

Research Article

Energy-Saving Virtual Machine Placement Method for User Experience in Cloud Environment

Shanchen Pang , Kexiang Xu, Shudong Wang , Min Wang, and Shuyu Wang

College of Computer Science and Technology, China University of Petroleum, Qingdao, Shandong 266580, China

Correspondence should be addressed to Shudong Wang; shudongwang2013@sohu.com

Received 20 November 2019; Revised 10 February 2020; Accepted 3 March 2020; Published 12 May 2020

Academic Editor: Haopeng Zhang

Copyright © 2020 Shanchen Pang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Green computing focuses on the energy consumption to minimize costs and adverse environmental impacts in data centers. Improving the utilization of host computers is one of the main green cloud computing strategies to reduce energy consumption, but the high utilization of the host CPU can affect user experience, reduce the quality of service, and even lead to service-level agreement (SLA) violations. In addition, the ant colony algorithm performs well in finding suitable computing resources in unknown networks. In this paper, an energy-saving virtual machine placement method (UE-ACO) is proposed based on the improved ant colony algorithm to reduce the energy consumption and satisfy users' experience, which achieves the balance between energy consumption and user experience in data centers. We improve the pheromone and heuristic factors of the traditional ant colony algorithm, which can guarantee that the improved algorithm can jump out of the local optimum and enter the global optimal, avoiding the premature maturity of the algorithm. Experimental results show that compared to the traditional ant colony algorithm, min-min algorithm, and round-robin algorithm, the proposed algorithm UE-ACO can save up to 20%, 24%, and 30% of energy consumption while satisfying user experience.

1. Introduction

Cloud computing [1] is a further extension of distributed computing, parallel computing, and grid computing. It provides a variety of hardware services, infrastructure services, platform services, software services, and storage services through the Internet to achieve cloud computing. Cloud computing users can use it on demand, pay for it on demand, and have good scalability.

With the development of cloud services, data centers become extremely large and need to consume a lot of energy. In [2], the authors pointed out that according to the statistics of the U.S. Department of Energy, the annual power consumption of data centers has accounted for 1.5% of the annual power generation in the United States. It is estimated that the annual power consumption of China's data centers is roughly the same as that of the United States and has reached the annual power generation of the Three Gorges. Energy consumption has become the most challenging problem in the current data center, so how to reduce energy

consumption is an urgent problem to be solved in cloud computing research.

Virtual machine consolidation is one of the common strategies to reduce energy consumption. The virtual machine consolidation includes host overload/underload detection, virtual machine selection, virtual machine placement [3–5], and virtual machine migration [6, 7]. Virtual machine migration can take a certain amount of time, consume a lot of resources, and affect the operation of other virtual machines on the server, resulting in system performance degradation. In addition, additional network storage system and high-speed communication network are needed to realize virtual machine migration. If there are a large number of virtual machine migrations in the cloud computing system, it will lead to failure to meet the service-level agreement. Therefore, the most direct way to reduce energy consumption in data centers is to optimize the initial placement of virtual machines, and in the placement process, in order to prevent excessively minimizing energy consumption and affecting the user experience, this article

introduces user experience indicators as a constraint on the placement of energy-saving virtual machines [8].

Because the ant colony algorithm has a good positive feedback mechanism and a distributed search method, it is very good to find suitable computing resources in unknown networks. This paper proposes a method for placing energy-saving virtual machines based on user experience. By modifying the pheromone and heuristic factor update methods and defining the parameter regulatory factor (RF), the improved roulette probability selection mechanism guides the ant to search, avoiding the algorithm to enter the local optimal due to premature convergence, and then finds the optimal solution for virtual machine placement, effectively reduces energy consumption, and achieves a balance between energy consumption and user experience.

On the basis of the above discussion, this paper optimizes the energy problem of dynamic virtual machine placement in the data center on the basis of satisfying the user experience. The contributions of our work to existing literature can be summarized as follows:

- (1) Considering the constraints of user experience while optimizing energy consumption, the energy-saving virtual machine placement system model of the cloud data center is established.
- (2) In order to adapt to the dynamic characteristics of cloud computing resources and meet user experience, solve cloud computing resource scheduling problems, and reduce data center power consumption, we propose a user experience-oriented energy-saving virtual machine placement method (UE-ACO).
- (3) The advantage of this algorithm is that it has a positive feedback mechanism, fast convergence speed, and can effectively solve the problem that the traditional ant colony algorithm easily falls into a local optimum. The algorithm does a good job of finding suitable computing resources in unknown networks.

2. Related Works

Virtual machine placement is a way to dynamically map virtual machines to physical machines to optimize resource utilization. The virtual machine is placed reasonably to achieve energy-saving purposes by using different scheduling algorithms.

