IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 2, 2020, accepted October 13, 2020, date of publication October 20, 2020, date of current version November 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032428

Balanced Cloud Edge Resource Allocation Based on Conflict Conditions

LEILEI ZHU[1,2], JIAHUI FENG[1], DAN LIU[1], HONGWEI YANG[1], ZHENGQI BAI[1], XIAOLONG SONG[1], AND LI LI[1]

[1]College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
[2]College of Medical Information, Changchun University of Chinese Medicine, Changchun 130117, China

Corresponding authors: Li Li (ll@cust.edu.cn) and Dan Liu (ld@cust.edu.cn)

This work was supported in part by the Science and Technology Project of the 14th Five-Year Plan of the Education Department of Jilin Province under Grant JJKH20200801KJ, and in part by the Key Project of the 14th Five-Year Plan of the Education Department of Jilin Province under Grant ZD19019.

ABSTRACT Under a multiscenario environment with frequent bursts of data in the edge cloud, the resource allocation in the edge cloud will affect the stability of its nodes. To address this problem, a balanced virtual resource allocation model based on conflict conditions is proposed. Based on a thorough study of the similarity between task attributes and resources used by the host, the concept of a task backlog is implemented to achieve a preliminary balanced allocation of tasks; thus, a conflict condition based on the remaining resources of the physical and virtual machines is proposed. Further, a matrix of phased conflict coefficients is built to establish a balanced virtual machine allocation model. The results of experiments comparing the performance of the proposed model with that of other existing models indicate that the proposed model can reduce the virtual machine scheduling time by up to 8.33%, save up to 6.25% of host energy consumption, and improve the algorithm efficiency by 20.47% compared with the other algorithms. To avoid the local optimal problem caused by dynamic virtual machine migration, an improved ant colony algorithm is combined with the above model, and concepts of a pheromone volatility factor and suppression factor are implemented to optimize the pheromone measurement function and ensure that the virtual machine migration path is globally optimal. Overall, the model reduces the conflict rate of resources on the physical machine and can maintain stable operation under CPU usage fluctuation, thus realizing a balanced allocation of node resources.

INDEX TERMS Backlog rate, conflict conditions, edge cloud, pheromone, resource allocation.

I. INTRODUCTION

Edge clouds are cloud computing platforms built on edge infrastructure based on the core of cloud computing technology and edge computing capabilities. These platforms may have a distributed cloud architecture (i.e., peer model) [1]. Each edge cloud can be independently calculated or coordinated with a central cloud. Fig. 1 illustrates the architecture of the edge cloud.

With the increase in the number of Internet of things devices, the traditional way of processing delivered resources by the central cloud has been unable to satisfy the requirements of tasks generated by edge devices. An established edge cloud model can accept the task of edge device delivery for the Internet of things so that the edge device can obtain the nearby resources dynamically and quickly to meet the needs of the edge device with multiple delays. However, due to the limited computing power of the edge cloud service nodes on the terminal side, while processing multiple scenarios and burst data, an uneven resource allocation will cause node stability problems. Some studies have attempted to solve these problems. You *et al*. [2] proposed the M/M/n queue model and a queue optimization model for mobile edge computing. The model designed a distributed energy-saving strategy from the aspects of clock frequency configuration and offload strategy; however, it could not design large-scale tasks, as property settings can easily cause tasks to pile up. In another study, a new nonlocally convergent particle swarm optimization algorithm (DNCPSO) was proposed that uses nonlinear inertial weights and searches in the corresponding direction to complete the selection and mutation operations of a group, thereby reducing the allocation time of certain resources [3]. However, while the algorithm improves execution efficiency, the system energy consumption and

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

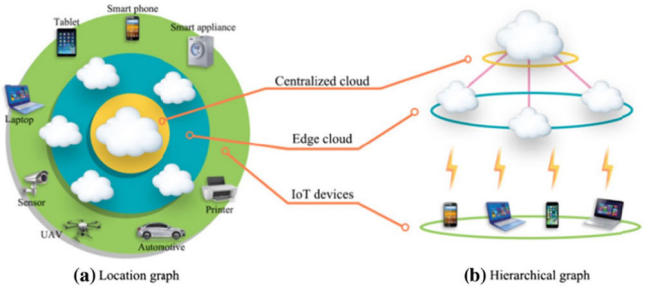


FIGURE 1. Edge cloud system architecture.

node load increase due to repeated inspection of inertial weights.

Base on the mixed integer nonlinear programming problem of the approximate optimal protest model of resource allocation strategy was proposed [4], the strategy takes the minimum value of each mobile user's transmission rate as the condition, establishes a fair resource allocation strategy with the goal of maximizing network throughput, and solves the approximate optimal solution by establishing time-sharing variables.However, this algorithm does not take into account energy efficiency and energy consumption, which is likely to cause large energy loss and cost waste. Yang *et al.*[5] proposed a green resource allocation model based on DRL framework, which can achieve the goal of green energy conservation in the network through a well-trained network training, and realize the effective allocation of resources, However, the simulation scenario is kind of simple and does not consider the burst of multi-user data.

Xu *et al*. [6] proposed the relaxed ant colony algorithm and considered the energy consumption of task scheduling.Tang *et al*. [7] proposed a task scheduling method aiming at optimizing the energy consumption of terminal mobile devices.Jian *et al*. [8] proposed a rational resource scheduling method based on the improved Chaotic bat algorithm. The metrics of above-mentioned references include resource utilization, load ratio,completion time, energy consumption and response time. However, while the above approach attempts to address resource constraints on mobile devices without placing tasks on edge cloud servers, doing so is more advantageous for delay-sensitive tasks.

In summary, the aforementioned research lacks consideration of the computing capacity and load capacity of edge nodes. Therefore, this study proposes a balanced resource allocation model based on conflict conditions (the SimCMA model). First, the similarity between tasks and resources used by the host is determined, and the concept of a backlog rate is used to ensure the rational allocation of tasks in the initial stage of resource allocation. Then, a staged conflict reduction coefficient matrix is constructed, based on which a resource allocation model is established. This model takes into account the characteristics of data mutation and uses an improved ant colony algorithm to ensure that the virtual machine migration path is globally optimal.

II. RELATED WORK

In this section, we discuss some of the resource allocation strategies developed for edge cloud computing.

Jararweh *et al.* [9] introduced a trustworthy smart city service delivery solution at the edge of the network. The scheme improves the availability, reliability and security of smart city terminal applications with the support of intrusion detection system by using the cooperation mode between distributed edge servers and intermediate nodes.The simulation results show that the request service efficiency of the proposed method is increased by 39.2% in a highly data-intensive environment, and the latency is reduced by 62.2% in a slightly data-intensive environment.However, the security of the system needs to be strengthened

Zhang *et al*. [10] realized task allocation according to the size of virtual machines while focusing on the goal of reducing energy consumption by using the least number of physical resources. However, this method does not consider the effect of physical machine resource usage on virtual machine allocation.

Guo *et al.* [11] combine game theory with cloudlet and propose a balanced task scheduling scheme, which reduces the workload of the centralized cloud and thus reduces energy consumption and computing cost.The strategy considers three attributes, namely response time, energy efficiency and service delay,but does not consider load balancing of nodes.However, if cluster nodes can reach the equilibrium state, the efficiency of edge nodes can be improved effectively.

Aujla *et al*. [12] combined support vector machine, two-stage game theory and software definition network, proposed a method to manage computing resources in edge environment.

Balasubramanian *et al.* [13] proposed an autonomous energy management architecture, called Droplet that learns the power-related statistics of the device.the paper based on the incentive strategy, the paper improved the device energy consumption strategy for different situations. The simulation experiment showed that the proposed framework effectively generated 10% cloud service benefits. However,time cost and calculation cost of the framework are high.

Wei *et al*. [14] based on the mobile edge computing method, focuses on the user's computing unloading problem in wireless cellular network, so as to optimize the computing unloading optimal solution.In this paper, model-free reinforcement learning (RL) framework is used to estimate and learn the value function of each mobile user's interaction with the environment, and then the optimal unload action is calculated.

