Optimization of Rational Scheduling Method for Cloud Computing Resources Under Abnormal Network

Lunqiang Ye(✉)

Southwest Minzu University, Chengdu, China
feiyun001b@163.com

Abstract. When the traditional heuristic algorithm was used to schedule the cloud computing resources under the abnormal network, there was a problem that the scheduling speed was slow and the effect was poor. Aiming at the above problems, combined with the characteristics of cloud computing and the actual needs of cloud computing resource allocation, based on the advantages of genetic algorithm and ant colony algorithm, a hybrid optimal cloud computing resource scheduling algorithm was designed. The improved algorithm combines the advantages of genetic algorithm and ant colony algorithm, and the genetic algorithm can effectively improve the search efficiency; The ant colony algorithm was used in the later stage of the algorithm to improve the accuracy of the optimal solution and to complete the reasonable scheduling of cloud computing resources under the abnormal network. The results show that the hybrid algorithm was faster than the single genetic algorithm and ant colony algorithm. It only took 10 s, the resource load was more balanced, and the scheduling effect was better.

Keywords: Abnormal network · Cloud computing · Resource scheduling · Genetic algorithm · Ant colony algorithm

1 Introduction

In the cloud computing environment, particle swarm optimization, genetic algorithm and ant colony algorithm are applied to the current resource scheduling [1]. They have adaptive search capabilities, they first establish a mathematical model for specific application problems, and then solve it [2]. Resource scheduling in the cloud computing environment is an NP-Hard problem. When solving a single algorithm, there are some limitations. For example, the population diversity of the late stage of genetic algorithm is seriously degraded, and the search ability of the bureau is poor. The ant colony algorithm has no initial hormone mechanism, and the search efficiency is low at the beginning stage.

In order to overcome the limitations of a single algorithm, it is better to balance the tasks between resources. The genetic algorithm and the ant colony algorithm are combined, and the genetic algorithm can be used to obtain the feasible solution of resource scheduling in the cloud computing environment. Then, using the characteristics of late search ability of ant colony algorithm, the resource scheduling scheme is

G. Gui and L. Yun (Eds.): ADHIP 2019, LNICST 301, pp. 207–215, 2019.
https://doi.org/10.1007/978-3-030-36402-1_22

searched twice to find the best scheduling scheme. Finally, the simulation experiment is carried out on Cloud Sin platform to test its performance. The results show that the improved hybrid algorithm is more efficient. It only takes 10 s, the resource load is more balanced, and the scheduling effect is better.

2 Hybrid Optimization Algorithm

In view of the advantages and disadvantages of traditional allocation algorithms, this paper designs a hybrid optimization algorithm. It is a new heuristic grid task scheduling algorithm with both time and efficiency [3]. The overall framework of its algorithm design is shown in Fig. 1.

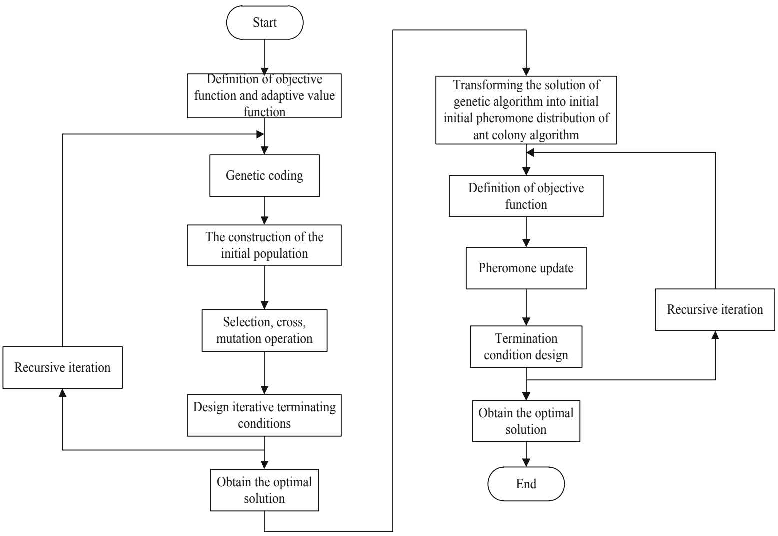

Fig. 1. Hybrid algorithm overall design framework

2.1 Dynamic Fusion Critical Point Determination of Two Algorithms

The key step of the new algorithm is the best combination of the two algorithms. The dynamic fusion strategy can determine the optimal fusion time between the genetic algorithm and the ant colony algorithm.

