

Research on Batch Scheduling in Cloud Computing

Jintao Jiao^{*1}, Wensen Yu², Lei Guo³

¹School of Mathematics and Computer Science, Wuyi University, 354300, Wuyishan, China

²The Key Laboratory of Cognitive Computing and Intelligent Information Processing of Fujian Education Institutions, 354300, Wuyishan, China

³Collaborative Innovation Center of Chinese Oolong Tea Industry, 354300, Wuyishan, China

*Corresponding author, e_mail: jiaojintao@163.com

Abstract

In the existing cloud computing environment, batch scheduling strategies mainly focus on the management of resources allocation. This paper provides the task scheduling algorithm based on service quality which fully considers priority and scheduling deadline. The improved algorithm combines the advantages of Min-min algorithm with higher throughput and linear programming with global optimization, considers not only all the tasks but also the high priority tasks. The experiment result shows that compared with the Min-min and DBCT the completed tasks of the improved algorithm increase about 10.6% and 22.0%, on the other hand the completed high priority tasks also increases approximately 20% and 40%.

Keywords: cloud computing, scheduling strategy, batch scheduling

Copyright © 2016 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Cloud computing is a new business model, and it is a product of information technology [1]. In cloud computing market, user scale is very large, and the QoS(Quality of Service) requests of each task are not identical. These factors make task schedule in cloud computing very complicated. So it is a technical difficulty to schedule the massive tasks in cloud computing in order to get a better scheduling results with the constrain of budget and deadline.

The main batch scheduling algorithms are Min-min [2] and DBCT [3] (under the deadline and budget constrain time optimization algorithm). The idea of Min-min algorithm is that the best resources will be allocated to the minimum task, thus the system throughput rate can be improved and the minimum time of the job can be obtained. While the idea of DBCT is to schedule the task on the resource that can complete the task earliest, also request the cost no more than user's budget and the finish time no more than deadline. With the popularity of cloud computing, some users with less budget may also want to use cloud computing; also there are some tasks with sufficient budget and intensive demand of deadline. For these cases, the number of completed tasks of Min-min and DBCT algorithms will be less. There are also a lot of people put forward other algorithms, but some did not consider the deadline [4], some did not consider the user's budget [5]; they did not balance the relationship between the deadline and the budget.

From the perspective of QoS, this paper proposes DBCMNT (deadline and budget constrained maximized number of completed tasks scheduling algorithm) algorithm in order to meet the budget and deadline constrains. Compared with Min-min and DBCT algorithm, DBCMNT algorithm can accomplish higher priority tasks better, at the same time the new algorithm can maximize the number of completed tasks. So the user's demands can be meet; on the other hand, the cloud computing can attract more consumers and get more benefits.

2. Cloud Computing Scheduling Market Model

Cloud computing scheduling market model[6-7]could manage and evaluate resources allocation more effectively. The model brings four benefits: (1) user's fair use of computing resources [8], (2) adjusting the balance of supply and demand of resources in cloud computing. When demand exceeds supply, raising the price of resources could help to reduce the number of users and tasks, on the other hand, when supply exceeds demand, lower price could help to

attract more users [9], (3) providing quality of service to users, such as task deadline, cost to complete the task, security of resources [10], (4) providing the effective resource management and allocation mechanism [11].

The model contains client, broker, resources, resources supporter and information service, the architecture shows in Figure 1.

