide a convenient access to shared configurable resources (e.g. servers, storage, network, applications and services) to consumers by cloud providers directly deploying geographically distributed cloud data centers around the world [1]. As the important underlying infrastructure of cloud computing, the scale of data centers becomes larger, and has received increasing attention in the improvement of both performance and quality of service (QoS) requirements. But the research on energy consumption is still insufficient [2].

Statistically, the electricity energy consumption of data centers is estimated up to 40% of total U.S. energy consumption, and the energy cost is accounted for 42% of the total operating expense of data centers [3]. Hence, the improvements of energy efficiency are crucially important for cloud data centers. Cloud providers need to insure that their profits and return on investment are not rapidly falling owing to increased energy costs, while satisfying the QoS requirement of consumers based on service level agreement (SLA). In addition, improving energy efficiency can reduce resource consumption, release negative effects of environmental pollution, and achieve sustainable development in cloud data centers. Nevertheless, there still remain a range of challenges in realizing, modeling and performance evaluation resources scheduling of cloud data centers with the energy efficient way [2].

Firstly, the physical resources (e.g. PMs) in underlying infrastructure of cloud data centers are heterogeneous. It means that service capacities and energy consumptions can vary with the resource types. Secondly, based on virtualization technology, cloud data centers can provide multiple virtual machine (VM) instances on fewer PMs for multiple consumers simultaneously. Although energy consumption can be reduced by switching idle PMs off or to a low-performance levels (e.g. using DVFS), the performance may be significantly degraded when multiple VMs are running on the same PM in cloud data centers [4]. In other words, the key is to find an appropriate trade-off between energy efficiency and QoS guarantee. Moreover, VMs can dynamically migrate from one PM to another, which will help to improve resource utilization, realize load balancing, and decrease failure rate by avoiding hot

spots. Finally, but one of the most important aspects, the properties of cloud data centers (such as large-scale, dynamics, heterogeneity and diversity) make the system performance evaluation becoming more and more complicated. But for now, little attention has been the focus on how to provide an intuitive model description and effective analysis method for cloud data centers.

Stochastic Petri Net (SPN) is a graphic modeling and analysis tool for distributed systems [5]. DSSPN is an extension of SPN, which introduces enabling predicates and random switches to describe the firing conditions of immediate transitions. Compared with SPN, DSSPN can provide rich semantics to depict the scheduling process by allowing tokens and the labels on arcs to be expressed by a tuple $\langle R_k, PM_j \rangle$. In this paper, we can better model and evaluate some important parameters of cloud system by introducing DSSPN. Moreover, system bottlenecks can be well detected through observation and analysis token backlogs in places.

The state explosion is the main difficulty of Petri nets. DSSPN not only can effectively reduce the scale of state space by merging transitions and places with equivalent transformation, but the refined technology can dynamically adjust model based on enabling predicates and random switches according to run-time states of the system. That is, immediate transitions can be disabled/enabled by setting parameters of enabling predicates and random switches without the model reconstruction to realize scalability.

Based on the above discussions, this paper is dedicated to design and model resource scheduling for cloud data centers, which can realize energy efficiency and avoid the degradation of performance. The main contributions of this study are organized as follows:

1. We abstract a task scheduling and VMs allocation model of energy-aware migration-enabled cloud data centers (*EAMEC*).
2. In order to improve energy efficiency and ensure performance by avoiding hot spots in clusters, we put forward an Energy-Aware Optimization (*EAO*) algorithm.
3. Based on Dynamic Scalable Stochastic Petri Net (DSSPN), we establish the stochastic model of *EAMEC* [6]. Furthermore, we evaluate some performance parameters (such as task backlogs, throughput, utilization, and energy consumption) of *EAMEC* by adopting *EAO* algorithm.
4. To validate the proposed approach and algorithm, we conduct extensive experiments through simulations, and receive performance results under different resources or different runtime.

The rest of this paper is organized as follows: Section 2 briefly discusses the related literature. The system model formal description is discussed in Section 3. Section 4 constructs a stochastic model for *EAMEC* data centers based on DSSPN, and proposes an *EAO* algorithm to enhance resource utilization and impair energy

consumption. Section 5 elaborates the parameters used in the performance analysis, and the experimental setup used for simulations is demonstrated in Section 6. Finally, we make a conclusion and discuss the future research.

2 RELATED WORK

Performance analysis usually concentrates on interrelation of system configuration, system load and performance indicators, which has already attracted some attention in the industry and academia. We divide the methods of performance analysis into three categories: measurement method, simulation method and model method.

Applying some measuring instruments, or measurement and simulation approaches, or measuring procedures, can directly attain the performance indicators and closely related quantities of systems. Then, performance indexes could be figured out by the corresponding calculation. An extensible cloud simulator CloudSim was proposed, which could evaluate the overall performance and also the energy consumption with taking into account I/O workload in a data center [7]. Based on CloudSim, a cloud framework CloudSimNFV was introduced to simulate several scheduling algorithms for resource allocation, and energy consumption was further evaluated [8]. Performance measurement framework (PMF) in virtualized cloud was studied to quantify the performance of profit and loss, and the significance of optimization for the application deployment was exposed [9]. The VM consolidation algorithm, which can consolidate VMs to PMs based on input task, was proposed to decrease the energy consumption by reducing the amount of active PMs, and is evaluated in CloudSim simulator to verify its effectiveness [10]. Based on Dynamic Voltage Frequency Scaling (DVFS), a cloud service framework with several power aware VM provisioning schemes was demonstrated to model the request process of VMs for real-time applications within data centers [11].

Measurement and simulation approaches are the most direct and effective way for the performance evaluation. However, the two approaches only can be applied to subsistent running systems, and are extremely time-consuming. Moreover, none of the two approaches is suitable for large scale and complicated cloud systems, especially involving numerous parameters in dynamic environments [6]. Therefore, neither of the measurement and simulation approaches is capable of finding out performance bottlenecks.

To overcome these challenges, some researchers propose some model methods to analyze and to evaluate the system performance. Based on network, a stochastic queuing approach is introduced to analyze the performance of migration-enabled clouds in error-prone environment, and to evaluate the performance metrics with different load conditions [1]. The cloud center is modeled as a M/G/m/m+r queuing system with single task arrivals and the finite task buffer, and a transformed analytical model based on Markov chain is proposed to obtain the probability distribution of the response time, blocking probability, and number of tasks [12]. The complex cloud system is divided into multiple submodels, and the interactive continuous

time Markov chain (CTMC) is introduced to study some important performance parameters of cloud data centers, such as task blocking probability, total waiting time, and time delay of users' service requests [13]. Multi-layered graph models are proposed to analyze various data center network (DCN) topologies and to compare the classic robustness metrics under different failure scenarios. In addition, based on the percentage change in the graph structure, a new metric named deterioration is also presented to quantify the DCN robustness [14].

