

Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing¹

Lu Huang

Software School of Xiamen University, Xiamen, China

Email: bangbang_4391@qq.com

Hai-shan Chen²

Software School of Xiamen University, Xiamen, China

Email: hschen@xmu.edu.cn

Ting-ting Hu

Software School of Xiamen University, Xiamen, China

Email: 815464551@qq.com

Abstract—Cloud computing is the product of the evolution of calculation. It is a new distributed computing model. As more and more people put into the research and applications on cloud computing, the technology of computing becomes more and more widely used. Cloud computing has a huge user group. It has to deal with a large number of tasks. How to make appropriate decisions when allocating hardware resources to the tasks and dispatching the computing tasks to resource pool has become the main issue in cloud computing. This paper is based on the current situation of resource allocation policy and job scheduling algorithms under cloud circumstance. It summarizes some methods to improve the performance, including dynamic resource allocation strategy based on the law of failure, dynamic resource assignment on the basis of credibility, ant colony optimization algorithm for resource allocation, dynamic scheduling algorithm based on threshold, optimized genetic algorithm with dual fitness and improved ant colony algorithm for job scheduling.

Index Terms—Cloud Computing, Resource Allocation, Job Scheduling, Ant Colony Algorithm, Genetic Algorithm

I. INTRODUCTION

Fig.1 shows the mode of calculation in different stages of development. Calculation mode has gone through a process from the original mode of gathering all the tasks to large-scale processor for processing (Fig.1(a)), to the distributed tasks processing mode based on Internet (Fig.1(b)) later, and then to the cloud computing mode for immediate processing (Fig.1(c))^[1]. Calculation is executed on distributed computer cluster in cloud computing. It is on kind of calculation mode driven by a large-scale commercial demand. Enterprises can get access to computing power, storage capacity, services and other abilities according to their needs. Cloud computing

is a super internet based computing model in which tens of thousands of computers and servers are connected into a computer cloud^[2]. Strictly speaking, Cloud Computing is a model of enabling ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resource(e.g. networks, servers, storage, applications, and service) that can be rapidly provisioned and released with minimal management effort or service provider interaction according to NIST(National Institute of Standards and Technology)^[3]. Cloud computing is a product from mixing traditional computer techniques and network technologies, such as grid computing, distributed computing, parallel computing, utility computing, network storage, virtualization, load balancing, etc^[4].

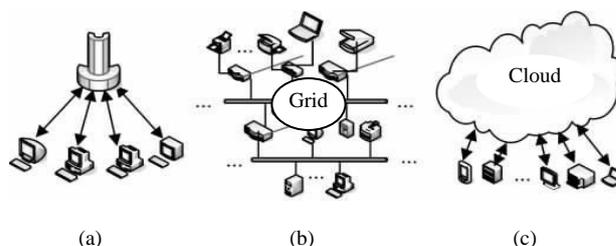


Fig.1 Evolution of Calculation Mode

II. RESOURCE ALLOCATION POLICY

A. Overview

Cloud computing evolves from grid computing, which is also regarded as its backbone and basic structure. It is arguable that cloud computing is a higher form of grid computing. But there is a big difference between them in reality. The specific content of which can be found in Evoy's paper [5]. The features that dispersion, heterogeneity and uncertainty of resources in the nodes bring challenges to resource allocation, which can not be

¹ Supported in part by a grant from the Leading Academic Discipline Program, "Project 211 (the 3rd phase)" of Xiamen University.

² Corresponding Author: hschen@xmu.edu.cn

satisfied with traditional resource allocation policies in cloud circumstance.

