Simulation of Multi-level Cloud Computing Task Balanced Allocation Based on Genetic Ant Colony

Gong Fanghai

School of Information Engineering, Guangzhou Huashang Vocational College, Guangzhou, 511300, Guangdong, China

Keywords: Genetic algorithm; Ant colony optimization; Cloud computing; Balance allocation

Abstract: The scale of cloud computing system is getting bigger and bigger, the topological structure is getting more and more complex, and the heterogeneity of resources makes how to effectively schedule cloud computing tasks become a very important research topic in the field of cloud computing. In this paper, the simulation research of multi-level cloud computing task balance allocation based on genetic ant colony is carried out. In this paper, users' requirements for QoS(Quality of Service) are divided into performance requirements and cost requirements. Performance requirements can shorten the time span by improving the computing performance, transmission performance and storage performance of physical resources, while cost requirements can reduce the computing cost by integrating performance requirements and scheduling costs. A GA _ ACO (Genetic Algorithm-Ant Colony Optimization) algorithm is proposed, which encodes these parameters and finds the optimal combination in the process of evolution. It fully combines the feedback mechanism of ACO, the global search ability of GA and the characteristics of fast convergence. The simulation results show that by combining the respective advantages of GA and ACO, the convergence speed is increased by 4. 07%, and the resource load rate of the algorithm decreases by 30. 28% on average compared with ACO. The load balance of the whole system is gradually improved.

1. Introduction

With the vigorous development of cloud computing technology, the research and implementation of various related technologies of cloud computing are also showing a trend of blooming flowers. Task scheduling refers to the assignment of tasks submitted by users to available resources for execution according to specific scheduling rules. Efficient task scheduling strategy is crucial to system operating cost, QoS(Quality of Service) and resource utilization [1-3]. The scale of cloud computing system is getting bigger and bigger, the topological structure is getting more and more complex, and the heterogeneity of resources makes how to effectively schedule cloud computing tasks become a very important research topic in the field of cloud computing.

Moori et al. proposed a multi-level cloud computing task balancing method based on supporting multi-link access [4]. Liu Guoqing et al. proposed a multi-level cloud computing task balanced allocation method based on the maximum comprehensive utilization rate, which fully considered the weight ratio of electronic information resources under the cloud computing platform, and adopted the maximum resource utilization rate algorithm based on weight calculation to configure virtual machines under the cloud computing platform, so as to achieve the multi-level cloud computing task balanced allocation [5]. Zhai Ling et al. used bee colony algorithm to improve the task allocation problem, which not only achieved the load balance but also made the task set complete in the shortest time under multi-objective constraints [6]. However, the tradeoff between load balancing and the shortest completion time of task set in the above literature is insufficient. In this paper, a multi-level cloud computing task balance allocation optimization method based on GA _ ACO (Genetic Algorithm-Ant Colony Optimization) is proposed.

DOI: 10.25236/icmmct.2023.037

2. Research method

2.1. Multi-level cloud computing task scheduling model

Because the computing tasks performed by the cloud computing system are relatively simple, and the tasks are basically used to deal with similar business requirements, the Master node in the system, that is, the Master node, uses the queue mode. When a Slave node applies to the master node for acquiring tasks, the master node will sort all tasks according to priority, then sort them according to arrival time, and then select a task with the highest priority and the earliest arrival time to assign to the Slave node. If there are some unallocated resources in the system, the unallocated resources will be allocated to the task pool that needs to use resources but the pre-allocated resources are not enough.

The goal of balanced assignment of multi-level cloud computing tasks is to quickly and reasonably assign the tasks submitted by users to the cluster resource nodes to run, to achieve the best scheduling of the cluster, and to maximize the utilization of the resources and throughput of the cluster [7-8]. Load balancing is a key issue to be considered when scheduling jobs in high performance computing clusters.

Load balancing means that the tasks submitted by users can be evenly distributed, and each node reasonably distributes the jobs submitted by users, so as to maximize the utilization of the resources of the scheduling system. Therefore, in order to ensure users' high-quality service and efficient task processing, high-performance computing task scheduling needs to have the characteristics of dynamic adaptability, dynamic scalability, large scale and high fault-tolerance mechanism.

