Research on Cloud Computing Resource Scheduling Strategy Based on Firefly Optimized Genetic Algorithm

IOP Conf. Series: Materials Science and Engineering **563** (2019) 052104 doi:10.1088/1757-899X/563/5/052104

Research on Cloud Computing Resource Scheduling Strategy Based on Firefly Optimized Genetic Algorithm

Yaning Han[1, 2], Jinbo Wang[2], Zhexi Yao[1, 2]

[1] University of Chinese Academy of Sciences, Beijing

[2] Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China

toxiaoning@163.com

Abstract. In the cloud computing system environment, combined with the first-level scheduling model of task-virtual machine resource nodes, the individual coding, fitness function, selection replication and cross-variation process are redesigned, and the cloud computing resource scheduling model based on genetic algorithm is established. Corresponding to fireflies and virtual machine resource nodes, this paper redesigned the firefly decision domain update method, selected attraction probability formula and location movement strategy, and combined with genetic algorithm to establish cloud computing resource scheduling model based on firefly-genetic algorithm. Experiment with the CloudSim cloud computing simulation platform. The results show that the task completion time of the resource scheduling model is smaller than that of the single genetic algorithm. The virtual machine load is more balanced, the task completion time is short, and the overall optimization effect of the resource scheduling scheme is obvious.

1. Research background

The large-scale use of virtual machine technology in the cloud computing system greatly enhances the scalability of the network, thereby accessing cloud computing resources anytime and anywhere according to the dynamic needs of users, improving the service quality, service capability and service efficiency of cloud computing service providers, in view of its The advantages of low cost, virtualization technology support, and diversified services have been rapidly developed on a large scale and widely used.

In the cloud computing environment, the common load scheduling algorithms include a polling algorithm, a random mode, a hash mode, a fastest response, and a minimum connection mode. These algorithms are relatively simple, and the direction of resource scheduling optimization is relatively simple. The advantages and disadvantages of each algorithm are relatively obvious, resulting in the failure of the load balancing effect of the virtual machine and the imperfect resource scheduling strategy. In addition to this there are some research load balancing algorithms, as follows:

Wenzhong X [3] proposed an improved genetic algorithm based on simulated annealing. In the iterative evolution process, the improved algorithm is based on the dual fitness function based on task average completion time and load balancing to judge the low-adaptity and retain excellent individuals. The adaptive cross-variation probability function was also redesigned when the genome was mutated.

Wenqing C [5] proposed an intelligent optimization strategy for virtual resource scheduling. Combining virtual machine resource characteristics, optimizing chromosome replication selection and

cross mutation in population, designing virtual machine load function and cloud task optimal span function as dual fitness function.

Lijiao G [6], an improved genetic algorithm (IGA) based on chromosome coding method and fitness function is proposed, and the algorithm simulation experiment is carried out on the cloud simulator CloudSim. The results show that the proposed algorithm is superior to traditional genetic algorithms in terms of performance and quality of service QoS, and can be better applied to cloud computing environment resource scheduling under large-scale tasks.

2. Resource scheduling description

2.1. Resource scheduling architecture

The computing data center has a large amount of computing resources and storage resources to provide users with a wealth of services, which are allocated to users on an as-needed basis. The resources in cloud computing are abstracted into virtual resources. The management of virtual resources is the core technology of cloud computing. The efficient resource scheduling method is the key to ensure the efficient operation of cloud systems. At present, cloud computing resource scheduling is mainly divided into two levels. The first-level scheduling is a matching problem between the cloud task and the virtual machine resource, and the cloud task is mapped to the virtual machine resource, so that the virtual machine completes the task at the fastest time and realizes the maximum virtual resource. The second-level scheduling is a mapping between virtual machines and physical machine resources, so that virtual machines are created or migrated on the appropriate hosts. The first-level scheduling studied in this paper, namely cloud tasks and virtual machine scheduling.