How to achieve dynamic configuration and shared use of computing resources is a core issue in the field of cloud. Researchers have proposed a variety of schemes for dynamic resources providing and management, which have been described in Irwin's paper [6] and Padala's paper [7]. Many scholars have put forward solutions for efficient resources allocation based on market mechanisms. CDA (Continuous Double Auction) is one of the common market mechanisms being used currently. It ensures high efficiency and effective coordination of resource allocation. Kenney's paper [8] proves that web resource allocation based on CDA framework is effective. And Li's paper [9] presents a cloud resource assignment policy based on CDA framework and Nash equilibrium to fulfill effective resource allocation in cloud environment. Meanwhile, Researchers have proposed several ideas to improve resource reliability, such as reliability verification method of resources based upon the law of failure of cluster nodes^[10], proposed by Heath, multiple checkpoint strategy for improving the ability of avoiding performance loss caused by system failure for HPC service^[11], presented and analyzed by Zhang, and the capacity of HPC job queue structure for improving the reliability of job running^[12], analyzed by Hacker. However, there is little discussion in the literature about the research on how to ensure the reliability of dynamic resource provision under cloud circumstance. It is important for both cloud computing infrastructure operators and service operators to guarantee the reliability of dynamic provision resource. Either the QoS (Quality of Services) and SLO (Service Level Objects) or the efficiency and effectiveness of dynamic resource that service providers should ensure depended on the reliability of dynamic resource providing. For this question, based upon the failure of rules that heterogeneous services present in time and space in cloud environment, researchers have proposed strategies such as dynamic resource provision on the basis of the law of failure^[13] and dynamic resource allocation based on credibility^[14] and so on.

Allocation of resources is an important component of cloud computing. Its efficiency will directly influence the performance of the whole cloud environment. Since cloud computing has its own features, original resource allocation policy and scheduling algorithms for grid computing are unable to work under this condition. To deal with this issue, Hua's paper [15] proposed an ant colony optimization algorithm for resource allocation, in which all the characteristics in cloud are considered. It has been compared with genetic algorithm and annealing algorithm, proving that it is suitable for computing resource search an allocation in cloud computing environment.

B. Dynamic Resource Allocation Strategy Based on The Law of Failure

As services on cloud computing platform can be broadly divided into two categories, which are data computation-intensive services and interaction-intensive

network processing services, the law of failure of these two services are took into account.

Research workers have made a lot of researches for the rule of system failure of node resources. Hua's paper[10], Schroeder's paper [16] and Sahoo's paper [17] present that through the study of log on the system failure, researchers found that the failure presents a strong time and spatial locality. And the spacing interval of node restarting failure without plans is regarded as a random process, which fits the Weibull (scale, shape) distribution with the parameters of shape less than 1. This suggests that the node which has just failed is more likely to fail again, while running, the node will become more and more stable. So it is necessary to make a period of test on the reliability of nodes which have just failed. Meanwhile, researchers point out that for a typical cluster system mixed by high-performance computing services and network processing services, the probability of the failure of node restarting randomly is from descending to ascending during the running time^[17]. This phenomenon is obvious for database nodes in the mixed cluster. At the same time, since service systems (such as data center services) accumulate more and more internal errors while running, software aging problem is common. In this regard, researchers propose software regeneration strategies. When the system is active regeneration, errors within the system will be cleared and the system will be back to normal state^[18]. Moreover, effective regeneration strategies can reduce system performance losses caused by software rejuvenation technology.

Combining the failure law, research workers put forward a dynamic node resource allocation policy based on failure rules^[13]. The policy includes two major categories of a single queue and multiple queue policies, in which the former is the basic of the latter. Dynamic provision strategy of node resource is an On-Demand strategy. The single queue strategy (Fig.2) maintains an ordered node resource pool according to the last recovery time (uptime). Failure nodes are placed on the top of the queue. When there is a request, this strategy will select a free node from the tail of queue to handle this request, ensuring that the node is the most reliable one in the idle resource pool. When the job is done, the resource node will be put back to the queue in order. For the basic single queue strategy, only one list is maintained, while for the extended multiple queue strategy (Fig.3), several queues are maintained. At each time, one appropriate queue is selected for resource providing and recycling operations.

Tian's paper [13] presents the comparison experiments between the single queue and multiple queue strategy under a variety of circumstances to prove the efficiency and reliability of them.

