Future Generation Computer Systems 80 (2018) 139–156

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method

Zhihua Li [a,*], Chengyu Yan [a], Lei Yu [b], Xinrong Yu [a]

[a] Department of Computer Science and Technology, School of Internet of Things Engineering, Jiangnan University, Jiangsu Wuxi 214122, PR China
[b] School of Computer Science, College of Computing, Georgia Institute of Technology, Atlanta, GA, 30332, USA

ARTICLE INFO

Article history:
Received 8 December 2016
Received in revised form 29 August 2017
Accepted 30 September 2017
Available online 31 October 2017

Keywords:
Virtual machine consolidation
Multi-resource constraints
Optimization model
Artificial bee colony-like search
Energy aware

ABSTRACT

In a data center, virtual machine consolidation has been proposed to improve the resource utilization and energy efficiency. An effective and efficient virtual machine consolidation method should achieve an appropriate balance among multiple goals, including guaranteeing service quality, reducing energy consumption and maximizing resource utilization. This problem is a multi-objective optimization problem with multiple resource constraints. To solve this problem, we propose an energy-aware dynamic virtual machine consolidation (EC-VMC) method that migrates virtual machines while satisfying constraints on the probabilities of multiple types of resources being overloaded. In our method, a series of algorithms for selecting and placing virtual machines to be migrated are utilized, with constraints on the probabilities of various resources in a physical machine being overloaded. Our algorithms integrate and cooperate similarly to artificial bee colony foraging behavior to perform an optimized search for the mapping relation between virtual machines and physical machines for consolidation. Extensive simulation is conducted to compare our EC-VMC method with previous virtual machine consolidation methods. The simulation results demonstrate that the EC-VMC method effectively overcomes the deficiencies of some existing heuristic algorithms and is highly effective in reducing VM migrations and energy consumption of data centers and in improving QoS.

1. Introduction

High energy consumption is a major challenge for resource management in data centers. As a data center continues to expand, the problem of high energy consumption becomes more prominent [1]. A survey by International Business Machines (IBM) [2] suggests that the average CPU utilization of physical machines in a data center is only 15%∼20%, and a physical machine in an idle state typically consumes 70% of its peak energy consumption [2,3]. Idle physical machines (PMs) with underutilized resources in a data center indicate low energy efficiency and tremendous energy waste.

Virtualization technology enables a cloud-computing service provider to create multiple VMs in a single physical machine (PM) and perform load balancing via virtual machine (VM) migration. As a major technology for improving energy efficiency and resource utilization in a data center, virtual machine (VM) consolidation has been extensively investigated [4–10]. A data center can periodically consolidate VMs and turn off some underutilized PMs based on VM and PM resource utilization to reduce energy consumption and improve resource utilization. However, because of the stochastic variation of workload in data centers, overly radical VM consolidation will negatively influence resource reservations in PMs, which leads to Service Level Agreement (SLA) violations. Therefore, minimizing energy consumption and maximizing resource utilization while guaranteeing quality of service (QoS) is a major challenge for VM consolidation in data centers.

The main strategy of VM consolidation is to define the static overload threshold or upper bound of CPU utilization to identify the overload status of the physical machine and the lower bound of CPU utilization to identify the underutilized PMs. VM migration or VM consolidation is triggered based on these thresholds to achieve the goal of reducing energy consumption and improving QoS. Such methods [3,11] are simple but lack flexibility to adapt to the dynamic workload in a data center. A dynamic overload threshold [12] is proposed for VM migration that considers the workload variations on source and destination hosts after VM migrations, but it does not consider load rebalancing of data centers if the workloads on these hosts have changed. Some existing methods [7–10] only focus on VM migration but ignore the impact of VM migration on resource consumption and service quality. For instance, as a VM in live migration suspends its service, prolonged VM migration likely affects the service quality [13].

* Corresponding author.
E-mail address: zhli@jiangnan.edu.cn (Z. Li).

https://doi.org/10.1016/j.future.2017.09.075
0167-739X/

Usually, VM consolidation methods can be either single resource based [3–11] or multiple resource based [14,15]. Because the CPU resource is a key factor that influences energy consumption of data centers, numerous studies have investigated how to improve the energy efficiency via the allocation of CPU resources, such as the methods proposed in [3–11]. However, these methods only consider the CPU resource. Factors such as memory, bandwidth and disk utilization should also be considered for VM consolidation, since they are also key factors that affect QoS. However, at the same time, multi-resource based VM consolidation can be more complex. For instance, previous work [14] proposed a multi-resource VM consolidation method that involves more complex assumptions and mathematical calculations due to multi-resource consideration.

To address the limitations of existing VM consolidation methods, we consider the VM consolidation problem with the following constraints: (1) VM dynamic consolidation should trigger as few VM migrations as possible to minimize the negative impact on QoS; (2) in a data center, the VM-generated workload is complex and dynamic, and the VM consolidation strategy should minimize the probability of the physical machines being overloaded; (3) VM dynamic consolidation should turn off as many underloaded PMs as possible to reduce energy consumption of data centers; and (4) resources such as memory and network are the major factors that affect QoS, so VM consolidation should allow a comprehensive treatment of such multi-resources. Based on these considerations, we propose an energy-aware dynamic optimization approach, called EC-VMC, for VM consolidation with constraints over overload probabilities for multiple types of resources. It consists of multiple algorithms corresponding to different phases of VM dynamic consolidation. Our algorithms simulate artificial bee colony foraging behavior to find the mapping relation between PMs and VMs. By using the searching mechanism and optimization strategy of the artificial bee colony (ABC) algorithm [16], an approximate feasible solution is obtained via the iterative searches. Our major contributions, in detail, are as follows.

(1) First, through deeper research, we note that VM consolidation is a multi-objective optimization problem with multiple resource constraints;

(2) Then, by assuming the mapping relation between PMs and VMs as the food source, the proposed sub-algorithms integrate and cooperate to simulate the artificial bee colony foraging behaviors; by using both the searching mechanism and optimization strategy of the artificial bee colony ABC algorithm, the optimum mapping relation with multi-resource constraints between PMs and VMs is obtained;

(3) Next, the issue of "where from and where to" for live VM migration is globally optimized and examined. Results on achieving appropriate balance among guaranteeing service quality, reducing energy consumption and maximizing resource utilization are promising;

(4) At last, the proposed EC-VMC method overcomes the limitation of falling into local optima that some existing algorithms have, such as the well-known heuristic BFD (Best Fit Decreasing) algorithm. Validation and experimental comparison are conducted using the CloudSim platform. The experimental results indicate that the proposed EC-VMC method has a distinct advantage in terms of reducing energy consumption, VM migrations and improving QoS.

The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 identifies and describes our optimization objective with multi-resource constraints. Section 4 presents the overview and the detailed design of our VM dynamic consolidation. Section 5 presents the performance evaluation of our scheme compared with other VM consolidation methods. Finally, Section 6 concludes this paper and discusses our future work.

2. Related works

A VM consolidation scheme should identify the VMs that should be migrated and the PMs that can be turned off. It subsequently solves the issue of to where the VMs should be migrated, i.e., the issue of identifying the source and destination hosts for the VMs in live migration. In summary, "where from where to" is the core issue of VM migration or VM consolidation. Many works [3,12,14,15,17–31] examine this issue from different perspectives, such as VM placement [19–22], host overload detection [3,12,22–27], and VM migration selection [22,28,29]. In this section, we mainly discuss single-resource VM consolidation and multi-resource VM consolidation from the aspect of resource allocation.

2.1. Single-resource VM consolidation

It is relatively simple to study host overload detection, VM migration and VM consolidation via the assignment of CPU resource. In previous studies [3,11], the upper and lower boundaries of CPU utilization were defined to classify PMs into three categories: underloaded, normal and overloaded. Some VMs were migrated from the overloaded physical machine to achieve the goals of performing load balancing, improving QoS and degrading the risk of overloading the CPU resource. The migrated VMs were redeployed in the normal PMs. All VMs in the underloaded physical machine were migrated, and the hosts were turned off to reduce energy consumption. However, the workload is random, and the host overload detection method, which is based on a fixed threshold, is incapable of adjusting the reserved idle resources according to the uncertain workload, which hinders the use of the VM consolidation method to properly allocate resources and causes undesired situations, such as poor service performance and high energy consumption.

Beloglazov A. et al. [22] made improvements based on the developments in [3] and proposed an adaptive heuristic VM dynamic consolidation method that analyzed the historical workload variation pattern of PMs and adaptively adjusted the overload threshold. Beloglazov A. et al. [22] proposed three host overload detection algorithms: the Median Absolute Deviation (MAD) algorithm, Interquartile Range (IQR) algorithm and Local Regression (LR) algorithm. The MAD and IQR algorithms measure the workload stability by calculating the median absolute deviation and interquartile range of recent CPU utilization, respectively. The overload threshold is decreased to reserve more resources for PMs with unsteady workloads. In this manner, the resource demands in the next phase are guaranteed and service quality is improved. However, the MAD and IQR algorithms disregard the recent workload variation trend. The result is that a physical machine with an unsteady load always needs to reserve a large number of resources, which actually decreases resource utilization and increases energy consumption. The LR algorithm forecasts CPU utilization via a local regression method and leverages the forecast value to proactively prevent physical machine overloading. However, for a highly fluctuating workload, obtaining an accurate prediction is difficult for the LR algorithm. To transfer workload by migrating some VMs from a physical machine with overload risk, Beloglazov A. et al. [22] proposed three VM migration selection algorithms: Minimum Migration Time (MMT), Maximum Correlation (MC) and Random Selection (RS). To realize VM placement, Beloglazov A. et al. [22] proposed the power aware best fit decreasing (PABFD) algorithm. First, the PABFD algorithm arranges the VMs in descending order based on the resource demand. Second, the migration of each VM is evaluated, and a physical machine with a minimum increase in energy consumption after VM placement is selected as the migration destination. The PABFD algorithm allocates the migrated VMs to the hosts with high energy efficiency; however, a reassessment of resource utilization and overload risk for the physical machine is not performed after VM migrations, which easily results in load imbalance.

On the basis of the developments in [3], some other approaches primarily forecast the resource utilization of PMs via an intelligent algorithm or adaptively adjust the threshold of CPU utilization. Shaw S. B. et al. [23] proposed a workload forecast via double-exponential smoothing. However, double-exponential smoothing parameters are difficult to determine in a dynamic load environment. Farahnakian F. et al. [24] predicted the upcoming resource utilization via the k-nearest-neighbor algorithm and combined two factors – current workload and forecasted workload – when determining physical machine overload risk. However, fast adaptive k-value selection is difficult to achieve. Farahnakian F. et al. [12] forecasted CPU utilization during VM consolidation via a linear regression method similar to that in [22]. Masoumzadeh S. S. et al. [25] learned the historical set of overload thresholds in different states of a data center via Fuzzy Q-learning and obtained a decision model to adjust the overload threshold to a proper value based on the state of the physical machine. However, this method required a long period of learning to converge and could not rapidly adapt to the environment. Moreover, the method is unable to use the recent workload variation pattern to determine whether the physical machine is overloaded. Masoumzadeh S. S. et al. [26] conducted a similar study, in which a proper overload threshold and a VM migration selection strategy was selected via a Q-learning method and were based on the CPU utilization of PMs and the quantity of VMs. However, this approach has the same deficiency as that in [25]: recent host workload variation is not considered in host overload identification.

Additionally, extensive research was also conducted based on the developments in [3]. For instance, Farahnakian F. et al. [12,18] forecasted the CPU or other resource utilization to ensure adaptive adjustment of the algorithm based on the dynamic workload. However, the aforementioned studies are based on heuristic algorithms, such as BFD [3,11], to redeploy the VMs. Due to the deficiencies of the algorithms themselves, these methods are likely to fall into local optima [3,11,12,18].

2.2. Multi-resource VM consolidation

In a real data center, VM consolidation is essentially a multi-resource scheduling or allocation issue because it is influenced by various factors, such as CPU utilization, memory utilization, bandwidth, and disk size. Farahnakian F. et al. [30] proposed the utilization prediction-aware best fit decreasing (UP-BFD) method. UP-BFD forecasts VM resource utilization, such as CPU and memory, based on the k-nearest-neighbor method, which is similar to the method in [24]. When conducting host overload detection and re-allocating VMs, both the current utilization and forecasted utilization of multiple resources are considered. However, UP-BFD cannot rapidly and adaptively determine a proper k value. Mishra M. et al. [15] proposed a vector-arithmetic-based multi-resource VM consolidation method. In this method, all resources are classified into three categories based on utilization—high, medium and low. VMs are classified into six categories based on VM resource utilization. VM placement is based on the proposed VM category matching rule. However, this method disregards the randomness of demands for each resource. VM dynamic consolidation is a nondeterministic polynomial-time (NP) hard problem, for which an optimization solution is difficult to obtain when the quantities of PMs and VMs are excessive. Therefore, Mishra M. et al. [15] investigated how to leverage greedy heuristic algorithms, such as BFD and FFD, to calculate a quasi-optimal solution. Although this solution reduces the computation cost, it easily falls into a local optimum. To address this problem, Ferdaus M. H. et al. [14] proposed a VM consolidation algorithm based on the ant colony optimization algorithm. This algorithm considers factors such as CPU, memory and network bandwidth resources and defines minimizing the energy consumption of data centers as an optimization objective. However, this algorithm does not consider the dynamic workload of VMs, lacks constraint on the scales of VMs and PMs to ensure the feasibility of VM consolidation and disregards the potential negative influence of high-volume VM migrations. To address these issues, Farahnakian F. et al. [31] proposed a VM consolidation algorithm based on ant colony system (ACS-VMC). In this algorithm, minimizing the number of running PMs and the number of VM migrations is defined as the optimization objective. The detection of overloaded and underloaded PMs is performed via the forecasting algorithm [12] and the method based on the upper and lower boundaries of CPU utilization [3]. These methods effectively reduce the problem search space and limit the total number of VMs to ensure the feasibility of VM consolidation. However, Farahnakian F. et al. [31] disregarded the randomness in demands for memory and network bandwidth. This algorithm detects the underloaded physical machines by defining the lower bound on CPU utilization, which makes it unable to adaptively determine which underloaded PMs should be turned off based on the load distribution of the data center.