One kind of algorithm for the virtual machine placement problem is the heuristic algorithm. Yadav et al. [9] proposed three adaptive models regarding high energy consumption and service-level agreement violations. While minimizing energy consumption, the SLA violations when selecting virtual machines from overloaded hosts were considered. The method can reduce energy consumption. At the same time, cloud data center performance can be guaranteed. Zhang et al. [10] studied reducing the energy consumption of mobile cloud data centers during periods of insufficient and overloaded virtual machines. They proposed an adaptive heuristic energy-aware algorithm that detects the history of

CPU utilization for virtual machine selection and dynamic merging, which minimizes total energy consumption and improves service quality. In [11], when solving the problems of cloud data center energy consumption and SLA violations, compared with most of the current works, they considered two additional important factors: (1) analyzing the robustness of the upper threshold of CPU usage. (2) Based on the CPU utilization prediction, when selecting the virtual machine of the overloaded host, the performance degradation time and SLA violations are reduced. A heuristic algorithm is proposed to detect the least squares regression of the overloaded host and select the virtual machine from the overloaded host minimum utilization prediction to reduce energy consumption with minimal SLA cost. In [12], for the problem of virtual machine integration, an energy-aware dynamic virtual machine selection algorithm was proposed to merge virtual machines from overloaded or underloaded to minimize energy consumption and maximize service quality.

Another kind of algorithm used to solve the VM placement problem is the metaheuristic algorithm (such as ant colony algorithm and particle swarm optimization), which are considered as good candidates for finding the approximate optimal solution. Gao et al. [13] proposed an energy-saving ant colony algorithm for network-aware virtual machine placement. This algorithm comprehensively considers CPU, memory, and bandwidth resources. Under these three constraints, the maximum cost reduces energy costs. Liu et al. [14] combined the order exchange and migration local search technology and proposed a virtual machine placement based on the ant colony algorithm. Experiments have proved that it works well in both homogeneous and heterogeneous cloud environments. On the premise of not wasting power and maintaining resource utilization, J et al. [15] proposed a direct guide ant colony system to achieve maximum resource utilization and minimum energy consumption. The algorithm also obtained better results. Saini [16] proposed a multiobjective ant colony system algorithm for virtual machine placement. The algorithm obtained a set of Pareto sets to minimize the total waste of resources and energy consumption. Zhang et al. [17] proposed a virtual machine layout optimization model based on an optimized ant colony algorithm based on virtual machine utilization parameters, multiobjective optimization strategies, and feature metrics, which solved redundant power consumption due to waste of idle resources of the physical machine problem.

In the study of green computing and virtual machine placement, we discussed the above methods and achieved good results. In the heuristic algorithm part, virtual machine placement takes into account the degradation of system performance during the virtual machine activity, so the user experience is considered while reducing energy consumption. In the metaheuristic algorithm, most choose to maximize resource utilization and minimize energy consumption, so they ignore system performance and user experience. Therefore, the virtual machine placement algorithm proposed in this paper combines the user experience as a constraint with the ant colony algorithm. Under the

premise of ensuring the user experience, the swarm intelligence algorithm is used to solve the virtual machine placement problem, thereby reducing energy consumption.

3. System Model

The system consists of a host management module, a user experience module, and a virtual machine scheduling module. The energy-saving virtual machine placement model is shown in Figure 1. The virtual machine demand flow enters the waiting queue. It is coordinated by the relevant module and is finally placed in the cloud computing data center. The host management module monitors the working status of each machine in real time, gives instructions for optimizing power consumption, and passes the load status of the machine and the virtual machine queue status to the user experience module. The user experience module receives the parameter values passed by the host management module and the waiting queue and calculates the user experience score. The virtual machine scheduling module accepts the information of the user experience module and waiting queue and allocates the virtual machine to a reasonably available host.

3.1. Host Management Module. Host management module manages the hosts of the cloud data center in an energy-saving way. The module monitors each machine for good operation to ensure the user experience and issues instructions for optimizing power consumption for machines with excessive CPU utilization. According to Pang et al. [18], the running state of each host is defined as follows:

$$\text{State}(j) = \frac{\text{RunVM}(j)}{\text{MaxVM}(j)}, \quad (1)$$

where $\text{RunVM}(j)$ represents the number of virtual machines carried by the current host and $\text{MaxVM}(j)$ represents the maximum number of virtual machines that the current host can host.

3.2. User Experience Module. Cloud computing user experience is the overall effect of users using cloud computing services, which determines users' satisfaction with cloud computing services [19]. In this paper, a user experience model with feedback is proposed combining the users and cloud services. User experience is described in terms of user needs, user perception, cloud service provision, and cloud service implementation.

There are various evaluation methods of user experience. According to [20–22], guided by some references of the objective evaluation method, this paper analyses the parameters affecting user experience in cloud computing services and then defines the user experience model. There are four parameters in the model (availability, SLA violation ratio, run time, and price). The user experience model and its parameters are defined as follows:

$$\text{Score}(\text{Avail}, \text{OTF}, T_{\text{total}}, P_{\text{price}}), \quad (2)$$

$$\text{Avail} = \frac{T_{\text{run}} - T_{\text{break}}}{T_{\text{run}}},$$

where T_{run} represents the total time available for the service and $T_{\text{break}}(i)$ denotes the total time of service interruption.

$$\text{OTF}(u_t) = \frac{t_0(u_t)}{T_{\text{run}}}, \quad (3)$$

where $t_0(u_t)$ represents the total time of the overload.

$$T_{\text{total}} = T_{\text{wait}} + T_{\text{run}}, \quad (4)$$

where T_{wait} represents the waiting time of the virtual machine in the queue; P_{price} represents the cost of this cloud service.