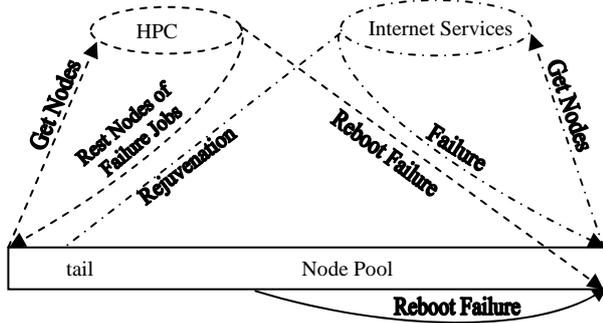


Fig.2 Basic Single Queue Strategy

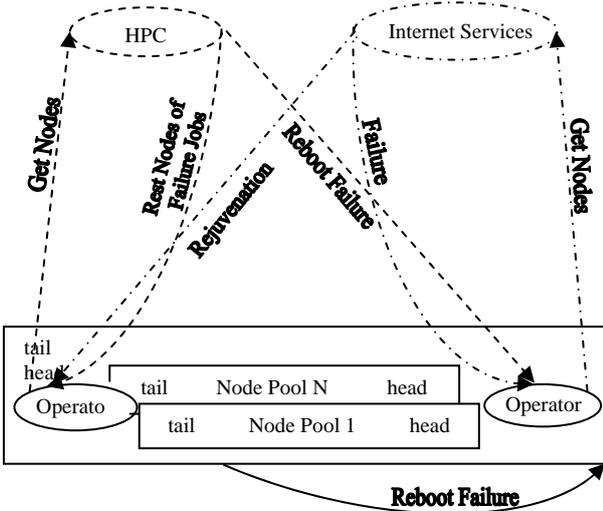


Fig.3 Extended Multiple Queue Strategy

C. Dynamic Resource Assignment on The Basis of Credibility

The requests for resource under cloud environment typically exhibit strong volatility. To ensure the credibility of dynamic resources without affecting its service efficiency, researchers propose a more credible dynamic resource provision strategy^[19].

In the cloud, resource allocation model based on CDA mechanism mainly includes cloud resource providing agent, cloud resource requirement agent, and information serving agent. They consult each other to achieve a balance on the price and the amount of resource for transaction. When entering or leaving a cloud resource system, both the owner of the resource and the cloud user need to be registered to the information serving agent. And the owner will set a price and allocate the resources through resource providing agent, while the user will allocate appropriate amount of resource through resource requirement agent to the jobs needed to be done. In CDA mechanism, resource providing agent, resource requirement agent and information serving agent correspond to the seller, the buyer and the arbiter in the auction respectively. The arbiter is responsible for organizing the auction and collecting market information. At any time unit during the auction, the seller and the buyer offer their own price to the arbiter, and the arbiter will match the resource transactions based on both sides' price lists and give an average price for both sides. The

buyers and the sellers will confirm the quote according to the market transaction environment and their own excitation mechanism. Taking the law of the node failure of cloud resource, and on the basic of CDA mechanism and the credibility of node, researchers proposed a dynamic resource allocation model (T-CDA).

First, both sides of supply and demand should confirm their own prices based upon their own pricing strategy. The auctioneer will sort the price of resource provider in a descending order and the price of resource demander in an ascending order. Then it will determine whether the resource transaction can be concluded based on the utility model of resource trading. Cheng's paper [14] presents a simulation experiment on this strategy in Matlab 7.1. Through the result of simulation and evaluation of successful execution ratio and a deviation from fairness in resource allocation, this strategy is proved to be markedly superior to the resource allocation strategies without considering the credibility of nodes.

D. Ant Colony Optimization Algorithm for Resource Allocation

Cloud computing distributed cluster uses a Master/Slaves structure (Fig.4). There is a Master node responsible for controlling and supervising all the Slave nodes. Since the specific condition of resource is unknown under cloud circumstance, and the networks do not have a fixed topology, the structure and the resource allocation of the whole cloud environment is unpredictable. In this case, the location and quality of computing resources for data nodes is unknown.

