Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.039076
Article

Virtual Machine Consolidation with Multi-Step Prediction and Affinity-Aware Technique for Energy-Efficient Cloud Data Centers

Pingping Li* and Jiuxin Cao

School of Cyber Science and Engineering, Southeast University, Nanjing, 211189, China
*Corresponding Author: Pingping Li. Email: 230188114@seu.edu.cn
Received: 09 January 2023; Accepted: 10 April 2023; Published: 09 June 2023

Abstract: Virtual machine (VM) consolidation is an effective way to improve resource utilization and reduce energy consumption in cloud data centers. Most existing studies have considered VM consolidation as a bin-packing problem, but the current schemes commonly ignore the long-term relationship between VMs and hosts. In addition, there is a lack of long-term consideration for resource optimization in the VM consolidation, which results in unnecessary VM migration and increased energy consumption. To address these limitations, a VM consolidation method based on multi-step prediction and affinity-aware technique for energy-efficient cloud data centers (MPaAF-VMC) is proposed. The proposed method uses an improved linear regression prediction algorithm to predict the next-moment resource utilization of hosts and VMs, and obtains the stage demand of resources in the future period through multi-step prediction, which is realized by iterative prediction. Then, based on the multi-step prediction, an affinity model between the VM and host is designed using the first-order correlation coefficient and Euclidean distance. During the VM consolidation, the affinity value is used to select the migration VM and placement host. The proposed method is compared with the existing consolidation algorithms on the PlanetLab and Google cluster real workload data using the CloudSim simulation platform. Experimental results show that the proposed method can achieve significant improvement in reducing energy consumption, VM migration costs, and service level agreement (SLA) violations.

Keywords: Cloud computing; VM consolidation; multi-step prediction; affinity relationship; energy efficiency

1 Introduction

With the rapid development of global informatization, the energy consumption and CO_2 emissions of cloud data centers have increased rapidly [1–3]. Recent studies have shown that cloud data centers account for approximately 7% of the total world's electricity consumption, and this number is expected to increase to 13% by 2030 [4]. However, the resource utilization of physical hosts in cloud data centers is generally low. Research shows that the average CPU utilization is less than 20%, while an idle

host consumes 70% of its peak energy consumption [5]. Inadequate utilization of resources leads to a massive waste of resources and unnecessary energy consumption, which increases the operating costs of cloud service providers and reduces the energy efficiency of the whole society.

In cloud computing, virtualization technology [6] has been used to encapsulate various resources of a cloud data center into virtual machines (VMs) for tenants to use. It should be noted that one physical host can host multiple VMs, so maximizing the number of VMs hosted by each host can maximize the utilization of cloud data center resources and reduce energy consumption. However, when a host hosts too many VMs, it will be overloaded, which will decrease the quality of service (QoS). VM consolidation has been proven to be one of the most effective techniques for optimizing resource utilization and guaranteeing QoS, and it needs to solve three problems [7–9]: 1) host load detection: detect host overloaded or underloaded; 2) migration VM selection: for overloaded hosts, select appropriate VMs to migrate; and 3) VM placement: based on the consideration of QoS and energy consumption, select appropriate hosts to place migration VMs. Through the VM consolidation, resources in the cloud data center can be dynamically adjusted to optimize their allocation.

However, the usage of resources in the cloud data center changes dynamically. For VM consolidation, the dynamically changing characteristics of cloud resources pose a challenge for assessing host load. Related studies use only the current utilization of a single resource (i.e., CPU) when determining host load status [10,11]. In addition, many schemes consider multiple resources (i.e., CPU, memory, disk, and network), but ignore the dynamically changing characteristics of host load [12,13]. The above two solutions fail to accurately describe the host load state, which may lead to unnecessary VM migration, resulting in system performance degradation and additional resource overhead [14]. Second, how to select migration VMs from overloaded hosts and select the optimal placement host for migration VMs is another key problem to be solved. Most of the existing VM consolidation schemes [15–17] do not consider the long-term relationship between VMs and hosts when selecting migration VMs and placement hosts. For a dynamically changing cloud data center, this leads to excessive consolidation and reduces the effective usage of resources, resulting in increased energy consumption and resource waste.

To address the aforementioned challenges encountered in VM consolidation, a VM consolidation algorithm based on multi-step prediction and affinity-aware technique is proposed in this paper. Using the historical data of the cloud data center, an improved linear regression (LR) prediction algorithm is employed to predict the next-moment resource utilization of VMs and hosts. Then, the long-term resource demand in the future period is obtained through multi-step iteration prediction. The current and multi-step prediction values of resources determine the host load state, avoiding aggressive consolidation. In addition, an affinity relationship model between VMs and hosts based on multi-dimensional resources and multi-step prediction values is proposed. Based on the proposed affinity model, affinity-aware VM selection and affinity-aware VM placement methods are designed to improve the utilization of resources, reduce the number of active hosts in a cloud data center, and achieve the purpose of reducing energy consumption.

For the rest of this paper, Section 2 describes the related work, Section 3 introduces the affinity model and multi-step prediction, Section 4 introduces the VM consolidation based on the multi-step prediction and affinity model, Section 5 specifically describes the experimental evaluation, and Section 6 introduces the conclusions.

2 Related Work

Energy efficiency and efficient usage of resources have been attracting increasing attention in recent years. VM consolidation has become an effective method to improve resource utilization and energy efficiency in cloud data centers. Relevant research mainly focuses on three aspects, namely, how to determine whether a host is overloaded or underloaded, how to select the appropriate migration VMs from the overloaded host, and how to reselect the appropriate host for migration VMs.

In terms of host load status detection, some recent studies [18–20] used static threshold of CPU as the only criterion. However, in cloud data centers, the usage of resources is constantly changing. Therefore, the static threshold of a single resource cannot accurately assess the load status of a host, resulting in improper resource utilization. Zhou et al. [21] proposed a dynamic threshold method based on the K-Means clustering algorithm, which divided hosts into four types according to the dynamic threshold, improving the detection performance of dynamically changing resources. Beloglazov et al. [11] proposed an adaptive host load threshold algorithm to determine the host load state based on the statistical analysis of historical CPU data, which further improved the performance of adaptive resource dynamic changes. However, these methods only consider the current CPU utilization when performing the host load detection. Therefore, the reliability of overloaded and underload host detection is reduced, resulting in unnecessary migration and energy waste. In addition, the above studies did not consider the effects of other resources on overloaded or underloaded host determination. However, excessive utilization of any resources on a host will affect the performance of the host and reduce its QoS.

Many studies have applied predictive techniques to VM consolidation to perceive dynamic changes in resources. Haghshenas et al. [22] predicted the CPU utilization based on LR and selected underloaded hosts and placement hosts according to the predicted CPU value. Similarly, Farahnakian et al. [23] proposed a combined prediction algorithm based on the LR and K-nearest neighbor algorithm to predict the host CPU utilization, which was used for host overload detection. Moghaddam et al. [24] used machine learning-based methods and neural networks to predict the CPU utilization of a host and designed a dynamic threshold according to the CPU prediction value to determine whether a host was overloaded. By predicting future resource utilization, the VM consolidation can effectively reduce the number of service level agreement violations (SLAv). However, these methods only consider the one-step prediction of current single resource utilization and lack the long-term usage consideration of multi-dimensional resources in the future period. Therefore, it is difficult to obtain long-term optimal management and increase migration times.

In terms of migration VM selection, the random selection (RS) method, minimum migration time (MMT) method, maximum correlation (MC) method, and maximum utilization (MU) method have been widely used [11,25]. In addition, Li et al. [26] used memory page similarity characteristics to select migration VM. They selected VMs with high memory page similarity on different hosts as the VM to be migrated to reduce migration time and data. Masoumzadeh et al. [27] proposed an adaptive VM selection method by integrating multiple VM selection technologies, which dynamically selects the migration VM selection method based on the current host load and the fuzzy Q-learning method. Laili et al. [28] proposed a new iterative budget algorithm for migration VM selection, which established a reverse selection mechanism for randomly selected hosts to find suitable migration VMs. However, the aforementioned studies did not consider the relationship between the future resource usage of VMs and hosts when selecting migration VMs, which not only increases the migration overhead but also brings potential SLAv.

In terms of VM placement, many studies have considered VM placement as a well-known bin-packing problem [29], which is also known as an NP-hard problem. The Best Fit Decreasing (BFD) algorithm, First Fit Decreasing (FFD) algorithm, First Fit (FF) algorithm, and Best Fit (BF) algorithm are widely used heuristic algorithms to solve the VM placement problem [30,31], whose purpose is to reduce energy consumption by reducing the number of running hosts. However, there is much that can be improved on the above widely used heuristics in terms of full utilization of resources, and some researchers have improved them. Beloglazov et al. [11] improved BFD and proposed a power-aware BFD (PABFD) algorithm. PABFD minimizes the increase in power consumption through energy consumption awareness when selecting the placement host for migration VMs. Murtazaev et al. [32] combined the FF and BF algorithms and proposed a Sercon algorithm to minimize the number of hosts and migration costs. Farahnakian et al. [33] used the host load prediction strategy based on the regression model and combined it with the BFD method to determine the placement host. Bobroff et al. [34] proposed a measure-forecast-remap VM placement algorithm by measuring historical workloads and predicting future resource demands to reduce the number of active hosts. However, the aforementioned algorithms do not consider the relationship between the resources required by a migration VM and the remaining resources of a host in the future period. Namely, they optimize only the current usage of resources and only guarantee the optimal current resource allocation after the VM are placed. Since the long-term relationship between multi-dimensional resources in the future period is ignored, a host is prone to overload again in the future, resulting in increased migration costs. Therefore, revealing the relationship between the VM's resources and the host's remaining resources in the future period can contribute to the effective utilization of resources and reduce the probability of host overloading in the future period.

