

Ant colony Algorithm based on Three Constraint Conditions for Cloud Resource Scheduling

Yang Zhaofeng, Fan Aiwan

Computer School, Pingdingshan University, Pingdingshan,
467002 Henan province, China
{ Yang Zhaofeng } pdsnciyang@163.com

Abstract

An improved ant colony algorithm based on the three constraint conditions that aiming at the problem of resource scheduling in cloud computing is proposed in this paper. this method is divided into three steps: Firstly, we describe the state transition probability by using the information heuristic factor and the expected heuristic factor. Secondly, the pheromone update strategy is used to design the scheduling process. Finally, the optimal path is based on the expected execution time, network delay and network bandwidth of three constraints. Experimental results show that the proposed method has faster execution speed than the traditional ant colony algorithm, and the load balancing of the results is more satisfactory.

Keywords: ant colony algorithm, cloud resource scheduling, cloud computing, cloudSim simulation

1. Introduction

Cloud computing is a hotspot of current study, which is to virtualize computing resources around, and with the resource scheduler integration in logic. Users will not need to continue to improve the local computing power and storage of personal resources, the required computing power and data resources will be widely distributed in a large number of distributed computers, simply access to the network will be able to access the application and service [1-3].

At present, there are more researches on user job scheduling of and resource scheduling in cloud computing. For the cloud of job scheduling, Guo proposed the Meta-scheduling model and the Meta-scheduling policy of a cloud environment. The model and job scheduling strategy can ensure that the entire data center's power consumption or carbon dioxide emissions minimum [4]. Alrokayan uses the cloud to expand the cluster to deal with the local cluster resource scarcity, and use some job scheduling strategy to evaluate the performance of the job, analyze the resource cost. The experimental results show that, in the case of heavy load, the Navie scheduling strategy to use the resources of the cloud will generate a lot of cost, while the use of request backfilling and redirection strategy based on the expansion factor to use the resources of the cloud, the cost is relatively low [5]. For the resource scheduling in the cloud, Pop proposed a resource load balancing scheduling algorithm based on trust driven. The algorithm considers the trust requirements of user tasks, takes the resource load balancing as the goal, and takes into account the time span of the task execution time and other factors. Simulation results show that the proposed algorithm can improve the load balance and reduce the relative execution time of the tasks [6]. Beghdad proposed the ant colony optimization algorithm, and is used to solve the problem of the classical traveling salesman problem in computer [7]. Subsequently, a large number of improved ant colony optimization algorithm began to appear, and was applied to the job-shop problem, network routing problem, vehicle routing problem and other problems [8-11]. At present, most of the ants in the ant colony

algorithm communicate mainly through the pheromone exchange, there is no direct communication between the ants. Therefore, the convergence of the algorithm is not high. Taking into account the cloud computing environment and the grid environment has a big difference, the size of the cloud is larger, and the computing power of each node is slightly lower than that of the grid site resources [12-15]. However, the service quality is not lower than the grid computing, so the efficient resource scheduling algorithm is the key to improve the performance of the cloud computing [16-18].

In this paper, we will improve the ant colony algorithm, and in order to better serve the cloud computing, we will propose a resource scheduling method based on the three constraint conditions.

2. Ant Colony Algorithm with Three Constraint Conditions

2.1 Traditional Ant Colony Algorithm

Ant colony optimization algorithm is a kind of swarm intelligence algorithm, which simulates the process of ant foraging [19]. Ants can leave a substance called pheromone on the path of its path in the process of feeding. Each ant searching for food in the sensing range, if not found, it will perceive the strength of this kind of material in a certain range [20-21]. The ants will move to the direction which has the most pheromones. The phenomenon that numerous ants foraging collectively can express as a positive feedback of pheromone. Ant colony algorithm is a random search algorithm, which has been applied in the research of TSP class problems. It uses natural ants routing way of artificial simulation, the main characteristic is to search for the optimal path through the positive feedback, the distributed cooperation [22-25].

Ant System (AS) developed by Dorigo, Maniezzo and Colomi in 1991, was the first ACO algorithm. In AS each ant is initially put on a randomly chosen city and has a memory which stores the partial solution it has constructed so far. Starting from the start point, an ant iteratively moves from city to city. When being at a city i , the ant k chooses to go to a still unvisited city j with a probability given by Eq. (1)

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \notin tabu_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta}, & j \notin tabu_k \\ 0 & , j \in tabu_k \end{cases} \quad (1)$$

Where $\eta_{ij} = 1/d_{ij}$ is a priori available heuristic information (d_{ij} is the distance of city i and city j), α and β are two parameters which determine the relative influence of pheromone and heuristic information, and the $tabu_k$ (the tabu table of the k -th ant) is the set of cities which the ant k has visited.