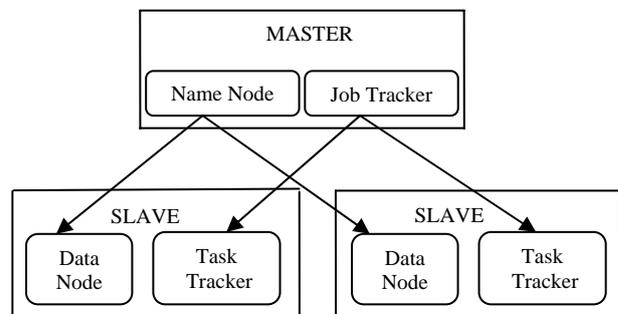


Fig.4 Master/Slaves Structure of Hadoop Cluster

Ant Colony Optimization (ACO) is an updated bionic optimization algorithm which is in simulation of ant foraging behavior. It is originated by M.Dorigo et al. who was inspired by the research result of the group behaviour of real ants^[20]. ACO algorithm shows characteristics of rapidity, distribution and global optimization when solving complex optimization problems. And the rapidity of finding the optimal solution is due to the regenerative feedback mechanism of pheromone. While its feature of distributed computation avoids premature convergence of the algorithm. Meanwhile, the ant system, with the feature of greedy heuristic search, can find an acceptable solution early in the search process.

The pseudocode of the prototype system of the ant colony algorithm can be expressed as follows:

Procedure: ACO Algorithm

```

Begin
  While (ACO has not been stopped) do
    Schedule activities
    Ant's allocation and moving ( ant distribution and
    movement )
    Local pheromone update ( local pheromone
    update )
    Global pheromone update ( global pheromone
    update )
    End schedule activities
  End While
End Procedure

```

Ant colony algorithm can find out computing resources in unknown network topology and select the most appropriate one or more resources to user's job until it meets user's requirements. When the search begins, query messages will be sent by slave node, and they will play the role of ants. All of the ants obey the principle of "the more pheromone one node has, the larger probability it will be, vice versa" to choose the next hop node, and it will leave pheromone on the node of the path as it goes through. In order to reflect the change of the pheromone, researchers adopt a local update strategy to modify the pheromone intensity onto the node. Hua's paper [15] provides a detailed description about ant colony algorithm in resource allocation and uses Gridsim to simulate local domain of cloud computing to inspect the operating conditions of the algorithm under cloud network environment. At the same time, this algorithm is compared with genetic algorithm and simulated annealing algorithm. Through GridResource class and a series of helper classes in Gridsim, researchers simulate the computation and network resources of cloud computing and constructs a relatively real cloud layout. After much experimentaiton, researchers found that ant colony algorithm is more effective than other two algorithms in the case that there are more nodes and fewer resources, which is just the characteristics of cloud environment. Ant colony algorithm aims at the large-scale, shared, dynamic and other characteristics of cloud environment. It assigns search and allocates computation resources to user's job dynamically. And it shows more advantages in cloud environment.

III. JOB SCHEDULING ALGORITHM UNDER CLOUD CIRCUMSTANCE

A. Background

Job scheduling of cloud computing refers to the process of adjusting resources between different resource users according to certain rules of resource use under a given cloud environment^[21]. Resource management and job scheduling are the key technologies of cloud computing. At present, there is not an uniform standard for job scheduling in cloud. Most alorithms focus on

job dispatcher, which is almost resopnsible for all the task allocations, responses and retransmissions^[22].

Over-reliance on the scheduler may lead to some virtual machines overload while others are idle after dispatcher allocating tasks according to the load of virtual machines. When this occurs, the only solution is to assign tasks for next period according to what the feedback scheduling device gets. The process in different vitual machines is independent. A virtual machine does not have access to other virtual machines' running conditions. So if one of the following two problems occures, the execution efficiency of the tasks will be affected. One of the problem is that the dispatcher is out of joint, and the other one is the environment of virtual machines changes, resulting in the problems of some virtual machines and making them unable to send the information back to the scheduler. This will also cause some virtual machines overload while others free.

B. Dynamic Scheduling Algorithm Based on Threshold

To get the real-time feedback of the state of virtual machine, there are two ways. One of them is to construct a set of feedback mechanism between dispatcher and virtual machines to get the real-time feedback of the tasks load on virtual machine, and then make a real-time adjustment on job allocation upon the fact of virtual machines. The other one is to use the dynamic scheduling among virtual machines themselves to get the real-time state of the load of virtual machines. If overload or idleness occurred, tasks could be readjusted and redistributed among virtual machines. By dynamic dispatch in virtual machines, dynamic scheduling algorithm based on threshold can allocate jobs and resources flexibly and reduce the efficiency impact caused by the synchronization among virtual machines. Fig.5 shows the model of the dynamic scheduling algorithm based on threshold^[21].

