Design and application of a green computing cloud platform based on scalable orchestration

Manni Xiong[a*], Chengming Wang[b], Hengjia Chang[c], Jiadong Song[d]

Changjiang Survey, Planning, Design and Research Co., Ltd., Wuhan, China

[*a] xiongmanni@cjwsjy.com.cn; [b]wangchengming@cjwsjy.com.cn; [c]changhengjia@cjwsjy.com.cn; [d]songjiadong@cjwsjy.com.cn

Abstract. With the gradual development and popularization of cloud computing technology, cloud computing has become an important support for government and enterprise information construction. However, traditional cloud computing platforms have problems such as low resource utilization and poor energy efficiency management, and current energy efficiency research cannot guarantee the parallel application quality and efficiency. Therefore, this article proposes a green computing cloud platform model based on scalable orchestration, and studies the implementation of a polymorphic ant colony algorithm improved on consistency hashing algorithm, which improves the resource utilization of the cloud platform, reduces energy consumption, and realizes the green development of cloud data centers. On the one hand, based on the energy consumption analysis of cloud data centers, models for resource management, prediction, and elastic scaling have been proposed to achieve dynamic switch control of physical machines in cloud data centers, improving the effective resource utilization and energy efficiency management of cloud data centers; On the other hand, in response to the problem of poor performance caused by traditional data center load imbalance, a dynamic feedback mechanism is adopted to calculate the load rate of each node based on the physical machine CPU, memory and other information collected by cluster monitoring. Based on the resource scheduling strategy that integrates the consistent hash load balancing algorithm and the polymorphic ant colony scheduling algorithm, the balanced distribution of scalable scheduling tasks is achieved, improving the efficiency of resource balancing scheduling.

Keywords: component; Green computing; Elastic expansion and contraction; Resource perception; Improving Ant Colony Algorithm; Resource allocation; Energy efficiency management

1 Introduction

With the increasing size of the cloud computing market, the scale of data centers carrying cloud services is also growing, which also leads to an increasing demand for energy in data centers. This not only brings high operating costs to operators, but also emits a large amount of carbon dioxide, exacerbating the greenhouse effect. At the same time, low resource utilization is also an important factor affecting energy consumption in cloud data centers. In order to find efficient resource management mechanisms to improve resource utilization and achieve green computing, domestic and foreign scholars have conducted many studies on the energy consumption of cloud computing resource management. Wang Yuan[1] proposed a CPU usage prediction algorithm based on deep belief networks and particle swarm optimization algorithm, as well as a multi cloud data center resource allocation method based on request prediction, for single or multiple cloud data centers. Although this method has high prediction accuracy, the use of swarm intelligence algorithm optimization results in longer time consumption than other single prediction algorithms, and only considers single step prediction without studying multi-step prediction. Zhou Zhi[2] proposed an online optimization system for cross domain data center energy consumption cost and carbon emissions collaborative optimization based on a progressive research approach of energy efficiency optimization, incentive mechanisms, and application expansion. However, the optimization and management of energy efficiency in data centers without considering multiple energy supply options. Ismaeel et al.[3] developed a predictive model that combines k-means clustering techniques and extreme learning machines. By using models to estimate future VM requests in historical data centers and improving the accuracy of prediction models, energy savings, performance improvements, and increased profits for customers and service operators have been achieved. Kholidy[4] proposed a swarm intelligence based method to predict CPU utilization, memory utilization, response time, and throughput. It combines multiple support vector regression models and autoregressive comprehensive moving average models for prediction, and uses PSO to select the best features and estimate the parameters of these two models.

Based on the above research status, although a large number of scholars at home and abroad have conducted many studies on green computing, mainly focusing on reducing energy consumption or energy costs to achieve the goal of energy conservation and emission reduction in data centers. But at present, research only focuses on energy efficiency management or resource scheduling, without considering the content of user business needs. In addition, during the promotion and application process, due to the random user business load, current research cannot guarantee the parallel energy efficiency and quality efficiency. This article proposes a green computing cloud platform model based on the above research and business operation requirements, and studies the construction of green cloud data centers from two aspects: automatic perception and orchestration of global resources and elastic scaling based on business attributes. On the one hand, based on the energy consumption analysis of cloud data centers, models for resource management, prediction, and elastic scaling have been proposed to achieve dynamic switch control of physical machines in cloud data centers; On the other hand, based on the physical machine CPU, memory, and other information collected by cluster monitoring, a dynamic feedback mechanism is adopted to calculate the load rate of each node. Based on the integration of consistent hash load balancing algorithm and polymorphic ant colony scheduling algorithm, a cloud platform resource balancing scheduling strategy is implemented to achieve balanced distribution of scalable scheduling tasks and improve the efficiency of resource balancing scheduling. To implement the "dual carbon" strategy, green computing has moved from an optional option to a mandatory one. This method has reference value for designing cloud platforms based on green computing.