2.2. Resource scheduling model

Suppose the cloud computing system has n cloud tasks T, ie T={t1, t2, t3, ..., tn}, m computing resources Vm, ie VM={Vm1, Vm2, Vm3, ..., Vmm}, then The following form is described:

ti={id，length, pesNumber, fileSize, outputSize…}

Where, ti represents the i($0 < i \le n$)th computing task in the task set T, id represents the task number, length represents the length of the task at execution time, pesNumber represents the number of processors to be used during execution, and fileSize represents the file size before the task is submitted. outputSize represents the output size after the task is executed.

Vmj={vmid，mips，ram，bw…}

Vmj represents the jth ($0 < j \le m$) resource node in the computing resource set VM, vmid represents the virtual machine number, which is the unique attribute for identifying the virtual machine, mips represents an indicator of the processing performance of the computer, represents the computing power of the virtual machine, and ram represents the virtual machine. Memory, bw represents network bandwidth.

The computing task is assigned to the computing resource node, and each task can only be executed on one resource node, and the mapping relationship between the task set T and the virtual resource VM is represented by the matrix X as formula (1):

$$X[n][m] = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \quad (1)$$

Wherein, x_{ij} is the distribution relationship between the task i and the virtual resource j, and the following relationship is also satisfied:

$$x_{ij}=\begin{cases}1,(t_i \rightarrow vm_j)\\ 0,(t_i \nrightarrow vm_j)\end{cases} \tag{2}$$

That is $x_{ij}=1$, if t_i is assigned to execute, otherwise $x_{ij}=0$. Therefore, an expected completion time to ETC (Excepted Time to Completion) of the task set T and the virtual resource set Vm can be derived, as expressed by the formula (3):

$$ETC[n][m]=\begin{bmatrix}ETC_{11} & ETC_{12} & \cdots & ETC_{1m}\\ ETC_{21} & ETC_{22} & \cdots & ETC_{2m}\\ \vdots & \vdots & \ddots & \vdots\\ ETC_{n1} & ETC_{n2} & \cdots & ETC_{nm}\end{bmatrix} \tag{3}$$

Among them, $ETC_{ij}=\frac{t_i.length}{vm_j.mips}$ is the ratio of the length of the task to the computing power of the virtual machine. Set BT_{ij} to the start time of the task t_i execution, then the task completion time CT_{ij} is as in formula (4):

$$CT_{ij}=BT_{ij}+ETC_{ij} \tag{4}$$

All task completion time for virtual resource j is as shown in formula (5):

$$AT(T)=\frac{1}{n}\sum_{j=1}^{m}Sum(vm_j) \tag{5}$$

For the resource scheduling problem, the virtual machine resource utilization is maximized and the resource waste is reduced, that is, the average completion time is required to be the smallest, that is, the objective function description is as shown in formula (6):

$$objFun(T)=\min\{\frac{1}{n}\sum_{j=1}^{m}Sum(vm_j)\} \tag{6}$$

3. Research on cloud computing resource scheduling strategy based on genetic algorithm

3.1. Individual coding and population initialization

In this paper, the resource-task real number coding method is used to initialize individual populations, and the generation method is random. Suppose that n computing tasks are deployed in m virtual machine resources, denoted by t and Vm respectively. Set the cloud computing task id={0,1,2,3,...,n}, virtual machine resource node id={0,1,2,3,...,m}, then the chromosome gene length is m The genotype is the virtual machine resource id, which ranges from 0 to m-1. Such as the following chromosome: {3,1,3,2, ... ,m-2,m-1}

3.2. Fitness function

For the resource scheduling problem, it is required to maximize the virtual machine resource node utilization and the task completion time is as small as possible. Therefore, it is decided to use the task average completion time as the objective function for description. Therefore, in the calculation, the reciprocal of the task average completion time is selected as the The fitness function of the individual, the objective function expression is as in formula (7):

$$\text{fitness(x)}=\frac{\text{n}}{\sum_{j=1}^{m} Sum(\text{vm}_\text{j})} \tag{7}$$