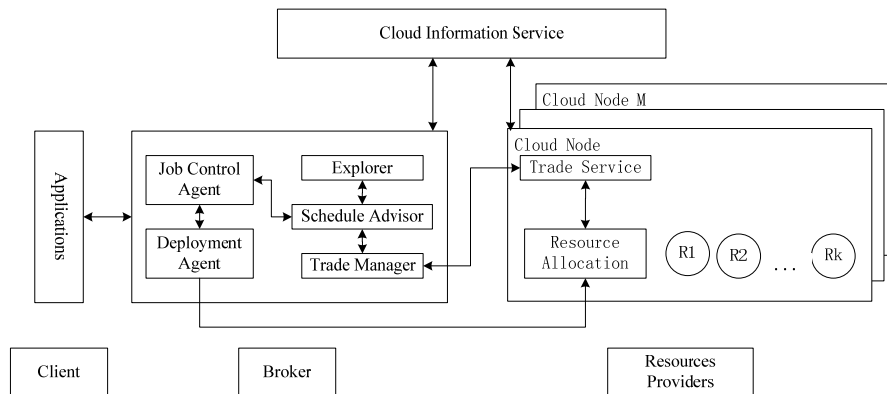


Figure. 1 Cloud Computing Scheduling Model

Usually the tasks that needed to be run can be divided into serial applications, parallel applications, parameter scanning applications, collaborative applications and etc. The system allows users to set the resource requirements and preferences for parameters, and the client pay for the using resources.

Broker is the intermediate interface between client and resource, function of which is to discover resources, select resources, receive tasks, return scheduling results and exchange information. And broker supports different scheduling policies which can find resource and schedule tasks according to user's demands. Broker is mainly composed of job control agent, schedule advisor, explorer, trade manager and deployment agent.

There is a variety of computing resources in cloud computing, resource has the attribute of computing performance and computing price, performance is measured by MIPS (Million Instructions Per Second), price is measured by unit time cost G\$. Generally the better the resource performance, the higher the price will be.

Information service mainly records available resources. Broker will query information service when searches for appropriate resources, then interact with resources providers after getting information of resources that meet user's demand. If resources providers have new resource to lease, they must register the resources information in information service so that broker could find it.

In the process of transaction between client and resources providers, resources providers register the resources information in the information service. After client submits a task to broker, broker will get available resources information from information service, and then schedules the task to the appropriate resources according to the scheduling algorithm. Broker also estimates the completion time and the cost before the task is executed. If the time exceeds the deadline or the cost is higher than user's budget, broker will refuse to accept the task. If the task is executed successfully, broker will return results to the client and obtain the profit, otherwise return the error information.

3. Description of Scheduling Tasks

In the cloud computing environment, resources provider can benefit by leasing computing resources, meanwhile client can complete the task of quality requirements by purchasing the right to use resources. We assume that the task budget and deadline is limited, client requires total cost within budget and hope as much as possible task could be completed

under the premise of task completion time before deadline. If client has sufficient budget and long enough deadline, the tasks would be completed within budget and deadline.

Due to the budget is less or the task is urgent, maybe some of the tasks could be completed. So, this paper mainly discusses how to maximize the number of tasks completed. Also, this paper considers the task priority under special circumstances, because client hopes that urgent and important tasks should be completed in time, while the rest of the tasks can be delayed.

With the above assumptions, client separates the tasks into two kinds by priority: the higher priority tasks and the lower priority tasks. This paper uses HS to represent the higher priority tasks and LS to represent the lower priority tasks. Figure 2 shows the deadline of the HS and LS tasks. We can figure out that the HS tasks must be completed before $T = T_{HS}$, and the LS tasks must be completed before $T = T_{LS}$, so the tasks are formalized as (HS, T_{HS}) and (LS, T_{LS}) . If tasks type is $(0, T_{HS})$ and (LS, T_{LS}) , it means that there are no HS tasks, as many tasks as possible should be completed within budget and before T_{LS} .

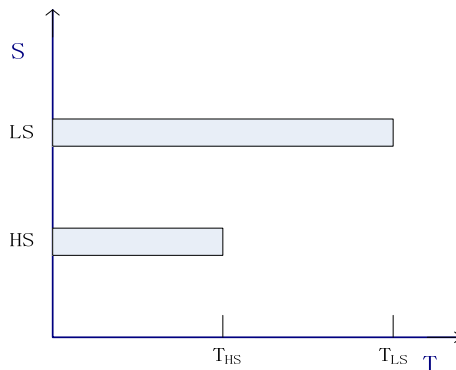


Figure. 2 Deadline of HS and LS tasks

4. Task Scheduling Algorithm Based on Quality of Service

In cloud computing market, resources supporter can't complete all tasks when client requires less budget or shorter time than needed. So this paper proposes the task scheduling algorithm based on quality of service to solve the problem.