When the solution construction ends after each ant has completed a tour, the pheromone trails are updated by the rule given by Eq. (2), (3) and (4)

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t+n) \quad (2)$$

$$\Delta\tau_{ij}(t+n) = \sum_{k=1}^m \Delta\tau_{ij}^k(t+n) \quad (3)$$

$$\Delta \tau_{ij}^k(t+n) = \begin{cases} \frac{Q}{L_k}, & \text{if ant } k \text{ uses edge } (i,j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $0 < \rho < 1$ is a evaporation factor of pheromone trails, m is the number of ants in the colony, Q is a const and L_k is the length of the tour which ant k has toured. With this rule, the shorter the tour is, the more pheromone is received by edges belonging to the tour, and it forms a positive feedback process.

2.2 Ant Colony Algorithm with Three Constraint Conditions

The ant colony algorithm is only suitable for a certain size. Although the algorithm is simple, it needs to spend a long time in the implementation process. Due to the number of paths in the tabu list records is few and prone to stagnation phenomenon in the initial stage, namely circles. This is because in the initial search, the differences of the amount of pheromone on each path are not obvious. It is possible that a large number of ants choose a path which is not ideal at the same time, and to the section of the road to bring an increase in the amount of information will lead to false information. Cause a large number of invalid search and reduce the efficiency of the search. This causes the system to appear the phenomenon of stagnation, the time to find the target will be longer. After a long period of time, to make the the pheromone on the better path is significantly higher than other paths. As time goes on, the differences become more and more obvious, which eventually converge. But when the group size is larger, the path number of the table records will be very huge; it is difficult to find a better path from a large number of random paths in a short time, so the running speed of the whole system is slow.

If we put the N ants into P random positions, then the L ants in the process of searching the target location, will determine its transfer direction according to the pheromone concentration in each path, and always moving in the direction of the larger concentration. At the initial stage, due to the difference in the amount of information on each path is small, the ants will select one path randomly. We use the tabu list Tv_l to record the path of the ant of l , and to make the dynamic adjustment depend on the ant movement process, used $q_{ij}^l(t)$ to express the state transition probabilities of the l ant chooses the k city to be the target location at the time of t :

$$q_{ij}^k = \begin{cases} \frac{v_{ij}^\theta(t)\pi_{ij}^g(t)}{\sum_{n \in y_l} v_{in}^\theta(t)\pi_{in}^g(t)} & j \in y_l \\ 0 & j \notin y_n \end{cases} \quad (5)$$

In the formula, y_l is the position that the ant l allows chooser in the next step;

$v_{ij}(t)$ is the left pheromone concentration at position i and j on the path of leaving at the time of t .

$\pi_{ij}(t)$ indicates that the initial information from position i to j , and the information can be obtained from the problem itself.

$\pi_{ij} = \frac{1}{L_{ij}}$ is represented by the position i to position j of a priori value, L_{ij} represents

the position of the distance between the location i to j, L_{ij} is smaller, π_{ij} is the greater, q_{ij} is greater.

θ is heuristic factor for the information which reflects the guiding role of information accumulated in the path of other ants during exercise played, indicates the relative importance of the track. The larger the value is, the ant is likely tend to choose a different path through.

ϑ is the expected heuristic factor, which reflects the degree of attention heuristic information when ants choose the path, indicate the relative weight of the heavy computing power predicted value.

In the actual calculation, if the amount of residual information on the path is not processed, the more and more information will be submerged by the path of the ant search process. we should use a certain strategy to adjust the pheromone after each ant has gone through a path or the completion of the traversal of all the p positions, and as the time goes on, it will be lower and lower, we use the following rules to adjust the amount of information on the path of (i, j) at the time of t+p:

$$v_{ij}(t+p) = (1-\eta)v_{ij}(t) + \Delta v_{ij}(t) \quad (6)$$

$$\Delta v_{ij}(t) = \sum_{l=1}^m \Delta v_{ij}^l(t) \quad (7)$$

In the formula, η represents the volatile coefficient of pheromone, then $1-\eta$ is the residual factor. In order to prevent the infinite accumulation of information, we set the range of η is from 0 to 1 in the closed interval.

$\Delta v_{ij}(t)$ indicates that the concentration of the residual information left on the path of i to j in the process of the time of t to t+p, amely, the pheromone increment on the path of (i,j) in this cycle.

