

The Intelligent Task Scheduling Algorithm in Cloud Computing with Multistage Optimization

XiaoLi He¹, Yu Song² and Ralf Volker Binsack³

¹*School of Computer Science, Sichuan University of Science and Engineering, Zigong Sichuan 643000, China*

²*Department of Network Information Management Center, Sichuan University of Science and Engineering, Zigong Sichuan 643000, China*

³*Winterstein 14, Niddatal, 61194, Germany*

Abstract

There're huge numbers of users and various tasks need to be handled in the cloud computing environment, the high effective task scheduling algorithm is one of the crucial problems that the cloud computing need to solve. Aiming to the model structure of cloud computing, in this article it introduces the Particle Swarm Optimization algorithm (PSO) and Ant Colony Optimization algorithm (ACO) to combine with optimized task scheduling algorithm. First it takes the particle swarm optimization algorithm to generate the initial scheduling results, and introduces the random inertia weight to improve the scheduling ability of the algorithm, then to take the generated results of improved particle swarm optimization algorithm as the initial pheromones of the ant colony algorithm to find out the optimal scheduling scheme, and use the elitist strategy and crossover operator in the genetic algorithm to improve the ant colony algorithm, among the algorithms to use multistage optimization algorithm to improve the operating efficiency. The experimental results show that under the same conditions, the total task completion time of improved algorithm has been reduced and its performance advantage are getting more obvious with the increased task measures.

Keywords: *Cloud computing; Task scheduling; Improved ant colony algorithm; Improved particle swarm optimization algorithm; Total task completion time*

1. Introduction

The cloud computing is such a computing model with rapid growth in recent years; it was rising with the technological development of distributed computing, grid computing and parallel computation. Cloud computing is a kind of computing model, which can obtain the resources quickly from the configurable computing resources sharing pool in real time according to the demand, those resources includes the server, storage, network, service and application, *etc*; the supply and release of resources can be finished in just shorter a time, so as to reduce the load of resource management and the interaction of service providers to a minimum [1]. The basic principles of cloud computing is that to break down the tasks reported by massive users into smaller tasks via the network, by using multiple computers connected in the network to search, compute and combine the results and then send them back to the users. The users quantity cloud computing shall face to is just huge and this causes the very heavy workload, thus the computing resource allocation and task debugging are all the very crucial nodes [2]; at present the more studied first-in-first-out scheduling algorithm, computing capacity scheduling algorithm and fair scheduling algorithm *etc.*, all have the certain deficiency [3]. In addition, the cloud computing task scheduling based on immune algorithm, ant colony algorithm and genetic algorithm have been turned into the research hotspots [4-6], but any algorithm shall have its own unique advantages and intrinsic disadvantages: for example, the genetic

algorithm has the global searching capacity and fast randomness, but it has too much process parameter setting and its realization algorithm is generally quite complicated, and it's quite easy falling into the local optimum; the ant colony algorithm has the better optimization capacity, but because of the less pheromone at the very beginning it may cause the lower convergence rate; however the particle swarm optimization algorithm has the quite fast convergence rate and better performance than the genetic algorithm; even its convergence rate is quite fast in the initial period of algorithm, at the later period the convergence rate is quite slower and the local searching capacity is also quite weak. Therefore, in this article it synthesizes the particle swarm optimization algorithm and ant colony algorithm that with better performances, to use these two algorithms in the cloud computing task scheduling model, first to utilize particle swarm optimization algorithm to generate initial scheduling results, make the scheduling results generated by particle swarm optimization algorithm as initial pheromone of ant colony algorithm, then use the strong optimization capacity of ant colony algorithm to seek the optimal scheduling scheme, meanwhile to make the targeted improvement on above two algorithms, thus to ensure the forming of the optimum scheduling scheme, promote the performance of cloud computing task scheduling algorithm.

