

The Load Balancing Ant Colony Optimization Based on Cloud Computing

Junyu Ye ^{a)}, Lichen Zhang ^{b)}

College of Computer Science, Guangdong University of Technology, Guangzhou, 510006.china

^{a)}364773360@qq.com

^{b)}lchzhang@gdut.edu.cn

Abstract. Ant Colony Optimization(ACO) is an intelligence-optimized algorithm Which can solve a large-scale task scheduling problem in cloud computing. However, the basic ant colony algorithm used in large scale task scheduling will easily lead to low distribution efficiency. To solve this problem, this paper analyzes the features of ACO and propose a load balancing ant colony optimization which will boost the utility of ants in the system. After that it use CloudSim, A tools for simulate the proposed algorithm as well as to compare it with the basic ACO. This result suggests that the proposed ACO can reduce the distribution scheduling execution time effectively and ensure the even distribution between nodes so that it can use in large scale scheduling problems.

Key words: Ant Colony Optimization, task scheduling, cloud computing, load balance.

INTRODUCTION

Originating from grid computing, cloud computing utilizes computer clusters to distribute computing on a large number of distributed computers [1], thus completing the processing of a large number of computing tasks submitted by users. How to carry out task scheduling effectively and maintain load balance is the most important research topic in cloud computing scheduling [2].

In the problem of cloud computing scheduling which mentioned above, the traditional allocation algorithm is difficult to achieve the load balancing requirements [3]. Some heuristic algorithm like Genetic Algorithm [4], Particle Swarm Optimization(PSO)[5-6], ant colony optimization[7-9] and other evolutionary algorithms have the characteristics of fast adaptability and low deviation degree of optimal solution.

Based on the characteristics of large-scale distribution scheduling, an improved ant colony algorithm is proposed to improve the efficiency and processing time of ant colony algorithm in large-scale scheduling problem by using the distinguish about effective ants.

CLOUD COMPUTING TASK SCHEDULING MODEL

The task scheduling of cloud computing combines two directions about user needs and resource allocation. The purpose is to make the task execution time span short and resources can achieve balanced load. The model of task scheduling can be expressed as $Q = \{T, VM, RT\}$. About that, $T = \{t_1, t_2, \dots, t_i, \dots, t_n\}$, is the set of tasks which the user wants to submit which included several independent tasks, $t_i (i=1, 2, \dots, n)$ represents the number i task. VM is a set of virtual machines which including several units and $vm_j (j=1, 2, \dots, m)$ represents a virtual machine that is no. J . 'RT' represents the processing time mapping of the set T and the virtual machine node VM. The matrix RT representation is as follows.

$$RT = \begin{pmatrix} RT_{11} & RT_{12} & L & RT_{1m} \\ RT_{21} & RT_{22} & L & RT_{2m} \\ M & M & O & M \\ RT_{n1} & RT_{n2} & L & RT_{nm} \end{pmatrix} \quad (1)$$

RT_{ij} represents the processing time required which the subtask t_i allocated to vm_j .

It can be seen from this that the task resource scheduling is transformed by the above mapping relation into the figure RT matrix. At this point, selecting the corresponding node assignment task can be regarded as the classic np-hard problem.

THE DESIGN OF EFFECTIVE ACO BASED ON LOAD BALANCING CONSTRAINT

Ant colony algorithm has some characteristics like strong robustness and it is a heuristic algorithm with strong adaptive and positive feedback. That characteristics makes ACO suitable for solving the large-scale task scheduling in cloud computing. But as the scale increases, the ACO diverges seriously. It is easy to get into local optimal solution when solving large-scale optimization problem and resulting in low efficiency of algorithm. This paper proposed an improved ant colony algorithm. The improved ant colony algorithm sets the load balance constraint condition to divides the effectiveness of the ant. After that it only updates Pheromone which those paths belong to effective ant.

Ant's path initialization

As we all known a VM node may assign multiple tasks. Therefore, in this improved algorithm, each ant generates a task scheduling sequence 'Qt' directly at the initialization stage for simulated randomly selected at the initialization time and selects different path behaviors in the basic ACO.

Qt = $\{q_1, q_2, \dots, q_i, \dots, q_n\}$. In queue Qt, q_i represents the unrepeatable task t_i for the queue location i. It also need to initialize VM selection sequence 'Qv' corresponding to Qt. Qv = $\{qv_1, qv_2, \dots, qv_i, \dots, qv_n\}$. At last it uses an array 'VMR' for used to calculate the total runtime of each virtual machine. VMR = $\{vmruntime_1, vmruntime_2, \dots, vmruntime_m\}$

Select computing nodes for the task

Like the basic ACO, it placed a number of ants randomly on nodes. p_{ij}^k represents the transfer probability of the No.k ant in the scheduling queue location I which choosing the No.j calculation node.

