IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 17, 2019, accepted April 15, 2019, date of publication April 23, 2019, date of current version May 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2912722

Multi-Objective VM Consolidation Based on Thresholds and Ant Colony System in Cloud Computing

HUI XIAO[1], ZHIGANG HU[1], AND KEQIN LI[2], (Fellow, IEEE)
[1]School of Computer Science and Engineering, Central South University, Changsha 410083, China
[2]Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

Corresponding author: Zhigang Hu (zghu@csu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61572525.

ABSTRACT With the large-scale deployment of cloud datacenters, high energy consumption and serious service level agreement (SLA) violations in datacenters have become an increasingly urgent problem to be addressed. Implementing an effective virtual machine (VM) consolidation methods is of great significance to reduce energy consumption and SLA violations. The VM consolidation problem is a well-known NP-hard problem. Meanwhile, efficient VM consolidation should consider multiple factors synthetically, including quality of service, energy consumption, and migration overhead, which is a multi-objective optimization problem. To solve the problem above, we propose a new multi-objective VM consolidation approach based on double thresholds and ant colony system (ACS). The proposed approach leverages double thresholds of CPU utilization to identify the host load status, VM consolidation is triggered when the host is overloaded or underloaded. During consolidation, the approach selects migration VMs and destination hosts simultaneously based on ACS, utilizing diverse selection policies according to the host load status. The extensive experiment is conducted to compare our proposed approach with the state-of-art VM consolidation approaches. The experimental results demonstrate that the proposed approach remarkably reduces energy consumption and optimizes SLA violation rates thus achieving better comprehensive performance.

INDEX TERMS Ant colony system, double thresholds, energy consumption, quality of service, VM consolidation.

I. INTRODUCTION

Cloud computing provides access to virtualized cloud resources as a service to users over Internet in an on-demand and pay-per-use style [1]. With the maturity of cloud computing business models and technology architectures, the number of cloud users has increased significantly. New datacenters, servers, and cooling equipment have been added to meet the increasing needs, generating high operational costs and carbon dioxide emissions. The minimization of datacenter energy consumption has become a critical challenge [2]. At the same time, taking into account the users' requirements for cloud service performance, the quality of service (QoS) [3] defined in service level agreement (SLA) needs to be met to avoid bringing unpredictable losses to users. Therefore, it is an urgent problem to be solved for the development of cloud computing to provide users with the desired QoS while reducing the energy consumption of datacenters.

The implementation of virtualization technology [4] in cloud computing enables hosts to share physical resources and offer users services flexibly by creating multiple VMs. Dynamic VM consolidation [5] periodically adjusts the current mapping relation between VMs and hosts according to time-varying resource requirements by migrating VMs between hosts to fully and evenly utilize computing resources. For example, migrating some VMs from an overloaded host targets at reducing SLA violation, and for an underloaded host, all VMs on it should be migrated away, then it is switched into sleep state to avoid energy waste. Obtaining the optimal mapping relation is beneficial to optimize the resource utilization, improve QoS, and reduce energy consumption.

The techniques addressing the VM consolidation problem primarily include heuristic greedy algorithms, constrained

The associate editor coordinating the review of this manuscript and approving it for publication was Mianxiong Dong.

programming, and meta-heuristic algorithms. Greedy algorithms are widely used for dynamic VM consolidation because of the low time complexity and simplicity to implement. However, traditional greedy approaches are easy to fall into local optimal solutions and miss the optimal solution. Constrained programming techniques can achieve the optimal solution, but they cannot be well extended to large datacenters with the limitation of the problem size. In recent years, researchers have proposed many bio-inspired meta-heuristic consolidation algorithms, such as Ant Colony Optimization (ACO) algorithm, Genetic Algorithm, Artificial Bee Colony (ABC) algorithm, which are effective in solving large-scale problems and avoiding local optimal solutions. Ant Colony System (ACS) [6]–[8], a kind of ACO algorithm, finds the near-optimal solution in polynomial time complexity through probabilistic search in the solution space, which has attracted more and more attention for the excellent performance in solving NP-hard problems and combinatorial optimization problems.

As far as we are concerned, most of the existing VM consolidation approaches only focus on saving energy consumption of cloud datacenters. However, SLA violation should also be considered to satisfy the QoS delivered by the cloud system. Noteworthily, VM consolidation can decrease energy consumption by consolidating VMs into a reduced number of hosts, but excessive consolidation might degrade system performance and lead to SLA violations [9]. Therefore, the optimal VM consolidation approach should strike a balance between energy consumption and QoS. In addition, the VM migration incurs additional workload and increases energy consumption. The service downtime caused by migration likely affects QoS. Hence, VM consolidation should trigger as few VM migrations as possible to minimize the consequent negative influence.

In this paper, we propose a multi-objective VM consolidation approach based on double CPU utilization thresholds [10] and ACS, called DA-VMC. In ACS, a number of artificial ants build solutions to the related optimization problem in parallel. They exchange quality information of these solutions via pheromone to find the optimal solution. Taking advantage of these characteristics of ACS, our approach exploits artificial ant colony to seek the optimal mapping relation between VMs and hosts by assuming that a mapping relation between VMs and hosts is a food source. The main contributions of this paper are summarized as follows.

- First, we abstract the VM consolidation problem as a multi-objective combinatorial optimization problem optimizing three conflicting objectives, including reducing energy consumption, ensuring QoS requirements, and reducing the number of migrations.
- Then, we employ double thresholds of CPU utilization to determine migration time. Based on double thresholds, the host load status is judged as overloaded, normal-loaded, or underloaded. In order to optimize QoS and energy consumption of datacenters, overloaded hosts and underloaded hosts will perform VM consolidation successively.
- Next, we apply ACS in a multi-stage VM consolidation, in which several selection policies corresponding to different host status are used to select migration VM and destination host. The problem of migration VM selection and destination host selection for VM consolidation is globally studied and optimized.
- At last, we evaluate the proposed DA-VMC approach by using CloudSim platform on real workload. The experimental results demonstrate that DA-VMC possesses an obvious advantage in the aspect of reducing energy consumption, SLA violations, and VM migrations.

The rest of the paper is organized as follows. Section 2 introduces the related work of VM consolidation. In Section 3, we introduce how to build the VM consolidation problem into a multi-objective combinatorial optimization problem. Section 4 proposes our VM consolidation approach based on double thresholds and ACS. Section 5 presents experiments results and performance evaluation; Section 6 concludes the paper and discusses our future work.

II. RELATED WORK

There are three main problems to be solved in VM consolidation [11]–[13]. First, VM consolidation time selection, to determine when VMs should be consolidated. Second, there is a need to select which VMs should be migrated, that is, migration VM selection. Third, VM deployment, to deploy those selected migration VMs. Depending on specific problems of VM consolidation, researchers have proposed a variety of different approaches.

A. CONSOLIDATION TIME SELECTION

The workload in the datacenter fluctuates in real time, which affects the resource utilization and energy efficiency of hosts [14]. Most of the existing studies determined the consolidation time on the basis of host resource utilization. VM consolidation is usually carried out for overloaded hosts and underloaded hosts to reduce energy consumption and improve QoS [15], [16].

Chen *et al.* [17] presented a method based on the sliding window concept to perform consolidation operation when host resource utilization, sampled at regular intervals and recorded in windows, exceeds the pre-defined high resource utilization threshold continuously. The consolidation action is also triggered if the host CPU utilization is lower than the underload threshold. Minarolli *et al.* [18] proposed a long-term forecast of VM resource demands based on Gaussian processes to detect when a host is overloaded or underloaded. A decision-theoretic approach using utility function was applied to execute migration decision considering live migration overheads.

To adapt to the dynamic variation of the workload in the datacenter, some approaches dynamically adjust the thresholds of CPU utilization to guide VM consolidation.

Masoumzadeh and Hlavacs [19] presented an adaptive threshold based approach using fuzzy Q-learning for host overloading detection. The fuzzy Q-learning technique learns the historical data of overload thresholds in different datacenter states, and yields the appropriate threshold for workload data input from hosts according to the online decision model. Zhou *et al.* [20] proposed an adaptive three-threshold framework based on K-means clustering algorithm. The hosts in the datacenter are divided into four categories: less loaded hosts, little loaded hosts, normal-loaded hosts, and overloaded hosts. Salimian *et al.* [21] proposed an adaptive algorithm based on fuzzy threshold to detect overloaded and underloaded hosts. The information of host resource usage is applied to the fuzzy inference engine to estimate the numerical value of the CPU utilization thresholds.

