Research on Improved Hybrid Particle Swarm Optimization Algorithm for Cloud Computing Task Scheduling

Xiaoguang Yang[1, a *], Qian Wang[2,b], Yimin Zhang[3,c]

[1,2,3]Dalian Neusoft Institute of Information, Dalian, Liaoning 116023, China

[a]yangxiaoguang@neusoft.edu.cn, [b]wangqian@neusoft.edu.cn, [c]zhangyimin@neusoft.edu.cn

Keywords: Cloud computing; Task scheduling; Ant colony algorithm; Hybrid PSO algorithm

Abstract. In the cloud computing environment, one of the hot spot of researches in cloud computing is how to accomplish the service request in numerous running tasks. This paper puts forward an improved hybrid particle swarm optimization, firstly using particle swarm algorithm as the main level algorithm, the initial solution rapidly, followed by the max min ant colony algorithm, as the algorithm to find the optimal solution based on the initial solution. Finally, the availability and advantage of the proposed algorithm can be tested through the simulation experiment.

Introduction

With the continuous development of information technology, the way in which human society disseminates information knowledge has undergone a revolutionary transformation. The emergence of Internet technology breaks the restriction of physical geography. Computers in different regions can transmit data through the Internet, which greatly facilitates the circulation of information knowledge. However, there are still many problems in the development of the Internet at present. Individual computer operating systems in the Internet are different, making information sharing difficult and operating costs high. With the development of network especially broadband network and the gradual maturity of virtualization technology, computer network and computer have become an integral whole. In order to solve such problems, people use "cloud computing" to share resources in the network. When there is a task requiring resources in the network, resources can be applied to achieve on demand [1].

Cloud computing is a new form for existing computing and service modes. Its resources exist in a virtual form and can be flexibly scaled according to customer needs. At the same time, cloud computing has incomparable advantages compared with other Internet technologies, such as running costs, resource utilization efficiency, etc. Therefore, cloud computing will play an indispensable role in the future development of Internet technology. When cloud computing is dealing with large-scale tasks, its scheduling algorithm will directly affect the efficiency of data processing in its resource pool, and then affect the customer's experience. Therefore, the task scheduling algorithm in cloud computing is the key part to determine the development of cloud computing [2].

Task scheduling in cloud environment is a kind of NP problem [3]. There are many ways to solve NP problem, such as genetic algorithm [4], ant colony algorithm [5], particle swarm optimization [6], and so on. In cloud computing environment, genetic algorithm is used to schedule tasks accordingly [7]. Particle Swarm Optimization (PSO) is used to optimize the task scheduling in cloud environment, which shortens the task scheduling time [8]. Ant colony algorithm is used to optimize the task scheduling in cloud environment, which minimizes the total and average time of task scheduling in limited resource environment [9].

But for the above optimization algorithm, there are some problems. For example, the optimized ant colony algorithm still relies heavily on the initial parameter setting, so it is easy to fall into the local optimal solution; after optimization, the search ability of particle swarm optimization is still obviously insufficient, which affects the effect of solving the optimal solution. In order to solve this problem, a Hybrid Particle Swarm Optimization (HPSO) algorithm is proposed in this paper, which uses PSO in the primary algorithm and improved Ant Colony Optimization algorithm (ACO) in the subordinate algorithm. The algorithm has both advantages of the two algorithms. The initial solution can be obtained quickly by PSO, and then the ACO is used on the basis of the initial solution. Finding the optimal solution improves the efficiency of cloud computing system scheduling.

Algorithm Design

Particle swarm optimization algorithm.

PSO is an evolutionary algorithm based on stochastic optimization technology, which has the advantages of easy implementation, high precision and fast convergence [10]. The basic principle is that in the N-dimensional search space, there is a group containing m particles and search for flight at a certain speed. In the search space, the position and speed of the No.i particles are set to be $Xi=\left[x_{i,1}, x_{i,2},\ldots\ldots, x_{i,n}\right]$ and $Vi=\left[v_{i,1}, v_{i,2},\ldots\ldots, v_{i,n}\right]$ $(1 \le i \le m)$.Then, by calculating the fitness of each particle in the population, the best position $V_i(t)=\left[p_{i,1}, p_{i,2},\ldots\ldots, p_{i,n}\right]$of each particle in the population and the best position $P_Q(t)$ of all particles in the population at that time are obtained, and the velocity and position of each particle in the population are updated according to the following Eq.1.