In this paper, the update strategy of pheromone is adopted as follows:

$$\Delta v_{ij}(t) = \begin{cases} \frac{R}{D_l} & \text{across}(i, j) \\ 0 & \text{not across}(i, j) \end{cases} \quad (8)$$

In the formula, R represents the pheromone intensity, which affects the convergence rate of the algorithm to a certain extent, and D_l indicates the total length of the path of the ant l in the cycle.

2.3. Ant Colony Algorithm with Three Constraint Conditions for Cloud Resource Scheduling

The typical resource scheduling model is showed in Figure 1.

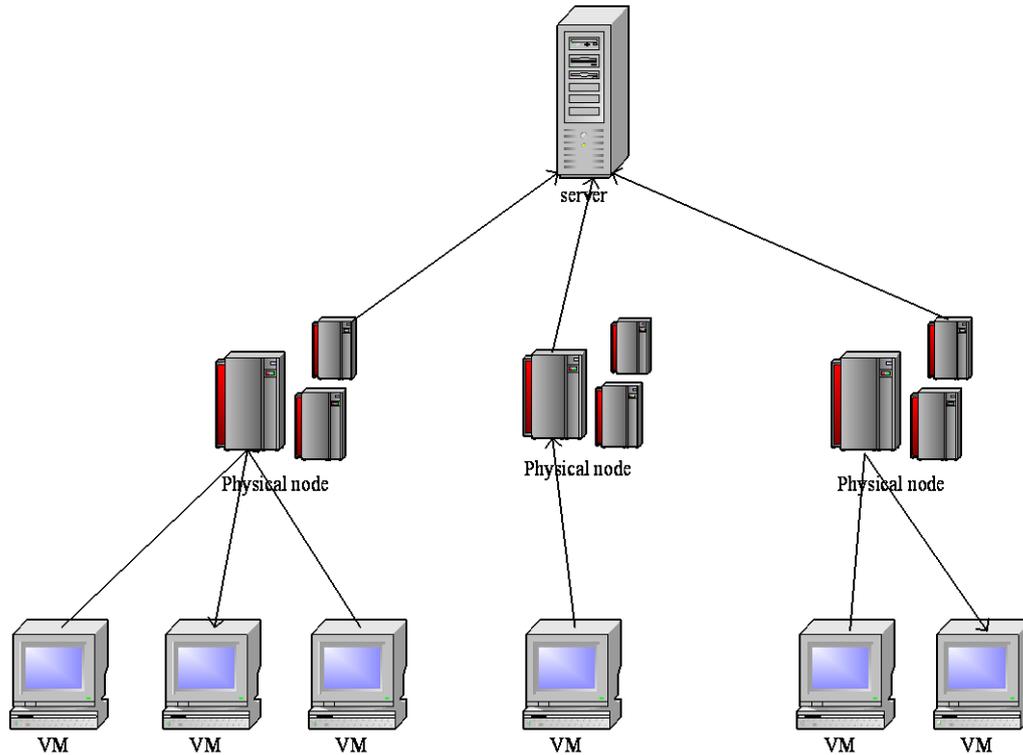


Figure 1. Resource Scheduling Model

$T = \{t_1, t_2, \dots, t_n\}$ represents the tasks waiting to be scheduled per unit of time, n is the number of tasks. $N = \{n_1, n_2, \dots, n_m\}$ represents the set of nodes in the cloud system, assuming that the cloud system has m nodes. For cloud computing system, n_i represents the computing resources on the n_i ; For cloud storage system, n_i represents the data on a n_i . $V = [v_1, v_2, \dots, v_n]$ represents the task scheduling vectors or a scheduling scheme. For cloud storage system, v_i represents the i -th task of data is provided by resources nodes that represented by v_i value, and the length of the vector is the total amount of scheduling tasks per unit time.

For example, a task scheduling vector [5, 1, 3, 2, 1, 6], the length of this vector is 6, and represents needs to schedule task number is 6 per unit of time. The value based on the position of No. 1 is 5, represent the data of the task 1 is provided by the system node 5.

So, the data of task 2 and 5 are provided by the node 1; the data of task 3 is provided by the node 3; the data of task 4 is provided by the node 2; the data of task 6 is provided by the node 6. For cloud computing system, it is on behalf of a task placed in a node.

Referring to the Map/Reduce cloud computing architecture model proposed by Google, each element in the cloud environment is composed of two parts: The first is a single Master JobTracker, the second is a Slave TaskTracker under the jurisdiction of each node of the cluster. The Master JobTracker is responsible for scheduling the tasks that constitute a job. The data resources of these tasks are distributed among the user image slice of different node storage resources. The output data of the Map is written to the local disk, and the JobTracker is notified when the machine is finished. Then, the Reduce will sent the HTTP request after it get the notices of JobTracker, and the JobTracker request is copied from the corresponding Map end to the output. Such a way can only wait for the

Map to complete the task to start the Reduce task, and the execution of the Map task and the Reduce task is separated.