Unlike traditional CPU utilization threshold frameworks, Fard *et al.* [22] employed the temperature threshold metric to conduct a thermal-aware consolidation. The temperature threshold is set as the temperature of the optimum host at 90% CPU utilization. For the other host, if its temperature exceeds the temperature threshold, it is identified as overloaded. For underloaded hosts, the consolidation is preferred to the host with most energy consumption but lowest operation per second. Abdelsamea *et al.* [23] proposed a multivariate regression method employing hybrid resource parameters including CPU, RAM, and bandwidth utilization to predict host resource utilization and detect overloaded hosts. However, the coefficients of the regression model are trained once per host, which adds complexity to the prediction.

B. MIGRATION VM SELECTION

VM consolidation adopts different selection strategies to select VMs for migration from hosts of undesirable workload status, aiming at different targets, such as optimizing resource utilization, improving QoS or reducing energy consumption.

Focused on the VM selection problem, Beloglazov *et al.* [10] presented three VM selection policies. The minimization of migrations (MM) policy selects the minimum number of VMs to migrate for overloaded hosts to decrease the CPU utilization. The highest potential growth (HPG) policy migrates the VM with the lowest usage of the CPU resource each time to minimize the potential enhancement of hosts' CPU utilization and prevent secondary SLA violations. The Random Selection (RC) policy randomly selects VMs for migration based on a uniformly distributed discrete random variable until the overload status of the host is eliminated. Further, Beloglazov and Buyya [24] presented two other VM selection policies. The minimum migration time (MMT) policy selects VMs requiring the shortest migration time for migration. The idea of the Maximum Correlation (MC) policy is that the higher the resource utilization correlation between VMs on the same host, the more likely the host overload, hence the VM having the highest correlation of CPU utilization with other VMs is selected for migration to reduce the risk of host overload. Cao and Dong [25] proposed the minimum utilization(MU) policy to select the VM with the lowest CPU utilization for migration. Masoumzadeh and Hlavacs [26] proposed a maximum utilization (MaxU) policy which selects the VM with the highest CPU utilization to migrate. In contrast to the MU policy, the MaxU policy eliminates the host overload as quickly as possible, whereas greatly increasing migration overheads.

Different from the above works which select migration VM based mainly on CPU utilization, Shidik *et al.* [27] proposed a VM selection policy based on RAM and CPU utilization. Fuzzy logic is employed to categorize both resource attributes of VM candidates and the Markov normal algorithm is used to select the migration VM according to the categorical attributes. Li *et al.* [28] selected VMs to migrate by utilizing the content similarity among the VM memory. The migration VM selection favors the VMs with the highest memory content similarity from different overloaded hosts to reduce the amount and time of data transfer during VM migration. Laili *et al.* [29] presented an iterative budget algorithm taking into account various resources including CPU, memory, disk and network. The algorithm builds a reverse selection mechanism that finds the most suitable VM from candidate VM sets for each randomly selected target host.

C. VM DEPLOYMENT

VM consolidation deploys the migrated VMs on a more suitable destination host to dynamically optimize the mapping relation between VMs and hosts in the cloud datacenter.

Mishra and Sahoo [30] studied how to use heuristic greedy algorithms, such as BFD (Best Fit Deceasing) and FFD (First Fit Deceasing), to obtain a quasi-optimal VM deployment solution. Murtazaev and Oh [31] proposed the Sercon algorithm which inherits some properties of FF (First Fit) and BF (Best Fit) to minimize the number of hosts and migrations. Farahnakian *et al.* [32] abstracted the VM consolidation problem into a multi-objective vector packing problem, aiming to reduce energy consumption, minimize migrations and avoid SLA violation. The target host is selected according to the current and future resource utilization of hosts and VMs. The correlation-based strategy of VM consolidation [33] consolidates VMs with inter-traffic as closely as possible to minimize the network traffic and alleviate network pressure, but it also increases the traffic load of the virtual switches on hosts. To address this problem, Li *et al.* [34] proposed virtual-switching-aware BFD(VSA-BFD) and virtual-switching-aware FFD(VSA-FFD) with comprehensive consideration of the traffic between VMs and the CPU overhead generated by virtual switches.

The greedy algorithm has low complexity, but it cannot guarantee to find the optimal solution. Chen *et al.* [17] formulated the VM deployment problem as a multi-criteria decision-making problem. The TOPSIS solution is applied to choose appropriate destination hosts. Huang and Tsang [35]

developed non-linear programming and proposed a distributed framework to automate VM consolidation. Based on the m-convex optimization theorem, the optimal VM deployment solution is obtained in an incremental manner. However, limited by the size and complexity of the problem, non-linear programming is incapable to scale well to big datacenters.

With the ability to handle the large-scale problems efficiently and avoid suboptimal solutions, the meta-heuristic algorithm is helpful to make up for the deficiency of the greedy algorithm and constrained programming. Li *et al.* [36] constructed the VM consolidation problem as a multi-objective optimization problem with multiple resource constraints, and simulated the artificial bee colony foraging behavior to search for the optimal mapping relation between VMs and hosts. Mosa and Paton [37] adopted the utility function considering income, energy cost, and violation costs to calculate the profit of VM deployment solutions. The genetic algorithm is used to search candidate deployments to maximize the utility function. Li *et al.* [38] proposed an VM reallocation algorithm based on the adaptive particle swarm optimization to achieve minimal energy consumption, where QoS is ensured by applying multi-resource utilization thresholds.

Focused on balancing the usage of various resources in hosts, Ferdaus *et al.* [39] proposed a vector algebra-based ACO algorithm for searching the optimal VM deployment. However, the algorithm reallocates all VMs during each VM consolidation, leading to high time complexity. Farahnakian *et al.* [40] proposed a VM consolidation method leveraging ACS to deploy VMs into the minimal amount of running hosts, based on the objective function defined with the number of sleep hosts and VM migrations. However, the energy consumption generated by VM migrations is not considered. Aryania *et al.* [41] extended the work of Farahnakian *et al.* [40] by considering the energy consumption of both running hosts and VM migrations, which obtains a better energy saving effect.

It can be known from above studies that most of them conduct research on VM consolidation from various optimization perspectives such as energy efficiency, resource usage balance, VM migration overhead. Accordingly, we construct the VM consolidation problem as a multi-objective combinatorial optimization problem taking into account multiple influential factors of VM consolidation. To figure out the problem, we propose a dynamic VM consolidation approach DA-VMC. Static double thresholds of CPU utilization are applied in the decision-making process of the consolidation time to reduce computational overhead and avoid the system instability. Assuming the mapping relation between VMs and hosts as a food source, we employ ACS which simulates the artificial ant colony foraging behavior to simultaneously select the migration VM and the destination host based on the specified objective function. After several rounds of searches and information exchanges, the ant colony acquires the near-optimal solution.

III. DYNAMIC VM CONSOLIDATION BASED ON MULTI-OBJECTIVE COMBINATORIAL OPTIMIZATION

A. DATACENTER MODEL

There exist many hosts with different configurations in the datacenter. VMs are deployed to the appropriate hosts according to their respective resource requirements. Assume that the host set in the datacenter is $H = \{h_1, h_2, \ldots, h_j, \ldots, h_m\}$, where m is the number of hosts; The VM set is denoted as $H = \{v_1, v_2, \ldots, v_i, \ldots, v_n\}$, where n is defined as the number of VMs. The variable $x_{ij} \in \{0, 1\}$ indicates whether the VM v_i is assigned to the host h_j. $x_{ij} = 1$ if assigned; otherwise $x_{ij} = 0$. The matrix $X = [x_{i,j}]_{n\times m}$ describes the mapping relation between the VMs and hosts. The host deploying v_i is denoted as $h(v_i)$, and $V(h_j)$ is the set of VMs deployed on the host h_j. VR_i is the demand of the VM v_i for the CPU resource. PR_j is the total CPU resource demands of the host h_j, which is expressed as

$$PR_j = \sum_{i=1}^{N} (x_{ij} \cdot VR_i). \tag{1}$$

CR_j is the CPU resource capacity of the host h_j. The actual utilization of CPU resource in the host h_j, denoted as U_j, is calculated with

$$U_j = \frac{PR_j}{CR_j}. \tag{2}$$

B. ENERGY CONSUMPTION MODEL

In general, the host has three states, i.e., running state, sleep state, and shutdown state. Chou *et al.* [42] suggested that different states of the host result in different power consumption levels. The energy consumption of hosts in the running state, denoted as HP_j^{on}, is highly positively correlated with the CPU utilization [43]. Compared to the running state, the host consumes only a small amount of energy in the sleep state, denoted as HP_j^{sp}, and consumes no energy in the shutdown state. Besides, switching back and forth between different states consumes energy and time. It takes a long time to restart the host from the shutdown state to the running state, which may cause degradation of service quality. Therefore, in this paper, VM consolidation is only considered in hosts that are running or sleeping.