3.3. Genetic operator

3.3.1. Individual choice and copy. Selective replication is to simulate the survival of the fittest mechanism in nature, search for better solution space that satisfies the constraints, and retain individuals with large fitness function (smaller task execution time) into the population of the offspring. Commonly used is the roulette selection method based on the fitness ratio, wherein the selection probability formula of the fitness function is:

$$\text{P(x)} = \frac{fitness(\text{x})}{\sum_{x}^{R} fitness(\text{x})} \tag{8}$$

3.3.2. Cross and variation. Select two individuals with higher probability from the population according to the cross-probability function (9) to exchange a certain or some genes

$$P_c = \begin{cases} k_1(\text{f}_{\max} - \text{f}^{'}) / (\text{f}_{\max} - \text{f}_{\text{avg}}), \text{f}^{'} \geq \text{f}_{\text{avg}} \\ \text{k}_2, \text{f}^{'} < \text{f}_{\text{avg}} \end{cases} \tag{9}$$

Among them, $k_1, k_2 (0 < \text{k}_1 < 1, 0 < \text{k}_2 < 1)$ is the constant, $\text{f}_{\max}$ is the largest fitness value in the population, and f_{avg} is the average fitness value at each iteration, $\text{f}^{'}$ is the larger fitness value among the two intersecting individuals. A single point crossing is a randomly selected position on a chromosome that divides a chromosome into two parts (except for the initial or end position of the position point). According to the calculated probability, two individuals are selected in the population, and the crossover position is used to randomly cross the genes at the two positions.

Mutation operation is another method of generating new individuals. Compared with cross-operation, mutation is considered locally, which enhances the ability to search locally, and makes individuals closer to the optimal solution. At the sa In order to test the effect of the proposed me time, mutation maintains the diversity of the population and the probability of mutation. Function as formula (10)

$$P_{\text{m}} = \begin{cases} k_3(\text{f}_{\max} - \text{f}^{'}), \text{f} \geq \text{f}_{\text{avg}} \\ \text{k}_4, \text{f} < \text{f}_{\text{avg}} \end{cases} \tag{10}$$

Among them, $k_3, k_4 (0 < \text{k}_3 < 1, 0 < \text{k}_4 < 1)$ is the constant, $\text{f}_{\max}$ is the largest fitness value in the population, and f_{avg} is the average fitness value at each iteration, $\text{f}^{'}$ is the fitness value of the mutant individual.

4. Resource scheduling based on firefly algorithm

In the population, each firefly represents a resource scheduling scheme, and fluorescein brightness represents the fitness value of the scheduling scheme.

4.1. Update the firefly fluorescein value in the middle group

Use random initialization, according to formula (1), In the t-th iteration of the firefly population, the fluorescein is updated as in formula (11):

$$l_i(\mathrm{t}) = (1-\rho)l_i(\mathrm{t}-1) + \gamma j(x_\mathrm{i}(\mathrm{t})) \tag{11}$$

Wherein, $l_i(\mathrm{t})$ is the fluorescein value representing the firefly i in the t generation updating, $j(x_\mathrm{i}(\mathrm{t}))$ represents the objective function value of the relative position, such as the formula (12), $x_i(\mathrm{t})$ is the position of the firefly at that time, ρ indicates the emission rate of fluorescein, γ indicates the rate of fluorescein updating.

$$j(x_\mathrm{i}(\mathrm{t}))=\frac{1}{1+\mathrm{e}^{-x_\mathrm{i}(\mathrm{t}-1)}} \tag{12}$$

4.2. Update dynamic decision domain

The decision domain refers to the field of view of this firefly, and the fireflies within this range can be attracted, which is the search space of the current scheduling scheme. The size of the decision domain directly affects the convergence state of this algorithm. If the decision domain is too large, it will cause the iteration to be slow and waste computing resources; if it is too small, it will easily cause premature convergence and convergence too fast, and the algorithm will fall into local optimum. Therefore, according to the resource scheduling scheme after each iteration The fitness value updates the search space. The dynamic decision domain update method is as shown in formula (13).