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{k=1}^m [\tau_{ik}]^\alpha [\eta_{ik}]^\beta} \quad (2)$$

τ_{ij} is the residual pheromone concentration of vm_j on task t_i . η_{ij} represents visibility. In this case $\eta_{ij} = 1/RT_{ij}$. α is the pheromone heuristic factor, and β is the visibility heuristic factor.

After calculating the outcome of the transfer probability p_{ij}^k , We use the method roulette to determine the vm_j selected by task t_i and write the corresponding vm_j to the VM selection sequence Qv. At the same time, the VMR update corresponds to the virtual machine run time.

$$vmruntime_j = vmruntime_i + RT_{ij} \quad (3)$$

Establishment of objective function.

Most of the cloud computing scheduling models use the MapReduce calculation model [10]. The tasks submitted by users are divided into multiple Map tasks and multiple Reduce tasks in parallel processing. Finally, the processing results are summarized and returned to the user. This MapReduce calculation model means that scheduling target is to reduce the maximum VM node processing time (the total run time of the computing node cluster).

Finish Time represents the maximum execution time of the VM node after the task assignment completes.

$$FinishTime = Max(VMR) \quad (4)$$

pheromone update

In ACO, when each iteration of the ant selects its path, it updates the pheromone.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij}$$

The ‘ ρ ’ is a parameter called Volatile coefficient, $\rho \in [0,1)$. $\Delta\tau_{ij}$ is Pheromone enhancement formula. In improved ACO, the pheromone enhancement strategy only updates the path selected by the effective ant. The effective ant is determined by the CPU utilization of the VM node, which called ‘util’.

$$util = \frac{\sum_{i=1}^m vmruntime_i}{FinishTime \times m} \times 100\% \quad (5)$$

The pheromone enhancement strategy uses the ant-cycle model.

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{FinishTime_k} & \text{no.k ant is effective ant} \\ 0 & \end{cases} \quad (6)$$

Algorithm steps

The improved algorithm flow chart is as follows.

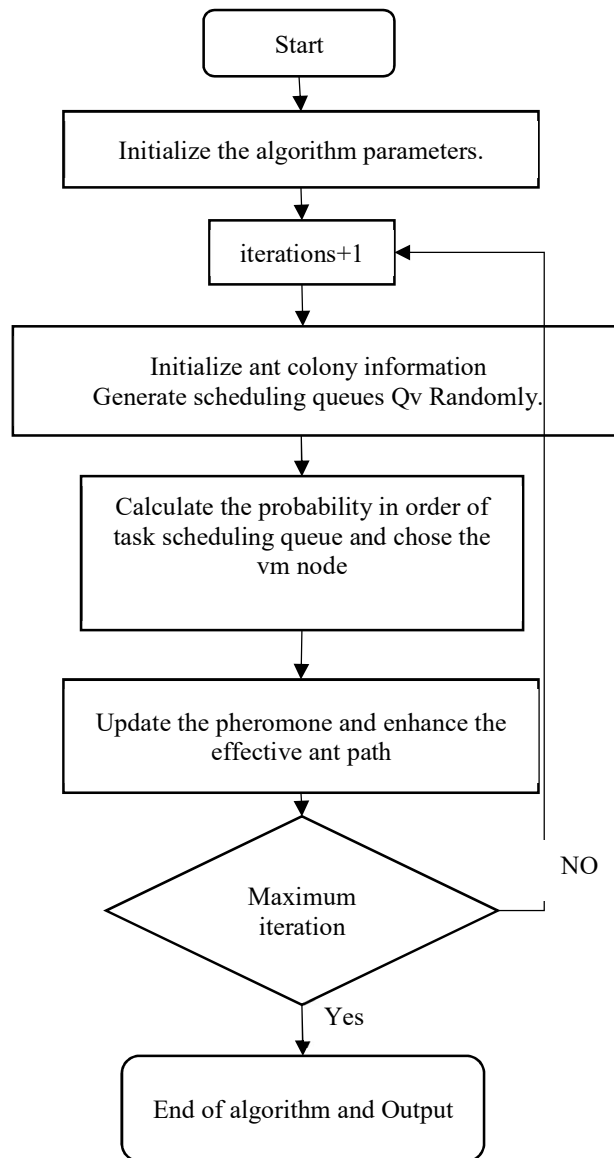


FIGURE1. Improved flow chart of ACO

ALGORITHM SIMULATION EXPERIMENT AND RESULT ANALYSIS

In this paper, it uses CloudSim[11] for experimental simulation. Table 1 shows the basic parameter and VM setting.

TABLE 1. the basic parameter and VM setting

Virtual Machine Setting		Algorithm parameter setting	
Number of virtual machines	50	The number of ants	5
		Initial pheromone τ	2
		α	1
The computing power of the virtual machine.	[200MI,400MI]	β	1
		ρ	0.3
Subtask length	[2000MI,5000MI]	Q	100
		Util	>80%

In the experiment, 50 virtual machine computing nodes were set up to test the processing time required for 500 to 5000 tasks respectively. Each experimental condition was run 20 times, and the average processing time was taken. The number of iterations sets 100 times. The experimental results are as Figure 2.