The user experience score is defined as follows:

$$\text{score} = \alpha_1 \cdot \text{Avail} + \alpha_2 \cdot \frac{1}{\text{OTF}(u_t)} + \alpha_3 \cdot \frac{1}{T_{\text{total}}} + \alpha_4 \cdot P_{\text{price}}, \quad (5)$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are the regression coefficients. This paper obtains a large number of data such as task execution time, CPU utilization, task waiting time, host overload execution time, memory capacity, and network bandwidth by the statistical method and calculates the regression coefficient.

Combining users, cloud services, and the perspectives of front and rear services, the relationship between the four perspectives of cloud computing user experience can be obtained, which is as shown in Figure 2.

The user requirements are the requirements before the user uses the cloud service, and the user perception is the perception after the user uses the cloud service. The cloud service provision is an ideal execution capability, and the cloud service implementation is the actual execution capability.

3.3. Virtual Machine Scheduling Module. The virtual machine scheduling module accepts virtual machine queue information and places the virtual machine on an available host. This paper applies the improved ant colony algorithm to the module. The goal is to effectively reduce the energy consumption of the data center based on the user experience. The specific method will be explained in Section 5.

4. Problem Formulation

4.1. Correspondence. To further illustrate this problem, we have used a matrix to describe the mapping between virtual machines and hosts. A set of virtual machine $\text{VM}_i = \{\text{VM}_1, \text{VM}_2, \dots, \text{VM}_m\}$; it indicates that m virtual machines are in the current waiting queue. A set of host machine $\text{PM}_j = \{\text{PM}_1, \text{PM}_2, \dots, \text{PM}_n\}$; it indicates n available computing resources. The distribution relationship between the virtual machine and the host is defined as an $m \times n$ matrix $X_{m \times n}$:

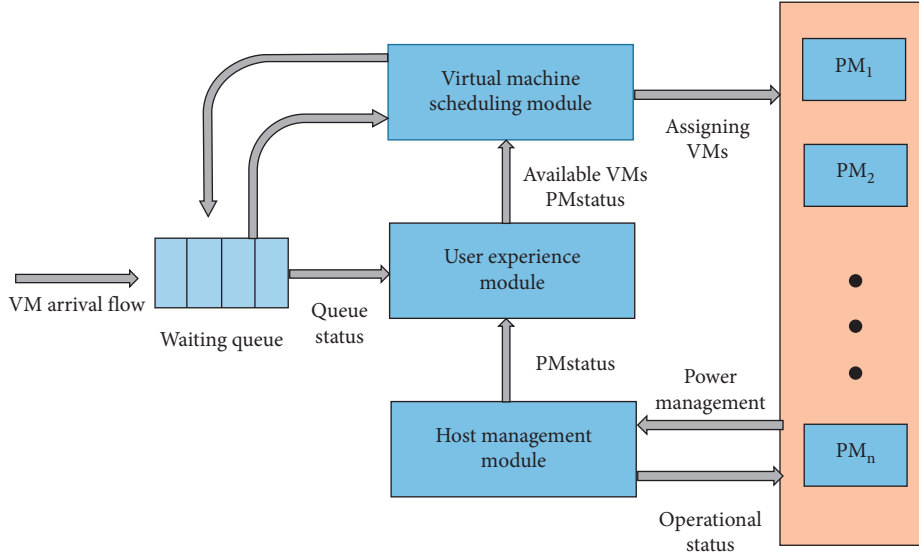


FIGURE 1: Energy-saving virtual machine placement system model.

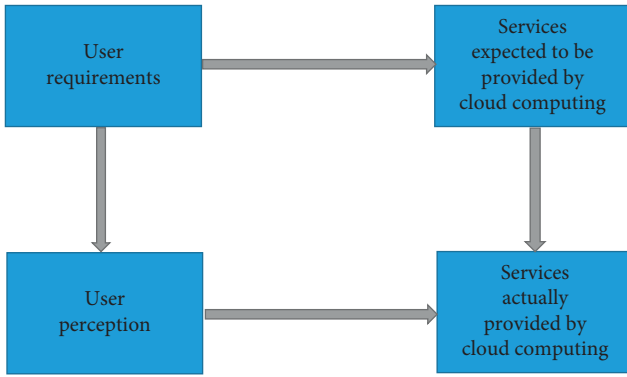


FIGURE 2: User experience model with feedback.

$$X_{m \times n} = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{pmatrix}, \quad (6)$$

where X_{ij} is VM_i running on the PM_j .

4.2. Energy Model. Relevant studies had shown that the power consumption of servers can be accurately described by the linear relationship between power consumption and CPU utilization. The analysis of this linear relationship on Dell servers confirmed this point. In order to save energy, the server was shut down when it was idle [23]. According to Malekloo et al. [24], the power consumption of the host is defined as a function of CPU utilization as follows:

$$P_j = (P_j^{\text{busy}} - P_j^{\text{idle}}) \times U_j^p + P_j^{\text{idle}}, \quad (7)$$

where P_j^{idle} and P_j^{busy} are the average power values when the host is idle and fully utilized and U_j^p represents the resource

utilization of the host j at current power. In our simulation experiments, this value was fixed at 162 and 215 watts.

