IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 June 2022, accepted 25 July 2022, date of publication 28 July 2022, date of current version 5 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3194514

RESEARCH ARTICLE

Dynamic Virtual Machine Consolidation Algorithm Based on Balancing Energy Consumption and Quality of Service

WEI LI[1], QI FAN[1], WENCHAO CUI[1], FANGFANG DANG[2], XIAOLIANG ZHANG[1], AND CHENG DAI[3]

[1]School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China
[2]State Grid Henan Information & Communication Company, Zhengzhou 450052, China
[3]State Grid Chonqing Information & Telecommunication Company, Chongqing 401120, China

Corresponding author: Wenchao Cui (cuzz@ncepu.edu.cn)

This work was supported by the State Grid Corporation of China Headquarters Management Science and Technology Project ''Research and Application of Key Technologies of Adaptive Protection for State Grid Cloud Security'' under Grant SGHAXT00YJJS2100034.

ABSTRACT Virtual machine consolidation (VMC) is an effective way to solve the problems of high power consumption and low utilization in cloud data centers. However, large-scale virtual machine migrations (VMMs) can result in additional workloads, service-level agreement violations (SLAVs), and considerable energy consumption (EC). Existing studies have made great progress in this respect, but the following problems remain: first, the potential overload of the physical host is not considered in the load detection of the physical host; second, the resource-demand scaling of physical hosts is not considered during virtual machine (VM) placement, which results in the lack of accuracy in selecting suitable hosts. In view of the above problems, this study firstly constructs a virtual resource consolidation model based on green energy conservation (GEC-VRCM), which defines the specific process and related attributes of VMC, which is beneficial to improve the consolidation efficiency of virtual resources. Second, based on this model, we propose a dynamic virtual machine consolidation algorithm based on balancing energy consumption and quality of service (EQ-DVMCA) to achieve efficient consolidation of virtual resources. Finally, experiments show that, compared with the selected 12 benchmark algorithms and two advanced VMC algorithms, EQ-DVMCA not only reduces the number of VMMs and EC, but also maintains a high level of Quality of Service (QoS) and achieves a balance between EC and QoS.

INDEX TERMS Virtual machine consolidation, energy consumption, virtual machine migration, quality of service.

I. INTRODUCTION

With the rapid development of cloud computing technology, infrastructure as a service (IaaS) has become an important service mode. Users can rent resources, including server, network, storage, and so on from IaaS providers on demand. The data center with the functions of elastic resource supply, the dynamic allocation of virtual services, and the virtualization and management of infrastructure resources has become an important carrier for building IaaS services [1].

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

However, in the past few years, the extreme energy consumption (EC) of cloud data centers has become a considerable problem [2]. Generally, due to various reasons such as network equipment, server utilization, and the low efficiency of the data center cooling system, data center energy is wasted [3]. According to Gartner's 2013 report, the power consumption of cloud data centers is usually huge, equivalent to the power consumption of 25,000 households [4]. Moreover, the power demand of global data centers is expected to increase by more than 66% from 2011 to 2035 [5]. In addition, the average utilization of the data center is between 12% and 18% [6]. The utilization rate of the Google data center has been between 10% and 50%, while idle servers use 70% of the maximum power of the server [7], which is a waste of power. To sum up, the utilization rate of the data center is very low. Low resource utilization leads to a large amount of energy waste and complexity, which expands the capacity of the data center and further worsens the waste of resources.

Therefore, the high power consumption and low utilization of cloud data center are the challenges faced by cloud computing. An effective and common method to solve this problem is virtual machine (VM) consolidation (VMC). VMC refers to placing VMs on fewer servers through virtual machine migration (VMM) according to their resource requirements and then changing some servers to sleep state, to reduce the energy cost of the data center. However, VMM will increase the cost of computing resources, and large-scale VMM may lead to additional workload, service-level agreement (SLA) violations (SLAV), and considerable EC. At the same time, the service will be suspended during VMM, and long-term migration may further affect the quality of service (QoS). Therefore, in the virtualized cloud data center, the effective consolidation of virtual resources should be to reduce the number of VMMs and the number of servers as much as possible on the premise of meeting the quality of cloud computing services. This method can improve the service resource utilization of the data center and reduce the energy cost, to realize the demand of energy conservation and the environmental protection of the data center.

Aiming at the above problems, this paper constructs a virtual resource consolidation model based on green energy conservation (GEC-VRCM), which defines the specific process and related attributes of virtual resource consolidation. Secondly, based on this model, we propose a dynamic virtual machine consolidation algorithm based on balancing energy consumption and quality of service (EQ-DVMCA) to achieve the efficient consolidation of virtual resources. The contributions of this paper are summarized as follows:

- This paper constructs a GEC-VRCM. The model defines the specific process and related attributes of virtual resource consolidation, which is beneficial to improve the consolidation efficiency of virtual resources.
- Based on GEC-VRCM, we propose an EQ-DVMCA to realize the efficient consolidation of virtual resources. The algorithm includes four parts: Physical Host Load Prediction, Physical Host Load State Detection, VM Selection, and VM Placement.
  - Physical Host Load Prediction: by sensing the load information in the data center, we propose a hybrid prediction algorithm based on the cubic exponential smoothing model and the Elman neural network model (HCESEA), to predict the physical host workload at the next moment. The HCESEA makes error predictions and corrections on the basis of the cubic exponential smoothing model (CES), so it can more accurately predict the load state of the physical host.
  - Physical Host Load State Detection: we propose a hybrid load detection algorithm (HLDA) to identify the current load state of the physical host and divide the load state of the physical host into the following four states: the under load state, the suitable load state, the potential overload state, and the overload state. The HLDA carefully divides the physical host status, which can reduce potential SLAV and improve the QoS.
  - VM Selection: we propose a VM selection algorithm based on CPU and memory perception (CM-VMSA) to select VMs that need migration on some unsuitable load states of hosts. CM-VMSA can reduce the migration time of VMs as much as possible on the basis of reducing the number of VMMs, and improve the QoS.
  - VM Placement: we propose a VM placement algorithm based on resource-demand scaling (RDS-VMPA), which selects suitable physical hosts for VMs migration according to the resource requirements of the VMM queue and the resource information of suitable hosts in the data center. This algorithm takes into account the resource-demand scaling of physical hosts and effectively prevents overloads caused by workload fluctuation. In this way, the VMs can be allocated to physical hosts reasonably, and the resource utilization of data centers is improved.
- We use CloudSim as a simulation framework for experimental evaluation to verify the effectiveness of the proposed method.

The remaining part of this paper is organized as follows. Section II discusses related works. The model is presented in Section III. Our proposed algorithms are introduced in Section IV. Experiments and results are discussed in Section V. Finally, Section VI presents the conclusion and future work.

II. RELATED WORK

The resource prediction of physical host is an important part of virtual machine consolidation algorithm. The higher accuracy of resource prediction, the virtual machine can be allocated to a more reasonable physical host. This approach can reduce SLAV and provide more reliable QoS.

To dynamically predict virtualized resources in order to handle variable workloads, Shyam *et al.* [8] proposed a Bayesian model to determine short and long-term virtual resources requirement of the CPU/memory intensive applications on the basis of workload patterns at several data centers in the cloud during several time intervals. The experimental results showed that the proposed model was able to predict virtual resources in a cloud environment with better accuracy compared to other models.

Targeting the challenging issues of cloud resource optimization, Tseng *et al.* [9] proposed a new prediction approach based on genetic algorithm (GA) for enhancing prediction accuracy in cloud data center. Authors have also proposed a VM placement algorithm for improving the average of resource utilization and reducing the EC of data center based on the prediction results from GA. Simulation results showed that the proposed GA was superior in prediction accuracy to the Grey model in terms of CPU utilization, memory utilization, and EC no matter in stable or unstable utilization tendency.

Berral *et al.* [10] present a methodology to discover resource usage behaviors of containers training Deep Learning (DL) models and observe repeating patterns and similitude of resource usage among containers training different DL models. The repeating patterns observed can be leveraged by the scheduler or the resource autoscaler to reduce resource fragmentation and overall resource utilization in a dedicated DL cluster. The experimental results showed that this method could auto-scale containers to reduce CPU and memory allocation by 30% compared to statistics based reactive policies.

Wang *et al.* [11] observed that Kubernetes Vertical Pod Autoscaler (VPA) used an autoscaling strategy that performs poorly on workloads that periodically change. Authors applied methods such as Holt-Winters exponential smoothing (HW) and Long Short-Term Memory (LSTM) artificial neural networks for time-series analysis, to predict future CPU demand. Results showed that the proposed LSTM-based autoscaler reduces CPU waste by a factor of 2× without incurring more CPU throttling than the default over-provisioned Kubernetes approach.

VMC is one of the effective ways to solve the problem of high power consumption and low utilization in cloud data centers. And the effective consolidation of virtual resources should be to reduce the number of VMMs and the number of servers as much as possible on the premise of meeting the quality of cloud computing service, which could improve the service resource utilization of the data center and reduce the energy cost.

The work related to VMC is shown in Table 1.

In [12], the author proposed an ant colony optimization meta heuristic algorithm based on vector algebra for VMC (AVVMC) to balance the use of computing resources in the data center. The experimental results showed that AVVMC performance had been improved in reducing EC and resource waste.

To reduce unnecessary expenses and energy waste, the authors [13] proposed a virtual machine consolidation algorithm with usage prediction (VMCUP) for improving the energy efficiency of cloud data centers. The results showed that consolidation with usage prediction reduces the total migrations and the power consumption of the servers while complying with the SLA.

In [14], authors proposed a new framework for VMC: Improved Underload Decision algorithm and Minimum Average Utilization Difference algorithm (IUMA) to achieve better energy efficiency. The experimental results showed that the algorithm could reduce the EC and SLAV rate of the data center, and the algorithm had a good effect on improving the energy efficiency of the data center.

Fard *et al.* [15] presented a Dynamic threshold Maximum fit (DthMf) VMC technique to achieve QoS temperature balance in the cloud data center. The principle was to consolidate VMs in high-performance servers instead of low-performance servers in order to produce less heat and less power consumption while having more workload.

In [16], authors developed a Bayesian network-based estimation model (BNEM) for live VMM, allowing a comprehensive treatment of nine actual factors in real data centers. By combining three algorithms corresponding to different phases in VMs consolidation, a hybrid Bayesian network-based VMs consolidation (BN-VMC) method was proposed. The simulation results showed that the method can significantly degrade EC, avoid inefficient VMMs, and SLAVs. Also, their method optimizes resource usage.

Wang and Tianfield [17] designed a new framework of energy-aware dynamic VMC for green cloud computing. Accordingly, the authors proposed a space aware best fit decreasing (SABFD) VM placement policy and a new migration VM selection method-based high-CPU utilization (HS). The scheme's main idea was to migrate VMs to the host with minimum available MIPS after VMs being allocated. The simulation results demonstrated that the proposed work outperforms alternative schemes by saving energy and meeting SLA.