Compared with the above works, we mainly focus on the energy-aware strategies for migration-enabled cloud data centers, and firstly introduce Dynamic Scalable Petri Net (DSSPN) to model and evaluate some important performance parameters (e.g. task backlog, average throughput, average reject ratio, resource utilization, and so on) of the proposed cloud system under different runtime and various quantities of PMs.

3 SYSTEM MODEL

The Energy-Aware Migration-Enabled Cloud (*EAMEC*) is a kind of green clouds, which can provide virtual configurable services on data centers by integrating themselves into networks. In addition, it can also increase the electricity efficiency in buildings, which accounts for about 40% of the total energy consumption [15]. Figure 1 shows the process of task scheduling and VM provision in *EAMEC*.

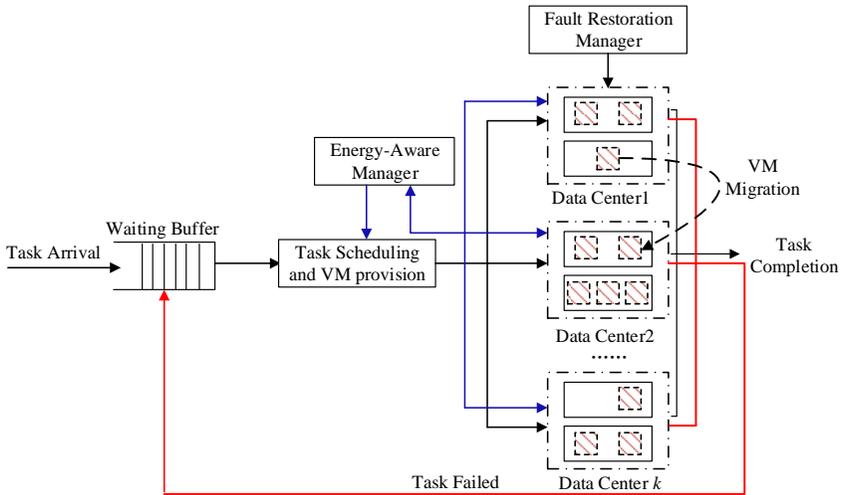


Figure 1. Task scheduling and VM provision in *EAMEC*

In this paper, the service model of the *EAMEC* system is the Infrastructure as a Service (IaaS). Without a loss of generality, we assume that the *EAMEC* system consists of k clusters (denoted as DC_i), each of which hosts np_i heterogeneous ma-

chines (i.e. PMs), such as high performance computers, workstations, and so on. Where, $k \in N^+$, $np_i \in N^+$, $i \in N^+$, $N^+ = \{1, 2, 3, \dots\}$. The the total number of PMs in the system is defined as follows:

$$tnp = \sum_{i=1}^k np_i. \quad (1)$$

Note that, in order to facilitate the analysis, it is assumed that these heterogeneous PMs (denoted as PM_j) have the same functionality, just for different capabilities in both CPU and memory. For example, a data center consists of two PMs, the CPU capability of each PM is 1000 MIPS, while another might be 800 MIPS. Where, $j \in \{1, 2, \dots, tnp\}$, MIPS is the unit of CPU.

Each PM_j can host at most mv instances of VM simultaneously, and the amount of VM instances concurrently running on PM_j is denoted as nv_j . The total number of VM instances running in the system is defined as follows:

$$tnv = \sum_{j=1}^{tnp} nv_j. \quad (2)$$

The set of VM in the system is expressed by $VM = \{VM_j^{nv_j} | VM_1^1, \dots, VM_1^{nv_1}, \dots, VM_{tnp}^1, \dots, VM_{tnp}^{nv_{tnp}}\}$, where $j \in \{1, 2, \dots, tnp\}$. Let $vpul_j^l$ indicate the processor utilization of l^{th} VM on machine PM_j , and pc_j expresses the processor capacity of machine PM_j . Where, $vpul_j^l \geq 0$, and $0 \leq \sum_{l=1}^{nv_j} vpul_j^l \leq pc_j$. Then the processor utilization PPU_j of machine PM_j during a given period of time is:

$$PPU_j = \left(\sum_{l=1}^{nv_j} vpul_j^l \right) / pc_j, \quad 0 \leq PPU_j \leq 1. \quad (3)$$

For the problem formulation, the following constraints are taken into consideration:

1. Tasks submitted to waiting buffer by users are independent, and they need to be allocated across the pool of VMs. The arrival rate of new tasks is λ , and it obeys the exponent distribution.
2. The capacity of waiting buffer is C , in which the tasks are served on the "first come, first served" (FCFS) basis. The VM is allocated in slots, each of which has the same length and denoted as ΔT . Where, $C \in N^+$, and $\Delta T \geq 0$.
3. Machines in the same cluster are homogeneous, while machines in different clusters might be heterogeneous. Each machine can be turned on or off, or configured to operate at low-performance levels (e.g. using DVFS) independently.
4. Each VM can be dynamically started and stopped on a PM according to the incoming tasks' requirements, and may lead to failure due to breakdown at runtime. Then failed VM can be repaired by the normal function. Let γ indicate

the failure rate, and η express the repair rate. In addition, both of them obey exponent distribution. Where, $\gamma \geq 0$, $\eta \geq 0$.

5. Once virtual machine VM_j^l malfunctions, the task running on VM_j^l will be broken down, and then resubmitted to the waiting buffer. The ratio of resubmission is β ($\beta \geq 0$), which obeys exponent distribution.
6. The service rate of each VM_j^l is uniform, and is expressed by u_j^l . Note, that the capacity of VM_j^l is determined by PM. That is, if VM_j^l hosted on PM_j , and the service rate of PM_j is u_j , then $u_j^l = u_j$. Where, $l \in 1, 2, \dots, nv_j$, and $j \in 1, 2, \dots, tnp$.
7. Each PM applies Dynamic Voltage and Frequency Scaling (DVFS) to achieve an appropriate trade-off between energy efficiency and performance. All machines have two service levels with different service rates. For $\forall PM_j$, if $n(PM_j)/m_j < \delta$, the service level of PM_j is 1, that represents serving at normal mode. Otherwise the service level of PM_j is 2, with providing a lower service mode to save energy. Where, $n(PM_j)$ indicates the amount of VMs currently running on PM_j , δ expresses the control threshold, $0 < \delta < 1$ (see below).

In order to analyze properly, this paper only considers the energy consumed by processors. There are several reasons [1]:

1. In cloud data centers, the total energy consumption is determined by CPU, memory, disk storage and network components. Compared to other resources, the energy consumption of CPU is dominant in cloud data centers. Therefore, we focus on the energy consumed by CPU in this paper.
2. The energy consumption by machines can be accurately described by a linear relationship between the energy consumption and CPU utilization.
3. The main goal of this paper is to reveal how the energy-aware strategies influence the energy consumption in *EAMEC* data centers.