Besides, other topics have also been addressed. Kaaouache M. A. et al. [19] studied the VM placement problem. The VM placement problem was abstracted as a bin-packing problem. A heuristic algorithm or intelligent optimization algorithm was enhanced to improve the mapping relation between VMs and PMs for a specific optimization objective. Aroca J. A. et al. [32] classified the VM placement issue based on a quantitative constraint between VMs and PMs; competitive ratio analysis was also performed to analyze the VM placement algorithm. Chen L. et al. [28] treated the VM migration decision issue as a Markov decision process. Based on the current workload, the VMs with a specific state were selected for migration to reduce the probability of VM re-migration and achieve the goal of load balancing. Sohrabi S. et al. [29] considered that live VM migration would suspend service, prolonged VM migrations could affect QoS and migrating VMs with "minimum migration time" could not effectively eliminate the resource overload risk of PMs. Therefore, the median migration time strategy was proposed to migrate the VMs with median migration time from the overloaded PMs. Voorsluys W. et al. [13] suggested that live VM migration may consume extra computing resources and network bandwidth since a virtual machine in migration suspends its service. Therefore, inefficient VM migration incurs SLA violations and induces extra resource and energy consumption. Beloglazov A. et al. [3] considered the negative effects of frequent VM migration, and therefore, the high-workload VMs were given priority to migrate from the overloaded physical machine. However, Mann Z. Á. et al. [17] theoretically proved that this method was unable to guarantee the minimal number of VM migrations, i.e., unnecessary VM migrations occurred. Additionally, some other potential issues such as the energy cost minimization using some effective schemes [33], data-oriented disaster recovery [34], VM-targeting attack [35] and so on, are also meaningful and benchmark works for data centers.

The proposed study conducts research on VM consolidation with sufficient consideration of the optimization requirements of energy efficiency; benefit of degrading the total number of VM migrations; comprehensive treatment of resources including CPU, memory and network bandwidth; and uncertainty in demands for each resource in data centers. Accordingly, we propose an EC-VMC method. First, multi-resource constraints with sufficient consideration of the randomness of each resource have been established to perform credible estimation of the energy consumption and VM migrations and to obtain multi-resource overload probability estimates via the statistical methods. Second, VM consolidation is abstracted as a multi-objective optimization problem with multiple resource constraints; a VM dynamic consolidation optimization model is established. Third, several sub-algorithms corresponding to different phases of VM dynamic consolidation are proposed, which complete each sub-task of VM consolidation. Finally, the mapping relation between PMs and VMs is assumed as a food source, and the inter-cooperation of sub-algorithms simulates artificial bee colony foraging behavior. By using both the searching mechanism and optimization strategy of the artificial bee colony (ABC) algorithm, an approximate feasible solution is obtained. Consequently, the EC-VMC method is capable of achieving appropriate balance among guaranteeing service quality, reducing energy consumption and maximizing resource utilization.

3. Virtual machine dynamic consolidation description

3.1. Data center resource assumption

This section describes the resource assumption of data centers in this paper. Assume that $H = \{h_1, h_2, \ldots, h_i, \ldots, h_n\}$ is the physical machine set; $VM = \{vm_1, vm_2, \ldots, vm_j, \ldots, vm_m\}$ is the virtual machine set; and VM_i is the virtual machine set deployed in the physical machine h_i. $x_{ij} \in \{0, 1\}$, where if $x_{ij} = 1$, the virtual machine vm_j is deployed in the physical machine h_i, which is denoted as $vm_j \in VM_i$; otherwise, the virtual machine vm_j is not deployed in the physical machine h_i, which is denoted as $vm_j \notin VM_i$. The matrix $\boldsymbol{X} = (x_{ij})_{n\times m}$ describes the mapping relation between the PMs and VMs. RS is the resource category set, including the CPU, memory and network bandwidth resources, i.e., $RS = \{CPU, MEM, BW\}$. $C^r(vm_j)$ is the actual configured capacity of resource r, $r \in RS$, in virtual machine vm_j; $C^r(h_i)$ is the configurable capacity of resource r in the physical machine h_i; $D^r(vm_j)$ is the demand of the virtual machine vm_j for the resource r (unit: percentage); and $U^r(h_i)$ is the actual utilization of the resource r in the physical machine h_i; the $U^r(h_i)$ calculation is shown in formula (1).

$$U^r(h_i) = \frac{1}{C^r(h_i)} \sum_{vm_j \in VM_i} D^r(vm_j) \cdot C^r(vm_j) \tag{1}$$

3.2. Data center energy consumption estimation

High energy consumption is a major challenge in data centers. With the widespread use of cloud-computing technology, the scale of cloud data centers is expanding, and the problem of high energy consumption is more prominent. Data center energy consumption is not only from the running physical machines but also the computer air conditioning unit (CRAC) and other network equipment such as routers. The green cloud-computing model [36,37] focuses on optimizing the energy consumption generated by network equipment and service computing as well, and is very promising. In this paper, we are mainly concerned with how to improve the resource utilization of PMs to reduce energy consumption in data centers.

According to the power consumption data set [38], a diagram for the relation between CPU utilization and the host power is provided in Fig. 1. When the CPU utilization is less than 70%, the approximately linear characteristic is more obvious. In previous works [3,10,14], the energy consumption of data centers was approximated by the linear relation between the power consumption of PMs and the CPU resource utilization. By assuming the linear relation between the CPU resource utilization and the power of PMs, a simple unary linear function was constructed to estimate the energy consumption of hosts in [3,10,14]. However, the accuracy is not good enough. In this paper, since the distribution of CPU

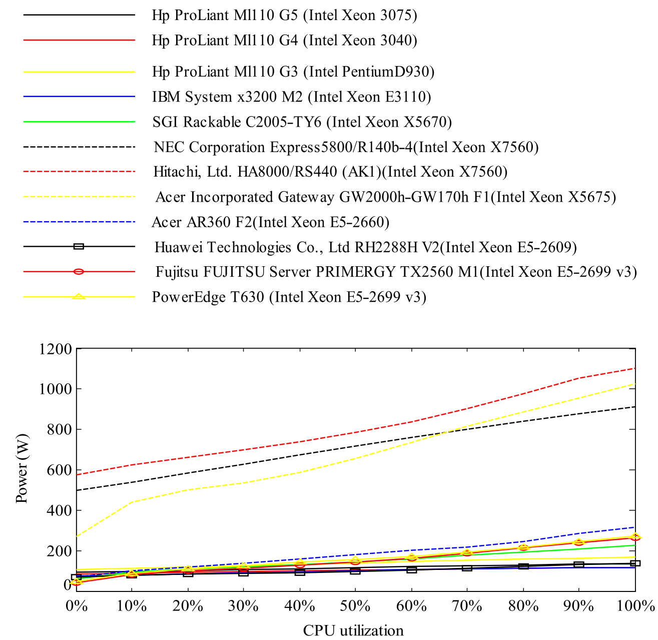


Fig. 1. The relation between CPU utilization and host power.

resource utilization in Fig. 1 is equidistant, with each ascending 10% representing a grade, and the relation is approximately linear in each interval, a method of estimating energy consumption based on local linearity is presented. Namely, we assume that the CPU resource utilization and the power of PMs are linearly related within each interval. Then, the power of the physical machine h_i can be calculated with formula (2),

$$f_{power}(h_i) = s_i \cdot \sum_{l=1}^{L} \left[p_l'(h_i) + \left(p_l(h_i) - p_l'(h_i)\right) \cdot rat(U^r(h_i), B_l) \right] \cdot I_l(h_i), r = CPU, s_i \in \{0, 1\} \tag{2}$$

where s_i is affected by the mapping relation matrix X. When $\sum_{j=1}^{m} x_{ij} > 0$, $s_i = 1$, which denotes that the physical machines are running; otherwise, $s_i = 0$, which indicates that the physical machines are asleep. Assume that CPU resource utilization is divided into L ranges $(B_1, B_2, \ldots, B_l, \ldots, B_L)$. If the CPU utilization of the physical machine h_i belongs to range B_l, then the function $I_l(h_i) = 1$; otherwise, $I_l(h_i) = 0$. If $I_l(h_i) = 1$ holds, the power of h_i can be estimated as $p_l'(h_i) + \left(p_l(h_i) - p_l'(h_i)\right) \cdot rat(U^r(h_i), B_l)$ according to the local linearity, where $p_l(h_i)$ represents the upper bound on the power of h_i within the range B_l, and $p_l'(h_i)$ represents the lower bound on the power of h_i within the same range. $rat(U^r(h_i), B_l)$ represents the ratio of CPU resource utilization within the range B_l. For example, if B_l is located inside [30%, 40%) and $U^r(h_i)$ is 35%,then $rat(U^r(h_i), B_l)$ is 0.5. When the $\boldsymbol{X}$=$(x_{ij})_{n\times m}$ distribution precondition is satisfied, the data center energy consumption estimation is calculated via formula (3).

$$F_{power}(\boldsymbol{X}) = \sum_{h_i \in H} f_{power}(h_i) \tag{3}$$

3.3. The number of VM migrations

The goal of VM consolidation is to determine the mapping relation between PMs and VMs for VM migration to achieve the goals of improving QoS and reducing energy consumption. Assume the mapping relation between PMs and VMs before VM consolidation is denoted as $\boldsymbol{X}=(x_{ij})_{n\times m}$, and the mapping relation after VM consolidation is denoted as $\widetilde{\boldsymbol{X}}=(\widetilde{x}_{ij})_{n\times m}$. If $x_{ij} \neq \widetilde{x}_{ij}$, i.e., the mapping between the virtual machine vm_j and the physical machine h_i changes, let $y_{ij} = 1$; otherwise, let $y_{ij} = 0$. Assume that the virtual machine vm_j migrates from the physical machine h_s to the physical machine h_d; $x_{sj} \neq \widetilde{x}_{sj}$ and $x_{dj} \neq \widetilde{x}_{dj}$. This case shows that one VM migration will generate two unequal relations in the mapping matrix $\boldsymbol{X}$. Therefore, the total number of VM migrations should be equal to half of $\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}$. Based on this rule, the number of VM migrations is estimated by formula (4).

$$F_{mig}(\boldsymbol{X}, \widetilde{\boldsymbol{X}}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} \tag{4}$$

3.4. Multi-resource overload probability estimation

Once the physical machine is overloaded, the service quality could be substantially affected. During VM consolidation, resource overloading should be avoided by effectively limiting the utilization of resources to within a certain range. However, the workload of a physical machine is the aggregated workload of the VMs that it hosts, and a virtual machine's demand for each resource varies randomly, which makes it more difficult to obtain deterministic resource consumption estimates. To address this issue, we use stochastic demand models for virtual machines' demands for each resource.

By constraining the resource overload probability, the optimal mapping between VMs and PMs can maximize the resource utilization and energy efficiency. The proposed resource overload probability estimation model is defined in (5),

$$P^r_{over}(h_i) = 1 - \Pr(R^r(h_i) < C^r(h_i)), \quad r \in RS \tag{5}$$

where $R^r(h_i) = \sum_{vm_j\in VM_i} D^r(vm_j)\cdot C^r(vm_j)$, $r \in RS$. Here, $R^r(h_i)$ is the workload for resource r generated by the VMs deployed in the physical machine h_i.