Li *et al.* [18] proposed an energy-aware dynamic VM consolidation (EC-VMC) method that migrates VMs while satisfying constraints on the probabilities of multiple types of resources being overloaded. The proposed algorithm achieved an optimum balance between improving energy efficiency, optimizing resource utilization and guaranteeing QoS.

Xiao *et al.* [19] proposed a merge-and-split-based coalitional game-theoretic approach (CGMS) for VMC in heterogeneous clouds. The proposed scheme was to partition physical machines according to their workload levels and then applying coalitional-game-based VMC algorithm to keep them running in a high energy efficiency state. Experimental results show that the proposed approach clearly outperforms traditional ones in terms of energy saving and load balancing.

In [20], the authors proposed the extension of the Modified Best Fit Decreasing Algorithm (MBFD-EX) and the extension of the First Fit Algorithm (FF-EX) as novel and effective evolutionary methods to enhance VM-to-PM placement. They proposed methodologies for VM allocation to amplify the energy proficiency of cloud computing systems while consolidating more held VMs. These methodologies could merge more VMs with less PMs to accomplish preferred energy proficiency over existing techniques. These approaches showcase the accomplishment of the benefit enhancement and energy-saving.

In order to solve local hotspots, Ilager *et al.* [21] proposed an Energy and Thermal-Aware Scheduling (ETAS) algorithm

TABLE 1. VMC related work.

References	Year	Algorithm	Indicator	Advantage
[12]	2014	AVVMC	EC Time Complexity	The proposed algorithm performance has been improved in reducing EC and resource waste.
[13]	2015	VMCUP	EC Number of VMMs SLAV	The proposed algorithm effectively reduces not only the number of migrations and the EC of the servers, but also the average number of SLAVs.
[14]	2016	IUMA	EC SLAV Number of VMMs	The algorithm can reduce the EC and SLAV rate of the data center, and has a good effect on improving the energy efficiency of the data center.
[15]	2017	DthMf	EC Number of VMMs SLAV	DthMf is optimized and improved in reducing EC and VMM times.
[16]	2017	BN-VMC	EC Number of VMMs SLAV	The proposed algorithm can significantly degrade EC, avoid inefficient VMMs, and improve QoS.
[17]	2018	SABFD HS	EC SLAV Number of VMMs QoS	The proposed scheme saves EC, reduces the number of VMM, and improves the QoS.
[18]	2018	EC-VMC	EC Resource utilization QoS	The proposed algorithm achieves an optimum balance between improving energy efficiency, optimizing resource utilization and guaranteeing QoS.
[19]	2019	CGMS	EC Load balancing Number of VMMs Number of physical host sleep	CGMS algorithm has advantages in energy saving and load balancing.
[20]	2019	MBFD-EX FF-EX	EC Number of physical host sleep Request acceptance ratio	These approaches cases the benefit enhancement and energy-saving are accomplished.
[21]	2019	ETAS	EC Local hotspots	ETAS outperforms other state-of-the-art algorithms by reducing overall energy without any hotspot creation.
[22]	2020	HUA	EC Number of physical host sleep	HUA can effectively detect overloaded hosts, so as to reduce the number of active hosts and reduce EC.
[23]	2020	EQC	EC Number of VMMs QoS	EQC in achieving proper trade-off between two conflicting parameters—energy and QoS.
[24]	2021	ARLCA	EC Load balancing SLAV	The proposed scheme finds the best balance in the process of distributing VMs in the data center, and also ensures the efficient operation of resources.
[25]	2021	EASVMC	EC Number of VMMs Resource utilization	The proposed algorithm has obvious advantages in EC, resource utilization and VMM.

that dynamically consolidates VMs to minimize the overall EC while proactively preventing hotspots. The results showed that ETAS outperforms other state-of-the-art algorithms by reducing overall energy without any hotspot creation.

Patel and Patel [22] proposed a host utilization-aware (HUA) algorithm for underloaded host detection and VM placement. The algorithm made use of the whole data center utilization to build lower threshold that will serve the overload detection policy. The experimental results had demonstrated the efficiency of HUA in detecting overloaded hosts and consequently vacating more hosts, which result on reduced number of active hosts and less EC.

To reduce the cost of cloud data centers, the authors [23] proposed an Energy and QoS-aware VM Consolidation approach (EQC) that can effectively consolidate the VMs among the heterogeneous hosts of a data center. The results validated the efficacy of EQC in achieving proper trade-off between two conflicting parameters—energy and QoS.

In order to reach new frontiers in energy efficient cloud infrastructure Shaw *et al.* [24] proposed an RL Consolidation Agent known as Advanced Reinforcement Learning Consolidation Agent (ARLCA) which is capable of driving both efficiency and delivery of service guarantees by dynamically adjusting its behavior in response to changes in workload variability. Through repeated interactions with the environment ARLCA could discover the optimal balance in the dispersal of VMs across the data center so as to prevent hosts becoming overloaded too quickly but also ensuring that resources are operating efficiently.

In order to effectively solve the problems of resource utilization and EC, the authors [25] proposed an energy-aware algorithm for workflow scheduling in cloud computing with VMC, called EASVMC. The results showed that the proposed algorithm had obvious advantages in EC, resource utilization and VMM.

Performance of each part of VMC is shown in Table 2.

To sum up, existing studies have made great progress in VMC in cloud-data centers, especially in the optimization of EC, VMM times, QoS, and other important indicators. However, most of the current studies on physical host-load detection divide the physical hosts into an overload state, an underload state, and a load state, according to

TABLE 2. Performance of each part of VMC.

References	Host Overload	Host Underload	VM Selection	VM Placement
[12]	The algorithm balances the resource utilization of the server.	This algorithm can reduce the waste of resources caused by idle hosts.	This algorithm can reduce the EC and resource waste in the data center.	The algorithm can improve the resource utilization of the data center and reduce the EC of the data center.
[23]	The algorithm reduces SLAV by predicting the future CPU usage of the PM.	The algorithm can shut down idle PMs and improve resource utilization.	The algorithm can reduce unnecessary migration caused by temporary resource load and improve the QoS.	This algorithm can reduce the frequency of server switching and EC.
[14]	This algorithm can reduce SLAV and improve energy efficiency.	The proposed the Improved Underload Decision (IUD) algorithm to reduce EC.	This algorithm can balance resources and reduce the migration time of VMs.	The proposed the Minimum Average Utilization Difference (MAUD) algorithm can realize load balancing, and reduce the number of VMMs and the EC of PMs.
[15]	The algorithm does a thermal-aware consolidation by defining upper threshold so that the server performance and temperature can be controlled simultaneously.	The algorithm reduces the number of running servers and the EC of idle servers.	The algorithm reduces the number of VMMs by calculating the deviation between the utilization of overloaded servers and their thresholds, and finding VMs close to the deviation.	This algorithm can reduce the performance degradation caused by real-time migration.
[16]	The algorithm detects overloaded PMs by considering their current resource utilization and potential overload probability.	The algorithm reduces EC and improves resource utilization.	The proposed the migration and capacity aware migration selection (MCAMS) algorithm to reduce the migration time of VMs.	The proposed the migration and power aware best fit decreasing (MPABFD) algorithm to reduce the migration probability of VMs and the EC.
[17]	This algorithm reduces SLAV and guarantees QoS.	This algorithm reduces the number of idle physical hosts.	The proposed the Space Aware Best Fit Decreasing (SABFD) to reduce the migrations of VMs and host shutdowns.	The proposed the High CUP utilization based migration VM Selection (HS) can help save energy and assure SLA as well.
[18]	The algorithm improves resource utilization and ensures energy efficiency.	The proposed the Underloaded Physical Host Sleep Selection (UPHSS) algorithm can reduce EC.	The proposed the Overload Probability Decreasing Migration Selection can reduce the migration time and migration times of VMs.	The proposed the Energy-aware and Overload Probability-estimation VM Placement (EOPVMP) algorithm can improve energy efficiency and ensure QoS.
[19]	The proposed algorithm can guarantee the QoS.	The proposed algorithm can improve resource utilization and reduce EC.	The proposed algorithm can optimize energy efficiency and reduce the number of VMMs.	The proposed algorithm ensures the QoS and achieves load balancing of PMs.
[20]	The proposed algorithm reduces the performance degradation caused by PM overload.	The proposed algorithm reduces idle PMs and EC.	The proposed algorithm reduces the EC of the PM.	The proposed algorithm improves the request acceptance ratio, QoS and energy efficiency.
[21]	The proposed algorithm reduces SLAV and improves resource utilization.	The proposed method reduces EC and the number of active hosts.	The proposed method reduces the overall EC of the data center.	The proposed algorithm reduces the impact of hotspots and guarantees the QoS.
[22]	The algorithm reduces SLAV and improves energy efficiency.	The proposed algorithm detects underloaded hosts by considering the overall utilization of the data center, thus reducing idle hosts.	This algorithm can reduce the times of VMM.	The proposed algorithm reduces EC and SLAV.
[23]	The proposed algorithm balances resources and reduces SLAV.	The proposed algorithm can reduce the EC of idle hosts.	The proposed algorithm can improve the QoS and reduce the migration time of VMs.	The proposed algorithm can improve resource utilization and reduce EC.
[24]	The proposed algorithm improves the QoS and reduces SLAV.	The proposed algorithm can improve resource utilization and reduce EC.	The proposed algorithm can reduce the migration times and migration time of VMs.	The proposed algorithm can ensure the resource balance of data center.
[25]	The proposed algorithm can balance resources and improve energy efficiency.	The proposed algorithm can improve resource utilization and reduce EC.	The proposed algorithm can reduce EC and VMM times.	The proposed method can improve the overall resource utilization and QoS of the data center.

the threshold, without considering the potential overload of physical hosts. Secondly, existing studies fail to take into account the resource-demand scaling of physical hosts during the placement of VMs, thus lacking accuracy in the selection of suitable physical hosts. Therefore, aiming at the above problems, we put forward the EQ-DVMCA to realize the efficient migration and consolidation of VMs.

III. MODEL

A. GEVRIM-C

In order to realize the efficient consolidation of virtual resources in a cloud environment, we construct GEVRIM-C. This model is shown in Figure 1. The model consists of a global manager and several local managers. The global manager is deployed inside data centers, and local managers are deployed on physical servers. The local manager collects the real-time load of CPU and the memory of physical hosts and VMs through the monitoring unit, and sends collected data set to the data information module in the global manager. According to the data set in the data information module, the scheduling module in the global manager uses the load prediction center to predict the load status of each physical host at the next moment, and provide the changing trend of the resource demand of each physical host. Secondly, the load detection center combines the real-time load and predicted load of the physical host to divide the load status of the physical host into the following four states: underload state, load state, potential overload state, and overload state. Then, the VM selection center selects VMs for migration from some non-suitable load states of hosts. Finally, a suitable host is selected for the VMs for migration through the VM Placement Center, and the migration scheme is sent to each local manager for migration. After all the VMs are migrated, set all the underloaded hosts to sleep mode to reduce data center energy costs.