(1) Set the minimum number of genetic iterations and set the maximum number of iterations.
(2) In the iteration process, we need to calculate the evolutionary rate of descendants and set the minimum evolution rate.

(3) Within the set number of iterations, if the continuous N generation, the evolution rate of the offspring population is less than the minimum evolution rate, indicating that the genetic algorithm optimization speed is low, so the genetic algorithm process can be terminated and enter the ant colony algorithm [4].

2.2 Objective Function

According to the characteristics of cloud computing resource scheduling, the following constraints are added, so that the algorithm can be better applied to practical problems. The specific constraints are: The processing time of the task is about ProcessTime(e), network bandwidth constraint Band math(e), and network delay constraint NetDelay(e) [5]. The objective function is defined as follows, depending on the situation after the resource is subject to the above constraints:

$$\mathrm{Re}s(e) = \frac{\left(A_{\mathrm{ProcessTime(e)}} + B_{\mathrm{NetDelay(e)}}\right)}{C_{\mathrm{Bandmath(e)}}} \tag{1}$$

$$s.t.\left\{ \begin{array}{l} \mathrm{ProcessTime(e)} < \mathrm{MaxT} \\ \mathrm{NetDelay(e)} < \mathrm{MaxN} \\ \mathrm{Band\ math(e)} < \mathrm{MinB} \end{array} \right\} \tag{2}$$

The maxT, minB, and maxN in the formula (2) represent the upper limit of the task processing time, the minimum limit of the bandwidth, and the upper limit of the network delay, respectively, and the weight values of the three constraints are represented by A, B, and C, respectively.

2.3 Fitness Function

The fitness index is used to evaluate the goodness of the individual in the population. The greater the fitness value, the better the individual's performance. By defining the fitness function as follows, the principles of the genetic algorithm can be well followed.

2.4 Genetic Coding

To perform a global search for resource scheduling problems based on genetic algorithms, it is first necessary to encode the scheduling scheme of the problem into chromosomes. The binary method is generally used for chromosome coding, that is, one chromosome corresponds to one binary string, and corresponds to a resource scheduling scheme [6].

2.5 Initial Population

Let the population size be M, the number of sub-tasks be n, the number of processing units be m, and the maximum depth of the task be h. The algorithm flow for generating the initial population is shown in Fig. 2.

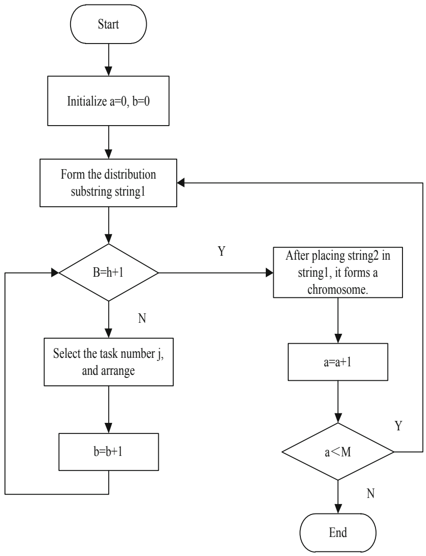

Fig. 2. Algorithm flow for generating an initial population

Assuming there are two resources, the two solutions that can be generated using this algorithm are:

Solution 1: (212212 212221)

Solution 2: (111212 122121)

Their corresponding resource rational scheduling schemes are:

Distribution plan 1: Y1: X2→X4→X5

Y2: X1→X3→X6

Distribution plan 2: Y1: X3→X1→X6

Y2: X5→X4→X2

2.6 Select Copy

After calculating the appropriate values of all chromosomes, it is necessary to select the chromosomes that satisfy the conditions with specific constraints, and set the appropriate values by roulette, to randomly select chromosomes with better performance from all chromosomes, follow the steps below:

Step1: The chromosomes are selected from the initial population according to the probability Pe, and then the appropriate values of the selected chromosomes are calculated, and the sum Mi of the moderate values of all the chromosomes is calculated.

Step2: A random number M0 is selected from [0, Mi] using a random number algorithm.
Step3: The appropriate value of the chromosome is added according to the numbering order of the chromosomes, and when the accumulated value is $\geq$Mo, the last accumulated chromosome is copied.
Step4: Step 2, Step 3 are repeated, and when the number of new chromosomes reaches the original number of chromosomes, the selection and copying operations are terminated.

