

Cloud Computing Task Scheduling Based on Pheromone Dynamic Adaptive Ant Colony Optimization

Junwei Ge^{1, a}, Yang Liu^{1, b} and Yiqiu Fang^{2, c}

¹*School of Computer science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;*

²*School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;*

^a*gejw@cqupt.edu.cn, ^b1445795785@qq.com, ^c994607515@qq.com*

Keywords: cloud computing; ant colony optimization; task scheduling; task completion time.

Abstract: Cloud computing is an Internet-based computing method that provides computers and other devices with shared software and hardware resources and information on the Internet according to their demands. Task scheduling is a core issue of cloud computing. Aiming at the resource scheduling problem of cloud computing, a task scheduling model of DQACO is proposed by improving the pheromone Q in the traditional ant colony optimization. Experiments show that the DQACO algorithm performs better in task completion time and can achieve better scheduling of tasks in the cloud environment.

1. Introduction

Cloud computing technology is a new technical model that emerged from the development of hardware and network technology to a certain stage^[1]. It contains most of the current information technology, such as distributed computing, virtualization, network, service, cloud computing platform and storage technology. It uses virtualization technology to form a huge resource pool of hardware and software resources^[2], which enables users to access resources on the Internet on demand and pay less. It is a new type of computing service model based on the Internet. Task scheduling is one of the core technologies of cloud computing, but there are some shortcomings in it. The ant colony optimization, which is one of the intelligent heuristic algorithms, is suitable for solving the NP problem. The cloud computing task scheduling is an NP-complete problem^[3]. However, because the ant colony optimization itself has the problem of lacking pheromone in the initial stage, which causes slow convergence and easy falling into local optimum, so its application still has some shortcomings in cloud computing task scheduling. Therefore, this paper starts with the dynamic adjustment of the pheromone accumulation, and optimizes the ant colony optimization to achieve the goal of minimum completion time, thus establishing a better task scheduling strategy.

2. Description of Cloud Computing Task Scheduling Problem

2.1 Task Scheduling Model

The essence of cloud computing task scheduling is to distribute n independent tasks to m virtual machines for execution. Most current cloud computing systems use the Map/Reduce working method, which is suitable for parallel computing of large-scale data sets. The cloud computing task scheduling process is shown in Figure 1.

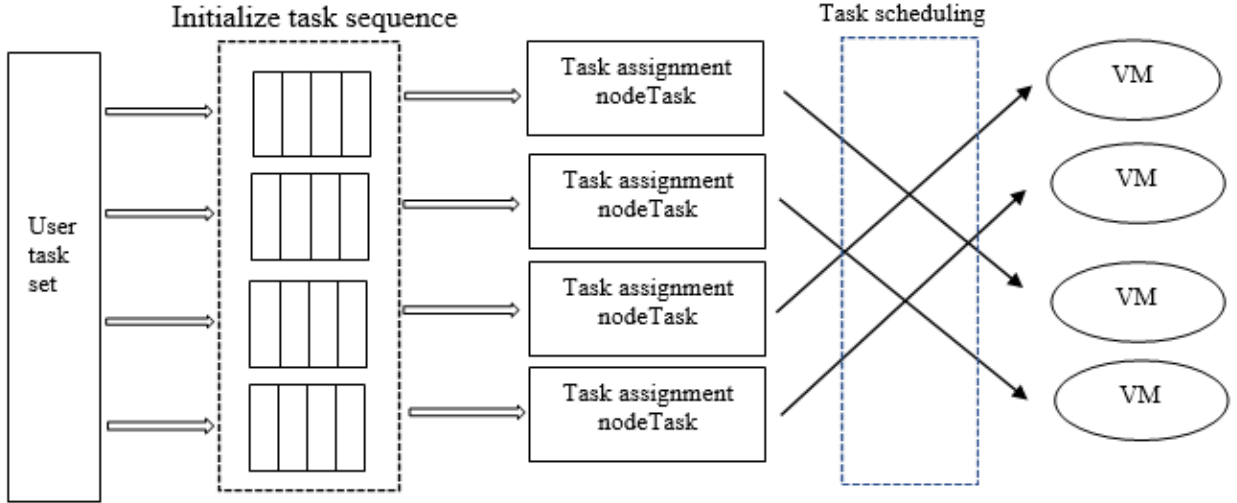


Figure 1

To describe the process of cloud computing task scheduling more clear, the following assumptions were made:

- (1) Tasks are independent and have no dependencies on each other;
- (2) The virtual machine will not be interrupted during task execution unless it fails.

