Cloud Computing Task Scheduling based on Dynamically Adaptive Ant Colony Algorithm and QoS Perception

Haiqin Liu

Department of Mathematics and Information Engineering, Dongchang College of Liaocheng University, Liaocheng Shandong 252000, China

Keywords: Cloud computing; Task scheduling; QoS; Dynamic adaptive

Abstract: Cloud computing is designed to integrate resources and to improve resource utilization. In the cloud computing environment, due to the varied and widely distributed resources, dynamic and changeling user demand, how to close automatically and efficiently task scheduling for the user to the appropriate virtual resources become one of the cloud computing data center is facing enormous challenges. To this problem, an ant colony optimization scheduling algorithm is proposed, dynamic adaptive adjusting pheromone volatilization factor, the user's QoS requirements was developed for the resource selection. By Cloud Sim experiment, the results show that the algorithm can improve the global search ability and avoid falling into local optimum prematurely, can improve the algorithm convergence speed and can balance resources load.

1. Introduction

Cloud computing is a service model that provides a service that provides users with resources such as infrastructure, applications, and development platforms. Cloud computing aims to integrate resources and improve resource utilization. Because of the diversity and wide distribution of resources in cloud computing environment and the dynamic change of user demand, the dynamic allocation and management of virtual resources is one of the great challenges facing cloud computing data centers. So how to schedule tasks reasonably and efficiently and achieve global optimization becomes a difficult and important problem in the research.

Many researchers have proposed some effective scheduling algorithms, such as Zuoliyun and others [2] An improved Min-Min algorithm combining task priority and minimum completion time is proposed. The algorithm divides resources according to the computing power and communication ability of resources, taking into account the minimum execution time and load balance. Wang Fang et al. [3] A cloud computing task scheduling algorithm based on dynamic adaptive ant colony algorithm is proposed. This algorithm uses chaotic disturbance strategy in resource node selection, which effectively reduces the probability of falling into local optimal solution and stagnation. In the process of solving the information element volatilization factor is set dynamically and the information element is updated dynamically. Deng Jianguang[4] A cost-driven cloud computing task scheduling strategy is proposed. The cloud computing task scheduling model established by this method can not only meet the user's QoS requirements, but also take into account the interests of the provider. Chayinghua[5] In this paper, an enhanced ant colony algorithm is proposed for cloud computing task scheduling, which not only considers the load balancing of resources and the minimum completion time of task scheduling, but also sets the ant search object as a task assignment on a virtual machine.

Although the existing ant colony algorithm can improve scheduling efficiency and load balancing to a certain extent, there are still problems that are prone to local optimal solutions and convergence speeds that need to be improved, and few algorithms can take into account the user's QoS needs. In this paper, based on adaptive ant colony algorithm, the task scheduling strategy of cloud computing is proposed, and according to the real-time solution of the algorithm, dynamically

  DOI: 10.25236/icemc.2019.100

adjust the pheromone volatilization factor, thereby improving the global search ability of the algorithm and preventing the algorithm from falling into the local optimal solution prematurely. The algorithm also takes into account the user's QoS requirements so that the system resources can be used more effectively.

2. Cloud Computing Task Scheduling Model

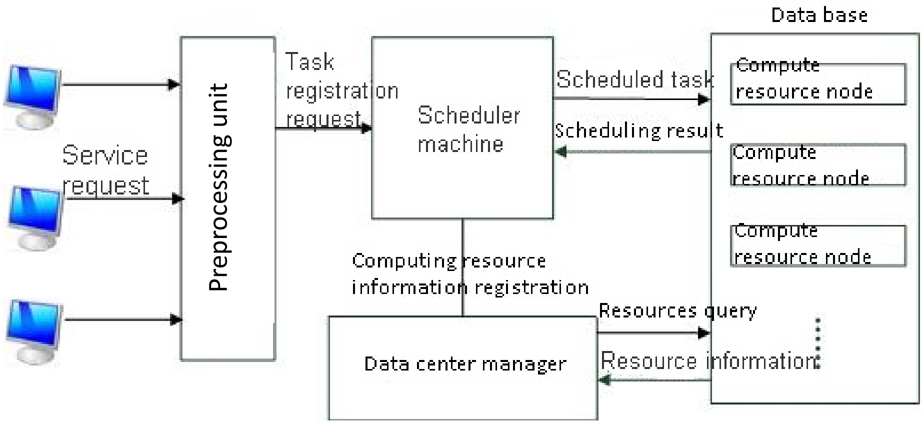

Fig. 1 Cloud computing environment task scheduling model

Users submit service task requests; The prepossessing unit sends the request to the appropriate data center according to the service types of the task (aaS, PaaS, and IaaS) and its QoS requirements. The scheduling manager of the data center selects the scheduling algorithm for task scheduling optimization. According to the feedback information of task execution, the assignment plan is further adjusted. The scheduling algorithm takes the tasks from each server's task request queue and assigns them to the corresponding server. The server performs the tasks and feeds the execution results back to the dispatch center.