2.7 Crossover Operator

First, two chromosomes are randomly selected, and then the position of the intersection is randomly determined, and the second half (including the intersection) is exchanged from the intersection to generate two new sub-individuals [7].

If the new individual does not satisfy the constraint, re-select the intersection. If the newly generated individuals satisfy the constraint relationship, calculate their fitness values separately. The number of individuals in the offspring is twice the number of individuals in the father. The elite retention strategy is used to rank the individuals in descending order according to the fitness value, and the first half of the individuals are retained to generate new populations.

2.8 Mutation Operator

The mutation operator here essentially transfers a resource to another task for execution. In order to prevent a certain resource from being migrated and the execution time increases and the population is degraded, it is stipulated that the resources occupied by the post-migration task are not randomly generated. Instead, in the collection of resources other than the resources currently occupied by the task, the resource that minimizes the execution time of the subtask is selected and migrated to the task for execution [8].

2.9 Iterative Termination Condition

The specific values of the parameters can be determined by combining empirical parameters related to the genetic algorithm.

2.10 Ant Colony Algorithm for Obtaining Optimal Solution

The chromosome population is evaluated according to the objective function, and when the evolution rate of the chromosome population is small, the final genetic algorithm is terminated; Since the grid scheduling has the shortest task scheduling length as the solution target, when used for the grid task scheduling problem, the basic ant colony algorithm needs to be modified appropriately. The specific changes are: in the traveling salesman problem, the basic ant colony algorithm uses pheromones to describe the path distance between the various attractions; In the problem of rational resource scheduling, pheromone is used to describe the real-time computing power of each computing node. The ant colony algorithm obtains the optimal solution process as follows:

Initial pheromone conversion:
In order to ensure the high efficiency of the algorithm, the optimal solution obtained by the genetic algorithm needs to be transformed and used as the initial pheromone of the ant colony algorithm. The calculation formula is as follows:

$$A_i(t) = A_0 + A_i^{GA}(t) \tag{3}$$

In the formula (3), A_0 represents a pheromone constant, and A_i^{GA} represents a pheromone value converted according to the calculation result of the genetic algorithm.

Objective function design:
The ant colony algorithm optimization solution is mainly a quadratic solution to the better solution obtained by the genetic algorithm. Here, the node domain of the ant colony algorithm is regarded as an undirected graph Gaco (V, E), and V and E represent nodes of all resources and a network set, respectively. The ultimate goal of the ant colony algorithm is to find an optimal path from the network geometry of all resources of the undirected graph.

Pheromone update:
In the actual solution, in order to avoid trapping the local optimal solution, the local pheromone of the ant selected line should be updated first. After the ant is transferred from node i to j, the pheromone of Aij can be updated by formula (4).

$$A_{ij}(t+1) = (1-\psi)A_{ij}(t) + \sum_{k=1}^{m} \Delta A_{ij}^{k}(t) \tag{4}$$

$$\Delta A_{ij}^{k}(t) = \begin{cases} \frac{B}{\mathrm{Re}source(e_{ij})}, \text{The k-th ant chooses } e_{ij} \text{ between t and t} + 1 \\ 0, \end{cases} \tag{5}$$

In the formula, $\psi \in (0,1)$ and $1-\psi$ respectively represent the update coefficient and residual coefficient of the pheromone; $\Delta A_{ij}^{k}(t)$ represents the pheromone left by the k-th ant on the path at time t; B represents the strength of the pheromone.

Equations (4) and (5) are mainly for global update of pheromone after an iterative update for all ants.

Termination condition design:
The ant colony algorithm can be terminated as long as any of the following conditions are met:

(1) The maximum number of design iterations is reached;
(2) The evolution rate of the offspring of the new population is lower than the design value.

3 Comparative Experiment

In order to verify the effectiveness of the hybrid algorithm, experiments are needed. CloudSim, a cloud computing simulation tool, is used to build an experimental simulation platform for resource rationality in cloud computing environment. The experimental environment is as follows: Select the cluster data that the network exposes, including 100 physical nodes, each node is configured as 24 Core, 64 G memory, 10 Gbps network. Using genetic algorithm, ant colony algorithm and hybrid algorithm, the reasonable scheduling of cloud computing resources under abnormal network is simulated.

The genetic algorithm, ant colony algorithm and hybrid algorithm were used to conduct reasonable scheduling experiments on 100 resources. The time required for the three algorithms to complete the rational scheduling of resources is shown in Fig. 3.