Traditional task scheduling mainly considers whether the task completion time is shorter, whether the load of the processor is balanced, and the task processing mode that does not support the interaction between tasks. The task scheduling object of cloud computing is the virtual computing resources encapsulated by various physical infrastructure devices such as servers, storage and networks, which are provided to users. These service resources have the same or different attributes, forming heterogeneous or isomorphic platforms.

In this paper, users' QoS requirements are divided into performance requirements and cost requirements. Performance requirements can shorten the time span by improving the computing performance, transmission performance and storage performance of physical resources, while cost requirements can reduce the computing cost by integrating performance requirements and scheduling costs [9].

If m resources are denoted as $R = \{r_1, r_2, \cdots, r_m\}$ and n tasks are denoted as $U = \{u_1, u_2, \cdots, u_n\}$ in the cloud computing environment, the cloud computing system can be described as:

$$Cloud = (R, T) \tag{1}$$

$E_{ct}(i, j)$ is the predicted execution time of task $i(i \in \{1,2,\cdots,n\})$ in resource $j(i \in \{1,2,\cdots,m\})$, and the calculation formula is as follows:

$$E_{ct}(i,j) = \frac{J_{Lengthi}}{P_{Compj}} \tag{2}$$

$J_{Lengthi}$ is the computational length of task i, and P_{Compj} is the computational performance of resource j.

In order to achieve the ultimate goal of task resource scheduling, the optimal time span is represented by $T(X)$, that is, $T(X)$ is minimized. Let L represent the set of load balancing indexes, and set $L = \{L_1, \cdots, L_n\}$, then $L_j(X)$ represents the load index of cluster node j under scheduling policy X. $L_j(X)$ definition:

$$L_j(X) = \frac{1}{T(X)} \sum_{i=1}^{m} M_{ij} * W_{ij}, \qquad j \in \{1,2,\cdots,m\} \tag{3}$$

$M_{ij} \neq -1$, can get $0 < L_j(X) \leq 1$.

2.2. Balanced distribution design based on genetic ant colony

Task scheduling in cloud computing is to schedule the n tasks submitted by users to be executed to the appropriate m computing resource nodes for processing under certain constraints, so as to complete the user's task requests. How to reduce the cost of computing center services while ensuring the QoS requirements of users, that is, to shorten the task execution time and reduce the energy consumption of computing nodes at the same time, so as to achieve the double-objective optimization of task execution time and energy consumption [10].

The QoS requirement of users depends on the time of task completion, while the service cost of service providers can be evaluated according to the actual task scheduling time. This paper will comprehensively evaluate the energy consumption cost by using the computing speed, transmission performance and storage capacity of resource nodes.

The first step of GA(genetic algorithm) is to choose an appropriate coding method according to the characteristics of the problem. After coding, each individual chromosome represents a temporary solution in the solution space, and all individuals' chromosomes gather together to form a solution space. A good coding method can express all the information needed by the algorithm operation and simplify the decoding process, so a good coding method is very important. It not only reflects the essence of the problem, but also simplifies the genetic process and improves the execution of the algorithm.

ACO(ant colony optimization) is a stochastic evolutionary algorithm, and its main application fields are TSP traveling salesman problem, quadratic assignment problem, etc. The research results show that this algorithm has many excellent characteristics, and it can obtain very good calculation results in related practical application fields. The more pheromones on a route, the greater the probability that the route will be selected, and eventually ants will completely select the route. This phenomenon is called the positive feedback effect of pheromones.

After a large number of experimental observations, the evolution speed of the algorithm will obviously decrease after GA runs to a certain stage, and the ant search path is in a blind state, with low efficiency. When the pheromone content accumulates to a certain extent, the ant's actions begin to show certain rules, and the evolution speed of the algorithm also begins to accelerate.