2 A solution for Green Computing Cloud Platform based on Scalable Orchestration

2.1 Green Computing Cloud Platform Model

Based on the dynamic start stop of idle devices, the green computing cloud platform model arranges and scales the actual occupied resources from the perspective of user business, and controls reserved resources with configurable load rates, effectively achieving energy efficiency control of green cloud data centers. The green computing cloud platform model is shown in Figure 1, which mainly includes modules such as orchestrator, scaling strategy, cluster monitoring, and multi cloud collaboration. The cluster monitoring service is equipped with an energy efficiency control switch, set to True, which will determine resource occupancy and shutdown quantity based on preset rules, and then operate the server shutdown. In the orchestration process, a scaling strategy is introduced to dynamically adjust the server's operating status (such as sleep, wake-up, etc.) based on workload, network traffic, and disk IO status, in order to optimize server energy consumption.

The energy consumption of data centers is mainly generated by components such as the CPU, disks, power supply, and memory of physical machines. Numerous studies have shown that, CPU utilization is an important factor affecting the energy consumption of physical machines[1]. This article uses CPU utilization to calculate the energy consumption of physical machines, as shown in formula 1.

$$PME_k^n = \begin{cases} \sigma \times \left[\alpha_k^n \times PM_k^{n,work} + (1-\alpha_k^n) \times PM_k^{n,idle}\right] \\ \sigma \times PM_k^{n,standby} \end{cases} \tag{1}$$

Among them, PME_k^n represents the energy consumption of the k-th physical machine in the nth data center, α_k^n represents the CPU utilization of the physical machine, $PM_k^{n,work}, PM_k^{n,idle}, PM_k^{n,standby}$ represents the power consumption of the k-th physical machine in the n-th cloud data center when it is in working, idle, and standby states, respectively, σ represents the duration.

Therefore, the energy consumption of the cloud data center DC_n is shown in formula 2.

$$DCE_n = \sum_{k=1}^{k=K} PME_k^n \tag{2}$$

The total energy consumption of all data centers is shown in formula 3.

$$E = \sum_{n=1}^{k=N} DCE_n \tag{3}$$

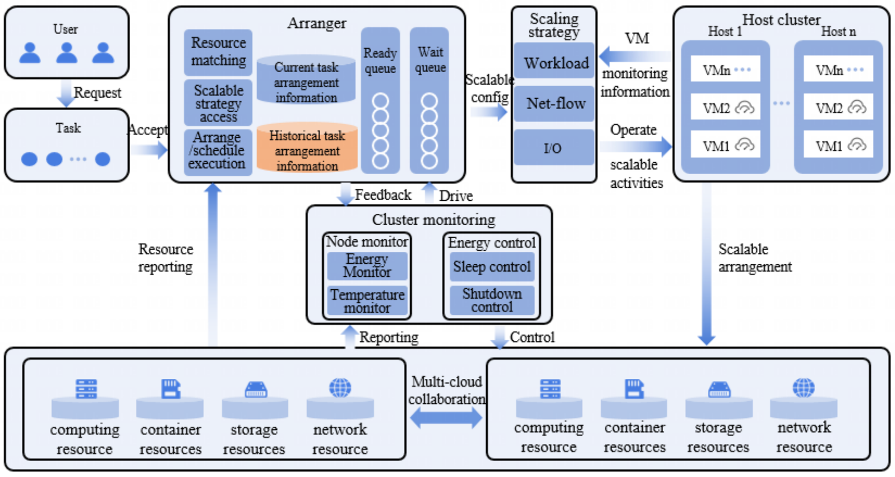

Fig. 1. Green computing cloud platform model

The green computing solution based on scalable orchestration proposed in this study starts from receiving task requests, arranges and matches appropriate nodes to carry user application services according to the cloud infrastructure type requested by the template, and combines scaling strategies to control the expansion and contraction of user applications. During this process, the orchestrator interfaces with cluster monitoring, dividing different host aggregates based on their service characteristics (computationally intensive, memory read/write, disk IO, etc.) based on the computing node information reported by each cloud platform. Then, the resource usage of each host aggregation is counted, and the remaining nodes are controlled to switch on and off according to the configured overload rate. For example, the computationally intensive host aggregation reported by the resource includes 10 physical machines, but their CPU and memory usage are quantified as 3 and 5, respectively. Based on the maximum resource consumption, virtual resources use 5 physical machines. At this time, cluster monitoring will shut down excess machines according to the configured overload rate (default of 50%), that is, shut down 2 servers. The number of server shutdowns is equal to the total number reported by the host aggregation minus the number of real-time occupied machines, and then minus the number of machines occupied by the overload rate. The resource control model is shown in formula 4.