The most important feature of the virtual resource consolidation model is that it forms a hybrid consolidation mechanism that combines active control based on workload prediction and passive control based on real-time load state. The advantage of this model is that the fluctuation of the workload can be known in advance by using forecasting technology, which can prevent the fluctuation of the workload. In addition, the model can obtain the actual status of the scheduling policy through feedback technology and implement the VMM operation for some non-suitable load states of hosts, to play a real-time correction and control role.

B. DEFINITION

Table 3 shows the definition of the symbols that are used in this section and the rest of the paper.

Definition 1: A physical host in a data center is $Q^H = \{h_1, h_2, \cdots, h_i, \cdots, h_m\}$. h_i is the *ith* physical host.

Definition 2: The VM queue in the physical host h_i is denoted as $VQ_{h_i} = \{v_1, v_2, \cdots, v_j, \cdots, v_n\}$, where v_j indicates the *jth* VM.

Definition 3: literature [26] showed that about 10% CPU overhead would be generated in the process of VMM, which will lead to the performance degradation of VM. Therefore, each VMM may result in some SLAVs. We need to minimize the number of real-time migration of VMs and ensure the QoS provided. The migration time and performance degradation formula of VM are as follows:

$$T_{v_j} = \frac{v_j^M}{B_j} \tag{1}$$

$$P_j^d = 0.1 \cdot \int_{t_0}^{t_0+T_{v_j}} v_j^C(t)\, dt \tag{2}$$

where, T_{v_j} is the VM v_j time taken to complete the migration, v_j^M is the VM v_j amount of memory used, B_j is the available network bandwidth, P_j^d is the amount of performance degradation caused by VM v_j migration, t_0 is the migration start time, v_j^C is the VM v_j CPU utilization.

TABLE 3. Definition of symbols.

Symbol	Definition
Q_l^H	Suitable host queue.
Q_o^H	Overload host queue.
Q_u^H	Underloaded host queue.
Q_{po}^H	Potentially overloaded host queue.
Q_m^V	VMM queue.
R_h^{vari}	Change in resource requirements for physical host h.
R_h^{CR}	The remaining resources of physical host h.
R_h^{ER}	The estimated remaining resources of physical host h.
Q_r^R	Resource demand reduction queue.
Q_g^R	Resource demand growth queue.
R_h^T	Total resource capacity of physical host H.
L_h^C	Current load of physical host H.
L_h^E	Expected load of physical host H.
MSF	The migration safety factor.
Q^H	Physical host queue.
η_u	Upper limit of the host load threshold
η_l	Lower limit of the host load threshold
δ	Priority factor
C_v^h	CPU usage of VM v on physical host h.
R_v^h	Memory capacity used by VM v on physical host h.
VQ_h	VM queue on physical host h.
T_{v_j}	Time taken for VM v_j to complete migration.
v_j^M	Amount of memory used by VM v_j.
B_j	Network bandwidth available to VM v_j.
P_j^d	Performance degradation caused by VM v_j migration.
v_j^C	CPU usage of VM v_j.
$L_{h_i}^C(t)$	Load of physical host h_i at time t.
f	The activation function.

Definition 4: The EC of the data center is expressed as:

$$EC = \sum\nolimits_{i=1}^{m} \int E\left(L_{h_i}^C(t)\right) dt \tag{3}$$

where EC is the EC of the data center, $L_{h_i}^C(t)$ is the physical host h_i is the load at time t, and E is the power consumption corresponding to the physical host load.

IV. THE PROPOSED ALGORITHMS

Based on GEVRIM-C, we propose an EQ-DVMCA to realize the efficient consolidation of virtual resources. The algorithm includes four parts: Physical Host Load Prediction, Physical Host Load State Detection, VM Selection and VM Placement.

A. PHYSICAL HOST LOAD PREDICTION

According to the dynamic and uncertain characteristics of physical host-load data, we propose a HCESEA. The prediction algorithm uses the cubic exponential smoothing model [27] (CES) to make predictions and then uses the Elman neural network model [28] (ENN) to predict the error of the CES model and finally obtains a prediction value after correcting the error. HCESEA alleviates the influence of model parameters on the overall performance, and the ENN model predicts the error of CES model, its prediction accuracy is better than that of the original data set, which further improves

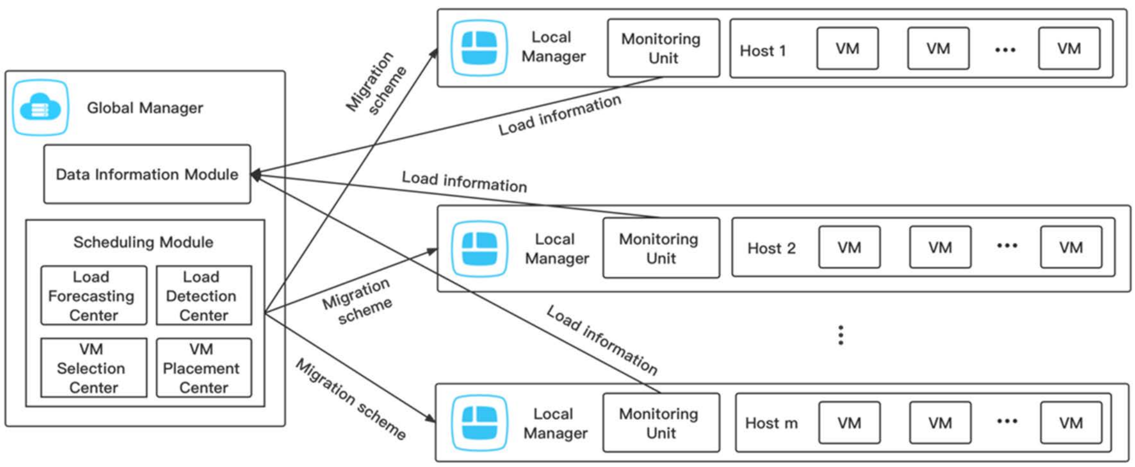

FIGURE 1. GEVRIM-C structure diagram.

the prediction performance of the model. The following is an introduction to the models involved in HCESEA.

1) CES MODEL

The calculation formula of cubic exponential smoothing value is shown as follows:

$$\begin{cases} S_t^{(1)} = \alpha x_t + (1-\alpha) S_{t-1}^{(1)} \\ S_t^{(2)} = \alpha S_t^{(1)} + (1-\alpha) S_{t-1}^{(2)} \\ S_t^{(3)} = \alpha S_t^{(2)} + (1-\alpha) S_{t-1}^{(3)} \end{cases} \tag{4}$$

In Formula (4): α represents the smoothing factor ($0 < \alpha < 1$); $S_t^{(1)}$ represents the primary exponential smoothing value of period t; $S_t^{(2)}$ represents the quadratic exponential smoothing value of period t; $S_t^{(3)}$ represents the cubic exponential smoothing value of period t.

The CES prediction model is as follows:

$$Y_{t+T} = A_t + B_t T + C_t T^2 \tag{5}$$

$$\begin{cases} A_t = 3S_t^{(1)} - 3S_t^{(2)} + S_t^{(3)} \\ B_t = \dfrac{\alpha[(6-5\alpha)S_t^{(1)} - 2(5-4\alpha)S_t^{(2)} + (4-3\alpha)S_t^{(3)}]}{2(1-\alpha)^2} \\ C_t = \dfrac{\alpha^2[S_t^{(1)} - 2S_t^{(2)} + S_t^{(3)}]}{2(1-\alpha)^2} \end{cases} \tag{6}$$

where, T is the number of forecast periods; A_t, B_t, C_t is the prediction parameter.

2) ENN MODEL

The basic structure of ENN consists of four parts: input layer, hidden layer, output layer, and context layer. The ENN model is shown in Figure 2. Unlike the general neural network, the ENN model adds a context layer, and the input of the context layer comes from the output of the hidden layer. This internal feedback mechanism enhances the processing ability of the network for dynamic time data.

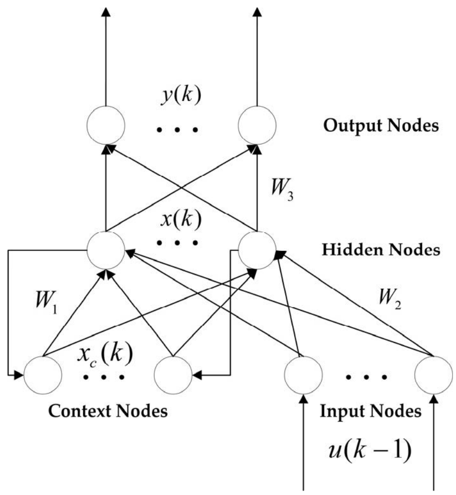

FIGURE 2. Schematic diagram of ENN model.

The mathematical model of ENN is as follows:

$$x(k) = f(w_1 x_c(k) + w_2 u(k-1)) \tag{7}$$

$$x_c(k) = \alpha x_c(k-1) + x(k-1) \tag{8}$$

$$y(k) = g(w_3 x(k)) \tag{9}$$

where, w_1 is the connection weight matrix between the context layer and the hidden layer, w_2 is the connection weight matrix between the input layer and the hidden layer, w_3 is the connection weight matrix between the hidden layer and the output layer, $x_c(k)$ and $x(k)$ represent the output of context layer and hidden layer respectively, and $y(k)$ represents the output of output layer. f is the activation function.

3) HCESEA

Let the dataset $L_h' = \{l_{t1}, l_{t2}, \cdots, l_{tn}\}$ be the real load of physical host h in the time period from t_1 to t_n. Y_h' represents the load prediction sequence of physical host h with length m

obtained by the CES model according to data set L'_h. Then, the load prediction sequence error of the physical host h can be expressed as $E_h = Y'_h - L'_h = e_1, e_2, \cdots, e_m$. The ENN model obtains the error correction sequence E'_h according to E_h, and the load prediction of the corrected physical host h is expressed as $Y^*_h = Y'_h - E'_h$. The structure of the HCESEA is shown in Figure 3.

Set m_{CES} and m_{ENN} be expressed as CES and ENN modules respectively. x_{CES} and x_{ENN} are input vectors of the two models respectively. Then, at t_{n+1}, the output of the model can be expressed as:

$$Y^* = m_{CES}(x_{CES}) - m_{ENN}(x_{ENN}) \tag{10}$$

HCESEA is shown in Algorithm 1.