4.1. Deadline and Budget Constrained Maximum Number of Completed Tasks(DBCMNCT)

Suppose that there are m machines $\{r_1, r_2, \dots, r_m\}$, n tasks $\{e_1, e_2, \dots, e_n\}$ and the following variables in the cloud computing environment.

Table. 1 Definition of Variables

variable	definition
n	number of tasks
m	number of computing resources
L_i	length of task i
B	task budget
Re_j	computing performance of resource j
P_j	price of resource j

According to the client's requirements, HS tasks should be completed before T_{HS} , LS tasks should be completed before T_{LS} . In order to ensure that the high priority tasks HS can be completed according to clients' demand good resources should be allocated to short tasks, so that more tasks can be completed in a period of time. Based on the two kinds of tasks, DBCMNCT gives higher priority to the allocation of resources to the HS tasks through Min-min algorithm. Because each machine performance is different and the allocation of the task size and number are not the same, so the completion time and cost of each machine will be different. If t_j and b_j are used to represent the completion time and cost when machine j is used to complete one of the HS tasks and b represent the service cost of all the HS tasks,

$$b = \sum_{j=1}^m b_j$$

However, if the budget is limited or the HS tasks deadline is urgent, part of the HS tasks may not be completed, then the uncompleted HS tasks should be added to the LS tasks set. When executing the LS tasks, a linear planning model LPM is used to get the mapping relationship between LS tasks and resources. The definition of task structure contains an attribute of sort, when sort=1 the task type is LS, when sort=2 the task type is HS.

There are some assumptions in the scheduling process:

Tasks in execution cannot be seized. Once the service provider begin to execute a task, it cannot interrupt the current task to perform others until the task is completed.

There is no priority for the same type tasks, the client only concerns about the number of completed tasks before deadline.

So the algorithm of deadline and budget constrained maximum number of completed tasks (DBCMNCT) is shown in algorithm 1.

Algorithm 1: DBCMNCT

Step 1: $S \leftarrow$ the set of all tasks;

Step 2: $S_{HS} \leftarrow$ the set of tasks that sort=1; $S_{LS} \leftarrow$ the set of tasks that sort=2;

Step 3: initialize $T_{HS}, T_{LS}, t_j, b_j, B, b$;

Step 4: if ($S_{HS} = \phi$) goto step 6;

Step 5: estimate the number of completed HS tasks by Min-min algorithm, return b, t_j ;

Step 6: $S_{LS} = S_{LS} \cup \{e_i \mid e_i \text{ uncompleted} \wedge e_i \in S_{HS}\}$;

Step 7: call LPM($S_{LS}, T_{LS}, t_i, B, b$) module, estimate the number of completed LS tasks and return the mapping relationship between tasks and resources;

Step 8: according to the mapping relationship, schedule tasks to the corresponding resources to performance;

Step 9: accept the schedule result returned from the resources provider;

Step 10: end;

4.2. LPM Module

The matrix A_{nm} is defined in formula 1,

$$A_{nm} = \begin{pmatrix} a_{11}, a_{12}, \dots, a_{1m} \\ a_{21}, a_{22}, \dots, a_{2m} \\ \dots \\ a_{n1}, a_{n2}, \dots, a_{nm} \end{pmatrix} \quad (1)$$

$$a_{ij} = \begin{cases} 1 & \text{if task } i \text{ is scheduled to resource } j; \\ 0 & \text{otherwise}; \end{cases}$$

Each task is executed at most on one resource, so there is formula 2.