Based on the above analysis and assumption, the following energy consumption model for VM consolidation is established with comprehensive consideration of the energy consumption generated by both host operation and state switching:

$$EC_j = \int_t \left[s_j^{on} \cdot HP_j^{on} + \left(1 - s_j^{on}\right) \cdot HP_j^{sp} \right] dt + s_j^{gs} \cdot ES_j^{gs} + s_j^{wa} \cdot ES_j^{wa}, \tag{3}$$

where $s_j^{on} \in \{0, 1\}$ represents the status of the host, $s_j^{on} = 1$ denotes that the host is running, $s_j^{on} = 0$ denotes that the host is in the sleep state. $s_j^{gs} \in \{0, 1\}$, s_j^{gs} is set to 1 when switching the host h_j from the running state to the sleep state, otherwise $s_j^{gs} = 0$. $s_j^{wa} \in \{0, 1\}$, s_j^{wa} is set to 1 when h_j is

waked up from sleep, otherwise $s_j^{wa} = 0$. ES_j^{gs} indicates the energy consumption of state switching from running to sleep, and ES_j^{wa} indicates that from sleep to running.

The total energy consumption of the datacenter can be calculated as

$$EC = \sum_{j=1}^{m} EC_j. \tag{4}$$

C. MULTI-OBJECTIVE COMBINATORIAL OPTIMIZATION

An efficient VM consolidation approach needs to develop corresponding strategies of resource management and scheduling for multiples goals of reducing energy consumption, improving QoS, and facilitating load balancing, so that the VM migration can be triggered under specific conditions to optimize the mapping relation between VMs and hosts. Accordingly, we take into account multiple factors affecting VM consolidation and define the following VM consolidation optimization objectives.

In order to reduce energy consumption and save energy, the total power consumption of the datacenter EC should be minimized, i.e., min (EC).

There exists resource competition between VMs on the same host. When the resource competition becomes more intense, especially when the resource utilization approaches or exceeds a certain threshold, the performance of the host turns worse, and the QoS level is more likely to decline [44]. Apparently, the QoS level of the host has a great relationship with the resource utilization. Hence, for purpose of optimizing the QoS more conveniently and effectively, we ensure the host QoS by limiting the host CPU resource utilization below the overload threshold Thr_o, as shown below:

$$U_j < Thr_o, \quad \forall j \in \{1, 2, \ldots, m\}. \tag{5}$$

Moreover, VM migrations due to consolidating VMs consume energy and computing resources, incurring performance interference and cost on both source and destination hosts [45]. Data transferred during migration burden network traffic, which inevitably interferes with the other VMs in the same datacenter. Thus minimizing the number of VM migrations, denoted as MG, is very necessary, i.e., min (MG).

Finally, the above objectives are integrated as a minimum combinatorial optimization problem with the resource constraint via a linear weighting method:

$$\begin{aligned} &\min(F) = \min\left(EC + \omega_{mg} \cdot MG\right) \\ &s.t.\ U_j < Thr_o, \forall j \in \{1, 2, \ldots, m\}, \end{aligned} \tag{6}$$

where ω_{mg} represents the weight of migrations with respect to the total energy consumption.

The optimization problem defined in (6) is studied and solved in Section IV.

IV. DA-VMC CONSOLIDATION APPROACH

The dynamic VM consolidation enables VMs to be dynamically migrated between hosts to accommodate the changeable workload in the datacenter. In VM consolidation, there are several key problems that must be addressed. For example, what kind of condition will trigger VM consolidation (consolidation time selection); which VMs should be migrated to achieve the best energy consumption and QoS (migration VM selection); Which destination hosts should be choosed to redeploy the selected migration VMs (destination host selection).

In order to solve the above problems, we propose a VM consolidation approach based on double thresholds and ACS. Based on the strategy of static double thresholds, the proposed DA-VMC approach sets the host CPU utilization upper threshold Thr_o and lower threshold Thr_u ($0 \leq Thr_u < Thr_o \leq 1$) to decide the consolidation time. The defined static thresholds are used to identify the load status of hosts, dividing all hosts into three sets: the overloaded host set H_o, the normal-loaded host set H_n, the underloaded host set H_u. For the host h_j, the host is overloaded if its CPU utilization $Thr_o \leq U_j < 1$; the host is normal-loaded if $Thr_u \leq U_j < Thr_o$; the host is underloaded if $0 < U_j < Thr_u$. VM consolidation is triggered when the host is overloaded or underloaded. According to the study by Beloglazov *et al.* [10] and experimental verification, the algorithm can obtain excellent performance when the upper threshold is set to 80% and the lower threshold is 40%.

In the datacenter, each host deploys one or more VMs. In this paper, we assume that all the VMs deployed on a host may be selected for migration, in which case the host is the source host of the migrated VMs. Likewise, a migrated VM may be redeployed to any other host, namely all hosts except the source host are its potential destination hosts. Accordingly, a tuple set of mapping relations T is defined as $T = \{(v_m, h_d)\}$, where each tuple consists of two elements: the VM to be migrated v_m and the destination host h_d. By treating the tuples in T as the ant food, the DA-VMC approach leverages ACS to search tuples in T to update the mapping relation between VMs and hosts.

The complexity of the consolidation approach depends primarily on the number of tuples in T. To reduce the time complexity of the approach, we apply a multi-stage consolidation which limits the number of tuples in T at each stage. On the one hand, VM consolidation is firstly performed for overloaded hosts, and then for underloaded hosts. On the other hand, when selecting the destination host, in order to minimize the number of underloaded hosts to reduce power consumption, the first choice is made in the set of normal-loaded hosts. If it fails, the range of choice turns to the set of underloaded hosts. Hosts in the sleep mode are activated only if the VM cannot be redeployed on an already active PM. In this way can restrict the solution space for each search and improve the efficiency of the ant's search, which helps greatly reduce the computation time of the approach without affecting the quality of the solution. Fig. 1 shows the multi-stage consolidation framework.

During the process of VM consolidation, the DA-VMC approach creates the pheromone information matrix $[\tau_{i,j}]_{n \times m}$

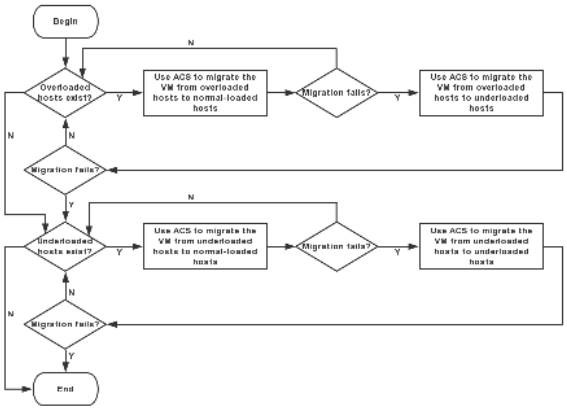

FIGURE 1. Multi-stage consolidation framework.

to save the experience in past searches of ants, where $\tau_{i,j}$ denotes the favorability of selecting the mapping relation tuple (v_i, h_j), namely redeploying the VM v_i on the host h_j. Two rules of local pheromone update and global pheromone update are applied to update the pheromone level of tuples. The heuristic factor is defined to guide the ant colony to select the optimal tuple for minimum power consumption and migration number based on the pseudo-random-proportional rule [6]. In the following, the detailed definition of these factors will be given.