2. The Brief Introduction of Cloud Computing Task Scheduling

In the cloud computing model, the computation of a large task shall need to break down the big tasks into many sub tasks first, according to certain principles the system shall assign the different nodes for completion of different sub tasks; while the operation of the sub tasks on each node were finished, it shall combine the output results from each nodes and output to the users, then to finish the computation of such large task [7]. Now in the cloud computing model, the Map-Reduce model [8] from Google Inc is generally used accordingly. In the Map-Reduce Model, the computation procedures are divided into the map procedure and reduce procedure; among them the map procedure is to break down the large task into smaller tasks, to make the parallel computation on assigned task server for the result output, the reduce procedure shall combine the output results from map procedure and handle them accordingly, feedback to the users, the implementation of above procedures can be shown in attached Figure 1.

The running procedure of the Map-Reduce Model can be expressed through mathematical form as following: to allocate the n sub tasks that can be independently run to the m ($m < n$) task servers for operation, it assumes that the sub task set is $T = \{t_1, t_2, \dots, t_n\}$, the task server set is $VM = \{vm_1, vm_2, \dots, vm_m\}$, according to the allocation principles of sub tasks, the sub task t_j ($j = 1, 2, \dots, n$) can be only run in any task server vm_i ($i = 1, 2, \dots, m$) among them, then the mapping relationship X between the T and VM shall be shown as following formula (1)

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

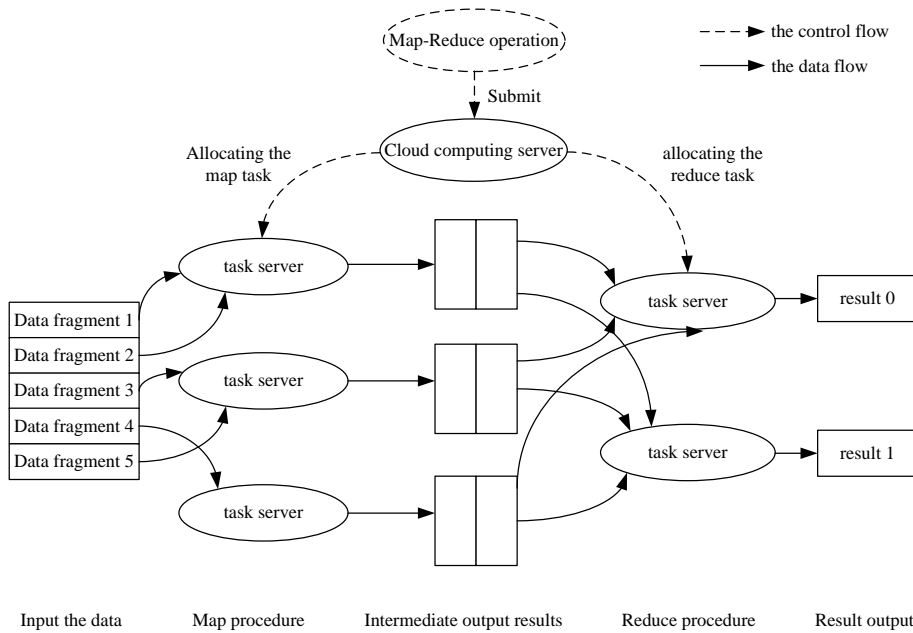


Figure 1. Running Process of Map-Reduce Model

Among them, x_{ij} represents whether the sub task t_j has been allocated to the task server vm_i accordingly, thus $x_{ij} \in \{0,1\}$ and there's the $\sum_{i=1}^m x_{ij} = 1, j \in \{1,2,\dots,n\}$, the value of x_{ij} shall be arranged as 1 or 0 according to that if the t_j has been allocated to the vm_i . It assumes that ET_{ij} is the expected runtime of the sub task t_j on the vm_i , similar to the mapping relationship x , it can get the runtime matrix ET as per formula (2) shown.

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

It assumes that the b_j is the start time for the sub task t_j running on the task server vm_i , then the expected completion time CT_{ij} for the sub task t_j shall be $b_j + ET_{ij}$, so T_{max} - the final completion time of each sub task should be $\max(CT_{ij})$, the target of cloud computing task scheduling model is to get the minimum T_{max} .