$$r_d^i(\mathrm{t}+1) = \min\{\mathrm{r_s}, \max\{0, r_d^i(\mathrm{t}) + \beta(|\,\mathrm{N_i(t)}\,| - \mathrm{n_t})\}\} \tag{13}$$

Where $r_d^i(\mathrm{t}+1)$ is the perceived radius of firefly i in the t+1 generation update, r_s is the maximum perceived radius, β is the decision domain update parameter, n_t is the threshold for the number of nearby fireflies, and $\mathrm{N_i(t)}$ is the firefly set in the decision domain, which is expressed as formula (14) .

$$\mathrm{N_i(t)} = \{\mathrm{j}:\|\mathrm{x_j(t)}\text{-}\mathrm{x_i(t)}\|<\mathrm{r}_d^i(\mathrm{t}); l_i(\mathrm{t}) < l_j(\mathrm{t})\} \tag{14}$$

4.3. Select the direction of movement for iterative updating

For the firefly set $\mathrm{N_i(t)}$ in the dynamic decision domain of firefly i, let firefly j be within the decision domain of i, ie $j \in N_i(\mathrm{t})$, the probability set of fireflies i being attracted as in formula (15).

$$\mathrm{j}=\max(\mathrm{P_{i1}}, \mathrm{P_{i2}}, \mathrm{P_{i3}}, ..., \mathrm{P_{iN_i(t)}}) \tag{15}$$

The probability that i chooses to move to j is as in formula (16).

$$\mathrm{P_{ij}(t)} = \frac{l_j(\mathrm{t}) - l_i(\mathrm{t})}{\sum_{k\in N_i(\mathrm{t})} l_k(\mathrm{t}) - l_i(\mathrm{t})} \tag{16}$$

Assuming firefly i chooses to move to firefly j, the position is updated as in formula (17).

$$x_\mathrm{i}(\mathrm{t}+1)=x_\mathrm{i}(\mathrm{t}) + s\times\left(\frac{x_j(\mathrm{t}) - x_\mathrm{i}(\mathrm{t})}{\|\,x_j(\mathrm{t}) - x_\mathrm{i}(\mathrm{t})\,\|}\right) \tag{17}$$

Where, x is the moving step size.

5. Resource scheduling based on firefly-genetic algorithm

The genetic algorithm obtains the optimal solution through adaptive adjustment and probabilistic search, and has good global search ability. The genetic algorithm tends to be precocious and easy to be trapped in local optimum, which leads to unreasonable resource scheduling scheme. The firefly algorithm has

fewer parameters, relatively simple operation, and better reliability. The relative optimal solution obtained by the genetic algorithm is used as the initial solution of the firefly algorithm, which not only ensures the excellent quality of the initial solution of the firefly algorithm, but also greatly improves the search efficiency of the firefly algorithm, and is beneficial to jump out of the local optimum and obtain the global optimal solution.

Table 1. Firefly-genetic algorithm.

INPUT：cloud computing tasks, virtual machine resources, and related parameters
OUTPUT：optimal resource scheduling scheme
1 BEGIN
2 population initialization, setting related parameters
3 IF within the number of iterations
4 calculating individual fitness within a population
5 choose to copy, cross and mutate
6 END IF
7 generating resource scheduling scheme
8 initialize the firefly population
9 IF within the number of iterations
10 update the value of fluorescein in the population
11 update the dynamic decision domain corresponding to the firefly to determine the direction of firefly movement
12 update location
13 END IF
14 decoding to get the optimal resource scheduling scheme
15 END

6. Analysis of results

In order to test the effect of the proposed algorithm in resource scheduling optimization in cloud computing environment and verify whether the load balancing expected effect can be achieved, this paper builds the CloudSim simulation platform under the local hardware resource environment and uses some test data randomly generated, which's range is 100~10000.