3. Design of QoS Perceptual Cloud Mission Scheduling Strategy Based on Dynamic Adaptive Ant Colony Algorithm

In the 1990s, Dorigo Macro et al. discovered a smart Bionic algorithm. They found that ants leave a substance called "pheromone" on the path when they were foraging. The higher the pheromone concentration on the path, the higher the number of ants that choose this path, this shows that the ant can perceive this pheromone and select the path by reference to its concentration. Then, the more ants passed, the more pheromones left on the path, forming a positive feedback mechanism. Dorigo Macro et al. designed an intelligent ant colony algorithm based on this mechanism. This algorithm is robust and suitable for solving NP-hard problems such as TSP problems.

According to the similarities and differences between the TSP problem and the cloud computing task scheduling problem, it is found that the ant colony algorithm can also be well applied to cloud computing task scheduling problems.

The following settings need to be done: (1) Analyze the characteristics of cloud computing task scheduling. According to the interaction between active virtual resources, the network topology is simulated and the pheromones directly act on the virtual resource node. When choosing a path, the ant will refer to the expected value of the path heuristic information in addition to the concentration of the pheromone. When the ant colony algorithm is applied to the task scheduling problem of cloud computing, the expected heuristic information is derived from the load and storage capacity of virtual resource nodes in the cloud environment.

3.1 Design Thinking.

In this paper, the task scheduling design idea of cloud computing: the current load of virtual resource node is taken into account when the task scheduling, and the specific scheduling algorithm uses the adaptive ant colony optimization algorithm to schedule the task submitted by the user to the appropriate computing resources. The task carrying ants in the adaptive ant colony algorithm have priority attributes and QoS requirements from task prepossessing units. The scheduling algorithm in this paper will take these attributes into account when selecting resource nodes, which not only satisfies the user's QoS requirements, but also effectively balances the resource load.

3.2 Algorithm Description.

The flow of this algorithm is as follows:

The first Step: Assign each ant a prioritized task list;

The second Step: Letting the number *n* of ants that have carried the task randomly distributed to the node to start searching;

The third Step: Setting the initial value of the pheromone according to formula (1);

The fourth Step: Judging whether the search condition is satisfied with it, and if it is suitable for it, than perform the following steps;

The fifth Step: Each ant in the ant colony constructs the solution to the next problem according to formula (2) and formula (3), that is, selects the next suitable node;

The sixth Step: Extracting the task with the highest priority from the ant-hosted task list to the selected node, and adding the node to the scheduled list;

The seventh Step: Each ant performs partial pheromone update according to formula (4) and formula (5);

The eighth Step: After the tasks all the ants have been scheduled, the global pheromone update is performed according to formula (6) (7), otherwise it is transferred to the fourth Step to continue execution;

The ninth Step: If the set number of iterations is exceeded, the algorithm execution will end, otherwise it starts again from The second Step.

3.3 Parameter Setting.

(1) Pheromone initial value setting

The initial pheromone value on each node path is often set to the same constant value in the TSP problem. In the algorithm of this paper, because of the different communication capabilities and computing power of the virtual resource nodes, the initial pheromone value of each virtual resource node is set according to its communication capability and computing power.

$$\varphi_j = a \cdot CalAblt_j + b \cdot CommAblt_j \qquad a,b \in [0,1] \tag{1}$$

Means the comprehensive ability calculation function of j node, $CalAblt_j$ means the inherent computing power of the j-node, $CommAblt_j$ and indicates the inherent communication capabilities of the j-node. a, b denotes a weighting factor, a+b=1, and the values of *a* and *b* are set according to different user' task requirements.