In addition, an increasing number of studies have modeled VM consolidation as a single- or multi-objective optimization problem, and various meta-heuristic algorithms have been proposed. Li et al. [35] obtained the optimal mapping relationship between VMs and hosts by simulating the foraging behavior of artificial bees. Besides, Li et al. [36] proposed a differential evolution meta-heuristic algorithm and applied it to VM consolidation. However, the above methods only consider the optimization of energy consumption and ignore the migration cost. Al-Moalmi et al. [37] proposed a grey wolf meta-heuristic algorithm considering the effective utilization of CPU and RAM, which effectively optimized energy consumption and QoS. However, the meta-heuristic algorithms are time-consuming in the evolutionary iteration process. For large-scale data centers, thousands of solution spaces need to be evaluated and compared, resulting in high algorithm complexity.

Existing VM consolidation studies have been mainly focused on reducing cloud data center energy consumption, SLAv, and migration overhead. The core of VM consolidation is to schedule and utilize various resources reasonably, but most studies have ignored the long-term relationship of the resources between VMs and hosts, and lack long-term consideration of resource optimization. To solve this limitation, this paper proposes an efficient and reliable strategy for VM consolidation with full consideration of migration overhead, energy consumption, and QoS. Compared with previous studies, for host load detection, the long-term load changes of the host are considered based on the proposed multi-step prediction, which effectively reduces the overload probability of the host. In addition, based on the proposed multi-step prediction and affinity model, on the one hand, consider the long-term relationship between the overloaded host and the VM when selecting migration VMs, which can quickly and accurately eliminate the overload state of the host with the minimum migration cost, and on the other hand, consider the long-term complementary relationship between the resources of the placement host and the VM when selecting placement hosts, which can make full use of resources.

3 Affinity Model Based on Multi-Step Prediction

3.1 System Framework

As shown in Fig. 1, the local monitor continuously monitors the resource utilization of VMs and hosts. The resource predictor uses the monitoring data to perform prediction. The resource scheduler obtains the current and future utilization of resources through the global monitor and calls the host load detection module, migration VM selection module, and VM placement module to complete the VM consolidation process. The host load detection module is used to determine whether a physical host is overloaded or underloaded. If the physical host is overloaded, the resource scheduler calculates the affinity value between the VM and the host using the migration VM selection module and selects migration VMs. The VM placement module selects the most suitable host in the cluster for the migration VMs.

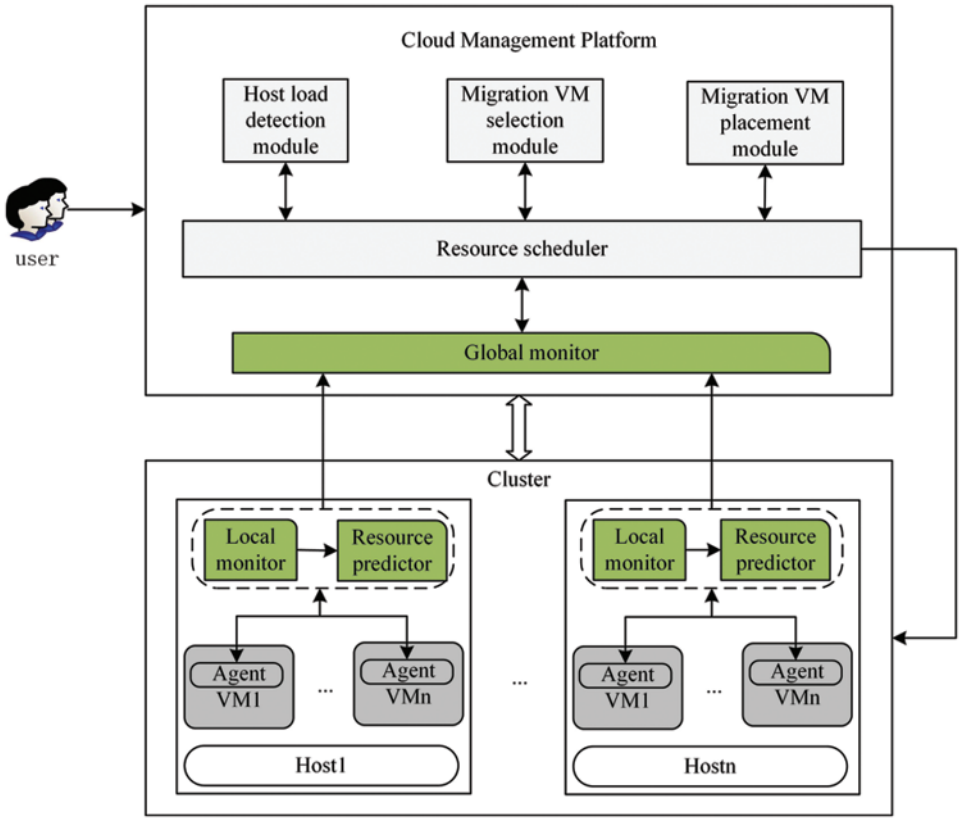


Figure 1: The system framework

3.2 Resource Model

In a cloud data center, assume that the number of hosts is n and the collection of hosts is expressed as $H = \{h_1, h_2, \cdots, h_j, \cdots, h_n\}$, where each host has D-dimensional resources. The $C_{h_j}^d$ is the capacity configuration of resource d of host h_j, where $d \in \{1, 2, \cdots, D\}$. Similarly, in a cloud data center, assume that the number of VMs is m and the collection of VMs is expressed as $V = \{v_1, v_2, \cdots, v_i, \cdots, v_m\}$, where each VM has D-dimensional resources. The $C_{v_i}^d$ is the capacity configuration of resources d of VM v_i, where $d \in \{1, 2, \cdots, D\}$.

Further, for host h_j, assume $U_t^d(h_j)$ is the utilization of resource d at moment t and as follows:

$$U_t^d\left(h_j\right) = \frac{1}{C_{h_j}^d} \sum\nolimits_{v_i \in VM\left(h_j\right)} U_t^d\left(v_i\right) \times C_{v_i}^d \tag{1}$$

where $U_t^d\left(v_i\right)$ indicates the utilization of resource d at moment t for VM v_i, and $VM(h_j)$ represents the VM collection of host h_j.

3.3 Resource Prediction Based on Improved LR

Assume that the last n utilization of resource d of host h_j are $U_{(t+1)-n}^d\left(h_j\right), \ldots, U_t^d\left(h_j\right)$. This paper uses the LR model to predict the next-moment resource utilization $U_{t+1}^d(h_j)$. Compared with other existing models, LR has great advantages in time complexity [38]. Resource utilization prediction based on the LR is defined as a linear prediction function, which can be expressed as Eq. (2).

$$U_{t+1}^d(h_j) = \beta_0 + \beta_1 U_t^d(h_j) \tag{2}$$

where β_0 and β_1 represent the regression coefficients estimated based on the latest n historical data samples. The most commonly used method to estimate coefficients β_0 and β_1 is the least square method, which minimizes the distance between the observed and predicted values to estimate the regression coefficient, as shown in Eqs. (3) and (4).

$$\beta_1 = \frac{\sum_i^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^n (x_i - \overline{x})^2} \tag{3}$$

$$\beta_0 = \overline{y} - \beta_1 \overline{x} \tag{4}$$

where $x \in \{x_1, x_2, \ldots, x_n\}$ and $y \in \{y_1, y_2, \ldots, y_n\}$, $\langle x, y \rangle$ represents the latest n observation samples of resource d, $\overline{x}$ is the mean of x, and $\overline{y}$ is the mean of y.

Previous research has shown that the absolute error of the least square method is not the smallest to solve the LR problem [39]. To improve the prediction accuracy, this study improves the above LR based on the new LR (NLR) model [40]. The basic idea of the NLR is to assume the model to pass through a straight line and a mean point. All sample points are required to be evenly distributed above and below the line. Subtract the slope of the assumed straight line from the slope connected by the sample point and intercept, and make their sum equal to 0. In this way, the slope and intercept can be solved by the simultaneous equation, namely coefficients β_0 and β_1. According to the new model, the regression coefficient is calculated by Eqs. (5)–(8). As above, $x \in \{x_1, x_2, \ldots, x_n\}$ and $y \in \{y_1, y_2, \ldots, y_n\}$, $\langle x, y \rangle$ represents the latest n observation samples of resource d, $\overline{x}$ is the mean of x, and $\overline{y}$ is the mean of y.

$$\beta_1 = \frac{\sum_{i=1}^n \frac{(y_i - \beta_0)}{x_i}}{n} \tag{5}$$

$$\overline{y} = \beta_0 + \beta_1 \overline{x} \tag{6}$$

There is a common factor β_0 in Eqs. (5) and (6), making Eqs. (5) and (6) simultaneous equations. Then, the coefficients β_0 and β_1 are calculated as follows:

$$\beta_1 = \frac{\sum_{i=1}^n \frac{y_i}{x_i} - \overline{y} \sum_{i=1}^n \frac{1}{x_i}}{n - \overline{x} \sum_{i=1}^n \frac{1}{x_i}} \tag{7}$$

$$\beta_0 = \overline{y} - \beta_1 \overline{x} \tag{8}$$

3.4 Multi-Step Prediction Based on Improved LR

The proposed multi-step prediction (MP) aims to predict the long-term usage of resource d in the future period $K \in \mathbb{N}$. MP can predict the usage of the kth step of resource d in advance, where $k \in \{1, 2, \ldots, K\}$ represents the MP prediction step length. Namely, the MP predicts the utilization $U^d_{t+k}(h_j)$ of the kth step based on the current utilization $U^d_t(h_j)$ of resource d. Specifically, the improved LR algorithm is employed to achieve the next k steps prediction values $U^d_{t+1}\left(h_j\right), U^d_{t+2}\left(h_j\right), \ldots, U^d_{t+k}(h_j)$ by iteration. That is, based on the value $U^d_t(h_j)$ at time t, NLR is used to calculate the predicted value $U^d_{t+1}\left(h_j\right)$ at time $t+1$, which is used as the observation sample at time $t+2$ to predict $U^d_{t+2}\left(h_j\right)$. In this way, the result of each prediction is used as the observation sample at the next time for iterative prediction to obtain multiple predicted values, which are calculated as follows.