$$SN_n = \lfloor DCN_n - \mathrm{MAX}(N_{CPU}, N_{mem})(1 - \beta) \rfloor \tag{4}$$

Among them, SN_n represents the number of physical machines that need to be shut down in the nth data center, DCN_n represents the number of physical machines in the nth data center, N_{CPU} represents the physical machine usage calculated based on CPU usage, N_{mem} represents the physical machine usage calculated based on memory usage, β represents the overload rate of the system configuration, which defaults to 50%.

When quantifying the actual usage of cluster servers, on the one hand, it is necessary to consider CPU and memory overmatching, and on the other hand, it is necessary to consider the CPU and memory reserved after physical machine virtualization. The number of machines controlled by overload rate is a dynamically changing quantification, mainly aimed at dynamically controlling the reserved amount of resources occupied by scalable user services based on the load of the cluster, and avoiding resource constraints during peak periods. In addition, in extreme cases, when the reserved resources cannot support business resource scaling, the reserved resources can maintain buffer time for turning on and off machines.

The cluster monitoring component utilizes API adaptation interfaces to interface with multiple data centers, collect device information of host clusters, including computing, storage, network, containers, and other resources, and then manage the resource utilization balance between nodes and data centers. The load of each physical machine refers to the average utilization rate of CPU, memory, disk, and network bandwidth within a scheduling domain. The imbalance degree of physical machines is shown in formula 5.

$$PS_{iunbalanced} = \frac{1}{n}\left(\sum_{i=1}^{n}\left|PM_{i\cdot cpu} - PM_{i\cdot cpu\cdot avg}\right| + \sum_{i=1}^{n}\left|PM_{i\cdot mem} - PM_{i\cdot mem\cdot avg}\right| + \sum_{i=1}^{n}\left|PM_{i\cdot disk} - PM_{i\cdot disk\cdot avg}\right| + \sum_{i=1}^{n}\left|PM_{i\cdot bw} - PM_{i\cdot bw\cdot avg}\right|\right) \quad (5)$$

Among them, $PS_{iunbalanced}$ refers to the imbalance of physical machines, n is the number of physical machines, $PM_{i\cdot cpu}$,$PM_{i\cdot mem}$,$PM_{i\cdot disk}$ and $PM_{i\cdot bw}$ are the utilization rates of physical machine i's CPU, memory, disk, and network bandwidth, respectively. $PM_{i\cdot cpu\cdot avg}$, $PM_{i\cdot mem\cdot avg}$, $PM_{i\cdot disk\cdot avg}$ and $PM_{i\cdot bw\cdot avg}$ are the average utilization rates of CPU, memory, disk, and network bandwidth for physical machine i, respectively.

2.2 Global Orchestration Module

Cloud resource global orchestration technology, as an important part of the cloud computing field, can achieve unified management and efficient scheduling of cloud resources, improve the utilization rate and service quality of cloud resources[5]. This chapter aims to propose an implementation scheme of global orchestration of cloud resources to provide technical support for the further development of green computing, as shown in Figure 2. The object of resource orchestration is the user's cloud resources and applications deployed and configured in cloud servers. In order to simplify the management of cloud computing resources, a unified management of a set of cloud resource lifecycles is achieved by defining templates, and the function of automated application deployment is provided. Write a template file based on the template specification defined by the resource orchestration service, defining the required cloud computing resources (such as cloud server instances, cloud hard disk instances, etc.), dependencies between resources, and installation scripts for applications in the template. The orchestration engine of the resource orchestration service will automatically create and configure all resources based on the template content, achieving automated deployment of applications. Resource orchestration also provides a wide variety of service templates, allowing users to quickly build the services they need using the provided service templates. Resource orchestration mainly provides capabilities such as template management, resource stack management, resource type management, application management, and template market management. The implementation plan for global orchestration includes:

(1) Support cross regional template configuration. Specify different resources when creating templates and resources

(2) Multi resource parameter configuration: supports the configuration of required resource parameter information for different regions, including mirrors, VPCs, subnets, etc.