An improved idea is that the Map task and Reduce task can be executed in a pipeline, That is, the Map task begins to produce the output directly to the corresponding Reduce task. This requires JobTracker to assign the corresponding Map and Reduce tasks, and the position of each Map task is sent to the Reduce task after the user submits the job. After each Map output Mapper decides to send the partition (Reduce task) and send it directly through the appropriate Socket.

Reduce task receive pipeline data received from each of the Map task and stores it in the memory buffer, and write to the buffer zone that has been sorted out to disk when it needed. Once the Reduce task is learned that each Map task has been completed, It performs the final merge of sorted content, then calls the user-defined Reduce function, and write output to HDFS.

When looking for a suitable computing node, first, it starts from the node to calculate its own resources. If the remaining resources are sufficient to meet the amount of jobs submitted by the user, then priority in the allocation of their own resources. If the resource is not enough to calculate the minimum amount of resources allocated to the user, then according to the search algorithm starts searching for other suitable cloud computing resources. The Ant colony optimization allocation algorithm proposed in this paper is realized in this part. In order to avoid stagnation search, the search will be carried out within a certain range, the purpose is to reduce the algorithm itself brings network overhead. If there is still no appropriate resources in all of the sub nodes, then the node report to ask for instructions to the main scheduling node removed user data mirroring the nodes in the cluster partition.

The slave node domain as an undirected graph $G(V, E)$, In which V is the set of all slave nodes in regional Area, E is the network set which connect to each slave node. In the cloud computing network, it is divided into several sub regions, and then the same number of ants are assigned to each region. Each group of ants search only in their own region to find a suitable computing node, that is, in the E to find an optimal path. The following parameters should be considered in the metric:

- (1) Expect the execution time, $ET(f)$ refers to the path of f at the end of the computing resources to cope with this job takes time;
- (2) Network latency: $ND(f)$ refers to the maximum network delay generated by path f ;
- (3) Network bandwidth: $BW(f)$ refers to the maximum bandwidth provided by the path f .

Assuming the feature set of a virtual machine resource VM_i is:

$$V_i = \{v_{i1}, v_{i2}, v_{i3}\} \quad (9)$$

In the formula, v_{i1} is the CPU feature, x_{i2} is the memory feature, x_{i3} is the bandwidth feature.

User demand for the diversity of cloud computing resources and preferences, how to make the QoS guarantee. QoS description of the task can usually be used to complete the task of time, memory, network bandwidth and other parameters to quantify the QoS. For example, task completion time QoS as the evaluation criteria, it include the start time of the task, the completion time, the end time, etc., can be selected to complete the task of all time as the evaluation index.

Usually the general expectation vector of the class i task can be described as:

$$X_i = \{x_{i1}, x_{i2}, x_{i3}\} \quad (10)$$

I In the formula, v_{i1} is the CPU feature, x_{i2} is the memory feature, x_{i3} is the bandwidth feature and at the same time to meet: $\sum_{j=1}^3 x_{ij} = 1$.

The constraint function for resource selection is:

$$Y(x) = \frac{\alpha ET(f) + \gamma ND(f)}{\beta BW(f)} \quad (11)$$

$$s.t. \begin{cases} ET(f) < T_1 \\ BW(f) > T_2 \\ ND(f) < T_3 \end{cases} \quad (12)$$

Ultimately, the process of determining the target resource and path is the process of finding the shortest path and the path and the resource of $Y(x)$. α, β, γ are the weight of three constraint conditions. T_1, T_2, T_3 are the boundary constraints, which meet the QoS constraints in cloud computing environment.

3. Experiment Result and Analysis

In order to verify the validity of the method in this paper, we have carried out experiments in the next work. The main platform of the experiment chose the CloudSim simulation platform. The detrusion of CloudSim simulation platform will help accelerate the algorithm of cloud computing, the development of methods and specifications. Due to a large number of computing nodes need to be deployed in the actual environment, it is more difficult for the general scientific research institutions to achieve, but it can be achieved through simulation platform to simulate the actual operating environment to achieve the purpose of our study. CloudSim component tools are open sources and free of charge. The framework and architecture components of CloudSim software structure including SimJava, GridSim, CloudSim, UserCode four levels.