$$U^d_{t+1}\left(h_j\right) = \beta^1_0 + \beta^1_1 U^d_t\left(h_j\right), \beta^1_1 = \frac{\sum_{i=1}^{n} \frac{y_i}{x_i} - \overline{y}\sum_{i=1}^{n} \frac{1}{x_i}}{n - \overline{x}\sum_{i=1}^{n} \frac{1}{x_i}}, \beta^1_0 = \overline{y} - \beta^1_1 \overline{x} \tag{9}$$

$$U^d_{t+2}\left(h_j\right) = \beta^2_0 + \beta^2_1 U^d_{t+1}\left(h_j\right), \beta^2_1 = \frac{\sum_{i=2}^{n+1} \frac{y_i}{x_i} - \overline{y}\sum_{i=2}^{n+1} \frac{1}{x_i}}{n - \overline{x}\sum_{i=2}^{n+1} \frac{1}{x_i}}, \beta^2_0 = \overline{y} - \beta^2_1 \overline{x} \tag{10}$$

$$U^d_{t+k}\left(h_j\right) = \beta^k_0 + \beta^k_1 U^d_{t+k-1}\left(h_j\right), \beta^k_1 = \frac{\sum_{i=k}^{n+k-1} \frac{y_i}{x_i} - \overline{y}\sum_{i=k}^{n+k-1} \frac{1}{x_i}}{n - \overline{x}\sum_{i=k}^{n+k-1} \frac{1}{x_i}}, \beta^k_0 = \overline{y} - \beta^k_1 \overline{x} \tag{11}$$

where t represents the current time, $t+k$ represents the kth time point in the future, and n represents the number of the latest observed samples at time t. When predicting the utilization $U^d_{t+k}(h_j)$ at time $t+k$, the predicted value $U^d_{t+k-1}(h_j)$ at time $t+k-1$ is taken as the observation sample, namely, $x_{n+k} = U^d_{t+k-2}\left(h_j\right)$ and $y_{n+k} = U^d_{t+k-1}(h_j)$. Assume that $\overline{x}$ is the mean of $x \in \{x_k, x_{k+1}, \ldots, x_{k+n}\}$, $\overline{y}$ is the mean of $y \in \{y_k, y_{k+1}, \ldots, y_{k+n}\}$, and $\langle x, y\rangle$ represents the latest n observation samples of resource d.

The MP algorithm is also applicable to the VM resources prediction. Thus, the next k steps prediction value $U^d_{t+1}(v_i), U^d_{t+2}(v_i), \ldots, U^d_{t+k}(v_i)$ of VM v_i can be obtained by the MP algorithm.

3.5 Affinity Model Based on Multi-Step Prediction

Since multiple VMs are usually running on a host, VMs with different requirements share and compete with each other for the host's resources. Therefore, each VM has a different effect on host overloading. To determine the key resources that cause host overload and eliminate host overload rapidly and effectively, this section proposes an affinity model by quantifying the relationship between the resources of VMs and hosts.

Assume that at time t, the multi-step prediction values of resource d of VM v_i and host h_j in the future period obtained by the MP are $P^d_{v_i} = \{U^d_{t+1}(v_i), U^d_{t+2}(v_i), \ldots, U^d_{t+K}(v_i)\}$ and $P^d_{h_j} = \{U^d_{t+1}\left(h_j\right), U^d_{t+2}\left(h_j\right), \ldots, U^d_{t+K}(h_j)\}$, respectively. So, $\varphi^d_{v_i h_j}$ can be expressed as Eq. (12).

$$\varphi^d_{v_i h_j} = \frac{cort(P^d_{v_i}, P^d_{h_j})}{dis(P^d_{v_i}, P^d_{h_j})} \tag{12}$$

$$cort\left(P^d_{v_i}, P^d_{h_i}\right) = \frac{\sum_{\delta k=1}^{K-1}(U^d_{t+1+\delta k}(v_i) - U^d_{t+\delta k}(v_i)) \cdot (U^d_{t+1+\delta k}(h_j) - U^d_{t+\delta k}(h_j))}{\sqrt{\sum_{\delta k=1}^{K-1}(U^d_{t+1+\delta k}(v_i) - U^d_{t+\delta k}(v_i))^2} \cdot \sqrt{\sum_{\delta k=1}^{K-1}(U^d_{t+1+\delta k}(h_j) - U^d_{t+\delta k}(h_j))^2}} \tag{13}$$

$$dis(P^d_{v_i}, P^d_{h_j}) = \sum_{\delta k}^{K} \left| U^d_{t+\delta k}(v_i) - U^d_{t+\delta k}(h_j) \right| \tag{14}$$

where $cort(P^d_{v_i}, P^d_{h_j})$ is the first-order correlation coefficient, indicating the similarity of changing trends of $P^d_{v_i}$ and $P^d_{h_j}$ in the future period K, and the larger its value is, the more similar the changing trends are; $dis(P^d_{v_i}, P^d_{h_j})$ represents the euclidean distance between $P^d_{v_i}$ and $P^d_{h_j}$ in the future period K, and the smaller its value is, the closer the resource usage in the future period is.

It should be noted that the larger the value of $\varphi^d_{v_i h_j}$ is, the closer the changing trends and usage of resources d of VM v_i and host h_j in the future period K are. Therefore, the affinity $\varphi_{v_i h_j}$ between VM v_i and host h_j can be calculated from $\varphi^d_{v_i h_j}$, that is, the sum of the affinity values of all resources, as follows:

$$\varphi_{v_i h_j} = \sum_{d=1}^{D} \varphi^d_{v_i h_j} \tag{15}$$

When selecting migration VMs from overloaded hosts, selecting the VM with the largest affinity value can rapidly and effectively reduce the overload degree of the host. The value of $\varphi_{v_i h_j}$ is calculated based on the predicted value of resources in the future period to prevent the host from being overloaded in the future period.

4 VM Consolidation Based on Multi-Step Prediction and Affinity-Aware

4.1 Host Load Detection Based on Multi-Step Prediction (MP-HLD)

The MP-HLD pseudo-code is given in algorithm 1. In the MP-HLD algorithm, a data center's host set $H = \{h_1, h_2, \cdots, h_j, \cdots, h_n\}$ is divided into three types: overload set H_o, normal-load set H_n, and underload set H_u. Assume that the maximum and minimum utilization thresholds of resource d on a host are denoted by Thr^d_{max} and Thr^d_{min}, respectively, and $k \in \{1, 2, \ldots, K\}$ is the prediction step length. Then, for host h_j, if the resource d satisfies the following conditions, the host h_j is classified as H_o:

(1) The utilization is larger than Thr^d_{max} for both the current and future periods; namely, $U^d_t(h_j) > Thr^d_{max}$ and $U^d_{t+k}(h_j) > Thr^d_{max}$;
(2) The current utilization is normal, but the utilization in the future period is larger than Thr^d_{max}; namely $U^d_t(h_j) \leq Thr^d_{max}$ and $U^d_{t+k}(h_j) > Thr^d_{max}$.

Accordingly, a host h_j is classified as H_u when it meets the following criteria:

(1) The utilization is less than Thr^d_{min} for both the current and future periods; namely $U^d_t(h_j) < Thr^d_{min}$ and $U^d_{t+k}(h_j) < Thr^d_{min}$;

(2) The current utilization is normal, but the utilization in the future period is less than Thr^d_{min}; namely $U^d_t(h_j) \geq Thr^d_{min}$ and $U^d_{t+k}(h_j) < Thr^d_{min}$.

The MP-HLD method detects the load status of hosts in advance based on the current and future period resource utilization. For overloaded hosts, it can reduce the SLAv and improve the QoS, and for underloaded hosts, it can reduce the number of running host and energy consumption.

Algorithm 1. The MP-HLD method

1. **Input:** H, Thr^d_{max}, Thr^d_{min}, $U^d_t(h)$, $U^d_{t+k}(h)$
2. **Output:** H_o, H_n, H_u
3. **for** h_j **in** H **do**
4. **for** $\forall d$ **in** D **do**
5. **for** $\forall k$ **in** K **do**

(Continued)

Algorithm 1. Continued

```
      if U_t^d(h_j) > Thr_max^d && U_{t+k}^d(h_j) > Thr_max^d then
        H_o ← h_j;
      else if U_t^d(h_j) ≤ Thr_max^d && U_{t+k}^d(h_j) > Thr_max^d then
        H_o ← h_j;
      else if U_t^d(h_j) < Thr_min^d && U_{t+k}^d(h_j) < Thr_min^d then
        H_u ← h_j;
      else if U_t^d(h_j) ≥ Thr_min^d && U_{t+k}^d(h_j) < Thr_min^d then
        H_u ← h_j;
      else
        H_n ← h_j;
      End if
    End for
  End for
End for
return H_o, H_n, H_u
```

4.2 Affinity-Aware VM Selection Algorithm (AF-VMS)

For overloaded hosts, any type of resource overload can become a bottleneck of the host's performance. Therefore, potential VMs need to be migrated from overloaded hosts to reduce the load. When selecting migration VMs, if the policy is unreasonable, it will lead to excessive VM migration, which increases energy consumption and affects the VM performance. The pseudo-code of AF-VMS is given in algorithm 2. For host h_j in the overloaded host set H_o, the affinity value $\varphi_{v_i h_j}$ between the host h_j and VM v_i running on it is calculated, and the VMs are sorted in descending order based on their $\varphi_{v_i h_j}$; then, migration VM is selected in order until host h_j returns to the normal load. For the underloaded host set H_u, all VMs in H_u will be migrated, and then the hosts will be shut down. Finally, the migration VM list migrate*VmList* is returned.