The studies have shown that the relationship between energy consumption and CPU utilization can be described by a linear function, even when DVFS is applied [16]. This is because that DVFS is only applied on CPU, which can adjust the voltage and frequency of CPU based on the number of states. In addition, these studies discover that the energy consumption of an idle PM is approximately 70% of the power consumed under a fully utilization [16]. Hence, the PM can be switched to the leisure mode for energy conservation. The linear function is defined as follows:

$$P(u) = k \times P_{wm} + (1 - k) \times P_{wm} \times u = P_{wm} \times (0.7 + 0.3u). \quad (4)$$

Where, P_{wm} indicates the energy consumption under normally working condition, $k = 70\%$, while u is the CPU utilization.

As mentioned above, the energy consumption at peak rate is much higher than other rates. To reduce the energy consumption, VMs hosting on the PMs at the peak rate can be migrated to those PMs with other service rates. The VM migration

can avoid hot spots, and implement load balancing of cloud data centers. Moreover, it can also improve the resource utilization, and decrease the failure rate caused by machine errors.

4 STOCHASTIC MODELING BASED ON DYNAMIC SCALABLE STOCHASTIC PETRI NET

As discussed above, we introduce Dynamic Scalable Stochastic Petri Net (DSSPN) to model the process of task scheduling in *EAMEC* data centers. DSSPN is an expanded formation of Stochastic Petri Net (SPN) with similar firing rules and dynamics. For the limitation of space, we will not work it here in detail. Figure 2 shows a DSSPN model with abstract subnets of the task scheduling in an *EAMEC* data center. Figure 3 further describes the DSSPN model with detailed flow of each subnet. It should be noted that the scheduling or decision is expressed by the enabling predicates and random switches associated with the transitions.

4.1 DSSPN Model of EAMEC

The DSSPN model of the task scheduling in Energy-Aware Migration-Enabled Cloud is defined as (Figures 2 and 3): $EAMC = (P, T, F, K, W, \lambda, TS, G, E, f, g, M_0)$ where P is the set of places, T is the set of transitions, consisting of immediate transitions and timed transitions. F expresses the set of arcs, K indicates the set of capacities combined with places, W is the set of weights, λ is the set of average fired rates mapping to timed transitions. TS denotes the set of types, G is a function that maps places or transitions to types. E is a function to set values for types, f and g are enabling predicates and random switches associated with transitions, respectively. M_0 represents the initial marking which models the initial status of a system.

The elements in the DSSPN model *EAMEC* are defined as follows:

1. $P = \{p_{wq}\} \cup \{p_{mj}, s_{wmj}, s_{smj}, q_{wmj}, q_{smj}, s_{errj}, s_{resj}\}$. Where, P is a finite set, and $j \in \{1, 2, \dots, tnp\}$.
2. $T_I = \{t_{mj}, t_{sj}, t_{wtsj}, t_{stwj}, t_{ij}, t_{errj}, t'_{errj}\}$. Where, T_I is the set of immediate transitions, $i, j \in \{1, 2, \dots, tnp\}$, and $i \neq j$.
3. $T_T = \{t_c, s_{1j}, s_{2j}, t_{repj}, t_{resj}\}$. Where, T_T is the set of timed transitions, and $j \in \{1, 2, \dots, tnp\}$.
4. The elements of finite set F , G and E signify arcs and labels on arcs, respectively (Figures 2 and 3). The detailed descriptions of K , λ , G , E , f , g and M_0 will be explained later.
5. $TS = \{R_k, PM_j, \langle R_k, PM_j \rangle\}$, where $k \in N^+$, and $j \in \{1, 2, \dots, tnp\}$.

The definitions of places and transitions included in *EAMEC* are described as follows:

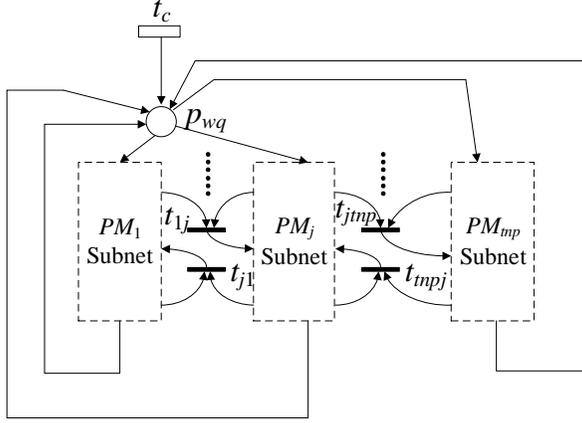


Figure 2. The DSSPN model of task scheduling in *EAMEC*

- (a) t_c : a timed transition with exponent distribution. It expresses assignment requests of VMs submitted by users, and fires with rate of λ . Once t_c fired, there is a task entering the place p_{wq} . Its enabling predicate is $f(t_c)$:

$$f(t_c) : M(p_{wq}) < C. \quad (5)$$

- (b) p_{wq} : a place expresses the waiting buffer, and is used to store task/VM provision requests. The capacity of p_{wq} is C , i.e., $p_{wq} = C$. For example, if $G(p_{wq}) = R_k$, then the type of each token in place p_{wq} is R_k , and $E(p_{wq}) \in N^+$ (indicates the size of tasks in the unit MB). That is, the attribute of tokens in a place are a tuple $\langle R_k, E(p_{wq}) \rangle$. Where, $k \in N^+$, and $j \in \{1, 2, \dots, tnp\}$.
- (c) p_{mj} : a place corresponds to machine PM_j . The tokens in p_{mj} indicates the current amount of available VMs on p_{mj} . The initial marking is $M_0(p_{mj}) = nv_j$. In addition, $K(p_{mj}) = nv_j$, $G(p_{mj}) = PM_j$, and $E(p_{mj}) \in [1, nv_j]$.
- (d) t_{wj} , t_{sj} : immediate transitions which are combined with place p_{mj} , p_{wq} , s_{wmj} , and s_{smj} to express allocation strategies of VMs for task requests in waiting buffer. Their enabling predicates and random switches will be described in the next section.
- (e) s_{wmj} , s_{smj} : state places, representing the working states of machine PM_j . Where, s_{wmj} indicates PM_j working with normal state, while s_{smj} denotes PM_j working with leisure state. The machine provides different service rates under different states. All machines can provide different service rates under different states.
- (f) t_{wtsj} , t_{stwj} : immediate transitions. Transition t_{wtsj} indicates that machine PM_j switches from normal state to leisure state, while t_{stwj} has the opposite

- (h) s_{1j}, s_{2j} : timed transition which expresses machine PM_j providing services with normal state or leisure state. The corresponding rate is exponent distribution, and $\lambda_{1j} = \mu_j, \lambda_{2j} = \mu_{sj}$, respectively. Where, μ_j denotes the service rate of PM_j with normal state, while μ_{sj} is the service rate with leisure state.
- (i) s_{errj}, s_{resj} : places are used to buffer the failed VMs and resubmitted tasks due to machinery breakdown, respectively. Where, $K(s_{errj}) = n_j, n_j \in N^+$.
- (j) t_{errj}, t'_{errj} : immediate transitions used to estimate whether faults of VMs occur. The failure rate is γ , the enabling predicates and random switches are:

$$g_{errj} = g'_{errj} = \begin{cases} \gamma, & \text{if } (M(S_{errj}) < K(S_{errj})) \wedge (M(s'_{errj}) < K(s'_{errj})), \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

