Scheduling Algorithm of Cloud Computing Based on DAG Diagram and Game Optimal Model

Liu Jun and Guo Zuhua

*Department of Computer Science, Henan Mechanical and Electrical Engineering
College, Henan, China
ddnt@163.com*

Abstract

In order to improve the efficiency of cloud computing task scheduling, we propose a cloud scheduling algorithm based on DAG task graph and game optimal. This method first constructed scheduling tasks of DAG task graph, and set the initial virtual machine for root and leaf nodes, then based on the optimization model of game, the effectiveness of difference between the task configuration after optimization judgment and the current task configuration until the effectiveness difference within a preset range, the experimental results show that the methods herein can achieve not only balancing scheduling between each virtual machine, but also has a faster rate scheduling.

Keywords: Cloud computing, task scheduling, DAG graph, game optimal

1. Introduction

Task scheduling is important content of cloud computing research. Simply put, task scheduling in cloud computing environment is to service requests submitted by user decomposed into multiple tasks, and assign these sub-tasks to the appropriate virtual machine. Nice task scheduling algorithm not only to meet users' requests but also to try to maximize improve the performance of cloud computing platform.

Bilgaiyan for high-throughput tasks in heterogeneous environments, designed a scheduling algorithm, and did a simulation for algorithm model, experimental results show that the algorithm model for high throughput problems has good advantages [1]. Jianfang made heterogeneous resources on the grid fuzzy partitioned, then use the DAG represent dependencies between tasks, designed algorithm greatly reduces the task execution time [2]. Benyi proposed an improved particle swarm optimization algorithm, at algorithm added dynamic multi-group collaboration mechanisms, to a certain extent, to avoid algorithm falling into local optimal, but it has poor convergence and slower solving speed [3]. Arya defines a allotting game in cloud computing resource, the game can be guaranteed price optimal paid by the user and resource usage of cloud computing center is also close to optimal, but did not consider the different quality of service requirements of multiple user classes [4]. Chaukwale used dynamic evolution of game model to solve the network routing problem, use the marginal cost expressed effectiveness of the participants, and prove the stability of evolution, using the dynamic of replicators making game converge to a stable point, designed evolutionary algorithm [5]. Verma made independently choose of physical paths in virtual network for underlying network abstracted into evolutionary game, virtual network through the repeatedly game learning to adjust the selection of the physical path, and analyzed the evolution of the game's using the potential game theory, proved evolution results will reached evolution equilibrium, to ensure that each virtual network have reached the optimal [6]. Quarati proposed a scheduling algorithm that comprehensive consideration of cost constraints and task priority, the ability the algorithm firstly computing priority of tasks and all resources of data centers, tasks and resources were sorted, complete matching of tasks and resources

[7]. Sadooghi consider the characteristics of resources and preferences of users or tasks using fuzzy clustering method for resources to be divided, narrowing the scope of resources the user selects, reducing the overhead of scheduling algorithms, the proposed scheduling algorithm is divided into two stages, with users as a unit to do overall scheduling, with task as unit to do cluster task scheduling [8].

Based on previous research, this method combined DAG task graph with game scheduling thought , trying to build a new algorithm that more suitable for cloud computing task scheduling to achieve better scheduling performance of cloud computing task.

2. Algorithm Design of this Article

In the cloud computing services, we usually put a service request submitted by users into multiple sub-tasks, and these sub-tasks are interdependent, that perform of tasks have certain sequences; we generally use DAG to describe such task.

The core idea of this algorithm design is: According to the dependencies between tasks in DAG figure, for cloud computing data center network design a network game model; in the model, all the sub-tasks seen as selfish participants in networks, compete limited virtual machine resources among them, and seek to maximize their own benefit; construct a task scheduling algorithm to solving the balanced state of game, solution results of that algorithm are equivalent with equilibrium state obtained by the task of free competition, extend this network game model, remove this assumption that "all participants are perfectly rational, so that more suitable to that condition that participants are limited rationality in realistic scenario.