(3) VPC mapping maintenance: Considering that the VPC resources in each region can only be used in the current region, it is necessary for VPC to support cross regional interoperability. When arranging, specify the corresponding VPC information for each region, and maintain relationship mapping for VPCs in different regions.

(4) Resource mounting configuration: Resources between multiple regions cannot be mounted, and resource mounting rules need to be limited.

(5) Service deployment configuration: involves the deployment of services in multiple regional resources, and it is necessary to specify the corresponding regional software package pull warehouse and corresponding regional script configuration.

(6) Monitoring and logging: Multi region resource activation and deployment, monitoring information of resources needs to be queried from monitoring systems and log collection systems in different regions.

(7) Layout engine parsing template: Manage resource information in different regions according to execution rules.

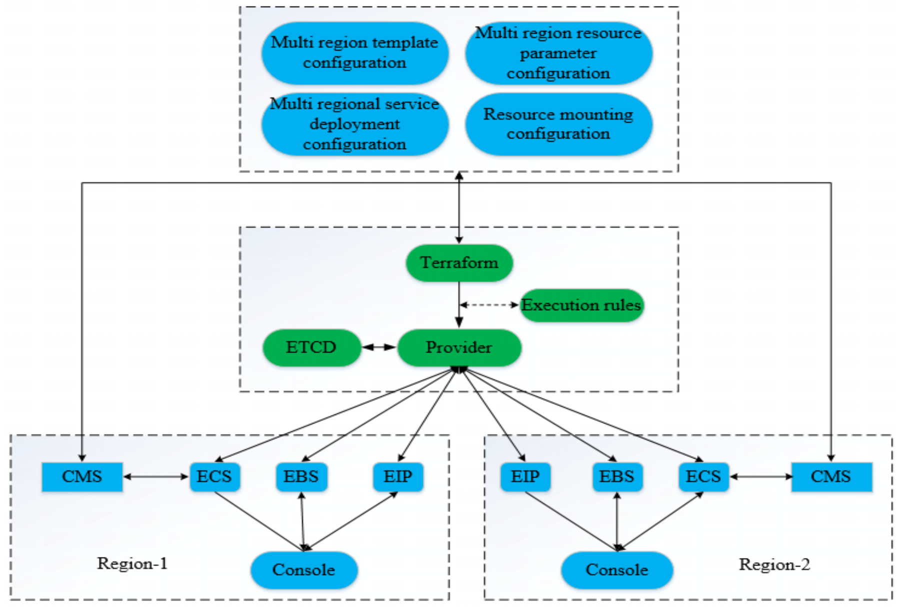
Multi region template configuration
Multi region resource parameter configuration
Multi regional service deployment configuration
Resource mounting configuration
Terraform
Execution rules
ETCD
Provider
CMS
ECS
EBS
EIP
Console
Region-1
EIP
EBS
ECS
CMS
Console
Region-2

Fig. 2. Implementation plan for global orchestration

2.3 Elastic Expansion Module

The elastic scaling module mainly realizes the function of automatically scaling user cloud resources. In terms of architecture, it includes Open API, scaling management, facility management, etc.

In terms of strategy management, elastic scaling supports timed strategy, periodic strategy, and indicator based monitoring strategy. Among them, the scheduled/periodic strategy supports four types of strategy scheduling: scheduled, daily, weekly, and monthly; The monitoring strategy supports triggering types such as maximum, minimum, and average monitoring indicators. In terms of scaling control, after the policy is triggered and executed, scaling control combines the policy content and cloud service instance information to identify scaling activities, generate scaling templates, and hand them over to the resource scheduling engine for cloud resource control. In terms of resource scheduling, the resource scheduling engine receives scalable template data, synchronizes resource status, and dynamically creates and deletes resources such as cloud servers, cloud hard drives, and elastic public IP addresses for cloud users. In terms of scaling group management, a scaling group is a logical grouping of cloud resources with the same scaling configuration, which can perform health checks on cloud resources and perform management operations. Figure 3 shows the scheduling task management process.