- (k) t_{repj} : timed transition indicates the resubmission of failed tasks, the resubmitted rate obeys exponent distribution, and $\lambda_{repj} = \beta$. Its enabling predicate is:

$$f(t_{resj}) : M(p_{wq}) < C. \quad (9)$$

4.2 VM Scheduling Algorithm

In cloud data centers, VMs are allowed to migrate from one PM to another, and can be completed in a very short time without suspending the services. However, VM dynamic migration will influence the performance of applications running on the VM. Moreover, the performance degradation and downtime during the migration process depend on behaviors of the application running on migrated VM, i.e., the capacity of memory is occupied by the application during execution [16]. The time depends on the capacity of memory occupied by the migrated VM and available network bandwidth. Since the overall objective of this paper is the energy consumption in cloud data centers, we only consider the energy consumed by migration. The energy consumption generated by machines in the leisure state is much less than that in the normal state. So only those VMs running on machines in normal state will be migrated, and the destination nodes are chosen from the machines in the leisure state. In this paper, we also propose an Energy-Aware Optimization (EAO) algorithm to allocate and migrate VMs in cloud data centers, shown in Algorithm 1.

The enabling predicates and random switches of transitions (Figures 2 and 3) are described as follows:

Algorithm 1 Energy-Aware Optimization (EAO) Algorithm

-
- 1: Input: the task requests in waiting buffer, the set of machines PM , the set of maximal VMs concurrently running on PMs nv , the set of available virtual machines VM , switching threshold δ , the set of working states S , the set of energy consumption in normal state $ER(PM(w))$, the set of energy consumption in leisure state $ER(PM(s))$, bandwidth B , and threshold δ ;
 - 2: **for** $j = 1$ to $|PM|$ **do**
 - 3: $count_j \leftarrow$ the number of VMs running on PM_j ;
 - 4: **if** $(\sum_{j=1}^{|PM|} (nv_j - count_j) \leq$ the number of task requests in the waiting buffer)
 - 5: no VM need to be migrated;
 - 6: **else**
 - 7: calculate the energy consumption EC_j of VM_j^l running on PM_j ;
 - 8: **for** $i = 1$ to $|PM|$ **do**
 - 9: **if** $(i \neq j)$
 - 10: **if** $((count_j + 1)/nv_j) < \delta)$
 - 11: calculate the energy consumption EC_i of VM_i^l running on PM_i ;
 - 12: **if** $((TEC_j^l + EC_i) < EC_j)$
 - 13: migrate VM_i^l to PM_i ;
 - 14: **if** $(count_j < nv_j)$
 - 15: **if** $((count_j/nv_j) < \delta) \wedge (((count_j + 1)/nv_j)$
 - 16: choose an available VM from PM_j to allocate to the first task request in the waiting buffer;
 - 17: **if** there is no match VM for above description
 - 18: **for** $j = 1$ to $|PM|$ **do**
 - 19: **if** $(count_j < nv_j) \wedge ((count_j/nv_j) < \delta)$
 - 20: choose an available VM from PM_j to allocate to the first task request in the waiting buffer;
 - 21: **else**
 - 22: randomly choose an available VM to allocate to the first task request in the waiting buffer;
-

1. The enabling predicate and random switch of t_{wj} are respectively:

$$f(t_{wj}) : (M(s_{wmj}) = 1) \wedge \left(\frac{M(q_{wmj}) + 1}{nv_j} < \theta \right) \wedge \left(\sum_{j=1}^{tnp} M(s_{smj}) = 0 \right) \\ \wedge ((M(q_{wmj}) + M(q_{smj})) < nv_j), \quad (10)$$

$$g_{wj}(M) = \begin{cases} 1/|WEA(M)|, & j \in WEA(M), \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

$$WEA(M) = \left\{ j \mid ((M(q_{wmj}) + 1)/nv_j) = \underline{\min} \text{wec} \right\}. \quad (12)$$

Where, θ is the upper threshold of machines, and \min_wec in $WEA(M)$ is that $\min \{(M(q_{wmj}) + 1)/nv_j\}$, $j \in \{1, 2, \dots, tnp\}$, $|WEA(M)|$ indicates the amount of elements in $WEA(M)$. When the ratio of virtual machine on a machine is equal to or larger than θ , it is not allowed to create new VM instance on this machine.

2. The enabling predicate and random switch of t_{sj} are respectively:

$$f(t_{sj}) : \left(\left(\frac{M(q_{smj}) + 1}{nv_j} < \delta \right) \vee \left(\frac{M(q_{smj})}{nv_j} < \delta \right) \right) \wedge (M(s_{smj}) = 1),$$

$$\wedge ((M(q_{wmj}) + M(q_{smj})) < nv_j), \quad (13)$$

$$g_{sj}(M) = \begin{cases} 1/|SEA(M)|, & j \in SEA(M), \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

$$SEA(M) = \left\{ j \mid ((M(q_{smj}) + 1)/nv_j) = \min_sec \right\}. \quad (15)$$

Where, δ represents the threshold of state transition, and \min_sec in $SEA(M)$ is $\min \{(M(q_{smj}) + 1)/nv_j\}$. $|SEA(M)|$ expresses the number of elements in $SEA(M)$.

3. t_{ij} combines with places q_{wmj} , q_{smj} and p_{mj} to describe the process of task scheduling and VM allocation, its enabling predicate and random switch are:

$$f(t_{ij}) : \left(M(p_{wq}) < \sum_{j=1}^{tnp} M(p_{mj}) \right) \wedge (M(s_{smj}) = 1)$$

$$\wedge (M(q_{wmi})/nv_i > \delta) \wedge \left(\frac{M(q_{wmj}) + M(q_{smj}) + 1}{nv_j} < \delta \right)$$

$$\wedge \left(\left(\frac{M(q_{wmi}) \cdot x}{B} + \frac{M(q_{wmi}) \cdot x}{pc_j} \right) < \frac{M(q_{wmi}) \cdot x}{pc_i} \right), \quad (16)$$

$$g_{ij}(M) = \begin{cases} \frac{1}{|MDP(M)|}, & \text{if } j \in MDP(M), i \neq j, \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

$$MDP(M) = \left\{ j \mid \left(\frac{M(q_{wmj}) + M(q_{smj}) + 1}{nv_j} < \delta \right) \wedge (M(s_{smj}) = 1) \right.$$

$$\left. \wedge \left(\left(\frac{M(q_{wmi}) \cdot x}{B} + \frac{M(q_{wmi}) \cdot x}{pc_j} \right) < \frac{M(q_{wmi}) \cdot x}{pc_i} \right) \right\}. \quad (18)$$

Where, $i \neq j$, and $i, j \in \{1, 2, \dots, tnp\}$. $|MDP(M)|$ expresses the amount of elements in $MDP(M)$.