Algorithm 2. The AF-VMS algorithm

```
Input: overloaded hosts set H_o, underloaded hosts H_u
Output: a VM list for migration migrateVmList
migrateVmList ← NULL;
for h_j in H_o do
  mapList ← NULL;
  for v_i in VM(h_j) do
    φ_{v_i h_j} = ComputeAffinity(v_i, h_j); // using Eq. (15).
    mapList ← (φ_{v_i h_j}, v_i);
  End for
  descending sort for mapList;
  for v_i in mapList do
    VM(h_j).remove(v_i);
    migrateVmList.add(v_i);
    Compute U_t^d(h_j) and U_{t+k}^d(h_j); // use the MP to compute U_{t+k}^d(h_j)
    until U_t^d(h_j) < Thr_max^d && U_{t+k}^d(h_j) < Thr_max^d;
```

(Continued)

Algorithm 2. Continued

16. **End for**
17. **End for**
18. **for** h_j **in** H_u **do**
19. **for** v_i **in** $VM(h_j)$ **do**
20. migrate$VmList.add(v_i)$;
21. **End for**
22. **End for**
23. **return** migrate$VmList$

4.3 Affinity-Aware VM Placement Algorithm (AF-VMP)

Next, the AF-VMP is introduced in detail and the pseudo-code of AF-VMP is given in algorithm 3. The main objective of the AF-VMP algorithm is to select a host with the maximum affinity from the H_n as the placement host for migration VMs. The calculation method of the affinity value in this algorithm is different from that in the AF-VMS algorithm. In order to maximize the usage of resources and ensure the minimum number of running hosts in a cloud data center, in the AF-VMP algorithm, the affinity value between the VM and host is calculated based on the remaining resources of a host, as follows:

$$\overline{\varphi}^d_{v_i h_j} = \frac{cort\left(P^d_{v_i}, \overline{P}^d_{h_j}\right)}{dis\left(P^d_{v_i}, \overline{P}^d_{h_j}\right)} \tag{16}$$

$$cort\left(P^d_{v_i}, \overline{P}^d_{h_j}\right) = \frac{\sum_{\delta k=1}^{K-1}\left(U^d_{t+1+\delta k}(v_i) - U^d_{t+\delta k}(v_i)\right)\cdot\left(\overline{U}^d_{t+1+\delta k}(h_j) - \overline{U}^d_{t+\delta k}(h_j)\right)}{\sqrt{\sum_{\delta k=1}^{K-1}\left(U^d_{t+1+\delta k}(v_i) - U^d_{t+\delta k}(v_i)\right)^2}\cdot\sqrt{\sum_{\delta k=1}^{K-1}\left(\overline{U}^d_{t+1+\delta k}(h_j) - \overline{U}^d_{t+\delta k}(h_j)\right)^2}} \tag{17}$$

$$dis\left(P^d_{v_i}, \overline{P}^d_{h_j}\right) = \sum\nolimits_{\delta k}^{K}\left|U^d_{t+\delta k}(v_i) - \overline{U}^d_{t+\delta k}(h_j)\right| \tag{18}$$

where $\overline{P}^d_{h_j} = 1 - P^d_{h_j}$ and $\overline{U}^d_{t+\delta k}(h_j) = 1 - U^d_{t+\delta k}(h_j)$ indicate the remaining value of resource d of host h_j.

In addition to the consideration of the affinity value when selecting a placement host, after VM v_i is migrated to host h_j, the host's resource usage cannot exceed its configured capacity $C^d_{h_j}$, and the host cannot be overloaded after placing the migration VM. Therefore, the following conditions must be met.

(1) After a VM is migrated to a host, the resource requirements of all VMs running on the host cannot exceed the configured capacity of the host, namely:

$$C^d_{h_j} > \sum\nolimits_{v\in VM\left(h_j\right)} C^d_v + C^d_{v_i}, \forall d \in \{1, 2, \cdots, D\} \tag{19}$$

(2) After a VM is migrated to a host, the resource utilization cannot exceed the maximum threshold in the future period, namely:

$$U^d_{t+k}(h_j) + \frac{U^d_{t+k}(v)\times C^d_v}{C^d_h} < Thr^d_{max}, \forall d \in \{1, 2, \cdots, D\} \text{ and } \forall k \in \{1, 2, \ldots, K\} \tag{20}$$

(3) The affinity value is maximal between migration VM v_i and placement host h_j. The affinity value is calculated as follows:

$$\varphi_{v_i h_j} = \sum_{d=1}^{D} \overline{\varphi}_{v_i h_j}^{d} \tag{21}$$

Algorithm 3. The AF-VMP algorithm

Input: VMs to be migrated migrate$VmList$,
resource configuration of h_j $C_{h_j}^d, d \in \{1, 2, \cdots, D\}$,
resource demand of VM v_i $C_{v_i}^d, d \in \{1, 2, \cdots, D\}$,
prediction value of the migrated VM and placed host $U_{t+k}^d(v_i)$, $U_{t+k}^d(h_i)$, $k \in \{1, 2, \ldots, K\}$,
normal-load host set H_n.
Output: $\boldsymbol{X}$ -mapping matrix between VMs to be migrated and target hosts
for v_i **in** migrate$VmList$ **do**
  for h_j **in** H_n **do**
    $flag =$ **true**; $\varphi_{v_i h_{j\_min}} = \infty$;
    // testing resource capacity
    for $\forall d$ **in** D **do**
      if $C_{h_j}^d \le \sum_{v \in VM(h_j)} C_v^d + C_{v_i}^d$ **then**
        $flag =$ **false**;
      End if
    End for
    // testing resource overload
    for $\forall d$ **in** D **do**
      for $\forall k$ **in** K **do**
        if $U_{t+k}^d(h_j) + \dfrac{U_{t+k}^d(v_i) \times C_{v_i}^d}{C_{h_j}^d} \ge Thr_{max}^d$ **then**
          $flag =$ **false**;
        End if
      End for
    End for
    // select the target host based on the affinity value
    if ($flag$) **then**
      $\varphi_{v_i h_j} = ComputeAffinity(v_i, h_j)$; // using Eq. (21)
      if $\varphi_{v_i h_j} < \varphi_{v_i h_{j\_min}}$ **then**
        $\varphi_{v_i h_{j\_min}} = \varphi_{v_i h_j}$;
        $x_{ij} = (v_i, h_j)$;
      End if
    End if
  End for
  $\boldsymbol{X} \leftarrow x_{ij}$
End for
return $\boldsymbol{X}$

4.4 MPaAF-VMC Algorithm

The detailed process of MPaAF-VMC is shown in algorithm 4. The prediction values $U^d_{t+k}(h_j)$ and $U^d_{t+k}(v_i)$ of the host and VM are obtained by the MP method (line 3). According to the thresholds Thr^d_{max} and Thr^d_{min}, the hosts are divided into three types (i.e., overloaded hosts H_o, normal-load hosts H_n and underloaded hosts H_u) using the MP-HLD method (line 4). Then, based on the proposed affinity model, the algorithm selects migration VMs (line 5) and placement hosts (lines 6–16). When selecting a placement host, the normal-load host should be selected first (lines 6–8), if no suitable host is found, the underloaded host is selected (lines 9–13), and if also no suitable host can be found, a new host is activated (lines 14–16), which can further reduce the number of running hosts.

Algorithm 4. The MPaAF-VMC algorithm

1. **Input:** Host set H, VM set V
2. **Output:** $\boldsymbol{X}$ -mapping matrix between VMs to be migrated and target hosts
3. $(U^d_{t+k}(v_i),\ U^d_{t+k}(h_j)) = \text{MP}(H),\ k \in \{1, 2, \ldots, K\}$
4. $(H_o,\ H_n,\ H_u) = \text{MP-HLD}\ (H,\ U^d_{t+k}(v_i),\ U^d_{t+k}(h_j))$
5. migrate$VmList = \text{AF-VMS}(H_o,\ H_u,\ U^d_{t+k}(v_i),\ U^d_{t+k}(h_j)),\ k \in \{1, 2, \ldots, K\}$
6. **for** h_j **in** H_n **do**
7. $\quad \boldsymbol{X} = \text{AF-VMP}(\text{migrate}VmList, U^d_{t+k}(v_i),\ U^d_{t+k}(h_j))$
8. **End for**
9. **if** $\boldsymbol{X}$ does not change **then**
10. $\quad$ **for** h_j **in** H_u **do**
11. $\quad\quad \boldsymbol{X} = \text{AF-VMP}(\text{migrate}VmList, U^d_{t+k}(v_i),\ U^d_{t+k}(h_j))$
12. $\quad$ **End for**
13. **End if**
14. **if** $\boldsymbol{X}$ does not change **then**
15. $\quad \boldsymbol{X} =$ activate a new host for migration VM
16. **End if**
17. **return** $\boldsymbol{X}$

5 Experimental Evaluation

5.1 Experimental Setup

The performance evaluation of the VM consolidation algorithm should be performed on a large-scale cloud data center infrastructure, but it is difficult to perform and repeat experiments in an actual data center to evaluate the proposed algorithm. Therefore, CloudSim [25] simulator was used to evaluate the proposed algorithm. CloudSim is a widely used cloud computing environment simulation software, which can build a virtualized environment and simulate various resources of the cloud data center.

The cloud data center simulated in the experiment included two types of physical hosts HP ProLiant G4 and G5. Detailed information on the host configuration is given in Tables 1, and 2 shows their power consumption characteristics. Table 3 shows the four types of Amazon EC2 VMs used in the experiment [11].

Table 1: The experimental host configuration

Host type	CPU type	Frequency (GHz)	Core	RAM (GB)
HP ProLiant G4	Inter Xeon 3040	1.86	2	4
HP ProLiant G5	Inter Xeon 3075	2.66	2	4

Table 2: The experimental host power consumption at different load levels (W)

Host	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP G5	93.7	97	101	105	110	116	121	125	129	133	135

Table 3: The experimental VM configuration information

Type	CPU frequency (MIPS)	RAM (GB)
High-CPU medium instance	2,500	0.85
Extra large instance	2,000	3.75
Small instance	1,000	1.7
Micro instance	500	0.613

5.2 Performance Metrics

This paper evaluates the performance of the proposed algorithm using several main evaluation metrics of VM consolidation, namely energy consumption, migration cost, and SLAv.