3. The Design for Improved Scheduling Algorithm

To improve the task scheduling algorithm, it mainly use the improved particle swarm optimization algorithm and the improved ant colony algorithm to make the combining optimization on the cloud computing task scheduling algorithm, to generate the better scheduling scheme.

3.1. The Design for Improved Particle Swarm Optimization Algorithm

The particle swarm optimization algorithm [8] is inspired by the group behavior of fishes and birds then developed into such a kind of evolutionary optimization algorithm to simulate the animal's group activity phenomenon, its basic principles are that: the system initially generates a set of random legal solutions, through iteration each particle shall track the individual optimal solution and the global optimal solutions of the whole population then to update the speed and position of particle continuously, the termination condition for the iteration is to reach the maximum iterations or to generate the optimal solutions within the error allowable range.

In cloud computing environment, according to the characteristic of task scheduling, it may need to adjust and set the speed and position for the particles, from formula (1), it can get the speed v_i of the i th particle as per shown in formula (3).

$$V_i = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix} \quad (3)$$

Among the formula, $v_{ij} \in \{-v_{max}, v_{max}\}$, then the position of the i th particle can be shown in formula (4) :

$$X_i = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad (4)$$

Among the formula, $x_{ij} \in \{0,1\}$, and there's $\sum_{i=1}^m x_{ij} = 1, j \in \{1,2,\dots,n\}$.

Through above formulas (3), (4) to complete the scheduling algorithm definition on the discrete particle group, the position of particles shall be limited as 0 or 1, the speed of particle then represents the probability for particle getting its position value, to achieve such effect, it defines the fuzzy function $sig(v_{ij}^{k+1})$ as per shown in formula (5):

$$sig(v_{ij}^{k+1}) = 1 / (1 + exp(-v_{ij}^{k+1})) \quad (5)$$

The iterative formula for the speed and position of particle are shown in formula (6) and (7):

$$v_{ij}^{k+1} = \omega \cdot v_{ij}^k + c_1 r_1 (p_{ij}^k - x_{ij}^k) + c_2 r_2 (p_{gj}^k - x_{ij}^k) \quad (6)$$

$$x_{ij}^{k+1} = \begin{cases} 1 & r_3 < sig(v_{ij}^{k+1}) \\ 0 & \text{ot her} \end{cases} \quad (7)$$

Among the formulas, p_{ij}^k , p_{gj}^k represents the individual optimum position and the optimum position of the population respectively, they're both the j dimensional vectors, k is the number of iterations, c_1 , c_2 are the learning factors, through previous research findings, it has certified that while $c_1 = c_2 = 2$ the effect could be better, r_1 , r_2 , r_3 are the random numbers uniformly distributed in [0,1] which are independent of each other, ω represents the inertia weight.

In standard particle swarm optimization algorithm, ω shall be changeless, thus It may cause the limited adjustment capacity for the algorithm [10], and then the particle swarm optimization algorithm shall be easily fell into local optimal solution; under such situation, ω shall have quite smaller influence on the adjustment of research, in order to

change such situation, it sets the ω as random number, the calculation formula can be shown in following formula (8).

$$\omega = \mu_{\min} + (\mu_{\max} - \mu_{\min}) \cdot \xi + \sigma \cdot \delta \quad (8)$$

In the formula, μ_{\max} 、 μ_{\min} are the maximum and minimum value of the ω respectively, σ is the variance of ω , the ξ 、 δ are the random numbers, they obey the uniformly distributed random numbers and the normal distribution in the interval [0,1]. Thus, the ω value shall be turned from constant into random value, this shall have the two advantages: the one is that the randomness of ω is similar to the mutation operator in the genetic algorithm, it shall have the positive significance to keep the particle swarm in diversity; the other one is that the mean value of the ω can be self-adaptively changed, the influence between the speed among different iterations shall be changed randomly, thus can adjust the searching capacity for the global optimal solution with more flexibility.