Compared with the grid computing, the most prominent feature of cloud computing is the use of sophisticated virtualization technology. All of the resources of the data center virtualize into a pool of resources, unified to provide services to the external users CloudSim embodies this feature. CloudSim provides a data center based virtual technology, modeling and simulation capabilities and resource monitoring, host to virtual machine mapping capabilities. Typically, a host of data centers can be mapped to a number of virtual machines based on the needs of users.

In addition to the software platform of CloudSim, the other configuration of the experiment as follows: the main frequency 2.0GHz of the dual core CPU, the size of the 8GB memory, 500GB hard drive, Windows7.0 operating system.

According to the ant colony algorithm and scheduling work flow, in this experiment, we set the number of tasks from 40 to 200, and the number of nodes is 8. In order to show the difference, we have to set the node's QoS property gap larger, mainly including CPU, memory and network bandwidth. At the same time, select the traditional ant colony algorithm and three constrained conditions ant colony algorithm is proposed in this paper, in order to form a comparison on the experimental results. The two algorithms are executed 5 times and the average value is taken. The time spent by the task execution is shown in Figure 2.

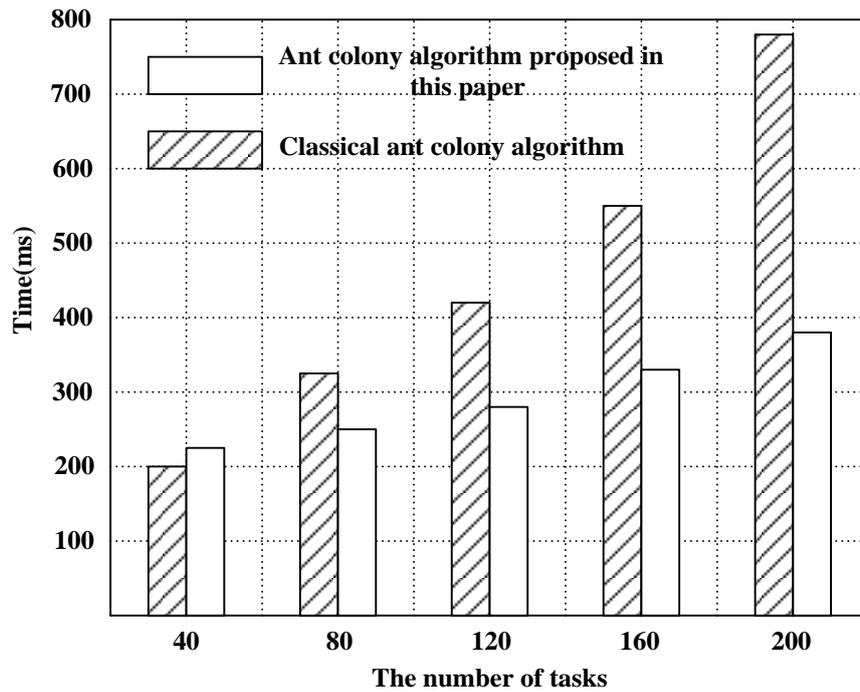


Figure 2. Comparison of the Execution Time of the Two Algorithms

From the graph, we can see that the three constraint conditions of ant colony algorithm, its execution time is significantly lower than the traditional ant colony algorithm in the implementation of cloud computing resource scheduling. In addition, with the increase of the amount of the task, the advantage of this kind of speed is more obvious.

Below, we examine the load balance of each node after the completion of the two algorithms. The results are shown in Table 1.

Table 1. Load Balance of Two Algorithms

Node	40 Tasks		80Tasks		120Tasks		160Tasks		200Tasks	
	CA	IA	CA	IA	CA	IA	CA	IA	CA	IA
1	0	0	2	4	8	10	12	14	18	20
2	0	6	4	10	10	16	14	20	20	26
3	4	6	10	10	16	16	20	20	26	26
4	6	10	10	14	16	20	20	24	26	30
5	4	4	8	10	10	14	16	20	20	24
6	8	4	12	8	16	12	22	18	26	22
7	8	4	18	10	22	14	26	20	32	24
8	10	6	16	14	22	18	30	24	32	28

In Table 1, CA represents the classical ant colony algorithm, and IA represents the three constraint conditions proposed in this paper. From the data in Table 1, we can see that the load on each node is more balanced, and the equilibrium degree can be seen from the following deviation.