2.1. DAG Graph and Cloud Computing Environments

DAG figure is a commonly used method described task structures in cloud computing. in a way of DAG task, the overall structure of the task can be divided into three levels: The first level is the root node, only one such node; second level is an intermediate node, this node can have multiple; third level is a leaf node, this node can only have one. Structure of a DAG task graph shown in Fig 1.

Figure 1: Structure of DAG Task Graph

Under this framework, there is a sequence of tasks, the root node to be the first execution, a leaf node to be the last execution, execution of intermediate nodes are not explicitly order requirements, but all the intermediate nodes must be executed before execution of leaf nodes.

Assuming cloud computing environment of this study, facing a distributed cloud computing scheduling tasks. In the cloud data center, there are multiple local scheduler and a task pool, all applications submitted by users requested is divided into various sub-tasks, these sub-tasks into in the tasks pool, Each sub-task has a task agents, these tasks agent can be freely selected to perform the task with which virtual machines of which

scheduler. These are communicating with each other between the scheduler and periodically obtain each other's load condition.

The actual task scheduling process is equivalent to such a problem, all users tasks are group participants, there is a game between these participants, they compete with each other about the limited virtual machines resources, the game finally reached an equilibrium state, i.e. without a participant can only change its strategy to increasing its utility selection.

2.2. Game Model of Task Scheduling

Based on the foregoing considerations, we first combine DAG task graph to build game model of cloud computing task scheduling. Task scheduling in cloud computing environment is to assigned appropriate the limited virtual machines for each sub-task. From the perspective of game theory is that the network participants - all tasks agents free compete virtual machine resources, we use game theory to design a two-stage game model for the task scheduling assignment.

First, root node and leaf nodes in DAG task graph will choose the virtual machine of the largest utility in current network. Then, nodes of the sub-tasks in middle layer selects virtual machines with the utilities largest on their own based on root node, leaf nodes decisions. In the course of the game, there may be multiple root node choose the same virtual machines, VM_o , there may be more than one leaf node selects the same virtual machine VM_D .

We make all choose that root node and leaf nodes may arise as a group, this choice is actually is a competition that intermediate node competing the virtual machines resource. Because this is the same general task service, so this competition is groups competition. Similarly, when there are multiple total tasks, there will be inter-group competition.

In the network game, competition policy for all subtasks of the middle layer in the network is to choose the virtual machines with the least costly, all participants competing the virtual machines in cloud computing data center; income of participants is the processing power of the virtual machine obtained, in this article, we make the summation of processing delay and marginal delay of virtual machine and as users' utility function, which is expressed as the specific formula:

$$C_i(vm) = - \left(t_i(vm_i) + vm \cdot \frac{dt_i(vm)}{dvm} \right) \quad (1)$$

Here, vm_i represents the i th virtual machines, $C_i(vm)$ represents the user utility function of the i -th virtual machine, $t_i(vm_i)$ represents processing delay of the i th virtual machine, as shown in the mathematical expression formula (2).

$$t_i(vm_i) = p \cdot vm_i + q \quad (2)$$

Based on principles of the user optimal and systems optimal of game equilibrium theory, each sub-task are selfish and rational, They have chosen the virtual machines the largest utility, eventually the game reach an equilibrium state, effectiveness of all the sub-tasks is the largest, that is no one sub-task can choose to change their strategy to increase their effectiveness. Ultimately, the game equilibrium problem becomes an optimization problem. For cloud computing tasks scheduling problem in this article, the most optimized mode as shown in equation (3), the objective function in Equation (4) below.

$$\min R(vm) = \sum_i \int_0^{vm_i} \left[t_i(vm) + vm \cdot \frac{dt_i(vm)}{dvm} \right] dvm \quad (3)$$

$$R(vm) = \sum_i vm_i \cdot t_i(vm_i) \quad (4)$$

2.3. Detailed Design of Task Scheduling Algorithm

If we assume that all participants in the network are fully obey the arrangement of system control center, we can design a tasks scheduling algorithm in the control center, the final results for solving algorithm are same with the state distribution resulting by all participants free compete. Therefore, the control center the task scheduling algorithm is as follows:

The first step, select a virtual machines with largest utility for each that root node and leaf nodes, respectively;

The second step, calculate the current processing delays and marginal delay each virtual machines, all the sub-tasks are assigned to virtual machines with minimum summation of the processing delay and marginal delay, to produce the amount of the virtual machines tasks vm_i , the iteration frequency is set as $n = 1$;

The third step, is to update the summation of the processing delay and marginal delay of each virtual machines and that is $t_i^n + vm_i^n \cdot \frac{dt_i(vm_i^n)}{dvm_i^n}$;

The fourth step, is to find the falling direction, based on the third step, make the amount of tasks assigned to the virtual machines having the smallest $t_i^n + vm_i^n \cdot \frac{dt_i(vm_i^n)}{dvm_i^n}$, resulting in the task amount of $dask_i^n$ in this case tasks;

The fifth step, by solving the optimization problem in equation (3) to determine the optimal iterative step, λ^n .

The sixth step, update the virtual machines assignments $vm_i^{n+1} = vm_i^n + \lambda^n (dask_i^n - vm_i^n)$.

The seventh step, check convergence and if it reaches the convergence condition $\frac{\sum_i (vm_i^{n+1} - vm_i^n)^2}{\sum_i (vm_i^n)^2} < \varepsilon$, then stop the iteration, this case the task amount of each virtual

machines is vm_i^{n+1} , if the convergence condition is not satisfied, turn into the second step, proceed to the second step to the first seven-step work.

3. Experimental Results and Analysis

In order to verify the effectiveness of the proposed scheduling algorithm, expand the experimental study below. Hardware environment used in the experiment: the computer is configured as CPU Core Duo 2.0G, memory 8G; experiment with the software environment, operating system with window XP, cloud computing simulation platform is CloudSim.

In experiment, set the number of the virtual machines is 20, each virtual machine configuration accordance with the formula (2), parameters a, b of each virtual machine as shown in Table 1.

Table 1. Parameters Configuration of Virtual Machine

VM	1	2	3	4	5
p	0.008	0.038	0.019	0.029	0.056
q	0.149	0.401	0.228	0.390	0.248
VM	6	7	8	9	10
p	0.039	0.040	0.029	0.028	0.039
q	0.588	0.491	0.478	0.292	0.547

VM	11	12	13	14	15
p	0.036	0.041	0.044	0.017	0.058
q	0.408	0.498	0.618	0.158	0.519
VM	16	17	18	19	20
p	0.039	0.078	0.063	0.084	0.025
q	0.318	0.692	0.701	0.818	0.177

Because this method concerned with the overall optimality of implementing the general tasks and consider in the game mechanism allows the virtual machine resources to be balanced loading task, so the first test is to test the number of virtual machines could undertake the task the algorithm is able to start when faced with different assignments, to examine its balance ability to undertake tasks.

In experiment, we let the total task began to increase from 100, and gradually increased to 600. Is shown in Figure 2, for the total 600, 20 virtual machines undertake the task situation.

Figure 2. For the Number of Tasks Is 600, the Number of Tasks That 20 Virtual Taken

As can be seen from the situation shown in Figure 2, in addition to the No. 1, No. 4, No. 11 virtual machine does not undertake tasks; the other 17 virtual machines have undertaken the tasks. This fully confirms the performance of the proposed method in terms of equilibrium scheduling, can make allocation of cloud computing tasks tend to the best condition, making the number of idle service resources as little as possible, so as to improve the efficiency of the entire cloud services.