Algorithm 1 HCESEA

Input: Q^H, L'
Output: Y^*
Step 1: Initialize CES model and its related parameters.
Step 2: Using the L', the predicted load sequence Y' of the physical host is obtained through the CES model.
Step 3: For each h in Q^H, get the real load L'_h of the physical host h. Obtain the predicted load Y'_h of the physical host h through Y'. Then the prediction sequence error of the physical host h is $E_h = Y'_h - L'_h$. Add E_h to E.
Step 4: Initialize ENN model and its related parameters. Train ENN model through E.
Step 5: The error correction sequence E' is obtained through the ENN model.
Step 6: For each h in Q^H, obtain the correction error E'_h of physical host h through E'. Then the load prediction of the modified physical host h is $Y^*_h = Y'_h - E'_h$. Add Y^*_h to Y^*.
Step 7: Return Y^*.

B. PHYSICAL HOST LOAD STATE DETECTION

Because the load of each physical host in the data center changes dynamically, the load of different physical hosts at the same time will be different. Therefore, we propose the HLDA. The HLDA first obtains the predicted load and real-time load of each physical host in the data center and then divides the load state of the physical host into the following four states base on the set threshold: the under-load state, the suitable-load state, the potential overload state, and the overload state.

The specific process of the HLDA is shown as follows. When the load of a physical host is higher than the upper threshold, the physical host is added to the overloaded host queue. When the load of a physical host is lower than the lower threshold, the physical host is added to the underloaded host queue. When the load of the physical host is within the threshold, it needs to be judged in combination with the predicted load of the physical host. If the expected load of the physical host is higher than the upper threshold, the physical host is added to the potentially overloaded host queue. On the contrary, the physical host is added to the suitable host queue.

HLDA is shown in Algorithm 2.

Algorithm 2 HLDA

```
Input: Q^H, Y*
Output: Q_u^H, Q_l^H, Q_o^H, Q_po^H
foreach h in Q^H do:
    L_h^C = h.getCurrentLoad();
    if (L_h^C > η_u)
        Q_o^H.addQueue)h);
    else if (L_h^C < η_l)
        Q_l^H.addQueue)h);
    else
        Y_h^* = h.getCorrectForecastLoad)Y*);
        if (Y_h^* > η_u)
            Q_po^H.addQueue)h);
        else
            Q_l^H.addQueue)h);
return Q_u^H, Q_l^H, Q_o^H, Q_po^H;
```

C. VM SELECTION

The real-time migration of VM will have a negative impact on the performance of applications running on the VM, resulting in some SLA conflicts. Moreover, the performance degradation during VMM is related to the migration time of the VM. Therefore, we propose CM-VMSA. CM-VMSA can reduce the migration time of VMs as much as possible on the basis of reducing the number of VMMs, to improve the quality of service. It can be seen from Section 4.2 that there are three types of physical hosts that need to migrate VMs, which are located in Q^H_u, Q^H_o, and Q^H_{po}, respectively. Therefore, we perform the following operations on these three types of queues:

Q^H_u*:* add all VMs on such physical hosts to Q^V_m. After all the VMs have migrated to other physical hosts, switch the physical host to sleep mode to reduce EC.

Q^H_o*:* first, arrange all VMs on such physical hosts in descending order according to CPU utilization; secondly, in the relative order of decreasing CPU utilization, arrange them in ascending order according to the amount of memory occupied by the VM; finally, try to migrate each VM in turn, and calculate the load of the physical host in real time. If $L^C < \eta_u$, stop the above operation and add the VM for migration to Q^V_m.

Q^H_{po}*:* first, arrange all VMs on such physical hosts in descending order according to CPU utilization; secondly, in the relative order of decreasing CPU utilization, arrange them in ascending order according to the amount of memory occupied by the VM; finally, try to migrate each VM in turn and calculate the load of the physical host in real time. If $(L^E - \sum_{i=i}^{k} C^h_i) < \eta_u$, stop the above operation and add the VM for migration to Q^V_m.

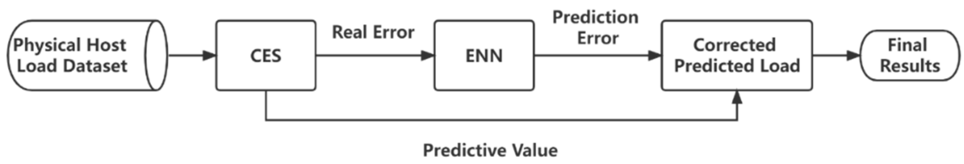

FIGURE 3. **HCESEA structure diagram.**

CM-VMSA is shown in Algorithm 3.

Algorithm 3 CM-VMSA

Input: Q_u^H, Q_o^H, Q_{po}^H
Output: Q_m^V
foreach h **in** Q_u^H **do:**
  foreach v **in** h **do:**
    Q_m^V.**addQueue**)v)$\mathcal{I}$
foreach h **in** Q_o^H **do:**
  VQ_h.**sortDescendUtilization**();
  VQ_h.**sortAscendMemory**();
while $(L^C \geq \eta_u)${
    $v = VQ_h$.**getFirstHost**();
    Q_m^V.**addQueue**)v)$\mathcal{I}$
    $L^C = L^C - v$.**get**C_v^h()$\mathcal{I}$
  }
foreach h **in** Q_{po}^H **do:**
  VQ_h.**sortDescendUtilization**();
  VQ_h.**sortAscendMemory**();
  $d = L^E$;
  while $(d \geq \eta_u)${
    $v = VQ_h$.**getFirstHost**();
    Q_m^V.**addQueue**)v)$\mathcal{I}$
    $d = d - v$.**get**C_v^h()$\mathcal{I}$
  }
return Q_m^V;

D. VM PLACEMENT

VM placement refers to selecting the suitable physical host for the migration VM according to the resource requirements of the VMM queue and the resource information of the host in the data center. Therefore, we propose an RDS-VMPA. RDS-VMPA first calculates the workload of all physical hosts that meet the resource allocation requirements of VMs in the future. Secondly, RDS-VMPA divides the physical hosts into resource demand reduction queue Q_r^R and resource demand growth queue Q_g^R according to the prediction, and sorts these queues in a specific way. Finally, RDS-VMPA determines the destination host through further screening. The specific algorithm steps are shown as follows:

1. Calculate the resource demand scaling on each physical host, and its value is the predicted load of the physical host minus the real-time workload of the physical host. Resource demand scaling reflects the changing trend of VM resource demand on the physical host.
2. Calculate the unallocated resources of the physical host. The unallocated resources are the total resources of the physical host minus the real-time workload of the physical host. And calculate the estimated remaining resources of the physical host. The estimated remaining resources are the difference between the unallocated resources and the resource demand scaling. Filter out the list of physical hosts with estimated remaining resources greater than zero to form a list of candidate physical hosts.
3. If the value of the resource demand scaling of the physical host is negative, the physical host will be added to the Q_r^R. And calculate the difference between the remaining resources of the physical host and the resource demand scaling of the physical host, and arrange the Q_r^R in descending order based on the difference. If the value of the resource demand scaling amount of the physical host is positive, the physical host will be added to the Q_g^R. The migration security factor *MSF* of the physical host is calculated. The security factor is the ratio of the resource demand scaling to the remaining resources of the physical host, and the Q_g^R is arranged in ascending order according to the *MSF*.
4. After two queues are generated, if one queue is empty, the head host of the other queue is directly selected as the placement host. If none is empty, the priority factors δ of the head hosts of the two queues are compared.
5. If Q_r^R and Q_g^R are empty and Q_m^V is not empty, it is necessary to judge whether the underloaded host queue Q_u^H is empty. If the Q_u^H is not empty, the Q_u^H is arranged in descending order according to the amount of unallocated resources. Assign the VM to be migrated to the queue head host of the Q_u^H, and add the host to Q_g^R. On the contrary, it indicates that the current running host can no longer meet the resource requirements. It is necessary to start a new physical host and allocate the VM to be migrated to the physical host.
6. After the VMM is completed, change the host in the Q_u^H to sleep state.

RDS-VMPA is shown in Algorithm 4.

RDS-VMPA takes into account the resource demand scaling of physical hosts and effectively reduces potential SLAVs. In addition, the algorithm can allocate VMs to physical hosts reasonably, improving the resource utilization of data centers.

Algorithm 4 RDS-VMPA

```
Input: Q_l^H, Y*, Q_m^V
Output: Allocation of Q_m^V
foreach h in Q_l^H do:
    R_h^T = h.getTotalResource();
    L_h^C = h.getCurrentLoad();
    Y_h^* = h.getCorrectForecastLoad)Y*);
    R_h^vari = Y_h^* - L_h^C;
    R_h^CR = R_h^T - L_h^C;
    R_h^ER = R_h^CR - R_h^vari;
if (R_h^vari > 0)
  Q_g^R.addQueue)h);
else
  Q_r^R.addQueue)h);
Q_r^R.sortDescendOrderByR_h^ER()I
foreach h in Q_g^R do:
  MSF = R_h^vari / R_h^CR;
Q_g^R.sortAscendOrderByMSF();
foreach v in Q_m^V do
  if (Q_r^R is NULL)
    allocateHost = Q_g^R.getFirstHost();
  else if (Q_g^R is NULL)
    allocateHost = Q_r^R.getFirstHost();
    else
      h1 = Q_r^R.getFirstHost();
      h2 = Q_g^R.getFirstHost();
      if (R_h1^CR > R_h2^CR)
        allocateHost = h1;
      else if (R_h2^CR > R_h1^ER)
        allocateHost = h2;
      else
        δ = R_h1^ER - R_h^CR;
        if (δ > v.getResource()/2)
          allocateHost = h1;
        else
          allocateHost = h2;
  if (allocateHost != NULL)
    allocate v to allocateHost;
  else if (Q_u^H != NULL)
    Q_u^H.sortDescendOrderByR_h^CR();
    h3 = Q_u^H.getFirstHost();
    allocatev to h3;
    Q_g^R.add)h3);
  else
    h4 = Q^H.startHost();
    allocate v to h4;
    Q_g^R.add)h4);
  if (Q_u^H != NULL)
    foreach h in Q_u^H do:
    h.changeSleep();
return Allocation of Q_m^V;
```

E. TIME COMPLEXITY ANALYSIS

This section mainly analyzes the time complexity of EQ-DVMCA. To analyze the time complexity of EQ-DVMCA, we first set the number of PMs to M and the number of VMs to N. EQ-DVMCA consists of four independent stages Physical Host Load Prediction, Physical Host Load State Detection, VM Selection and VM Placement.

First of all, Physical Host Load Prediction mainly predicts the load of physical hosts. It needs to traverse the collection of physical hosts running in the cloud data center once. The time complexity of this phase is $O(M)$.

Secondly, Physical Host Load State Detection divides the load status of the physical host according to the predicted load and real-time load of the physical host. The time complexity of this stage is $O(M)$.