Existing research [39–42] assumes that the virtual machine's demand for a resource follows a normal distribution. Yu L. et al. [42] analyzed the PlanetLab trace [43] and the Google cluster trace and discovered that the distribution of these workloads resembled a normal distribution. Therefore, this paper also assumes that virtual machines' demand for each resource follows a normal distribution. Assume the demand of the virtual machine vm_j for resource r satisfies $D^r(vm_j) \sim N(\mu^r(vm_j), \sigma^r(vm_j)^2)$, where $\mu^r(vm_j)$ is the average demand of the virtual machine vm_j for resource r and $\sigma^r(vm_j)$ is the standard deviation of the demand of the virtual machine vm_j for resource r. $\mu^r(vm_j)$ and $\sigma^r(vm_j)$ are obtained by analyzing the recent requirement data for resource r. Therefore, $\Pr(R^r(h_i) < C^r(h_i))$ is calculated via formula (6),

$$\Pr(R^r(h_i) < C^r(h_i)) = \Phi\left(\frac{C^r(h_i) - \mu^r(h_i)}{\sigma^r(h_i)}\right) \tag{6}$$

where $\mu^r(h_i) = \sum_{vm_j\in VM_i} C^r(vm_j)\cdot\mu^r(vm_j)$ and $\sigma^r(h_i) = \sqrt{\sum_{vm_j\in VM_i}\left[C^r(vm_i)\cdot\sigma^r(vm_j)\right]^2}$. Here, Φ is the probability distribution function for the standard normal distribution; $\mu^r(h_i)$ is the average workload for the resource r in the physical machine; and $\sigma^r(h_i)$ is the standard deviation of the workload for resource r in the physical machine.

3.5. Optimization objective

The dynamic consolidation of virtual machines achieves the goal of degrading the energy consumption of data centers, optimizing resource utilization and improving QoS by adjusting the mapping relation between VMs and PMs. However, frequent VM migrations have a negative influence. Voorsluys W. et al. [13] suggest that live VM migration consumes extra computing resources; thus, excess VM migrations spur a significant workload and increase energy consumption. Because a virtual machine in live migration suspends its service, the service quality is affected. Therefore, reducing the total number of VM migrations is very critical.

To address this issue, multiple factors that affect VM consolidation are considered, and accordingly, the following VM consolidation optimization objectives are defined:

$$\begin{cases} \min F_{power}(\widetilde{\boldsymbol{X}}) \\ \min F_{mig}(X, \widetilde{\boldsymbol{X}}) \\ \min P^r_{over}(h_i), \forall h_i \in H, r \in RS \end{cases}$$

where the data center energy consumption $F_{power}(\widetilde{X})$, the number of VM migrations $F_{mig}(X, \widetilde{X})$ and the overload probability for each resource $P^r_{over}(h_i)$.

To simplify the optimization objectives, these multi-objective optimization problems are converted. First, min $P^r_{over}(h_i)$ is converted to an "ε -constraint" problem, i.e., $P^r_{over}(h_i) < \varepsilon$. Here, ε is the upper limit of the overload probability for resource r. There is a credibility problem with the resource r overload probability estimation. Higher current physical machine resource utilization suggests higher overload risk for the physical machine. Therefore, ε is adaptively changed according to the physical machine workload. In this manner, the "ε-constraint" is converted to formula (7),

$$P^r_{over}(h_i) < \frac{1 - U^r(h_i)}{s}, \forall h_i \in H, r \in RS \tag{7}$$

where parameter s defines the ε variation scope. A smaller s indicates that the variation in ε is more sensitive, and higher physical machine resource utilization means a lower ε, which strengthens the alert for resource overloading.

Two optimization objectives, the data center energy consumption and the number of VM migrations, are integrated into formula (8) via a linear weighting method,

$$F(X, \widetilde{\boldsymbol{X}}) = F_{power}(\widetilde{\boldsymbol{X}}) + w_{mig}F_{mig}(X, \widetilde{\boldsymbol{X}}) \tag{8}$$

where w_{mig} is the relative weight of migrations, which is an experience parameter that is learned from several tests.

To facilitate the solution of this problem, the minimization problem is converted to a maximization problem, i.e., formula (8) is converted to formula (9),

$$F_{con}(X, \widetilde{\boldsymbol{X}}) = \left[F^{\max}_{power} - F_{power}(\widetilde{\boldsymbol{X}})\right] + w_{mig}\left[m - F_{mig}(X, \widetilde{\boldsymbol{X}})\right] \tag{9}$$

where $F^{\max}_{power} = \sum_{h_i\in H} P_L(h_i)$ represents the maximum energy consumption of data centers and m is the total number of VMs or the upper limit of the VM migrations.

After this transformation, the optimization objective of the VM dynamic consolidation is simplified as the optimization problem shown in formula (10).

$$\begin{cases} \max F_{con}(X, \widetilde{\boldsymbol{X}}) \\ s.t. \\ P^r_{over}(h_i) < \dfrac{1 - U^r(h_i)}{s}, \forall h_i \in H, r \in RS \end{cases} \tag{10}$$

The optimization problem shown in formula (10) is investigated and solved in Section 4.

4. EC-VMC dynamic consolidation method

4.1. VM migration selection

To solve formula (10) and obtain $\widetilde{X}$, all resources in all PMs should satisfy the constraint defined in formula (7). Some VMs are migrated from the PMs that do not satisfy the constraint or are overloaded. Therefore, the VM migration selection criteria are very critical. Improper selection criteria may incur excessive VM migrations, extra computational workloads and prolonged virtual machine service suspension due to prolonged VM migration, which can affect QoS. Therefore, the following VM migration selection strategy is defined: the VMs that have relatively shorter migration times and can significantly reduce the multiple-resource overload risk of PMs after migration are selected. Migration of these VMs is helpful for reducing both the VM migration time and the number of VM migrations. After the virtual machine vm_j is migrated from the physical machine h_i, the resource r overload probability is estimated via formula (11) and the resource overload probability descending gradient is calculated via formula (12).

$$P^r_{over}(h_i, -vm_j) = 1 - \Pr(R^r(h_i) - D^r(vm_j) < C^r(h_i)) \tag{11}$$

$$\Delta p_{over}(h_i, -vm_j) = \frac{1}{|RS|} \sum_{r\in RS} P^r_{over}(h_i) - P^r_{over}(h_i, -vm_j) \tag{12}$$

Live VM migration generally uses the pre-copying method. In each cycle of iteration, the dirty pages in memory will be retransmitted until the number of dirty pages is less than the set threshold, and then it conducts the last round of dirty page memory transfer. According to the model developed in [44,45], the migration time of the virtual machine from the physical host is the sum of the times spent for multiple memory page transfers. In terms of this rule, Liu H. et al. [45] proposed a base model of migration performance, which is based on the current usage of memory, dirty page rate, and data transfer rate, to estimate the migration time of the virtual machine and the total network traffic; this paper also uses this model to estimate the migration time $T_{mig}(h_i, vm_k)$ from the physical host h_i to the virtual machine vm_k. Moreover, the probability of VM migration integrates the migration time of the virtual machine, and the change in resource overload probability after migration of a virtual machine is given in formula (13). In formula (13), $\sigma_{mig} > 0$, which represents the weight of migration time. It can be seen from formula (13) that the larger the reductions in the resource overload probability of the physical machine after migration of a virtual machine and the shorter the migration time, the higher the probability that the virtual machine is selected for migration.

$$P_s(vm_k) = \frac{\Delta p_{over}(h_i, -vm_k) - \sigma_{mig} \cdot T_{mig}(h_i, vm_k)}{\sum_{vm_j \in VM_c} \Delta p_{over}(h_i, -vm_k) - \sigma_{mig} \cdot T_{mig}(h_i, vm_k)} \tag{13}$$

4.2. Virtual machine placement

A new destination host should be selected for VM placement to complete VM migration. To completely utilize the physical machine resource, a physical machine with low utilization is selected. The following two principles for priority are defined: (1) an underutilized physical machine has higher priority, and (2) a physical machine with low overload probability has higher priority. Because different PMs have heterogeneous energy consumption, the workload saturation level of the physical machine is measured by the remaining effective power after the migrated VMs have been deployed in this physical machine, which maximizes its energy efficiency. The PMs with low-overload probability after VM placement are selected with higher priority as the new destination hosts, which guarantee QoS. When these two principles for placement priority are combined, the heuristic criteria for selecting the destination physical machine h_i for the virtual machine vm_j placement are defined in formula (14),

$$\eta(h_i, +vm_j) = \left[\frac{p_L(h_i) - f_{power}(h_i, +vm_j)}{p_L(h_i) - p_1(h_i)}\right] + w_{over}\left[1 - \frac{1}{|RS|}\sum_{r\in RS} P^r_{over}(h_i, +vm_j)\right] \tag{14}$$

where $f_{power}(h_i + vm_j)$ represents the power of the physical machine h_i after deploying the migrated virtual machine vm_j ;$P^r_{over}(h_i + vm_j)$ is the resource γ overload probability of the physical machine h_i after deploying the migrated vm_j, whose calculation method is identical to that for $P^r_{over}(h_i, -vm_j)$ (e.g., shown in formula (11)); and w_{over} represents the relative weight of the overload probability, which is an experience parameter that is learned from multiple experiments. A large $\eta(h_i, +vm_j)$ indicates that the physical machine h_i is highly suitable as a host for the migrated virtual machine vm_j.

This paper is inspired by an idea [14], in which the pheromone matrix τ is created to save past experience that is acquired when searching for the feasible mapping relation between VMs and PMs. A similar strategy is employed in this paper: after each iteration, the pheromone matrix τ is updated via formula (15) based on the current optimum mapping relation between VMs and PMs. The pheromone matrix τ provides prior knowledge and experience for improving the mapping relation in the next iteration.

$$\begin{cases} \tau(h_i, vm_j) = (1-\rho)\cdot\tau(h_i, vm_j) + \Delta\tau(h_i, vm_j) \\ \Delta\tau(h_i, vm_j) = \begin{cases} F_{con}(X, \widetilde{\boldsymbol{X}}^*), & \widetilde{x}^*_{ij} = 1 \\ 0, & otherwise \end{cases} \end{cases} \tag{15}$$

where ρ is the pheromone attenuation coefficient; X is the mapping relation before VM consolidation; $\widetilde{X}^*$ is the currently identified optimum mapping relation; and $\widetilde{x}^*_{ij} = 1$ indicates that the virtual machine vm_j in the current optimum mapping relation is deployed in the physical machine h_i. To completely leverage the experience from $\widetilde{X}^*$, the influence of the mapping relation $\langle h_j, vm_j\rangle$ in $\widetilde{X}^*$ is enhanced via formula (15). A larger pheromone $\tau(h_i, vm_j)$ means that it is better to deploy the migrated virtual machine vm_j in physical machine h_i based on the experience accumulated during the solution process.

In terms of the heuristic criteria (14) and past experience from the pheromone matrix, the probability of the physical machine h_i being selected as the destination host for the migrated virtual machine vm_j is calculated via formula (16).

$$P_d(h_i, +vm_j) = \frac{\left[\tau(h_i, vm_j)\right]^\alpha \left[\eta(h_i, +vm_j)\right]^\beta}{\sum_{h_k\in H_a - H_o}\left[\tau(h_k, vm_j)\right]^\alpha \left[\eta(h_k, +vm_j)\right]^\beta} \tag{16}$$

where α and β represent the weights of heuristic information and pheromone, respectively, whose values are obtained via the method in [14]. H_a is the set of running PMs and H_o is the set of overloaded PMs. In combination with the idea developed in [14], $\left[\boldsymbol{\tau}(h_i, vm_j)\right]^\alpha \left[\eta(h_i, +vm_j)\right]^\beta$ integrates the accumulated experience and heuristic information during the iteration search process. The greater the value of $\left[\boldsymbol{\tau}(h_i, vm_j)\right]^\alpha \left[\eta(h_i, +vm_j)\right]^\beta$ is, the more advantageous the deployment of the migrated virtual machine vm_j in physical machine h_i is. A higher value is helpful to improving the resource utilization of physical machines and is desirable for obtaining a globally optimal solution. If the value is small, it is not advantageous for the migrated virtual machine vm_j to be deployed in physical machine h_i.

Based on the above, we propose an energy-aware and overload probability-estimation VM placement (EOPVMP) algorithm, which is shown as Algorithm 2.

4.3. Underloaded physical machine identification

In this part, we intend to discuss how to identify the underloaded PMs and the benefits of doing so.

Generally, a low CPU utilization means the physical machine is in an underloaded status. Assume that the function $e(z)$ represents the relation between the utilization z of CPU and the power of host h. As seen from Fig. 1, with the increase in CPU utilization,

Algorithm 1.Overload Probability Decreasing Migration Selection

1: Input: h_i, X
2: Output: VM_m, X
3: $VM_m \leftarrow \varnothing$
4: while $P^r_{over}(h_i) > [T_r - U^r(h_i)]/s, \exists r \in RS$ **do**
5: **for all** $vm_k \in VM_i$ **do**
6: Use equation (12) to calculate $\Delta p_{over}(h_i, -vm_k)$
7: According Base model of Migration Performance to estimate $T_{mig}(h_i, vm_k)$
8: **end for**
9: Use equation (13) to calculate probability distribution $P_{vms} = \{P_s(vm_k) \mid vm_k \in VM_i\}$
10: Select $vm_s \in VM_i$ following P_{vms} randomly
11: $VM_i \leftarrow VM_i - \{vm_s\}$
12: $VM_m \leftarrow VM_m \cup \{vm_s\}$
13: Update X, $x_{is} \leftarrow 0$
14: end while

Algorithm 2. Energy-aware and Overloaded Probability-estimation VM Placement

1: Input: $VM_m, H_a, H_o, H_s, \tau, X$
2: Output: X
3: for all $vm_k \in VM_m$ **do**
4: **Use (16) to calculate** $P_{vmd} = \{P_d(h_i, +vm_k) \mid h_i \in H_a - H_o\}$
5: **According to** P_{vmd} **select** $h_d \in H_a - H_o$ **randomly**
6: **if** $h_d = \varnothing$ **then**
7: $h_d \leftarrow \arg\min_{h_i \in H_s} f_{power}(h_i, +vm_k)$
8: $H_a \leftarrow H_a \cup \{h_d\}, H_s \leftarrow H_s - \{h_d\}$
9: **end if**
10: $VM_m \leftarrow VM_m - \{vm_k\}$
11: $VM_d \leftarrow VM_d \cup \{vm_k\}$
12: $x_{dk} \leftarrow 1$
13: end for

the uptrend of the increased host power gradually declines; this hypothesis is established in Eq. (17).

$$\frac{e(z+\Delta z)-e(z)}{\Delta z} \le \frac{e(z'+\Delta z)-e(z')}{\Delta z}, z \ge z' \tag{17}$$

Then, the following theorem holds.