$$\sum_{j=1}^m a_{ij} \leq 1, i = 1, 2, \dots, n, e_i \in S_{LS} \quad (2)$$

Before the execution of LS tasks, each resource has already worked for t_j and gained service cost b due to the execution of HS tasks. Based on deadline T_{LS} and the available budget left $B-b$, there are formula 3 and formula 4.

$$\sum_{i=1}^n a_{ij} * L_i / Re_j \leq T_{LS} - t_j, j = 1, 2, \dots, m, e_i \in S_{LS} \quad (3)$$

$$\sum_{i=1}^n \sum_{j=1}^m a_{ij} * L_i * P_j / Re_j \leq B - b, e_i \in S_{LS} \quad (4)$$

When executing the LS tasks, the scheduling objective is to maximize the number of completed tasks, so there is formula 5.

$$\max = \sum_{i=1}^n \sum_{j=1}^m a_{ij}, e_i \in S_{LS} \quad (5)$$

According to formula 2 to formula 5, a linear planning model can be established, LPM algorithm can be used to solve the LPM module in algorithm 1, as shown in algorithm 2.

$$\begin{aligned} \max &= \sum_{i=1}^n \sum_{j=1}^m a_{ij} \\ \text{s.t.} & \\ &\sum_{j=1}^m a_{ij} \leq 1, i = 1, \dots, n \\ &\sum_{i=1}^n a_{ij} * L_i / Re_j \leq T_{LS} - t_j, j = 1, \dots, m \\ &\sum_{i=1}^n \sum_{j=1}^m a_{ij} * L_i * P_j / Re_j \leq B - b \\ &a_{ij} = 0 \text{ or } 1, i = 1, 2, \dots, n, j = 1, 2, \dots, m \\ &e_i \in S_{LS} \end{aligned}$$

Algorithm 2: LPM

Step 1: $M \leftarrow$ the set of all machines;

Step 2: initialize $A_{nm}, R_{e_i}, P_j, L_i, T_{LS}, t_j, B, b$;

Step 3: establish and calculate the linear planning model, return A_{nm} ;

Step 4: for($i=1; i \leq n; i++$)
for($j=1; j \leq m; j++$) {

```

if( $a_{ij} == 1$  &&  $e_j \in S_{LS}$ ) schedule task  $i$  to machine  $j$ ;
else continue;

```

```

}

```

Step 5: return the expected completion time and mapping result;

Step 6: end;

5. Experiment and Result Analysis

Two experiments are designed in order to test the proposed DBCMNT algorithm. The simulation platform is Cloudsim [12-13]. Cloudsim is an event driven cloud computing simulation toolkit based on Java; its main goal is to research in the effective resource allocation method based on the computing economy model. In the designed cloud computing market, the price is proportional to the computing ability.

5.1. Experiment without HS Tasks

The first experiment is designed to test the total number of completed tasks under different budget and deadline when there are no HS tasks. And the size of each task is distributed randomly between. between 600 MI(Million Instruction) and 12600 MI. There are 7 parameter group of budget and deadline; the parameter is increased step by step, such as (450000,5000) (700000,7000) (850000,9000) (1000000,10000) (1150000,13000) (1300000,16000) (1450000,19000).

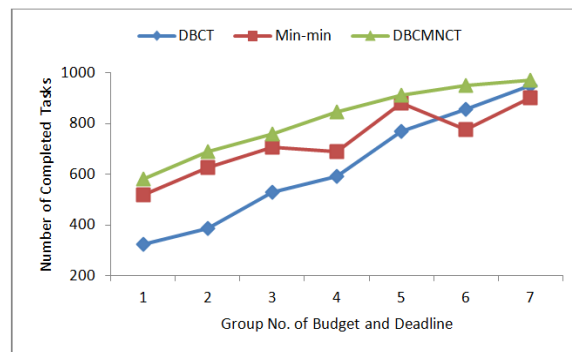


Figure. 3 Number of Completed Tasks Corresponding to Each Group of Budget and Deadline