Energy consumption: In this study, it is considered that the energy consumption of a cloud data center is generated by running hosts. Thus, for the same number of VMs, the fewer hosts there are, the lower the energy consumption is. Research has shown that the energy consumption of a host depends on the CPU, memory, network, and storage devices, and it is directly proportional to the CPU utilization [41]. Therefore, CPU utilization is usually used to express the relationship between host energy consumption and resource utilization. In this experiment, the host energy consumption was measured based on the real power consumption data provided by the SPECpower Benchmark results. The energy consumption of HP G4 and G5 servers under different load levels is shown in Table 2.

Migration cost: The performance of the VM will be degraded during the migration, which may lead to the SLAv. Related research [11] has shown that the average performance reduction during the VM migration can be evaluated as 10% of CPU utilization, which is used as an evaluation metric of migration costs in this study. In addition, the number of VM migrations is also used as an evaluation metric of migration costs.

SLA violations: SLA is a conformance agreement between the subscriber and the cloud service provider on services, which is an important metric to comprehensively measure the QoS. SLAv [11] can be measured in two independent aspects: SLA violations due to host overload (SLAVO) and

SLA violations due to VM migration (SLAVM). Therefore, the comprehensive SLAv is calculated as follows:

$$SLAv = SLAVO \times SLAVM \tag{22}$$

where SLAVO is the duration of the host overload, which is the percentage of time that the host's resource utilization is close to 100%. SLAVM represents the performance reduction caused by VM migration. Eqs. (23) and (24) are the calculation method of SLAVO and SLAVM, respectively.

$$SLAVO = \frac{1}{N}\sum_{j=1}^{N}\frac{T_{h_j}}{T_{a_j}} \tag{23}$$

$$SLAVM = \frac{1}{M}\sum_{i=1}^{M}\frac{C_{v_i}}{C_{a_i}} \tag{24}$$

In Eq. (23), T_{a_j} indicates the total running time of host h_j, T_{h_j} represents the duration during which the resource usage of host h_j approaches 100%, and N represents the number of hosts. In Eq. (24), C_{a_i} represents the total CPU requirement of VM v_i, C_{v_i} represents the unsatisfied resource request capacity caused by VM migration, and M represents the number of VMs. According to research [11], SLAVM can be calculated as 10% of CPU utilization.

5.3 Workload Data

This paper evaluates the effectiveness of the proposed methods based on two real-world workload datasets.

PlanetLab Data [42]: PlanetLab collected data from more than 1,000 VMs at more than 500 locations worldwide and tracked the CPU usage every 5 min. During March and April 2011, from the workload traces, we randomly selected 10 days of data, whose statistical characteristics are given in Table 4, showing that the average CPU utilization was much below 50%.

Table 4: PlanetLab data statistics

Date	Number of VMs	Mean (%)	St. dev. (%)	Median (%)
03/03/2011	1,052	12.31	17.09	15
06/03/2011	898	11.44	16.83	13
09/03/2011	1,061	10.70	15.57	13
22/03/2011	1,516	9.26	12.78	12
25/03/2011	1,078	10.56	14.14	14
03/04/2011	1,463	12.39	16.55	17
09/04/2011	1,358	11.12	15.09	15
11/04/2011	1,233	11.56	15.07	16
12/04/2011	1,054	11.54	15.15	16
20/04/2011	1,033	10.43	15.21	12

Google Cluster Data (GCD) [43]: GCD tracked the usage of multiple resources for approximately a month in May 2011. Table 5 shows the statistical characteristics of critical resources for 1,600 randomly selected VMs.

Table 5: The GCD statistics

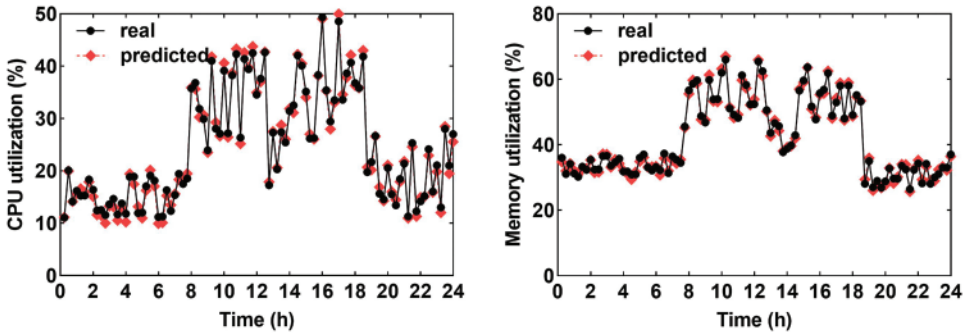
Resource	Number of VMs	Mean (%)	St. dev. (%)	Median (%)
CPU	1,600	21.23	12.78	18
Memory	1,600	18.57	15.83	13

5.4 Performance Evaluation

5.4.1 Multi-Step Prediction Performance

The GCD was used to evaluate the performance of the proposed MP scheme. The host resource utilization data were collected periodically by capturing the CPU and memory consumed by all tasks when the GCD workload was running. The collected data samples were divided into training and validation datasets. For observation data of a length n, the ratio of training to validation data was 3:2. The mean absolute percentage error (MAPE) and the root mean square error (RMSE) were used to evaluate the accuracy of the prediction model.

The prediction performance of MP for the six-step prediction (corresponding to the predicted value at 30 min) is presented in Fig. 2. Experimental results showed that the predicted value of the MP was close to the real value.



(a) (b)

Figure 2: Comparison of the six-step prediction of CPU and memory. (a) CPU. (b) Memory

The results of the multi-step prediction are shown in Fig. 3, which indicates that the MAPE and RMSE values were the smallest for one-step prediction. Then, as the step length increases, the MAPE and RMSE values gradually increased because the error of multi-step prediction accumulated as the step length increases, resulting in a decline in the prediction accuracy. However, the maximum increase in the MAPE value for the CPU and memory was only 0.041 and 0.019, respectively. Therefore, the error values of one- and twelve-step predictions were not very different. This result proved the correctness of the MP algorithm. In other experiments, this paper set the prediction step length to six to ensure the accuracy of the prediction.

5.4.2 Performance of MP-HLD

The experiment was conducted based on cloud data centers with 100 to 1500 hosts and used the PlanetLab and GCD workloads data to perform VM consolidation every 5 min. We recorded 24 h of data as experimental results. The static detection method and dynamic detection method were used as

the benchmark for comparison. For using the static threshold to detect host load (THR-HLD), the host overload threshold was set to 80%, while for using the dynamic threshold to detect host load (LR-HLD), the method of local regression (LR) was used to predict the host resource usage and determine the host load state. The experimental analysis focused on the impact of MP-HLD, THR-HLD, and LR-HLD on the number of overloaded, underloaded, and active hosts under different workloads and cloud data center sizes. The test results are shown in Figs. 4–6.

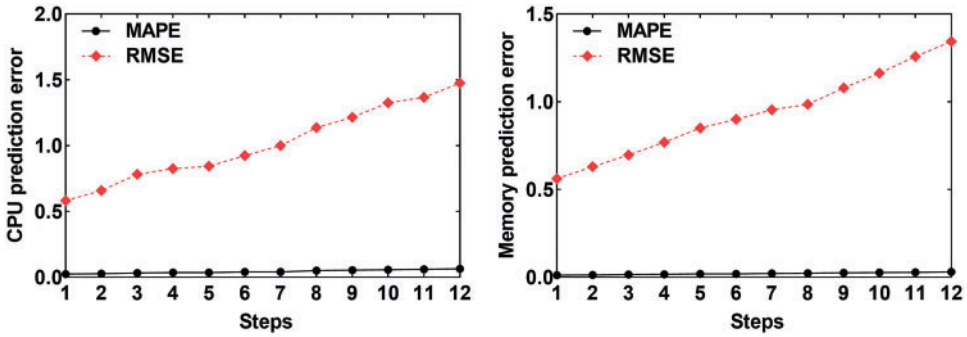


Figure 3: Comparison of the multi-step prediction. (a) CPU prediction error. (b) memory prediction error