Ganesan *et al*. [15] based their study on the mapping relationship between SaaS-based applications and virtual machines. From the perspective of the size of virtual machines, they focused on the joint capabilities of virtual machines but did not consider the fact that eliminating redundancy in a virtual machine affects the resource allocation

efficiency to a certain extent. In addition, the study considered only the situation of an application mapped to a single virtual machine.

Yang *et al*. [16] proposed a multiterminal load balancing algorithm based on the minimum weight of the load. This method reflects the comprehensive load of each terminal node according to different types of load factors, dynamically adjusts the weight of the node according to the above results, and calculates the actual load of the previous node. Thus, the method ensures that the node with the minimum load in a cluster provides services, and a multiterminal adaptive load balance is realized; however, the method has the disadvantage of a long average delay.

In summary, the aforementioned resource allocation models mainly target traditional cloud computing centers and have certain limitations on the delay and allocation efficiency when processing concurrent resources. In fact, the edge cloud has limited node computing capabilities and high latency requirements. Therefore, maximizing the use of physical machine resources and system operating efficiency is the key to the stable operation of edge cloud service nodes.

To address the abovementioned issues, this study proposes the SimCMA model, which considers the effect of physical machine thresholds on resource allocation. In the model, the network transmission delay is ignored, and the threshold of unloading tasks to an edge cloud node is set. Then, a task set with high similarity to the resources used by the host is separated, and the concept of a task backlog rate is proposed to realize the balanced unloading of tasks to the node, thereby reducing the startup rate of the virtual machine. Thus, the initial equilibrium of virtual resource allocation is achieved. Subsequently, this study considers that the resource utilization of the physical machine has the optimal threshold. The threshold value should be dynamically adjusted according to the demand of resources and the real load situation at the nodes. The core aim of this study is to establish a conflict model between resources and the remaining resources of the physical machine according to a set threshold. A conflict model between a resource and the remaining resources of the physical machine is established to filter conflicting resources.

Thereafter, the SimCMA model is optimized on the basis of the number of virtual machine migrations and system energy consumption. Based on the premise that the number of edge cloud hosts has a certain limit, the model reduces virtual machine redundancy in nodes and improves the number of virtual machine migrations by using an ant colony algorithm to consider the cost of virtual machine migration. Some existing studies on this aspect were conducted from the perspective of physical hosts, while others focused on the computing power of virtual machines [17], [18]. The content of a physical host mostly includes memory, space, CPU operation, and so on [19], [20], whereas that of the virtual machine mostly includes I/O, CPU, and so on. The optimal allocation of resources is an np-hard problem, and heuristic algorithms are often used to find approximate optimal solutions. Classic algorithms include particle swarm, genetic, and simulated annealing algorithms [21], [22].

III. NUMERICAL MODEL

A. FORMAL DESCRIPTION OF THE PROBLEM

In this study, the SimCMA model is established in the following steps: (1) Initial task offload: based on the similarity between the task and the resources used by the host, the concept of a backlog rate is used to offload tasks requested by users to edge service nodes. (2) Establishment of resource allocation model: based on the computing power of the host, a matrix of conflict coefficients is established for the physical machine, a resource allocation model is built according to the conflict conditions, and a balanced allocation of resources is achieved. (3) According to the conflict model between the physical machine and a resource and using the improved ant colony algorithm, the optimal path of virtual machine migration is optimized to reduce the cost of virtual machine migration.

The core aim of this study is the establishment of a resource allocation model. The main medium for resource allocation is virtual machines [23], [24]. Although a virtual machine plays an important role because it is connected to the tasks of the end user, relevant security and manageability are the focus of consideration. The end user is the producer of data, and the virtual machine can be regarded as a tool for mapping the end tasks to each server node. The purpose of the edge cloud is to achieve interconnection between the physical hosts of the task. The physical nodes of the edge cloud data center bear a large amount of workload on the edge. The task is encapsulated by the virtual machine in a node, and the encapsulation process requires a certain cost.

The balanced allocation strategy of virtual machines will directly affect resource allocation. Fig. 2 illustrates the role of the resource allocation model.

B. REASONABLE EXPERIMENTAL ASSUMPTIONS

In establishing the resource allocation model, to ensure the authenticity of experiments, we first propose the following hypothetical conditions:

1) To design a resource allocation model, we consider cloud task mapping and virtual machine scheduling of different nodes in the same edge cloud data center. Because the edge cloud center is limited by a certain communication distance and cost, this condition is reasonable.
2) The effect of network bandwidth on cloud task execution is not considered in the establishment of the model. We assume that when an edge user task method is offloaded to the edge cloud data center node, the transmission is lossless; that is, we ignore the transmission delay and consider the task to reach the edge cloud center node directly.
3) Each host in the edge cloud center has its own stable independent CPU, memory unit, and other attributes,

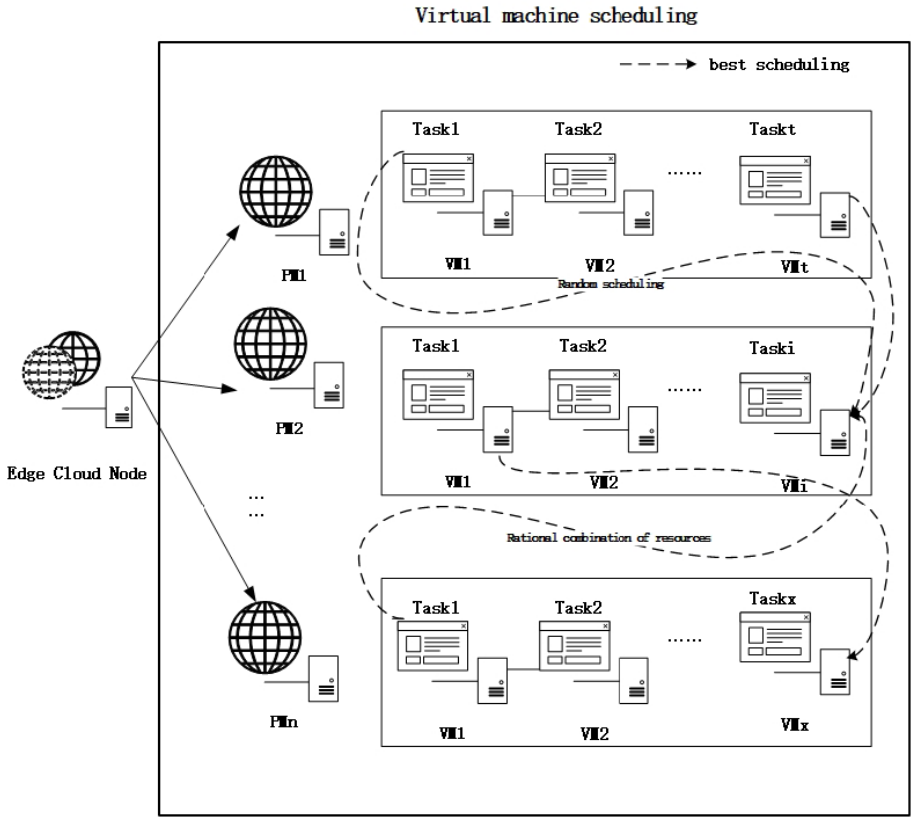


FIGURE 2. Schematic of the virtual machine allocation strategy.

which follow certain constraint relationships with each other. We consider that the decision condition for the host to be able to perform a task is that the maximum number of CPU Mips (Mips: million instructions per second) and the memory footprint required for a task are less than the remaining amount of these two resources in the host.

4) Finally, we assume that each physical host in the cluster has the same computing power and energy consumption.