Comparative experiments were performed using a single genetic algorithm (GA) and the firefly-genetic(GA-FA) algorithm proposed in this paper. The core parameters of the experimental hardware platform are Intel® CoreTM i7-4710MQ CPU @2.5GHz, 8G memory. The proposed algorithm setting parameters are shown in **Table 2** and **Table 3**:

Table 2. GA's parameters.

	population size	number of iterations	k_1	k_2	k_3	k_4
value	50	50	0.6	0.4	0.08	0.01

Table 3. FA's parameters.

	population size	number of iterations	ρ	γ	r_s	β	n_t	s
value	30	50	0.4	0.6	5	0.2	5	5

In the experiment, the cloud computing task length is set from 1000 to 40000, and the virtual machine computing performance mips is set from 1000 to 5000. The control variable method is used to analyze the change of the number of cloud tasks and the number of virtual machine resources to the average task completion time and virtual machine load. Impact. Due to the large randomness of the algorithm, we take the average of multiple simulations as a reference result to ensure the effect of the experiment.

(a)The number of virtual machine resource nodes is 5. Change the number of computing tasks and record the average time for task completion in **Table 4**.

Table 4. Task completion time for each number of computing tasks

	20	40	80	150	300	500	1000	2000
GA	20.71	43.24	70.20	126.08	244.17	459.28	898.66	1870.66
GA-FA	23.98	41.78	67.69	128.60	206.65	418.04	810.54	1658.39

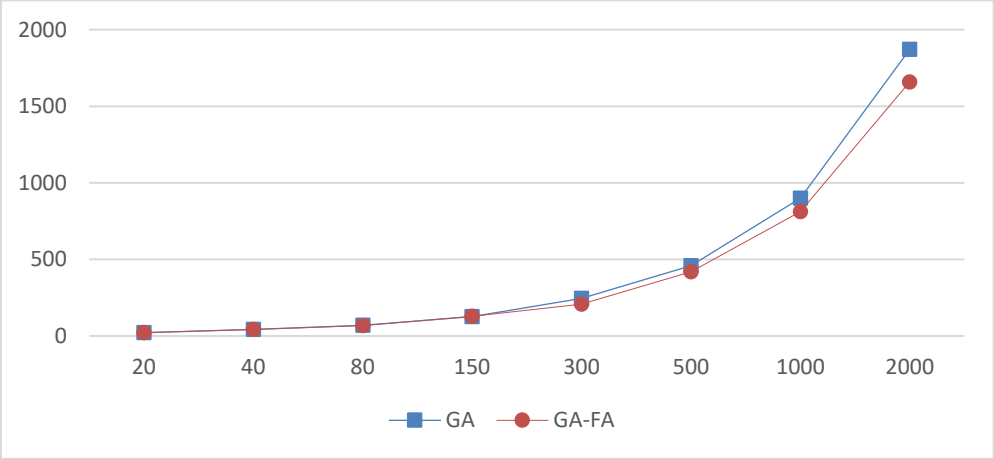

Figure 1. Task completion time for each number of computing tasks.

As **Figure 1**, When the number of tasks is small, there is almost no difference in the completion time of the same number of tasks under the two algorithms. With the gradual increase of the number of cloud computing tasks, the improved algorithm task completion time is obviously shorter than the single genetic algorithm, and the improved algorithm optimizes the original allocation scheme.

(b)When the total number of computing tasks is 300, the number of virtual machine resources changes, and the average time to complete the task is recorded, as shown in **Table 5**.