5 PERFORMANCE ANALYSIS

As described above, we will further analyze the performance and energy consumption of *EAMEC* based on DSSPN model and the probabilities of stability. Compared with other approaches, such as Markov decision process, DSSPN can employ the integrated functions of Stochastic Petri Net Package (SPNP) to automatically deduce the probabilities of stability, without having to calculate them by stochastic math formulas. This is beneficial to model and evaluate the performance the cloud systems, because the number of states might reach thousands even if only consisting of few machines, shown in Table 1.

| | 1 Machine | 2 Machines | 3 Machines | 4 Machines |
|-------------------|-----------|------------|------------|------------|
| Reachable states | 283 | 569 | 1 088 | 1 594 |
| Fired transitions | 923 | 1 977 | 3 928 | 5 842 |

Table 1. Number of states and fired transitions

- At time t , the amount of running VMs on PM_j is:

$$nvw_j(t) = M(q_{wm_j}(t)). \quad (19)$$

- At time t , the amount of idle VMs on PM_j is:

$$nsv_j(t) = M(p_{m_j}(t)). \quad (20)$$

- At time t , the average queue length of system $AQL(t)$ is:

$$AQL(t) = \left(\sum_{y=0}^t M(p_{wq}(y)) \right) / t. \quad (21)$$

- At time t , the average throughput of system $ATP(t)$ is:

$$ATP(t) = \left(\sum_{y=0}^t \sum_{j=1}^{tnp} nvw_j(y) \right) / t. \quad (22)$$

When the service transitions corresponding to machines are saturated, we can use the accumulating character of tokens in place p_{wq} to analyze the throughput of the *EAMEC* systems. The average throughput is related to the capacity of waiting buffer, service rates of machines and the maximal number of available VMs.

- At time t , the average probability that machine PM_j works with leisure state $ASR_j(t)$:

$$ASR_j(t) = \left(\sum_{y=0}^t P(M(q_{wm_j}(y)) \leq (\delta \cdot nv_j)) \right) / t \quad (23)$$

where $P(\text{enabled}(t_{err_j}(y)))$ is the enabling probability of error transition t_{err_j} at time t , and $P(M(q_{wm_j}(y)) \leq (\delta \cdot nv_j))$ represents the probability that tokens are in place q_{wm_j} is equal to or less than $\delta \cdot nv_j$.

- At time t , the utilization of machine PM_j is:

$$UR_j(t) = M(q_{wm_j}(t)) / nv_j. \quad (24)$$

- Based on the Equation (5), we can deduce the average energy consumption of the system at time t :

$$AEER(t) = \frac{\sum_{y=0}^t \sum_{j=1}^{tnp} (0.7 + 0.3 \cdot UR_j(y)) \cdot P_{wm_j}}{t \cdot tnp} \quad (25)$$

where P_{wm_j} is the energy consumption of PM_j in normal state, and the unit is watt.

6 CASE STUDY AND SIMULATION

In this section, we make simulated experiments to study the applicability of DSSPN in the framework of *EAMEC* data centers. We consider a sample of data center on laptop with Intel i5-4210 multi-core processors. In addition, SPNP platform is used to automatically deduce the analytical solutions of performance for *EAMEC* model, as shown in Figures 4, 5, 6, 7, 8, and 9. The machines are depicted by places in DSSPN, the analysis results are calculated based on simulated analytical solutions of SPNP and the tokens in places. The processor specification is given in [18], which is illustrated in Table 2.

| State Level | Normalized Service Rate | Energy Consumption |
|-------------|-------------------------|--------------------|
| 1 | 0.3333 | 0.279 |
| 2 | 0.5000 | 0.390 |
| 3 | 0.6666 | 0.570 |
| 4 | 1.0000 | 0.925 |

Table 2. Normalized specification for processors

The number of machines in the cloud data center varies from 1 to 4, the capacity of waiting buffer C is from 30 to 50, the threshold of switching transformation δ is 0.5. The arrival rate of task request is 20, the error rate is 0.2, the repair rate is 0.1, and the resubmitted rate of failed tasks is 0.3. In addition, we suppose that there are two classes of machines in the system. One can host 3 VMs at most with the state levels being 2 and 4. Another can host 2 VMs at most with the state levels being 1 and 3. For the convenient analysis, we assume that all rates obey the exponent distribution.

Figure 4 shows how the average queue length varies with the runtime t . When the system consists of 1 PM or 2 PMs, the arrival rate of tasks is larger than the

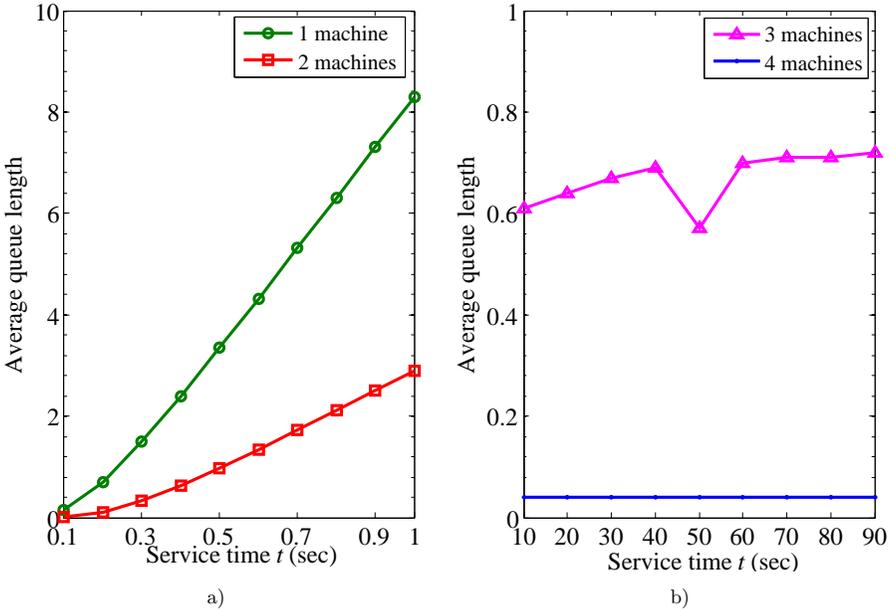


Figure 4. Simulation results of the average queue length

service rate of the system, so the queue length increases with the runtime t . And the queue length is gradually close to the capacity of the waiting buffer. When the system consists of 3 PMs or 4 PMs, and $0 \leq t \leq 1$, the tasks in the waiting buffer are less than the capacity of the system. At this point, there is no backlog of task requests in waiting buffer. In particular, Figure 4b) illustrates that 3 PMs can satisfy the requirement of the system hardly without backlogs.

Figure 5 illustrates how the number of PMs affects the average throughput of the system. When the capacity of waiting buffer is 30, the arrival rate is 20 tasks per second, $0 \leq t \leq 1$, and the number of machines is 1 or 2, the throughput of system is limited by the processing capacity of the system. When the number of PMs is 3 or 4, the processing capacity of the system can satisfy the requirements of users without waiting.