C. RESOURCE ALLOCATION NUMERICAL MODEL

The proposed SimCMA model focuses on maximizing performance while reducing energy consumption. The premise of the experiments is reasonable offloading of tasks by modeling the use of an ideal experimental environment, irrespective of factors such as hardware defects, the number of virtual machines (hereinafter referred to as VMs), security, and fault tolerance. When implementing task mapping, to improve performance to the maximum extent possible, a VM with better performance should perform multiple tasks as much as possible. However, when a massive task generated by the burst data of the edge cloud arrives, it will cause some VMs to be idle. By contrast, some cases of excessive load will result in the wastage of resources and a reduction in operating efficiency. In addition, there is a reasonable threshold of resource utilization for each host when processing tasks of the same type. We argue that if the similarity between a task and the resources used by a host is relatively high, the task cannot be placed in a specific service node but in a wait state. Furthermore, when a VM processes a task, it needs to combine with other VMs when its available resources cannot load the current task. However, due to the different resource types, the VM tends to combine into a balanced physical machine model with a low resource conflict rate to ensure its long-term operation.

Fig. 3(a) shows that the difference between the current VM's memory usage and CPU usage is greater than 5%; therefore, it is impossible to continue to load memory-demanding tasks, so a memory conflict host is generated. Similarly, Fig. 3(c) shows a CPU conflict host. Therefore, for the host, our goal is to build a low-conflict physical machine model, as shown in Fig. 3(b).

In summary, the SimCMA model is established by the following three steps.

1) SIMILARITY OF PREVIOUS TASK ASSIGNMENT

(a) When a task arrives, the decision variables quantize the task mapped to a VM. First, a VM must exist, or the system would produce a VM.

(b) When the task is unloaded, the optimal computing threshold of VM space should be fully considered to ensure the long-term operation of the VM. Clustering task features and superimposing similar types of tasks will result in the

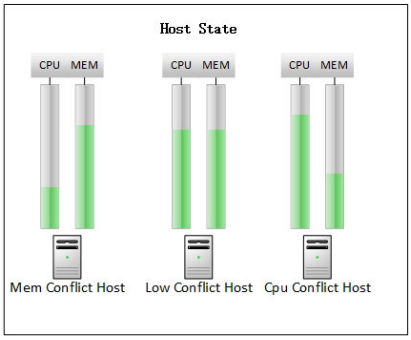


FIGURE 3. Schematic of host-based conflict conditions.

corresponding VM utilization exceeding the optimal range. Therefore, the CPU and memory requirements of the physical machine are used as decision variables to build a similarity model between tasks.

In this study, the set of physical hosts contained in cluster J is assigned as $J = \{H1, H2, \ldots, Hn\}$. Each physical host processes concurrent tasks through the virtual machine. A host is defined as $Hx = \{VM1, VM2, \ldots\ldots VMn\}$, and the task set $T = \{T1, T2, \ldots, Tn\}$ is used in the cloud data center. Task set Tx represents the CPU, and the memory attributes are abstracted in the form of an n × k matrix as follows. It is assumed that the first two columns are the CPU and memory attributes in set T.

$$\begin{bmatrix} T_1^i & T_1^j & \cdots & T_1^k \\ T_2^i & T_2^i & \cdots & T_2^k \\ \vdots & \vdots & \ddots & \vdots \\ T_n^i & T_n^j & \cdots & T_n^k \end{bmatrix}_{n*k}$$

After all the nodes of the cloud data center are started, each task T is assigned according to the arrival order. The relationship between task Tx and the host is random, that is, any Tx—>$rand(Hx)$. A task can be offloaded to the host such that the load range of the host is not exceeded. In this study, taking the CPU and memory (MEM) as examples, it is considered that the remaining resources of the host are more than the resource requirements of the task. This can be expressed as follows:

$$\mathrm{Hx}\ (CPU\_left) > \mathrm{Tx}\ (CPU) \tag{1}$$

$$\mathrm{Hx}\ (MEM\_left) > \mathrm{Tx}\ (MEM) \tag{2}$$

The above relations indicate that a constraint condition for current task Tx to be offloaded normally to a host is that the remaining CPU utilization of the host is greater than the CPU utilization required by the current task. The similarity between two vectors $V_1\ (y1, y2 \ldots\ldots yn)$ and $V_2(x1, x2 \ldots\ldots xn)$ is generally expressed as the cosine of the angles xi and yi (Sim). The general formula is as follows [25], [26]:

$$Sim = \frac{\sum_{i=1}^{n} (x_i * y_i)}{\sqrt{\sum_{i=1}^{n} x_i^2} * \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{3}$$

Similarly, based on formula (3), we propose using the Sim index to measure the similarity between task attributes and the resources used by the host. The average execution time of tasks is expressed as T_{a-excu}, and the execution time of the similarity algorithm is expressed as T_{s-excu}, where

$$T_{a-excu} \gg T_{s-excu} \tag{4}$$

Therefore, the effect of the similarity algorithm on data center tasks is ignored. The following similarity formula is then established:

$$Sim = \frac{\left(T_i^{k1} * PM_i^{u1}\right) + \left(T_i^{k2} * PM_i^{u2}\right)}{\sqrt{\left(T_i^{k1}\right)^2 + \left(T_i^{k2}\right)^2} * \sqrt{\left(PM_i^{l1}\right)^2 + \left(PM_i^{l2}\right)^2}} * 100\% \tag{5}$$

Here, PM_i is the edge of the cloud cluster physical machines, T_i is the current task, and u1 and u2 are the CPU and memory resources used by the physical machine, respectively.

The higher the Sim index is, the greater the similarity of the task attributes to be allocated to the resources used by the current physical machine. The initial similarity threshold is 65% [27]. If the threshold is exceeded, it is considered that processing the task by the current physical machine will cause a certain index of the CPU or MEM to be overused, and as a result, the task is not suitable for running on the physical machine.

In this study, we set the initialization threshold on the basis of historical data. However, due to the uncertainty of the actual cloud task data, setting the threshold extremely low may cause some tasks to fail to run in the host in a timely manner, resulting in a task backlog. Therefore, to ensure the normal offloading of tasks, the concept of a backlog is proposed here. An unsuccessfully assigned task is defined as $overstock\_T$. The threshold of the task backlog rate is set to 10% or below to avoid an excessive task backlog. The backlog rate is calculated by formula (6):

$$Backlog = \frac{num(overstock\_T)}{num(Tx)} \times 100\% \tag{6}$$

Based on formula (6), the similarity adjustable space is set to 65%–95%. Here, we define init_Sim = 65% and max_Sim = 95%. For values above 95%, CloudSim fails to initialize. The threshold ranges are shown in formulas (7) and (8):

$$\mathrm{Sim} \in [65\%, 95\%] \tag{7}$$

$$\mathrm{Backlog} \in [10\%, 90\%] \tag{8}$$

The similarity optimization model considering the backlog rate is shown in formula (9):

$$Sim\ (t) = \begin{cases} init\_Sim, , Backlog < 10\% & (9) \\ init_{Sim} + \dfrac{max\_Sim - init\_Sim}{max\_Backlog - min\_Backlog} * \\ (Backlog\ (t) - Backlog\ (t-1)), Backlog \geq 10\% & (10) \end{cases}$$

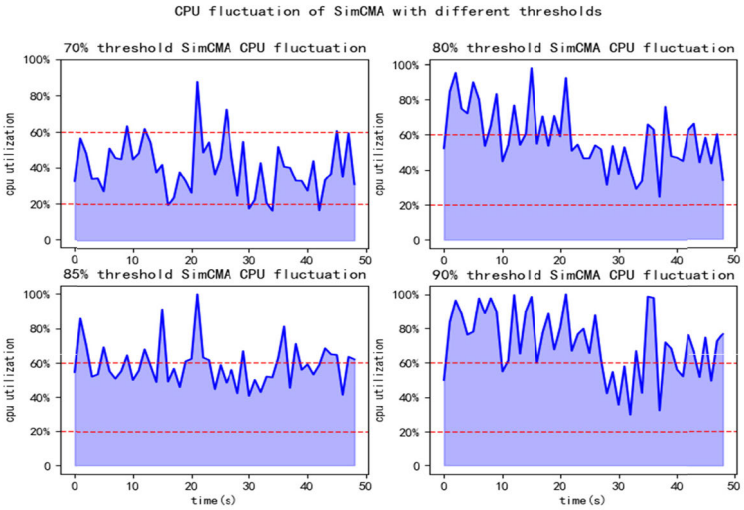


FIGURE 4. Comparison of selected CPU threshold experiments.