Table 5. Task completion time under the number of virtual machine resources

	5	10	20	30
GA	244.17	141.38	84.79	51.32
GA-FA	206.65	145.06	79.52	48.86

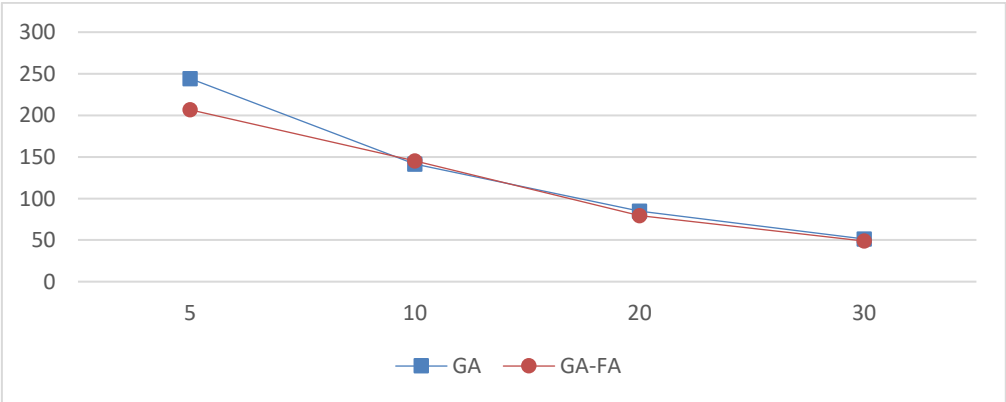

Figure 2. Task completion time under the number of virtual machine resources.

As **Figure 2**, When the number of cloud computing tasks is constant, when the virtual machine computing resources are small, the task completion time under the improved algorithm is significantly shorter than the task completion time under the improved algorithm. As the number of virtual machine resources increases, the task completion time under the algorithm is significantly reduced, and the trend

tends to be consistent. This is because the virtual machine computing cluster is expanding, its task processing capability has fully met the task requirements.

7. Summary

The cloud computing resource scheduling strategy based on genetic algorithm and firefly algorithm is proposed. From the experimental results, the optimized algorithm task completion time is shortened and the scheduling efficiency is improved. However, this algorithm still has some shortcomings, such as ideal experimental environment, fixed virtual machine performance parameters, and computing task parameters.

References

[1] Yu L, Zhiwen Z, Xiaolan L, Lingrong K, Shuhuan Y and Yanfang Y 2012 resource scheduling strategy based optimized generic algorithm in cloud computing environment *Journal of Beijing Normal University* **48(04)** p 378-84

[2] Ruiqi C 2016 study of cloud computing resource management algorithm based on improved genetic algorithm *Beijing University of Technology*

[3] Wenzhong X, Zhiping P and Jinglong Z 2015 research on cloud computing resource scheduling strategy based on genetic algorithm *Computer Measurement & Control* **23(05)** p 1653-56

[4] Feng L, Li B and Jun Y 2016 An Improved Genetic Algorithm for Cloud Computing Resource Scheduling *Computer Measurement & Control* **24(05)** p 202-06

[5] Wenqing C, Xueying C 2016 resource scheduling and optimization method in cloud computing environment *Laser Journal* **37(06)** p 115-18

[6] Lijiao G, Qingsheng W 2015 intelligent optimisation strategy for virtual resource scheduling in cloud environment *Computer Applications & Software* **32(06)** p 308-11

[7] A. Yousif, A. H. 2012 Optimizing job scheduling for computational grid based on firefly algorithm *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT), Kuala Lumpur* pp 97-101

[8] J. Lu. 2016 Improved Genetic Algorithm-Based Resource Scheduling Strategy in Cloud Computing *International Conference on Smart City and Systems Engineering (ICSCSE), Hunan* pp 230-34

[9] S. Chen, J. Wu and Z. Lu 2012 A Cloud Computing Resource Scheduling Policy Based on Genetic Algorithm with Multiple Fitness *IEEE 12th International Conference on Computer and Information Technology, Chengdu* pp 177-84

[10] S. S. Rajput and V. S. Kushwah 2016 A Genetic Based Improved Load Balanced Min-Min Task Scheduling Algorithm for Load Balancing in Cloud Computing *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), Tehri* pp 677-81

[11] Xiao-Ke Li, Chun-Hua Gu 2015 Virtual machine placement strategy based on discrete firefly algorithm in cloud environments *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu* pp 61-66