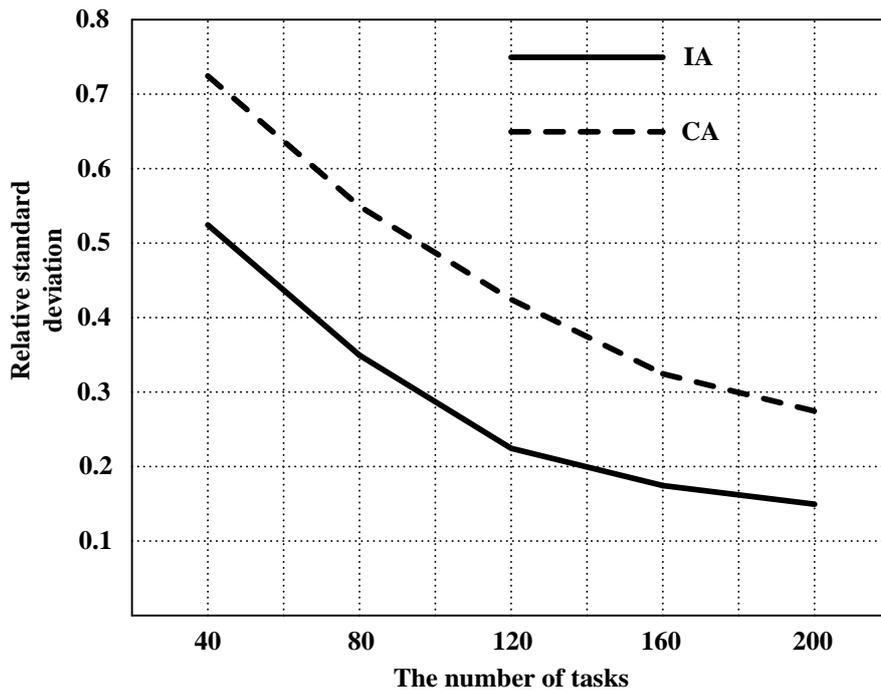


Figure 3. Load Balance of the Two Algorithms

The three constraint conditions of ant colony algorithm, which can be seen in Figure 3, is presented in this paper, which is obviously lower than that of the traditional ant colony algorithm. Also, with the increase of the amount of the task, the degree of the reduction is more obvious.