When Backlog $< 10\%$, init_Sim is calculated normally; when Backlog $\geq 10\%$, the total amount of similarity changes over time is calculated according to the offset of the change in the backlog rate over time.

2) ESTABLISHMENT OF THE RESOURCE ALLOCATION MODEL FOR CONFLICT CONDITIONS

According to the similarity calculation described above, after a task is mapped to a VM, overload of the VM can still occur, which would affect the operational stability of the VM. This is because the remaining resources of the host are not fully considered during the calculation. Therefore, in this paper, the SimCMA model is established based on the factors influencing the resources used by the host. From the perspectives of CPU and MEM, hosts are divided into three types according to the differences between the host and each attribute: MEM conflict host (type A), CPU conflict host (type B), and low-conflict host (type C). Type A: CPU_left-MEM_left $> 5\%$, type B: MEM_left-CPU_left $> 5\%$, type C: the absolute difference between the two types is less than or equal to 5%, which is our target physical machine type. These types are expressed in formulas (11)-(13):

MEM conflict host:

$$H_i^{CPU\_leftRate} - H_i^{MEM_{leftRate}} > 5\%, \quad H_i \in A; \tag{11}$$

CPU conflict host:

$$H_j^{MEM\_leftRate} - H_j^{CPU_{leftRate}} > 5\%, \quad H_j \in B; \tag{12}$$

Low-conflict host:

$$\left| H_t^{MEM\_leftRate} - H_t^{CPU\_leftRate} \leq 5\% \right|, \quad H_t \in C; \tag{13}$$

We consider the load capacity of the physical machine where the VM is located. Taking a MEM conflict host as an example, if only the abovementioned formula (11) is considered, this will easily result in the host's CPU resource request being extremely large, which will affect the normal operation of the host. Therefore, the maximum threshold of the CPU and memory usage of the host is set to 85% [28]. To prove that the selected threshold is reasonable, this study takes CPU utilization as an example and sets the thresholds of CPU utilization to 70%, 90%, 80% and 85% (Fig. 4).

The results show that, when the threshold value is 70%, the CPU utilization rate of the host is generally low, and the CPU utilization rate of some hosts is even lower than 20%, and the overall CPU utilization rate is low.When the threshold value is 90%, the CPU utilization rate will usually reach 100%. The overall CPU utilization rate is too high, which is not conducive to maintaining the long-term operation of the host computer.When the threshold value is set to 85%, the overall performance of the CPU is good and the curve is relatively smooth.

In addition, to reduce energy consumption overhead,the host is set to automatically shut down when its current CPU usage is less than 20% [29]. The host CPU and memory boundary conditions are thus set as formulas (14)–(16):

$$H\left(CPU\_used\right) < H\left(CPU\right) \times 85\%, \tag{14}$$

$$H\left(MEM\_used\right) < H\left(MEM\right) \times 85\%, \tag{15}$$

$$H\left(CPU\_used\right) > 20\% \tag{16}$$

Based on the analysis of the abovementioned conflict condition model, the core aim of the proposed VM scheduling model design is to construct a VM scheduling method that tends to produce low-conflict hosts. Therefore, the following scheduling priority is set:

$$Priority\left(AB\right) > \begin{cases} Priority\left(AA\right) \\ Priority\left(BB\right) \end{cases} > \begin{cases} Priority\left(AC\right) \\ Priority\left(BC\right) \end{cases} \tag{17}$$

When the hosts of the two VMs participating in the scheduling are types A and B, the scheduling priority is the

highest. The second priority of VM scheduling is that the hosts are of type A or B because any VM is itself balanced. When the two VMs are considered to have reached the equilibrium model, they do not participate in the scheduling. The specific VM scheduling process is set as the following three steps:

(a) Divide the conflict rate model for each physical machine in the data center as follows:

$$A = \{H1, H2 \ldots\ldots Hk\}, \tag{18}$$

$$B = \{H1, H2 \ldots\ldots Hn\}, \tag{19}$$

$$C = \{H1, H2 \ldots\ldots Hm\}, \tag{20}$$

(b) Design the weights s, w, and t corresponding to the priorities for the VM scheduling.

(c) By traversing the current host cluster on the basis of the different weights mentioned above, construct a conflict reduction coefficient matrix for each host according to the priority reduction characteristics in the conflict model and by combining the corresponding weights.

The corresponding conflict reduction coefficient symmetric matrix is

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kn} \end{bmatrix}_{k*n}$$

Here, considering formula (22) as an example for analysis, when H_i and H_j correspond to two types of conflicting physical machines, that is, formulas (11) or (12), if the absolute value of the difference in formula (22) (e.g., CPU_usedRate) approaches 0, H_i and H_j are considered to have similar CPU resource usage, so it is not suitable to accept CPU-type VMs. To make full use of the remaining resources of different hosts and build a balanced virtual unit combination, the value of $\left|H_i\left(CPU\_usedRate\right) - H_j\left(CPU\_usedRate\right)\right|$ in formula (22) should be increased to the maximum possible; that is, the combination mode with the largest Crash coefficient should be selected for VM scheduling. This enables H_i and H_j to take full advantage of their own resources to receive the corresponding VM. In formula (24), when both H_i and H_j are of the low-conflict type, the effect of t is minimal because its weight has the lowest priority. Therefore, in the SimCMA model, the cases where either H_i or H_j is of the low-conflict type are ignored.

3) DYNAMIC GLOBAL OPTIMIZATION OF THE MODEL

Based on formulas (21)–(23), as shown at the bottom of the page. we construct the conflict condition model. This model sets the combination strategy before VM migration, thus reducing the migration times of the VMs to some extent. However, because the migration time of VMs is also a key factor affecting the host performance, the SimCMA model incorporates the improved ant colony algorithm to ensure a global optimal VM migration path and the minimum VM migration time.

The ant colony algorithm uses the walking paths of ants to represent the feasible solution of a problem to be optimized, and all the paths of the entire ant colony constitute the solution space of the problem to be optimized [30]. Ants with shorter paths release more pheromones. As time progresses, the concentration of pheromones accumulated on shorter paths gradually increases, and the numbers of ants that choose those paths increase [31]. In this study, the released amount of pheromones is directly related to the physical machine conflict coefficient matrix. For host combinations with high conflict coefficients, the greater the amount of pheromone, the more likely the VM is to choose the scheduling route that contains the destination of the host. In summary, the derivation of the original ant colony algorithm used in this study follows the following process:

Given the initial state VM $_{(K)}$, the probability of the VM migrating is given by formula (24):

$$P_{ij}^{k}(t) = \frac{message_{ij}^{\alpha}(t) * distance_{ij}^{\beta}(t)}{\sum_{s \in allowed_k} message_{is}^{\alpha}(t) * distance_{is}^{\beta}(t)} \tag{24}$$

Here, α and β represent the pheromone acquisition probability for a distance from H_i to H_j within t and the probability of obtaining the corresponding bandwidth for a distance from H_i to H_j within t, respectively.