$$v_{i,j}(t+1)=\omega v_{i,j}(t)+c_1r_1\left[p_{i,j}(t)-x_{i,j}(t)\right]+c_2r_2\left[p_{g,j}\left|(t)-x_{i,j}(t)\right|\right] \quad (1)$$

$$x_{i,j}(t+1)=x_{i,j}(t)+v_{i,j}(t), j=1,2\cdots n \quad (2)$$

Where, ω is the weight of inertia, *c1* and *c*2 are learning factors, *r*1 and *r*2 are mutually independent pseudo-random Numbers, subject to uniform distribution on [0, 1].

As can be seen from the speed update Eq.2, the *c*1 parameter enables the particle to adjust its step length and then fly towards its best position. The *c*2 parameter can also adjust the step length of the particle itself and make the particle fly towards the best global position.

ω is the influence of time *t* on the velocity at time *t*+1. When ω large, time t is affects the velocity at time *t*+1 more. At this point, the particle search capability is strong in the global scope. When ω is relatively small, the velocity at time *t* has little influence on the velocity at time *t*+1. At this point, particle search capability is relatively strong in the local range. Therefore, the value of ω can be used to make the algorithm difficult to fall into the local optimal solution.

Under the action of velocity update formula, the particle constantly adjusts its position, so that the optimal solution of the problem can be solved. The particle position update mode in particle swarm optimization algorithm can be described by Fig.1.

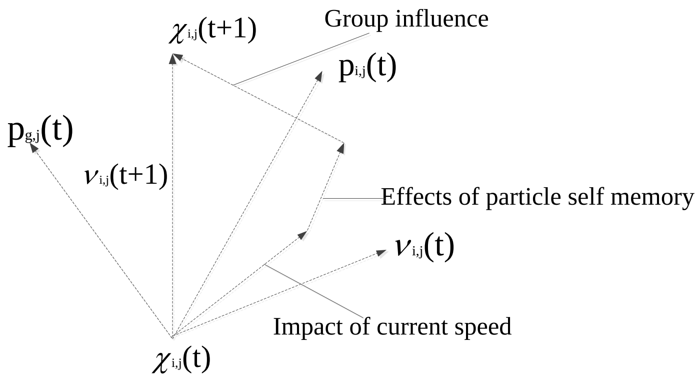

Figure 1. Schematic diagram of particle position update

Ant colony algorithm.

The principle of the basic ant colony algorithm is as follows: firstly, the constant *n* is set as the number of ants in the algorithm, the *k* position of the individual at time *t* is $p(t)=r$, and the next step of the individual is $P(\mathrm{t}+1)=\mathrm{s}$, which is determined according to Eq.3.

$$s=\begin{cases}\arg \max\limits_{s\in allowed_k(t)}\left\{\tau_{rs}^{\alpha}(t)\eta_{rs}^{\beta}(t)\right\}, & q\le q_0\\ S, & others\end{cases} \tag{3}$$

Among them, between [0,1], the system is set artificially, S is the position of the next moment of the individual, and the selection is made according to a certain probability distribution, the rule is shown in Eq.4.

$$p_s^k(\mathrm{t})=\begin{cases}\dfrac{\tau_{\mathrm{rs}}^{\alpha}(\mathrm{t})\eta_{\mathrm{rs}}^{\beta}(\mathrm{t})}{\sum\limits_{u\in allowed_k(\mathrm{t})}\tau_{\mathrm{ru}}^{\alpha}(\mathrm{t})\eta_{\mathrm{ru}}^{\beta}(\mathrm{t})}, & s\in allowed_k(\mathrm{t})\\ 0, \quad others\end{cases} \tag{4}$$

Among them, $\tau_{rs}(\mathrm{t})$ and $\eta_{rs}(\mathrm{t})$ represents the concentration and expectation of pheromone left by the individual ant on the path. α represents the relative importance of pheromone concentration, β represents the relative importance of expectation for reaching the next position, and the allowable position of the individual is represented by $allowed_k(\mathrm{t})$.