A. PHEROMONE INFORMATION

The ant deposits pheromone on the path to finding food source, and when other ants smell the deposited pheromone, they tend to choose paths with a higher pheromone concentration. The quality of the solution found by ants depends greatly on the definition of the pheromone. Thus, choosing a rational definition of pheromone is very critical. In the initialization phase of the pheromone matrix $[\tau_{i,j}]_{n\times m}$, we employ the solution quality assessment method used by Dorigo and Gambardella [6] to calculate the initial pheromone amount as follows:

$$\tau_0 = \frac{1}{EC_0 \cdot MG}, \tag{7}$$

where EC_0 is the total power consumption of the datacenter in the initial state. MG is the approximate optimal number of migrations, which can be estimated by using the nearest neighborhood heuristic algorithm [6].

For the pheromone update in ACS algorithm, we present the local pheromone update rule and the global pheromone update rule to increase the pheromone amounts corresponding to high quality solutions or decrease those corresponding to the low quality ones. After selecting a new mapping relation tuple (v_i, h_j), the ant updates the pheromone level of this traversed mapping relation using the following local pheromone update rule:

$$\tau_{ij} = (1 - \rho_l) \cdot \tau_{ij} + \rho_l \cdot \tau_0, \tag{8}$$

where $\rho_l \in [0, 1]$ is the local pheromone evaporating parameter. The local pheromone update rule applied to a traversed tuple decreases the tuple's pheromone concentration by a certain level and weakens its attraction to other ants. Hence, the local pheromone update rule can avoid a premature convergence of the ACS algorithm towards a suboptimal solution.

After all the ants complete building solutions, the quality of all current built solutions is evaluated according to the objective function. The following global pheromone update rule is performed to preserve the experience of the global optimal solution:

$$\begin{cases} \tau_{ij} = (1 - \rho_g) \cdot \tau_{ij} + \rho_g \cdot \Delta\tau \\ \Delta\tau = \begin{cases} \dfrac{1}{F(X^+)}, & if\ x_{ij} = 1\ in\ X^+ \\ 0, & otherwise, \end{cases} \end{cases} \tag{9}$$

where $\Delta\tau$ is the amount of additional pheromone increment. $\rho_g \in [0, 1]$ is the global pheromone evaporation parameter. X^+ is the global best solution.

B. HEURISTIC FACTOR

In ACS, the heuristic factor is used in combination with the pheromone to guide the solution construction of ants. The heuristic factor is expressed as $\eta_{i,j}$, indicating the desirability of selecting the tuple (v_i, h_j). Calculated in a problem-specific style, the heuristic factor reduces the blindness of ant searches, which is an important factor affecting the efficiency of ACS.

In order to fully utilize host resources while minimizing degradation of service quality due to resource competition, the proposed approach defines different calculation criterions of the heuristic factor according to different consolidation stages, considering two partial contribution of migration VM selection and destination host selection. The calculation criterion of the heuristic factor $\eta_{i,j}$ for selecting the tuple (v_i, h_j) is defined as

$$\eta_{i,j} = \eta^v\left(h(v_i), -v_i\right) \cdot \eta^h\left(h_j, +v_i\right), \tag{10}$$

where $\eta^v\left(h(v_i), -v_i\right)$ is the heuristic factor that selects the VM v_i for migration from its source host $h(v_i)$, and $\eta^h\left(h_j, +v_i\right)$ is the heuristic factor that redeploys v_i to the destination host h_j. A hybrid heuristic factor is defined with consideration of both migration VM selection and destination host selection.

1) MIGRATION VM SELECTION

Based on the predefined double thresholds, the main idea of VM selection is to control the CPU utilization of hosts between the double thresholds. Inappropriate selection criteria may increase VM migrations, aggravate power consumption, or affect QoS due to long service downtime in migration. Hence, selection criteria for migration VM should be reasonably defined. In the following, we discuss the proposed VM selection policies with corresponding heuristic factor calculation rules.

If the host CPU utilization exceeds the upper threshold, some VMs have to be migrated from the host to reduce the utilization, preserving some free resources to prevent SLA violations. Accordingly, we introduce three VM selection policies for overloaded hosts as follows.

First, the Highest CPU Priority Selection policy (HCPS). The HCPS policy defines that the higher the CPU utilization of a VM, the higher the priority that the VM is selected for migration. For the overloaded host $h_j \in H_o$, the heuristic factor of selecting the VM v_i for migration is calculated as

$$\eta^v\left(h_j, -v_i\right) = \frac{VR_i}{PR_j}. \tag{11}$$

Second, the Minimum CPU Priority Selection policy (LCPS). When the upper threshold is violated by the host h_j, the idea of the LCPS policy is that a VM v_i with lower CPU utilization are selected with higher priority as the migration VM. The heuristic factor is defined as

$$\eta^v\left(h_j, -v_i\right) = 1 - \frac{VR_i}{PR_j}. \tag{12}$$

Third, the Random CPU selection policy (RCS).For the overloaded host $h_j \in H_o$, the RCS Policy randomly selects the VM v_i for migration. All VMs are selected with the same priority, and the corresponding heuristic factor is defined as

$$\eta^v\left(h_j, -v_i\right) = \frac{1}{\left|VM_j\right|}, \tag{13}$$

where $\left|VM_j\right|$ represents the number of VMs deployed on the host h_j.

For a underloaded host, all VMs on this host should be migrated and the host should be switched to the sleep state to avoid the idle power consumption. Thus, with the aim of minimizing invalid migrations and underloaded hosts, the underloaded host should prefer to migrate the VM that significantly reduces its resource utilization after migration. The heuristic factor for selecting the VM v_i to migrate in the underloaded host $h_j \in H_u$ is defined as

$$\eta^v\left(h_j, -v_i\right) = 1 - U_j\left(-v_i\right), \tag{14}$$

where $U_j\left(-v_i\right)$ represents the CPU utilization of the host h_j after migrating the VM v_i.

2) DESTINATION HOST SELECTION

A new destination host should be selected to deploy the selected migration VM. When the host CPU utilization gets closer to the overload threshold, the system QoS drops more rapidly. Therefore, when selecting the destination host in the normal-loaded host set H_n, choosing the host with low resource utilization rate after VM redeployment is advantageous for avoiding the QoS degradation due to the excessive CPU utilization of hosts. The heuristic factor definition of selecting the destination host $h_j \in H_n$ for the VM v_i is computed as

$$\eta^h\left(h_j, +v_i\right) = \begin{cases} 1 - U_j(+v_i), & if\ U_j(+v_i) < Thr_o \\ 0, & otherwise, \end{cases} \tag{15}$$

where $U_j(+v_i)$ is the CPU utilization of the host h_j after deploying the VM v_i. Besides, the CPU resource utilization constraint in the heuristic factor is to prevent migrations that cause the overload of destination hosts.

Underloaded hosts have low resource utilization and weak resource competition, which guarantee QoS but waste energy. Hence, when selecting the destination host h_j in the underloaded host set H_u, a host with higher resource utilization after VM deployment is more preferable, which contributes to the minimization of underloaded hosts. The corresponding heuristic factor is defined as

$$\eta^h\left(h_j, +v_i\right) = \begin{cases} U_j(+v_i), & if\ U_j(+v_i) < Thr_o \\ 0, & otherwise. \end{cases} \tag{16}$$

C. PSEUDO-RANDOM-PROPORTION RULE

Based on the heuristic factor and pheromone information, the ant selects the next tuple for traversal according to the following pseudo-random proportion rule:

$$\left(v_m, h_d\right) = \begin{cases} \arg\max_{\left(v_i, h_j\right) \in \Omega_k} \{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}\}, & if\ q \le q_0 \\ \left(v_s, h_g\right), & otherwise, \end{cases} \tag{17}$$

where α and β are parameters that control the influence of the pheromone and the heuristic factor respectively. Ω_k is the set of tuples currently allowed to be traversed by the ant ant_k. q is a random number uniformly distributed in [0, 1], $q_0 \in [0, 1]$ is a fixed parameter that determines the relative importance of cumulative experience with random selection. (v_s, h_g) is a random tuple variable selected according to the probability distribution given below:

$$p_{md}^k = \begin{cases} \dfrac{\tau_{md}^{\alpha} \cdot \eta_{md}^{\beta}}{\sum_{\left(v_i, h_j\right) \in \Omega_k} \left(\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}\right)}, & if\ \left(v_i, h_j\right) \in \Omega_k \\ 0, & otherwise, \end{cases} \tag{18}$$

where p_{md}^k denotes the probability that the ant ant_k chooses to traverse the tuple (v_m, h_d) in the next step.