Thirdly, VM Selection mainly selects the VM to be migrated from the physical host. In this phase, there are three types of physical hosts that need to migrate VMs: Q_u^H, Q_o^H, and Q_{po}^H. Let $Q_u^H = m_1, Q_o^H = m_2, Q_{po}^H = m_3$, and the number of VMs in Q_u^H, Q_o^H, and Q_{po}^H is n_1, n_2, and n_3, respectively. VMs on a physical host use quicksort. Therefore, the time complexity of this stage is $O(m_1 \cdot n_1 logn_1 + m_2 \cdot n_2 logn_2 + m_3 \cdot n_3 logn_3)$, that is, $O(M \cdot NlogN)$.

Finally, VM Placement needs to allocate the VMs to be migrated to appropriate PMs. The time complexity of this phase is $O(M + MlogM + N \cdot MlogM)$, that is, $O(N \cdot MlogM)$.

To sum up, the time complexity of EQ-DVMCA is $O(M + M + M \cdot NlogN + N \cdot MlogM)$, that is, $O(M \cdot NlogN)$ or $O(N \cdot MlogM)$.

V. EXPERIMENTS AND RESULTS

A. SIMULATION ENVIRONMENT SETTINGS

We use the CloudSim [29] simulation platform to evaluate our proposed EQ-DVMCA and compare it with some benchmark algorithms and current advanced methods. We use CloudSim4.0, which is an event-driven emulator for simulating cloud computing infrastructure and application services. CloudSim supports virtual resource management and modeling, EC, VMM and other functions [30].

Due to the large memory gap between different physical host models, the experimental environment of this study simulates the experimental environment of literature [22, 26]. The experiment simulated a data center consisting of 800 heterogeneous physical hosts, including three types of physical hosts: HP ProLiant ML110 G4, HP ProLiant ML110 G5, and HP ProLiant DL360 G7. The host configuration is shown in Table 4.

Four types of VMs were selected in the experiment: High-CPU Medium Instance, Extra Large Instance, Small Instance and Micro Instance. Table 5 lists the VM attributes.

In order to make the simulation results authentic, we have used data provided as a part of the CoMon project, a monitoring infrastructure for PlanetLab [31]. This data provides CPU

TABLE 4. Host configuration.

Type	MIPS	Core	Ram (MB)	Bandwidth (Gbps)	Number
HP ProLiant ML110 G4-Xeon 3040	1860	2	4096	1	300
HP ProLiant ML110 G5-Xeon 3075	2660	2	4096	1	300
HP ProLiant DL360 G7-Xeon X5675	3067	12	16384	1	200

TABLE 5. VM attributes.

Type	MIPS	RAM
High-CPU Medium Instance	2500	0.85 GB
Extra Large Instance	2000	3.75 GB
Small Instance	1000	1.7 GB
Micro Instance	500	613 MB

TABLE 6. Workload informations for PlanetLab.

Date	Number of VMs	Mean (%)	St.dev. (%)
2011/03/03	1052	12.31	17.09
2011/03/06	898	11.44	16.83
2011/03/09	1061	10.70	15.57
2011/03/22	1516	9.26	12.78
2011/03/25	1078	10.56	14.14
2011/04/03	1463	12.39	16.55
2011/04/09	1358	11.12	15.09
2011/04/11	1233	11.56	15.07
2011/04/12	1054	11.54	15.15
2011/04/20	1033	10.43	15.21

requirements collected from more than 1,000 VMs running on servers in more than 500 locations around the world. CPU utilization within this data set was measured at 5-minute intervals. We randomly selected 10 days from the March and April 2011 data sets as the workload data set. Table 6 shows the information for the workload data. After creating PM and VM instances on the CloudSim platform, we use this data set to generate VM workloads, which are then randomly deployed to PM based on VM resource requirements.

In order to ensure that the algorithm proposed in this paper is still effective in different working environments, we also use the real workload from the distributed data center Bitbrains as the experimental data [32], [33]. The data was obtained by monitoring and managing 1750 VMs in the Bitbrains data center every five minutes. The data includes the running status of VMs in Bitbrains within 4 months. Considering the comparison with PlanetLab data set, the data of 10 days in this data set are selected in this paper. Table 7 shows the workload data information.

B. PERFORMANCE INDEX

This paper adopts the following six performance evaluation indicators: VMM, SLAV Time Per Active Host (SLATAH), Performance Degradation due to Migration (PDM), SLAV, EC, EC and SLAV (ESV) [26].

TABLE 7. Workload informations for Bitbrains.

Date	Number of VMs	Mean (%)	St.dev. (%)
2013/08/02	1237	7.20	5.97
2013/08/04	1233	8.05	4.83
2013/08/05	1232	8.99	6.36
2013/08/08	1209	10.27	6.64
2013/08/11	1202	9.06	6.92
2013/08/15	1191	8.71	5.67
2013/08/19	1188	8.13	5.34
2013/08/20	1186	8.99	3.45
2013/08/22	1183	5.89	3.16
2013/08/24	1175	9.56	4.68

SLATAH refers to the percentage of time when the CPU utilization of the physical host reaches 100% during operation. The definition is as follows:

$$\mathrm{SLATAH} = \frac{1}{m}\sum\nolimits_{i=1}^{m}\frac{T_{h_i}^{S}}{T_{h_i}^{a}} \tag{11}$$

where, $T_{h_i}^{S}$ represents physical host h_i the time of SLAV caused by CPU utilization reaching 100% during operation, $T_{h_i}^{a}$ represents the running time of physical host h_i.

PDM indicates the degradation of QoS caused by VMM. The definition is as follows:

$$\mathrm{PDM} = \frac{1}{n}\sum\nolimits_{j=1}^{n}\frac{P_j^{d}}{v_j^{c}} \tag{12}$$

where, n represents the number of VMs, v_j^c represents VM v_j CPU size requested during runtime.

SLAV is a comprehensive evaluation of SLATAH and PDM, which is an evaluation index of data center QoS. The definition is as follows:

$$\mathrm{SLAV} = \mathrm{SLATAH}\cdot\mathrm{PDM} \tag{13}$$

The lower the SLATAH and PDM values of the data center, the smaller the comprehensive index SLAV and the higher the QoS of the data center.

Energy conservation can not only reduce the operation and maintenance cost of data centers, but also is an important way to build green data centers. We used the SPECpower benchmark to provide actual power consumption data for the server. Table 8 lists the power consumption characteristics of the host.

At the same time, low EC also means high utilization of resources. Considering the requirements of data center for low EC and high QoS, the comprehensive evaluation index ESV of QoS and EC is defined. The expression is as follows:

$$\mathrm{ESV} = \mathrm{EC}\cdot\mathrm{SLAV} \tag{14}$$

The lower the ESV value, the lower the EC of the data center and the higher the QoS.

C. BENCHMARK ALGORITHM AND PARAMETER SETTING

The authors of [26] carried out in-depth research on the VMC process on CloudSim platform and proposed the corresponding algorithms. These include the host-load detection

TABLE 8. Power consumption characteristics of host (Watt).

Host	0 (%)	10 (%)	20 (%)	30 (%)	40 (%)	50 (%)	60 (%)	70 (%)	80 (%)	90 (%)	100 (%)
HP Proliant G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP Proliant G5	93.7	97	101	105	110	116	121	125	129	133	135
HP Proliant G7	55.6	95.4	107	115	124	133	142	155	173	192	216

TABLE 9. EQ-DVMCA related parameters.

Algorithm		Parameter	Value
EQ-DVMCA	HCESEA	α	0.6
		Initial value	The average value of historical CPU utilization in the first three periods
		The number of nodes in the input layer	12
		The number of nodes in the hidden layer	6
		The number of iterations	15000
		f	$\frac{1}{1+e^{-x}}$
		The error threshold	0.00001
	HLDA	$\boldsymbol{\eta_u}$	0.8
		$\boldsymbol{\eta_l}$	0.2

algorithms: static threshold (THR); interquartile range (IQR); local regression robust (LRR); median absolute deviation (MAD); VM selection algorithm: minimum migration time (MMT); random choice (RC); maximum correlation (MC); and VM placement algorithm: power-aware best fit decreasing (PABFD). By combining these algorithms, we can obtain 12 different VMC algorithms. Each consolidation method includes the host-load detection algorithm, the VM selection algorithm, and VM placement algorithm. The four load detection algorithms have their own safety factor: when the safety factor is set too large, it indicates that the load detection will pay attention to the stability of the physical host to ensure the QoS; on the contrary, the load detection algorithm will give up some stability to pursue the full utilization of the physical host resources to achieve the purpose of energy saving. According to the literature [26], it is found that when the safety parameters of THR, IQR, LRR, and MAD are set to 0.8, 1.5, 1.2, and 2.5, respectively, the corresponding consolidation algorithm can achieve relative balance in terms of energy saving and QoS assurance. Therefore, we selected the above 12 VMC algorithms as the benchmark algorithm in this paper.

At the same time, in order to verify the effectiveness of EQ-DVMCA in VMC, we also selected two current advanced algorithms: CGMS [19] and HUA-LRR-MC-1.2 [22], for experimental comparison.

The relevant parameters of EQ-DVMCA proposed in this paper are shown in Table 9. All these experimental parameters are set according to empirical values.

D. EXPERIMENTAL RESULTS

The load prediction of the physical hosts is very important to understand the future load status of the physical hosts. We first verified the effectiveness of HCESEA. We adopted mean absolute error (MAE) and mean squared error (MSE) as the evaluation criteria for the model prediction results, as shown in Table 10. The smaller the prediction value of MAE, the better the prediction accuracy of MAE, where y_i is the true value, and $\bar{y}_i$ is the predicted value.

TABLE 10. Error index.

Index	Formula
MAE	$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \bar{y}_i\rvert$
MSE	$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2$

The data set of 2011/03/03 in PlanetLab and 2013/08/02 in Bitbrains were used in the experiment. The load information of 1000 VMs is randomly selected from the above two data sets as the data data1 (2011/03/03) and data2 (2013/08/02) in this experiment. The ratio of training data set to test data set is 7:3.

We selected Bayesian [8], GA [9], ENN, and CES as the comparison algorithms of this experiment. To ensure the fairness of the experiment, the parameter settings of ENN and CES were consistent with those of HCESEA. The experimental results are shown in Table 11.

TABLE 11. Experimental results.

Data	Index	Bayesian	GA	CES	ENN	HCESEA
Data1	MAE	0.1242	0.1615	0.4016	0.2142	**0.0987**
	MSE	0.0873	0.0936	1.1039	0.1043	**0.0826**
Data2	MAE	0.1537	0.1812	0.4563	0.2316	**0.1134**
	MSE	0.0914	0.0982	1.4267	0.1328	**0.0842**

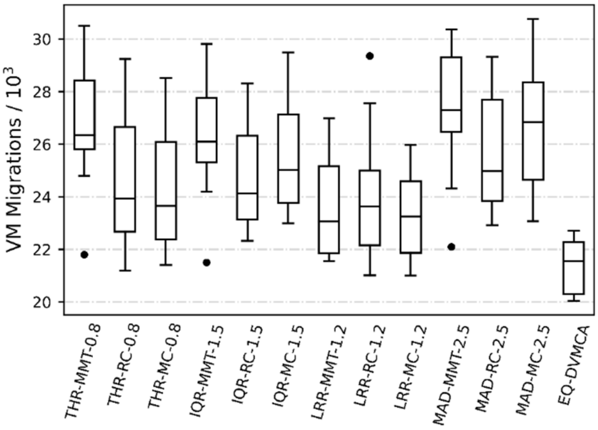

FIGURE 4. VMM comparison of EQ-DVMCA and benchmark algorithm in PlanetLab dataset.