The total power of the host is defined as follows:

$$\text{Min} \sum_{j=1}^n P_j = \sum_{j=1}^n \left[Y_j \times \left((P_j^{\text{busy}} - P_j^{\text{idle}}) \times \sum_{i=1}^m (X_{ij} \times R_i^{VM}) + P_j^{\text{idle}} \right) \right], \quad (8)$$

where R is a set of CPU resources required by each virtual machine. The variable Y_i represents the on/off state of the host (1 means open; 0 means closed) and X_{ij} is whether to assign a virtual machine to the host.

4.3. Resource Allocation. The ant colony algorithm is a heuristic algorithm with distinct advantages and disadvantages.

4.3.1. The Positive Feedback Mechanism. The search process will converge and eventually reach the optimal solution.

4.3.2. Distributed Computing. The search process adopts the distributed computing mode, and many units carry out parallel computing at the same time, which greatly improves the computational ability and efficiency of the algorithm.

4.3.3. Easy to Combine. It is easy to combine with other optimization algorithms.

4.3.4. Inherent Defect. The disadvantage is that it is easy to fall into the local optimum, and it is difficult to jump from local optimum to global optimal. The ant colony algorithm does a good job of finding the right computing resources in an unknown network topology [25].

4.4. Initialization. During the initialization phase, the ants are randomly placed on the host, and the pheromone strength is initialized:

$$\tau_j(0) = \left(\frac{\text{Mips}_j}{N} \right), \quad (9)$$

where Mips_j represents the computing power of the current host PM_j and N is a constant.

4.5. Probability Selection Mechanism. Each ant selects the host based on the pheromone and heuristic factors. In the iterative process, P_j^k (probability selection mechanism) depends on host information (user experience score and energy consumption information), path pheromone concentration changing, and adjustment factor. Individual ants represent the allocation scheme of the virtual machine. At time t , the probability that the ant k selects the next host for the virtual machine is defined as follows:

$$P_j^k(t) = \begin{cases} \frac{[\tau_j(t)]^\alpha [\eta_j]^\beta \text{RF}}{\sum_{i=1}^n [\tau_i(t)]^\alpha [\eta_i]^\beta}, & \text{if } l, j, \dots, n, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where α and β are the weighting factors; it represents the importance of pheromones and heuristic factors. $\tau_j(t)$ is the pheromone concentration on the host PM_j at time t . η_j is a heuristic factor, $\eta_j = f(t)$; $f(t)$ represents the profit function of the physical machine while it is running the virtual machine:

$$f(t) = \lambda_1 \cdot \text{Score}(t) + \lambda_2 \cdot \frac{1}{\sum_{j=1}^n P_j(t)}, \quad (11)$$

where λ_1 and λ_2 are model-specific coefficients.

The heuristic factor of the original ant colony reflects the relative importance of the amount of information accumulated by the ant during the movement process in guiding the ant colony search process. The larger the value, the greater the possibility that the ant chooses to walk through the path before, and the randomness of the search is reduced. Therefore, it is easy to fall into local optimum. Based on the shortcomings of the original ant colony heuristic factor, setting the parameter RF can effectively solve the problem.

RF is an adjustment factor. It is related to the CPU utilization of the current host and the initial CPU utilization threshold. The value of RF directly affects the subsequent placement of the virtual machine. Set the CPU utilization threshold during the initialization phase (by data analysis, this article sets it to 75%).

If the CPU utilization of the current host PM_j is higher than the set threshold, the state transition probability of the host PM_j is reduced by adjusting the factor RF_1 , and the probability of the system placing the virtual machine on the host PM_j is reduced. At the same time, if the current allocation scheme is not globally optimal, then the adjusting factor RF can also increase the probability of the ant jumping out of local optimal and increase the randomness of the

search. To a certain extent, it alleviates the inherent problem of the original ant colony algorithm that is easy to fall into local optimum.

If the current host PM_j 's CPU utilization is below the set threshold, the host can continue to place the virtual machine and update it by adjusting the factor RF_2 . The adjustment factor RF is defined as follows:

$$\text{RF} = \begin{cases} \text{RF}_1: 1 - \frac{U_j - U_0}{\sum_{j=1}^n U_j}, & U_j \geq U_0, \\ \text{RF}_2: 1, & U_j < U_0, \end{cases} \quad (12)$$

where U_j is the CPU utilization of the current host PM_j and U_0 is the CPU utilization threshold preset by the user experience module.

4.6. Pheromone Update. The pheromone update based on ants that propose executable virtual machine placement is

$$\tau_j(t+1) = (1 - \rho) \times \tau_j(t) + \Delta\tau_j, \quad (13)$$

where $\tau_j(t+1)$ represents the pheromone content of the host PM_j at the next moment. ρ represents the pheromone volatilization coefficient: $0 < \rho < 1$. $\Delta\tau_j$ represents the increment of the pheromone:

$$\Delta\tau_j = \max[f(t)_{k,r}], \quad (14)$$

where $\max[f(t)_{k,r}]$ represents the maximum profit of the ant k in the iteration r . If ant k completes the search, indicating that the ant has found an executable allocation plan, then the pheromone concentration of the host in the allocation scheme is updated. If all ants have completed the search, find the best solution among all feasible allocations and update the pheromone concentration of the host in the best scenario.