The pseudo-random proportional rule favors the tuple $\left(v_i, h_j\right)$ with large heuristic value η_{ij} and high pheromone level τ_{ij}. In each iterative step, if the generated random number q is not greater than q_0, the tuple $\left(v_i, h_j\right)$ with maximum value of $\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}$ in Ω_k is selected according to (17), which helps ants quickly converge to a high quality solution. Otherwise, the tuple is randomly selected in Ω_k in accordance with the probability distribution p_{md}^k in (18), at which time ants conduct a broader search to avoid premature stagnation. Combining multiple heuristic factor calculation criteria and pseudo-random-proportional rule defined above, we propose the MTS (Mapping Relation Tuple Selection) algorithm. The algorithm selects the next tuple of mapping relation (v_m, h_d) based on the load status identifier *heuP* of the source host and destination host. The MTS algorithm pseudocode is as shown in Algorithm 1.

Algorithm 1 Mapping Relation Tuple Selection (MTS)

Input: T, $heuP$
Output: (v_m, h_d)
switch ($heuP$)
// VM from overloaded hosts to normal-loaded hosts.
case "ON"**:**
  Choose a VM selection policy SP from (11), (12), (13)
  Compute $\eta_{md}, \forall (v_m, h_d) \in T$ with SP, (15), (10)
  Break
// VM from overloaded hosts to underloaded hosts.
case "OU"**:**
  Choose a VM selection policy SP from (11), (12), (13)
  Compute $\eta_{md}, \forall (v_m, h_d) \in T$ with SP, (16), (10)
  Break
// VM from underloaded hosts to normal-loaded hosts.
case "UN"**:**
  Compute $\eta_{md}, \forall (v_m, h_d) \in T$ with (14), (15), (10)
  Break
// VM from underloaded hosts to underloaded hosts.
case "UU"**:**
  Compute $\eta_{md}, \forall (v_m, h_d) \in T$ with (14), (16), (10)
  Break
Compute $p_{md}, \forall (v_m, h_d) \in T$ with (18)
Choose $(v_m, h_d) \in T$ with (17)

The pseudocode of the proposed VM consolidation algorithm DA-VMC is shown in Algorithm 2. In the initialization phase, the pheromone matrix is set as τ_0 (line 2).The algorithm iterates over nI times (line 3). In each iteration, nA ants build new mapping relation between VMs and hosts in parallel by sequentially performing VM consolidation for overloaded hosts and underloaded hosts (line 5-31). The ant firstly selects the VM in the overloaded hosts for redeployment (line 5-17). Under the premise of ensuring service performance, VMs in the underloaded hosts are redeployed to achieve energy saving (line 18-30). The local pheromone update rule is applied to each traversed mapping relation (line 16 and 29). After all ants have constructed their solutions, all ant-specific solutions are added to the solution set S (line 31). The global optimal solution X^+ is selected by using the objective function in (6) to evaluate each solution in S (line 33). The global pheromone update rule is performed consequently with X^+ (line 34). Finally, when all the ants have iterated through nI rounds, the algorithm outputs the global optimal solution of the mapping relation X^+.

Algorithm 2 DA-VMC

Input: X
Output: X^+
1: $S \leftarrow \phi, X^+ \leftarrow \phi, X^k \leftarrow \phi, t \leftarrow \phi$
2: Initialize all pheromone values τ_0 with (7)
3: **for** $i \in [1, nI]$ **do**
4: **for** $k \in [1, nA]$ **do**
5: **while** overloaded hosts exist **do**
6: $T_{on} \leftarrow \{(v_m, h_d) \,|h(v_m) \in H_o \wedge h_d \in H_n\}$
7: $t \leftarrow MTS\ (T_{on},\ "ON")$
8: **if** t *is null* **then**
9: $T_{ou} \leftarrow \{(v_m, h_d) \,|h(v_m) \in H_o \wedge h_d \in H_u\}$
10: $t \leftarrow MTS\ (T_{ou},\ "OU")$
11: **if** t *is null* **then**
12: Break
13: **end if**
14: **end if**
15: Update mapping relation matrix X
16: Apply local update rule with (8)
17: **end while**
18: **while** underloaded hosts exist **do**
19: $T_{un} \leftarrow \{(v_m, h_d) \,|h(v_m) \in H_u \wedge h_d \in H_n\}$
20: $t \leftarrow MTS\ (T_{un},\ "UN")$
21: **if** t *is null* **then**
22: $T_{uu} \leftarrow \{(v_m, h_d) \,|h(v_m) \in H_u \wedge h_d \in H_u\}$
23: $t \leftarrow MTS\ (T_{uu},\ "UU")$
24: **if** t *is null* **then**
25: Break
26: **end if**
27: **end if**
28: Update mapping relation matrix X
29: Apply local update rule with (8)
30: **end while**
31: $S \leftarrow S \cup \{X^k\}$
32: **end for**
33: $X^+ \leftarrow \arg\max_{X^k \in S} \{F(X^k)\}$
34: Apply global update rule on X^+ using (9)
35: **end for**

V. PERFORMANCE EVALUATION

A. EXPERIMENT SETUP

In this section, we conduct the simulations to evaluate the performance of our proposed approach using CloudSim [46] as the simulation platform. CloudSim is a discrete event simulator that enables virtual environment modeling and virtual resource management. The simulated cloud datacenter in the experiment consists of 800 physical hosts, half of which is HP ProLiant G4 and the other half is HP ProLiant G5. The host configuration details are listed in Table 1. In addition, four kinds of Amazon EC2 VMs [47] are used in this experiment, whose corresponding characteristics are depicted in Table 2. After creating host instances and VM instances on the CloudSim platform, the VMs are deployed on different hosts through the PABFD method [24]. During each VM consolidation cycle, VM consolidation is performed based on the new workload and resource requirements of hosts and VMs.

The workload employed in the experiment came from the CoMon project, which is responsible for monitoring the operation of the infrastructure in PlanetLab [48]. The data used in the experiment came from more than 1,000 VMs at more than 500 places around the world. The statistical properties

TABLE 1. Host configuration.

Type	CPU Type	Frequency (GHz)	Core	RAM (GB)
HP ProLiant G4	Intel Xeon 3040	1.86	2	4
HP ProLiant G5	Intel Xeon 3075	2.66	2	4

TABLE 2. VM types.

Type	CPU frequency (MIPS)	RAM (GB)
High-CPU medium instance	2500	0.85
Extra large instance	2000	3.75
Small instance	1000	1.70
Micro instance	500	0.61

TABLE 3. The properties of PlanetLab data.

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

TABLE 4. DA-VMC parameters.

ω_{mg}	α	β	q_0	ρ_l	ρ_g	nA	nI
9	1	1.5	0.9	0.1	0.1	10	10

of the data are shown in Table 3. Since the actual workload of the real datacenter is periodic and regular, we extract one-day workload from the workload datasets to conduct experiments. The algorithm related parameter settings are shown in Table 4.

B. EVALUATION INDICES

In cloud datacenter, SLA is used to specify the QoS requirements of the user and the consequences of violation, enabling service providers and users to reach an agreement on the services, priorities and responsibilities. Once an SLA violation happens, the user's interests may not be guaranteed, for which the provider may pay an expensive penalty to the user as compensation. For the better optimization of QoS, Beloglazov and Buyya [24] proposed several methods to measure SLA violations.

When the resource utilization of the host reaches 100%, host overload occurs and the available resources are less than the total resource demand of the VMs, which may result in an SLA violation. *SLAVO* is defined as the proportion of time when the host resource utilization reaches 100% to measure the SLA violation caused by the host overload, as shown below:

$$SLAVO = \frac{1}{m}\sum_{j=1}^{m}\frac{T_j^o}{T_j^a}, \tag{19}$$

where m indicates the number of hosts, T_j^o is the total time that the resource utilization of the host h_j experiences 100%, and T_j^a is the total running time of the host h_j.