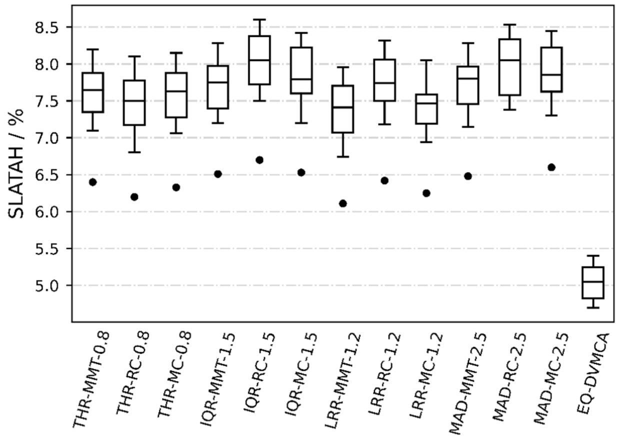

FIGURE 5. SLATAH comparison of EQ-DVMCA and benchmark algorithm in PlanetLab dataset.

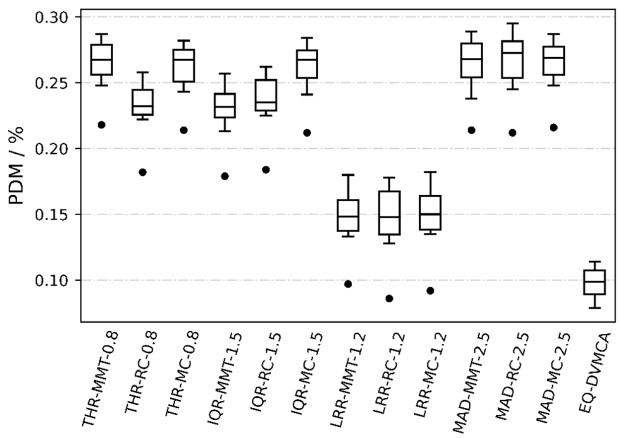

FIGURE 6. PDM comparison of EQ-DVMCA and benchmark algorithm in PlanetLab dataset.

According to Table 10, compared with Bayesian, GA, CES, and ENN, the HCESEA proposed in this paper has the best effect. Bayesian is easily affected by extreme values, which leads to the decline of prediction accuracy. GA is prone to premature convergence, which leads to unstable accuracy of prediction. The ENN model in HCESEA is the prediction of the error of CES model; thus, correcting the error of the model can obtain a better prediction accuracy. As a result, the load state of the physical host can be predicted more accurately using HCESEA.

Next, in order to verify the performance advantages of EQ-DVMCA, we compared it with 12 benchmark algorithms on the data set of PlanetLab. Table 12 shows the simulation results of the algorithm.

It can be seen from Figure 4 that compared with other benchmark algorithms, the VMM value of EQ-DVMCA proposed in this paper is the smallest. This is because the HCESEA used in this paper has high prediction accuracy and can predict the future load value of the host according to the historical load data. In addition, the CM-VMSA proposed in this paper is subsequently adopted to further keep the PM in a stable state in a short time, thus avoiding frequent migration of VMs to a large extent.

It can be seen from Figure 5 that the SLATAH value of EQ-DVMCA proposed in this paper is the lowest. This is because EQ-DVMCA can more accurately predict the changes of physical host workload through HCESEA, thus reducing the probability of physical host overload. Secondly, RDS-VMPA can effectively cope with workload fluctuations and maintain the normal working status of the PMs in the data center for a long time.

It can be seen from Figure 6 that the PDM value of EQ-DVMCA proposed in this paper is the lowest. This is because EQ-DVMCA can effectively avoid invalid VMM through CM-VMSA and reduce the loss of computing resources caused by VMM to the data center. Secondly, RDS-VMPA can effectively cope with workload fluctuations, thus fundamentally reducing the losses caused by VMM.

It can be seen from Figure 7 that the SLAV value of EQ-DVMCA proposed in this paper is significantly smaller than that of other benchmark algorithms. This is because the EQ-DVMCA obtains the resource-demand scaling of physical hosts through CM-VMSA, thus taking into account the changing trend of the resource demand of the VM on the physical host. This method can effectively prevent overload caused by load fluctuation, and reduce the probability of resource competition generated by PMs as much as possible.

It can be seen from Figure 8 that the EC value of EQ-DVMCA proposed in this paper is the smallest. This is because the CM-VMSA used in this paper can use the objective function of CPU and memory utilization to select VMs, so as to solve the pressure of overloaded hosts as soon as possible. Secondly, RDS-VMPA is used to place VMs, which can not only improve the utilization of PM resources, but also shut down more idle hosts, thus saving more EC.

TABLE 12. Simulation results of EQ-DVMCA and benchmark algorithm in PlanetLab dataset.

Method	VMM	SLATAH (%)	PDM (%)	SLAV (10^{-5})	EC (kW·h)	ESV (10^{-2})
THR-MMT-0.8	26705	7.55	0.26	19.63	200.18	3.93
THR-RC-0.8	24677	7.40	0.23	17.02	194.07	3.30
THR-MC-0.8	24317	7.52	0.26	19.55	190.40	3.72
IQR-MMT-1.5	26214	7.65	0.23	17.60	187.92	3.31
IQR-RC-1.5	24634	7.96	0.24	19.10	176.78	3.38
IQR-MC-1.5	25579	7.76	0.26	20.18	182.96	3.69
LRR-MMT-1.2	23548	7.30	0.15	10.95	170.34	1.87
LRR-RC-1.2	24100	7.67	0.15	11.51	165.19	1.90
LRR-MC-1.2	23303	7.38	0.15	11.07	170.59	1.89
MAD-MMT-2.5	27227	7.65	0.26	19.89	188.62	3.75
MAD-RC-2.5	25569	7.99	0.27	21.57	183.16	3.95
MAD-MC-2.5	26601	7.81	0.26	20.31	184.96	3.76
EQ-DVMCA	**21345**	**5.04**	**0.10**	**5.04**	**142.39**	**0.72**

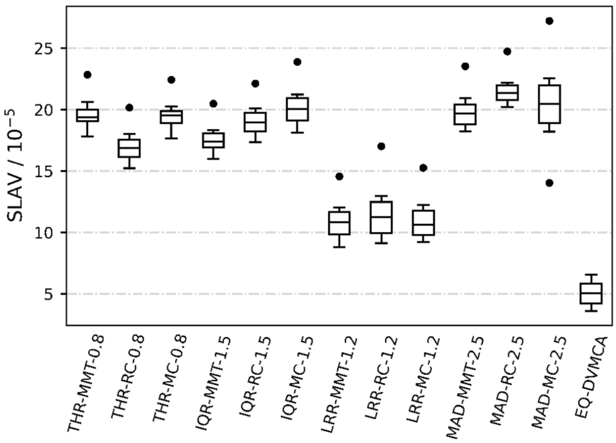

FIGURE 7. SLAV comparison of EQ-DVMCA and the benchmark algorithm in PlanetLab dataset.

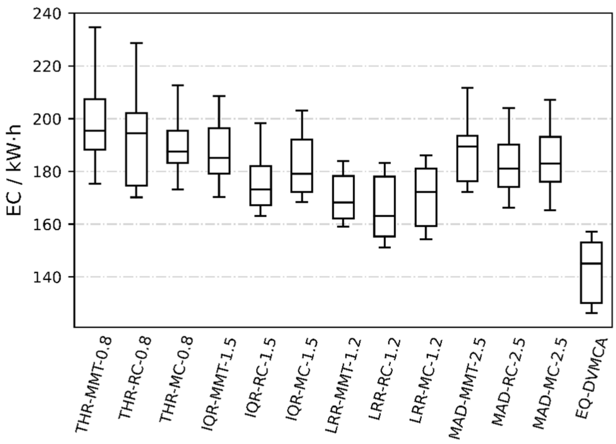

FIGURE 8. EC comparison of EQ-DVMCA and benchmark algorithm in PlanetLab dataset.

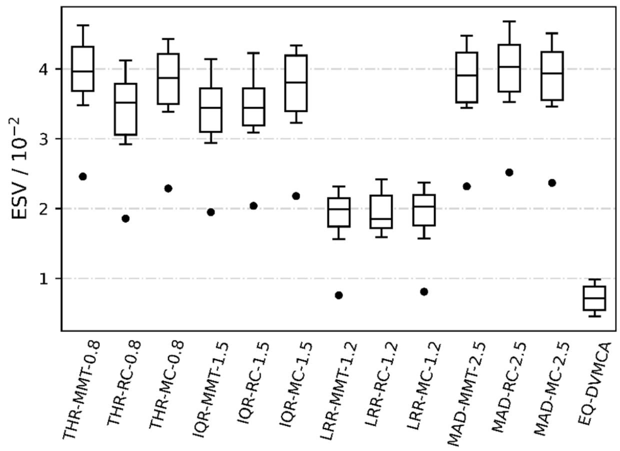

FIGURE 9. ESV comparison of EQ-DVMCA and benchmark algorithm in PlanetLab dataset.

It can be seen from Figure 9 that the ESV value of EQ-DVMCA proposed in this paper is the lowest. This is because EQ-DVMCA not only guarantees the QoS, but also reduces the EC of the data center by reducing the number of VMMs. This makes EQ-DVMCA strike a balance between EC and QoS.

To sum up, EQ-DVMCA performs best in all the six performance tests. Compared with the benchmark algorithm, EQ-DVMCA not only significantly reduces the number of VMM and EC but also provides reliable QoS. According to the calculation in Table 12, compared with the benchmark algorithm, EQ-DVMCA is reduced by 8.40–21.60% in the VMM index, 30.96–36.92% in the SLATAH index, 33.33–62.96% in the PDM index, 53.97–76.63% in the SLAV index, 13.80–28.87% in the EC index, and 61.50–81.77% in the ESV index. These prove the superiority of EQ-DVMCA.

Finally, in order to further prove the advantages of EQ-DVMCA and the effectiveness of EQ-DVMCA in different working environments, we compared it experimentally with CGMS, HUA-LRR-MC-1.2 on the Bitbrains dataset. Table 13 shows the simulation results of the algorithm.

The comprehensive performance pairs of the algorithms are shown in Table 14.

As can be seen from Figure 10, the VMM value of EQ-DVMCA is the smallest. This is because EQ-DVMCA

TABLE 13. Simulation results of EQ-DVMCA and advanced algorithm in Bitbrains dataset.