To avoid the excessive effect of pheromones in time t and reduce the effect of the distance factor, a pheromone volatility factor is proposed.

$$message_{ij}(t+1) = (1-\rho)\, message_{ij}(t) + \Delta message_{ij}(t) \tag{25}$$

Δmessage$_{ij}$ (t) can be regarded as a pheromone increment, and ρ can be regarded as a pheromone volatility factor. In this study, we ignore the network bandwidth under the edge cloud; therefore, we focus on the relationship between pheromones and VM scheduling. To reduce the number of VM scheduling

$$Crash = max \begin{cases} \left|H_i\left(CPU\_usedRate\right) - H_j\left(CPU\_usedRate\right)\right| * \\ \left|H_i\left(MEM\_usedRate\right) - H_j(MEM\_usedRate)\right| * s, i \in A, j \in B & (21) \\ \left|H_i\left(CPU\_usedRate\right) - H_j\left(CPU\_usedRate\right)\right| * \\ \left|H_i\left(MEM\_usedRate\right) - H_j(MEM\_usedRate)\right| * w, i, j \in AorB & (22) \\ \left|H_i\left(CPU\_usedRate\right) - H_j\left(CPU\_usedRate\right)\right| * \\ \left|H_i\left(MEM\_usedRate\right) - H_j(MEM\_usedRate)\right| * t, i \; or \; j \in C & (23) \end{cases}$$

tasks and algorithm traversals and to increase the weight of pheromones, the concept of a volatility inhibition factor ($\hat{\rho}$) is proposed. Formula (25) can be rewritten as formula (26):

$$message_{ij}(t+1) = (1-\rho * \hat{\rho})\, message_{ij}(t) + \Delta message_{ij}(t) \tag{26}$$

To reduce the resources available to the host, the inhibitor is reduced, and to prevent excessive use of physical resources and degradation of the computing machine performance, when the CPU and memory resources of the host reach the threshold condition of 85%, the remaining pheromones on the scheduling path will be quickly emptied at the next time node. Thus, the possibility that other unmigrated VMs will continue to migrate to the host is significantly reduced, thereby maintaining the high performance state of the host. Therefore, in formula (26), $message_{ij}(t+1) = 0$; then, formula (26) can be written as follows:

$$\begin{aligned} 1 - (\rho * \hat{\rho}) &= 0 \\ \hat{\rho} &= \frac{1}{\rho} \end{aligned} \tag{27}$$

Because the CPU and memory threshold conditions are set to 85%, the $\hat{\rho}_{ij}$ value must be computed in multiple stages. The value of the initial $\hat{\rho}_{ij}$ is determined according to the minimum value of the remaining resources of the host in time t, as shown in formula (28), as shown at the bottom of the page. Formulas (29), (30), and (31), as shown at the bottom of the page illustrate changes in $\hat{\rho}_{ij}$ over time under the conditions that the remaining resources of the destination host are greater than the boundary threshold, equal to the boundary threshold, and less than the boundary threshold, respectively. It can be seen that the greater the remaining resources of the destination host are, the stronger the effect of the inhibitor. When the boundary conditions are not met, that is, when the destination host has entered a saturated state, the inhibitor quickly volatilizes pheromones and reduces the concentration, and the destination host continues to schedule other VMs. In formula (26), $\Delta message_{ij}(t)$ represents the largest conflict, implying the largest conflict coefficient in the above conflict condition model. This can be expressed in terms of $Crash^{t}_{max} - Crash^{t+1}_{max}$. The pheromone function of SimCMA is expressed as formula (32):

$$\begin{aligned} message_{ij}(t+1) = {} & (1 - \rho * \hat{\rho}_{ij})\, message_{ij}(t) \\ & + (Crash^{t}_{max} - Crash^{t+1}_{max}) \end{aligned} \tag{32}$$

The virtual resource allocation model established as described above can significantly reduce the number of scheduling tasks while achieving a balanced VM migration strategy to ensure long-run edge cloud data centers.

IV. SIMULATION EXPERIMENTS AND RESULTS ANALYSIS

A. TYPES OF GRAPHICS

The experiments in this study were conducted using the CloudSim cloud simulator for simulating the access of concurrent tasks to edge cloud users (data reference from internal simulation data of the CloudSim platform). To observe the energy consumption of different algorithms for the same task, 500, 800, 1000, 1200, and 1500 task data with correct CPU and memory requirements were used as test data. The parameters of heterogeneous nodes in the cluster are shown in Table 1:

TABLE 1. Configuration information of the cluster.

Node	Core	Memory	CPU
Node 1-400	2	4GB	Xeon 3075 2660 MHz
Node 401-800	2	4GB	Xeon 3040 1860 MHz

In the experiments, the resource types used were CPU and memory (unit: MB) and energy consumption (unit: kwh). Based on the above settings of the CPU and memory, a relevant reference value range of 85% was used, following previous studies [32], [33]. The specific performance evaluation indicators are as follows: 1) the total number of VM scheduling times of the cloud center, 2) the energy consumption of the host, 3) the number of VM migrations, and 4) the scheduling time of the cluster. The experiments were divided into two phases: 1) the SimCMA model is established, and 2) it is optimized to ensure global solutions. For the analysis of the proposed model, the abovementioned phase division and performance indicators for other scheduling algorithms of the CloudSim simulation platform were compared with those for our model.

B. MULTIPART FIGURES

1) INITIAL SIMCMA MODEL EXPERIMENTS BASED ON CONFLICT CONDITIONS

Five common combination algorithms of VM, namely, IqrMu (inter quartile range allocation/minimum utilization selection policy), Lrrs (local regression/random selection policy), MadMmt (median absolute deviation allocation/minimum

$$\hat{\rho}_{ij}(t) = \begin{cases} \hat{\rho}_{init} * \min\left\{H^{j}_{t}\left(CPU_{left}\right), H^{j}_{t}\left(MEM_{left}\right)\right\}, & (28) \\ ifH\,(CPU\_left) > 0.15 * H(CPU) and MEM\_left > 0.15 * H\,(MEM), & (29) \\ \dfrac{1}{\rho}, & (30) \\ ifCPU\_left < 0.15 * H\,(CPU)\, orMEM\_left < 0.15 * H\,(MEM), & (31) \end{cases}$$

migration time selection), MadRs (median absolute deviation allocation/minimum random selection, and MadMc (median absolute deviation allocation/maximum correlation), in the CloudSim platform were selected for comparison with the proposed algorithm. This experiment simulated the VM scheduling times when different algorithms were applied for the processing of 30000 identical tasks (task interval $d = 5000$) by VMs. The experimental results obtained using the SimCMA model are shown in Figs. 5 and 6. The results indicate that because the similarity ratio between the resources used by the host and the current task is considered in the proposed model, at the beginning of the experiment, tasks that might cause excessive load can be filtered out. This approach reduces the number of VM migrations compared with that of the other algorithms, with the maximum value reduced by 8.33%. For example, compared with MadRs, because our model uses the approximate median selection method of absolute median difference, it is similar to the SimCMA model. The scheduling times of the two algorithms are similar but that of the SimCMA model is still slightly lower. After calculating the energy consumption of the host, the energy consumption interval was set as $d = 25$. Note that the overall energy consumption of the SimCMA model is still lower than that of the other algorithms, saving up to 6.25%. Consequently, the SimCMA model can optimize nodes from two aspects: VM scheduling times and host energy consumption.

In a subsequent experiment, the execution efficiency of each algorithm was specifically considered. From the VM selection and host selection algorithms, the process of task processing is defined in both directions, and the obtained results are shown in Fig. 5. The overall algorithm running time indicates that the average running time of the SimCMA model is lower than that of the other algorithms. The overall running time of the selected algorithm is analyzed based on the experimental results. During the initial running of the task, because the IqrMu algorithm uses the concept of quartiles, the difference between two VMs participating in the operation is divided into four intervals, so the remaining resources of the host match well and there are fewer scheduling tasks. This algorithm requires four operations due to resource matching, whereas SimCMA has more idle time in the early stage of the operation. Therefore, not only is the running time short but also the algorithm execution efficiency can be improved by an average of 20.47%. A comparison of the average value of the SimCMA model and those of the other algorithms is shown in Fig. 7.