Theorem 1. *CPU utilization d is generated by any VM that satisfies the following formula,*

$$e(z+d)+e(y-d) \le e(z)+e(y), z \ge y \ge 0 \tag{18}$$

where y represents the CPU utilization of the other hosts.

Proof. See Appendix.

This theorem indicates that migrating a virtual machine from a low-utilization physical machine to another physical machine that has relatively high utilization is helpful to reduce the energy consumption. Because the underloaded PMs usually deploy few VMs, switching the underloaded PMs to sleep mode is a feasible solution to minimize the energy consumption of data centers and reduce VM migrations. Based on these investigations, this paper proposes to estimate the probability that a physical machine is to be switched to sleep mode by the CPU utilization of the physical machine. Based on the estimated probability, we identify an underloaded physical machine by randomly selecting a physical machine that should be switched to sleep mode. The probability for the physical machine h_k to be selected is defined in (19).

$$P_s(h_k, H_c) = \frac{1-U^r(h_k)}{\sum_{h_k \in H_c} 1-U^r(h_i)}, r = CPU \tag{19}$$

where H_c is the candidate physical machine set, which is typically the difference set between the running PM set and the overloaded PM set. As shown in formula (19), lower CPU utilization indicates higher probability for the physical machine to be switched to sleep mode.

In terms of the aforementioned analysis, an iterative strategy is applied: in each round of iteration, a physical machine is randomly selected based on the probability defined in (19). Based on the EOPVMP allocation algorithm in Section 4.2, we attempt to migrate all VMs in the selected physical machine to other destination hosts with higher workload. If all destination hosts satisfy the resource constraint after hosting the migrated VMs, then this selected physical machine is switched to sleep mode. Otherwise, this physical machine is untouched and the next iteration is performed until all running and non-overloaded PMs are traversed. Based on this, we propose the following underloaded physical host sleep selection (UPHSS) algorithm.

Algorithm 3. Underloaded Physical Hosts Sleep Selection

Input: H_a, H_o, H_s, τ, X
Output: X
$H_e \leftarrow \varnothing, H_c \leftarrow H_a - H_o - H_e$
while($H_c \neq \varnothing$)
  Use (19) to calculate $P_{vmu} = \{P_s(h_i, H_c) \mid h_i \in H_c\}$
  According to P_{vmu} **select** $h_s \in H_c$ **randomly**
  $H_c \leftarrow \{h_i \mid U^r(h_i) \geq U^r(h_s) \wedge h_i \in H_c\}, H_e \leftarrow H_e \bigcup \{h_s\}$
  $VM_m \leftarrow VM_s$
  $\tilde{X} \leftarrow EOPVMP(VM_s, H_c, H_o, H_s, \tau, X)$
  if $VM_s \neq \varnothing$ **then**
    $VM_s \leftarrow VM_m$
  else
    $X \leftarrow \tilde{X}$
    $H_s \leftarrow H_s \bigcup \{h_s\}, H_a \leftarrow H_a - \{h_s\}$
  end if
  $H_c \leftarrow H_a - H_o - H_e$
end while

4.4. Virtual machine consolidation algorithm

The artificial bee colony (ABC) algorithm [16] is a meta-heuristic algorithm. The ABC algorithm treats the feasible solution as bee food, where a sufficient amount of food source means a higher-quality solution. The search optimization process in the ABC algorithm defines three types of bee colonies: each employed bee forages in a food source area and corresponds to a food source; an onlooker waits in the beehive, shares information with the employed bee to select a food source and then forages; and a scout searches for a new food source when the employed bee's forage becomes scarce.

Inspired by this strategy, assume that a feasible mapping relation between VMs and PMs in VM consolidation is a food source. The above three sub-algorithms integrate and cooperate to simulate the artificial bee colony foraging behaviors, and both the searching mechanism and optimization strategy of the ABC algorithm are employed in the iterative solutions. In foraging, an artificial bee shares the pheromone matrix to continuously improve the mapping relation between PMs and VMs and find the optimum solution. In the ABC algorithm, each bee has a dedicated task, which significantly improves the algorithm reliability, effectively prevents search stagnation and achieves global optimization results. Here, OPDMS, EOPVMP and UPHSS are integrated to create the novel ABC-like foraging for VMs consolidation (ABC-like FVC) algorithm, whose primary goal is to obtain the optimum mapping relation between PMs and VMs at the current cycle.

First, the ABC-like FVC algorithm selects VMs from the overloaded PMs for migration; second, these VMs are reallocated. Finally, the underloaded PMs are switched to sleep mode, and the quality of the mapping relation after VM consolidation is evaluated.

In the ABC algorithm, an onlooker randomly selects a honey source that is guarded by an employed bee based on the solution quality. In this paper, a pheromone matrix is created for each honey source, i.e., the mapping relation between PMs and VMs, to save the past experience accumulated during the process of searching for a feasible mapping relation. The probability that a honey source is selected is redefined in (20).

$$P_s(bee) = \frac{f_{best}(bee)}{\sum_{bee \in employed} f_{best}(bee)} \tag{20}$$

where $f_{best}\ (bee)$ is the employed bee's historically best honey source.

Therefore, host overload detection, VM migration selection and VM placement in VM consolidation are abstracted into the bee colony foraging optimization problem in the ABC algorithm, which is comprised of four phases: (1) the employed bee starts to forage, i.e., search for a feasible mapping relation between PMs and VMs by the ABC-like FVC algorithm; when a newly discovered mapping relation is superior to the optimal honey source owned by the employed bee, i.e., $f\ (bee) > f_{best}\ (bee)$, the optimization solution is updated; (2) an onlooker randomly selects a honey source based on the probability of the honey source being selected. The feasible mapping relation is searched based on the pheromone matrix of this honey source, i.e., a new mapping relation is recalculated via Algorithm 4, and the mapping relation is improved via the pheromone matrix of the selected honey source. When the newly discovered allocation relation is superior to the optimized honey source, the optimization solution is updated; (3) a scout resets each honey source whose optimization solution has not been updated for several iterations by initializing the pheromone matrix of this honey source and applying the discovered feasible mapping relation as the optimization solution for this honey source, and (4) the pheromone matrix of each honey source is updated via formula (15). These four procedures constitute a complete cycle of iteration in VM consolidation.

Based on the above, these four procedures are hybridized to form an energy-aware and multiple-resource overload probability constraint-based VM consolidation (EC-VMC) method.

5. Implementation and simulation

5.1. Experiment arrangements

This study employed the CloudSim toolkit [46] as the simulation platform. Since the lower bound on the number of VMs in the employed workload traces is approximately 800, the experiments simulated a cloud data center comprised of 800 heterogeneous physical machines. These physical hosts are classified into two categories: HP Enterprise ProLiant DL360 Gen9 and Huawei Technologies Co., Ltd. Fusion Server RH2288H V3. The detailed configurations are listed in Table 1. Besides, eight types of VMs are used

Algorithm 4. ABC-like Foraging for VMs Consolidation

1: **Input:** $H_a, H_o, H_s, \tau_{bee}, X$
2: **Output:** $f_{bee}, \tilde{X}_{bee}$
4: $VM_m \leftarrow \varnothing, \tilde{X}_{bee} \leftarrow X$
5: **for all** $h_j \in H_o$ **do**
6: $(VM_m^i, \tilde{X}_{bee}) \leftarrow OPDMS(h_i, \tilde{X}_{bee})$
7: $VM_m \leftarrow VM_m \cup VM_m^i$
8: **end for**
9: $\tilde{X}_{bee} \leftarrow EOPVMP(VM_m, H_a, H_o, H_s, \tau_{bee}, \tilde{X}_{bee})$
10: $\tilde{X}_{bee} \leftarrow UPHSS(H_a, H_o, H_s, \tau_{bee}, \tilde{X}_{bee})$
11: $f_{bee} \leftarrow F(X, \tilde{X}_{bee})$

Table 1
The physical machine instances.

Types	CPU	RAM (GB)
HP Enterprise ProLiant DL360 Gen9	Intel Xeon E5-2699 v3 36 Cores 2300 MHz	64
Fusion Server RH2288H V3	Intel Xeon E5-2698 v4 40 Cores 2200 MHz	64

Table 2
The virtual machine instances.

Types	CPU frequency (MIPS)	RAM (GB)
High-CPU medium instance	2500	0.85
Extra large instance	2000	3.75
Small instance	1000	1.7
Micro instance	500	0.613
M3.medium	1 × 2500	3.75
M3.large	2 × 2500	7.5
M3.xlarge	4 × 2500	15
M3.2xlarge	8 × 2500	30

Table 3
The properties of Planetlab trace.

Date	Number of VMs	Mean (%)	St. dev. (%)
03/03/2011	1052	12.31	6.68
06/03/2011	898	11.44	6.77
09/03/2011	1061	10.70	7.35
22/03/2011	1516	9.26	6.24
25/03/2011	1078	10.56	6.32
03/04/2011	1463	12.39	7.03
09/04/2011	1358	11.12	6.95
11/04/2011	1233	11.56	7.13
12/04/2011	1054	11.54	7.22
20/04/2011	1033	10.43	8.10

in the presented experiments, four of them are introduced in [22] and the remainder are the type of M3 of the well-known EC2 [47]. The virtual machine instances are shown in Table 2. In Table 2, the frequency of the hosts' CPUs is mapped onto MIPS ratings [22]: 2300 MIPS each core of the HP Enterprise ProLiant DL360 Gen9 server and 2200 MIPS each core of the Fusion Server RH2288H V3 host. And more, the number in front of the expression of CPU frequency, such as the "2" in "2×2500" for "M3.large" in Table 2, denotes two pieces of vCPU [47]. After creating physical host instances and VM instances on the CloudSim platform, the VMs are deployed to different PMs by the method of PABFD [22]. After each cycle of VM consolidation, the VMs upload new workloads and change the overall resource demands.

A VM consolidation is conducted every 5 min, so the total number of consolidations is 288 one day. The comparison experiment include two parts: (1) the analytical comparison between the EC-VMC algorithm and the single-resource VM consolidation methods that address the optimization of energy consumption of data centers; and (2) the analytical comparison among EC-VMC algorithm and several existing multi-resource VMs consolidation methods. In this paper, two types of workload trace, the PlanetLab trace [43] and Bitbrains trace [48], are employed. PlanetLab only records CPU usage and packages the text file of time series of CPU usage; thus, the PlanetLab trace is used for part (1) of the comparison experiments. The statistical properties of the PlanetLab trace are shown in Table 3. Bitbrains includes CPU usage, memory, and workload of network bandwidth; thus, its trace for a multi-resource workload is taken for part (2). Table 4 shows the relevant statistical properties. In a real data center, since the actual workload is regular and periodic, we take one-day workloads from the different datasets of workload traces respectively, thus can get two experimental samples, one is for part (1) and another for part (2). At last, in the two parts of the experiments, the EC-VMC-related parameters are set as follows: $w_{mig} = 9$, $w_{over} = 0.5$, $\rho = 0.2$, $\alpha = 1$ and $\beta = 1.5$. The numbers of employed bees and scout bees are both 10, and the number of onlooker bees is 5. The maximum number of iterations is 25. All these experimental parameters are set based on the experience values.

Additionally, to simulate the process of live VM migration more accurately, this paper draws on the Base Model of Migration Performance [45] to simulate the variation of dirty pages in memory, and uses this model to get the virtual machine migration time T_{mig} and the total network traffic V_{mig}. With respect to the model in [45], it is necessary to set a dynamically changing dirty page memory rate D (in MB/s) for each virtual machine and a memory transmission rate R (in MB/s) during migration; D is assumed to be a random variation, and it satisfies $D/V_{mem} \sim N(0.3, 0.1)$, where V_{mem} is the current size of VM memory during migration, so that we set $R = D + 100$ MB/s in the experiments according to the adaptive data transmission rate strategy in [45].

