Study on Resources Scheduling Based on ACO Algorithm and PSO Algorithm in Cloud Computing

Xiaotang Wen[1], Minghe Huang[1], Jianhua Shi[1]
[1]School of Software
[1]Jiangxi Normal University
[1]Jiangxi, China
xiaotangwen2010@163.com, huangmh0093@sina.com, 0793shijianhua@163.com

***Abstract*—It improves the algorithm because of the shortcoming that the ACO algorithm is easy to fall into local optimal solution in the cloud computing resource scheduling. The improved algorithm makes particle optimization inosculated into ant colony algorithm, which first finds out several groups of solutions using ACO algorithm according to the updated pheromone, and then gets more effective solutions using PSO algorithm to do crossover operation and mutation operation so as to avoid the algorithm prematurely into the local optimal solution.**

Keywords-ACO algorithm; PSO algorithm; Cloud Computing; Resources Scheduling

I. INTRODUCTION

Allocation of computing resources is an important component of cloud computing technology, its efficiency directly affects the performance of the entire cloud computing environment. Currently, Domestic and foreign researchers study resource scheduling based on the ant colony algorithm(ACO) in cloud computing environment, which overcomes the shortcomings of huge size of the node and lower allocation of resources for a single node, ensuring user's jobs can be completed on time. But it still exist that the algorithm is easy to fall into local optimal solution, making inefficient resource scheduling. For this problem, it will improve algorithm by combining ACO with particle swarm optimization algorithm(PSO) to improve the efficiency of resource scheduling in cloud computing environments.

II. ALGORITHMS DESCRIPTION

2.1 ACO Description

Ant colony algorithm, also known as ant algorithm, is a bionic optimization algorithm which used to find an optimal path in a directed graph. The ants will leave a pheromone on the way of foraging, they communicate with others using pheromone to find a shorter path. The amount of pheromone on each path will reflect probability the other ants choose the path. Ants of a place is more, the strength of pheromone is greater. Ultimately, all ants will choose the shortest path. Ant colony algorithm is well suited for resources scheduling in cloud computing because of its concurrency and expandability.

- pheromone definition and update
  Inspired by Ref [1], we measure a node's pheromone by virtual machine's hardware resources. m is the number of CPU, p(MPIS) is processing capability, r is memory capacity, h is external memory capacity, b is bandwidth, which used to measure nodes' pheromone. This sets threshold with parameter according to Formula [1], if it exceeds the threshold, then it should calculate with threshold.
  $m_{max}=m_0$, $p_{max}=p_0$, $r_{max}=r_0$, $h_{max}=h_0$, $b_{max}=b_0$ (1)
- Initialize the hardware pheromone of the node :
  CPU's pheromone:

$$\tau_{ij}(0)=\frac{m*p}{m_0*p_0}\times 100\% \quad (2)$$

Memory's pheromone:

$$\tau_{ij}(0)=\frac{r}{r_0}\times 100\% \quad (3)$$

External memory's pheromone:

$$\tau_{ij}(0)=\frac{h}{h_0}\times 100\% \quad (4)$$

Bandwidth's pheromone:

$$\tau_{ij}(0)=\frac{b}{b_0}\times 100\% \quad (5)$$

Node i's pheromone is weighted sum of each pheromone above, that is

$$\tau_{ij}(0)=a\tau_{if}(0)+b\tau_{ir}(0)+c\tau_{ih}(0)+d\tau_{ib}(0)$$
$$a+b+c+d=1 \quad (6)$$

978-0-7695-4818-0/12 $26.00 © 2012 IEEE
DOI 10.1109/DCABES.2012.63

When a new task is assigned to the compute nodes, the CPU utilization ratio will increase and the pheromone decrease. Stipulate that when the new tasks are assigned to the compute nodes, the pheromone is updated by Formula (7).

$$\tau_i(t_1) = \tau_i(t) - \lambda\tau_i(t) \quad , \quad 0 < \lambda < 1 \tag{7}$$

$\tau_i(t)$ represents the pheromone concentrations of the compute nodes at time t, $\tau_i(t_1)$ represents the pheromone concentration that new tasks reach node i at time t_1, is regulatory factor.

$$p_{ij} = \frac{\tau_j^{\alpha} / A_j^{\beta}}{\sum_{m \notin N_s} \left(\tau_m^{\alpha} / A_m^{\beta}\right)}, \qquad j \notin N_s \tag{8}$$

$A_j = ET_j^{n_d}(J_d(t))$, p_{ij} represents the probability that ants select next node j in node i. τ_j represents the pheromone concentration that node i observed node j. N_s represents node set of ants path. m is neighbor of node i. α, β are regulatory factor., which denote importance of τ_j and A_j.[1]

2.2 Description of PSO Algorithm

Initialize a group of random particles in a space, the particle position represents possible solution, each particle advances to a certain speed, particle swarm gradually approaches to the optimal location after repeated advances which are also called iteration, thus the optimal solution will be got. In each iteration, the particles update themselves according to two extreme values: one is the optimal solution finding by a single particle, namely the individual extremum; the other is the optimal solution finding by whole particle swarm, that is, namely the global extremum.