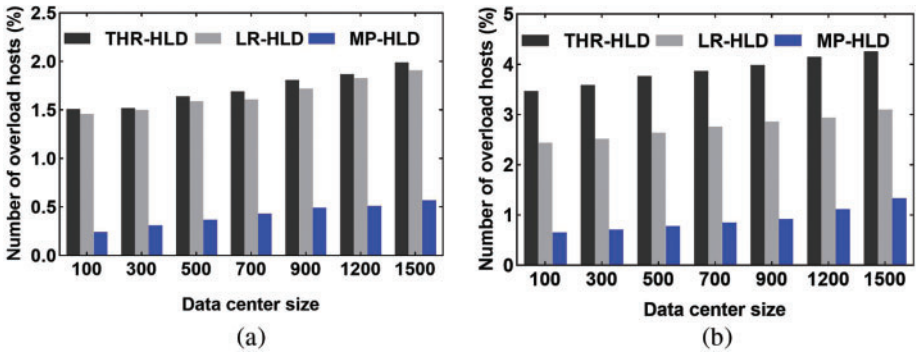


Figure 4: Comparison of the number of overloaded hosts. (a) PlanetLab. (b) GCD

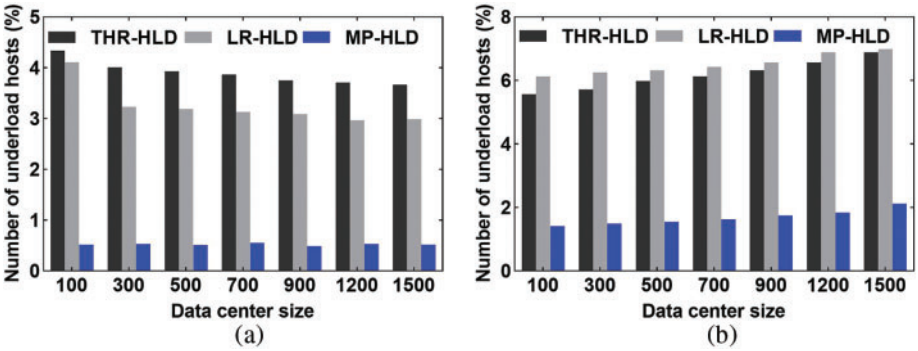


Figure 5: Comparison of the number of underloaded hosts. (a) PlanetLab. (b) GCD

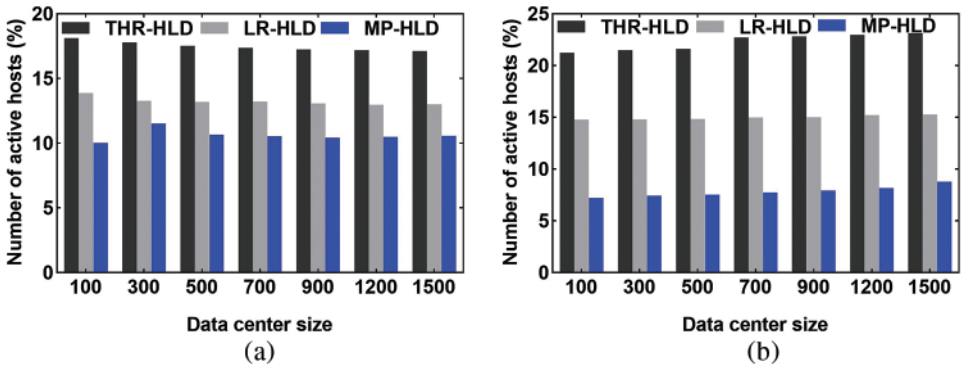


Figure 6: Comparison of the number of active hosts. (a) PlanetLab. (b) GCD

It can be seen from Figs. 4 and 5 that MP-HLD has the least number of hosts in overload and underload host detection. For a data center with 1,500 nodes, compared with static and dynamic threshold schemes, the number of overloaded hosts detected by MP-HLD reduced by 71.4% and 70.2%, respectively, and the number of underloaded hosts detected by MP-HLD reduced by 85.8% and 82.6%, respectively, in PlanetLab workload tests. The static and dynamic threshold algorithms determine the load status of a host only according to the current or short-term resource utilization. Even if the host load fluctuates occasionally or changes in a short time, the host will be judged to be overloaded or underloaded, which leads to misjudgment of the host load status, so more overloaded or underloaded hosts are detected. MP-HLD uses multi-step prediction to predict the host resource usage in the future period. When detecting the host load state, it considers the current and future period of resource usage, which effectively reduces the misjudgment caused by load changes through long-term forecasting of resource usage.

Fig. 6 shows that MP-HLD requires the least number of active hosts. Based on the above analysis, it can be seen that THR-HLD and LR-HLD methods do not fully consider the long-term dynamic changes of cloud data center resources in the future period, so they detect more underloaded or overloaded hosts. During every consolidation, more overloaded and underloaded hosts lead to more VM migrations, and more migration VMs need to activate more hosts to place these VMs.

5.4.3 VM Selection Performance

This section evaluates the performance of the AF-VMS. In addition to using the AF-VMS method, several widely used VM selection algorithms proposed in [25] were used for comparison, namely, the MU, MMT, and MC algorithms. The experiment performed VM consolidation every 5 min using PlanetLab and GCD workloads, and the data over 24 h were used as the experimental results. The comparisons of energy consumption and SLAv of different VM selection algorithms under different VM numbers are presented in Figs. 7 and 8. As shown in Figs. 7 and 8, the AF-VMS method had the lowest energy consumption and SLAv among all methods regardless of the number of VMs.

Next, the effects of the number of VM migrations and the average number of overloaded hosts per consolidation were analyzed. As shown in Figs. 9 and 10, AF-VMS achieved better performance than the other VM selection strategies. Fig. 9 shows that AF-VMS has the minimum number of migration VMs, achieving reductions of 39.5% and 40.6% on PlanetLab and GCD, respectively, for the cluster with 800 VMs compared to the suboptimal MC scheme. In addition, as presented in Fig. 10, the

average number of overloaded hosts per consolidation was reduced by 29.2% and 23.3% on PlanetLab and GCD, respectively, for the cluster with 800 VMs compared to the suboptimal MC scheme.

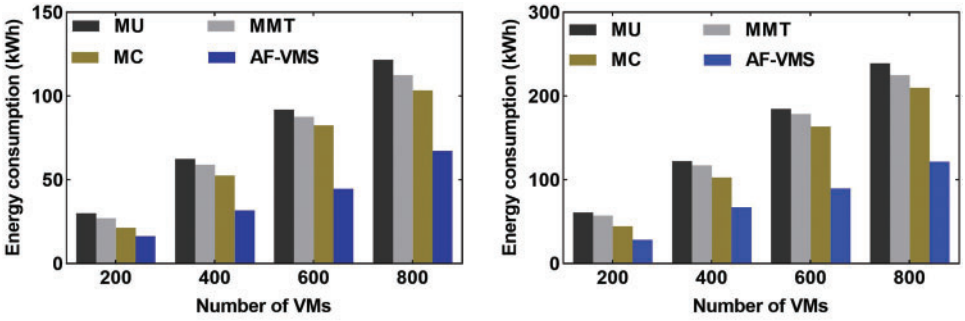


Figure 7: Comparison of energy consumption. (a) PlanetLab. (b) GCD

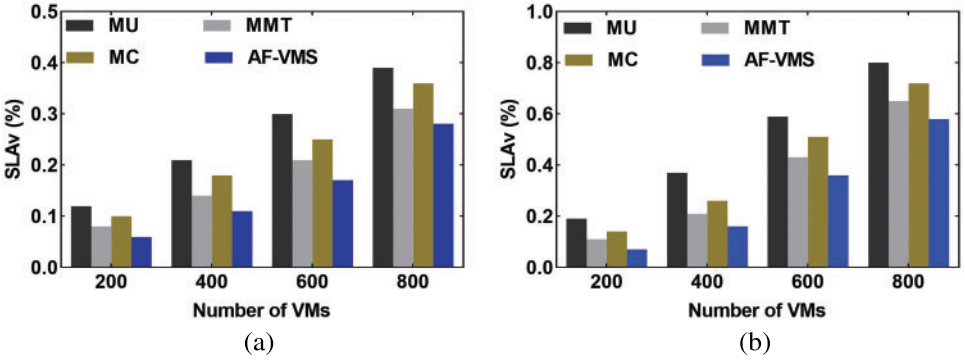


Figure 8: Comparison of SLAv. (a) PlanetLab data. (b) GCD

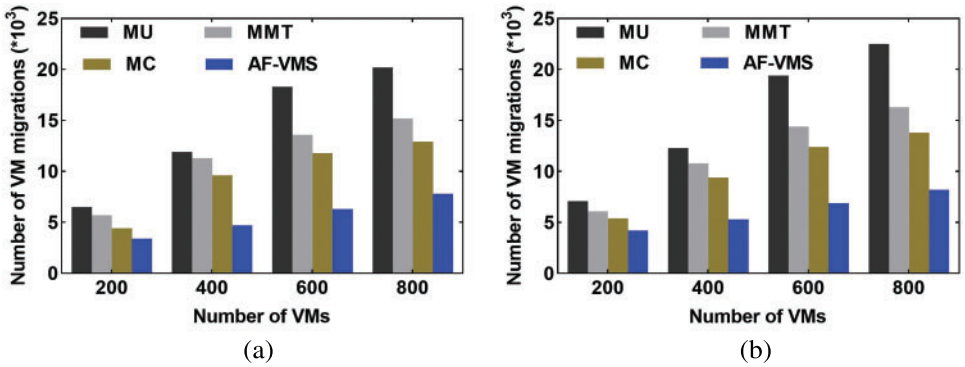


Figure 9: Comparison of the number of VM migrations. (a) PlanetLab data. (b) GCD

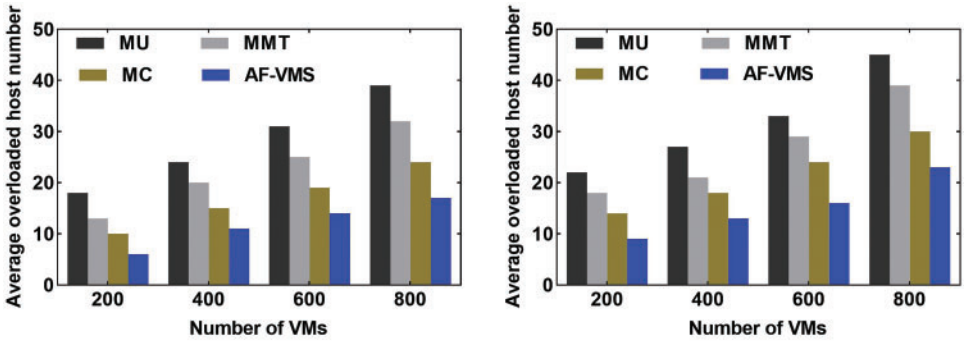


Figure 10: Comparison of the average number of overloaded hosts. (a) PlanetLab. (b) GCD

The above performance improvement was mainly attributable to the affinity model based on the multi-step prediction. The multi-step prediction could correctly predict the resource utilization of the host and VM in the future period. Based on the predicted value, the affinity model calculated the relationship between the VM and the overloaded host, which could accurately select a VM with the greatest impact on the overloaded host. This is because the resource change trend of the selected migration VM and the overloaded host was the closest in the future period, and the resource usage of the selected migration VM and the overloaded host was also the closest in the future period. Consequently, the overloaded host could accurately and rapidly recover to normal load with the minimum number of VM migrations, and it could avoid being overloaded in the future period. This effectively reduces the number of VM migrations and the probability of host overload.