In summary, the SimCMA model can improve the overall task processing speed of the host. The Lrrs algorithm adopts a random selection strategy in the VM selection stage and a local regression strategy in the VM allocation stage. Therefore, when the number of task sets is relatively small, the regression prediction accuracy is relatively high and the speed is relatively fast, but when the number of tasks is higher than 1000, the running time of the algorithm significantly increases. For the MadMmt algorithm, the VM selection

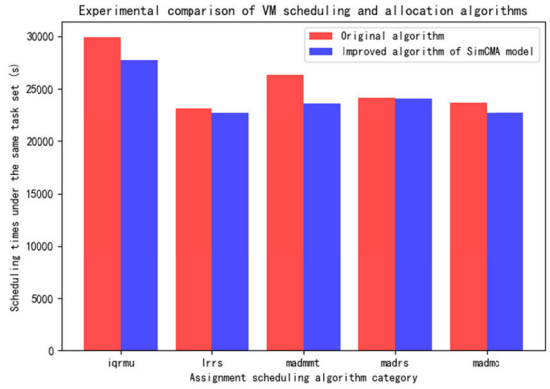


FIGURE 5. Comparison of VM scheduling times among the SimCMA model and other algorithms under the same task set.

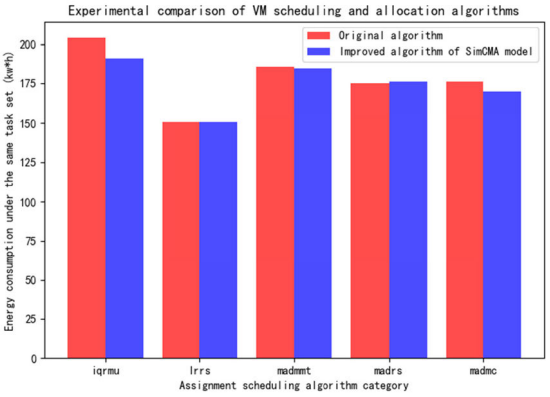


FIGURE 6. Comparison of mainframe energy consumption of the SimCMA model and other algorithms under the same task set.

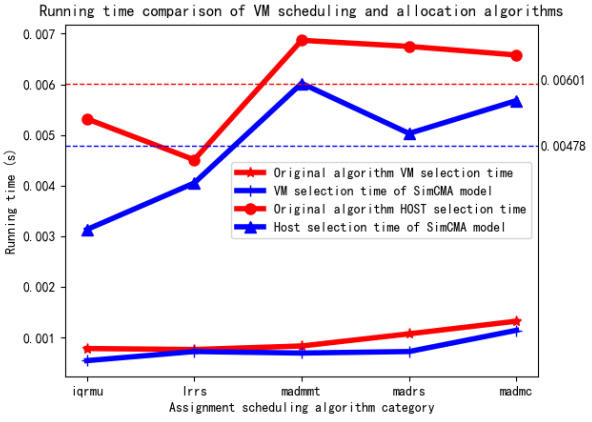


FIGURE 7. Comparison of the running times of the SimCMA model and other algorithms for VM scheduling.

strategy of the algorithm adopts the minimum migration time method. Therefore, in the algorithm selection, memory resources are used to select the least expensive VM migration scheme from the migration VM queue (the priority of small memory reflects the minimum time). The algorithm uses 12 (recommended safety index) historical data to calculate the absolute deviation, which specifies the overload threshold for allocation. Because only memory is used as an indicator,

the CPU of the host may be in an abnormal working state, so the execution time is long. For the MadMt algorithm, since the absolute median value of this point considers the condition of whether the median resource between two VMs exists, when the condition does not exist, the algorithm traverses other hosts. Therefore, the scheduling time is the highest, and the algorithm is similar to the AB conflict model in this paper. At this point, the task execution time of SimCMA reaches the maximum for the MadRs algorithm. Because this algorithm adopts a random selection strategy in the VM selection stage and a median absolute deviation method in the VM allocation stage, the execution time is shorter when the number of tasks is small, and the overall execution time is also slightly lower. For the MadMc algorithm, because the algorithm is in the VM selection stage, the maximum correlation selection method is used, and a linear regression model is used to fit the CPU historical data of the VM to select the maximum correlation of historical data. Therefore, with increases in the time and the number of task sets, the increase in historical data will be very fast, so the calculation time of the linear model is relatively long in all algorithms.

2) GLOBAL OPTIMIZATION OF THE SIMCMA MODEL

Next, a dynamic VM scheduling problem with time slots was experimented. As described earlier, under the same task set conditions, compared with that of other algorithms, the VM scheduling time of the SimCMA model is lower. Therefore, the arrival of a task is often sudden, and the dynamic migration of the VM is the focus of this experiment.

In this experiment, the VM scheduling times and host energy consumption were considered under the conditions of 1500 task sets and a task growth interval of d = 500. Here, again, we compared the proposed algorithm with the five basic algorithms considered in the previous experiments. The results are shown in Figs. 8 and 9. From the pheromone suppression of volatile factors calculated by formula (27), the following reasons describe why the SimCMA model is better in terms of VM scheduling time and host energy consumption.

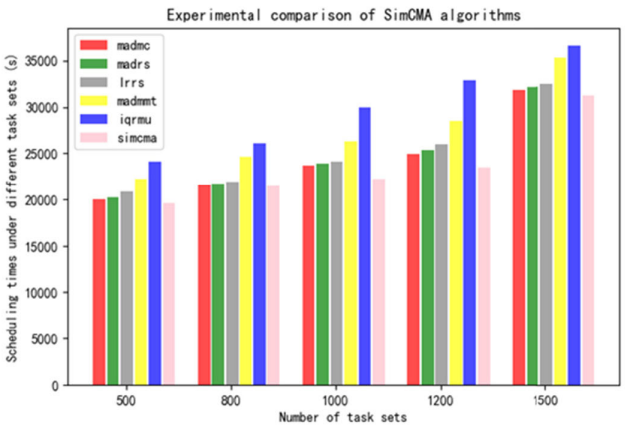


FIGURE 8. Comparison of VM scheduling times of the SimCMA model and other algorithms under different task sets.

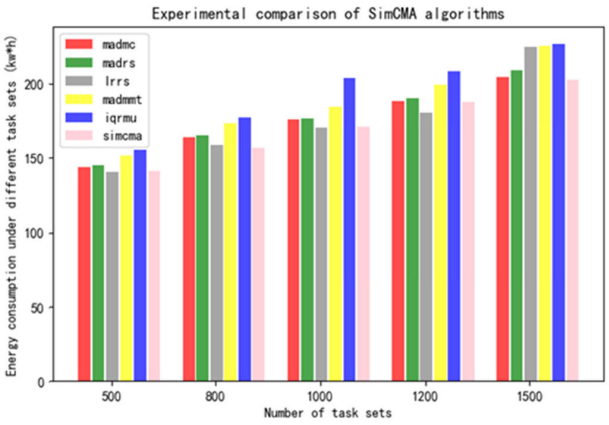


FIGURE 9. Comparison of mainframe energy consumption of the SimCMA model and other algorithms under different task sets.

(a) Due to the effect of suppressing volatility factors, in the initial stage of VM scheduling, when the host has more idle resources, the amount of pheromones increases on a certain path, and the priority of the conflict coefficient matrix tends to be the lowest. Therefore, the model tends to be a balanced VM combination for AC, BC, and even CC. Thus, the initial SimCMA algorithm performs well in both aspects shown in Figs. 10 and 11.

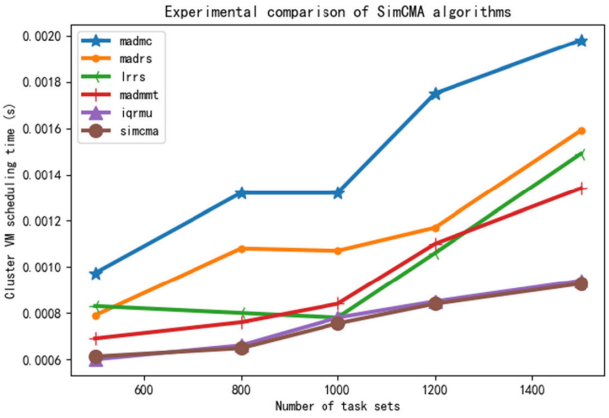


FIGURE 10. Comparison of VM scheduling times of the SimCMA model and other algorithms.