First, set a minimum and maximum evolutionary algebra, and set them as $gene_{\min}, gene_{\max}$, so as to prevent the algorithm from running without convergence. Then calculate the evolution rate of GA, which is calculated by the following formula:

$$R_i = \frac{F_i - F_{i-1}}{F_{i-1}}, \qquad i = 1,2,\cdots,m \tag{4}$$

m represents the population size, F_i represents the fitness value of the i th individual in the population, and the average of the evolution rates of all individuals in each generation is the evolution rate of that generation.

In the evaluation of chromosome population, we use the small evolution rate of five successive generations to evaluate. The first 10% of the initial population is taken as the optimal solution, and it is converted into the initial pheromone of ant colony. The specific initialization rules are as follows:

$$T_i^G(0) = \rho S_n \tag{5}$$

ρ represents a set constant; Represents the optimal solution of S_n GA. Through the results of

GA, we can get the distribution of pheromones.

The shorter the time required to complete the task, the larger the individual's time fitness value, and the more it can represent the optimal solution of the problem. Similarly, the smaller the energy consumption of computing nodes, the larger the fitness value of individuals. In order to give consideration to both, this paper combines the time fitness function with the energy consumption fitness function, and defines a comprehensive fitness function $F(I)$ to ensure QoS, as follows:

$$F(I) = aFit_{time}(I) + bFit_{\cos t}(I) \tag{6}$$

Coefficient a,b represents the weight of time and energy consumption, respectively, and $a+b=1, a,b \in [0,1]$.

The specific solution flow chart of GA_ACO algorithm is shown in Figure 1:

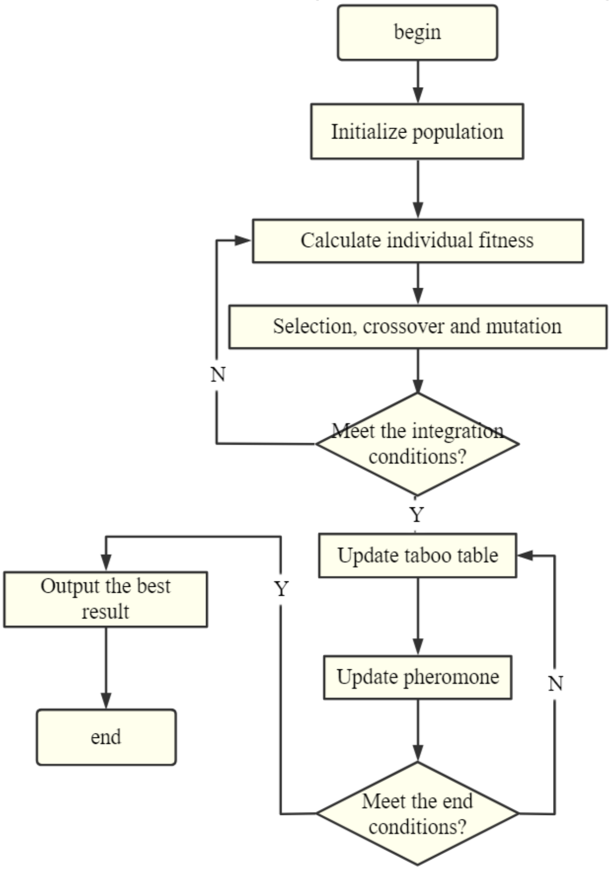

Figure 1 GA_ACO algorithm flow

3. Simulation analysis

The algorithm idea of this paper is to modify CloudSim's own task scheduling algorithm (that is, to modify the DataCenterBroker class), use the hybrid algorithm of GA and ACO to schedule tasks, and extend Cloudlet class to introduce multi-dimensional QOS, so as to meet users' various QOS requirements. Establish and set virtual machine ID and performance parameters such as memory, CPU and storage. Establish the tasks in the cloud computing system and set the task ID, the end conditions of the tasks, the QOS objectives, etc.

In the simulation environment, the number of nodes is 50, each node contains 2 CPUs, the memory size of the node is 128G, and the bandwidth is 56GB. Each virtual machine resource includes CPU number, memory size, bandwidth, instruction processing speed and other indicators. Create CloudSim object to add cloud computing tasks.