(2) Options of the next node

The ant *k* begins from the current node at time *t*, and selects the next target node *j* according to the probability function shown in formula (2) and formula (3):

$$P_j^k(t) = \begin{cases} \dfrac{\left[\tau_j(t)\right]^\alpha \cdot \left[\eta_j(t)\right]^\beta}{\sum\limits_{k \in CL} \left[\tau_k(t)\right]^\alpha \cdot \left[\eta_k(t)\right]^\beta} & k \in CL \\ 0 & otherwise \end{cases} \tag{2}$$

$$\eta_j(t)=\varphi_j \tag{3}$$

The taboo table tabu k (k=1...m) means the scheduled list, *A* represents a list of all available resource nodes, $\tau_j(t)$ and CL=A-tabu k, representing a list of $\eta_i(t)$ select-able resource α and β nodes. It indicates the pheromone concentration remaining on node *j* at time *t*, indicating heuristic information. It is a weighting factor that indicates the degree of emphasis on pheromone concentration and heuristic information.

(3) pheromone update

When a resource node *j* is chosen, its corresponding pheromone concentration will be updated according to formula (4) and formula (5), that is, local pheromone update.

(4)

$$\Delta\tau_j=\phi_j \tag{5}$$

$\Delta\tau_j$ Indicates the pheromone concentration increase value of the node *j* at time t+1. Indicates the pheromone volatilization factor. The larger the value, the faster the pheromone on the node evaporates, otherwise the slower, the value is dynamically adjusted according to formula (8).

When an iterative process is completed, the pheromone update,then the global pheromone update, which is performed on all resource nodes on the selected optimal path according to formula (6) and formula (7).

(6)

$$\Delta\tau_j^{'}=\varphi_j/n \tag{7}$$

$\Delta\tau_j$ indicates the pheromone concentration update increment of each resource ρ node on the optimal path, which is the average comprehensive capability value of all resource ρ nodes on the path, and n is the number of resource nodes.

(4) Adaptive adjustment of pheromone volatilization factor

In the traditional ant colony algorithm, the value of the pheromone volatilization factor is constant. When the problem ρ size is large, the value of the ubiquity affects the global search ability of the algorithm, and the pheromone value of some resource nodes that have never been searched. Decreasing to 0 is not conducive to load balancing of the entire system resources; if the value ρ is too small, the pheromone on the resource node volatilizes too slowly, which will reduce the convergence speed and prolong the total solution time. Therefore, in this paper, the value of the dynamic adaptive adjustment according to the solution condition will be used in the solution process.

$$\rho(t)=\begin{cases}0.95\rho(t-1) & if\,0.95\rho(t-1)>=\rho_{\min}\\ \rho(0)=1 & \rho_{\min}\ else\end{cases} \tag{8}$$

$\rho_{\min}$ Is the minimum value of ρ, It can prevent the convergence rate of the algorithm from ρ being reduced because the value is too small.

4. Experiment

This article employs the cloud computing simulation platform, and Cloud Sim is developed by the University of Melbourne to conduct experiments. The Data center class simulates various hardware resources in the data center and presents them as virtual resources. The Data Center Broker class simulates a user agent, and selects the appropriate virtual resource provider on behalf of the users through the Cloud Information Service class. The Cloud Information Service class provides resource information registration, Search and other functions.

The experimental software and hardware environment configuration: win7 system, Eclipse3.2, JDK1.6.0, 3G memory, CPU clocked at 2.13GHz. Set the number of virtual resource nodes to five; the inherent attribute parameters of the resource nodes are set to two, one is the computing power (in MIPs), and the other is the communication capability (bandwidth), so that the difference in the capability parameter values of each node is widened; The number of tasks is 10, 20, ... 100. The algorithm of this paper and Cloud Sim's own polling scheduling algorithm and standard ant colony algorithm were respectively carried out for 10 experiments and averaged. The population size of the algorithm and the standard ant colony algorithm is m=50, the heuristic factor α=1, and the expected heuristic factor β=2.7. The pheromone volatilization coefficient of the standard ant colony is ρ=0.6, and the ρ value of this paper is dynamically determined. After iteration 100 times, the results of the operation are shown in Figure 2.