The situation showed in Fig.5 is that if there are some virtual machines overload and some idle at a certain time, dynamic job adjustment is conducted to shorten the total cost time, thereby enhancing efficiency. However, task allocation between virtual machines refers to synchronization problems, which is also the biggest problem of the dynamic scheduling algorithm based on threshold. Since each virtual machine is independent to each other, in the other word they are non-interfering. They can perform tasks in parallel. If virtual machines are synchronized, they inevitably bring effects to their performance. Therefore, the synchronization operation should be kept to minimum range.

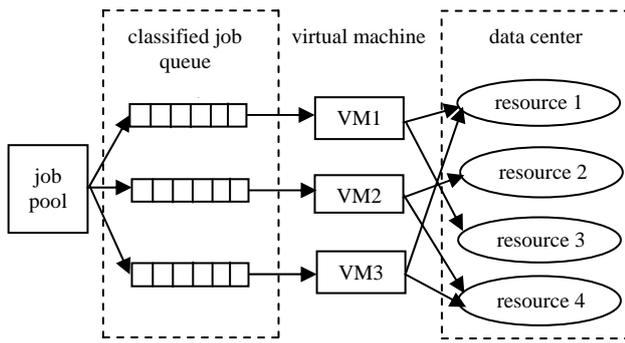


Fig.5 Model of the Dynamic Scheduling Algorithm Based on Threshold

In order to reduce the impact of synchronization, two measurements are taken. First, set the threshold. Synchronization is executed only when the virtual machine reaches a threshold. The larger the threshold is, the smaller the impact of synchronization will be. Second, limit the synchronization down to two virtual machines. The smaller the number of virtual machines for synchronization is, the weaker the impact it brings.

Fig.6 indicates the flow of the dynamic scheduling algorithm based on threshold.

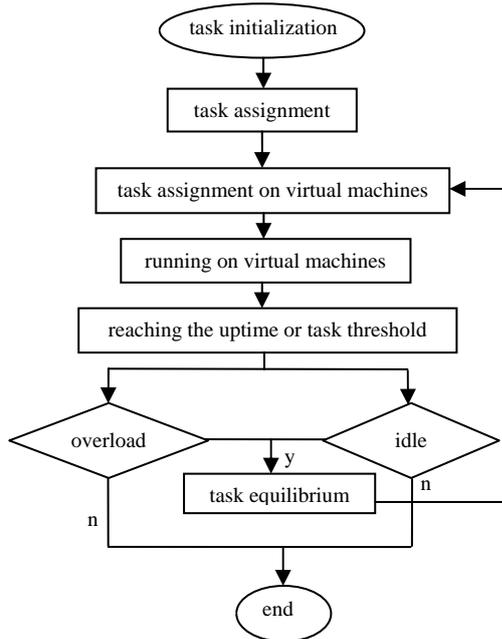


Fig.6 Flow of the Dynamic Scheduling Algorithm Based on Threshold

Task assignment involves in setting task classification according to PRI. A task that has a higher execution priority has higher PRI.

Reaching the uptime or task threshold means that the time threshold of task running or the number of tasks that are waiting in the line is reached. It includes two conditions. There are two types of task threshold. One is the number of tasks waiting to be done in the queue on one virtual machine. The other one is the number of tasks that have been finished on another virtual machine. If both numbers were larger than the threshold value at the same time, these two virtual machines would be

synchronized. And their tasks will be balanced and will continue working.

Task equilibrium means that if there is at least one idle virtual machine and at least one overload virtual machine, other virtual machines will execute tasks independently.

Xian's paper [23] compares between dynamic scheduling algorithm based on threshold and virtual machines with the static independent job scheduling algorithm on CloudSim platform. The result suggests that when there are a fairly large number of tasks, the former can complete task allocation efficiently and reduce the running time greatly. It shows an obvious advantage over the latter.

C. Optimized Genetic Algorithm

Genetic Algorithm (GA) is proposed by Holland, who was inspired by biological evolution, in 1975. Parallelism and global solution space search are the two notable features of the GA^[24]. Fig.7 shows the flow of GA.