5.4.4 VM Placement Performance

Next, the performance of AF-VMP was evaluated. The host overload detection algorithms THR-HLD, LR-HLD, MP-HLD, the VM selection algorithms MU, MMT, MC, and the widely used heuristic VM placement algorithms FF and PABFD constituted VM consolidation schemes as the comparison benchmarks.

The comparison results of energy consumption, SLAv, and VM migration under different VM selection algorithms and VM placement algorithms are shown in Figs. 11–13. Compared with other schemes, the scheme proposed in this paper significantly reduced power consumption and SLAv with minimal migration cost.

As shown in Fig. 11, based on the proposed MP, AF-VMS, and AF-VMP VM consolidation scheme, for the data center with 800 hosts and 1000 VMs, the energy consumption per day using the PlanetLab and GCD was 117 kWh and 163 kWh, respectively, and compared with the other algorithms, the maximum reductions were 44% and 42%, respectively. This was because when a VM was relocated, the proposed affinity model considered the complementarity of resources, which maximizes the usage of the host resources. Therefore, the number of running hosts was effectively reduced, thus reducing energy consumption.

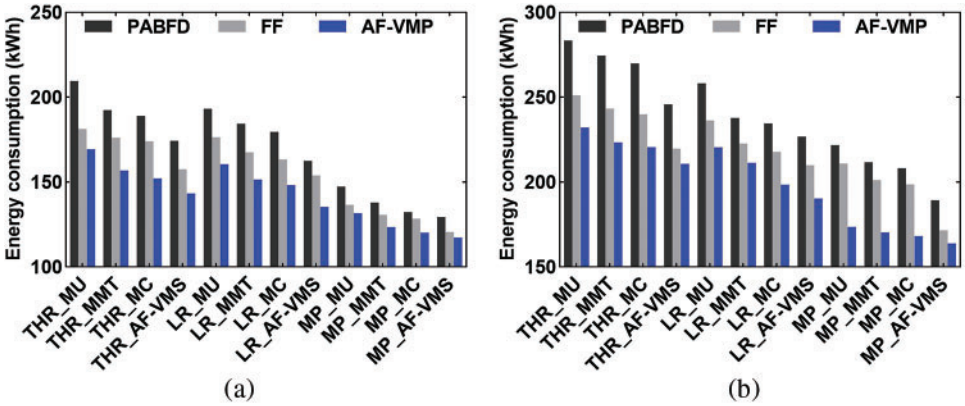


Figure 11: Comparison of energy consumption. (a) PlanetLab. (b) GCD

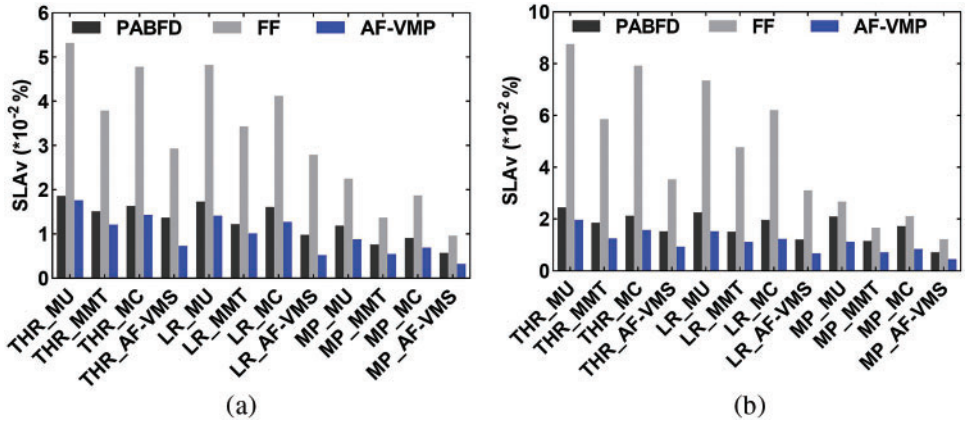


Figure 12: Comparison of SLAv. (a) PlanetLab. (b) GCD

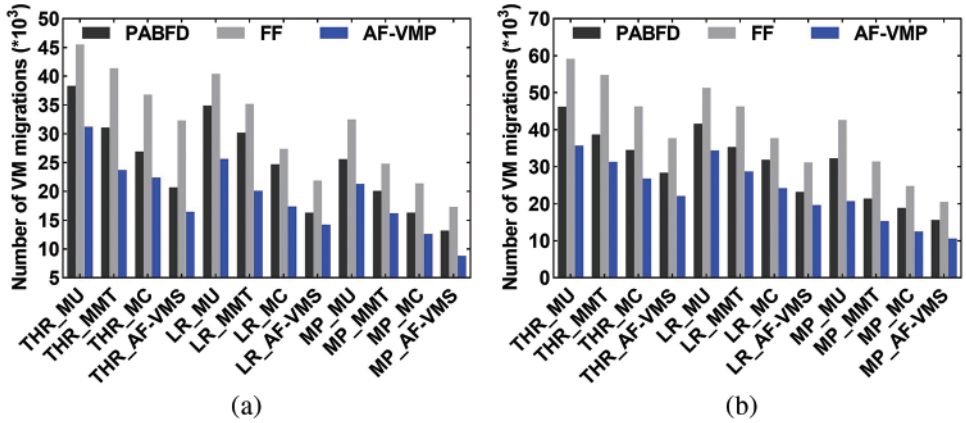


Figure 13: Comparison of the number of VM migrations. (a) PlanetLab. (b) GCD

Fig. 12 shows that the proposed algorithm had the minimum SLAv, which proved that it achieved a good performance in terms of QoS assurance. The affinity model considered the VM impact on

the overloaded host when selecting migration VMs, which could restore overloaded hosts to normal-load hosts with the fastest speed and the smallest migration cost. In addition, AF-VMS and AF-VMP used the resource utilization prediction in the future period as a measurement factor, which effectively reduced the probability of a host being overloaded again in the future period, thus effectively reducing the SLAv. However, the FF algorithm considered only whether the host had sufficient resources when selecting placement hosts, and ignored the effects of SLAv and energy consumption. Therefore, the SLAv generated by the FF algorithm was the highest among all algorithms. The PABFD algorithm mainly focused on the impact on energy consumption and did not consider SLAv.

Similarly, the proposed algorithm had the best performance in terms of migration cost compared with other schemes. As shown in Fig. 13, the FF algorithm migrated the most VMs among all algorithms. This is because the FF algorithm considered only whether the host had sufficient resources at the current moment when migrating VMs, and ignored both the fact of whether the host would be overloaded in the future period and the future dynamic changing trend of host resources. However, based on the affinity model, the proposed algorithm fully considered the impact of these two factors and selected the migration VM that had the greatest impact on the overloaded host which reduced the number of VM migrations. In addition, based on the affinity model, when selecting a host for VM placement, the proposed algorithm not only considered the complementary characteristics of the resources between VM and host but also ensured that the host's resources were within a normal load range at the current and future periods, thus effectively avoiding the occurrence of host overload and reducing the VM migration number.

5.4.5 MPaAF-VMC Performance

Considering the diversity of cloud computing environments, different cluster scales were used to further test the comprehensive performance of the proposed comprehensive algorithm MPaAF-VMC. Table 6 shows the cluster scales. MPaAF-VMC was compared with the LR + MC + PABFD and LR + MC + FF algorithms. In addition, the TVRSM algorithm [44] was used as a comparison benchmark. The TVRSM selected migration VMs from overloaded hosts based on the multiple correlation coefficient (MCC) and selected placement hosts based on the minimum power increasing strategy (MPIS) that selected the host with the smallest power increase.

Table 6: Cluster scale configuration

Scale	Number of servers	Number of VMs
Scale-1	100	200
Scale-2	500	800
Scale-3	900	1,200

The performance of different schemes on PlanetLab for various cluster scales is presented in Table 7. The results show that the proposed algorithm has the best performance regarding all metrics among all algorithms. As the cluster scale increased, the energy consumption of each solution increased, but the proposed algorithm had the least energy consumption, VM migrations, and overloaded host number among all methods. The performance improvement was due to the long-term optimization considerations of resources based on the affinity model, which guaranteed long-term optimization and full utilization of resources. In contrast, the other algorithms considered only the current resource utilization, ignoring dynamic changes and the long-term relevance of cloud

resources when selecting migration VMs and placement hosts, so they could not guarantee the long-term optimal utilization of resources, which increased energy consumption. Although the TVRSM algorithm selected migration VM based on the MCC strategy that considered the resource correlation between the host and VM, it ignored the long-term correlation of resource usage, which led to more VM migrations and overloaded or underload hosts detection.

Table 7: Performance of various schemes under different cluster scale scenarios

Scale	Algorithm	Energy consumption (kWh)	SLAv (%)	Migration number	Avg. overloaded host number
Scale-1	LR + MC + PABFD	74	1.8	3,186	7
	LR + MC + FF	61	4.4	4,687	11
	TVRSM	53	1.2	2,518	9
	MPaAF-VMC	45	0.36	1,282	2
Scale-2	LR + MC + PABFD	154	2.6	13,762	24
	LR + MC+FF	146	5.2	19,021	31
	TVRSM	123	1.9	9,874	19
	MPaAF-VMC	96	0.49	5,965	10
Scale-3	LR + MC + PABFD	216	3.5	19,747	35
	LR + MC + FF	189	6.3	28,148	49
	TVRSM	167	2.4	14,493	28
	MPaAF-VMC	135	0.62	9,126	17

6 Conclusions

To reduce the energy consumption and SLAv of a cloud data center, this paper proposes an MPaAF-VMC algorithm. The MPaAF-VMC uses a multi-step prediction scheme based on the improved LR prediction model to forecast resource demand in the future period. In addition, an affinity model between the host and VM is developed based on the multi-step prediction result. During the VM consolidation, based on the affinity model, for selecting migration VMs, the VM with the maximum affinity value is selected from overloaded hosts, which not only effectively eliminates the overload state of a host but also decreases the number of VM migrations and SLAv, and for selecting placement hosts, the host with the maximum affinity value is selected for migration VMs, which ensures that VMs are deployed on the minimum number of hosts to reduce energy consumption. Finally, many simulation experiments are conducted using PlanetLab and GCD workload data, and the results show that the performance of the MPaAF-VMC algorithm is significantly better than other algorithms. However, as the prediction step length increases, the prediction error will increase, which can be a limitation of this work. This is due to the accumulation of errors with the increase in prediction step length.