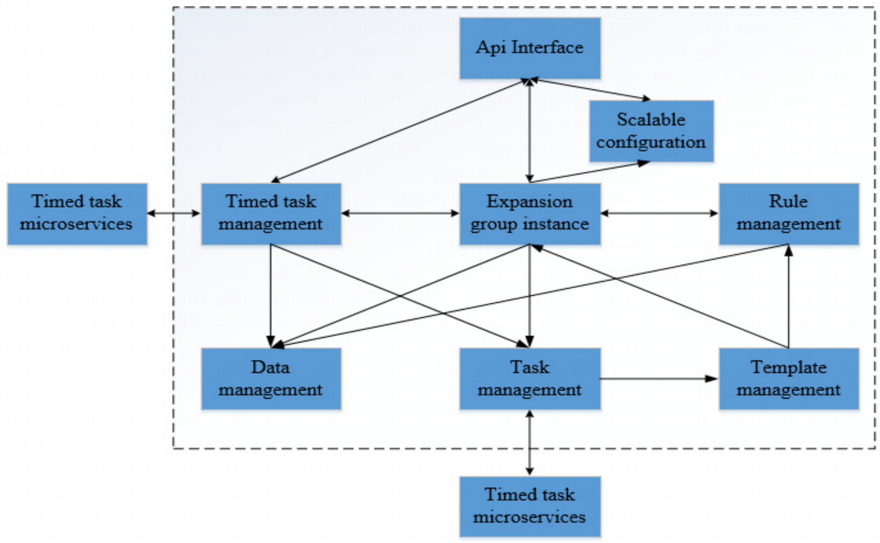

Fig. 3. Scalable timed task management process

The scaling group instance management module is the core functional module responsible for creating, deleting, and querying scaling group instances, using scaling strategies, moving cloud server instances in and out, and triggering activities. Scaling group instance management module, which internally maintains the creation, query, and deletion operations of instances, as well as simultaneously maintains the move in and move out operations of cloud servers; In addition, use the scaling rule management module to manage scaling strategies, and complete initialization and manual triggering operations through the scheduled task management module and activity management module. In order to achieve efficient resource allocation, the scalability module can predict the resource situation of virtual machines in the future, and the predicted results are used to guide resource allocation[6]. The resource prediction model is shown in formula 6.

$$\widehat{F_m}(T) = \widehat{F_m}(t + \alpha) = DP - CPUA(Q_m) \tag{6}$$

Among them, $\widehat{F_m}(T)$ represents the predicted total number of application requests deployed on the virtual machine VM_m^n at time T, while Q_m represents the historical dataset of the actual total number of application requests deployed on the virtual machine VM_m^n in the nth cloud data center.

Generally speaking, the more complete the collection and fusion of resource data, the more accurate the calculation results of elastic expansion conditions will be. Let φ represent the established cloud platform resource data collection coefficient, j represents the resource data fusion processing coefficient per unit time, h represents the maximum collection amount of cloud platform resource data per unit time, and the conditions for elastic expansion are shown in formula 7.

$$\hat{X} = \frac{\left[\beta U - \frac{f(\theta_2 - \theta_1)^2}{\bar{G}}\right]}{2\delta k_0}\sqrt{\varphi h^2 + j^2} \tag{7}$$

Among them, θ_1 and θ_2 represent two different cloud platform sensing demand conditions, where the amount of resource data search is equal to β, f represents the resource data transfer coefficient in the cloud platform environment, $\bar{G}$ represents the average amount of cloud platform resource searches per unit time, δ represents the transmission efficiency of resource data, k_0 represents the initial storage value of data information.

Scaling activity management is responsible for the execution of elastic scaling actions, including five activities: initialization of scaling group instances, maintenance of expected number of instances, maintenance of maximum/minimum number of instances, manual triggering of rule activities, and timed task triggering of rule activities. The activity management module receives trigger inputs from the scheduled task management and scaling group instance management modules, uses the template management module to produce and execute templates, calls the interface provided by the facility management service, and completes the execution of activity content[7]. Each activity is completed by an independent thread. The state switching of the five activities is shown in Figure 4.

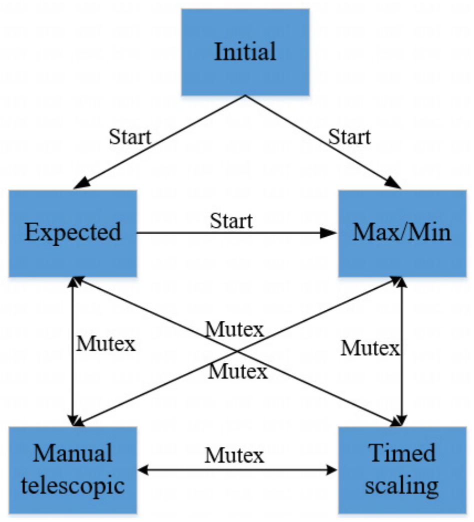

Fig. 4. Scaling activity state switching

The scaling activity sets the trigger execution time for different priority activities based on the type of strategy, achieving time priority. The execution logic is as follows[8]:

(1) At the same time point, multiple strategies are triggered simultaneously (alarm strategy, timing strategy, and periodic strategy), with the priority order being alarm strategy>timing strategy>periodic strategy. The scaling group executes according to the priority of the strategy, and can only execute the same type of strategy at the same time point.