4.7. Algorithm Flow

Step 1: the host management module gets the real-time running status of the machine in the cloud data center and then performs state control on each host.

Step 2: transfer the running status of the host and the virtual machine queue state to the user experience module, and get the available host resources according to the preset CPU utilization threshold.

Step 3: initialize pheromone for available host resources in the virtual machine scheduling module.

Step 4: place all the ants randomly on the available hosts.

Step 5: the ant k calculates the probability selection mechanism based on the pheromone concentration, the heuristic factor, and the adjustment factor and selects the next host.

Step 6: update local pheromone when ant k completes the search; if not, return to Step 5.

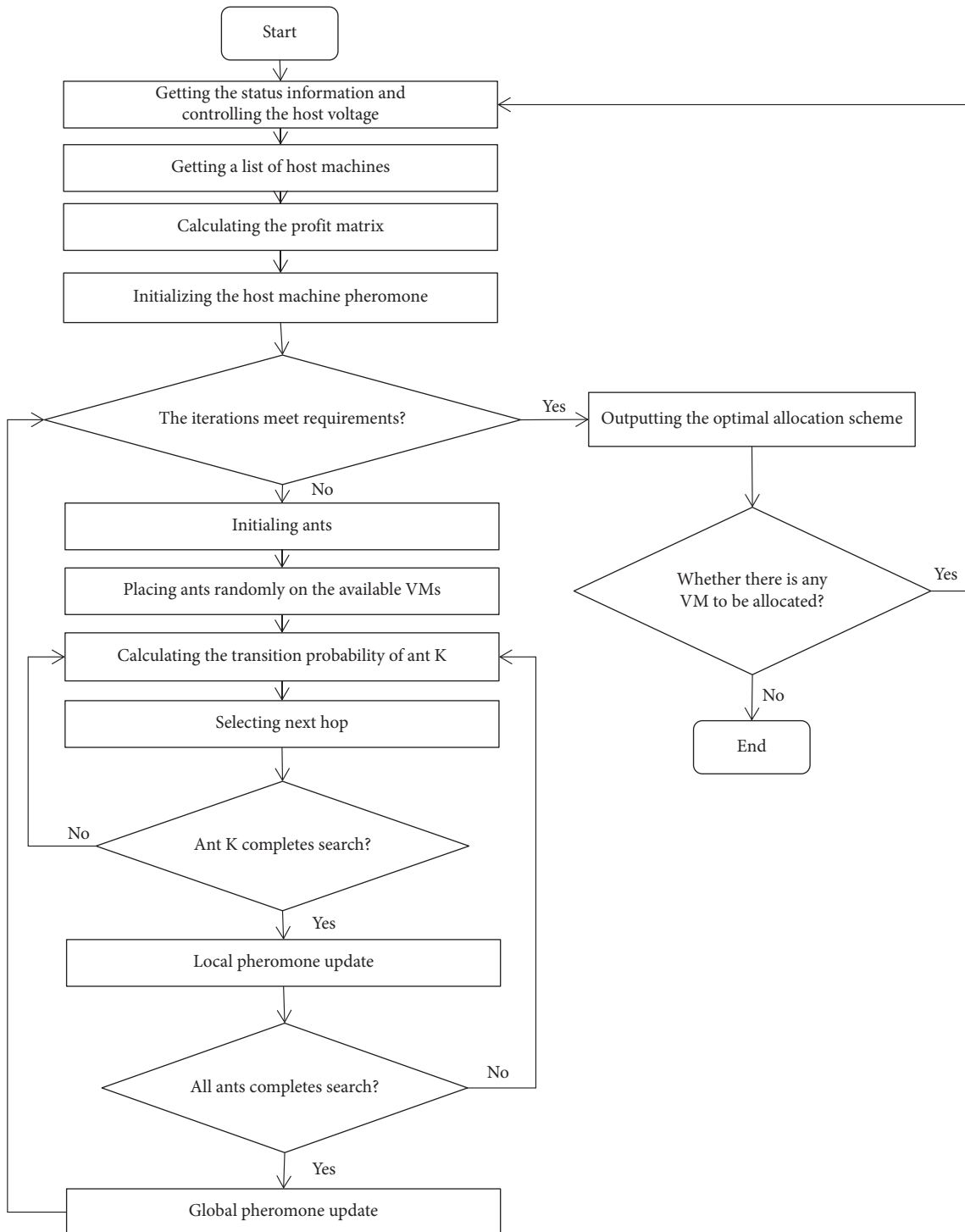


FIGURE 3: Flowchart of energy-saving ant colony algorithm.

Step 7: update global pheromone when all ants complete the search; if not, return to Step 4.

Step 8: when the number of initialization iterations is reached, the system outputs the optimal allocation scheme; if not, it returns to Step 4.

Step 9: determine if there are still virtual machine requirements in the waiting queue that are not

allocated. If yes, return to the first step; if not, end this virtual machine allocation.

The energy-saving ant colony algorithm is executed periodically, and Figure 3 depicts the main flow of the algorithm.