Though the ω was treated with randomization, it shall be still kept in the interval $[-v_{\max}, v_{\max}]$, and it shall also be treated as per following formula (9) shown:

$$v_{ij}^{k+1} = \begin{cases} -v_{\max} & v_{ij}^{k+1} < -v_{\max} \\ v_{\max} & v_{ij}^{k+1} > v_{\max} \\ v_{ij}^{k+1} & \text{ot her} \end{cases} \quad (9)$$

3.2. The Improved Ant Colony Algorithm

While finding foods the ants shall release the pheromone along the path, the more pheromone on the path, the higher possibility the ants may choose to follow, then to release even more pheromone, thus it makes a virtuous cycle accordingly, the ant colony algorithm is such an optimization algorithm to simulate the foraging behavior of ant colony [11]. By using the ant colony algorithm in the cloud computing task scheduling algorithm, according to the mapping relationship matrix x shown in formula (1), it generates the node set $\{x_{ij}\}_{m \times n}$, by using ant colony algorithm to generate the nodes $\{x_{y_1, 1}, x_{y_2, 2}, \dots, x_{y_j, j}, \dots, x_{y_n, n}\}$ from the node set, among them $y_j \in \{1, 2, \dots, m\}$, the to complete the minimum T_{\max} .

The ant colony algorithm can be mainly divided into three procedures: (1) initiation; (2) pheromone updating; (3) solving the optimal solution. While in initiation procedure, it shall reinforce the results generated by the particle swarm optimization algorithm, so as to accelerate the convergence rate of the ant colony algorithm and then get the optimal solution as sooner as possible, though the experiment it was tested that the better effect shall be made while the reinforcing times is 5.5, in scheduling algorithm it uses the reinforcing times as 5.5.

Before the updates of pheromone, the ant will follow on the path accordingly, it assumes that at the t time the transition probability of the k th ant climbing to the node x_{ij} can be shown in formula (10).

$$v_{ij}^{k+1} = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{x_{ik} \notin tabu_k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta} & x_{ij} \notin tabu_k \\ 0 & \text{ot her} \end{cases} \quad (10)$$

Among the formula, the $\eta_{ij}(t)$ represents the inspiring information, the $\tau_{ij}(t)$ represents the pheromone on node x_{ij} , the $tabu_k$ ($k = 1, 2, \dots, m$) is the taboo list of the k th ant, after the task was break down the n sub tasks shall match with the node $\{x_{ij}\}_{i=1}^m$ respectively, while the k th ant selected the node, such node need to be added into $tabu_k$ accordingly. And α 、 β represent respectively the weight of information content and the inspiring information. After that to update the pheromone, allocating the sub tasks is equivalent to that the ant walking into the corresponding node. And the pheromone shall be updated accordingly, it shall be shown as formula (11).

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (11)$$

Among the formula, the $\Delta \tau_{ij}^k(t)$ represents the information content released by the k th ant on the path (i, j) in this iteration, while the ant selecting x_{ij} it shall be $\frac{P}{C \tau^k(t)}$, P is the constant, while the ant not selecting x_{ij} it shall be 0, $1 - \rho$ is the residual pheromone proportion, ρ represents the volatility coefficient of the pheromone and there's $0 \leq \rho < 1$.

The genetic algorithm is quite easily fallen into local extremum point [12], and brings negative influence on the model; in order to solve such problem, it introduces the elitist strategy and crossover operator to make the improvement on the updating principle of pheromone. The elitist strategy is also originated from the genetic algorithm, its main target is to protect the superior individual, then to keep the pheromone of the present generated optimal solution in the stronger level during the next iteration, after each iteration it shall bring additional pheromone to the local optimal solution, but this shall not aims to all the ants, the algorithm shall only use the elitist strategy on the ants that obtained the optimal solutions, to update the formula (11) then get the formula (12) accordingly:

$$\tau_{ij}(t+n) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) + e \Delta \tau_{ij}^*(t) \quad (12)$$

Among the formula the e is the weight value, but while the elite ant passing by it shall get the value as $\frac{P}{D^*}$, the D^* is the present optimal target value, while the elite ant not passing by the x_{ij} to get the value of $\Delta \tau_{ij}^*(t)$ as 0, means that not to use the elitist strategy on it accordingly.