Method	Index	2013/8/2	2013/8/4	2013/8/5	2013/8/8	2013/8/11	2013/8/15	2013/8/19	2013/8/20	2013/8/22	2013/8/24
EQ-DVMCA	VMM	2391	2319	2345	2316	2324	2276	2215	2243	2237	2246
	SLATAH (%)	5.37	5.33	5.35	5.29	5.24	5.16	5.18	5.19	5.16	5.13
	PDM (%)	0.15	0.14	0.14	0.13	0.12	0.11	0.12	0.09	0.10	0.09
	SLAV (10^{-5})	8.06	7.46	7.49	6.88	6.29	5.68	6.22	4.67	5.16	4.62
	EC (kW·h)	16.71	16.68	16.55	16.12	15.87	15.66	15.49	14.63	14.42	13.89
	ESV (10^{-2})	0.13	0.12	0.12	0.11	0.10	0.09	0.10	0.07	0.07	0.06
HUA-LRR-MC-1.2	VMM	3418	3394	3386	3297	3357	3316	3241	3219	3228	3241
	SLATAH (%)	5.81	5.79	5.80	5.74	5.76	5.73	5.71	5.70	5.68	5.69
	PDM (%)	0.29	0.29	0.28	0.29	0.26	0.24	0.25	0.27	0.23	0.24
	SLAV (10^{-5})	16.85	16.79	16.24	16.65	14.98	13.75	14.28	15.39	13.06	13.66
	EC (kW·h)	16.60	16.47	16.59	16.01	15.74	15.57	15.47	14.84	14.21	13.74
	ESV (10^{-2})	0.28	0.28	0.27	0.27	0.24	0.21	0.22	0.23	0.19	0.19
CGMS	VMM	2716	2649	2653	2546	2603	2504	2418	2432	2339	2462
	SLATAH (%)	6.47	6.49	6.45	6.41	6.34	6.24	6.21	6.23	6.15	6.13
	PDM (%)	0.17	0.18	0.16	0.15	0.14	0.15	0.13	0.13	0.12	0.14
	SLAV (10^{-5})	11.00	11.68	10.32	9.62	8.88	9.36	8.07	8.10	7.38	8.58
	EC (kW·h)	18.49	18.54	18.12	17.76	17.54	17.33	17.41	17.29	17.25	17.26
	ESV (10^{-2})	0.20	0.22	0.19	0.17	0.16	0.16	0.14	0.14	0.13	0.15

TABLE 14. Comparison of comprehensive performance between EQ-DVMCA and advanced algorithms in Bitbrains dataset.

Method	VMM	SLATAH (%)	PDM (%)	SLAV (10^{-5})	EC (kW·h)	ESV (10^{-2})
EQ-DVMCA	**22912**	**5.24**	**0.12**	**6.25**	156.02	**0.99**
HUA-LRR-MC-1.2	33097	5.74	0.26	15.16	**155.24**	2.36
CGMS	25322	6.31	0.15	9.30	176.99	1.65

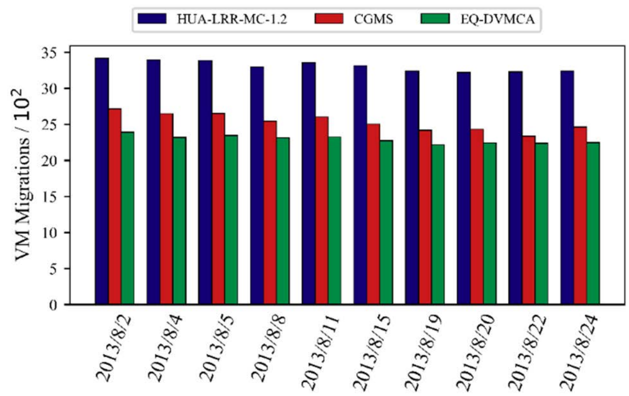

FIGURE 10. VMM comparison of EQ-DVMCA and advanced algorithm in Bitbrains dataset.

classifies the load status of physical hosts more carefully through the HLDA, to obtain a more accurate load status of the physical hosts. This method reduces the possibility of overloading the physical host again and reduces the probability of VMM. Secondly, when selecting VMs that need to be migrated, EQ-DVMCA can preferentially select more effective VMs for migration through CM-VMSA while avoiding the occurrence of invalid VMM, which greatly reduces the number of VMM. According to the calculation, compared with CGMS and HUA-LRR-MC-1.2, the VMM index of EQ-DVMCA decreased by 9.52% and 30.77%, respectively. This proves the feasibility of EQ-DVMCA on VMM.

As can be seen from Figure 11, the SLATAH value of EQ-DVMCA is the lowest, which indicates that the algorithm has the least overload violations in the data center at runtime. This is because EQ-DVMCA can predict the change of physical host workload through HCESEA more accurately, so that VMs can be migrated before physical host overload to avoid host overload. Secondly, when selecting suitable hosts for VMs for migration, EQ-DVMCA obtains the resource-demand scaling of physical hosts through RDS-VMPA, to consider the changing trend of VM resource demand on physical hosts and effectively prevent overload caused by workload fluctuation. RDS-VMPA ensures that physical hosts in data centers work properly for a long time after VMs are placed, preventing secondary overload. According to the calculation, compared with CGMS and HUA-LRR-MC-1.2, the SLATAH index of EQ-DVMCA decreased by 16.98% and 8.73%, respectively. This proves that EQ-DVMCA can effectively cope with workload fluctuations and maintain the normal working status of physical hosts in the data center for long periods of time.

As can be seen from Figure 12, EQ-DVMCA has the best performance in reducing PDM. This is because EQ-DVMCA is first optimized from the number of VMM, which fundamentally reduces the loss caused by VMM. Secondly, when selecting VM through CM-VMSA, EQ-DVMCA quantifies the effectiveness of VMM, which effectively avoids invalid VMM and further reduces the loss of computing resources caused by VMM to the data center. According to the calculation, compared with CGMS and HUA-LRR-MC-1.2, the PDM index of EQ-DVMCA decreased by 19.45% and 54.92% respectively. This shows the superiority of EQ-DVMCA in PDM index.

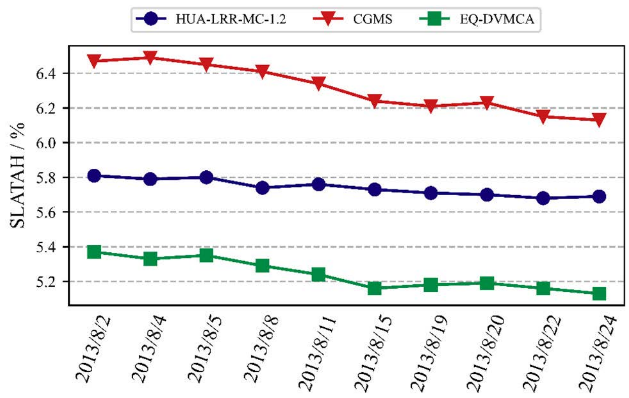

FIGURE 11. SLATAH comparison of EQ-DVMCA and advanced algorithm in Bitbrains dataset.

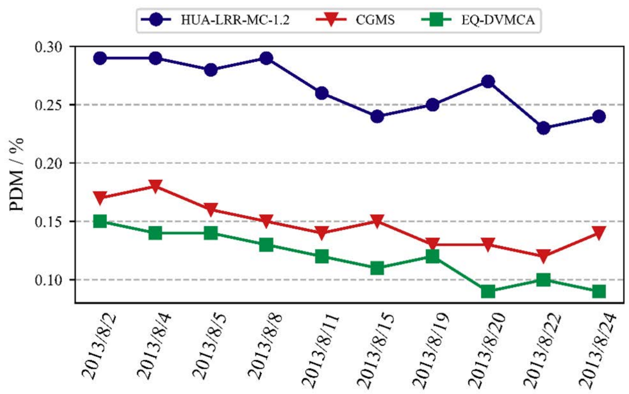

FIGURE 12. PDM comparison of EQ-DVMCA and advanced algorithm in Bitbrains dataset.

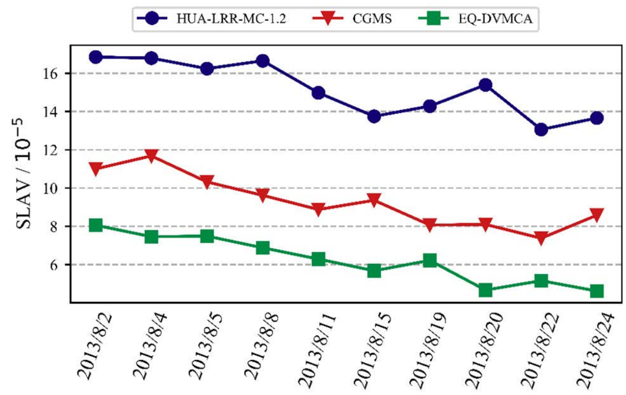

FIGURE 13. SLAV comparison of EQ-DVMCA and advanced algorithms in Bitbrains dataset.

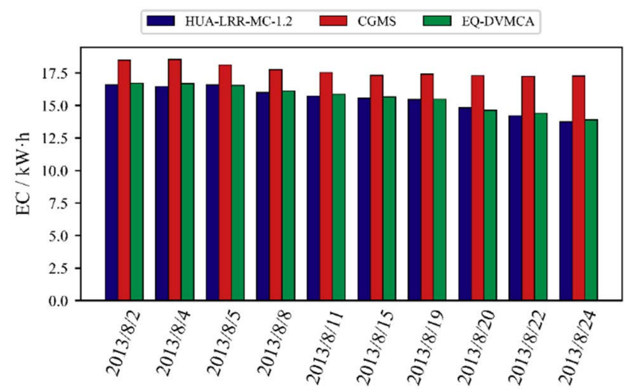

FIGURE 14. EC comparison of EQ-DVMCA and advanced algorithm in Bitbrains dataset.

As can be seen from Figure 13, The SLAV value of EQ-DVMCA is the lowest. Combined with the experimental results in Figure 11 and 12, it can be seen that EQ-DVMCA can deal with workload fluctuations more effectively and avoid migration of invalid VMs, so it can provide users with better QoS. According to the calculation, compared with CGMS and HUA-LRR-MC-1.2, the SLAV index of EQ-DVMCA decreased by 32.77% and 59% respectively. This proves that the EQ-DVMCA can guarantee QoS in the normal operation of data center.