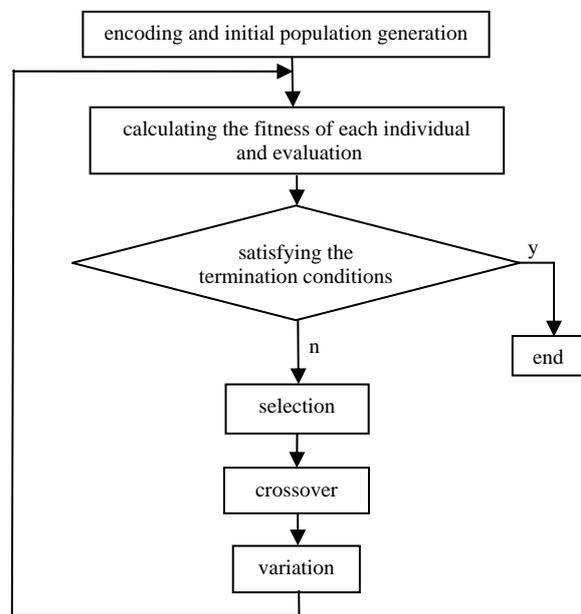


Fig.7 Flow of GA

On the basis of Map/Reduce model in cloud^[25], in order to cut down both the total running time and the average time of task execution, researchers add one more fitness to improve the GA. That is the optimized genetic algorithm with dual fitness (DFGA), which has two fitness functions.

DFGA algorithm uses the indirect encoding method of resource—task. The length of chromosome is the number of sub tasks. And the value of each gene on the chromosome is corresponding to the resource number which is allocated to the sub task on this location. Initialization is to generate a SCALE number of chromosome, which has a length of M, and the value range of gene is a random number in [1, WORKER]. Among them, M stands for the total number of sub tasks, and WORKER is the number of resources. There are two fitness functions, one is the total time of job running on all virtual machines and the other is the average time.

Both of them should be short. The selection operator uses Roulette method.

Li's paper [24] describes the comparison between adaptive genetic algorithm (AGA) and DFGA under a local environment of cloud computing, which is simulated on Gridsim. The result shows that after many generations of evolution, both the total time and the average time of tasks execution using DFGA are significantly superior to the AGA. DFGA is an effective job scheduling algorithm.

D. Improved Ant Colony Algorithm

The essence of job scheduling is to select a way of dynamic combination of resource with relatively good performance among all the resource allocation methods. From the perspective of problem-solving, optimized ant colony algorithm is very suitable for resource allocation in cloud environment^[26]. As the randomness of ant colony algorithm is large, it is easily trapped in local optimal solution and slow convergence. Thus, research workers introduce GA, which has a capability of rapid and random global search, to each iteration process of ant colony algorithm. This can greatly accelerate the speed of convergence and ensure the accuracy of the original algorithm.

For each resource requester, cloud computing service cluster should give a fairly good combination of tasks and resources. In improved ant colony algorithm, at the same time, the factors affecting the resource state can be described by pheromone, and the scheduling process can get predictable results simply and quickly.

In cloud circumstance, take ACS (Ant Colony System) algorithm model based on ACO algorithm for example, the flow of job scheduling process based on ACO algorithm can be described as Fig.8^[27].

Wang's paper [21] describes the analog simulation of the improved ACO algorithm based on an extended cloud computing simulation platform. It was compared with the Round Robin (RR) algorithm and the original ACO algorithm. Generally, improved ACO algorithm takes less time and has a higher efficiency than other two algorithms.

IV. CONCLUSIONS

This paper analyzes resource allocation and job scheduling issues under cloud environment and describes interrelated solutions proposed by research workers. Progress has been made in the existing strategies, which can significantly improve the efficiency of the use of resources according to certain situation in cloud. As the application area of cloud computing becomes wider and wider, resource allocation and job scheduling algorithms will be further improved in order to adapt to a variety of specific application environments.