This elitist strategy reduces the possibility of optimal solution missing, and more obviously promotes the pheromone content at each node of the optimal path, and ensure the ant colony standing on present optimal solution to seek for new optimal solution; once the better solution was found, the added pheromone could assure that the new optimal path shall not be missed, and the pervious path shall be reduced gradually due to the residual pheromone $1 - \rho$ influence, more and more ants even the elite ants shall move to the new optimal path to search the pheromone, then further reinforce the pheromone content accordingly.

But just because of the increase of pheromone, those nodes with much pheromone content shall have the bigger possibility to be searched accordingly, it may be easily fallen into local optimal solution but cannot find out the global optimal solution, thus it introduces the crossover operator of the genetic algorithm based on the elitist strategy [13], by taking the method of two point crossover to balance and the optimal path established by the ant colony and choose the best. For example, after generating the node

scheme $\{x_{y_1}, x_{y_2}, \dots, x_{y_j}, \dots, x_{y_n}\}$, it randomly chooses two nodes $x_{\xi_1 \xi_2}, x_{\xi_3 \xi_4} \in \{x_{y_1}, x_{y_2}, \dots, x_{y_j}, \dots, x_{y_n}\}$, and change their position oppositely, then to generate the new node set – which is the new scheduling scheme. To evaluate the new scheduling scheme and make comparison with the pervious scheme, if the T_{max} becomes smaller then to use the new scheduling scheme and exchange the pheromone accordingly, so as make the ants converging at the new optimal path; otherwise do not make the replacement and enter into next iteration directly.

3.3. The Procedure of Combining Optimization to Improve Algorithm

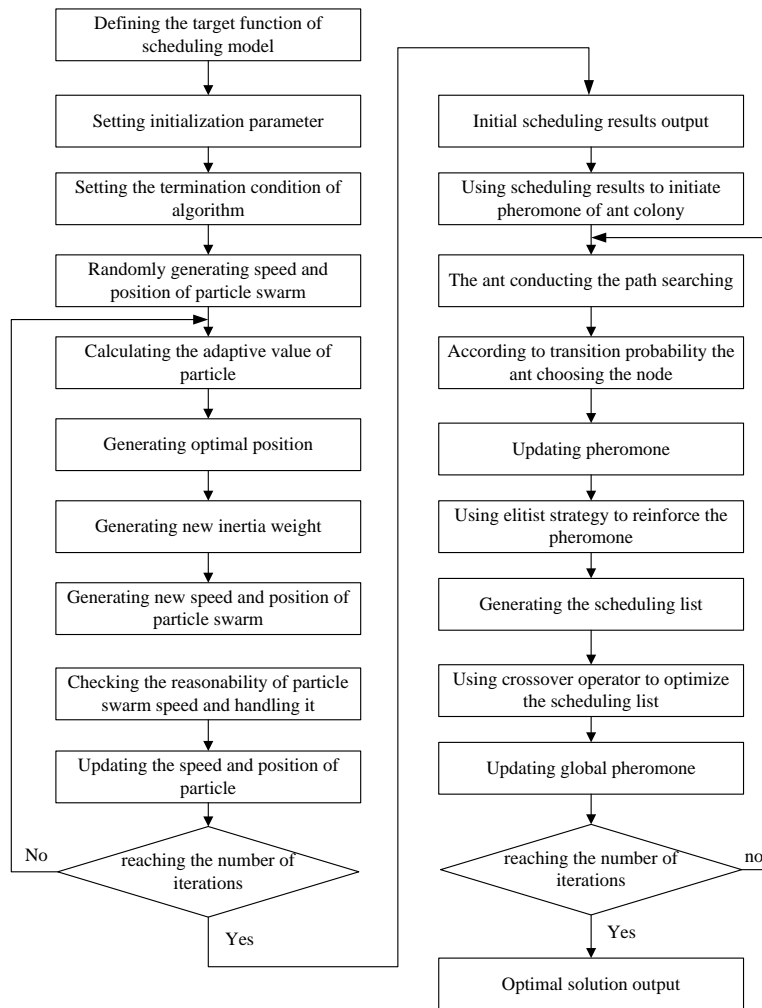


Figure 2. Improved Algorithm Flowchart

After improvement on the particle swarm optimization algorithm and ant colony algorithm, to introduce them into cloud computing task scheduling algorithm, the flowchart is shown in Figure 2.