5.2. Comparison with single-resource constraint-based VM consolidation

5.2.1. Evaluation indices

In [22], SLA violation Time per Active Host (SLATAH), Performance Degradation due to Migrations (PDM), SLA Violations (SLAV), Energy Consumption (EC), and Energy and SLA Violations (ESV) were used for performance valuation. In this section, these indices are employed to evaluate the performances of the compared algorithms objectively.

SLATAH is defined in (21). SLATAH measures the service quality of a running physical machine.

Algorithm 5. EC-VMC

Substituting (7) to calculate $H_o = \{h_i \mid \neg(P^r_{over}(h_i) < \frac{1-U^r(h_i)}{s}), \exists r \in RS, h_i \in H_a\}$
repeat
  for all $bee \in employed$ **do**
    $(f(bee), \tilde{X}(bee)) \leftarrow ABC-FVC(H_a, H_o, H_s, \tau_{bee}, X)$
    if $f(bee) > f_{best}$ **then**
      $f_{best} \leftarrow f(bee), \tilde{X}_{best} \leftarrow \tilde{X}(bee)$
    end if
    if $f(bee) > f_{best}(bee)$ **then**
      $f_{best}(bee) \leftarrow f(bee), \tilde{X}_{best}(bee) \leftarrow \tilde{X}(bee)$
      $stag(bee) \leftarrow 0$
    else
      $stag(bee) \leftarrow stag(bee)+1$
    end if
  end for
  for all $looker \in onlookers$ **do**
    According to $\{P_s(bee) \mid bee \in employed\}$ **select** $look \leftarrow bee$ **randomly**
    $(f(look), \tilde{X}(look)) \leftarrow ABC-FVC(H_a, H_o, H_s, \tau_{look}, X)$
    if $f(look) > f_{best}$ **then**
      $f_{best} \leftarrow f(look), \tilde{X}_{best} \leftarrow \tilde{X}(look)$
    end if
    if $f(look) > f_{best}(look)$ **then**
      $f_{best}(look) \leftarrow f(look), \tilde{X}_{best}(look) \leftarrow \tilde{X}(look)$
      $stag(look) \leftarrow 0$
    else
      $stag(look) \leftarrow stag(look)+1$
    end if
  end for
  for all $scout \in employed$ **do**
    if $stag(scout) > stag_{\max}$ **then**
      $\tau_{scout} \leftarrow \tau_0$
      $(f(scout), \tilde{X}(scout)) \leftarrow ABC-FVC(H_a, H_o, H_s, \tau_{scout}, X)$
      if $f(scout) > f_{best}$ **then**
        $f_{best} \leftarrow f(scout), \tilde{X}_{best} \leftarrow \tilde{X}(scout)$
      end if
      $f_{best}(scout) \leftarrow f(fcout), \tilde{X}_{best}(scout) \leftarrow \tilde{X}(scout)$
      $stag(scout) \leftarrow 0$
    end if
  end for
  for all $bee \in employed$ **do**
    $\tilde{X}^* \leftarrow \tilde{X}_{best}(bee)$
    Use (15) update τ_{bee}
  end for
  $iter \leftarrow iter+1$
until $iter > iter_{\max}$

$$SLATAH = \frac{1}{n}\sum_{i=1}^{n}\frac{T_i^s}{T_i^a} \tag{21}$$

where T_i^s is the SLAV duration resulting from overloaded CPU resources for the physical machine h_i , T_i^a denotes the running time of the physical machine h_i, and n represents the number of PMs.

PDM is defined in (22). It reflects the extent of the performance decline resulting from VM migrations.

$$PDM = \frac{1}{m}\sum_{j=1}^{m}\frac{C_j^d}{C_j^r} \tag{22}$$

where c_j^d denotes the size of the unsatisfied demand for CPU resources as a result of vm_j migration, c_j^d is the size of the demand for CPU resources from the virtual machine vm_j, and m represents the number of VMs.

SLAV comprehensively evaluates the QoS of a data center on a single day. It is calculated by formula (23).

$$SLAV = SLATAH \times PDM \tag{23}$$

The lower the values of SLATAH, PDM, and SLAV are, the better the QoS is.

Table 4
The properties of Bitbrains trace.

Date	Number of VMs	CPU		Memory		Bandwidth	
		Mean (%)	St. dev. (%)	Mean (%)	St. dev. (%)	Mean (%)	St. dev. (%)
02/08/2013	1237	7.20	5.97	8.83	4.35	0.76	1.84
04/08/2013	1233	8.05	4.83	9.75	4.12	0.80	1.77
05/08/2013	1232	8.99	6.36	9.13	4.10	0.83	1.72
08/08/2013	1209	10.27	6.64	9.69	4.52	0.70	1.67
11/08/2013	1202	9.06	6.92	9.74	4.41	0.96	2.04
15/08/2013	1191	8.71	5.67	9.68	4.20	0.95	1.93
19/08/2013	1188	8.13	5.34	9.20	3.63	0.88	1.76
20/08/2013	1186	8.99	3.45	9.10	3.05	0.79	1.46
22/08/2013	1183	5.89	3.16	8.83	3.21	0.94	1.60
24/08/2013	1175	9.56	4.68	9.40	3.75	1.29	2.56

Table 5
Simulation results using Planetlab trace with four indices.

Method	EC (kWh)	SLAV	VMMs	ESV (%)
EC-VMC	31.35	0.00068	6153	0.0213
MAD-MMT-2.5	64.73	0.04025	33 854	2.6054
MAD-MC-2.5	65.42	0.05126	35 343	3.3534
MAD-RS-2.5	58.57	0.07205	27 916	4.2200
LR-MMT-1.2	62.89	0.03299	31 744	2.0747
LR-MC-1.2	63.05	0.04431	31 891	2.7937
LR-RS-1.2	55.28	0.06167	24 716	3.4091
IQR-MMT-1.5	65.93	0.04540	34 614	2.9932
IQR-MC-1.5	66.91	0.05100	35 850	3.4124
IQR-RS-1.5	58.68	0.06923	27 514	4.0624
ST-MMT-0.8	65.44	0.04082	34 250	2.6713
ST-MC-0.8	66.30	0.04777	35 196	3.1672
ST-RS-0.8	59.01	0.06516	28 194	3.8451

EC indicates the energy consumption of a data center in a single day. A low EC value indicates high energy utilization and energy efficiency of the data center. The comprehensive evaluation index ESV, which is defined in formula (24), reflects the energy consumption, number of VM migrations, and service quality.

$$ESV = EC \times SLAV \tag{24}$$

A low ESV value indicates good performance in saving energy and guaranteeing the service quality of the data center.

Since VMs always suspend service when they are in live migration, prolonged VM migration may also further affect QoS. Reducing the number of insignificant VM migrations and the total number of VM migrations is beneficial to improving the QoS. Therefore, if limited VM migrations can yield ideal effects of VM consolidation, it indicates that the VM consolidation method is highly efficient.

5.2.2. Results analysis

In this section, four host overload detection algorithms (ST [3], MAD [22], IQR [22], LR [22]) and three VM migration selection algorithms (MMT, MC and RS) are hybridized to form 12 combination methods, which are compared with the EC-VMC algorithm. These combination methods employ PABFD [22] to conduct VM placement. They only consider the effect of a single-factor CPU resource; the experimental parameters are set based on the values in [3,22].

Table 5 lists the simulation results for various combination methods versus the EC-VMC algorithm. The numerical value after each algorithm's name is the current parameter setting for this algorithm, which is the experience parameter. The results indicate that the EC-VMC method consumes the least amount of energy and that IQR-MC-1.5 consumes the greatest amount of energy. The experimental results based on the PlanetLab trace indicate that the EC index of EC-VMC is only 46.9% of IQR-MC-1.5 and 56.7% of LR-RS. Compared with other combination methods, EC-VMC significantly reduces the energy consumption because the UPHSS in EC-VMC switches as many underloaded PMs to sleep mode as possible, which yields excellent energy savings.

Next, the methods are compared and analyzed in terms of SLAV. In Table 5, EC-VMC has optimal SLAV, followed by LR-MMT. The simulation results using the PlanetLab trace show that EC-VMC's SLAV is only 2.1% of LR-MMT's SLAV. The experimental results using the PlanetLab trace also indicate that MAD-RS has the highest SLAV and that EC-VMC's SLAV is only 0.9% of MAD-RS's SLAV. Therefore, EC-VMC's capability of guaranteeing QoS is superior to those of the other algorithms.

Because SLAV addresses SLATAH and PDM, SLATAH and PDM are analyzed as follows. Fig. 2 shows the comparison in terms of SLATAH. As shown in Fig. 2, EC-VMC effectively guarantees the QoS of running PMs and reduces their resource overload risk. The primary reasons are that EC-VMC considers the randomness of demands for each resource and constrains the utilization of each resource, which effectively guarantees superior QoS of the running PMs. When redeploying a virtual machine, the EOPVMP algorithm considers the overload probability of the physical machine after it hosts the migrated virtual machine, which guarantees its running QoS from another perspective. Fig. 3 shows the comparison of PDM among the methods; the EC-VMC method effectively reduces the probability of influencing service quality by "service suspension during live VM migration". According to the combined analysis with Fig. 2, the main reason for this situation is that the EC-VMC method effectively guarantees excellent QoS of the running PMs and reduces host overload risk, which results in a reduction in the number of VM migrations triggered by host overload. The "VMMs" index in Table 5 also proves this point. The VMMs index reflects the VM migrations. The OPDMS algorithm reduces the time of VM migration and avoids the occurrence of resource demand violations from VMs due to prolonged migration time. Therefore, the EC-VMC method has a lower PDM than the other methods, and the EC-VMC method has an outstanding SLAV compared with the other methods.

Regarding the number of VM migrations, in Table 5, EC-VMC has the lowest number of triggered VM migrations, followed by the LR-RS algorithm. However, the experimental results using the PlanetLab trace reveal that the number of VM migrations triggered by EC-VMC is only 24.9% that of LR-RS. The results also show that IQR-MC has the greatest number of VM migrations, which is 5.8 times that of EC-VMC. This is primarily because EC-VMC accurately identifies the variation patterns of the workload and effectively reduces the host overload risk and number of VM migrations.

The performances obtained using the ESV index are shown in Table 5, which reveal that the comprehensive performance of EC-VMC is the best. LR-MMT is the second best, with a performance that is considerably lower than that of EC-VMC. According to the simulation results using the PlanetLab trace, the ESV index of EC-VMC is only 1.0% of that of LR-MMT, whereas the ESV index of MAD-RS is the worst, being 19.8 times as great as that of EC-VMC. Therefore, having the best comprehensive performance, EC-VMC is

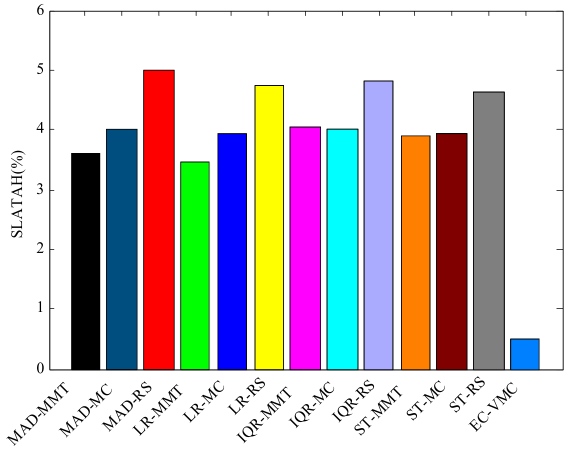


Fig. 2. Comparison of SLATAH.

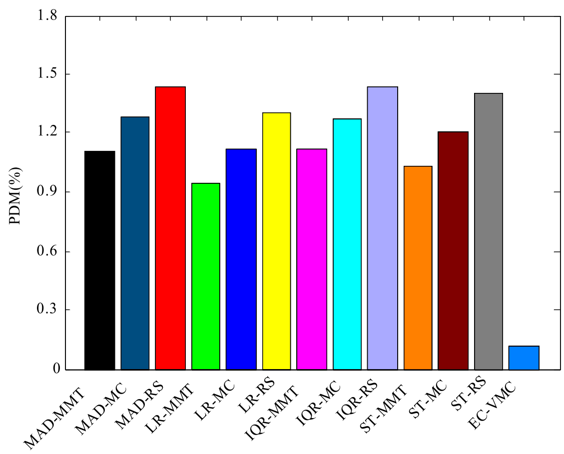


Fig. 3. Comparison of PDM.