Figure 3 shows the result of the first experiments. With the increase of budget and deadline the number of tasks completed by Min-min algorithm is increased, but not absolute; the completed tasks of group 4 is less than group 3, also the completed tasks of group 6 is less than group 5; the budget and deadline is increased while the number of completed tasks is decreased. The main reason is when some of the tasks cannot be completed, the budget and deadline will have an impact on the schedule of tasks that the number of completed tasks is decreased. In comparison, the number of completed tasks of DBCMNT and DBCT algorithm is increased with the increment of budget and deadline. For each group of budget and deadline, the performance of DBCMNT is better than that of DBCT and Min-min, there are about 22% and 10.06% performance improvement.

5.2. Experiment with HS Tasks

The proportion of HS tasks in the second experiment changes from 10% to 80%; and the number of all tasks is randomly distributed from 500 to 1000.

Figure 4 shows that the HS tasks completion rate of DBCMNT algorithm is much more higher than the other two algorithms, this is mainly because the priority is given to HS tasks in DBCMNT algorithm while not in the other two algorithms. This means more urgent tasks can be completed with DBCMNT algorithm than the other two.

Figure 5 shows that with the increase of the HS tasks, the total tasks completion rate of Min-min and DBCT algorithm is relatively stable. And the total tasks completion rate of DBCMNET algorithm is higher than the other two algorithms if the proportion of HS tasks is less than 50%, but when the proportion of HS tasks is more than 50% the performance of DBCMNET algorithm is worse than the other two. This is mainly because the HS tasks will occupy the good resources, so the completion is affected with the increase of the HS tasks. In the actual cloud computing system the number of HS tasks should be limited so that the performance could be better.

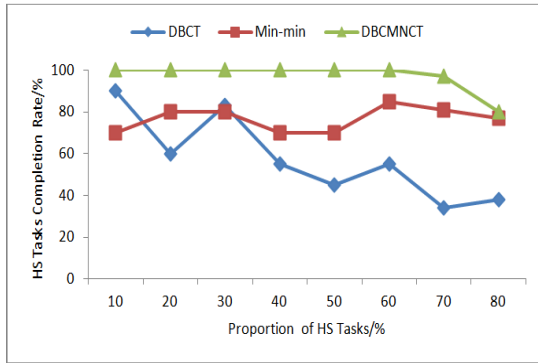


Figure. 4 HS Tasks Completion Rate with Different Proportion of HS Tasks

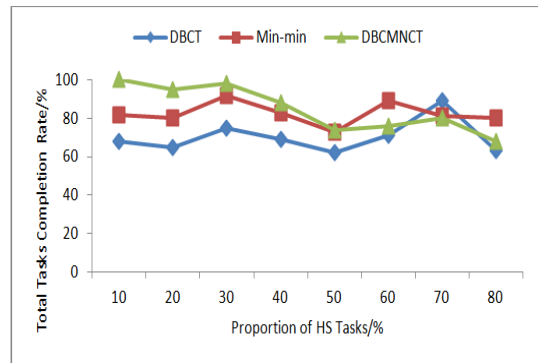


Figure. 5 Total Tasks Completion Rate with Different Proportion of HS Tasks

5.3. LPM Algorithm Performance Analysis

The branch and bound algorithm [14] is used to solve the linear planning model in LPM Algorithm, the time and space costs of the whole algorithm are also mainly consumed in solving the linear planning model. Table 2 and Table 3 show that the time and space costs are increased when the number of tasks is increased.

Table 2. Time and Space Cost with Different Number of Tasks

Number of Tasks	50	100	500	1000
Time Cost(ms)	10	10	20	40
Space Cost(K)	101	192	922	1825

Table 3. Time and Space Cost with Different Number of Resources

Number of Resources	50	100	500	1000
Time Cost(ms)	10	20	20	40
Space Cost(K)	866	1704	5100	8421

The time cost of the algorithm is in the millisecond level, which is far less than the execution time of the task, so the time cost of the scheduling algorithm can be ignored. The space cost is increased linearly, so the number of tasks or resources should be limited.