In the improved algorithm, it first uses the particle swarm optimization algorithm to generate the initial scheduling results, meanwhile for improving the adjustment capacity of particle swarm optimization algorithm so as to generate initial scheduling results even faster, it introduces the inertia weight ω in randomness, to enhance the efficiency of particle swarm optimization algorithm. By taking the scheduling results generated by the

particle swarm optimization algorithm as the initial pheromone of the ant colony algorithm, it utilizes the stronger optimizing ability of ant colony algorithm to seek for the optimal scheduling scheme, meanwhile to improve the effectiveness and avoid the algorithm falling into local optimal solution, it uses the elitist strategy and crossover operator in genetic algorithm to make improvement on ant colony algorithm, in order to gain the better path, finally to generate the optimal scheduling scheme.

4. The Experimental Results and Conclusion Analysis

Table 1. Experiment Parameter Setting

Parameter		Parameter value
Particle swarm optimization algorithm	Population size	100
	Iteration numbers	50
	Population size	100
Ant colony algorithm	α	1
	β	1
	ρ	0.15
	Iteration numbers	150

In order to make evaluation and analysis on the performance of improved cloud computing model task scheduling algorithm, it made the experimental test on the CloudSim cloud computing simulation platform [14] developed by Melbourne University of Australia, and totally 5 virtual computing resources have been set accordingly, with the computing power as {400,600,800,1000,1200}, computing unit is million commands per second, the main parameter setting of the experiment was shown in Table 1.

By making simulated computation and comparison on both the combining optimization scheduling algorithm with improved particle swarm optimization algorithm and ant colony algorithm and the scheduling algorithm with the standard particle swarm optimization algorithm and ant colony algorithm, the used *ETC* matrix was generated by Matlab 7.0, the parameter setting for particle swarm optimization algorithm and ant colony algorithm are all the same, iteration numbers are totally 200 times, the experimental results for 100 sub tasks and 500 sub tasks are set respectively, after repeat operation for 50 times, it got the implemation time curve for the total task as per Figure 3 and Figure 4 shown respectively.

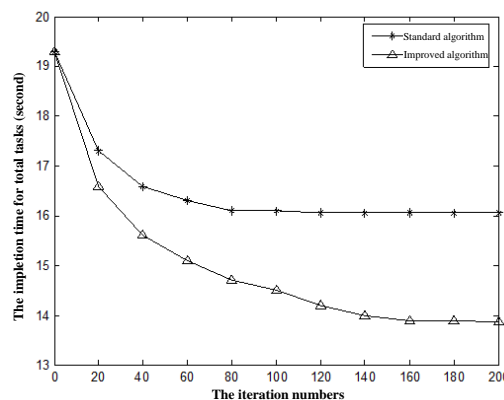


Figure 3. Task Completion Time of 100 Tasks

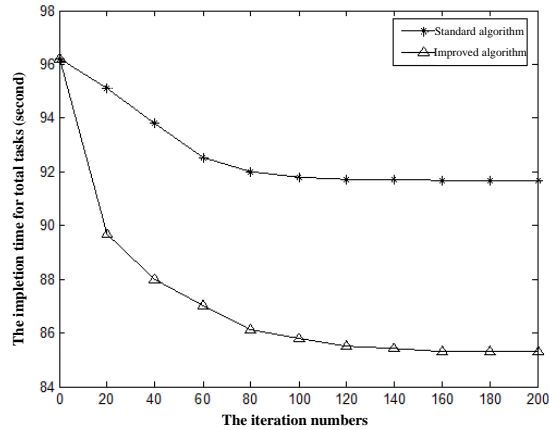


Figure 4. Task Completion Time of 500 Tasks

From the results shown in Figure 3 it can be seen that, while the experiment setting 100 sub tasks, after 80 iterations the impletion time for total tasks by standard algorithm before improvement is levelling off, but for the algorithm after improvement, after the 140th iteration the impletion time for total tasks is levelling off accordingly, it is thus clear that the algorithm before improvement has the quicker convergence rate. But the algorithm after improvement has the better impletion time of total tasks than that of algorithm before improvement; after the convergence, the algorithm after improvement has the shorter impletion time for total tasks in 2 seconds.