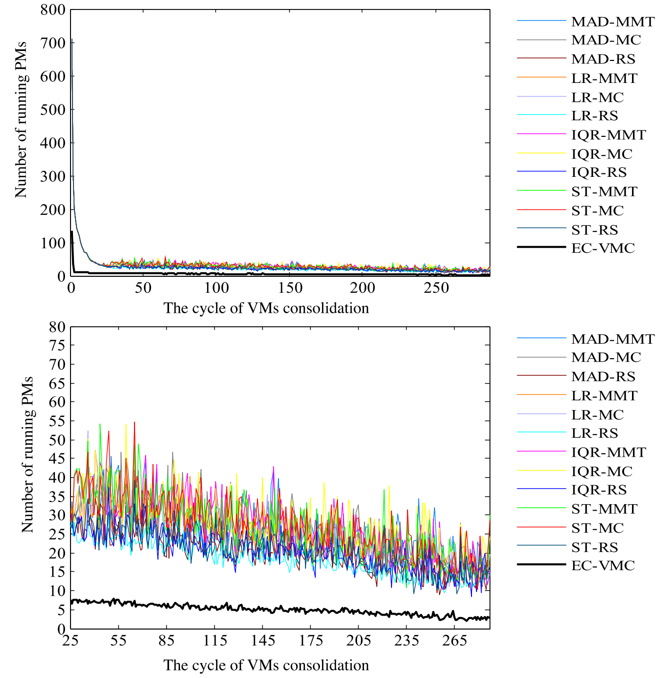


Fig. 4. The number of running PMs varying with the cycle of VMs consolidation.

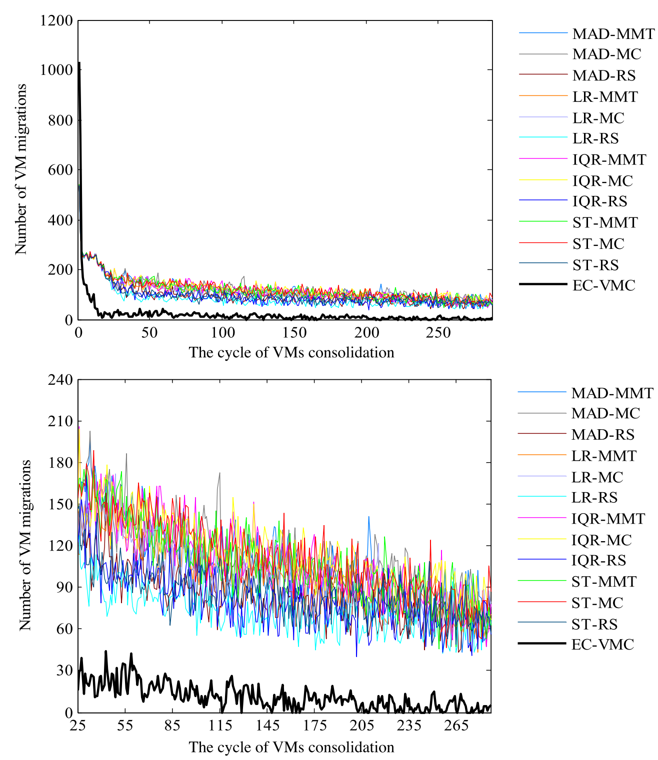


Fig. 5. The number of VM migrations varying with the cycle of VMs consolidation.

apparently superior to the other algorithms. These results indicate that EC-VMC has realized its ultimate optimization objectives of reducing the energy consumption of data centers and number of VM migrations.

To conduct a thorough analysis of the EC-VMC algorithm's efficiency, the variations in the numbers of running PMs and migrated VMs during all 288 consolidation cycles are analyzed, as shown in Fig. 4 and Fig. 5. In the graphs, the lines in different colors represent different algorithms, and their corresponding relations are shown on the right-hand side of the graph. Figs. 4 and 5 are divided into upper and lower sub-graphs. The upper graph represents the results from the initial VM consolidation to the 288th VM consolidation. To improve the visual discrimination, the lower sub-graph is a partially enlarged image of the upper sub-graph, representing the results from the 25th VM consolidation to the 288th VM consolidation.

Fig. 4 shows the variation in the number of running PMs with respect to the VM consolidation, which continues to be performed periodically. As shown in the upper subgraph in Fig. 4, EC-VMC can turn off many PMs faster than the other algorithms to conserve a massive amount of energy. The lower subgraph in Fig. 4 shows that changes occur in each cycle from the 25th VM consolidation to the 288th VM consolidation. EC-VMC can maintain no more than 10 running PMs, which is lower than the upper limits of the other compared algorithms; thus, the energy consumption for EC-VMC is apparently lower.

Fig. 5 shows the variation in the number of VM migrations with respect to the cycle of the ongoing VM consolidation. The upper subgraph in Fig. 5 shows that changes occur from the initial VM consolidation to the 288th VM consolidation. The number of VM migrations triggered by the EC-VMC algorithm within the initial 25 cycles is greater than those triggered by the other algorithms. In other words, many PMs have been turned off by the EC-VMC

Table 6
Simulation results using Bitbrains trace with four indices.

Method	EC (kWh)	SLAV	VMMs	ESV (%)
EC-VMC	27.49	0.00061	4491	0.0168
VectorGreedy	51.10	0.00134	8596	0.0685
UP-BFD	42.29	0.00219	8725	0.0926
ACS-VMC	60.71	0.00141	7814	0.0856

algorithm in the previous cycles, resulting in excess VM migrations. As shown in the lower subgraph in Fig. 5, after the 25th VM consolidation, the number of VM migrations triggered by EC-VMC remains within 0–30 approximately, which is remarkably lower than those triggered by the other algorithms. This result indicates that EC-VMC can perform appropriate load balancing through limited VM migrations. Efficient VM consolidation limits the number of VM migrations in the subsequent consolidation. Therefore, the total number of VM migrations with the EC-VMC algorithm is lower than with other algorithms.

Compared with the existing single-resource VM consolidation algorithms, EC-VMC has distinct advantages in terms of reducing energy consumption, guaranteeing QoS and making more reasonable VM migration decisions.

5.3. Comparison with multi-resource constraint-based VM consolidation

5.3.1. Evaluation indices

In this section, besides employing the evaluation indices in Section 5.2.1, we present two additional indices: one is memory demand violations (MDV), which aims to evaluate the capability to meet the memory demands of VMs; the other is bandwidth demand violations (BDV), which aims to evaluate the capability to meet the bandwidth demands of VMs. MDV is defined in formula (25).

$$MDV = \left(\frac{1}{|VM|}\sum_{vm_j\in VM}\frac{T_v^r(vm_j)}{T_a^r(vm_j)}\right) \times \left(\frac{1}{|VM|}\sum_{vm_j\in VM}\frac{C_v^r(vm_j)}{C_a^r(vm_j)}\right), r = MEM \tag{25}$$

where $T_v^r(vm_j)$ denotes the duration of the resource demand violation resulting from the vm_j requests to the memory resource, $T_a^r(vm_j)$ represents the total duration resulting from the vm_j requests to the memory resource; $C_v^r(vm_j)$ denotes the unsatisfied memory resource required capacity required by the vm_j, and $C_a^r(vm_j)$ is the memory resource capacity required by vm_j. BDV, formulated using the same template as MDV, is rewritten as formula (26), which addresses bandwidth resource.

$$BDV = \left(\frac{1}{|VM|}\sum_{vm_j\in VM}\frac{T_v^r(vm_j)}{T_a^r(vm_j)}\right) \times \left(\frac{1}{|VM|}\sum_{vm_j\in VM}\frac{C_v^r(vm_j)}{C_a^r(vm_j)}\right), r = BW \tag{26}$$

5.3.2. Results analysis

The EC-VMC algorithm is also compared with three multi-resource VM consolidation algorithms, i.e., VectorGreedy [15], UP-BFD [30], and ACS-VMC [31], using the Bitbrains trace.

Table 6 shows the results of the comparison of EC-VMC with the three existing multi-resource VM consolidation algorithms by employing the indices EC, SLAV, VMMs, and ESV. Undoubtedly,

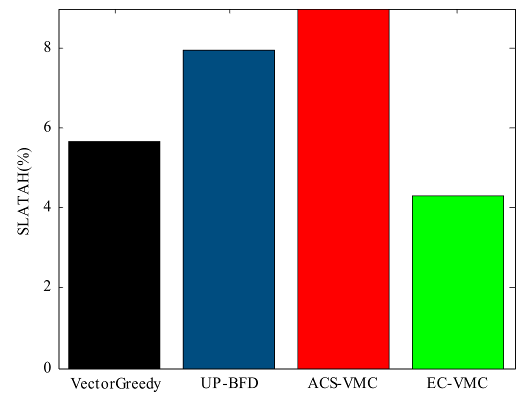


Fig. 6. Comparison of SLATAH.

the energy consumption for EC-VMC is the lowest, whereas the electrical energy consumed by ACS-VMC is the highest. According to the simulation results using the Bitbrains trace, the EC index of EC-VMC is 45.3% that of ACS-VMC and 65.0% that of UP-BFD. Compared with other methods, EC-VMC has significantly better capability to save energy because the sub-algorithm UPHSS in EC-VMC switches as many underloaded PMs as possible to sleep mode; and EC-VMC effectively avoids local optima due to the multiple iterations, such as the mechanism in the ABC algorithm. In this manner, global optimization results and better energy savings are achieved.

As shown in Table 6, the SLAV-driven performance of EC-VMC is the best, and that of VectorGreedy is the second best. The performances of ACS-VMC and VectorGreedy are approximately the same, whereas UP-BFD exhibited the worst performance. According to the experimental results using the Bitbrains trace, the SLAV value of EC-VMC is only 45.5% that of VectorGreedy and 27.9% that of UP-BFD. The SLAV index integrates both SLATAH and PDM. Thus, the in-depth analytical study of SLATAH and PDM is discussed in the following.

Fig. 6 shows the performance comparison using the SLATAH index. As shown in Fig. 6, the SLATAH index of EC-VMC is the lowest because it limits resources utilization depending on the randomness of resource demands and guarantees superior QoS of the running PMs. Furthermore, when a migrated VM is redeployed, the overloading risk of the destination physical machine is considered by the proposed EOPVMP algorithm; thus, the QoS of the destination physical machine is further guaranteed. Fig. 7 shows the comparison in terms of the PDM index. Compared with the other algorithms, the PDM index of EC-VMC is the lowest. EC-VMC can effectively reduce the probability of QoS deterioration due to the service suspension resulting from VM migrations. According to the comprehensive analysis of Fig. 6, EC-VMC can effectively guarantee QoS of the running PMs, thereby reducing the overloading risks. Consequently, the number of insignificant VM migrations resulting from the resource overloading of PMs is reduced. The simulation results of the number of VM migrations in Table 6 also demonstrate this outcome. Furthermore, the OPDMS algorithm can effectively avoid the demand violations of VMs resulting from prolonged VM migration. Therefore, the PDM index of EC-VMC is lower than those of the other algorithms. Moreover, its SLAV index is the lowest among the compared algorithms.

Regarding the number of VM migrations, in Table 6, the number of VM migrations triggered by EC-VMC is the lowest, followed by ACS-VMC. The number of VM migrations for EC-VMC is only 57.5% that of ACS-VMC. Furthermore, the number of VM migrations triggered by UP-BFD is the highest, being 1.94 times as great as that

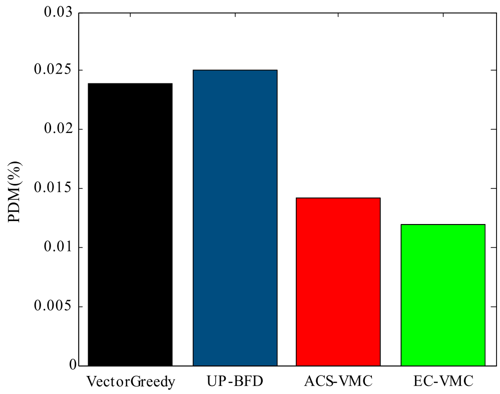


Fig. 7. Comparison of PDM.

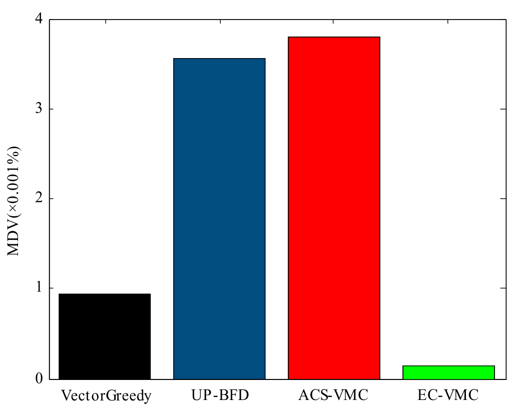


Fig. 8. Comparison of MDV.

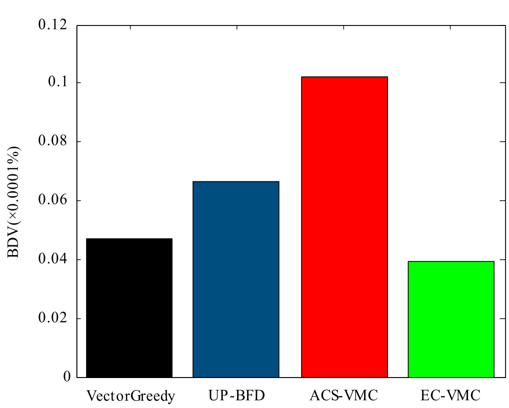


Fig. 9. Comparison of BDV.

triggered by EC-VMC. Overall, EC-VMC has the capability to reduce the number of VM migrations and overloading risks of hosts.