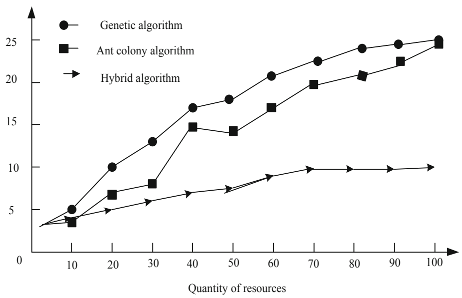

Fig. 3. The time required for the three algorithms to complete the rational scheduling of resources

It can be seen from Fig. 3 that compared with the other two algorithms, the hybrid algorithm requires the least time for reasonable resource scheduling, and it takes only 10 s to complete the scheduling. And from the time span, although the number of resources is constantly increasing, the average value has good stability.

In order to further verify the advantages of the hybrid algorithm in the scheduling effect, the resource load situation after three algorithms is analyzed. The result is shown in Fig. 4.

Figure 4 shows the load data of each resource point when the number of resources is 100. From the ant colony algorithm, the ant colony calculation and the resource load degree of the hybrid optimization algorithm, it can be seen that some resource points have a higher load level and some resource points have a lower load level. The ant colony algorithm and the genetic algorithm load on the resource points 5, 7, and 8 respectively, and the load on the resource points 3, 4, and 8 is small. The load of the hybrid optimization algorithm is more balanced, indicating that the overall load balancing performance of the hybrid algorithm is optimized.

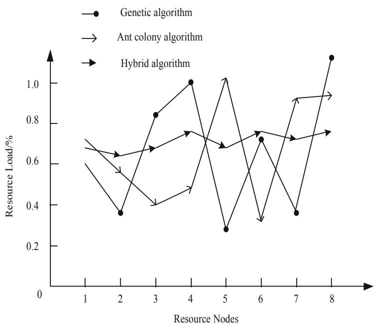

Fig. 4. Resource load situation after scheduling by three algorithms

4 Conclusions

In summary, this study discusses some issues about the rational scheduling of cloud computing resources under abnormal networks, and analyzes the advantages and disadvantages of current commonly used resource scheduling algorithms. Through in-depth research and comparison of genetic algorithm and ant colony algorithm, it is found that the two scheduling strategies have the characteristics of complementary advantages; On this basis, a rational resource scheduling strategy based on genetic-ant colony algorithm is proposed. This strategy integrates the dual advantages of genetic algorithm and ant colony algorithm in resource scheduling problems. The simulation experiment results show that the proposed scheduling strategy is superior to genetic algorithm and ant colony algorithm in large-scale task environment, and is superior to ant colony genetic algorithm in general. How to use the genetic ant colony algorithm based resource scheduling strategy for practical work will be the next research goal.

Acknowledgements. Supported by the Fundamental Research Funds for the Central Universities, Southwest Minzu University, “A Study on the Cloud Load Balancing of Data Stream Storage in Desktop Cloud Environment” Project No. 2019NQN54.

References

1. Qiu, X., Dai, Y., Xiang, Y., et al.: Correlation modeling and resource optimization for cloud service with fault recovery. IEEE Trans. Cloud Comput. **7**(3), 693–704 (2017)
2. Mao, L., Qi, D.Y., Lin, W.W., et al.: An energy-efficient resource scheduling algorithm for cloud computing based on resource equivalence optimization. Int. J. Grid High Perform. Comput. **8**(2), 43–57 (2016)
3. Zuo, L., Shu, L., Dong, S., et al.: A Multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. IEEE Access **3**, 2687–2699 (2017)
4. Jin, A., Song, W., Wang, P., et al.: Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing. IEEE Trans. Serv. Comput. **9**(6), 895–909 (2016)
5. Zhang, R., Wu, K., Li, M., et al.: Online resource scheduling under concave pricing for cloud computing. IEEE Trans. Parallel Distrib. Syst. **27**(4), 1131–1145 (2016)

6. Cui, X., Zeng, C., Xu, Z., et al.: Resource scheduling strategy in cloud computing based on greedy algorithm. Microelectron. Comput. **33**(6), 41–43 (2016)
7. Kumar, D., Baranwal, G., Raza, Z., et al.: A systematic study of double auction mechanisms in cloud computing. J. Syst. Softw. **125**, 234–255 (2017)
8. Meng, L.X., Meng, L.W.: Simulation of resource load balancing scheduling under cloud computing. Comput. Simul. **35**(4), 386–389 (2018)