(b) During the scheduling process, the increase in the amount of pheromones on a certain path in the initial stage is within a reasonable threshold range due to the limitation of the volatility inhibition factor. Therefore, the overall VM scheduling times of the SimCMA algorithm increase smoothly.

(c) As the number of tasks increases, the resource consumption of the host also increases. At this point, the role of the volatility inhibition factor gradually decreases. When the volatility inhibition factor exceeds the threshold, the increase in pheromones on the path is 0. Consequently, an excessive increase in the node load is avoided; therefore, the host energy consumption of the SimCMA algorithm increases steadily and is generally low.

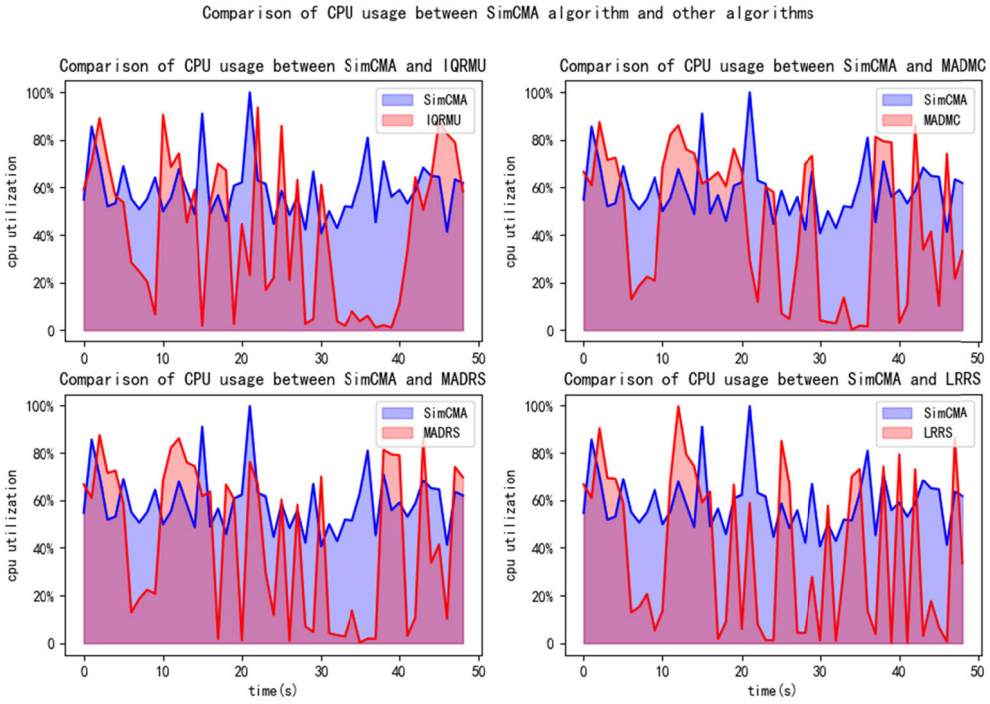


FIGURE 11. Comparison of CPU usage fluctuations of the SimCMA model and other algorithms.

After considering the scheduling time of the VMs in the cluster, as shown in Fig. 8, because of the effect of the volatility inhibition factor on pheromones in the SimCMA model, when the critical value of the resource is reached, the amount of pheromones on the path will rapidly return to 0, thereby realizing rapid VM scheduling, and the VM scheduling time in the cluster suddenly increases. Additionally, from the results of the SimCMA model, the overall VM scheduling time of the SimCMA model is still lower than those of the other algorithms.

Finally, we considered the fluctuation of the host CPU as the number of tasks increases to analyze the impact of SimCMA on the stability of the host. The four subgraphs in Fig. 11 compare the CPU usage of this algorithm with that of the other algorithms. Although the CPU usage of the proposed algorithm is slightly higher at the initial moment, the overall CPU usage of the algorithm is more stable than that of the other algorithms; therefore, this algorithm is more suitable for edge nodes to process a certain stable operation when there are data bursts within a time interval.

V. CONCLUSION

Through analysis of edge cloud tasks and the cluster environment, this study established a balanced resource allocation model based on conflict conditions (SimCMA model).

First, the similarity principle was used to consider the task backlog to achieve a reasonable unloading of tasks. Then, based on the conflict conditions, a resource allocation model with advantages in reducing VM scheduling times and host energy consumption was constructed. According to the degree of resource allocation, the model considers that the local pheromones may increase in the traditional ant colony algorithm. The concepts of a pheromone volatility factor and inhibition factor were implemented to avoid redundant VM scheduling or resource overload when resources reach a critical value. From the results of experiments comparing the proposed SimCMA model with other existing models, the proposed model has certain advantages in terms of VM scheduling time, host energy consumption, and execution efficiency. In addition, the SimCMA model can ensure stable operation of the host CPU when the magnitude of the data increases.

In summary, the balanced resource allocation model proposed in this study has certain advantages in maintaining the stability of edge cloud nodes. This model was verified using the CloudSim simulation platform, which limited the obtained attribute types. Therefore, in the future, we will build clusters and use real data to expand the types of decision variables to further verify the practical application of this model.

ACKNOWLEDGMENT

The authors would like to thank all the authors and reviewers for their efforts to make this Special Section. The authors also appreciate the contributions of many experts in the field who have participated in the review process and provided constructive suggestions to the authors to improve the content and presentation of the article.

REFERENCES

[1] X. L. Zhang, J. H. Yang, X. J. Sun, and J. P. Wu, "Survey of geo-distributed cloud research progress," *J. Softw.*, vol. 29, no. 7, pp. 2116–2132, Jul. 2018, doi: 10.13328/j.cnki.jos.005555.

[2] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017, doi: 10.1109/TWC.2016.2633522.

[3] Y. Xie, Y. Zhu, Y. Wang, Y. Cheng, R. Xu, A. S. Sani, D. Yuan, and Y. Yang, "A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment," *Future Gener. Comput. Syst.*, vol. 97, pp. 361–378, Aug. 2019, doi: 10.1016/j.future.2019.03.005.

[4] T. Yang, Y. Hu, M. C. Gursoy, A. Schmeink, and R. Mathar, "Deep reinforcement learning based resource allocation in low latency edge computing networks," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Lisbon, Portugal, Aug. 2018, pp. 1–5.

[5] M. Yang, X.-L. Huang, W.-W. Miu, P.-F. Yu, W. Li, and R.-X. Yang, "Deep reinforcement learning based green resource allocation mechanism in mobile edge network for ubiquitous power IoT," in *Proc. 2nd Int. Conf. Adv. Control, Automat. Artif. Intell. (ACAAI)*. Wuhan, China: Science and Engineering Research Center, 2020, p. 7.

[6] J. Xu, Z. Hao, R. Zhang, and X. Sun, "A method based on the combination of laxity and ant colony system for cloud-fog task scheduling," *IEEE Access*, vol. 7, pp. 116218–116226, 2019.

[7] C. Tang, S. Xiao, X. Wei, M. Hao, and W. Chen, "Energy efficient and deadline satisfied task scheduling in mobile cloud computing," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018.

[8] C. Jian, J. Chen, J. Ping, and M. Zhang, "An improved chaotic bat swarm scheduling learning model on edge computing," *IEEE Access*, vol. 7, pp. 58602–58610, 2019.

[9] Y. Jararweh, S. Otoum, and I. A. Ridhawi, "Trustworthy and sustainable smart city services at the edge," *Sustain. Cities Soc.*, vol. 62, Nov. 2020, Art. no. 102394.

[10] H. B. Zhang, H. Li, S. X. Chen, and X. F. He, "Computing offloading and resource optimization in ultra-dense networks with mobile edge computation," *J. Electron. Inf.*, vol. 41, no. 5, pp. 183–190, 2019, doi: 10.11999/JEIT180592.