When the individual *k* in the ant colony moves in the search space, the pheromone concentration in the path will be updated. The formula Eq.5 is as follows:

$$\tau_{\mathrm{rs}}\left(\mathrm{t}+1\right)=\left(1\text{-}\lambda\right)\tau_{\mathrm{rs}}\left(\mathrm{t}\right)+\lambda\tau_0 \tag{5}$$

Among them, $0<\lambda\le 1$ is an artificial setting constant, $\tau_0=\left(nL_{mm}\right)^{-1}$ is the initial pheromone on the path, and L_{mm} is the length of the adjacent nodes. It is not difficult to see from the above formula that pheromone updating formula can reduce the concentration of pheromone, reduce the attraction of a certain path to ants, so that ants have the possibility to explore other paths.

When the ants in the colony have finished searching, the shortest path will be found and the pheromone on all paths will be updated. The overall pheromone update formula Eq.6 is as follows.

$$\tau_{rs}(\mathrm{t}+1)=\left(1\text{-}\rho\right)\tau_{rs}(\mathrm{t})+\rho\Delta\tau_{\mathrm{rs}}(\mathrm{t}) \tag{6}$$

Among them, $0<\rho\le 1$ is a constant, indicating the speed of pheromone decay, $\Delta\tau_{rs}(t)=1/r$, L_{gb} is the shortest path found in the search process.

The max-min ant system is an improvement on the basic ant colony algorithm. A cycle does not update pheromones for all ants, but updates pheromones for only one ant. This ant can only be of two types: ants that find the best solution in the current cycle or ants that may find the best known path. In order to prevent the extreme situation of too large or too small pheromone on a certain path, the concentration range of pheromone is set$[\tau_{\mathrm{min}},\tau_{\mathrm{max}}]$. In order to attract more ants in the beginning of the search, pheromone concentration initialization value is no longer a constant, but set to the upper limit of the interval, and select a smaller volatility coefficient, in order to get more search paths. The whole pheromone update becomes Eq.7.

$$\tau_{ij}(t+n)=(1-\rho)*\tau_{ij}(t)+\Delta\tau_{ij}(t)^{best} \qquad (7)$$

Among them, $\Delta\tau_{ij}(t)^{best}=1/f(s^{best})$. $f(s^{best})$ represents the value of the current iteration optimal or global optimal solution.

Mixed particle swarm algorithm process.

The basic implementation steps of cloud computing task scheduling algorithm based on improved hybrid particle swarm optimization algorithm are as follows:

1. Define the target function for the scheduling task in the cloud environment.
2. The parameters and stopping conditions in the improved hybrid particle swarm optimization algorithm are artificially set.
3. The fitness value of the individual in the particle swarm is calculated by the objective function, and the optimal position of the sum group is calculated.
4. According to formula Eq.1 and Eq.2, we update the speed and position of individuals in a particle swarm.
5. Determine whether the number of iterations of particle swarm optimization algorithm reaches the set number, if the number of iterations, then output the initial results, and turn to step 6; otherwise, return to step 3, recalculate the particle fitness value.
6. According to the initial results obtained in the fifth step, we set the initial pheromone value in the slave algorithm.
7. The ants are randomly assigned to each node and searched.
8. According to formula Eq.3, the ant selects the next node and updates the local pheromone according to formula Eq.5, and adds the selected node to the scheduling table.
9. When all the ants in the ant colony complete a search, all tasks are scheduled, according to the task scheduling list to calculate the fitness of the scheduling results, and according to formula Eq.7 update the global pheromone, otherwise jump to step 8.
10. If the algorithm reaches the maximum and minimum number of iterations, stop the algorithm and get the global optimal solution; otherwise, go back to step 7 and recalculate.

Simulation experiment

In order to test the superiority of the algorithm proposed in this paper, a comparison experiment was conducted in two cases, namely, 100 virtual computing resources and 100 sub-tasks, and 100 virtual computing resources and 300 sub-tasks. At the same time, the pure ant colony optimization algorithm, the pure particle swarm optimization algorithm and the hybrid example group algorithm proposed in this paper were used to carry out experiments in the above two cases. Set related parameters of hybrid particle swarm algorithm: the particle swarm size is 200, $C_1=\mathrm{C}_2=2$ and the maximum number of iterations is 100;The maximum and minimum ant colony algorithm has a population size of 200, $\alpha=\beta=1$, $\rho=0.2$,and the maximum number of iterations is 150, and the two algorithms repeat 30 times each. The comparison simulation experiment is as follows:

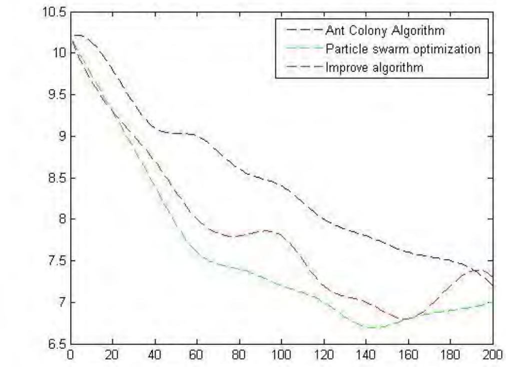

Figure 2. Three algorithms complete the simulation of 100 subtasks

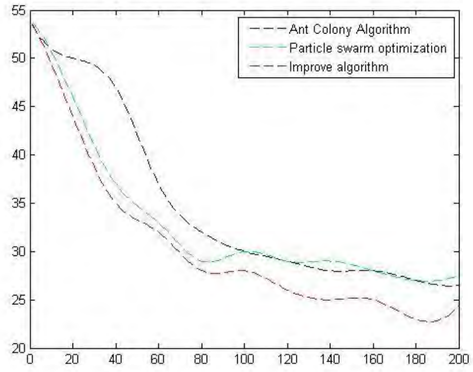

Figure 3. Three algorithms complete the simulation of 300 subtasks

In the graph above, the horizontal axis is the number of iterations, the vertical axis is the completion time, the star-shaped broken line is the pure ant colony algorithm, the circular broken line is the pure particle swarm optimization algorithm, and the square broken line is the hybrid particle swarm optimization algorithm proposed in this paper. From the above diagram, we can see that the hybrid particle swarm optimization algorithm takes less time to complete the same task than the simple particle swarm optimization and ant colony algorithm, and its advantages are more obvious when the task increases.

Summary

This paper proposes an improved hybrid particle swarm optimization algorithm and applies it to cloud computing. This method adopts the master-slave structure, and its master-level algorithm employs the particle swarm algorithm, and the slave algorithm applies the improved ant colony algorithm - the maximum and minimum ant colony algorithm. The simulation experiment shows that the proposed algorithm is more real-time and searchable than the simple particle swarm algorithm and ant colony algorithm. At the same time, this paper only considers the time factor of cloud computing scheduling task and ignores some other factors, such as cost factor, load factor, etc., so further research should be carried out on the basis of this paper to make the cloud computing scheduling task reach the optimal state under the action of comprehensive factors.

References

[1] H. Zhao, H. Wang and X. Liao: Systems Engineering-Theory & Practice, (2016) No.3, p.768-778.

[2] W.H. Tian, Y. Zhao: *Cloud Computing - Resource Scheduling Management* (National Defense Industry Press, Beijing 2011).

[3] B.Y. Fang, Y.Y. Zhang and Y. Cheng: Telecommunications Science, (2010) No.s1,p.1-5. (In Chinese)

[4] S. Wang: *Research on load balancing scheduling algorithm of cloud computing resources based on minimum transfer cost*(Master of Engineering, University of Electronic Science and technology of China, China 2013). (In Chinese)

[5] L.X. Wang, H. Zhao: Industrial instrumentation and automation devices, Vol. 2 (2016) p.3-6. (In Chinese)

[6] L.Z. Guo, Y.J. Geng，C.Y. Jiang and et al: Computer engineering and design, Vol. 34 (2013) No.7,p. 2358-2362.

[7] T. Wang, Z. Liu, Y. Chen and et al: *Control Conference. IEEE*(Chiang Mai, Thailand, June 26-27, 2016) p.146-152.

[8] L.Z. Guo, S.G. Zhao and SH.G. Shen: Journal of Networks, Vol. 7 (2012) No.3,p.547-553.

[9] J.F. Li, J. Peng and X.Y. Cao: Energy Procedia, Vol. 13 (2011) p. 6833-6840.

[10] S.J. Nirmala, S.M.S. Bhanu: Computing, Vol. 98 (2016) p. 1-19.