Figure 6 shows how the number of PMs affects the average reject rate of the system. When the task requests are up to 30, the system will reject new requests submitted by users. When $1 \leq t \leq 10$, one machine can meet user demands, the average reject rate is up to 56.65%, while the average reject rate is only 10.07% in 2 machines scenario. In addition, the average reject rate increases over time.

Figure 7 shows how the number of machines impacts the average failure rate. When $1 \leq t \leq 10$, the average failure rate is higher, but gradually decreases over time. The figure also illustrates that the more machines in the system, the lower the

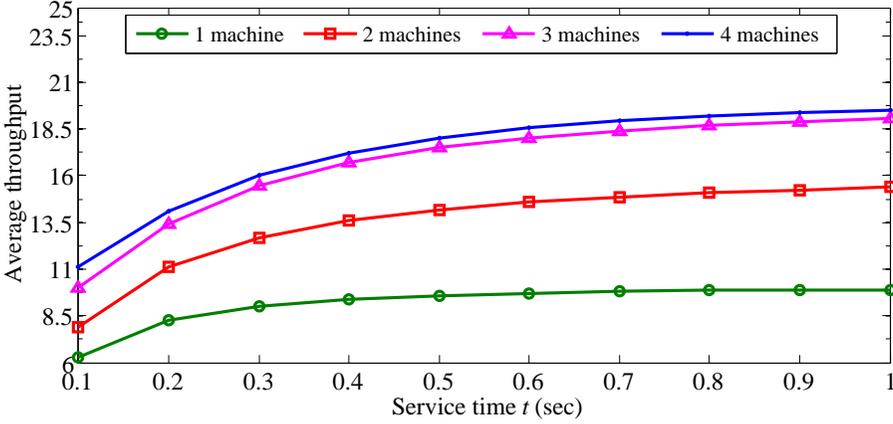


Figure 5. Simulation results of the average throughput

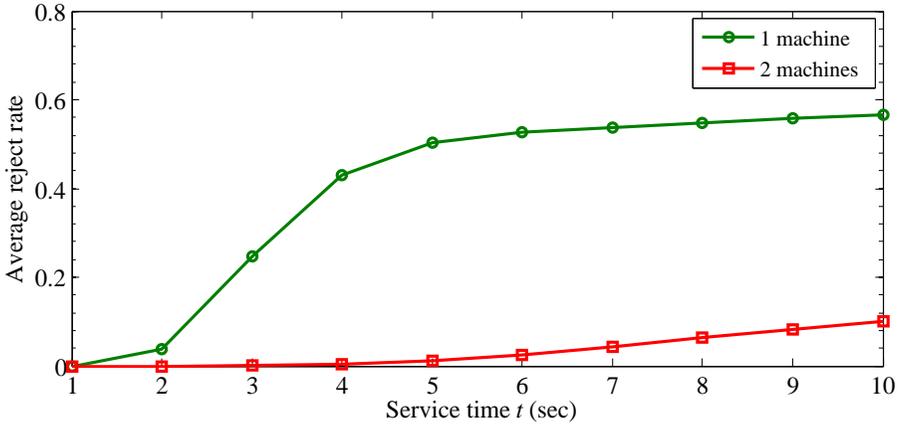


Figure 6. Simulation results of the average reject rate

average failure rate. This is due to the risk spread out by virtual machine dynamic migration.

Figure 8 shows how the number of PMs affects the average energy consumption of the system. When $0 \leq t \leq 1$, the average energy consumption is gradually close to a constant. The energy consumption increases with the number of machines included in the system. In 1 PM and 2 PMs scenarios, PMs provide services to peak. However, in 3 PMs and 4 PMs scenarios, the utilization of each machine is approaching a certain stability over time. In 3 PMs scenarios, machines serve in leisure state with the probability of 55.46% on average, while in 4 PMs scenarios, the probability is 60.49%.

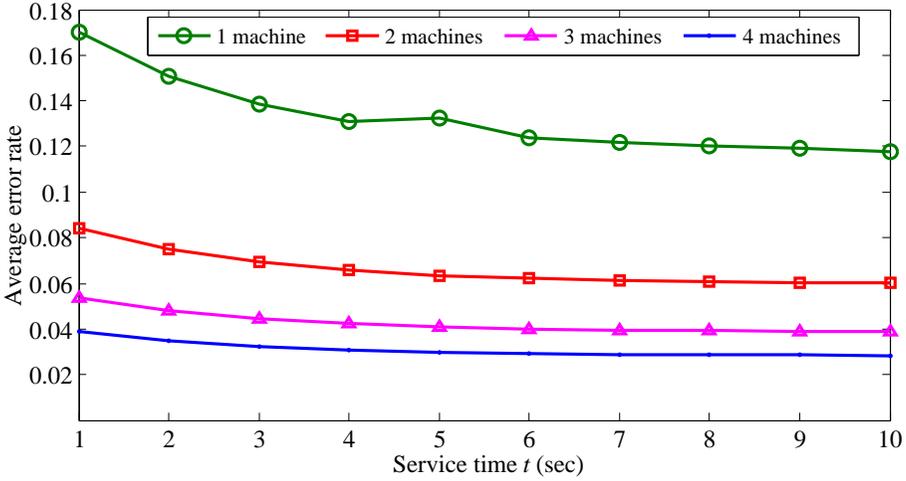


Figure 7. Simulation results of the average error rate

Figure 9 shows how the number of PMs affects the average resource utilization of the system. When $0 \leq t \leq 1$, the utilization in 1 PM scenario is the highest and up to 98.39%. In 2 PMs scenario, the utilization is about 48%, while the utilization is 69.04% and 78% for 3 PMs scenario and 4 PMs scenario, respectively. The reason for the reduced resource utilization of machines is that when the number