From the results shown in Figure 3 it can be seen that, while the experiment setting 500 sub tasks, after 80 iterations the impletion time for total tasks by standard algorithm before improvement is levelling off, but for the algorithm after improvement, after the 140th iteration the impletion time for total tasks is levelling off accordingly, it is thus clear that the algorithm before improvement has the quicker convergence rate; but the algorithm after improvement has the better impletion time of total tasks than that of algorithm before improvement; after the convergence, the algorithm after improvement has the shorter impletion time for total tasks in about 6.2 seconds.

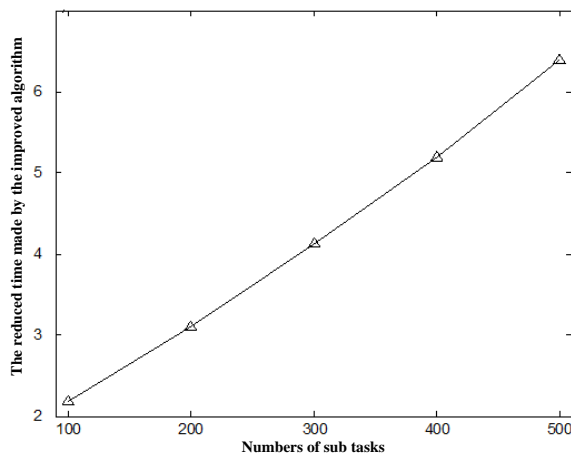


Figure 5. Reduces Time Value of Improved Algorithm

To test the reduction degree that the improved algorithm made to the impletion time for total tasks while there're different numbers of sub tasks, it assumes that the sub task are respectively 100, 200, 300, 400, 500 and the other parameters are the same as above, after

making the experiment the results were get as per Figure 5 shown, it is clear that with the increased quantity of sub tasks, the reduced time that the improved algorithm made shall also become increased accordingly, means that the effects of improved algorithm has been reinforced accordingly.

5. Closing Remarks

In this article, the main work is that based on the combination optimization on cloud computing task scheduling algorithm by introducing particle swarm optimization algorithm and ant colony algorithm, to make improvement on it accordingly, using the inertia weight in randomness to reinforce the adjustment ability of algorithm and promote the efficiency of particle swarm optimization algorithm; meanwhile, in order to improve the effectiveness and avoid ant colony algorithm falling into local optimal solution, it uses the elitist strategy and crossover operator of the genetic algorithm to improve the ant colony algorithm, so as to obtain the better path, finally to generate the optimal scheduling scheme, The experimental results show that, based on the high performance of previous algorithm the improved algorithm could further promote the impletion time for total tasks, and with the increased sub tasks quantity, the impletion time for total tasks shall get the bigger range in decrease. But there's still space for making progress of such algorithm, the focal [pot of the works for next step is to make improvement on the aspects such as convergence rate.