Particles update their velocity and position according to the two extreme values above and the following two formulas.

$$V = \omega * v + c_1 * rand() * (pBest - X) + c_2 * rand() * (gBest - X) \tag{9}$$

$$X = X + V \tag{10}$$

Among them, V=[v1,v2,⋯,v d] is the velocity of the particle, X=[x1, x2, ⋯, x d] is the current position of the particle, d is the dimension of solution space. pBest is the individual extremum, gBest is the global extremum, rand( ) is random number between 0 and 1. c_1, c_2 are called learning factors, which are used to adjust the particle update step, ω is a weighted factor.

The particle swarm gradually closes to the optimal solution location through continuous learning and updating, finally, the gBest which is the global optimal solution will be got using the algorithm.[2]

III. IMPROVED RESOURCES SCHEDULING ALGORITHM IN CLOUD COMPUTING

Ant colony algorithm uses of pheromones to transmit information, and particle swarm optimization algorithm uses own information, individual extremum information and global extremum information to guide the particles to select next iteration position. Ant colony algorithm combines positive feedback principle organically with some kind of heuristic algorithm, but it is prone to premature convergence and falling into local optimal solution. Hybrid algorithm, the idea is to let the ants also have the characteristics of the "particles". First of all, ants are in accordance with the ant colony algorithm to complete a traverse, and then they adjust solutions according to the local optimal solutions and global optimal solutions. Adjustment is as follows: to resource scheduling, its current position is behalf of the basic path node set. If it uses particle swarm algorithm to solve the problem, the speed is hard to express, so here the idea of genetic algorithm is adopted. $\omega * v$ can be seen as the mutation operation of genetic algorithm, and $c_1 * (pBest - X) + c_2 * (gBest - X)$ in formula (9) can be seen as the crossover operation of genetic algorithm, which lets the current solution do crossover operator with the individual extremum and the global extreum so as to produce a new location.

In this paper, the crossover strategy as follows: select a cross- region randomly in the second string, old_2 is added to the old_1 corresponding position, delete the nodes in old_1 appeared in the old_2 cross- region. Variation strategy: select the node of j_1 visits from the nodes of 1-n visit, exchange the j_1 visit node and the j_1+1 visit node in node set N_0, and the rest keep unchanged. At this time, the path set node is N_1.

A node may have to run multiple tasks at the same time in the cloud computing. But, it will improve performance of entire cloud that tasks are assigned to the most efficient node. Therefore, a task expected execution time model is established to predict execution time of the new tasks on the node, the formula as follows:

$$ET_j^{n_d}(J_d(t_2)) = \frac{n_d}{n_v} \times \left(\rho ET_j^{n_v}(J_v(t_0)) + (1-\rho) RT_j^{n_v}(J_v(t_1))\right) \tag{11}$$

$ET_j^{n_d}(J_d(t_2))$ indicates that the new task J_d estimated the expected execution time on the j node at time t_2, n_d indicates the load of the node j at time t_2, here the number of task running on node j indicates the load size. $ET_j^{n_v}(J_v(t_0))$ indicates the expected execution time of the previous completed task J_v at time t_0. $RT_j^{n_v}\left(J_v\left(t_1\right)\right)$ indicates the actual execution time of the previous task completed at time t_1. n_v indicates the load of previous task.[1]

The tasks become less and less on a node, so the expected completion time of the tasks is also less and less. Therefore, the expected execution time should be modified at intervals according to the following formula:

$$ET_e^{n_d}\left(J_d\left(t+1\right)\right)=\left(1-\rho_1\right)\times ET_e^{n_d}\left(J_d\left(t\right)\right),\quad 0<\rho_1<1 \tag{12}$$

The algorithm steps of resources scheduling based on ACO algorithm and PSO algorithm in cloud computing are as follows:

- Step1: Initialize nc=0(nc is iterations), use ACO algorithm to complete the first traverse (form m path node set);
- Step2:Calculate the fitness value based on the current path node set(the expected execution time), set the current fitness value to individual extremum called ptbest, and set the current path nodes set to individual extreme nodes set called pcbest, then find out the global extremum called gtbest and the global extreme nodes set called gcbest;
- Step3:Put m ants randomly on n nodes, put initial staring node of each ant in the current path node set, to each ant k, move to the next node j according to the probability p_{ij}, put node j in the current path node set, compute task expected execution time based on formula(11).
- Step4: Operate as follows for each ant, the path node set of ant j termed N_0(j) is crossed by gcbest which gets N_1'(j), and N_1'(j) is crossed by pcbest which gets N_1''(j), N_1''(j) variants with a specified probability which gets N_1(j). accept the new value if the new objective function gets better, otherwise refuse and N_1(j) remains N_0(j), then, rediscover individual extremum ptbest and extremal node set pcbest for each ant, find out global extremum gtbest and global extremal node set gcbest;
- Step5: Calculate task expected execution time for each ant node set, record the current best solution;
- Step6: Update the pheromone concentration according to formula(7), modify the node task expected execution time according to formula(12);
- Step7: nc=nc+1;
- Step8: If nc<a predetermined number of iterations, go to step3;
- Step9: Output the optimal solution, allocate the tasks in the nodes which included in the optimal solution node set.

IV. THE EXPERIMENT

This paper does simulation using matlab7.0 to verify the improved algorithm. In order to verify the superiority of the improved algorithm, we are using ACO algorithm and PSO algorithm for task scheduling in this experiment. The expected execution time of task is shown in figure1. Experimental parameter settings: number of ants m=10, maximum number of iterations nc_{max}=500, α=1.5, β=2, ρ=0.9.

Overall, the tasks execute slowly because of a little pheromone when the ACO algorithm is put into use, but as the pheromone increasing later, and the positive feedback enhancing, the time increase rate becomes smaller. It is clear that the execution time for improved algorithm is shorter than the former, so it has higher efficiency.

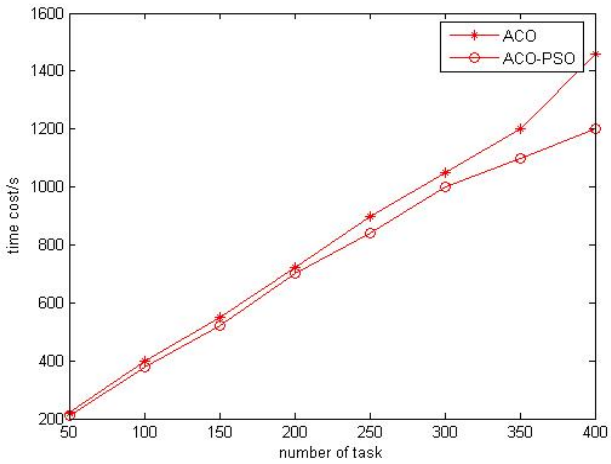

Figure 1.

V. CONCLUSION

This paper considered the series of characteristics of cloud computing, made particle optimization inosculated into ant colony algorithm and introduced crossover strategies and mutation strategies in genetic algorithm to overcome the defect that the algorithm is easy to fall into local optimal solution in the cloud computing resource scheduling. Experiment shows that the improved algorithm not only accelerated the convergence speed, but also avoided falling into local optimum solution, and achieved the purpose that the user tasks were efficiently provided appropriate resources in cloud computing, which improved the resource utilization ratio.

REFERENCES

[1] Y. Liu, X. H. Wang, C. M. Xing and S. Wang, "Resources scheduling strategy based on ant colony optimization algorithms in cloud computing," Computer Technology and Development, vol. 21(9), pp. 19-23, 2011.

[2] S. Gao, X. Z. Jiang, K. Tang and J. Y. Yang, "Hybrid Algorithm Combining Ant Colony Optimization Algorithm with Pariticle Swarm Optimization," Proceedings of the 25th Chinese Control Conference, pp. 1428-1432, 2006.

[3] Z. H. Zhang and X. J. Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation," 2010 2nd International Conference on Industrial Mechatronics and Automation, pp. 240-243, 2010.

[4] H. W. Tian, F. Xie and J. M. Ni, "Resource Allocation Algorithm Based on Particle Swarm Algorithm in Cloud Computing Environment," Computer Technology and Development, vol. 21(12),pp. 22-25, 2011.

[5] Y. G. Wang and R. L. H, "Study on Cloud Computing Task Schedule Strategy Based on MACO Algorithm," Computer Measurement and Control, vol. 19(5), pp. 1203-1204, 2011.

[6] S. H. Liu, Y. Zhang and H. Y. Wu, "Multi-robot task allocation based on swarm intelligence," Journal of Northeast Normal University (Natural Science Edition), vol. 41(4), pp. 69-72, 2009.