4.8. Algorithm Time Complexity. The total time complexity is defined as $O(\text{set CPU utilization threshold}) + O(\text{initializing$

distribution matrix) + $O(\text{calculate total energy consumption}) + O(\text{calculate Profit function}) + O(\text{comparing current CPU utilization of all hosts})$. To be specific, $O(\text{set CPU utilization threshold})$ is $O(1)$; $O(\text{initializing distribution matrix})$ is $O(m \times m)$; $O(\text{comparing current CPU utilization of all hosts})$ is $O(n)$; $O(\text{calculate total energy consumption})$ is $O(m \times n)$; and $O(\text{calculate Profit function})$ is $O(m)$. Thus, total time complexity is $O(m \times n)$.

5. Experiment and Analysis

Due to the constraints, it is very difficult to perform iterative experiments on a real data center, so we use simulation methods to evaluate the user experience and energy consumption of the proposed strategy. Therefore, we can carry out a large number of experiments with statistical significance. Specifically, by implementing the proposed algorithm on a PC with Core i7, 3.2 GHz, 8 GB RAM, and Windows 10, the simulation framework adopts the CloudSim simulation platform and uses the Java programming language to set up 6 host machines, 30 virtual machines, and 200–1,200 tasks in the simulation environment. We have written experimental data generation tools, which can generate the corresponding dataset files by setting the number of tasks, and then under the same environment, four algorithms read the same dataset files to generate a solution. The optimum is emancipated into CloudSim and simulated. Finally, the results are counted through CloudSim API.

According to the parameter settings in Tables 1 and 2, the results of the three indicators and the curve of the profit function are obtained, as shown in Figures 4–7. Table 1 describes the host, virtual machine, and task information in the simulation environment. Table 2 sets the parameters of the UE-ACO algorithm in this paper.

Figure 4 shows the difference in total execution time for the four algorithms, BACO, RR, Min-Min, and UE-ACO. Task completion time is the total completion time of a batch of submitted tasks, including task waiting time and task execution time. The RR algorithm takes the longest time to complete tasks, and the UE-ACO algorithm is the shortest. BACO algorithm is a basic traditional ant colony algorithm, the heuristic factors do not change with the dynamic allocation of virtual machines, ants will choose to move directly to resource-rich hosts, and its effect will deteriorate with the increase of the number of tasks. RR algorithm is a self-contained polling algorithm in CloudSim. The RR algorithm involved in this paper uses the minimum virtual machine to run on the largest idle host. This algorithm takes into account the changes of resources, but it ignores the high increase of energy consumption.

Figure 5 shows a comparison of the average waiting time using four algorithms. Task waiting time is an important indicator of user experience, which directly affects the score of user experience and then affects the execution process of the algorithm. The results show that the energy-saving virtual machine placement method oriented to user experience can better meet user needs.

Figure 6 shows a comparison of energy consumption using four algorithms. The energy consumption of the cloud

TABLE 1: Parameters of CloudSim.

Type	Parameter	Value
Data center	Number of data centers	1
	Number of hosts per data center	6
Host	Total number of hosts	6
	Host memory	2048–4096
	Bandwidth	10000
VM	Total number of VMs	30
Task	Total number of tasks	200–1,200

TABLE 2: Parameters of the UE-ACO algorithm.

Parameter	Value
Number of ants in the colony	100
Number of iterations	20
α	1
ρ	0.5

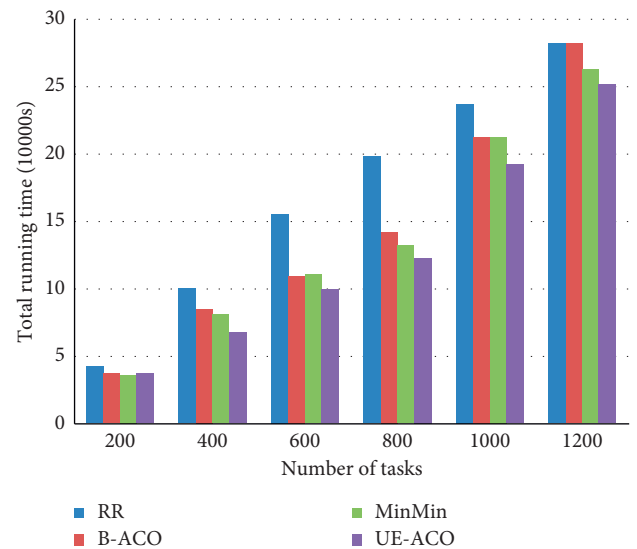


FIGURE 4: Execution time of tasks.

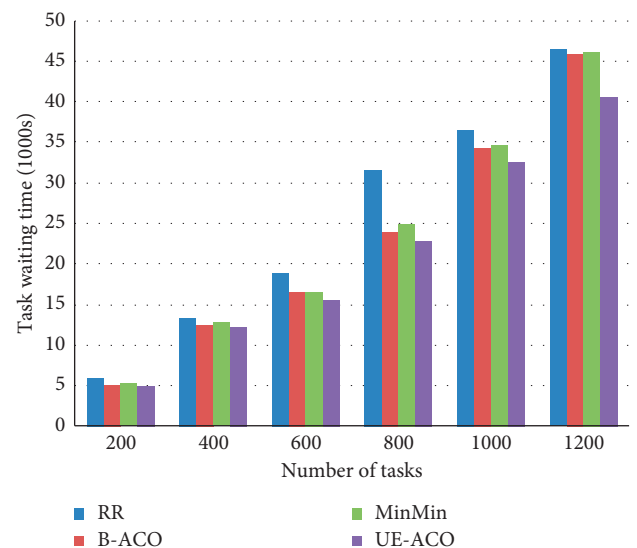


FIGURE 5: Task's average waiting time.