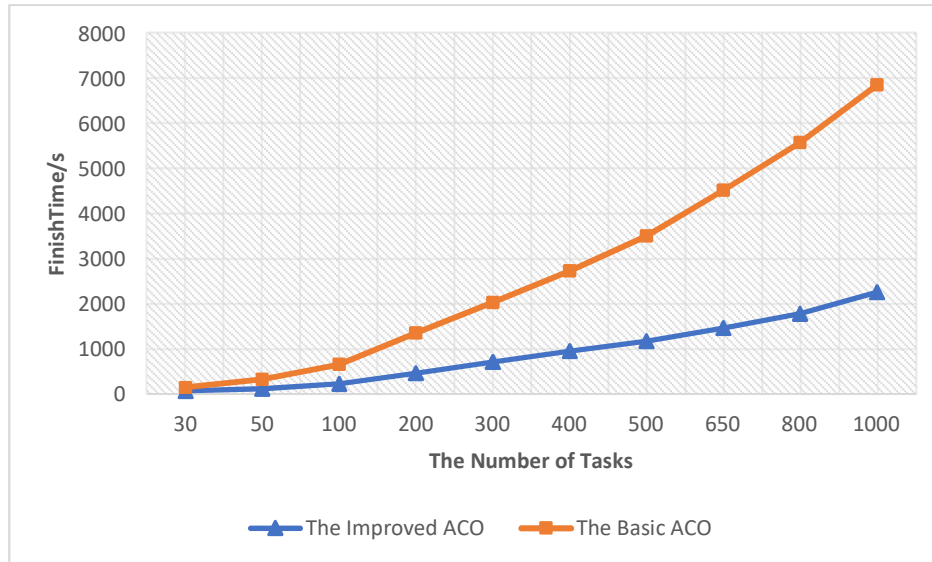


FIGURE 2. The processing time for multiple tasks.

It can be seen from the experimental results that, with the increase of the number of tasks submitted, the two algorithms correspond to the linear increase in the maximum processing time of VM. Due to the improved ant colony algorithm for pheromone enhanced add virtual machine CPU utilization as a constraint condition, the task to a more balanced distribution to different virtual machine, so the node total processing time is greatly reduced. The algorithm improves the effect greatly.

CONCLUSION

This paper studies the cloud computing task scheduling model and propose an improved ant colony algorithm model based on effective ant. The improved algorithm is added to the effective ant pheromone updating strategy of CPU utilization, so that the load of each node of the virtual machine is more balanced and the maximum node processing time is greatly reduced. This can reduce the execution time of ACO and accelerate the response speed. The results show that the improved algorithm is more suitable for cloud computing task scheduling.

ACKNOWLEDGMENTS

This paper is supported by the National Natural Science Foundation of China(No.61572142) and Natural Science Foundation of Guangdong Province (No.2015A030313490).

REFERENCES

1. Weiwei Lin, Deyu Qi. Survey of Resource Scheduling in Cloud Computing[J]. Computer Science,2012,39(10).
2. Liyun Zuo, Zhibo Cao. Review of scheduling research in cloud computing[J]. Application Research of Computers,2012,29(11):4023-4027.
3. Quan Chen, Qianni Deng. Cloud computing and its key techniques[J]. Journal of Computer Applications,2009,9(29): 2562-2567.
4. Jianfeng Li, Jian Peng.Task scheduling algorithm based on improved genetic algorithm in cloud computing environment[J]. Journal of Computer Applications,2011,31(1):184-186.

5. Dewen Wang, Xiaomeng Liu. Resource scheduling of cloud computing platform based on improved particle swarm optimization[J]. *Application Research of Computers*,2015,32(11):3230-3246.
6. Kaitao Shen, Demin Hu. Research on Task Scheduling Based on Cloud Computing and Improved Discrete Particle Swarm Algorithm[J]. *Computer Measurement & Control*,2012,20(11):3070-3072.
7. Yun Wei, Yuanyuan Chen. Cloud Computing Task Scheduling Model Based on Improved Ant Colony Algorithm[J]. *Computer Engineering*,2015,41(2):12-16.
8. Dorigo M,Blum C. Ant colony optimization theory: A survey[J]. *International Journal of Productivity Performance Management*,2010,59(2):811-828.
9. Gao Y. A Multi-Objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing[J]. *Journal of Computer and System Sciences*,2013,79(8):1230-1242.
10. Min Zhou. Overview of MapReduce [D]. Guangzhou: College of Information Science and Technology of Jinan University,2008.
11. Xiajun Wang. Study and Application of A Toolkit for Cloud Computing Simulation--CloudSim[J]. *Microcomputer Applications*,2013,29(8):59-61.