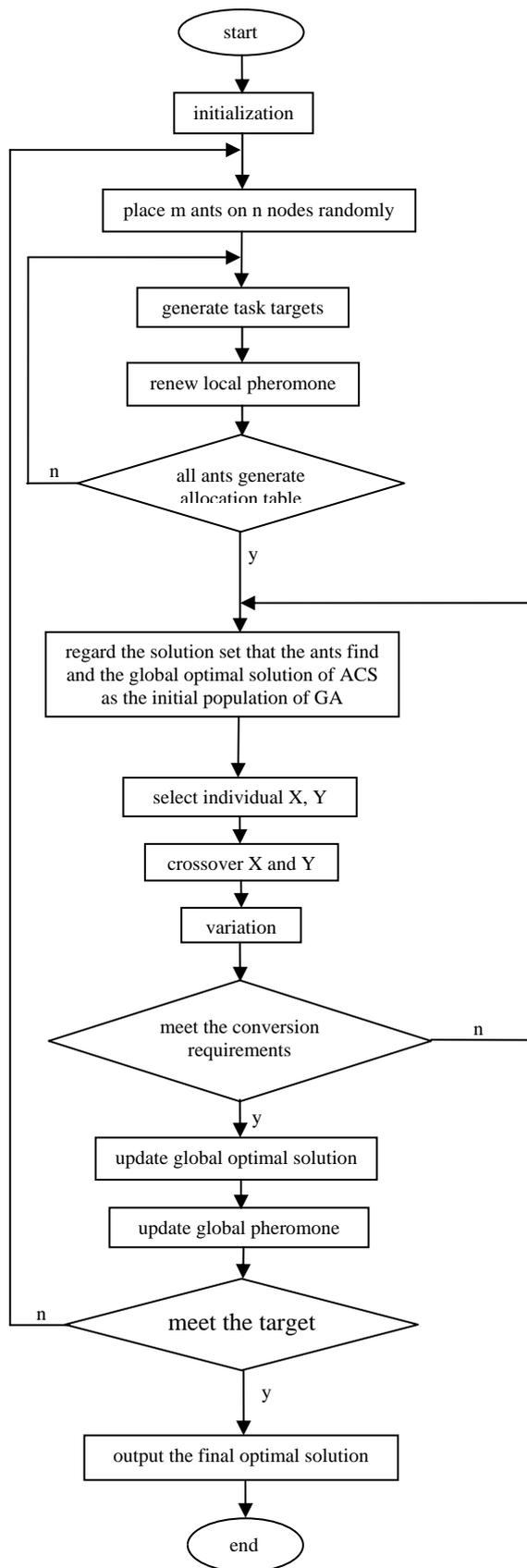


Fig.8 Flow of Job Scheduling Process Based on ACO Algorithm

ACKNOWLEDGMENT

This work was supported in part by a grant from the Leading Academic Discipline Program, "Project 211 (the 3rd phase)" of Xiamen University.