In this paper, experiments are conducted according to the above parameters, and task completion time, CPU utilization and load balance are taken as the performance evaluation indexes of the algorithm. The comparative experiments of GA, ACO and GA_ACO in this paper are carried out. Task completion time refers to the total time spent from the first task being assigned to the computing node to the last task being completed. There are 20 server nodes and 20-200 tasks in the experiment. The comparison results of task completion time are shown in Figure 2.

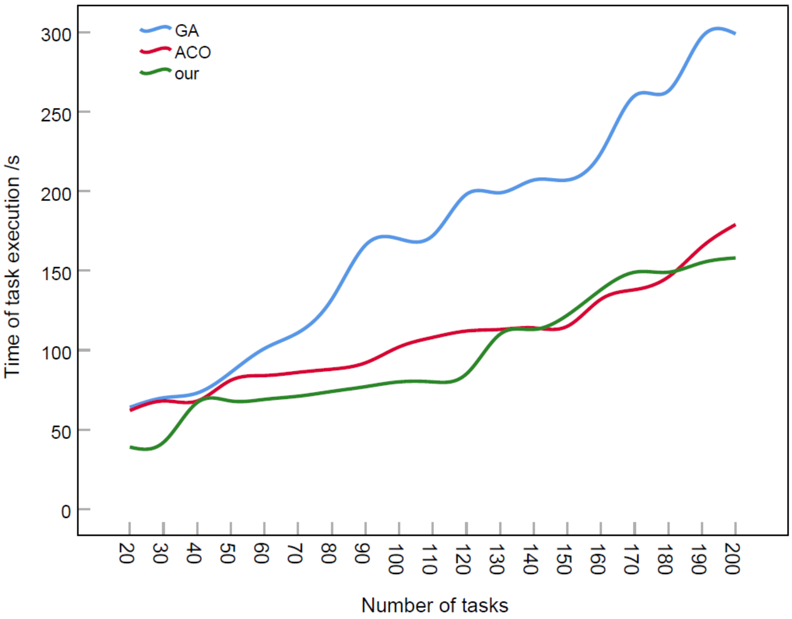

Figure 2 Comparison of task completion time

It can be seen that the task execution time of ACO algorithm and GA_ACO algorithm is much faster than that of GA algorithm. The reason is that the server load factor is added to these two algorithms, and both of them can choose the server node with better load to perform the task. Moreover, due to the poor solving ability of GA algorithm in the later stage, the more tasks, the greater the task completion time. Therefore, the convergence speed of the fused GA_ACO algorithm is stronger than that of ACO, so the task execution time is also the shortest.

In order to more accurately describe the system load balancing performance of each scheduling algorithm in the solution process, the standard deviation of resource node load is used for evaluation. Based on different task scheduling algorithms, compare the system load balance after different tasks are scheduled, as shown in Figure 3 below.