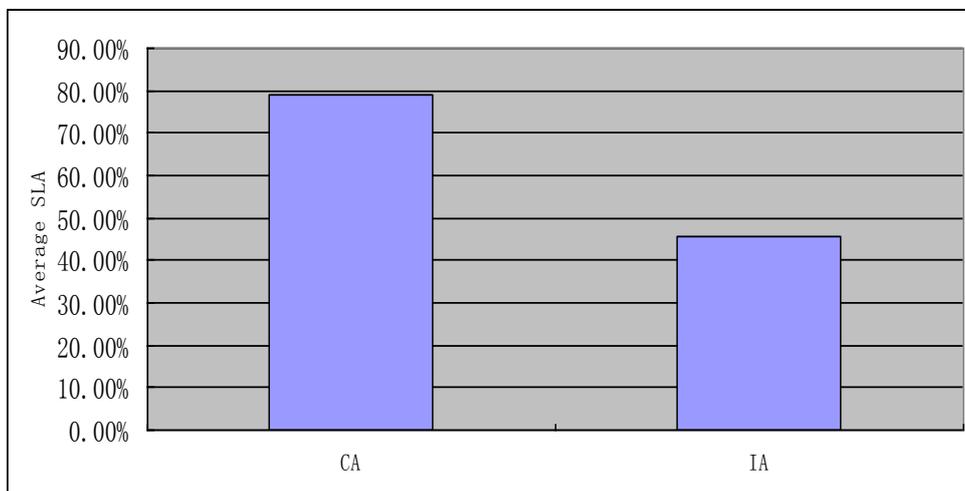


Figure 4. Average SLA of Three Algorithms

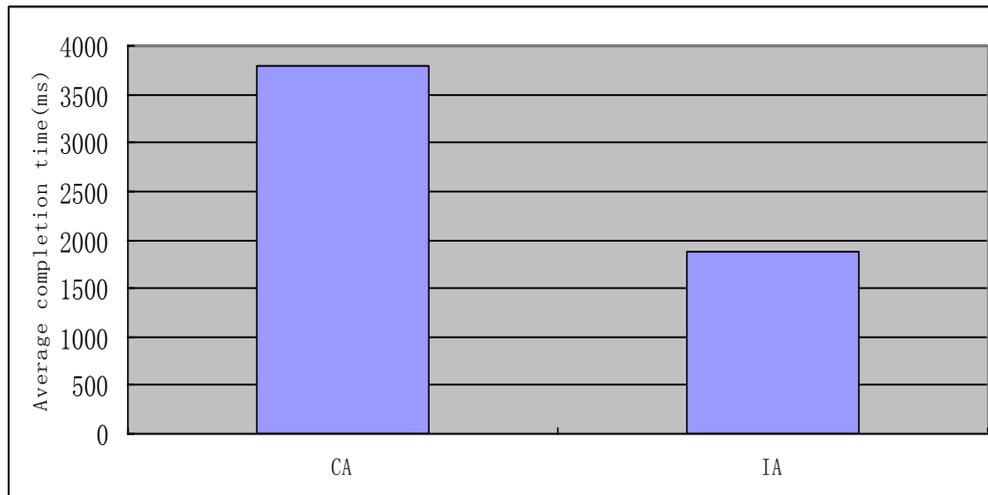


Figure 5. Average Completion Time of Three Algorithms

From Figure 4, 5, we can see this IA algorithm is maintaining lower average SLA and shortest average completion time relative to CA algorithm.

4. Conclusions

In cloud computing, resource scheduling is one of the core issues. In this paper, based on the ant colony algorithm, a kind of ant colony algorithm based on three constraint conditions is constructed, which is used for resource scheduling in cloud computing. With the help of the information elicitation factor and expect inspiration factor, with the help of the pheromone intensity design pheromone update strategy, with the help of three constraints to complete the optimization process. Firstly, we describe the state transition probability by using the information heuristic factor and the expected heuristic factor. Secondly, the pheromone update strategy is used to design the scheduling process. Finally, the optimal path is based on the expected execution time, network delay and network bandwidth of three constraints. Experimental results show that the three constraint conditions ant colony scheduling algorithm can complete the resource scheduling in a shorter time, and the load distribution of the scheduling results is more balanced.

Appendix

This article is a revised and expanded version of a paper entitled Application of Ant colony Algorithm in Cloud Resource Scheduling Based on Three Constraint Conditions presented at The 5th International Conference on Computer Science and Technology April 22-23, 2016 at Suzhou Youngor Leidisen Hotel, Suzhou, China.

Acknowledgments

The authors thank all the reviewers and editors for their valuable comments and work.