REFERENCES

- [1] Li Qiao, Zheng Xiao. Research Survey of Cloud Computing[J]. Computer Science, 2011, 4(38): 32~37.
- [2] Chunmei Chi, Feng Gao. The Trend of Cloud Computing in China[J]. Journal of Software, VOL.6, NO.7, 2011,7: 1230~1234.
- [3] Tong Yang, Ben-Chang Shia, Jinrui Wei, Kuangnan Fang. Mass Data Analysis and Forecasting Based on Cloud Computing[J]. Journal of Software, VOL.7, NO.10, 2012,10: 2189~2194.
- [4] Lin Hua, Yang Dongri, Liu Longgeng. Master interview – cloud computing promotes business and technological change[M]. Beijing: Electronic Industry Press, 2011.
- [5] MC Evoy G V, Schulze B. Using clouds to address grid limitations[C]. MGC'08. Belgium: Leuven Press, 2008.
- [6] Irwin D, Chase J S, Grit L et al. Sharing networked resources with brokered leases[C]. Proceedings of the USENIX Technical Conference, Boston, MA, USA, 2006, pp299~212.
- [7] Padala P, Shin K G, Zhu Xiao Yun et al. Adaptive control of virtualized resources in utility computing environments[C]. Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007. Lisbon, Portugal, 2007, pp289~302.
- [8] Kenney J, Eberhart R. Particle Swarm Optimization[C]. Proc. of IEEE International Conf. on Neural Networks. Perth. USA [s. n.], 1995.
- [9] Li Li, Niu Ben. Particle swarm optimization[M]. Beijing: Metallurgical Industry Press, 2009.
- [10] Heath T, Martin R P, Nguyen T D. Improving cluster availability using workstation validation[C]. Proceedings of the ACM SIGMETRICS. Marina Del Rey, California, USA, 2002, pp217~227.
- [11] Zhang Y, Squillante M S, Sivasubramaniam A et al. Performance implications of failures in large-scale cluster scheduling[C]. Proceedings of the 10th Worksh-op on Job Scheduling Strategies for Parallel Processing JSSPP 2004. New York, NY, USA, 2004, pp233~252.
- [12] Hacker T J, Meglicki Z. Using queue structures to improve job reliability[C]. Proceeding of the ACM HPDC 2007. Monterey, California, USA, 2007, pp43~54.
- [13] Tian Guanhua, Meng Dan, Zhan Jianfeng. Reliable Resource Provision Policy for Cloud Computing[J]. Chinese Journal of Computers, 2010, 10(33): 1859~1872.
- [14] Cheng Shiwei, Pan Yu. Credibility-based dynamic resource distribution strategy under cloud computing environment[J]. Computer Engineering, 2011, 6(37): 45~48.
- [15] Hua Xiayu, Zheng Jun, Hu Wenxin. Ant colony optimization algorithm for computing resource allocation based on cloud computing environment[J]. Journal of East China Normal University (Natural Science). 2010, (1): 127~134.
- [16] Schroeder B, Gibson G A. A large-scale study of failures in high-performance computing systems[C]. Proceedings of DSN 2006. Philadelphia, Pennsylvania, USA, 2006, pp249~258.
- [17] Sahoo R K, Sivasubramaniam A, Squillante M S et al. Failure data analysis of a large-scale heterogeneous server environment[C]. Proceedings of the DSN 2004. Florence, Italy, 2004, pp772~784.
- [18] Huang Y, Kintala C, Kolettis N et al. Software rejuvenation: Analysis, module and applications[C]. Proceedings of the 25th Symposium on Fault Tolerant Computer Systems. Pasadena, California, 1995, pp381~390.
- [19] Valkenhoef G, Famchurn S D, Vytelingum P, et al. Continuous Double Auctions with Execution Uncertainty[C]. Proc. of Workshop on Trading Agent Design and Analysis. Pasadena, California, USA [s.n.], 2009.
- [20] Dorigo, Maniezzo, Colomi. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on SMC, 1996, 26(1): 8~41.
- [21] Wang Yonggui, Han Ruilian. Study on cloud computing task schedule strategy based on MACO algorithm[J]. Computer Measurement & Control, 2011, 19 (5): 1203~1204, 1211.
- [22] Wu Jiyi, Ping Lingdi, Pan Xuezheng, etc. Cloud computing form concept to platform[J]. Telecom Technology, 2009, (12): 23~25.
- [23] Xian Jin, Yu Guicheng. Research on scheduling algorithm based on cloud computing[J]. Computer & Digital Engineering, 2011, (7): 39~42.
- [24] Li Jianfeng, Peng Jian. Task scheduling algorithm based on improved genetic algorithm in cloud computing environment[J]. Journal of Computer Applications, 2011, 1 (31): 184~186.
- [25] DEAN J, GHEMAWATS. MapReduce simplified data processing on large clusters[C]. Proceedings of the 6th Symposium on Operating System Design and Implementation. New York ACM, 2004, pp137~150.
- [26] Wang Tianqing, Xie Jun, Zeng zhou. Grid resource scheduling strategy based on ant colony algorithm[J]. Computer Engineering and Design, 2007, 28(15): 3611~3613.
- [27] Dorigo M, Caro GD. Ant colony optimization: A new meta-heuristic[A]. Proc. of the 1999 Congress on Evolutionary Computation. Washington: IEEE Press, 1999, pp1470~1477.



Lu Huang received the Bachelor of Digital Media Arts from Xiamen University in 2011.

At present, she is a postgraduate student at the Software School of Xiamen University. Her research areas are cloud computing and web data mining.



Hai-shan Chen received the Bachelor of Science from Xiamen University in 1982.

At present, He is the professor of software school of Xiamen University. His research interests include database technology, Web services and cloud computing.



Ting-ting Hu received the Bachelor of Digital Media Arts from Xiamen University in 2011.

At present, she is a postgraduate student at the Software School of Xiamen University. Her research area includes data mining and cloud computing.