

# Task Scheduling Policy Based on Ant Colony Optimization in Cloud Computing Environment

Lin Wang and Lihua Ai

**Abstract** Cloud computing can provide strong processing capacity to tackle huge amounts of requests from many users. Task scheduling problem is the keystone to Cloud computing. In this paper we propose a task scheduling policy based on Ant Colony Optimization. This policy can minimize the makespan of the tasks submitted to the cloud system.

**Keywords** Cloud computing • Task scheduling • Ant Colony Optimization

## 1 Introduction

Cloud computing is a new computing model which aims at providing services on demand. It is development of distributed computing, parallel computing and grid computing. With the features of flexibility, virtualization etc., it can provide dynamic and scalable resources to users [1].

Task scheduling policy plays a significant role in Cloud computing. Efficient task scheduling can improve the resource utilization and enhance the performance of Cloud computing system. Task scheduling problem is considered as a NP-hard problem. It is usually be solved by heuristic algorithms which can converge to the optimal or near-optimal solution.

Ant Colony Optimization (ACO) is a heuristic algorithm which is based on the behavior of ants seeking the shortest path between anthill and the location of food source. With the mechanism of positive feedback and distributed cooperation, it is proved to be a useful heuristic algorithm for solving NP-hard problems.

---

L. Wang (✉) • L. Ai  
School of Computer and Information Technology, Beijing Jiaotong University,  
Beijing 100044, People's Republic of China  
e-mail: [wanglin@bjtu.edu.cn](mailto:wanglin@bjtu.edu.cn); [lhui@bjtu.edu.cn](mailto:lhui@bjtu.edu.cn)

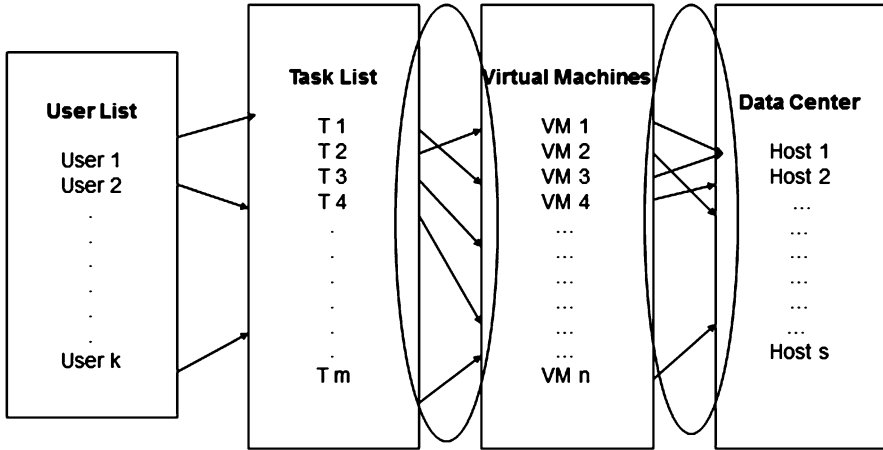


Fig. 1 Scheduling model of Cloud computing

A task scheduling policy based on ACO is proposed to improve the efficiency of Cloud computing system, which has been proved by the simulation experiment results in a Cloud simulator called CloudSim.

## 2 Scheduling Model

As shown in Fig. 1, task scheduling problem is finding a solution to assign the users' tasks to the virtual machines with the minimum time from the submission of the first task to the completion of the last task.

The set of tasks is  $T = \{T_1, T_2, \dots, T_m\}$  ( $m$  is the total number of tasks).

The set of virtual machines is  $VM = \{VM_1, VM_2, \dots, VM_n\}$  ( $n$  is the total number of virtual machines). Some restrictions are as follows:

1. One task can only be allocated to one virtual machine.
2. There is no dependency relationship between the tasks.

## 3 Scheduling Policy Based on Ant Colony Optimization

The original ACO is designed to solve the traveling salesman problem. We change some mechanisms of ACO to adapt to task scheduling problem [2].

### 3.1 Initialization

Set the initial values of  $\eta_i$  according to the capacity of the virtual machine resources. Every ant chooses a VM for the first task randomly.