As shown in Table 6, the ESV value of EC-VMC is the lowest, indicating that it has the most comprehensive performance among the compared algorithms. The ESV value of EC-VMC is only 24.5% that of VectorGreedy, whereas the ESV value of UP-BFD is the worst, being only 5.51% times as great as that of EC-VMC. Therefore, the comprehensive performance of EC-VMC is the best, and the proposed algorithm is apparently superior to the other algorithms.

Fig. 8 shows the MDV index of the compared algorithms. Compared with other related algorithms, EC-VMC is designed to allow for proper allocation of memory resources and to ensure a low probability of demand violation of VMs for memory resources. The MDV values of ACS-VMC and UP-BFD are relatively high compared with that of EC-VMC. Fig. 9 shows the comparison results based on the BDV index among the related algorithms. EC-VMC and VectorGreedy have approximately the same capability to guarantee QoS of networks. The MDV and BDV indices of ACS-VMC are the highest among those of the compared algorithms. The primary reason is that the frequent VM migrations by ACS-VMC result in extra consumption of memory and network resources; this is supported by the experimental results in Table 6. Another reason is that ACS-VMC does not take into account the influence of dynamic changes, especially the stochastic resource demands for memory and network bandwidth. In summary, EC-VMC has relatively stronger capability to guarantee QoS among the compared algorithms.

The optimization objectives of EC-VMC are to reduce the energy consumption of the data center and the number of VM migrations. Since the energy consumption of a data center is proportional to the number of running PMs, for a sufficient analysis of the effectiveness of EC-VMC, the number of running PMs in the data center at each cycle along with the ongoing VM consolidation is analyzed, and the results are shown in Fig. 10.

Fig. 10 shows the variation in the number of running PMs with respect to each cycle of the ongoing VM consolidation. The upper subgraph in Fig. 10 shows the changes from the initial VM consolidation to the 288th VM consolidation. Compared with other algorithms, EC-VMC turns off many PMs relatively fast. To enhance the ability to distinguish among the compared algorithms, the lower subgraph in Fig. 10 is a partially enlarged image of the upper subgraph; it shows the changes from the 25th VM consolidation to the 288th VM consolidation. EC-VMC maintains the number of running PMs at 5–10, and this range is apparently lower than those of the other algorithms. In the latter period, i.e., after the 25th cycle of VM consolidation, the numbers of running PMs by the UP-BFD and VectorGreedy have been stably maintained at 10–25 respectively. ACS-VMC maintains the number of running PMs at approximately 25. In conclusion, the total number of running PMs with EC-VMC throughout all 288 cycles of VM consolidation is lower than those of the compared algorithms; thus, its energy consumption is lower than those of the others.

Fig. 11 shows that the constant changes in the times of the triggered VM migrations during each cycle vary with VMs consolidation. The upper subgraph in Fig. 11 shows the changes over the 288 cycles of VMs consolidation, from the initial VM consolidation to the 288th VM consolidation. The number of VM migrations triggered by EC-VMC within the initial 25 cycles is greater than those of the other algorithms. Since many PMs have been turned off in the initial 25 cycles, many VM migrations occur during the period. To enhance the ability to distinguish among the compared algorithms, the lower subgraph is a partially enlarged image of the upper subgraph. It shows the changes from the 25th VM consolidation to the 288th VM consolidation. After the 25th VM consolidation, the number of VM migrations triggered by EC-VMC is approximately 0–25, which is lower than those of other algorithms. This result indicates that the probability of re-migration for the migrated VMs after deployed by EC-VMC declined. The result, from another aspect, indicates that the resource allocation has been optimized and load balancing has been performed; as a result, the number of VM migrations triggered by EC-VMC is lower than those of the other algorithms.

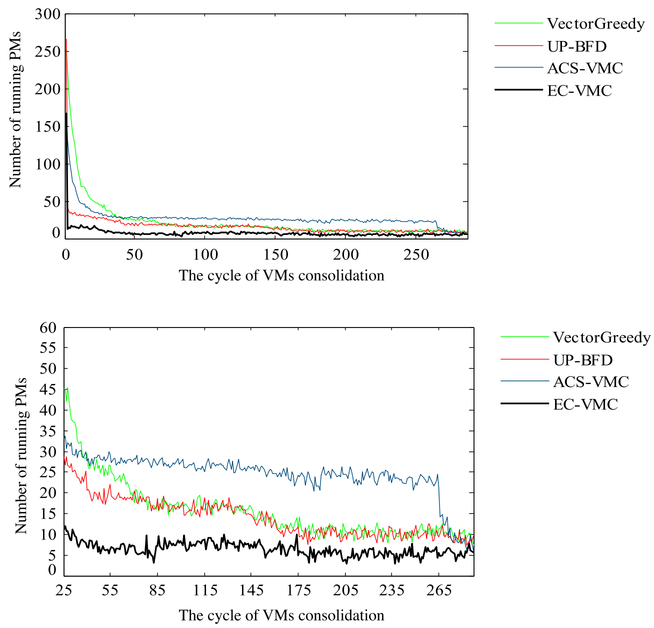


Fig. 10. The number of running PMs varying with the cycle of VMs consolidation.

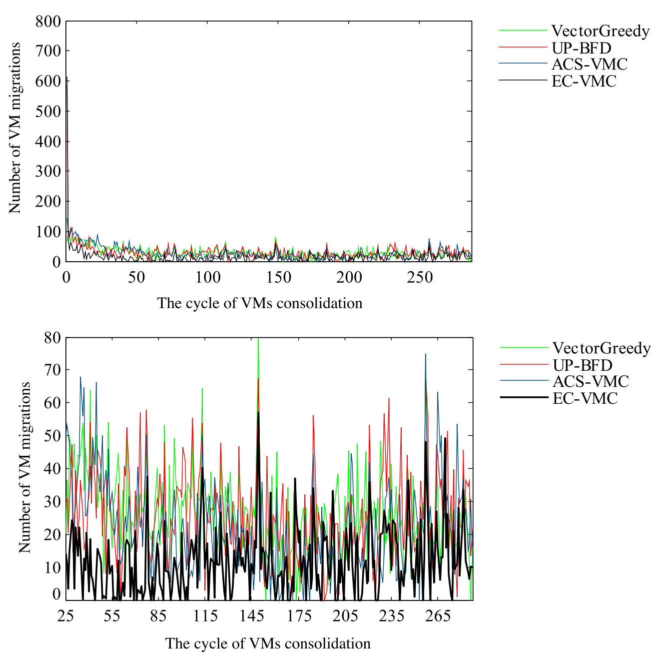


Fig. 11. The number of VM migrations varying with the cycle of VMs consolidation.

The analytical comparison of the EC-VMC with the single-resource and multi-resource VM consolidations show that the proposed method is prominent in various indices, since the foraging strategy and search mechanism in the ABC algorithm have comprehensively solved the issue of the optimization of resource allocation. The EC-VMC algorithm obtains a global optimization result, whereas the other compared algorithms prematurely fall into local optima. On the other hand, the results also indicate that the proper multi-resource overload probability constraint prevent frequent resource overloading, which is helpful to improving resource efficiency. In general, the EC-VMC method is effective and efficient and can reduce energy consumption, guarantee QoS, and make reasonable VM migration decisions.

5.4. Effectiveness of the proposed energy consumption model

The proposed energy consumption model, the linear model [3,10,14] and the nonlinear model [49] are experimentally compared to validate the effectiveness of the energy consumption model proposed in this paper. The linear model in [3] is defined as $P(u) = k \cdot P_{\max} + (1-k) \cdot P_{\max} \cdot u$, where u is the CPU utilization, $P_{\max}$ is the maximum power of hosts, and k is usually set to 70% since a physical machine in an idle state typically consumes 70% of its peak energy consumption. The nonlinear model in [49] is defined as $P(u) = P_{idle} + (P_{peak} - P_{idle}) \cdot u^{\alpha}$, $\alpha > 0$, where P_{idle} is the power when the host is idle and P_{peak} is the peak power. The above energy consumption models are combined with the EC-VMC method to conduct experiments respectively. The simulation platform for the experiment is CloudSim, and PlanetLab is used for the workload.

As VM consolidation proceeds, the total power of the running PMs and the changes in the situation are calculated; the experimental results are shown in Fig. 12.

Fig. 12 shows the changes in the sum of the powers of all running PMs with respect to the ongoing VM consolidation. In Fig. 12, "linear" represents EC-VMC using the linear model [3], "non-linear" denotes EC-VMC using the model in [49], and "local linear" represents EC-VMC using the proposed energy consumption model. It can be seen from the figure, through a limited number of VM consolidations, that initially, the use of three different energy consumption models associated with EC-VMC can cause the total power of the hosts to rapidly decline. The direct reason is that through the VM consolidation, a large number of physical machines are switched to sleep mode, and the deeper reason is that the global optimization of the ABC algorithm makes the allocation between VMs and PMs in the data center highly efficient. Then, the downward trend of the sum of the host power becomes slow; the reason for this phenomenon is that the total number of running PMs changes little, so the energy consumption does not change much. Fig. 4 also proves this point, and in the lower subgraph of Fig. 4, the number of running hosts only gradually drops from 10 to 5 in the subsequent cycles of VM consolidation, making it impossible to cause a big change in energy consumption.

In the three comparison schemes, EC-VMC using the local linearity energy consumption model consumes less energy; this proves the validity of the local linearity power model. In addition, we can see from Fig. 12 that using three different energy consumption models, the host energy variation trends are very similar; this fully demonstrates that the EC-VMC method is not sensitive to the energy consumption model and that EC-VMC's effectiveness and efficiency in saving energy and VM consolidation are due to the global optimization using the ABC algorithm.

6. Conclusions

VM migration is an approach to technical enablement of VM consolidation. VM migration facilitates load balancing, and VM consolidation aims to improve the energy efficiency and QoS of data centers. Both VM migration and VM consolidation are influenced by various factors. These factors include resource properties, such as CPU utilization, memory, bandwidth, disk size, and energy consumption, and performance properties of data centers, such as the stochastic workload and the dynamic and uncertain resource demands of VMs. In a data center, the highly efficient VM consolidation method attempts to achieve an appropriate balance among reducing energy consumption, optimizing resource utilization, and guaranteeing QoS, which is a multi-objective optimization problem with multiple resource constraints.

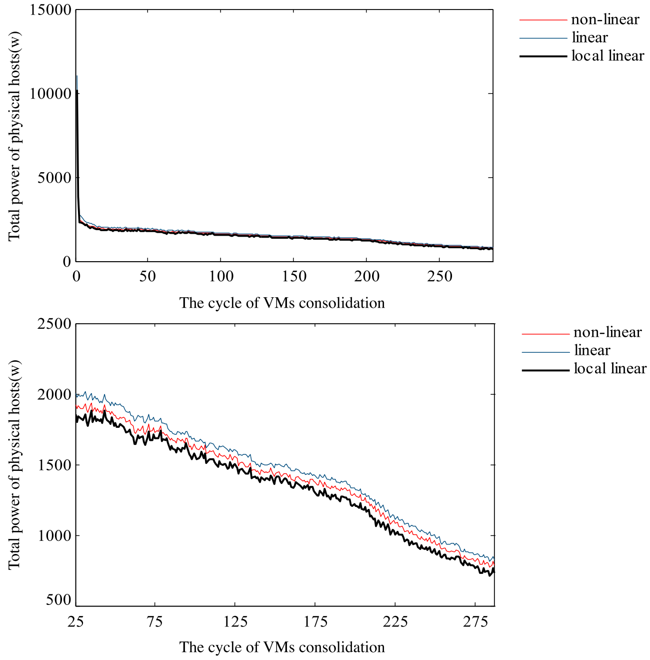


Fig. 12. Total power of physical hosts varying with the cycle of VMs consolidation.

This study focuses on the energy efficiency optimization of data centers, the reduction of the number of VM migrations, the uncertain characteristics of various resource demands (e.g., CPU, memory, and network bandwidth), and the essential requirements of global optimization for VM consolidation. By reliably estimating the energy consumption of data centers, the number of VM migrations and the probability of multi-resource overloading, as well as the effectively adaptive constraint of resource overloading probability, algorithms for selecting the VMs to be migrated and VM placement, and for determining the underloaded PMs, have been proposed. By taking the mapping relation between PMs and VMs as the food source, the proposed sub-algorithms integrate and cooperate to simulate the artificial bee colony foraging behaviors; by using both the searching mechanism and optimization strategy of ABC algorithm, the optimum mapping relation with multi-resource constraints between PMs and VMs is obtained globally. As a result, the issue "where from and where to" for live VM migration is naturally resolved. The simulation results and their analytical comparison demonstrate the apparent improvement of EC-VMC in terms of SLATAH, PDM, SLAV, EC, ESV, the number of VM migrations and QoS. The proposed algorithm also achieves an optimum balance among improving energy efficiency, optimizing resource utilization, and guaranteeing QoS. Thus, its effectiveness and efficiency have been validated.