Reference

- [1] Z. Lu, C Esteve, J Chirivella and P Gagliardo, "A Game Based Assistive Tool for Rehabilitation of Dysphonic Patients". 3rd International Workshop on Virtual and Augmented Assistive Technology (VAAT) at IEEE Virtual Reality 2015 (VR2015), Arles, France, IEEE, (2015).
- [2] Zi Chen, W Huang, Lv Zhihan. "Uncorrelated Discriminant Sparse Preserving Projection Based Face Recognition Method". Multimedia Tools and Applications. (2016).
- [3] Lv Zhihan, A Halawani, S Feng, H Li, and S Ur Rehman. 2014. "Multimodal Hand and Foot Gesture Interaction for Handheld Devices". ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). 11, 1s, Article 10 (October 2014), 19 pages.
- [4] K Leng, W Shi, J Chen, Lv Zhihan. "Designing of a I-shaped less-than-truckload cross-dock: A simulation experiments study". International Journal of Bifurcation and Chaos. (2015).
- [5] Y Lin, J Yang, Lv Zhihan, W Wei, H Song. "A Self-Assessment Stereo Capture Model Applicable to the Internet of Things". Sensors. (2015).
- [6] J He, Y Geng, K Pahlavan, "Toward Accurate Human Tracking: Modeling Time-of-Arrival for Wireless Wearable Sensors in Multipath Environment", IEEE Sensor Journal, vol. 14, no. 11, (2014) Nov., pp. 3996-4006
- [7] W Ou, Lv Zhihan, Z Xie. "Spatially Regularized Latent topic Model for Simultaneous object discovery and segmentation". The 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC2015).
- [8] W Wang, Z Lu, X Li, W Xu, B Zhang, X Zhang. "Virtual Reality Based GIS Analysis Platform". 22th International Conference on Neural Information Processing (ICONIP 2015), Istanbul, Turkey.
- [9] W Ke. "Using Simulation to Explore Distributed Key-Value Stores for Exascale System Services." 2nd Greater Chicago Area System Research Workshop (GCASR). (2013).
- [10] J He, Y Geng, F Liu, C Xu, "CC-KF: Enhanced TOA Performance in Multipath and NLOS Indoor Extreme Environment", IEEE Sensor Journal, vol. 14, no. 11, Nov. (2014), pp. 3766-3774.
- [11] S Zhou, L Mi, H Chen, Y Geng, "Building detection in Digital surface model", 2013 IEEE International Conference on Imaging Systems and Techniques (IST), Oct. (2012).
- [12] N Lu, C Lu, Z Yang, Y Geng, "Modeling Framework for Mining Lifecycle Management", Journal of Networks, vol. 9, no. 3, (2014) Jan, pp. 719-725.
- [13] L Tonglin. "ZHT: A light-weight reliable persistent dynamic scalable zero-hop distributed hash table." Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on. IEEE, (2013).
- [14] W Ke. "Optimizing load balancing and data-locality with data-aware scheduling." Big Data (Big Data), 2014 IEEE International Conference on. IEEE, (2014).
- [15] M Zhou, G Bao, Y Geng, B Alkandari, X Li, "Polyp detection and radius measurement in small intestine using video capsule endoscopy", 2014 7th International Conference on Biomedical Engineering and Informatics (BMEI), Oct. (2014).
- [16] X Song, Y Geng, "Distributed community detection optimization algorithm for complex networks", Journal of Networks, 9(10), 2758-2765, Jan. (2014).

- [17] G Liu, Y Geng, K Pahlavan, "Effects of calibration RFID tags on performance of inertial navigation in indoor environment", 2015 International Conference on Computing, Networking and Communications (ICNC), Feb. (2015).
- [18] G Yan, Lv Yuxiang, Q Wang, Y Geng, "Routing algorithm based on delay rate in wireless cognitive radio network, Journal of Networks", vol. 9, no. 4, Jan. (2014), pp. 948-955,
- [19] DJiang, X Ying, Y Han, Lv Zhihan. "Collaborative Multi-hop Routing in Cognitive Wireless Networks". Wireless Personal Communications. (2015).
- [20] Z, Xiaobing "Exploring Distributed Resource Allocation Techniques in the SLURM Job Management System." Illinois Institute of Technology, Department of Computer Science, Technical Report (2013).

Authors



XiaoLi He, she was born in Sichuan, China, in 1982. She received her M.S. degree in computer application technology from China West Normal University, Nanchong, China, in 2005 and 2008. She is currently a lecturer in the Sichuan University of Science & Engineering. Her research interest is mainly in the area of Computer Network, Computer Application Technology and Electronic Commerce. She has published several research papers in scholarly journals in the above research areas and has participated in several books.