### 3.2 Selection Mechanism

Ant  $k$  selects a task in sequence from the deferred task list and assigns the task to resource  $j$  according to the probability formulation:

$$P_j^k(t) = \frac{[\tau_j(t)]^\alpha \cdot [\eta_j]^\beta}{\sum [\tau_j(t)]^\alpha \cdot [\eta_j]^\beta} \quad (1)$$

In Formula (1),  $\tau_j(t)$  is the amount of pheromone which resource  $j$  owns at time  $t$ ,  $\eta_j$  is the inherent capacity of the resource  $j$ ,  $\alpha$  is a parameter to control the influence of  $\tau_j(t)$ ,  $\beta$  is a parameter to control the influence of  $\eta_j$ .

### 3.3 Global Update

When every ant finds a complete solution for all the tasks, we can update the pheromone values of the virtual machines selected by the best ant which finds the shortest makespan according to this formulation:

$$\tau_j^{new} = (1 - \rho) * \tau_j + \Delta\tau_j \quad (2)$$

In Formula (2),  $\Delta\tau_j$  is increment and  $\rho$  is the pheromone evaporation coefficient.

Besides, we limit the range of the pheromone values to avoid the pheromone converging too fast.

## 4 Pseudo Code

Pseudo code of resource selection process as follows:

```

Begin
Nc : number of iterations;
Nmax : number of the maximum iteration;
Initialize; Every ant selects a virtual machine randomly for the first task;
do
    Every ant processes a solution according to the selection mechanism;
    record the best solution and update pheromone;
while(Nc < Nmax)
End.
```

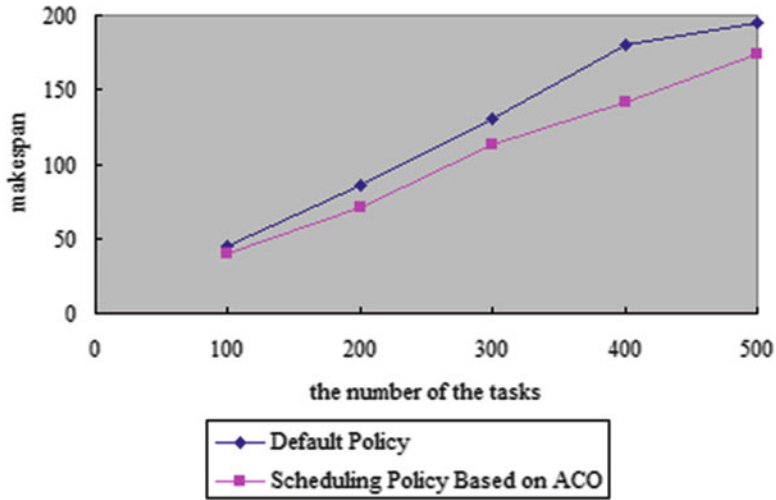


Fig. 2 Comparison of simulation results

### 5 Experiment and Result Analysis

CloudSim is a simulation toolkit for Cloud computing. The default scheduling policy provided in CloudSim schedules sequentially between a list of virtual machines and a list of tasks [3].

By extending *DatacenterBroker* class, we can design the proposed scheduling policy based on ACO [4].

We use makespan (the total finish time of the tasks) to evaluate the performances of the default policy and the proposed policy.

In Fig. 2, the result shows that the scheduling policy based on ACO can bring smaller makespan in comparison to the default policy of CloudSim.

### 6 Conclusion

In this paper we propose a task-scheduling policy based on ACO and evaluated it in a Cloud simulator with the number of tasks varying from 100 to 500. The results of above experiment demonstrate the effectiveness of the algorithm.

## References

1. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Gener Comput Syst* 25:599–616
2. Dorigo M, Stützle T (2004) *Ant colony optimization*. MIT Press, Cambridge
3. Calheiros RN, Ranjan R, Beloglazov A, De Rose CAF, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw Pract Exp* 41:23–50
4. The CLOUDS (2012) Lab: CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services. <http://www.gridbus.org/cloudsim/>. Accessed on 2012