Although the linear planning model is a NP problem, it has already been used in other scheduling algorithm. Nowadays there are a lot of research results can make the time complexity of serial algorithm control in polynomial level. And with the improvement of parallel technology, the serial linear planning algorithm can be easily parallelized, which also greatly reduces the time and space costs.

6. Conclusion

This paper discusses how to maximize the number of completed tasks with constrain of budget and deadline; also when there are some urgent or important tasks, the algorithm can

ensure these tasks completed according to user's demand. HS tasks will be allocated on good resources; then higher HS task completion rate and total task completion rate can be achieved by using the DBCMNET algorithm based on LPM.

In this paper, we pay more attention to the user. Henceforth by combining the character of the system itself, we will continue to improve the algorithm in order to make it more comprehensive, more widely used and more simple to use.

Acknowledgements

This work was supported by Fund of Fujian Education Institutions(JB14101),Fund of Wuyi University(XD201406) and research project of university teaching reform in Fujian Province (JAS151342).

References

- [1] Shuo Wang, Shihong Liu. Key Technology of Agricultural Production and Market Information Matching in Big Data Era. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(3): 2212-2218.
- [2] Patel G, Mehta R, Bhoi U. *Enhanced Load Balanced Min-min Algorithm for Static Meta Task Scheduling in Cloud Computing*. *Procedia Computer Science*. 2015; 57: 545-553.
- [3] Chintapalli, Venkatarami Reddy. A deadline and budget constrained cost and time optimization algorithm for cloud computing. *Communications in Computer and Information Science*. 2011; 193(4): 455-462.
- [4] Sujan S, Devi RK. *A batchmode dynamic scheduling scheme for cloud computing*. *Proceedings of 2015 Global Conference on Communication Technologies*. Nagercoil. 2015: 297-302.
- [5] Fan Yuanyuan, Liang Qingzhong, Chen Yunsong, Yan Xuesong, Hu Chengyu, Yao Hong, Liu Chao, Zeng Deze. Executing time and cost-aware task scheduling in hybrid cloud using a modified DE algorithm. *Communications in Computer and Information Science*. 2016; 575: 74-83.
- [6] Rabi Prasad Padhy, Manas Ranjan Patra. Architecture & Design of Affordable and Highly Available Enterprise Cloud Service. *International Journal of Cloud Computing and Services Science*. 2013; 2(2): 85-105.
- [7] Hongsheng Su, Ying Qi. A chaos cloud particle swarm algorithm based available transfer capability. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(1): 38-47.
- [8] Mazalov V, Lukyanenko A, Luukkainen S. Equilibrium in cloud computing market. *Performance Evaluation*. 2015; 92: 40-50.
- [9] Hamsanandhini S, Mohana RS. *Maximizing the revenue with client classification in Cloud Computing market*. 2015 International Conference on Computer Communication and Informatics. Coimbatore. 2015: 1-7.
- [10] Mansouri Dou El Kefel, Benyettou Mohammed. Price optimisation of cloud computing service in a monopolistic competitive market. *Multiagent and Grid Systems*. 2016; 11(4): 259-271.
- [11] Xu Bolei, Qin Tao, Qiu Guoping, Liu Tie-Yan. *Competitive pricing for cloud computing in an evolutionary market*. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*. Istanbul. 2015; 3: 1755-1756.
- [12] Rodrigo N. CloudSim: A Novel Framework for the Modeling and Simulation of Cloud Computing Infrastructures and Services. The University of Melbourne Australia. 2009.
- [13] RN Calheiros, R Ranjan, A Beloglazov, CAF De Rose, R Buyya. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software Practice and Experience*. 2011; 41(1): 23-50.
- [14] Jin Fang, Yuan Zhi-hua, Liu Qing-Quan, Xing Zhang. Quantized feedback control of network empowerment ammunition with data-rate limitations. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2014; 12(3): 525-532.