References

- [1] Kumar Dinesh, Raza Zahid. A PSO based VM resource scheduling model for cloud computing[C]. Proceedings-2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT, (2015), pp.213-219.
- [2] Sousa, Marcelo Portela., Kumar, Ajey., Lopes, Rafael F., Lopes, Waslon T.A., Alencar, Marcelo Sampaio. Cooperative Space-Time Block Codes for Wireless Video Sensor Networks. Wireless Personal Communications, (2012), pp.1-15.
- [3] Wang Dewen, Liu Xiaomeng. A resource scheduling strategy for cloud computing platform of power system simulation based on dynamic migration of virtual machine[J]. Dianli Xitong Zidong hua/Automation of Electric Power Systems, (2015), vol.39, no.12, pp. 97-105.
- [4] Paoli, Roberto., Fernández-Luque, Francisco J., Doménech, Ginés., Martínez, Félix., Zapata, Juan., Ruiz, Ramón.A system for ubiquitous fall monitoring at home via a wireless sensor network and a wearable mote. Expert Systems with Applications, (2012), vol.39, no.5, pp.5566-5575.
- [5] Guo Fengyu, Yu Long, Tian Shengwei, Yu Jiong. A workflow task scheduling algorithm based on the resources fuzzy clustering in cloud computing environment[J]. International Journal of Communication Systems, (2015), vol.28, no.6, pp.1053-1067.
- [6] Moschakis, Ioannis A., Karatza, Helen D. Evaluation of gang scheduling performance and cost in a cloud computing system. Journal of Supercomputing, (2012), vol.59, no2, :pp.975-992.
- [7] Alrokayan Mohammed, Vahid Dastjerdi Amir, Buyya Rajkumar. SLA-aware provisioning and scheduling of cloud resources for big data analytics[C]. 2014 IEEE International Conference on Cloud Computing in Emerging Markets, (2015), no.19, pp.665-673.
- [8] Al-Ameen, Mahdi Nasrullah., Hasan, Md. Rakib. The mechanisms to decide on caching a packet on its way of transmission to a faulty node in wireless sensor networks based on the analytical models and mathematical evaluations. Proceedings of the 3rd International Conference on Sensing Technology, ICST 2008, (2008), pp.336-341.
- [9] Pop Florin, Dobre Ciprian, Cristea Valentin, Bessis Nik, and et.al. Deadline scheduling for aperiodic tasks in inter-cloud environments: a new approach to resource management[J]. Journal of Super computing, (2015), vol.71, no.5, pp.1754-1765.
- [10] Misra, Satyajayant., Hong, Seung Don., Xue, Guoliang., Tang, Jian. Constrained relay node placement in wireless sensor networks: Formulation and approximations. IEEE/ACM Transactions on Networking, 2010, (2010), vol.18, no.2, pp.434-447.
- [11] Beghdad Bey Kadda, Benhammadi Farid, Sebbak Faouzi, Mataoui Mhamed. New tasks scheduling strategy for resources allocation in cloud computing environment[C]. 6th International Conference on Modeling, Simulation, and Applied Optimization, ICMSAO 2015-Dedicated to the memory of late Ibrahim El-Sadek, (2015), no.7, pp.768-777.
- [12] Ahuja, Ravin., De, Asok., Gabrani, Goldie. SLA based scheduler for cloud for storage & computational services. Proceedings of 2011 International Conference on Computational Science and Its Applications, ICCSA 2011, (2011), pp. 258-262.
- [13] Chen Niansheng, Fang Xiaoping, Wang Xin. A cloud computing resource scheduling scheme based on estimation of distribution algorithm[C]. 2014 2nd International Conference on Systems and Informatics, ICSAI, (2014), pp.304-308.
- [14] Assuncao M, Costanzo A, Buyya R. Evaluating the cost benefit of using cloud computing to extend the capacity of clusters. Proc of the 18th ACM International Symposium on High Performance Distributed Computing, Munich, Germany, (2009), pp.141-150.
- [15] Shin Saemi, Kim Yena, Lee Sukyoung. Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing[J]. 2015 12th Annual IEEE Consumer Communications and Networking Conference CCNC, (2015), pp.814-819.
- [16] Moschakis, Ioannis A., Karatza, Helen D. Evaluation of gang scheduling performance and cost in a cloud computing system. Journal of Supercomputing, (2012), vol.59, no.2, pp.975-992.
- [17] Ai, Lifeng., Tang, Maolin., Fidge, Colin. Resource allocation and scheduling of multiple composite web services in cloud computing using cooperative co-evolution genetic algorithm. Lecture Notes in Computer Science, (2011), pp.258-267.
- [18] Devi K.Vimala, Mehata K.M. Modelling resource estimation and scheduling for processing in cloud computing[C]. 2013 5th International Conference on Advanced Computing, ICoAC 2013, (2014), no.12, pp. 325-330.
- [19] Song, Xiangqian.,Gao, Lin., Wang, Jieping. Job scheduling based on ant colony optimization in cloud computing. 2011 International Conference on Computer Science and Service System, CSSS 2011, (2011), pp.3309-3312.
- [20] Uskenbayeva R.K, Kuandykov A.A, Cho Y.I., Kalpeyeva Zh.B. Tasks scheduling and resource allocation in distributed cloud environments[C]. International Conference on Control, Antomation and Systems, (2014), no.16, pp.1373-1376.
- [21] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, (1997), vol.1, no. 1, pp. 53-66.

- [22] Fang, Yadong., Wang, Fang., Wang, Hui. Research of multi-objective optimization study for job shop scheduling problem based on grey ant colony algorithm. *Advanced Materials Research*, (2011), pp.1033-1036.
- [23] Mazloumi, Ehsan., Mesbah, Mahmoud., Ceder, Avi., Moridpour, Sara., Currie, Graham. Efficient transit schedule design of timing points: a comparison of ant colony and genetic algorithms. *Transportation Research Part B: Methodological*, (2012), vol.46, no.1,;pp.217-234.
- [24] Mirabi, Mohammad. Ant colony optimization technique for the sequence-dependent flowshop scheduling problem. *International Journal of Advanced Manufacturing Technology*, (2011), vol.55, no.1, pp. 317-326.
- [25] Yang Zhaofeng, Fan Aiwan. Application of Ant colony Algorithm in Cloud Resource Scheduling Based on Three Constraint Conditions[C].The 5th International Conference on Computer Science and Technology April 22-23, 2016 at Suzhou Youngor Leidisen Hotel, Suzhou, China.

Authors



Yang Zhaofeng, an associate professor in Computer School of Pingdingshan University, who was born in 1978 in Xiangcheng County, Henan Province, China. He received the M.S. degree of Computer Application Technology and Network Security in Xi'an Electronic and Science University in 2010. He is mainly engaged in the research of network information security and cloud computing security.



Fan Aiwan, an associate professor in Computer School of Pingdingshan University, who was born in 1978 in Neixiang County, Henan Province, China. He received the M.S. degree of Computer Application Technology and Network Security in Xi'an Electronic and Science University in 2009. He is mainly engaged in the research of network information security and cloud computing security.