(2) During the cooling time, the scaling group will reject scaling activities triggered by alarm policies. Scaling activities triggered by other types of scaling policies (such as timed and periodic policies) are not restricted, but the cooling time will be recalculated in seconds.

(3) There are multiple alarm policies set, and if multiple policies are met at the same time, the multiple policies for the alarm will be executed simultaneously.

2.4 Implementation of Multi state Ant Colony Algorithm Improved by Consistent Hashing Algorithm

Traditional cloud resource scheduling algorithms often struggle to achieve ideal results when facing the dynamically changing cloud environment in the green computing cloud platform model mentioned earlier. Therefore, this article proposes an improved polymorphic ant colony cloud resource scheduling algorithm based on consensus hashing algorithm, aiming to improve the efficiency and performance of cloud resource scheduling.

The consistency hash algorithm is mainly used to establish a dynamic hash ring based on scalable orchestration, identify resource node mapping, and load balancing[9]. The specific implementation is as follows:

(1) Hash the resource nodes in the green computing cloud platform model that are in the on state to obtain the hash value of each node.

(2) Map nodes to a circular hash space based on hash values, forming a dynamic hash ring. The so-called dynamic hash ring refers to when a node is in a shutdown state, it will be removed from the hash ring. When the node is in a power on state, it will retry step (1) and add the node to the hash ring.

(3) Using the cluster monitoring module in the green computing cloud platform model to dynamically feedback the load rate and ranking of all hash nodes, in order to determine whether load adjustment is needed for hash nodes. Collect the CPU, memory, disk, and network usage of each server node at a fixed cycle, set CPU, memory, disk, and network weight coefficients, and calculate the dynamic load. The calculation method is as shown in formula 8.

$$PD_i = w_c c_i U_{cpu}(i) + w_m m_i U_{mem}(i) + w_t t_i U_{thou}(i) + w_n n_i U_{net}(i) \tag{8}$$

Among them, c_i,m_i,t_i and n_i are the normalized number of CPU cores, memory capacity, disk, and network bandwidth of the node, $U_{cpu}(i)$,$U_{mem}(i)$,$U_{thou}(i)$ and $U_{net}(i)$ are the CPU utilization, memory utilization, disk utilization, and network bandwidth utilization of the node, respectively. w_c,w_m,w_t and w_n are the weight coefficients of each load evaluation indicator information, and$w_c + w_m + w_t + w_n = 1$.

(4) When a node is overloaded, according to the characteristics of the consistency hash algorithm, some tasks are migrated to adjacent nodes to achieve load balancing.

Based on the above algorithm, an improved ant colony algorithm is used to establish a hash ring for dynamic nodes of the green computing cloud platform before scaling and scheduling. This algorithm is used to schedule m virtual machines to n hash nodes, improve scheduling efficiency and system load balancing. Combined with virtual machine hot change technology, the energy consumption of the cloud data center is minimized during normal operation[10]. The improved ant colony algorithm in this article requires the following steps to solve the problem:

(1) Initialize the number of iterations to $t = 0$, and set the original pheromone of the node to $\tau_i = \varphi \cdot PS_{iunbalanced} + \omega \cdot PD_i$.

(2) Each ant in the AntList represents a virtual machine creation request, and m ants are arranged based on the size of resource requirements.

(3) Formula 9 represents the host selection mechanism of ants. In the formula, P_i represents the probability of the i-th hash node being selected by ants, τ_i represents the pheromone of the i-th host, and μ_i represents the cost-effectiveness of the i-th host. α and β represent the importance of τ_i and μ_i, and $allowed_k$ represents the set of hash nodes that can be matched, while $tabu_k$ represents the set of hash nodes that cannot be matched or have already been matched. When the hash node i is selected by the ant, it is placed in $tabu_k$, and the virtual machine creation request dispatches resources to the hash node, and these virtual machines are deleted from the AntList.

$$P_i = \begin{cases} \frac{[\tau_i(t)]^{\alpha} \cdot [\mu_i]^{\beta}}{\sum [\tau_i(t)]^{\alpha} \cdot [\mu_i]^{\beta}} & i,k \in allowed_k \\ 0 & otherwise \end{cases} \tag{9}$$

(4) If AntList is not empty, return execution. Then, if all ants are empty, execute (2).