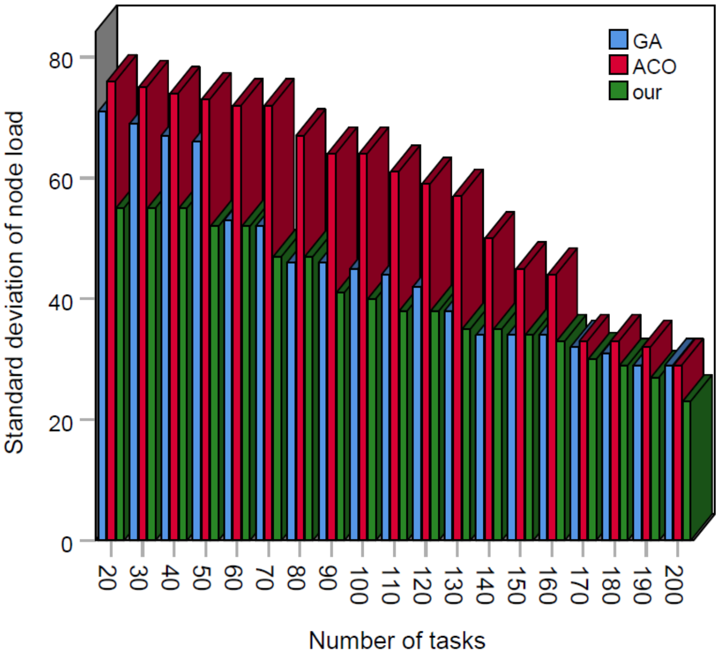

Figure 3 Comparison of system load balancing with different task numbers

Among the three scheduling algorithms, the system load balance level based on GA_ACO task scheduling strategy is always optimal, which is determined by the solution accuracy of the scheduling algorithm itself based on the double-objective optimization model of time and energy consumption. By combining the advantages of GA and ACO, the convergence speed is increased by 4. 07% by avoiding the local optimal solution. With the increase of task scale, the resource load rate of the algorithm is reduced by 30. 28% on average compared with ACO.

The load standard deviation of each resource node in the system corresponding to all scheduling algorithms generally shows a downward trend with the increase of the number of tasks, which indicates that the load balance of the system is constantly improving. This is because when the number of scheduled tasks is small, the task allocation is random, which affects the performance of system load balance. However, as the number of tasks increases, resource allocation tends to be rationalized gradually, and the load balance of the whole system is gradually improved.

4. Conclusions

Task scheduling refers to assigning tasks submitted by users to available resources for execution according to specific scheduling rules. Efficient task scheduling strategy is crucial to system operating cost, QoS and resource utilization. In this paper, the simulation research of multi-level cloud computing task balance allocation based on genetic ant colony is carried out. In this paper, the calculation speed, transmission performance and storage capacity of resource nodes will be used to comprehensively evaluate the energy consumption cost. Genetic operator is used to filter the optimal solution globally and ant colony operator is used to improve the accuracy of the solution. The simulation results show that by combining the respective advantages of GA and ACO, the convergence speed is increased by 4. 07%, and the resource load rate of the algorithm decreases by 30. 28% on average compared with ACO.

Acknowledgements

2022KTSCX380 Research and application of college students' innovation and entrepreneurship cloud service platform based on big data technology

References

[1] Li Bo, Gao Peixin, Zhang Ming,&Fan Panlong. (2018). Cross-cloud task allocation method of unmanned combat system based on network load balancing. Command and Control and Simulation, 40(5), 7.

[2] Wang Dongliang, Yi Junyan, Li Shihui, & Wang Hongxin. (2017). Cloud computing task scheduling integrating load balancing and bat algorithm. Information network security, 2017(1), 6.

[3] Zeng Zhaomin. (2017). Research on Cloud Computing Load Balancing Based on Improved Genetic Algorithm. Electronic Design Engineering, 2017(4), 4.

[4] Moori, A., Barekatain, B., & Akbari, M. (2022). Latoc: an enhanced load balancing algorithm based on hybrid ahp-topsis and opso algorithms in cloud computing. The Journal of Supercomputing, 78(4), 4882-4910.

[5] Liu Guoqing,&Shi Xiaochun. (2018). Improved fuzzy clustering time mode cloud computing task scheduling scheme. Control Engineering, 25(11), 5.

[6] Zhai Ling, Shen Si, & Cheng Shixing. (2019). Optimization simulation of balanced distribution of electronic information resources under cloud computing platform. Computer Simulation, 36(7), 397-400,440.

[7] Zhang, G. (2017). Simulation research on the cloud computing data scheduling and distribution technology based on balanced energy consumption modelling. Boletin Tecnico/Technical Bulletin,

55(19), 625-633.

[8] Gan, C., Feng, Q., Zhang, X., Zhang, Z., & Zhu, Q. (2020). Dynamical propagation model of malware for cloud computing security. IEEE Access, 2020(99), 1-1.

[9] Gao, W. (2021). Intelligent prediction algorithm of cross-border e-commerce logistics cost based on cloud computing. Scientific programming, 2021(10), 2021.

[10] Shah-Mansouri, H., Wong, V., & Schober, R. (2017). Joint optimal pricing and task scheduling in mobile cloud computing systems. IEEE Transactions on Wireless Communications, 2017(8), 1-1.