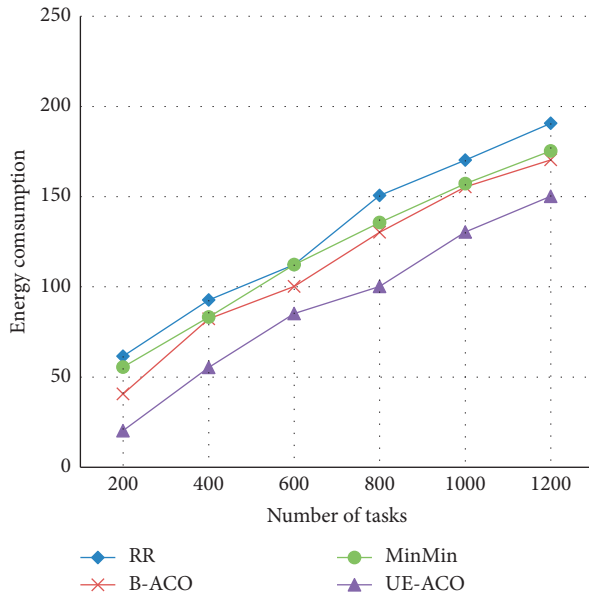


FIGURE 6: Energy consumption of task running.

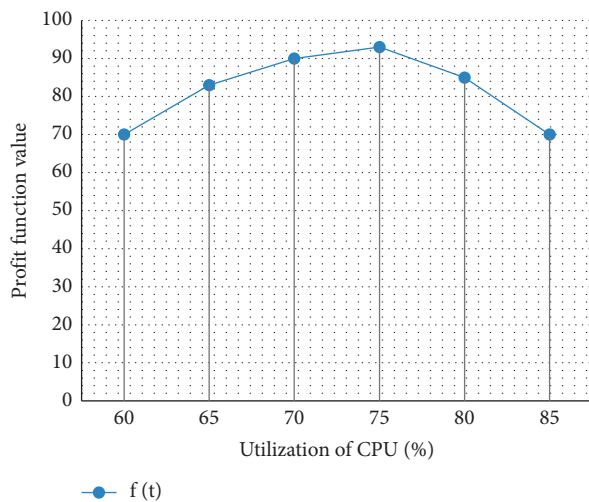


FIGURE 7: The profit function of the algorithm.

computing data center increases with the number of tasks. Compared with RR, BACO, and Min-Min, the proposed UE-ACO algorithm can significantly reduce the energy consumption of the system. RR, BACO, and Min-Min algorithms only consider the completion time of tasks while ignoring energy consumption. UE-ACO algorithm guides pheromone and heuristic factor updates by the user experience score and system energy consumption, so it can effectively reduce energy consumption on the basis of meeting task completion time.

Figure 7 shows the change of profit function of the proposed algorithm when setting different CPU utilization thresholds. With the increase of threshold, user experience scores are good, and energy consumption has a greater impact on the profit function. The profit function reaches its maximum when the threshold is set to 75%. When the threshold value exceeds 75%, it can continue to reduce

energy consumption to a certain extent, but the execution state of the host PM_j is significantly worse, and the user experience score has a greater impact on the profit function, so the value of the profit function decreases as a whole.

6. Conclusion

In order to solve the problems of user experience and energy consumption in the data center, the key indicators affecting user experience are analyzed, the user experience is modeled mathematically, the system model of virtual machine allocation in the cloud data center is described, and a comprehensive management strategy is formed. A user experience-oriented placement method of the energy-saving virtual machine is proposed. By modifying the pheromone updating method and heuristic factor in the ant colony algorithm and setting CPU utilization threshold, the energy consumption of the cloud computing data center is effectively reduced on the basis of guaranteeing user experience and improved the inherent defect that the traditional ant colony algorithm is easy to fall into local optimum. In the future work, we plan to establish a more perfect virtual machine placement system model. In addition, more details of user experience models can be described or a new model can be constructed. This kind of model is very necessary in the scheduling problem of virtual machines in cloud data centers. Finally, memory and network bandwidth are added to the energy model to calculate energy consumption more accurately.

Data Availability

The cloud resource allocation data used to support the findings of this study have not been made available because of secrecy.

Conflicts of Interest

Regarding the publication of this article, the authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank International Networks Service and Bio-Computing Innovation Team from the college of Computer and Communication Engineering in China University of Petroleum, thanks to the team members for their discussion and technical support. This work was supported by the National Natural Science Foundation of China (61672033, 61873280, 61873281, 61972416, 61672248, and 61902430), National Key Research and Development Project (no.2018YFC1406204), Key Research and Development Program of Shandong Province (no.2019GGX101067), the Natural Science Foundation of Shandong Province (no. ZR2019MF012), Taishan Scholars Fund (no. ZX20190157), Independent Innovation Research Project (no. 18CX02152A), and Fundamental Research Funds for the Central Universities (no. 19CX02028A).