(5) $t = t + 1$

(6) Hash nodes use equation 10 to calculate the volatilization of pheromones, where ρ is the volatilization rate.

$$\tau_i(t) = (1-\rho) \cdot \tau_i(t-1) \tag{10}$$

(7) The ant colony releases pheromones according to formulas 11 and 12, where ${\Delta\tau_i}^k$ is the pheromone released by the k-th ant on i between $t-1$ and t. If the k -th ant passes through host i from $t-1$ to t, then ${\Delta\tau_i}^k = Q/Pk$, where Q is a constant and P is the total power of k passing through the hash node; Otherwise ${\Delta\tau_i}^k = 0$.

$$\tau_i(t) = \tau_i(t) + \Delta\tau_i \tag{11}$$

$$\Delta\tau_i = \sum_{k=1}^{M} {\Delta\tau_i}^k \tag{12}$$

(8) If the total power of this optimal scheduling is smaller than the optimal scheduling in the previous path process, then this scheduling is selected as the optimal scheduling. If the maximum number of times is reached or the optimal scheduling power no longer changes, the ant colony algorithm ends. Otherwise, repeat the above steps.

3 Test and Verification

Deploy the services of two cloud platforms in a containerized manner according to the previous green cloud platform model, including orchestration services, scaling services, cluster monitoring services, virtualization services, etc. Among them, the cloud platform obtained from the cluster monitoring service includes 5 computationally intensive, 10 memory optimized, 4 I/O read-write, and 3 general-purpose computing. And the CPU over ratio of the cloud platform is 6, the memory over ratio is 1, and the overload rate is configured at 50%. The actual number of servers occupied by the cloud platform and the number of servers that can be shut down are shown in Table 1.

Table 1. Cloud Platform Resource Occupation and Shutdown Information

	Actual occupancy quantity	Number of shutdowns
Computationally intensive	2	2
Memory optimized	3	5
I/O read-write type	1.8	1
Universal computing type	1.2	1

The number of tasks to be assigned has gradually increased from 50 to 300, and other parameters such as network bandwidth and network delay are randomly generated by the platform. Combined with the parameter selection in Section 2.4 of this article, the various parameters are set as shown in Table 2.

Table 2. Cloud Platform Resource Occupation and Shutdown Information

Parameter	Value
α	0.8
β	1.0
ρ	0.6
φ	0.5
ω	0.5

The task completion time and hash node imbalance of the original ant colony algorithm and the improved ant colony algorithm in this article are shown in Figures 5 and 6.