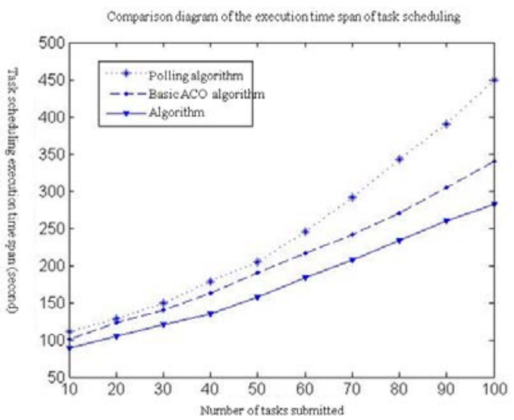

Figure 2 Comparison of three algorithm task execution time

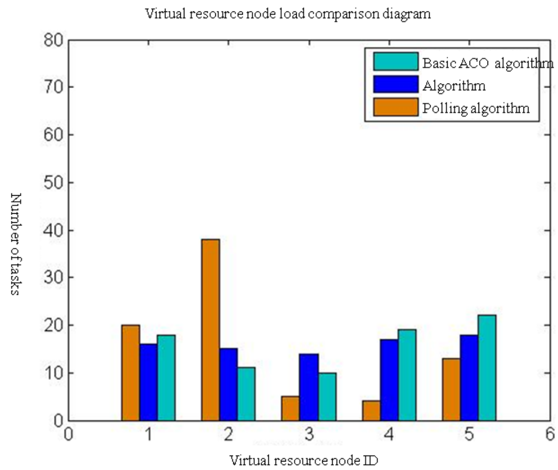

Figure 3 Three algorithms load balance degree relative standard deviation

It can be seen from Fig. 2 that with the increase of the task size, the performance gap between the three algorithms is widened, the performance of the polling algorithm is the worst one, and the basic ant colony algorithm is second. The reason is that the algorithm uses dynamic adaptive adjustment of the volatile factor. The mechanism speeds up the convergence and improves the search efficiency.

Figure 3 shows the load of virtual resource nodes in the three algorithms when the number of tasks is 80. In the polling algorithm, because the tasks are sequentially allocated, the resource load is the most unbalanced; the basic ACO algorithm does not consider the load capacity of the resource nodes (such as storage capacity, computing power, etc.) in the scheduling process, so the load balancing situation fluctuates greatly. The algorithm in this paper considers the comprehensive load capacity of the node in the scheduling, so the resource load is more balanced.

5. Conclusion

We can know that the search efficiency and load balancing of the algorithm in this paper have improved a lot from the experimental results. However, we can also possible to further optimize the values of the heuristic factor α and the expected heuristic factor β, which can further improve search efficiency; In terms of setting QoS, we can bring in more parameters such as storage capacity, task execution cost, and so on.

Acknowledgements

Fund project: science and technology program of institutions of higher learning in Shandong province (J15LN78).

References

[1] Marco Dorigo, Mauro Birattari, Thomas St’’utzle. Ant colony optimization artificial ants as a computational intelligence technique[R].Belgium: IRIDIA, 2006.

[2] ZUO Liyun, ZUO Lifeng. Cloud computing scheduling optimization algorithm based on reservation category[J].Computer Engineering&Design,2012,33(4):1357-1361(in Chinese).

Zuo Liyun, Zuo Lifeng. A scheduling optimization algorithm based on pre-classification in cloud computing[J].Computer Engineering and Design,2012,33(4):1357-1361.

[3] WANG Fang, LI Mei-an, DUAN Wei-jun. Cloud computing task scheduling based on dynamic adaptive ant colony algorithm[J].journal of Computer Applications,2013,33(11) : 3160 -3162,3196.

[4] Deng Jianguang, Zhao Yuelong et al. A cost-driven cloud computing task scheduling strategy[J]. Journal of Jiangsu University: Natural Science Edition, 2014, 35(2): 214-219.

[5] Zha Yinghua, Yang Jingli. Application of improved ant colony algorithm in cloud computing task scheduling[J]. Computer Engineering and Design, 2013, 34(5): 1716-1719.

[6] Lu X, Gu Z. A load-adapative cloud resource scheduling model based on ant colony algorithm [C]. Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on. IEEE, 2011: 296-300.

[7] Tian H, Dong X, Zhao X, et al. A novel SLA-driven verification and planning framework for QoS [C]Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on. IEEE, 2010: 159 -164.

[8] Ding Jian, Chen Zeng, Yuan Zhuo. Fusion of genetic algorithm and ant algorithm [J]. Computer Research and Development, 2003, 9 (40): 1351-1356.

[9] Zhang Jihui, Xu Xinhe. A New Evolutionary Algorithm—— Ant Colony Algorithm[J]. Systems Engineering - Theory & Practice, 1999, 19(3): 84-87.

[10] Wang Ying, Xie Jianying. An Adaptive Ant Colony Algorithm and Its Simulation Research[J]. Journal of System Simulation, 2002, 14(1): 31-33.

[11] Zhang Chunyan, Liu Qinglin, Meng Yu. Cloud computing task assignment based on ant colony algorithm [J]. Computer Applications, 2012, 32(5): 1418-1420.

[12] Wang Jiawei, Lu Zhihui, Wu Jie et al. Analysis of cloud computing technology development and its application[J]. Computer Engineering and Design, 2010, 31(20): 4404-4409.