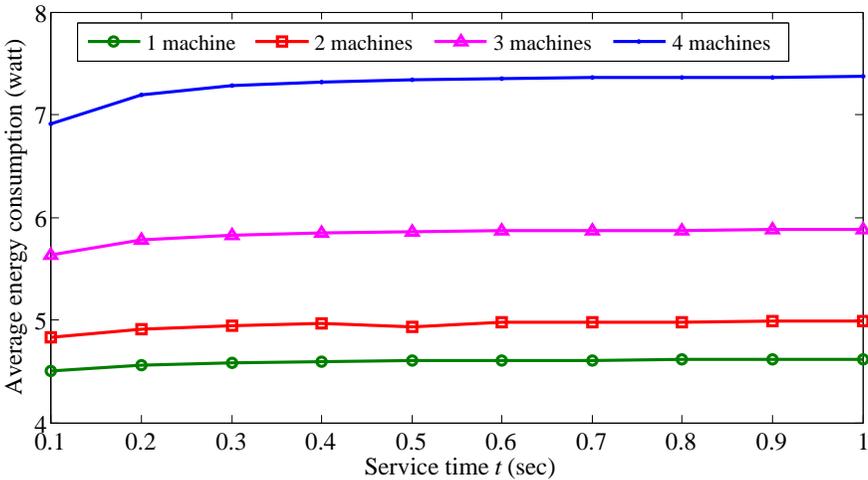


Figure 8. Simulation results of the average energy consumption

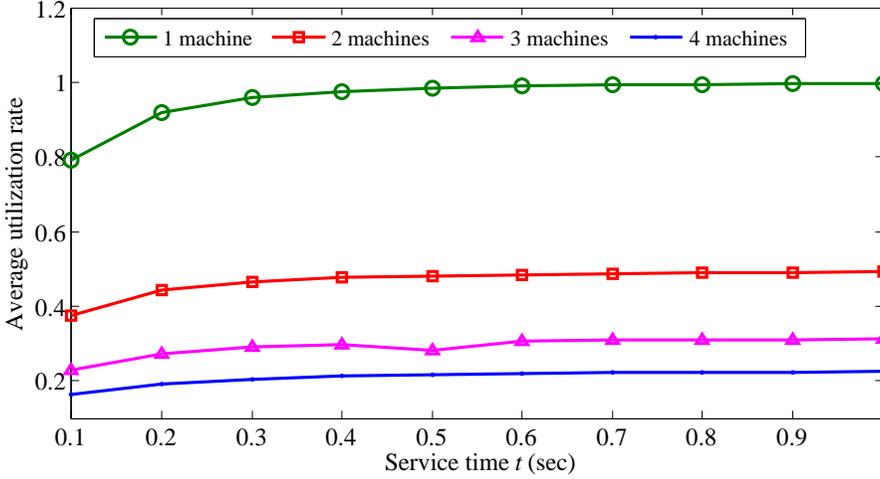


Figure 9. Simulation results of the average resource utilization

of machines increases the arrival rate of tasks and the capacity of waiting buffer remains unchanged.

In the simulation, the queue length increases with the runtime t , and it is gradually close to the capacity of the waiting buffer. When the system consists of 3 PMs or 4 PMs, and $0 \leq t \leq 1$, the tasks in the waiting buffer are less than the capacity of the system. At this point, there is no backlog of task requests in the waiting buffer. That is, available resources are greater than the task demands under the maximum capacity of the waiting buffer. So the resource utilization rate goes up when more machines are used.

To demonstrate the effectiveness of EAO, we compare the approach with VM balancing based on the SCORE tool [19]. The VM balancing approach splits VM requests over multiple cloud datacenters, which can avoid the performance degradation by resource contention. But VM balancing causes the large energy consumption owing to a large number of active servers [20]. The experiment parameters are shown in Table 3.

| | avgTaskPerJob | avgJobDuration | avgCpuPerTask | avgMemPerTask |
|----------------|---------------|----------------|---------------|---------------|
| Batch | 30 | 50 | 0.3 | 0.2 |
| Service | 9 | 500 | 0.5 | 0.7 |

Table 3. Experiment parameters

Figure 10 shows the performance between EAO and existing VM balancing, at numMachines = 100, cpuPerMachine = 4, and memPerMachine is 1. Figures 10 a) and 10 b) show the average utilization of CPU and memory between EAO and the VM balancing during runtime, respectively. The minimal memory utilization of VM

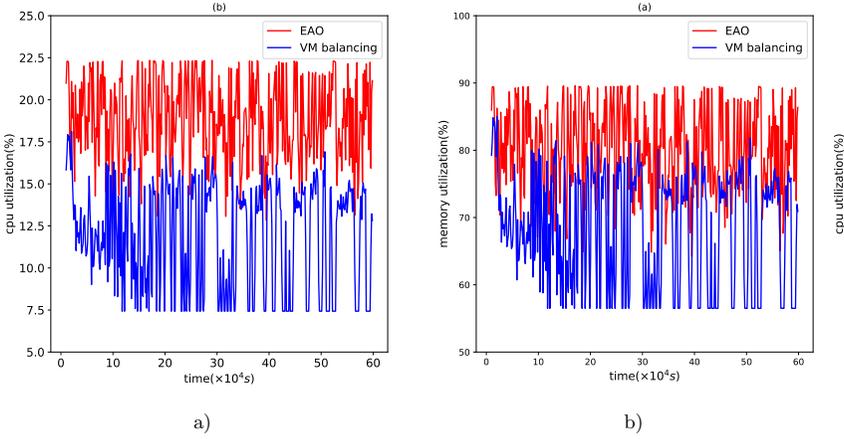


Figure 10. Experiment results between EAO and VM balancing

balancing is 49.49 %, but that of EAO is up to 64.20 %. Moreover, EAO has a 5.08 % higher maximum than VM balancing. The minimum of average CPU utilization for EAO is 12.82 %, but for VM balancing is 7.43 %. The maximum of average CPU utilization in EAO and VM balancing are 22.35 % and 18.12 %, respectively. Compared to VM balancing, the EAO approach achieves the improvements in resource utilization.

| Scheduling | Shut-Down Policy | Runtime [s] | Energy Consumed [kwh] | Energy Saved [kwh] | Number of Jobs |
|--------------|------------------|-------------|-----------------------|--------------------|----------------|
| EAO | always | 604 800 | 1 314 831 | 92.86 | 30 347 |
| VM balancing | never off | 604 800 | 1 470 019 | 0.00 | 27 918 |

Table 4. Energy-efficiency experiment

Except performance parameters, the energy-efficiency parameters are also included in the SCORE tool [19]. The results corresponding to Batch jobs between EAO and VM balancing are presented in Table 4. Note, that EAO approach further improves the performance of energy. Table 4 shows that the total energy consumption of EAO is approximately 10 % smaller than that of VM balancing. In addition, EAO completes more jobs than VM balancing at the same time. Our proposed approach takes into account both the energy consumption and resource utilization, while VM balancing only considers the performance of VM requests.

7 CONCLUSION

Based on DSSPN, this paper proposed a stochastic model to analyze the performance and energy consumption of *EAMEC* data centers, which can decrease systematic energy consumption by applying DVFS to CPUs. The *EAMEC* model depicts the logical relations among workload, failure and recovery of virtual machines, number of machines, VM dynamic migration and scheduling strategy. With the improving energy efficiency and increasing utilization, we proposed *EAO* algorithm to realize the dynamic migration of VMs. Then, we described the EAS strategy by setting enabling predicates and random switches of transitions in the model. Finally, a simulation tool called SPNP was introduced to work out the analytical solutions. The results could be used to analyze both the performance and energy consumption for cloud data centers. The simulation results also showed that DSSPN was convenient to model and evaluate complex cloud systems. Even if the states were up to thousands, we still obtained simulation results easily, without exhaustive complicated computations.

In the subsequent work, the authors plan to conduct their research on the model mechanism and formal semantics of dynamic Petri nets, and investigate the flexible modeling and dynamic optimization of service composition in large-scale mobile cloud system.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (grants No. 61702307, No. 61572523, and No. 61902222).