References

- [1] X. Luo, "CloudTalking about the development of cloud computing," *Electronic World*, vol. 566, no. 8, p. 104, 2019.
- [2] S. Feng, L. Gao, and F. Zhou, "Analysis of energy saving status and challenges in China's data centers," *China Energy*, vol. 37, no. 11, pp. 23–26, 2015.
- [3] S. Jangiti, E. Sriram, R. Jayaraman et al., "Resource ratio based virtual machine placement in heterogeneous cloud data centres," *Sādhanā*, vol. 44, no. 12, pp. 1–6, 2019.
- [4] T. P. Shabeera, S. D. M. Kumar, S. M. Salam et al., "Optimizing VM allocation and data placement for data-intensive applications in cloud using ACO metaheuristic algorithm," *Engineering Science & Technology An International Journal*, vol. 20, no. 2, pp. 616–628, 2016.
- [5] D. Pei, "Dual objective optimal deployment method of virtual machine based on ant colony algorithm," *Computer Engineering and Design*, vol. 39, no. 11, pp. 164–169, 2018.
- [6] M. H. Ferdaus, M. Murshed, R. N. Calheiros et al., "Multi-objective decentralized dynamic virtual machine consolidation using ACO metaheuristic in computing clouds," *Cornell University*, 2017.
- [7] H. Xu, Y. Liu, W. Wei, and Y. Xue, "Migration cost and energy-aware virtual machine consolidation under cloud environments considering remaining runtime," *International Journal of Parallel Programming*, vol. 47, no. 3, pp. 481–501, 2019.
- [8] C. Liu and C. Zou, "Research on cloud service strategy based on user experience," *Journal of Chinese Computer Systems*, pp. 1203–1206, 2016.
- [9] R. Yadav, W. Zhang, O. Kaiwartya et al., "Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing," *IEEE Access*, p. 1, 2018.
- [10] R. Yadav and W. Zhang, "MeReg: managing energy-SLA tradeoff for green mobile cloud computing," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 6741972, 11 pages, 2017.
- [11] R. Yadav, W. Zhang, K. Li et al., "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center," *Wireless Networks*, vol. 26, no. 3, pp. 1905–1919, 2020.
- [12] R. Yadav, W. Zhang, H. Chen et al., "Mums: energy-aware vm selection scheme for cloud data center," in *Proceedings of the 2017 28th International Workshop on Database and Expert Systems Applications (DEXA)*, IEEE, Lyon, France, pp. 132–136, August 2017.
- [13] C. Gao, H. Wang, L. Zhai et al., "An energy-aware ant colony algorithm for network-aware virtual machine placement in cloud computing," in *Proceedings of the 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, Wuhan, China, January 2017.
- [14] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 113–128, 2018.
- [15] J. Srija, R. R. John, and G. M. Kanaga, "An element search ant colony technique for solving virtual machine placement problem," *Journal of Physics: Conference Series*, vol. 892, no. 1, Article ID 012007, 2017.
- [16] R. Saini, "A multi-objective ant colony system Algorithm for virtual machine placement," *International Journal of Engineering Research and Applications*, vol. 7, no. 1, pp. 95–97, 2017.
- [17] L. Zhang, Y. Wang, L. Zhu, and W. Ji, "Towards energy efficient cloud: an optimized ant colony model for virtual machine placement," *Journal of Communications and Information Networks*, vol. 1, no. 4, pp. 116–132, 2016.
- [18] S. Pang, W. Zhang, T. Ma et al., "Ant colony optimization algorithm to dynamic energy management in cloud data center," *Mathematical Problems in Engineering*, pp. 1–10, 2017.
- [19] M. R. Mahmud, M. Afrin, M. A. Razzaque, M. M. Hassan, A. Alelaiwi, and M. Alrubaian, "Maximizing quality of experience through context-aware mobile application scheduling in cloudlet infrastructure," *Software: Practice and Experience*, vol. 46, no. 11, pp. 1525–1545, 2016.
- [20] M. J. OSullivan and D. Grigoras, "Context aware mobile cloud services: a user experience oriented middleware for mobile cloud computing," in *Proceedings of the the 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (Mobile Cloud 2016)*, IEEE, Oxford, UK, May 2016.
- [21] Y. Liu, C. Li, and Z. Yang, "Tradeoff between energy and user experience for multimedia cloud computing," *Computers & Electrical Engineering*, vol. 47, pp. 161–172, 2015.
- [22] J. Wei, E. Juarez, M. Garrido, and F. Pescador, "Maximizing the user experience with energy-based fair sharing in battery limited mobile systems," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 3, pp. 690–698, 2013.
- [23] H. Kasture, *A Hardware and Software Architecture for Efficient Datacenters*, Electrical Engineering and Computer, Bern, Switzerland, 2017.
- [24] M.-H. Malekloo, N. Kara, and M. El Barachi, "An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments," *Sustainable Computing: Informatics and Systems*, vol. 17, pp. 9–24, 2018.
- [25] Y. Xiao, J. Jiao, D. Qiao et al., "Review of the basic principles and applications of ant colony algorithm," *Light Industry Technology*, vol. 34, no. 232, pp. 73–76, 2018.