2.2 Formal Description of the Problem

The user task set is defined as $T = \{t_1, t_2, \dots, t_m\}$, where m is the number of tasks and each task can only be executed on one virtual machine. The virtual machine set is defined as $VM = \{vm_1, vm_2, \dots, vm_n\}$. The matching relationship between the task and the virtual machine is represented by the matrix MT:

$$MT = \begin{bmatrix} mt_{11} & mt_{12} & \dots & mt_{1n} \\ mt_{21} & mt_{22} & \dots & mt_{2n} \\ \dots & \dots & \dots & \dots \\ mt_{m1} & mt_{m2} & \dots & mt_{mn} \end{bmatrix} \quad (1)$$

In the 1st formula, mt_{ij} denotes the distribution relationship between the No.i task and the No.j virtual machine, where $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, m\}$. Then the expected execution time of the task model can be represented by the matrix ET:

$$ET = \begin{bmatrix} et_{11} & et_{12} & \dots & et_{1n} \\ et_{21} & et_{22} & \dots & et_{2n} \\ \dots & \dots & \dots & \dots \\ et_{m1} & et_{m2} & \dots & et_{mn} \end{bmatrix} \quad (2)$$

In the 2nd formula, $e\tau_{ij}$ denotes the time required for the No.i task to operate on the No.j virtual machine, where $i \in \{1,2, \dots, n\}$, $j \in \{1,2, \dots, m\}$.

3. Improved Ant Colony Optimization

3.1 Standard Ant Colony Optimization

The ant colony optimization is an artificial intelligence optimization algorithm^[4] that simulates the behaviors of ants searching for food and then backing to home. Ants collaborate through pheromones and are widely used in TSP issues and network routing. The state transition probability of the ant colony optimization in the process of ant k selecting the next node at time t is:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in allowed_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & \text{If } j \in allowed_k, \\ 0, & \text{or} \end{cases} \quad (3)$$

in the 3rd formula, $allowed_k$ represents the set of nodes that the ant has not visited, α is the information heuristic factor; β is the expected heuristic factor; η_{ij} is the heuristic function associated with the edge (i, j); τ_{ij} is the pheromones on the edge (i, j).

In order to avoid residual pheromone covering the heuristic information, the ant colony optimization introduces a pheromone volatilization mechanism. When the ant completes a search, it updates the pheromone on each side. The rules are as follows:

$$\begin{cases} \tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \\ \Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \end{cases} \quad (4)$$

In the formula, ρ represents the volatilization coefficient of the pheromone, then $1-\rho$ represents the pheromone residual factor. $\Delta\tau_{ij}(t)$ and $\Delta\tau_{ij}^k(t)$ respectively represent the pheromone increments of all ants and ant k remaining on the edge (i, j) in this cycle.

$\Delta\tau_{ij}^k(t)$ uses the Ant-Cycle model, the specific rules are as follows:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{if the No.k ant passes (i,j) in the circle} \\ 0 & \text{or} \end{cases} \quad (5)$$

3.2 Improvement of Ant Colony Optimization

The pheromone intensity Q in the ant colony optimization will affect the convergence speed of the algorithm to some extent. The standard ant colony optimization is prone to stagnation and local optimality when it is scheduled for cloud computing tasks. For the pheromone intensity Q is a fixed value and it will affect the convergence speed to a certain extent in the standard ant colony optimization, the pheromone dynamic adaptive ant colony optimization^[5] is applied to cloud computing task scheduling. The specific adjustment rules are as follows:

$$Q_{g+1} = \begin{cases} Q_g + k \cdot \frac{(Q_{max} - Q_{min})}{MAX_GEN} & (Q_{g+1} < Q_{max}) \\ Q_{max} & \text{or} \end{cases} \quad (6)$$

In the formula, MAX_GEN represents the maximum number of iterations set in the algorithm, g represents the number of iterations, and Q_g represents the value of the No.g generation pheromone intensity Q. Q_{max} and Q_{min} represent the maximum and minimum values allowed for the

pheromone intensity respectively. The best values are obtained by the experiment.