VM migrations cause overall performance degradation. *SLAVM* represents the SLA violation due to VM migrations, which is defined as

$$SLAVM = \frac{1}{n}\sum_{i=1}^{n}\frac{C_i^d}{C_i^r}, \tag{20}$$

where n represents the number of VMs. C_i^d denotes the unsatisfied requirement for CPU resources of the VM v_i due to the migration. C_i^r is the total CPU resource requirement during the lifetime of the VM v_i. According to the previous research [24], the overhead C_i^d caused by VM migration is set to 10% of the VM's CPU utilization.

SLAV is employed to assess the overall QoS level of the cloud datacenter, which comprehensively reflects the total performance degradation caused by host overload and VM migrations, as shown in

$$SLAV = SLAVO \times SLAVM. \tag{21}$$

The smaller value of the variable *SLAV* indicates less SLA violations and higher QoS level.

VM consolidation should optimize the energy consumption and SLA violation of the datacenter in a balanced manner. The comprehensive evaluation index *ESV* is obtained by combining the energy consumption index *EC* and the SLA violation index *SLAV*, as shown in

$$ESV = EC \times SLAV, \tag{22}$$

where *EC* represents the total energy consumption of the data center. A lower value of *EC* denotes higher energy efficiency of the datacenter. And a low *ESV* value demonstrates that the datacenter has excellent performance in both energy consumption and QoS.

Dynamic migration of VMs generates certain overhead, such as the occupation of computing resources and energy consumption. Further, the service suspension due to the VM migration may degrade QoS. Therefore, reducing the number of VM migrations *MG* facilitates saving resource and energy consumption, as well as improving QoS, which helps VM consolidation achieve the desired performance.

VI. RESULTS ANALYSIS

In this section, we firstly evaluate three types of VM selection policies proposed for overloaded hosts, which are the HCPS policy, the LCPS policy, and the RCS policy. The evaluation results of the three VM selection policies in energy consumption and SLA violation rates are shown in Table 5. From the results, we can see that among the three VM selection

TABLE 5. Performance evaluation of VM selection policies.

Policy	EC (kWh)	$SLAV(\%)$	$ESV(\%)$
HCPS	100.37	0.00357	0.3583
LCPS	97.10	0.00366	0.3554
RCS	95.92	0.00349	0.3348

policies, the RCS policy has the lowest energy consumption, followed by the LCPS policy and the HCPS policy. With respect to *SLAV*, the RCS policy achieves the lowest value, followed by the HCPS policy and the LCPS policy. Obviously, the RCS policy achieves better performance of energy consumption and SLA violation rate. This is due to that the RCS policy attains better randomness and thus ants have more access to different solutions. In the HCPS policy, VMs with larger CPU utilization have a higher probability of being selected for migration, resulting in higher energy consumption, but at the same time, the risk of host overload is easier to be mitigated, leading to a lower *SLAV*. The LCPS policy prefers to migrate out VMs with low CPU utilization, which saves energy whereas increasing the overload risk of the host, resulting in the highest *SLAV*. According to the experimental results, the RCS policy has the best comprehensive performance to select migration VM for the overloaded host. Therefore, we choose the RCS policy as the VM selection policy for overloaded hosts in the subsequent experiments.

To evaluate the performance of the DA-VMC algorithm, we compare the proposed algorithm DA-VMC with two heuristic VM allocation algorithms (i.e., ST [10], DT [10]) and two ACO-based VM consolidation algorithms (i.e., ACS-VMC [40], EVMCACS [41]). Fig. 2 shows the comparison of energy consumption using real workload among the approaches. The value after the algorithm name is the current parameter value of this algorithm. Compared with ST, DT, ACS-VMC, and EVMCACS, DA-VMC saves 38.3%, 34.1%, 17.7%, and 15.1% of energy consumption respectively. Since DA-VMC prioritizes VM migration to a normal-loaded host, many underloaded hosts are enabled to be switched to sleep mode, thus reducing the number of active hosts in the datacenter and saving a lot of energy.

Regarding *SLAV* of the approaches as shown in Fig. 3, the proposed DA-VMC algorithm achieves the lowest value of *SLAV*. In order to prevent SLA violations, DA-VMC ensures to keep the CPU utilization of hosts below the overload threshold by moving VMs from overloaded hosts. Besides, the heuristic criteria guarantee that the destination host does not exceed the overload threshold after deploying the migrated VM. Therefore, DA-VMC obtains better *SLAV* performance than other algorithms.

Since *SLAV* is a comprehensive index obtained from *SLAVO* and *SLAVM*, *SLAVO* and *SLAVM* of the approaches are evaluated respectively as follows. Fig. 4 depicts the comparison of *SLAVO*, which clearly reveals that DA-VMC has a lower *SLAVO* compared with other approaches. This is

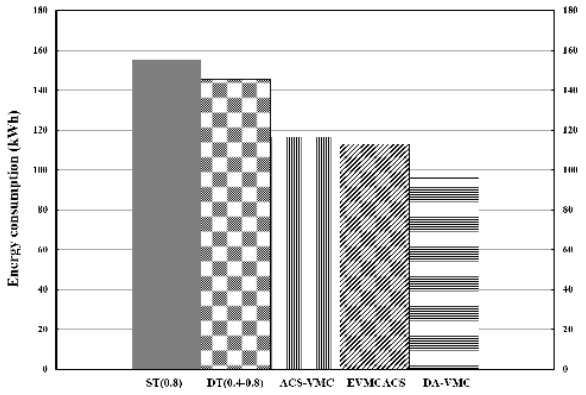

FIGURE 2. Comparison of energy consumption.

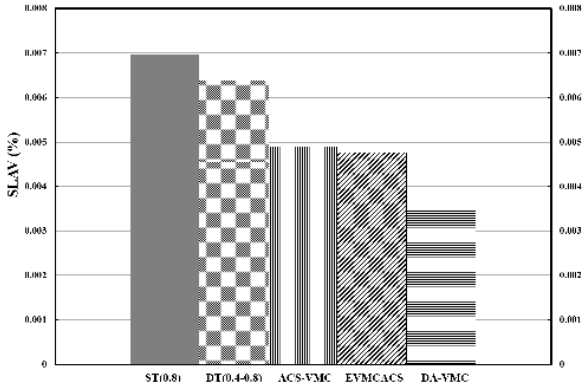

FIGURE 3. Comparison of SLAV.

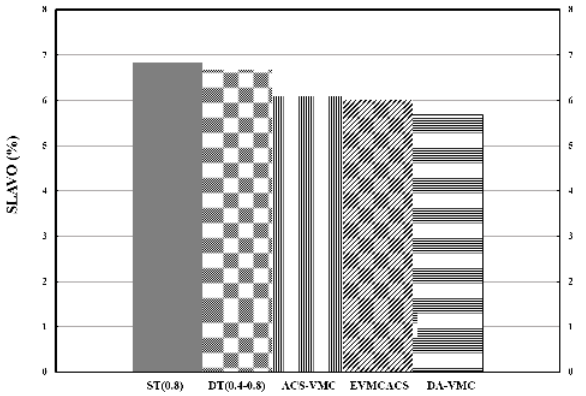

FIGURE 4. Comparison of SLAVO.

primarily due to that DA-VMC limits the resource utilization of the host under the overload threshold and reduces the risk of host resource overload, thus guaranteeing the QoS of the running host. In addition, when redeploying the migrated VM, the DA-VMC algorithm preferentially selects the host with lower resource utilization in the normal-loaded hosts, which guarantees the QoS of running hosts from another perspective. Fig. 5 shows the comparison of *SLAVM* among the approaches. As observed from Fig. 5, DA-VMC has the optimal performance in terms of *SLAVM* among the approaches.

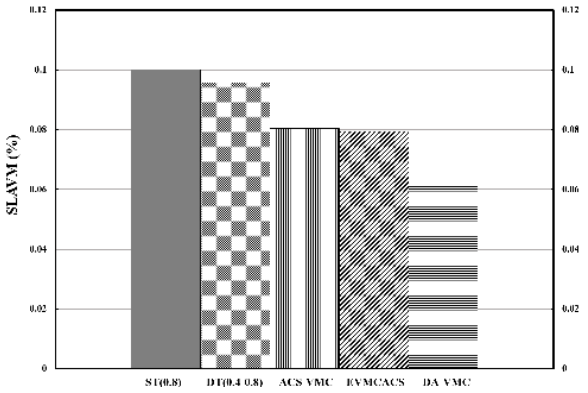

FIGURE 5. Comparison of SLAVM.