With the development of big data and artificial intelligence, higher requirements are put forward for cloud computing systems. In addition to energy consumption challenges, data processing faces new challenges. In future work, during the VM consolidation, the affinity relationship between VMs and between hosts could be analyzed to achieve better energy consumption optimization and data processing performance. In addition, the accuracy of multi-step prediction and the prediction step

length could be further explored to perceive potential risks as early as possible and plan resources to meet the challenges.

Acknowledgement: Thanks to our tutors and researchers for their assistance and guidance.

Funding Statement: The project is supported by the National Natural Science Foundation of China (62172089, 61972087, 62172090).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Ahmad, A. Shahnaz, M. Asfand-e-Yar, W. Khalil and Y. Bano, "A service level agreement aware online algorithm for virtual machine migration," *Computers Materials & Continua*, vol. 74, no. 1, pp. 279–291, 2023.

[2] S. S. Panwar, M. M. S. Rauthan and V. A. Barthwal, "A systematic review on effective energy utilization management strategies in cloud data centers," *Journal of Cloud Computing*, vol. 11, no. 1, pp. 1–29, 2022.

[3] S. Bharany, S. Sharma, O. I. Khalaf, G. M. Abdulsahib, A. S. AlHumaimeedy *et al.,* "A systematic survey on energy-efficient techniques in sustainable cloud computing," *Sustainability*, vol. 14, no. 10, pp. 6256, 2022.

[4] M. Avgerinou, P. Bertoldi and L. Castellazzi, "Trends in data centre energy consumption under the European code of conduct for data centre energy efficiency," *Energies*, vol. 10, no. 10, pp. 1470, 2017.

[5] R. Birke, L. Y. Chen and E. Smirni, "Data centers in the cloud: A large scale performance study," in *Proc. of the 5th IEEE Int. Conf. on Cloud Computing*, Honolulu, HI, USA, pp. 336–343, 2012.

[6] J. Wang, H. Gu, J. Yu, Y. Song, X. He *et al.,* "Research on virtual machine consolidation strategy based on combined prediction and energy-aware in cloud computing platform," *Journal of Cloud Computing*, vol. 11, no. 25, pp. 1–18, 2022.

[7] R. Shaw, E. Howley and E. Barrett, "Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers," *Information Systems*, vol. 107, no. 10, pp. 101722, 2022.

[8] X. Liu, J. Wu, L. Chen and L. Zhang, "Energy-aware virtual machine consolidation based on evolutionary game theory," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 10, pp. 1–16, 2022.

[9] W. Li, Q. Fan, W. Cui, F. Dang, X. Zhang *et al.,* "Dynamic virtual machine consolidation algorithm based on balancing energy consumption and quality of service," *IEEE Access*, vol. 10, pp. 80958–80975, 2022.

[10] K. Haghshenas, A. Pahlevan, M. Zapater, S. Mohammadi and D. Atienza, "MAGNETIC: Multi-agent machine learning-based approach for energy efficient dynamic consolidation in data centers," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 30–44, 2022.

[11] 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 and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[12] A. R. Madireddy and K. Ravindranath, "Dynamic virtual machine relocation system for energy-efficient resource management in the cloud," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 3, pp. 1–17, 2022.

[13] B. M. P. Moura, G. B. Schneider, A. C. Yamin, H. Santos, R. H. S. Reiser *et al.,* "Interval-valued fuzzy logic approach for overloaded hosts in consolidation of virtual machines in cloud computing," *Fuzzy Sets and Systems*, vol. 446, no. 4, pp. 144–166, 2022.

[14] I. Takouna, E. Alzaghoul and C. Meinel, "Robust virtual machine consolidation for efficient energy and performance in virtualized data centers," in *Proc. 2014 IEEE Int. Conf. Green Comput. Commun*, Taipei, Taiwan, pp. 470–477, 2014.

[15] W. Lin, W. Wu and L. He, "An on-line virtual machine consolidation strategy for dual improvement in performance and energy conservation of server clusters in cloud data centers," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 766–777, 2022.
[16] J. Zeng, D. Ding, K. kang, H. Xie and Q. Yin, "Adaptive DRL-based virtual machine consolidation in energy-efficient cloud data center," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2991–3002, 2022.
[17] P. Li and J. Cao, "A virtual machine consolidation algorithm based on dynamic load mean and multi-objective optimization in cloud computing," *Sensors*, vol. 22, no. 23, pp. 9154, 2022.
[18] C. H. Hsu, K. D. Slagter, S. C. Chen and Y. C. Chung, "Optimizing energy consumption with task consolidation in clouds," *Information Sciences*, vol. 258, no. 3, pp. 452–462, 2014.
[19] Y. Liu, Y. Zhao, J. Dong, L. Li, C. Wang *et al.,* "I-Neat: An intelligent framework for adaptive virtual machine consolidation," *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 13–26, 2022.
[20] C. Mastroianni, M. Meo and G. Papuzzo, "Probabilistic consolidation of virtual machines in self-organizing cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 125–228, 2013.
[21] Z. Zhou, J. Abawajy, M. Chowdhury, Z. Hu, K. Li *et al.,* "Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms," *Future Generation Computer Systems*, vol. 86, no. 6, pp. 836–850, 2018.
[22] K. Haghshenas and S. Mohammadi, "Prediction-based underutilized and destination host selection approaches for energy-efficient dynamic VM consolidation in data centers," *Journal of Supercomputing*, vol. 76, no. 12, pp. 10240–10257, 2020.
[23] F. Farahnakian, T. Pahikkala, P. Liljeberg and J. Plosila, "Energy-aware consolidation algorithm based on K-nearest neighbor regression for cloud data centers," in *Proc. IEEE/ACM 6th Int. Conf. Utility Cloud Computing*, Dresden, Germany, pp. 256–259, 2013.
[24] S. M. Moghaddam, M. O'Sullivan, C. G. Walker, S. F. Piraghaj and C. P. Unsworth, "Embedding individualized machine learning prediction models for energy efficient VM consolidation within Cloud data centers," *Future Generation Computer Systems*, vol. 106, no. 1, pp. 221–233, 2020.
[25] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.
[26] 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 Generation Computer Systems*, vol. 84, no. 5, pp. 98–107, 2018.
[27] S. S. Masoumzadeh and H. Hlavacs, "Integrating VM selection criteria in distributed dynamic VM consolidation using fuzzy Q-learning," in *Proc. of the 9th Int. Conf. on Network & Service Management*, Zurich, Switzerland, pp. 332–338, 2014.
[28] Y. Laili, F. Tao, F. Wang, L. Zhang and T. Lin, "An iterative budget algorithm for dynamic virtual machine consolidation under cloud computing environment," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 30–43, 2021.
[29] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang *et al.,* "Power-aware and performance-guaranteed virtual machine placement in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1385–1400, 2018.
[30] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *IEEE Transactions on Services Computing*, vol. 3, no. 4, pp. 266–278, 2010.
[31] A. Beloglazov, J. Abawajy and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.
[32] A. Murtazaev and S. Oh, "Sercon: Server consolidation algorithm using live migration of virtual machines for green computing," *IETE Technical Review*, vol. 28, no. 3, pp. 212–231, 2011.

[33] F. Farahnakian, P. Liljeberg and J. Plosila, "LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers," in *Proc. 2013 39th Euromicro Conf. on Software Engineering and Advanced Applications*, Santander, Spain, pp. 357–364, 2013.
[34] N. Bobroff, A. Kochut and K. Beaty, "Dynamic placement of virtual machines for managing sla violations," in *Proc. of 10th IFIP/IEEE Int. Symp. on Integrated Network Management*, Munich, Germany, pp. 119–128, 2007.
[35] Z. Li, C. Yan, L. Yu and X. Yu, "Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method," *Future Generation Computer Systems*, vol. 80, no. 7, pp. 139–156, 2018.
[36] Z. Li, X. Yu, L. Yu, S. Guo and V. Chang, "Energy-efficient and quality-aware VM consolidation method," *Future Generation Computer Systems*, vol. 102, no. 5, pp. 789–809, 2020.
[37] A. Al-Moalmi, J. Luo, A. Salah and K. Li, "Optimal virtual machine placement based on grey wolf optimization," *Electronics*, vol. 8, no. 3, pp. 283, 2019.
[38] S. Islam, J. Keung, K. Lee and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computing Systems*, vol. 28, no. 1, pp. 155–162, 2012.
[39] R. W. Hawley and N. C. Gallagher, "On Edgeworth's method for minimum absolute error linear regression," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 2045–2054, 1994.
[40] Y. Yu, "Research of a new method for solving linear regression," in *Proc. of the 2018 Int. Conf. on Transportation & Logistics, Information & Communication, Smart City (TLICSC 2018)*, Chengdu, China, pp. 484–488, 2018.
[41] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, no. 1, pp. 1–15, 2009.
[42] K. Park and V. S. Pai, "CoMon: A mostly-scalable monitoring system for planetLab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006.
[43] N. Rashid and U. K. Yusof, "Literature survey: Statistical characteristics of google cluster trace," in *Proc. 2018 Fourth International Conf. on Advances in Computing, Communication & Automation (ICACCA)*, Subang Jaya, Malaysia, pp. 1–5, 2018.
[44] W. Zhu, Y. Zhuang and L. Zhang, "A three-dimensional virtual resource scheduling method for energy saving in cloud computing," *Future Generation Computer Systems*, vol. 69, no. 6, pp. 66–74, 2017.