Due to the accumulation of pheromones, the search process will fall into a local optimum. If the optimal solution obtained in successive n generations does not change, the search is considered to be caught in a certain extreme point (may not be the global optimal solution). At this time, mandatory measures are taken to reduce the content of the pheromone, so that the maximum pheromone volatilization factor ρ is adopted to try to make it jump out of the local minimum.

3.3 Steps of the solutions for the Improved Algorithm in the Cloud Computing Task Scheduling Problem

- (1) Initializing the parameters of the ant colony optimization;
- (2) Randomly deploying all ants on the initial node;
- (3) Calculating the state transition probability of the ant, and move the ant to the next node according to the result;
- (4) Locally updating the pheromone and make corresponding modifications to the taboo table of a single ant;
- (5) Repeating 3) ~ 4) until all individuals of the ant colony find a solution, and then updating global pheromone and changing the information intensity strength Q ;
- (6) Judging whether it is locked in a local optimum, and if so, changing the pheromone volatilization factor;
- (7) The number of iterations increases. If the number of iterations reaches the maximum number of iterations, the search is stopped. The current result is the solution for this optimization.

4. Experiment and Result Analysis

4.1 Experimental Environment and Parameter Settings

4.1.1 Task and virtual machine settings

The number of tasks is between [100,250], the task length is between [30000, 3600000], the number of virtual machines is between [5,20], the virtual machine performance is between [500,1500], and the unit is Mips. The virtual machine memory range is [256, 2048], the number of virtual machine is 1, the bandwidth is 100M/s, and the storage space is 10240M.

4.1.2 Parameter settings in the algorithm

The parameters of the algorithm are set in the reference ^[6], and are determined after the test: $\alpha=2.0$, $\beta=5.0$, $\rho=0.6$, the number of ants is 10, and the maximum number of iterations of the algorithm is $N_{max} = 20$.

4.2 Simulation Results and Analysis

Experiment 1 compares the optimization results of the two algorithms by setting different numbers of virtual machines when the number of tasks is fixed. Experiment 2 compares the optimization results of the two algorithms by setting different task numbers when the number of virtual machines is fixed. The experimental results are shown in Figure 2 and Figure 3 respectively.