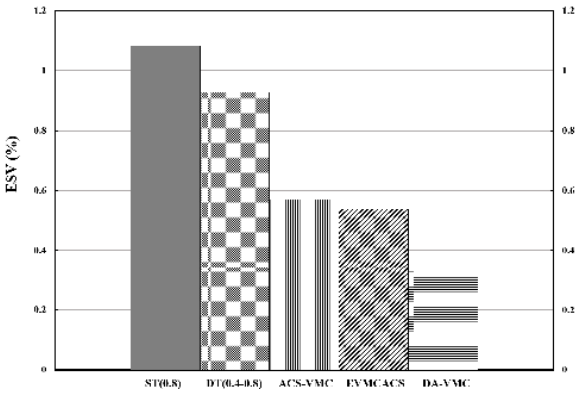

FIGURE 7. Comparison of ESV.

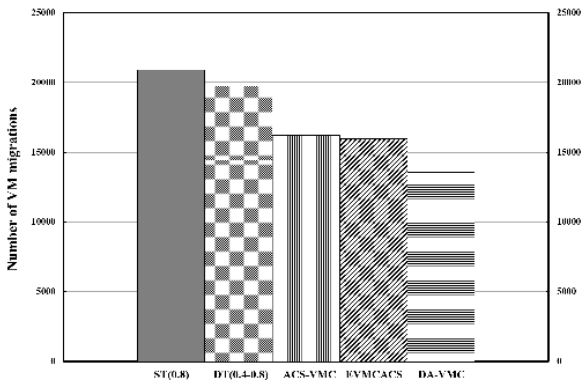

FIGURE 6. Comparison of the number of VM migrations.

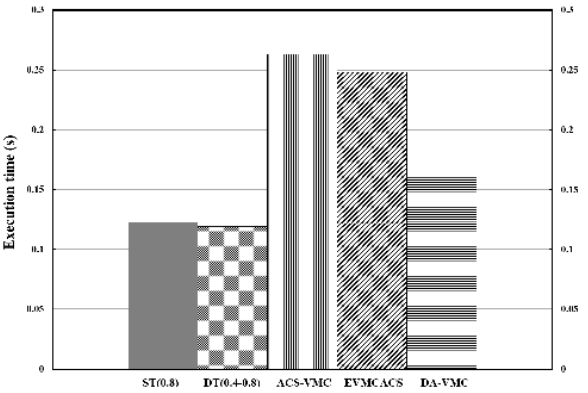

FIGURE 8. Comparison of execution time.

DA-VMC reduces the impact of migration on service quality mainly because that DA-VMC effectively reduces the number of triggered VM migrations as shown in Fig. 6. The combining evaluation of *SLAVO* and *SLAVM* proves that DA-VMC's capability of ensuring QoS is preferable to that of the other approaches.

As shown in Fig. 6, DA-VMC improves the efficiency of migration and obtains the minimum number of migrations. The DA-VMC approach efficiently ensures remarkable QoS of the active hosts and reduces the risk of host overload, leading to a reduction in the number of VM migrations triggered by host overload. For another, the objective function defined in the DA-VMC approach tends to minimize the number of VM migrations.

Fig. 7 depicts the comprehensive performance of the approaches evaluated by employing the *ESV* index. From Fig. 7, we can see that DA-VMC has the best overall performance. According to the experimental results, the *ESV* index obtained by DA-VMC is only 36.1% of that of DT, and the *ESV* index of ST is the worst, which is up to 3.3 times that of DA-VMC. The main reason is that DA-VMC effectively reduces the risk of host overload and the number of VM migrations by precisely identifying the host load status.

The above experiment results employing the PlanetLab workload prove that DA-VMC achieves the goal of reducing datacenter energy consumption, ensuring QoS and making more rational VM migration decisions.

In order to deeply analyze the efficiency of DA-VMC, the execution time of the five approaches is analyzed and compared, as shown in Fig. 8. The heuristic algorithms ST and DT have shorter execution time than other three meta-heuristic algorithms based on ACS because of the low time complexity. Among the three ACS-based approaches, DA-VMC is superior to ACS-VMC and EVMCACS in terms of execution time performance, because it consolidates in stages and limits the size of candidate sets of migration VMs and destination hosts at each consolidation stage. Furthermore, we observed that the execution time of DA-VMC is relatively close to that of the two heuristic algorithms (i.e., ST and DT).

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a VM consolidation approach (DA-VMC) based on double thresholds and ACS. It addresses the problems of high PM power consumption and QoS degradation in datacenters by consolidating VMs into appropriate hosts. The VM consolidation problem is built as a multi-objective optimization problem. The double thresholds are used to make decision of the consolidation conditions that trigger VM consolidation. By treating the mapping relation between VMs and hosts as the food source, the mapping

relation is optimized through a multi-stage consolidation based on ACS. The optimal mapping relation between VMs and hosts is acquired globally through the distributed search and cooperation of the artificial ants. The performance of the proposed approach is evaluated using real workload. The simulation results indicate that compared with other approaches, our approach effectively reduce energy consumption and guarantee excellent QoS of the datacenter.

In the future work, we plan to conduct a further study of employing adaptive thresholds aiming at variable workload to make reasonable decisions of VM migration. Furthermore, we intend to conduct more simulations to evaluate the proposed approach on the real workload.

REFERENCES

[1] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Comput. Surv.*, vol. 47, no. 4, p. 63, Jul. 2015.

[2] Z. Zhou *et al.*, "Fine-grained energy consumption model of servers based on task characteristics in cloud data center," *IEEE Access*, vol. 6, pp. 27080–27090, 2018.

[3] D. Ardagna, G. Casale, M. Ciavotta, and J. F. Pérez, and W. Wang, "Quality-of-service in cloud computing: Modeling techniques and their applications," *J. Internet Services Appl.*, vol. 5, no. 11, pp. 11–27, Sep. 2014.

[4] N. Saswade, V. Bharadi, and Y. Zanzane, "Virtual machine monitoring in cloud computing," *Procedia Comput. Sci.*, vol. 79, pp. 135–142, Dec. 2016.

[5] M. C. S. Filho, C. C. Monteiro, and P. R. M. Inácio, and M. M. Freire, "Approaches for optimizing virtual machine placement and migration in cloud environments: A survey," *J. Parallel Distrib. Comput.*, vol. 111, pp. 222–250, Jan. 2018.

[6] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[7] M. Dorigo, G. Caro, and L. Gambardella, "Ant algorithms for discrete optimization," *Artif. Life*, vol. 5, no. 2, pp. 137–172, Apr. 1999.

[8] P. R. Theja and S. K. K. Babu, "Evolutionary computing based on QoS oriented energy efficient VM consolidation scheme for large scale cloud data centers," *Cybern. Inf. Technol.*, vol. 16, no. 2, pp. 97–112, Jun. 2016.

[9] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang, and Q. Zheng, "Power-aware and performance-guaranteed virtual machine placement in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1385–1400, Jun. 2018.

[10] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generat. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[11] F. Zhang, G. Liu, X. Fu, and R. Yahyapour, "A survey on virtual machine migration: Challenges, techniques, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1206–1243, 2nd Quart., 2018.

[12] C. D. Martino, S. Sarkar, R. Ganesan, Z. T. Kalbarczyk, and R. K. Iyer, "Analysis and diagnosis of SLA violations in a production SaaS cloud," *IEEE Trans. Rel.*, vol. 66, no. 1, pp. 54–75, Mar. 2017.

[13] Z. Zhou, Z. Hu, J.-Y. Yu, J. Abawajy, and M. Chowdhury, "Energy-efficient virtual machine consolidation algorithm in cloud data centers," *J. Central South Univ.*, vol. 24, no. 10, pp. 2331–2341, Oct. 2017.

[14] E. Arianyan, H. Taheri, and S. Sharifian, "Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers," *Comput. Elect. Eng.*, vol. 47, pp. 222–240, Oct. 2015.