REFERENCES

- [1] XIA, Y. N.—ZHOU, M. C.—LUO, X.—PANG, S. C.—ZHU, Q. S.: A Stochastic Approach to Analysis of Energy-Aware DVS-Enabled Cloud Datacenters. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 45, 2014, No. 1, pp. 73–83, doi: 10.1109/TSMC.2014.2331022.
- [2] DINESH REDDY, V.—GANGADHARAN, G. R.—RAO, G. S. V. R. K.: Energy-Aware Virtual Machine Allocation and Selection in Cloud Data Centers. *Soft Computing*, Vol. 23, 2019, No. 6, pp. 1917–1932, doi: 10.1007/s00500-017-2905-z.
- [3] KLASS, A. B.—WILSON, E. J.: Energy Consumption Data: The Key to Improved Energy Efficiency. *San Diego Journal of Climate and Energy Law*, Vol. 6, 2015, pp. 69–115.
- [4] KRZYWDA, J.—ALI-ELDIN, A.—CARLSON, T. E.—ÖSTBERG, P.-O.—ELMROTH, E.: Power-Performance Tradeoffs in Data Center Servers: DVFS, CPU Pinning, Horizontal, and Vertical Scaling. *Future Generation Computer Systems*, Vol. 81, 2018, pp. 114–128, doi: 10.1016/j.future.2017.10.044.

- [5] MOLLOY, M. K.: Discrete Time Stochastic Petri Nets. *IEEE Transactions on Software Engineering*, Vol. SE-11, 1985, No. 4, pp. 417–423, doi: 10.1109/TSE.1985.232230.
- [6] HE, H.—PANG, S.—ZHAO, Z.: Dynamic Scalable Stochastic Petri Net: A Novel Model for Designing and Analysis of Resource Scheduling in Cloud Computing. *Scientific Programming*, Vol. 2016, 2016, Art.No. 9259248, 13 pp., doi: 10.1155/2016/9259248.
- [7] OUARNOUGHI, H.—BOUKHOBZA, J.—SINGHOF, F.—RUBINI, S.: Integrating I/Os in Cloudsim for Performance and Energy Estimation. *ACM SIGOPS Operating Systems Review*, Vol. 50, 2017, No. 2, pp. 27–36, doi: 10.1145/3041710.3041715.
- [8] YANG, W.—XU, M.—LI, G.—TIAN, W.: CloudSimNFV: Modeling and Simulation of Energy-Efficient NFV in Cloud Data Centers. *arXiv preprint arXiv:1509.05875*, 2015.
- [9] BAUTISTA, L.—ABRAN, A.—APRIL, A.: Design of a Performance Measurement Framework for Cloud Computing. *Journal of Software Engineering and Applications*, Vol. 5, 2012, No. 2, pp. 69–75, doi: 10.4236/jsea.2012.52011.
- [10] MISHRA, S. K.—PUTHAL, D. et al.: Energy-Efficient VM-Placement in Cloud Data Center. *Sustainable Computing: Informatics and Systems*, Vol. 20, 2018, pp. 48–55, doi: 10.1016/j.suscom.2018.01.002.
- [11] KIM, K. H.—BELOGLAZOV, A.—BUYYA, R.: Power-Aware Provisioning of Virtual Machines for Real-Time Cloud Services. *Concurrency and Computation: Practice and Experience*, Vol. 23, 2011, No. 13, pp. 1491–1505, doi: 10.1002/cpe.1712.
- [12] KHAZAEI, H.—MISIC, J.—MISIC, V. B.: Performance Analysis of Cloud Computing Centers Using M/G/m/m + r Queuing Systems. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, 2012, No. 5, pp. 936–943, doi: 10.1109/TPDS.2011.199.
- [13] KHAZAEI, K.—MISIC, J.—MISIC, V. B.: A Fine-Grained Performance Model of Cloud Computing Centers. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, 2012, No. 11, pp. 2138–2147, doi: 10.1109/TPDS.2012.280.
- [14] BILAL, K.—MANZANO, M.—KHAN, S. U.—CALLE, E.—LI, K.—ZOMAYA, A. Y.: On the Characterization of the Structural Robustness of Data Center Networks. *IEEE Transactions on Cloud Computing*, Vol. 1, 2013, No. 1, 14 pp., doi: 10.1109/TCC.2013.6.
- [15] Clean Energy Australia Report 2015. Melbourne, VIC, Australia, Clean Energy Council, 2015.
- [16] KUSIC, D.—KEPHART, J. O.—HANSON, J. E.—KANDASAMY, N.—JIANG, G.: Power and Performance Management of Virtualized Computing Environments via Lookahead Control. *Cluster Computing*, Vol. 12, 2009, No. 1, pp. 1–15, doi: 10.1007/s10586-008-0070-y.
- [17] ELBAY, S. K.—HEGAZY, I.—EL-HORBATY, E.-S. M.: Live Migration Overhead-Aware Dynamic VM Consolidation Algorithm in Cloud Computing. *Egyptian Computer Science Journal*, Vol. 42, 2018, No. 4, pp. 75–88.
- [18] CHANDNANI, L.—KAPOOR, H. K.: Formal Approach for DVS-Based Power Management for Multiple Server System in Presence of Server Failure and Repair.

IEEE Transactions on Industrial Informatics, Vol. 9, 2012, No. 1, pp. 502–513, doi: 10.1109/TII.2012.2198656.

- [19] FERNÁNDEZ-CERERO, D.—FERNÁNDEZ-MONTES, A.—JAKÓBIK, A.—KOŁODZIEJ, J.—TORO, M.: SCORE: Simulator for Cloud Optimization of Resources and Energy Consumption. *Simulation Modelling Practice and Theory*, Vol. 82, 2018, pp. 160–173, doi: 10.1016/j.simpat.2018.01.004.
- [20] PENG, Y.—KANG, D.K.—AL-HAZEMI, F.—YOUN, C.-H.: Energy and QoS Aware Resource Allocation for Heterogeneous Sustainable Cloud Datacenters. *Optical Switching and Networking*, Vol. 23, 2017, Part 3, pp. 225–240, doi: 10.1016/j.osn.2016.02.001.



Hua HE received her Ph.D. degree in computer application technology from Tianjin University, China, in 2017. Her research interests include Petri nets, big data processing, performance analysis and cloud computing. She is currently Lecturer at Shandong University of Technology, Zibo, China.



Yu ZHAO received his Ph.D. degree in Monte-Carlo modeling from the Brunel University London, Middlesex, UK, in 2014. His research interests include cloud computing, MapReduce. He is currently Assistant Research Fellow from the Shandong Academy of Social Sciences, Ji'nan, China.



Shanchen PANG received his Ph.D. degree in computer software and theory from the Tongji University, Shanghai, China, in 2008. His research interests include theory and the application of Petri nets, service computing, trusted computing. He is currently serving as Professor at the China University of Petroleum, Qingdao, China.