[11] X. Guo, L. Liu, Z. Chang, and T. Ristaniemi, "Data offloading and task allocation for cloudlet-assisted ad hoc mobile clouds," *Wireless Netw.*, vol. 24, no. 1, pp. 79–88, Jan. 2018.

[12] G. S. Aujla and N. Kumar, "MEnSuS: An efficient scheme for energy management with sustainability of cloud data centers in edge–cloud environment," *Future Gener. Comput. Syst.*, vol. 86, pp. 1279–1300, Sep. 2018.

[13] V. Balasubramanian, F. Zaman, M. Aloqaily, S. Alrabaee, M. Gorlatova, and M. Reisslein, "Reinforcing the edge: Autonomous energy management for mobile device clouds," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 44–49.

[14] Y. Wei, Z. Wang, D. Guo, and F. R. Yu, "Deep Q-learning based computation offloading strategy for mobile edge computing," *Comput., Mater. Continua*, vol. 59, no. 1, pp. 89–104, 2019.

[15] R. Ganesan, S. Sarkar, and A. Narayan, "Analysis of SaaS business platform workloads for sizing and collocation," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Honolulu, HI, USA, Jun. 2012, pp. 868–875.

[16] T. Yang, L. Wan, S. J. Ma, and L. J. Ma, "LVS cluster load balancing algorithm with adaptive weight leastload," *Commun. Technol.*, vol. 50, no. 4, pp. 741–745, 2017, doi: 10.3969/j.issn.1002-0802.2017.04.027.

[17] M. Guazzone, C. Anglano, and M. Canonico, "Exploiting VM migration for the automated power and performance management of green cloud computing systems," in *Energy Efficient Data Centers* (Lecture Notes in Computer Science), J. Huusko, H. D. Meer, S. Klingert, A. Somov, Eds. Berlin, Germany: Springer, 2012, pp. 81–92.

[18] Z. Á. Mann, "Rigorous results on the effectiveness of some heuristics for the consolidation of virtual machines in a cloud data center," *Future Gener. Comput. Syst.*, vol. 51, pp. 1–6, Oct. 2015, doi: 10.1016/j.future.2015.04.004.

[19] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera, "A stable network-aware VM placement for cloud systems," in *Proc. 12th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, Ottawa, ON, Canada, May 2012, pp. 498–506.

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

[21] Z. Y. Li, S. M. Chen, B. Yang, and R. F. Li, "Multi-objective memetic algorithm for task scheduling on heterogeneous cloud," *Chin. J. Comput.*, vol. 39, no. 2, pp. 377–390, Feb. 2016, doi: 10.11897/SP.J.1016.2016.00377.

[22] A. Dalvandi, M. Gurusamy, and K. C. Chua, "Application scheduling, placement, and routing for power efficiency in cloud data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 947–960, Apr. 2017, doi: 10.1109/TPDS.2016.2607743.

[23] F. Ahamed, S. Shahrestani, and B. Javadi, "Security aware and energy-efficient virtual machine consolidation in cloud computing systems," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Tianjin, China, Aug. 2016, pp. 1516–1523.

[24] K. Liu, "Greedy algorithm-based virtual machine migration strategies in cloud data center," *Comput. Eng.*, vol. 45, no. 10, pp. 33–39, 2019.

[25] W. L. Wang, Y. Xu, Y. W. Zhao, and W. C. Zhu, "Research on location-routing problem based on hyper-heuristic algorithm," *J. Zhejiang Univ. Technol.*, vol. 47, no. 6, pp. 604–610, 2019.

[26] Y. Zhou, L. Kong, Z. Wu, S. Liu, Y. Cai, and Y. Liu, "Ensemble of multi-objective Metaheuristic algorithms for multi-objective unconstrained binary quadratic programming problem," *Appl. Soft Comput.*, vol. 81, Aug. 2019, Art. no. 105485, doi: 10.1016/j.asoc.2019.105485.

[27] K. Z. Shi, H. Q. Yu, F. Luo, and G. S. Fan, "VMC strategy based on the mutual exclusion conditions for cloud data center," *J. East China Univ. Sci. Technol.*, vol. 43, no. 1, pp. 119–128, Feb. 2017, doi: 10.14135/j.cnki.1006-3080.2017.01.019.

[28] C. Zhao, L. S. Yan, Y. H. Cui, H. L. Xing, and B. Feng, "Dynamic adjusting threshold algorithm for virtual machine migration," *J. Comput. Appl.*, vol. 37, no. 9, pp. 2547–2550, Feb. 2017, doi: 10.11772/j.issn.1001-9081.2017.09.2547.

[29] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007, doi: 10.1109/MC.2007.443.

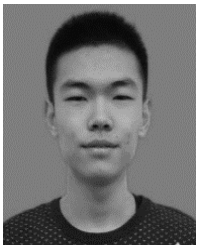
[30] X. Sheng and Q. Li, "Template-based genetic algorithm for QoS-aware task scheduling in cloud computing," in *Proc. Int. Conf. Adv. Cloud Big Data (CBD)*, Chengdu, China, Aug. 2016, pp. 25–30.

[31] F. Alharbi, Y.-C. Tian, M. Tang, W.-Z. Zhang, C. Peng, and M. Fei, "An ant colony system for energy-efficient dynamic virtual machine placement in data centers," *Expert Syst. Appl.*, vol. 120, pp. 228–238, Apr. 2019, doi: 10.1016/j.eswa.2018.11.029.

[32] H. Q. Zhang, X. Zhang, H. Wang, and Y. Liu, "Task scheduling algorithm based on load balancing ant colony optimization in cloud computing," *Microelectron. Comput.*, vol. 32, no. 5, pp. 31–35 and 40, 2015, doi: 10.19304/j.cnki.issn1000-7180.2015.05.007.

[33] Y. Zhou, Y. Zhang, H. Liu, N. Xiong, and A. V. Vasilakos, "A bare-metal and asymmetric partitioning approach to client virtualization," *IEEE Trans. Services Comput.*, vol. 7, no. 1, pp. 40–53, Jan. 2014, doi: 10.1109/TSC.2012.32.

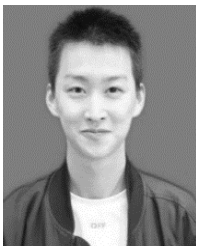
LEILEI ZHU received the master's degree from the Changchun University of Science and Technology, where she is currently pursuing the Ph.D. degree with the School of Computer Science and Technology. She is also a Lecturer with the School of Medical Information, Changchun University of Chinese Medicine.



JIAHUI FENG was born in June 1999. He is currently pursuing the bachelor's degree with the Changchun University of Science and Technology. His research interests include cloud computing, data analysis, and data mining.

DAN LIU received the Ph.D. degree from the Changchun University of Science and Technology, Changchun, China. She is currently an Associate Professor with the School of Computer Science and Technology, Changchun University of Science and Technology. She is a member of CCF.

HONGWEI YANG received the master's degree from the Changchun University of Science and Technology, Changchun, China. He is currently an Associate Professor with the School of Computer Science and Technology, Changchun University of Science and Technology.



ZHENGQI BAI was born in June 1999. He is currently pursuing the bachelor's degree with the Changchun University of Science and Technology. His main research interests include cloud computing, data analysis, and data mining.

XIAOLONG SONG received the Ph.D. degree from the Changchun University of Science and Technology. He is currently a Lecturer and the Director of the Changchun Branch of Cloud Storage, Chinese Academy of Sciences. He is also a Lecturer with the School of Computer Science and Technology. His main research interests include big data, cloud computing research, and project implementation. He is a member of CCF.

LI LI received the master's and Ph.D. degrees from the Changchun University of Science and Technology, China. She is currently a Professor with the School of Computer Science and Technology, Changchun University of Science and Technology. She is a member of CCF.