Further testing the performance of the proposed method in this paper on the execution time is mainly the time delay of results this method returned of the implementation of different number of tasks. In order to facilitate the formation of intuitive contrast, this paper selects Maxmin method and Minmin methods on Cloudsim platforms, perform delay experiments with this method together, the experimental results shown in Figure 3.

Figure 3. Time Delay Test Results

As can be seen from Figure 3, the proposed method to complete the same amount of tasks takes the smallest time. Moreover, with the increasing number of tasks (from 100-600), the time delay curves of this method showed slowing growth trend, while the time delay curve that Maxmin method and Minmin method corresponds to is showing a trend of increasing growth.

4. Conclusion

Task scheduling is an important subject in cloud computing field, if all idle service resources as much as possible devoted to the process of completing the task, it is bound to improve the efficiency and quality of cloud services. Therefore, this article combine scheduling method of DAG task graph and game optimal ideas, to build a new cloud computing scheduling algorithm. This method based on the total amount of tasks firstly, to build a DAG task graph, that root node and leaf nodes set to the largest utility initial virtual machines; later, according to each virtual machines processing delays and marginal delay, combined with the game optimization model, through utility function ratio to constantly update setting of the initial virtual machine; when the utility difference is less than the threshold setted in advance, iterative update stops, formate ultimate task allocation of individual virtual machines.

Experimental results show that: the proposed method has good equilibrium scheduling performance, you can make more virtual machines resource serve for specific tasks in cloud computing; addition, this method has the advantage of a higher execution time than the two methods of Maxmin and Minmin, when performing tasks time delay is much smaller.

Acknowledgements

This work was supported by the National 863 high technology research and development projects (No.2013aa01a215), the colleges and universities youth backbone teachers funded projects in Henan province (No.2011GGJS-198) Science and technology key research projects in Henan province department of education (No.14a520045)

References

- [1] B. Saurabh and S. Santwana, "Workflow scheduling in cloud computing environment using cat swarm optimization", *Souvenir of the 2014 IEEE International Advance Computing Conference*, (2014), pp. 680-685.

- [2] J. Cao, J. Chen and Q. Zhao, "An optimized scheduling algorithm on a cloud workflow using a discrete particle swarm", *Cybernetics and Information Technology*, vol. 14, (2014), pp. 25-39.
- [3] B. Attila, D. J. Daniel and K. Attila, "Energy-aware VM scheduling in IaaS clouds using pliant logic", *Proceedings of the 4th International Conference on Cloud Computing and Services Science*, (2014), pp.519-526.
- [4] A. L. Kumar and V. Amandeep, "Workflow scheduling algorithms in cloud environment- A survey". *Recent Advances in Engineering and Computational Science*, vol. 22, (2014), pp. 44-51.
- [5] C. Rajesh and S. S. Kamath, "An equal share ant colony optimization algorithm for job shop scheduling adapted to cloud environments", *Lecture Notes in Electrical Engineering*, (2014), pp.81-92.
- [6] V. Amandeep and K. Sakshi, "Bi-Criteria priority based particle swarm optimization workflow scheduling algorithm for cloud", *Recent Advances in Engineering and Computational Science*, vol. 22, (2014), pp.101-109.
- [7] Q. Alfonso and D. Emanuele, "Scheduling strategies for enabling meteorological simulation on hybrid clouds". *Journal of Computational and Applied Mathematics*, vol. 273, (2013), pp.438-451.
- [8] S. Iman and P. Sandeep, "Achieving efficient distributed scheduling with message queues in the cloud for many-task computing and high-performance computing", *14th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, (2014), pp. 404-413.

Authors

Liu Jun, Lecturer, Henan Mechanical and Electrical Engineering College. Born in 1981, Mr. Liu graduated and got master degree from Wuhan University of Technology, and his main research interests are Database Technology and Cloud Computing.

Guo Zuhua, Associate Professor, Henan Mechanical and Electrical Engineering College. Born in 1977, Mrs Guo graduated and got master degree from Zhengzhou University and her main research interests are Information security and Algorithm research.