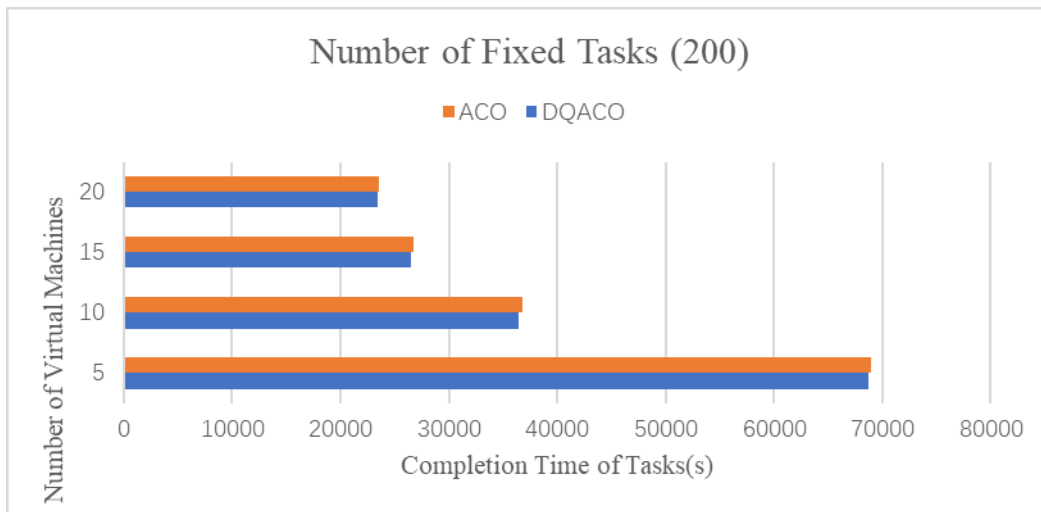


Figure 2

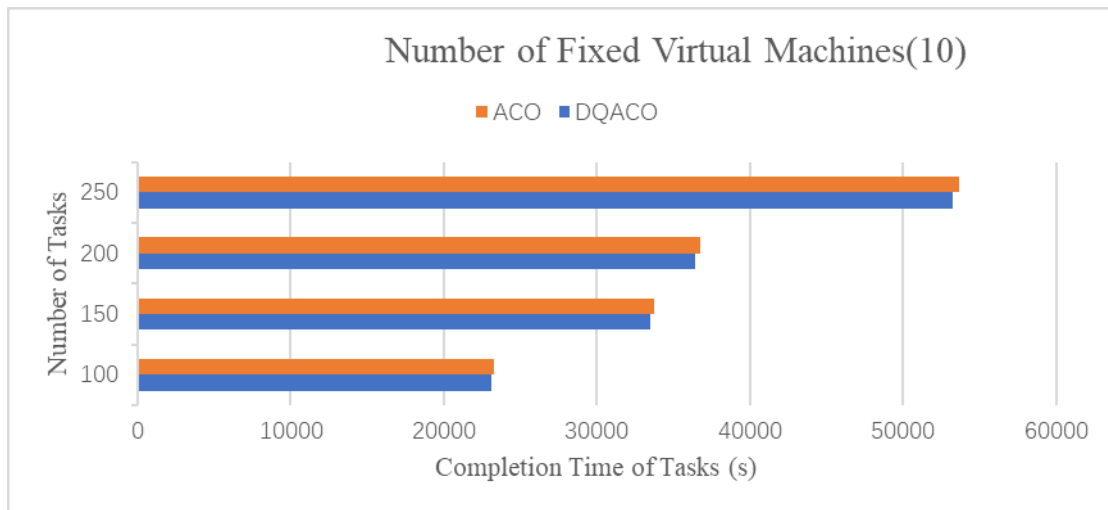


Figure 3

It can be seen from Figure 2 and Figure 3 that, compared with the standard ant colony optimization, the improved ant colony optimization DQACO has a better performance than it in the completion time of the cloud computing scheduling task, whether the number of virtual machines is fixed, the number of tasks is fixed or unfixed, and the number of virtual machines is uncertain or not. Thus, the DQACO algorithm is feasible in cloud computing task scheduling to reduce task completion time.

5. Summary

For the task scheduling problem in cloud computing, the pheromone dynamic adaptive is applied to cloud computing task scheduling. The algorithm achieves the purpose of fast convergence by dynamically adjusting the pheromone intensity Q and adopts mandatory measures to make it jump out of the local optimum when it is trapped in local optimum. The simulation results show that the proposed algorithm outperforms the standard ant colony algorithm in reducing the completion time of cloud computing task scheduling.

References

- [1] Peng Wang. *Cloud computing and big data technology*[M]. People's Posts and Telecommunications Press,2014, p.1-2.
- [2] Yong Wei, Kaixin Zhao, Songqing Zhang, et al. *Research on Cloud Computing Task Scheduling Based on Improved Ant Colony Algorithm*. *Fire command and control*.Vol.42(2017) No.5, p.130-133.
- [3] Thusoo A, Shao Z, Anthony S, et al. *Data warehousing and analytics infrastructure at facebook* [C]. Indianapolis, Indiana: SIGMOD'10, ACM, 2010,p.1013:1020.
- [4] Yanfei Zou, Shuying Liu. *Cloud computing resource scheduling model based on improved ant colony algorithm*[J]. *Journal of Jilin University (Science Edition)*. Vol.55(2017) No.3, p.679-683.
- [5] Gangli Qin, Jiaben Yang. *Ant Colony Algorithm for Adaptive Adjustment of Pheromone*[J]. *Information and control*.Vol.31(2002) No.3, p.198-201.
- [6] Junwei Ge, Qiang Guo, Yiqiu Fang. *Research on The Task Scheduling Strategy Based on Multi-Objective Optimization in Cloud Environment*[J]. *Microelectronics and Computer*. Vol (2017) No.11, p.63-67.