However, there are a few limitations that need to be further addressed in our future works. First, the correspondence relations between bee colony behaviors and various operations in VM consolidation such as host overload detection, VM placement, and VM migration selection need further study to improve the proposed algorithm. Second, determining how to address the issue "where from and where to" for the live VM migration based only on the well-known ABC algorithm or other similar intelligence algorithms should be given priority. Finally, the current study only focuses on VM consolidation in a single data center. Thus, the consolidation of VMs among data centers should be studied to improve the QoS and energy efficiency and to adapt to the increased demands of virtual resource management in large-scale data centers.

Acknowledgments

This work is supported by the Future Research Projects Funds for the Science and Technology Department of Jiangsu Province (Grant No. BY2013015-23) and the Fundamental Research Funds for the Ministry of Education (Grant No. JUSRP211A 41).

Appendix. Proof of Theorem 1

Proof. Given $\forall z \geq y \geq 0$, we have, according to (17),

$$\frac{e(z+d)-e(z)}{d} \leq \frac{e(y+d)-e(y)}{d}, z \geq y \geq 0 \tag{a}$$

Using $y+d \geq y = (y-d)+d$ and (17) gives

$$\frac{e(y+d)-e(y)}{d} \leq \frac{e((y-d)+d)-e(y-d)}{d} \tag{b}$$

Combining (a) and (b) yields

$$\frac{e(z+d)-e(z)}{d} \leq \frac{e(y+d)-e(y)}{d} \leq \frac{e((y-d)+d)-e(y-d)}{d}, z \geq y \geq (y-d) \geq 0 \tag{c}$$

We can then obtain, using (c),

$$e(z+d)-e(z) \leq e((y-d)+d)-e(y-d) = e(y)-e(y-d), \quad z \geq y \geq (y-d) \geq 0 \tag{d}$$

Equivalently, $e(z+d)+e(y-d) \leq e(z)+e(y), z \geq y \geq 0$. This completes the proof of (18).

References

[1] M.F. Li, J.P. Bi, Z.C. Li, Resource-scheduling-waiting-aware virtual machine consolidation, J. Softw. 25 (7) (2014) 1388–1402 (in Chinese).

[2] R. Birke, L.Y. Chen, E. Smirni, Data centers in the cloud: A large scale performance study, in: Proceedings of the 5th IEEE International Conference on Cloud Computing, IEEE, 2012, pp. 336–343.

[3] A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Gener. Comput. Syst. 28 (5) (2012) 755–768.

[4] A. Gandhi, M. Harchol-Balter, R. Das, C. Lefurgy, Optimal power allocation in server farm, SIGMETRICS Perform. Eval. Rev. 37 (1) (2009) 157–168.

[5] S. Srikantaiah, A. Kansal, F. Zhao, Energy aware consolidation for cloud computing, Clust. Comput. 12 (2009) 1–15.

[6] A. Verma, P. Ahuja, A. Neogi, pMapper: power and migration cost aware application placement in virtualized systems, in: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, Springer, 2008, pp. 243–264.

[7] A. Verma, G. Dasgupta, T.K. Nayak, P. De, R. Kothari, Server workload analysis for power minimization using consolidation, in: Proceedings of the 2009 Conference on USENIX Annual Technical Conference, USENIX Association, 2009, pp. 28–28.

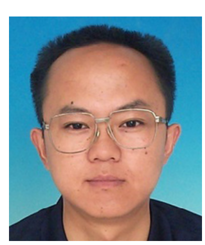
[8] D.G. Lago, E.R.M. Madeira, L.F. Bittencourt, Power-aware virtual machine scheduling on clouds using active cooling control and dvfs, in: Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and E-Science, ACM, 2011 p. 2(1–6).

[9] M. Guazzone, C. Anglano, M. Canonico, Exploiting VM migration for the automated power and performance management of green cloud computing systems, in: Proceedings of the International Workshop on Energy Efficient Data Centers, Springer Berlin Heidelberg, 2012, pp. 81–92.

[10] E. Feller, L. Rilling, C. Morin, Energy-aware ant colony based workload placement in clouds, in: Proceedings of the IEEE/ACM International Conference on Grid Computing, IEEE, 2011, pp. 26–33.

[11] F. Hermenier, X. Lorca, J.M. Menaud, G. Muller, J. Lawall, Entropy: a consolidation manager for clusters, in: Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, ACM, 2009, pp. 41–50.

[12] F. Farahnakian, P. Liljeberg, J. Plosila, LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers, in: Proceedings of the 39th Euromicro Conference on Software Engineering and Advanced Applications, IEEE, 2013, pp. 357–364.
[13] Zhihua Li, Chengyu Yan, Xinrong Yu, Ning Yu, Bayesian network-based virtual machines consolidation method, Future Gener. Comput. Syst. 69 (2017) 75–87.
[14] M.H. Ferdaus, M. Murshed, R.N. Calheiros, R. Buyya, Virtual machine consolidation in cloud data centers using ACO metaheuristic, in: Proceedings of the European Conference on Parallel Processing, Springer International Publishing, 2014, pp. 306–317.
[15] M. Mishra, A. Sahoo, On theory of vm placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach, in: Proceedings of the IEEE International Conference on Cloud Computing, IEEE, 2011, pp. 275–282.
[16] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. J. 8 (1) (2008) 687–697.
[17] Z.Á. Mann, Rigorous results on the effectiveness of some heuristics for the consolidation of virtual machines in a cloud data center, Future Gener. Comput. Syst. 51 (2015) 1–6.
[18] N.T. Hieu, M. Di Francesco, A. Ylä-Jääski, Virtual machine consolidation with usage prediction for energy-efficient cloud data centers, in: Proceedings of the 8th International Conference on Cloud Computing, IEEE, 2015, pp. 750–757.
[19] M.A. Kaaouache, S. Bouamama, Solving bin packing problem with a hybrid genetic algorithm for VM placement in cloud, Proc. Comput. Sci. 60 (2015) 1061–1069.
[20] M.R. Chowdhury, M.R. Mahmud, R.M. Rahman, Implementation and performance analysis of various VM placement strategies in CloudSim, J. Cloud Comput. 4 (1) (2015) 1–21.
[21] Y. Gao, H. Guan, Z. Qi, Y. Hou, L. Liu, A multi-objective ant colony system algorithm for virtual machine placement in cloud computing, J. Comput. Syst. Sci. 79 (8) (2013) 1230–1242.
[22] A. Beloglazov, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurr. Comput. Pract. Exp. 24 (13) (2012) 1397–1420.
[23] S.B. Shaw, A.K. Singh, Use of proactive and reactive hotspot detection technique to reduce the number of virtual machine migration and energy consumption in cloud data center, Comput. Electr. Eng. 47 (2015) 241–254.
[24] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, Energy aware consolidation algorithm based on K-nearest neighbor regression for cloud data centers, in: Proceedings of the 6th International Conference on Utility and Cloud Computing, IEEE/ACM, 2013, pp. 256–259.
[25] S.S. Masoumzadeh, H. Hlavacs, An intelligent and adaptive threshold-based schema for energy and performance efficient dynamic VM consolidation, in: Proceedings of the European Conference on Energy Efficiency in Large Scale Distributed Systems, Springer, Berlin Heidelberg, 2013, pp. 85–97.
[26] S.S. Masoumzadeh, H. Hlavacs, Dynamic virtual machine consolidation: A multi agent learning approach, in: Proceedings of the IEEE International Conference on Autonomic Computing, IEEE, 2015, pp. 161–162.
[27] S.S. Masoumzadeh, H. Hlavacs, A cooperative multi agent learning approach to manage physical host nodes for dynamic consolidation of virtual machines, in: Proceedings of the IEEE Fourth Symposium on Network Cloud Computing and Applications, IEEE, 2015, pp. 43–50.
[28] L. Chen, H. Shen, K. Sapra, Distributed autonomous virtual resource management in datacenters using finite-markov decision process, in: Proceedings of the ACM Symposium on Cloud Computing, ACM, 2014, pp. 1–13.
[29] S. Sohrabi, I. Moser, The effects of hotspot detection and virtual machine migration policies on energy consumption and service levels in the cloud, Proc. Comput. Sci. 51 (2015) 2794–2798.
[30] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, H. Tenhunen, Utilization prediction aware VM consolidation approach for green cloud computing, in: Proceedings of the IEEE 8th International Conference on Cloud Computing, IEEE, 2015, pp. 381–388.
[31] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, H. Tenhunen, Using ant colony system to consolidate VMs for green cloud computing, IEEE Trans. Serv. Comput. 8 (2) (2015) 187–198.
[32] J.A. Aroca, A.F. Anta, M.A. Mosteiro, C. Thraves, L. Wang, Power-efficient assignment of virtual machines to physical machines, Future Gener. Comput. Syst. 54 (2016) 82–94.
[33] Zhangjin Li, Jidong Ge, Chuanyi Li, Hongji Yang, Haiyang Hu, Bin Luo, Victor Chang, Energy cost minimization with job security guarantee in Internet data center, Future Gener. Comput. Syst. 73 (2017) 63–78.
[34] Victor Chang, Towards a big data system disaster recovery in a private cloud, Ad Hoc Netw. 35 (2015) 65–82.
[35] Shahid Anwar, Zukira Inayat, Mohanmad Fadi ZoIkipli, Jasni Mohamad Zain, Abdullah Gani, Nor Badrul Anuar, Mahammad khurram Khan, Cross-VM cache-based side channel attacks and proposed prevention mechanisms: A survey, Future Gener. Comput. Syst. 93 (2017) 259–279.
[36] T. Baker, M. Asim, H. Tawfik, et al., An energy-aware service composition algorithm for multiple cloud-based iot applications, J. Netw. Comput. Appl. (2017).
[37] T. Baker, Y. Ngoko, R. Tolosana-Calasanz, et al., Energy efficient cloud computing environment via autonomic meta-director framework, in: Developments in ESystems Engineering (DeSE), 2013 Sixth International Conference on, IEEE, 2013, pp. 198–203.
[38] First Quarter 2011 SPECpower_ssj2008 Results, 2011. URL: https://www.spec.org/power_ssj2008/results/res2011q1/.
[39] M. Wang, X. Meng, L. Zhang, Consolidating virtual machines with dynamic bandwidth demand in data centers, in: Proceedings of the IEEE INFOCOM, IEEE, 2011, pp. 71–75.
[40] M. Chen, H. Zhang, Y.Y. Su, X. Wang, G. Jiang, K. Yoshihira, Effective VM sizing in virtualized data centers, in: Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management and Workshops, IEEE, 2011, pp. 594–601.
[41] H. Lin, X. Qi, S. Yang, S. Midkiff, Workload-driven VM consolidation in cloud data centers, in: Proceedings of the IEEE International Parallel and Distributed Processing Symposium, IEEE, 2015, pp. 207–216.
[42] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang, Y. Pan, Stochastic load balancing for virtual resource management in datacenters, IEEE Trans. Cloud Comput. 99 (PP), (2016), 1-1. http://dx.doi.org/10.1109/TCC.2016.2525984.
[43] K.S. Park, V.S. Pai, CoMon: a mostly-scalable monitoring system for PlanetLab, Oper. Syst. Rev. 40 (1) (2006) 65–74.
[44] Z. Li, G. Wu, Optimizing VM live migration strategy based on migration time cost modeling, in: Proceedings of the 2016 Symposium on Architectures for Networking and Communications Systems, ACM, 2016, pp. 99–109.
[45] H. Liu, H. Jin, C.Z. Xu, et al., Performance and energy modeling for live migration of virtual machines, Clust. Comput. 16 (2) (2013) 249–264.
[46] R.N. Calheiros, R. Ranjan, A. Beloglazov, César A.F. De Rose, R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Softw. Pract. Exp. 41 (1) (2011) 23–50.
[47] http://www.amazonaws.cn/en/ec2/details/.
[48] S. Shen, V.V. Beek, A. Iosup, Statistical characterization of business-critical workloads hosted in cloud datacenters, in: Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE, 2015, pp. 465–474.
[49] C.H. Hsu, S.W. Poole, Power signature analysis of the SPECpower_ssj2008 benchmark, in: IEEE International Symposium on PERFORMANCE Analysis of Systems and Software, IEEE, 2011, pp. 227–236.



Zhihua Li holds a Diploma and Ph.D. in Computer Science from Jiangnan University, Jiangsu, Wuxi, China in 2009. His research interests include network technology, parallel/distributed computing, information security, data mining, pattern recognition. He is currently faculty of the Dept. of Computer Science and Technology at the Jiangnan University.

Chengyu Yan is a M.S. student in the Dept. of Computer Science and Technology at the Jiangnan University, Jiangsu, Wuxi, China. His current research interests include cloud computing, grid computing, distributed computing.

Lei Yu received his B.S. degree and M.S. degree in computer science from Harbin Institute of Technology, China. He is currently working towards the Ph.D. degree in the School of Computer Science at Georgia Institute of Technology, USA. His research interests include cloud computing, data privacy, and sensor networks, wireless networks, and network security.

Xinrong Yu is a M.S. student in the Dept. of Computer Science and Technology at the Jiangnan University, Jiangsu, Wuxi, China. His research interests include cloud computing, parallel computing, distributed computing.