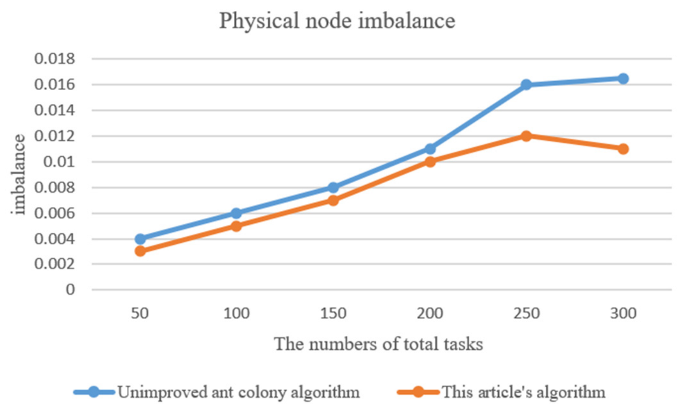

Fig. 5. Physical node imbalance

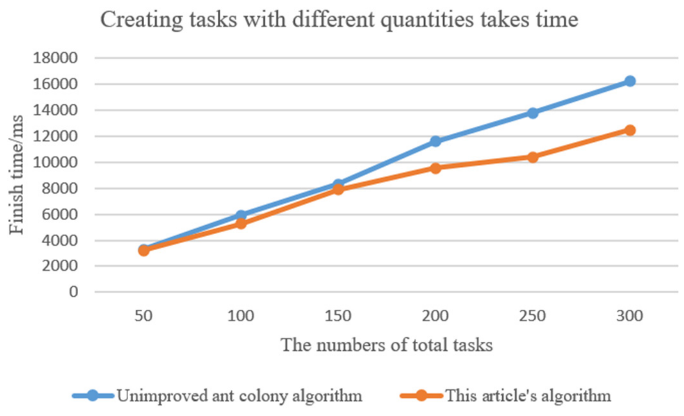

Fig. 6. Creating tasks with different quantities takes time

In addition, Modify the number of replicas for the cluster monitoring service to 0, then set the value corresponding to energy_ efficiency switch in the configmap on which it depends to True. Finally, through the deploy of the edit service, set the number of replicas to 1, and the service will restart. Observing the service operation log, you will see that the cluster monitoring service issues shutdown operations to the corresponding server according to the calculated number of shutdowns in Tables 2.

According to the threshold values of user application configuration workload, network traffic, and disk IO status, the cloud platform will run continuously for one month and count the number of times the shutdown machine has been restarted. Among them, the cloud platform restarted one computationally intensive server and shut down one memory optimized server on the 10th; On the 15th, one memory optimized and one general-purpose computing were restarted; On the 23rd, one memory optimized model was shut down. The energy consumption recorded by the cloud platform has decreased by 23.4% compared to the previous month.

In summary, a green computing cloud platform based on orchestration scaling can not only effectively control the energy consumption of data centers, but also greatly improve resource balance and scheduling efficiency. Prove the feasibility of this plan.

4 Conclusions

In summary, this article focuses on the solution of a green computing cloud platform based on scalable orchestration, including the green computing cloud platform model, global resource orchestration, and elastic scaling. Based on the energy consumption analysis of cloud data centers, resource control, prediction, and elastic scaling models are proposed to achieve dynamic switch control of physical machines in cloud data centers, improve the effective resource utilization and energy efficiency management of cloud data centers. Not only has energy efficiency management and resource scheduling achieved the goal of energy conservation and emission reduction in data centers, but also the goal of controlling resources according to the actual business needs of users. In the process of promoting applications, problems such as random user business loads have also been solved, and the parallel quality and efficiency of data center energy efficiency and user applications have been ensured. However, in terms of reducing energy efficiency costs in data centers, the impact of new energy sources such as wind and solar power on energy consumption costs has not been taken into account, and the differences in electricity prices among different regions in China have not been taken into account.

References

1. Y, Wang. Research on Resource Prediction and Configuration Methods for Green Cloud Computing in Data Centers[D]. Hunan University of Science and Technology, 2021. DOI:10.27738/d.cnki.ghnkd.2021.000474.
2. Z, Zhou. On the Energy Efficiency of Green Geo-distributed Datacenters [D]. Huazhong University of Science and Technology, 2017.
3. Ismaeel S, Miri A. Using ELM techniques to predict data centre VM requests[C]. IEEE International Conference on Cyber Security and Cloud Computing. IEEE, 2015: 80-86.
4. Kholidy H A. An Intelligent Swarm Based Prediction Approach For Predicting Cloud Computing User Resource Needs[J]. Computer Communications, 2020(151): 133-144.

5. JX, Wan.YY, XU. Design and Implementation of Service orchestration Technology for Enterprise Cloud Computing Platform [J]. Journal of Shanghai Institute of Ship Transportation Science, 2021, 44 (02): 52-57+65.
6. JW, He. LC, Qi. A Resource Elastic Scaling Method for Cloud Platforms Based on Hybrid Models [J]. Electronic Design Engineering, 2022, 30 (24): 90-94. DOI:10.14022/j.issn1674-6236.2022.24.019.
7. LJ, Luo. Research on container elastic scaling technology based on load prediction [D]. Wuhan Textile University, 2021. DOI:10.27698/d.cnki.gwhxj.2021.000102.
8. GQ, Xu, Research on Adaptive Management and Collaborative Scheduling of Cloud Resources for Scalable Cloud Services. Tianjin Key Laboratory of Cognitive Computing and Applications, Tianjin University, 2018-07-09.
9. YY, Chen. D Liu. etc. Consistent Hash Database Load Balancing with Endogenous Dynamic Feedback [J]. Modern Electronics Technique, 2021, 44 (23): 111-116. DOI:10.16652/j.issn.1004-373x.2021.23.022.
10. ZD, Sun. Q, Jiao.etc. Research on Task Scheduling Strategy for Power Cloud Data Center Based on Improved Ant Colony Algorithm [J]. Power System Protection and Control, 2022, 50 (02): 95-101. DOI:10.19783/j.cnki.pspc.210466.