According to Figure 13 and Figure 14, unlike the previous results, HUA-LRR-RC-1.2 has the lowest EC value, followed by EQ-DVMCA. However, the EC values of HUA-LRR-RC-1.2 are not significantly different from those of EQ-DVMCA. Based on Figure 13, it can be found that HUA-LRR-RC-1.2 actually sacrifices the QoS of the data center to improve the energy efficiency level of data center. Compared with HUA-LRR-RC-1.2, although EQ-DVMCA has less extra EC, it can effectively improve the QoS of the data center, so it is reasonable. The reasons why EQ-DVMCA can consume less energy and maintain a high level of QoS are as follows: first, considering the impact of VMM on EC, EQ-DVMCA improves the effectiveness of each VMM during operation and completes the consolidation of VMs in the data center with a small amount of VMM, thus reducing the extra EC generated by VMM. Second, when placing VMs, EQ-DVMCA allocates VMs to reasonable physical hosts through RDS-VMPA, which improves the resource utilization of physical hosts and effectively reduces the EC of physical hosts. According to the experimental results, the EC index of EQ-DVMCA was reduced by 11.85% compared with CGMS. This proves that it is feasible to reduce the number of VMM to save energy in the data center.

As can be seen from Figure 15, the ESV value of EQ-DVMCA is much lower than that of other algorithms. This shows that compared with other algorithms, the performance of EQ-DVMCA is optimal. According to the experimental results, compared with CGMS and HUA-LRR-MC-1.2, the ESV index of EQ-DVMCA decreased by 40.32% and 58% respectively. This result further proves that EQ-DVMCA achieves a balance between EC and QoS.

To sum up, EQ-DVMCA can not only significantly reduce the number of VMM, but also save energy and effectively ensure QoS. This enables cloud service providers reducing

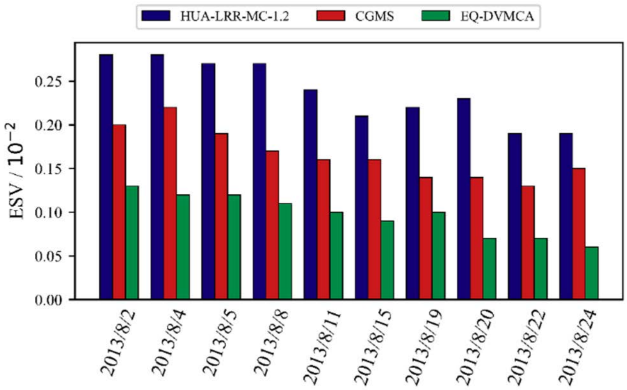

FIGURE 15. ESV comparison of EQ-DVMCA and advanced algorithm in Bitbrains dataset.

the cost of data centers and improving the user experience, further boosting cloud computing.

VI. CONCLUSION AND FUTURE WORK

VMC is one of the most effective methods to solve the high power consumption and low utilization of cloud data centers. However, large-scale VMM can result in additional workloads, SLAV, and considerable power consumption. Therefore, efficient VMC is one of the hotspots of current research. Existing studies have made great progress in this respect, however, the following problems still exist: first, the potential overload is not considered in the load detection of physical host; second, the resource-demand scaling of physical hosts is not considered during VM placement, which results in a lack of accuracy in selecting suitable hosts. In order to solve the above problems, this paper first constructed the GEC-VRCM model, which defines the specific process and the related attributes of VMC. GEC-VRCM is beneficial to improve the consolidation efficiency of virtual resources. Secondly, on the basis of this model, we propose EQ-DVMCA to realize the efficient consolidation of virtual resources. We proved the effectiveness of EQ-DVMCA through three experiments:

1. In the load prediction assessment, we compared HCESEA, CES, and ENN. The experimental results show that the prediction error of HCESEA is much smaller than that of CES and ENN under different working conditions. This proves that the load state of the physical host can be more accurately predicted using HCESEA.
2. We selected 12 VMC algorithms as the benchmark algorithm. Then, we compared these algorithms under six performance indexes. Experimental results show that compared with the benchmark algorithm, EQ-DVMCA has obvious advantages in terms of VMC.
3. We compared EQ-DVMCA with CGMS and HUA-LRR-MC-1.2, two advanced algorithms. The experimental results show that EQ-DVMCA can not only reduce the migration times of VMs but also maintain a high level of QoS with low EC and achieve a balance between EC and QoS.

To sum up, EQ-DVMCA can be well applied to VMC.

In future work, we will consider the impact of other resources on EC and SLAV.

REFERENCES

[1] T. Chen, Y. Zhu, X. Gao, L. Kong, G. Chen, and Y. Wang, "Improving resource utilization via virtual machine placement in data center networks," *Mobile Netw. Appl.*, vol. 23, no. 2, pp. 227–238, 2018.

[2] M. Tarahomi, M. Izadi, and M. Ghobaei-Arani, "An efficient power-aware VM allocation mechanism in cloud data centers: A micro genetic-based approach," *Cluster Comput.*, vol. 24, no. 2, pp. 919–934, Jun. 2021.

[3] S. F. Piraghaj, R. N. Calheiros, J. Chan, A. V. Dastjerdi, and R. Buyya, "Virtual machine customization and task mapping architecture for efficient allocation of cloud data center resources," *Comput. J.*, vol. 59, no. 2, pp. 208–224, 2016.

[4] F. Birol, *World Energy Outlook*. Paris, France: Int. Energy Agency, 2013.

[5] W. Li, Y. Xia, M. Zhou, X. Sun, and Q. Zhu, "Fluctuation-aware and predictive workflow scheduling in cost-effective infrastructure-as-a-service clouds," *IEEE Access*, vol. 6, pp. 61488–61502, 2018.

[6] L. Barroso and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synth. Lectures Comput. Archit.*, vol. 8, no. 3, pp. 1–154, 2013.

[7] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, and W. Lintner, "United States data center energy usage report," Tech. Rep., 2016.

[8] G. K. Shyam and S. S. Manvi, "Virtual resource prediction in cloud environment: A Bayesian approach," *J. Netw. Comput. Appl.*, vol. 65, pp. 144–154, Apr. 2016.

[9] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. M. Leung, "Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1688–1699, Jun. 2018.

[10] J. L. Berral, C. Wang, and A. Youssef, "AI4DL: Mining behaviors of deep learning workloads for resource management," in *Proc. 12th USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, 2020, pp. 1–7.

[11] T. Wang, S. Ferlin, and M. Chiesa, "Predicting CPU usage for proactive autoscaling," in *Proc. 1st Workshop Mach. Learn. Syst.*, Apr. 2021, pp. 31–38.

[12] 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.* Cham, Switzerland: Springer, Aug. 2014, pp. 306–317.

[13] N. T. Hieu, M. Di Francesco, and A. Yla-Jaaski, "Virtual machine consolidation with usage prediction for energy-efficient cloud data centers," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jun. 2015, pp. 750–757.

[14] D. Deng, K. He, and Y. Chen, "Dynamic virtual machine consolidation for improving energy efficiency in cloud data centers," in *Proc. 4th Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Aug. 2016, pp. 366–370.

[15] 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.

[16] 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.

[17] H. Wang and H. Tianfield, "Energy-aware dynamic virtual machine consolidation for cloud datacenters," *IEEE Access*, vol. 6, pp. 15259–15273, 2018.

[18] 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.

[19] X. Xiao, W. Zheng, Y. Xia, X. Sun, Q. Peng, and Y. Guo, "A workload-aware VM consolidation method based on coalitional game for energy-saving in cloud," *IEEE Access*, vol. 7, pp. 80421–80430, 2019.

[20] X. Zhang, T. Wu, M. Chen, T. Wei, J. Zhou, S. Hu, and R. Buyya, "Energy-aware virtual machine allocation for cloud with resource reservation," *J. Syst. Softw.*, vol. 147, pp. 147–161, Jan. 2019.

[21] S. Ilager, K. Ramamohanarao, and R. Buyya, "ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation," *Concurrency Comput., Pract. Exper.*, vol. 31, no. 17, pp. e5221.1–e5221.15, Apr. 2019.

[22] N. Patel and H. Patel, "Energy efficient strategy for placement of virtual machines selected from underloaded servers in compute cloud," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 6, pp. 700–708, Jul. 2020.
[23] A. Tarafdar, M. Debnath, S. Khatua, and R. K. Das, "Energy and quality of service-aware virtual machine consolidation in a cloud data center," *J. Supercomput.*, vol. 76, no. 11, pp. 9095–9126, Nov. 2020.
[24] R. Shaw, E. Howley, and E. Barrett, "Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers," *Inf. Syst.*, vol. 107, Jul. 2022, Art. no. 101722.
[25] R. Medara and R. S. Singh, "Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization," *Simul. Model. Pract. Theory*, vol. 110, Jul. 2021, Art. no. 102323.
[26] 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.
[27] E. S. Gardner, "Exponential smoothing: The state of the art—Part II," *Int. J. Forecasting*, vol. 22, no. 4, pp. 637–666, Oct. 2006.
[28] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 19, no. 2, pp. 179–211, 1990.
[29] 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.
[30] N. Patel and H. Patel, "A comprehensive assessment and comparative analysis of simulations tools for cloud computing," *Int. J. Eng. Comput. Sci.*, vol. 5, no. 11, pp. 18972–18978, Nov. 2016.
[31] K. Park and V. S. Pai, "CoMon: A mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Operating Syst. Rev.*, vol. 40, pp. 65–74, Jan. 2006.
[32] S. Shen, V. Van Beek, and A. Iosup, "Statistical characterization of business-critical workloads hosted in cloud datacenters," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 465–474.
[33] S. Anoep, C. Dumitrescu, D. Epema, A. Iosup, M. Jan, H. Li, and L. Wolters. *The Grid Workloads Archive: Bitbrains*. Accessed: Feb. 5, 2022. [Online]. Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains

WEI LI received the B.S. degree in software engineering from North China Electric Power University, Beijing, China, in 1987, where she is currently a Professor with the School of Control and Computer Engineering. Her research interests include smart grid software technology, network security, and machine learning.

QI FAN is currently pursuing the master's degree with the School of Control and Computer Engineering, North China Electric Power University, Beijing. His research interests include cloud computing, virtual machine consolidation, and network security.

WENCHAO CUI was born in January, in 1983. He received the bachelor's and doctor's degrees from North China Electric Power University, in 2007 and 2014, respectively. In 2018, he was employed as a master's Supervisor. He is mainly engaged in the research of power information security, aiming at the application requirements of internal and external network boundary security isolation, terminal security access, and internal security audit.

FANGFANG DANG worked at State Grid Henan Electric Power Company. She joined the company, in 2014, engaged in network security related work, published more than 20 papers, applied for four national patents, one software copyright, participated in a number of management innovations, won one provincial and ministerial first prize, two provincial company first prizes, and one second prize.

XIAOLIANG ZHANG received the bachelor's and master's degrees from North China Electric Power University, in July 2009 and April 2012, respectively. He has been engaged in the research of power information security and power intelligent software.

CHENG DAI works in the information and communication branch at State Grid Chongqing Electric Power Company. Engaged in information construction, business, and platform operation management of the company. He has published more than ten papers, applied for five national patents, participated in a number of management innovation and scientific and technological innovation competitions, and won one provincial and ministerial first prize and one second prize.