[15] Z. Zhou, J. Yu, F. Li, and F. Yang, "Virtual machine migration algorithm for energy efficiency optimization in cloud computing," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 24, p. e4942, Aug. 2018.

[16] A. Horri, M. S. Mozafari, and G. Dastghaibyfard, "Novel resource allocation algorithms to performance and energy efficiency in cloud computing," *J. Supercomput.*, vol. 69, no. 3, pp. 1445–1461, Sep. 2014.

[17] X. Chen, J.-R. Tang, and Y. Zhang, "Towards a virtual machine migration algorithm based on multi-objective optimization," *Int. J. Mobile Comput. Multimedia Commun.*, vol. 8, no. 3, pp. 79–89, 2017. doi: 10.4018/IJMCMC.2017070106.

[18] D. Minarolli, A. Mazrekaj, and B. Freisleben, "Tackling uncertainty in long-term predictions for host overload and underload detection in cloud computing," *J. Cloud Comput.*, vol. 6, no. 4, pp. 4–21, Feb. 2017.

[19] S. S. Masoumzadeh and H. Hlavacs, "An intelligent and adaptive threshold-based schema for energy and performance efficient dynamic VM consolidation," in *Proc. Eur. Conf. Energy Efficiency Large Scale Distrib. Syst.* Vienna, Austria, 2013, pp. 85–97.

[20] Z. Zhou *et al.*, "Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms," *Future Gener. Comput. Syst.*, vol. 86, pp. 836–850, Sep. 2018.

[21] L. Salimian, F. S. Esfahani, and M.-H. Nadimi-Shahraki, "An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines," *Computing*, vol. 98, no. 6, pp. 641–660, Jun. 2016.

[22] S. Y. Z. Fard, M. R. Ahmadi, and S. Adabi, "A dynamic VM consolidation technique for QoS and energy consumption in cloud environment," *J. Supercomput.*, vol. 73, no. 10, pp. 4347–4368, Oct. 2017.

[23] A. Abdelsamea, A. A. El-Moursy, E. E. Hemayed, and H. Eldeeb, "Virtual machine consolidation enhancement using hybrid regression algorithms," *Egyptian Inform. J.*, vol. 18, no. 3, pp. 161–170, Nov. 2017.

[24] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput., Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.

[25] Z. Cao and S. Dong, "Dynamic VM consolidation for energy-aware and SLA violation reduction in cloud computing," in *Proc. 13th Int. Conf. Parallel Distrib. Comput. Appl. Technol.*, Beijing, China, Dec. 2012, pp. 363–369.

[26] S. S. Masoumzadeh and H. Hlavacs, "Dynamic virtual machine consolidation: A multi agent learning approach," in *Proc. IEEE Int. Conf. Autonomic Comput.*, Grenoble, France, Jul. 2015, pp. 161–162.

[27] G. F. Shidik, A. Azhari, and K. Mustofa, "Improvement of energy efficiency at cloud data center based on fuzzy Markov normal algorithm vm selection in dynamic vm consolidation," *Int. Rev. Comput. Softw.*, vol. 11, no. 6, pp. 511–520, Jun. 2016.

[28] H. Li, W. Li, H. Wang, and J. Wang, "An optimization of virtual machine selection and placement by using memory content similarity for server consolidation in cloud," *Future Gener. Comput. Syst.*, vol. 84, pp. 98–107, Jul. 2018.

[29] Y. Laili, F. Tao, F. Wang, L. Zhang, and T. Lin, "An iterative budget algorithm for dynamic virtual machine consolidation under cloud computing environment (revised December 2017)," *IEEE Trans. Serv. Comput.*, to be published. 2018.

[30] M. Mishra and A. Sahoo, "On theory of VM placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach," in *Proc. IEEE 4th Int. Conf. Cloud Comput.*, Washington, DC, USA, Jul. 2011, pp. 275–282.

[31] A. Murtazaev and S. Oh, "Sercon: Server consolidation algorithm using live migration of virtual machines for green computing," *IETE Tech. Rev.*, vol. 28, no. 3, pp. 212–231, 2011.

[32] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu, and H. Tenhunen, "Energy-aware VM consolidation in cloud data centers using utilization prediction model," *IEEE Trans. Cloud Comput.*, to be published. doi: 10.1109/TCC.2016.2617374.

[33] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," *Comput. Netw.*, vol. 57, no. 1, pp. 179–196, 2013.

[34] M. Li, J. Bi, and Z. Li, "Improving consolidation of virtual machine based on virtual switching overhead estimation," *J. Netw. Comput. Appl.*, vol. 59, pp. 158–167, Jan. 2016.

[35] Z. Huang and D. H. K. Tsang, "M-convex VM consolidation: Towards a better VM workload consolidation," *IEEE Trans. Cloud Comput.*, vol. 4, no. 4, pp. 415–428, Oct./Dec. 2016.

[36] Z. Li, C. Yan, L. Yu, and X. Yu, "Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method," *Future Gener. Comput. Syst.*, vol. 80, pp. 139–156, Mar. 2018.

[37] A. Mosa and N. W. Paton, "Optimizing virtual machine placement for energy and SLA in clouds using utility functions," *J. Cloud Comput.*, vol. 5, no. 1, pp. 1–17, Dec. 2016.

[38] H. Li, G. Zhu, C. Cui, H. Tang, Y. Dou, and C. He, "Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing," *Computing*, vol. 98, no. 3, pp. 303–317, Mar. 2016.

[39] M. H. Ferdaus, M. Murshed, R. N. Calheiros, and R. Buyya, "Virtual machine consolidation in cloud data centers using ACO metaheuristic," in *Proc. Eur. Conf. Parallel Process.*, Porto, Portugal, 2014, pp. 306–317.
[40] F. Farahnakian *et al.*, "Using ant colony system to consolidate VMs for green cloud computing," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 187–198, Mar. 2015.
[41] A. Aryania, H. S. Aghdasi, and L. M. Khanli, "Energy-aware virtual machine consolidation algorithm based on ant colony system," *J. Grid Comput.*, vol. 16, no. 3, pp. 477–491, Sep. 2018.
[42] L.-D. Chou, H.-F. Chen, F.-H. Tseng, H.-C. Chao, and Y.-J. Chang, "DPRA: Dynamic power-saving resource allocation for cloud data center using particle swarm optimization," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1554–1565, Jun. 2018.
[43] Q. Zheng, R. Li, X. Li, and J. Wu, "A multi-objective biogeography-based optimization for virtual machine placement," in *Proc. 15th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput.*, Shenzhen, China, May 2015, pp. 687–696.
[44] Q. Chen, J. Chen, B. Zheng, J. Cui, and Y. Qian, "Utilization-based VM consolidation scheme for power efficiency in cloud data centers," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., Jun. 2015, pp. 1928–1933.
[45] Z. Li, C. Yan, X. Yu, and N. Yu, "Bayesian network-based virtual machines consolidation method," *Future Gener. Comput. Syst.*, vol. 69, no. 3, pp. 75–87, Apr. 2017.
[46] R. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.
[47] F. Durao, J. F. S. Carvalho, A, Fonseka, and V. C. Garcia, "A systematic review on cloud computing," *J. Supercomput.*, vol. 68, no. 3, pp. 1321–1346, Jun. 2014.
[48] K. Park and V. S. Pai, "CoMon: A mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 65–74, 2006.

HUI XIAO received the B.S. degree from the School of Software, Shandong University, in 2017. She is currently pursuing the master's degree in software engineering with Central South University. Her main research interests include cloud computing, energy efficient datacenter, and virtual machine management.

ZHIGANG HU received the M.S. and Ph.D. degrees from Central South University, in 1988 and 2002, respectively, where he is currently a Professor with the School of Computer Science and Engineering. He has published over 200 research papers. His research interests include high performance computing, cloud computing, energy-efficient resource management, and virtual machine deployment.

KEQIN LI is a SUNY Distinguished Professor of computer science with the State University of New York. He is also a Distinguished Professor with Hunan University, China. He has published over 640 journal articles, book chapters, and refereed conference papers. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyberphysical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent, and soft computing. He is an IEEE Fellow. He has received several best paper awards. He currently serves or has served on the Editorial Boards of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON SERVICES COMPUTING, and the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING.