A Benefit-driven Task Scheduling Algorithm based on Genetic Algorithm in Cloud Computing

ZHAO Jie

Mudanjiang Normal University, China

zhaojiezhjie@sina.com

Keywords: Clouding computing, Genetic algorithm, Benefit drive, Task scheduling.

Abstract. To solve the benefit-driven task scheduling problem in clouding computing, a benefit-driven task scheduling algorithm based on genetic algorithm is proposed. Under the precondition of meeting the QoS constraints, the proposed algorithm takes computation overhead, service profit and delay compensation into account. And the mathematical model of the benefit-driven task scheduling problem is introduced. Meanwhile, genetic algorithm is used to solve the he benefit-driven task scheduling problem. Simulation demonstrates that compared with the Min-Min algorithm and QoS Min-Min algorithm, the proposed algorithm can significantly reduce the scheduling completion time, balance the load and improve the profit per unit computing cost.

Introduction

Clients do not need to purchase and deploy IT infrastructure, and they only need to rent resources of cloud computing system according to own demand in cloud computing system. Cloud computing system can provide task processing and data storage [1]. People put forward different task scheduling algorithms in order to better deal with different tasks, ensure that the system can satisfy QoS (Quality of Service) requirement of client, such as taboo search algorithm, multiple QoS constrained task scheduling algorithm, Min-Min algorithm, QoS Min-Min algorithm, etc. [2,3]. However, cloud computing, as commercial service, not only should meet QoS requirements of various tasks submitted by the client, but also should maximize profits of cloud service provider, namely, maximization of unit cost benefit. However, existing cloud computing task scheduling algorithm only starts from the perspective of clients and aims at satisfying client task scheduling QoS requirements without consideration of service benefits. Therefore, Benefit-driven Task Scheduling Algorithm based on Genetic Algorithm (BTSAGA) is proposed from the perspective of cloud service provider in the paper.

Mathematic modeling of task scheduling

It is assumed that one scheduling task can only be assigned to a compute node for performance in order to facilitate analysis. The system does not support to be preempted. Next, meanings of related symbols for BTSAGA are firstly introduced.

$C = \{c_1, c_2, \cdots, c_n\}$: cloud computing resource pool，n for calculating node number;

$T = \{t_1, t_2, \cdots, t_m\}$: to-be-scheduled task set, m for to-be-scheduled task quantity;

X: $m \times n$ task scheduling distribution matrix, if t_i ($t_i \in T$) is assigned to c_j ($c_j \in C$), corresponding position element of matrix X is $x_{i,j}$=1, otherwise $x_{i,j}$=0;

T_finish_i : anticipated time for completing t_i ($t_i \in T$);

I_count_i : t_i ($t_i \in T$) scale, namely instruction quantity contained in t_i;

$I\_\cos t_i$: c_j ($c_j \in C$) cost for performing one instruction;

$Deadline_i$: t_i ($t_i \in T$) completion time limit, if cloud service provider can complete task t_i, within the time limit, benefit can be obtained. Otherwise client loss should be compensated;

$D\_\cos t_i$: amount that should be paid to client within unit time when the cloud service provider fails to complete t_i ($t_i \in T$) before $Deadline_i$;

$Budget_i$: benefit that can be obtained by cloud service provider when cloud service provider completes t_i ($t_i \in T$) before $Deadline_i$;

L_line_i : the latest completion time of t_i ($t_i \in T$) that can be accepted by client;

$Fine_j^i$: liquidated damages that should be paid to client by cloud service provider due to delay when c_j ($c_j \in C$) is used for treating t_i ($t_i \in T$);

$Comp\_\cos t$: computing cost of to-be-scheduled task set T;

$T\_benefit$: benefit from to-be-scheduled task set T;

$Worline_i$: when time for cloud service provider to complete t_i ($t_i \in T$) exceeds L_line_i , the provider should pay liquidated damage to clients. In the paper, it is assumed that when $T\_finish_i > Worline_i$, cloud service provider should pay liquidated damages of $Budget_i$ to client.

In the process of task scheduling, the client should firstly evaluate the urgency of task and combine with expense budget situation to determine QoS requirements of the task, and then task scheduling request should be submitted to cloud service provider. Cloud service provider can estimate the benefits that can be obtained according to QoS requirements of the task and the usage of own cloud computing resources, thereby deciding whether the task scheduling request can be accepted or not. After the request is accepted, once cloud service provider fails to complete the task scheduling within the prescribed time, the provider should compensate certain loss for the client, and the compensation amount is determined by delay time and D_cost. Both client and cloud service provider hope to complete the task before L_line. Otherwise, Qos can not be guaranteed, thereby causing loss on client. Meanwhile, cloud service provider also should compensate corresponding liquidated damages to client.

After cloud service provider accepts client's request for scheduling tasks t_i ($t_i \in T$), t_i is handled by c_j ($c_j \in C$) in the cloud computing resource pool, namely $x_{i,j}$=1. Then, cost for c_j is shown as follows aiming at t_i:

$$Comp\_\cos t_j^i = I\_\cos t_j \times I\_count_i \tag{1}$$

Accordingly, cloud service provider can obtain benefit $Budget_i$- $Fine_j^i$ from the task t_i ($t_i \in T$), and $Fine_j^i$ expression is shown as follows:

$$Fine_j^i = \begin{cases} 0 \\ D\_\cos t_i \times (T\_finish_i - Deadline_i), Deadline_i < T\_finish_i < L\_line_i \\ Budget_i, T\_finish_i \geq L\_line_i \end{cases} \tag{2}$$

Obviously, $T\_benefit = \sum_{j=1}^{n}\sum_{i=1}^{m}(Budget_i - Fine_j^i) \times x_{i,j}$ aiming at T, while corresponding computing cost is $Comp\_\cos t = \sum_{j=1}^{n}\sum_{i}^{m} Comp\_\cos t_j^i \times x_{i,j}$.

Cloud computing services provided by cloud service provider is still business service. Pursuit of commercial service benefit maximization can embody businessman profit pursuit. Cloud service provider expects that the most value can be produced aiming at limited cloud computing resources. Therefore, task scheduling algorithm should be able to maximally improve the benefits brought by the unit cost overhead. Then, task scheduling algorithm has the following scheduling purpose:

$$target = \max\left(\frac{T\_benefit}{Comp\_\cos t}\right), \quad \text{s.t.} \quad T\_finish_i < L\_line_i \tag{3}$$

Benefit-driven task scheduling algorithm based on genetic algorithm

It can be seen from the above analysis that there are a total of n^m scheduling modes for m to-be-scheduled tasks and n computing resources. It is impossible to obtain optimal solution of formula (3) within polynomial time. Therefore, the task scheduling is an NP - hard issue.

Because task scheduling belongs to NP - hard issue, heuristic algorithm BTSAGA is utilized for solving task scheduling problem in the paper. Figure 1 shows BTSAGA workflow. It can be seen from figure 1 that the client firstly submits a task scheduling request to pretreatment unit. Secondly, pretreatment unit can estimate according to task properties and QoS scale to task. Next, task registration request is sent to scheduler. Meanwhile, task relevant information expected forecast value is sent to the scheduler. Scheduler can judge whether the request can be accepted or not according to related information of the task and usage condition of cloud computing resource pool. If the request is accepted, related information of the task and usage condition of the cloud computing resource pool can be sent to task scheduling decision-making unit by the scheduler. Genetic algorithm is used in the unit. Usage condition of current resource pool and related information of the task can be combined for solving task scheduling problem. The obtained task scheduling plan can be fed back to scheduler. Then, scheduler can appointed corresponding computing nodes in the cloud computing resource pool for handing task according to task scheduling plan. Finally, handling result can be fed back to scheduler by cloud computing resource pool. In addition, once resource information in cloud computing resource pool is changed, related information can be sent to scheduler for scheduler to use.

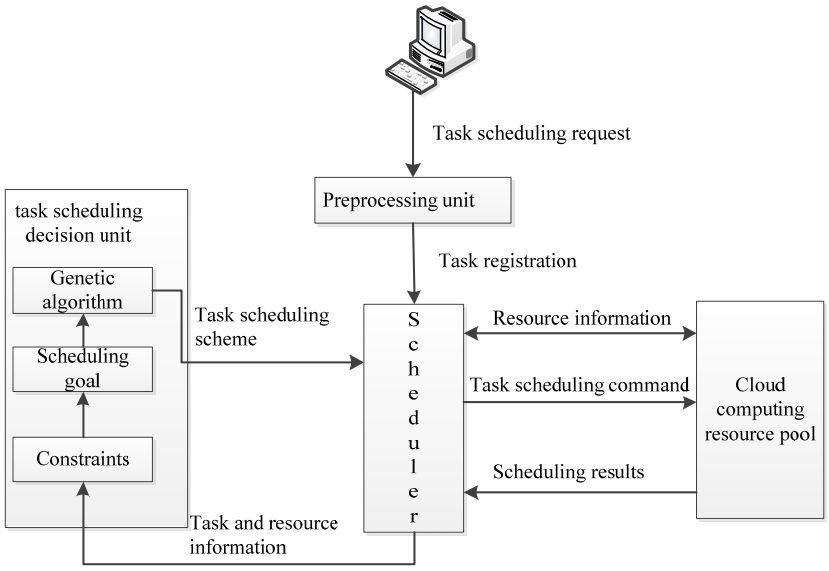

Figure 1. BTSAGA flow chart

In figure 1, genetic algorithm is used in solving scheduling objective. Its solving process is shown in figure 2. It can be seen from figure 2 that genetic algorithm involves generation of initial population, genetic iterative operation, fitness value calculation and determination of iterative terminating condition. The will be introduced respectively next.

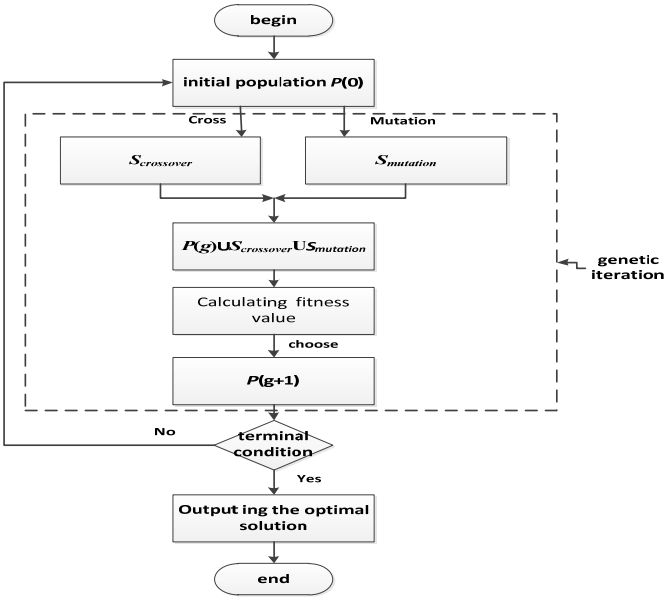

Figure 2. Flow chart of solving scheduling problem by genetic algorithm

Initial population.

First of all, some encoding method should be adopted for converting scheduling scheme into corresponding chromosome so that genetic algorithm can implement subsequent genetic iteration operations in order to use the genetic algorithm for solving scheduling problems. In BTSAGA, binary code is used in chromosome encoding. According to the content in the second section, it is obvious that a scheduling task can only be assigned to a compute node for performance, so only one factor can be equal to 1 in each line of matrix X. Next, n=4， m＝6 is adopted as an example for explaining how to generate corresponding chromosome with binary coding. Concrete encoding process is given in Figure 3.

$$X=\begin{bmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\\1&0&0&0\\0&1&0&0\end{bmatrix}\rightarrow\begin{bmatrix}10\\11\\00\\01\\00\\01\end{bmatrix}\rightarrow(101100010001)$$

Figure 3 Chromosome encoding process

It can be seen from figure 3 that t_1 is assigned to c_3, 10 is used for representing $t_1 \rightarrow c_3$ during binary encoding, and so on, thereby obtained required chromosomes (101100010001). It is assumed that population size is N, scheduling methods in the same quantity are generated in a random mode. The above binary encoding method is used for transforming coding scheme into corresponding chromosome, thereby obtaining initial population $P(0)$.

Fitness function.

Fitness function is used for measuring the quality of chromosomes, namely fitness value of chromosome is larger, its corresponding scheduling plan has higher benefits, and it is more impossible to be retained in genetic iterations. In BTSAGA, benefit of unit computing cost is regarded as fitness function, and the expression is shown as follows:

$$fitness=\frac{T\_benefit}{Comp\_cost} \tag{4}$$

Genetic iteration.

After initial population P (0) is generated, mutation, crossover and selection genetic operators should be used for iterative evolution of chromosome in population, the procedure will be repeated for many times until the evolved population index reaches the preset goals or maximum iteration frequency.

Variation: firstly, several chromosomes can be randomly selected from current population P (g), and genes in the selected chromosomes can undergo mutation with mutation probability p_m. Finally the new generated chromosome can be placed into set $S_{mutation}$.

Crossover: first of all, two chromosomes are randomly selected from current population P (g) with crossover probability pc. A section of gene is randomly selected for crossover from two chromosomes, namely chromosome cross breeding. Finally, the new generated chromosome is placed in set $S_{crossover}$.

Choice: firstly, chromosomes after crossover and mutation as well as current population are combined, thereby obtaining P(g)U$S_{crossover}$U$S_{mutation}$. Fitness value of all chromosomes in the set is calculated. They are sequenced according to the order large to small. Finally, the retained first N chromosomes are regarded as next generation of population P (g+1).

Iteration terminating conditions.

Firstly, formula (3) is utilized for assessing new population P(g+1). If iteration frequency reaches the largest iteration frequency, or the fitness function value of chromosomes in a population is less than the preset value in multiple iterations, iteration can be terminated. The chromosome with the largest fitness function value is output. The algorithm is terminated; Otherwise, the above genetic iterations are repeated.

Simulation analysis

Simulation parameter settings.

It is assumed that there are a total of 10 computing nodes in cloud computing resource pool, namely n=10. The processing unit number, computing cost and computing efficiency of each computing node are shown in Table 1.

There are a total of 10 different attribute tasks in the process of task scheduling. Attribute setup of 10 tasks is shown in Table 2.

Table 1. Cloud computing resource list

No.	1	2	3	4	5	6	7	8	9	10
Processing unit quantity	5	5	10	10	15	15	20	20	30	30
Computing cost efficiency	6	8	10	12	14	16	18	20	22	24
Computing efficiency	23	28	33	38	43	48	53	58	63	68

Table 2. Task Attribute Table

No.	1	2	3	4	5	6	7	8	9	10
Budget	160	170	250	250	270	310	320	370	380	410
I_count	210	240	300	340	390	460	510	560	610	630
Dealine	9	10	10	15	18	25	25	30	30	40
D_cost	10	10	12	12	14	14	16	16	18	18
L_line	15	16	16	25	30	36	36	40	45	60

Meanwhile, it is assumed that population size is N = 100, mutation probability is p_m=0.16, crossover probability is p_c＝0.75, the maximum number of iterations is 200 aiming at genetic algorithm in BTSAGA, objective function minimum improvement rate is 0.2% after population iteration for three times.

Analysis on simulation results.

In the section, performances of BTSAGA, Min-Min algorithm and QoS Min-Min algorithm are compared from the aspects of schedule completion time and unit cost efficiency based on the above simulation reference. In the simulation, the task submitted by client randomly is selected randomly from table 2. Each group of simulation experiment is repeated for 100 times, thereby eventually getting the average value of simulation experiment results for 100 times.

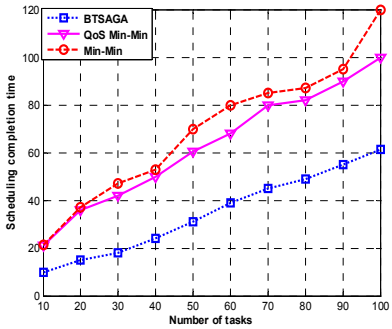

Figure 4. Schedule completion time vs. task quantity

Figure 4 shows the changing condition of schedule completion time with task quantity. It can be seen from the figure 4 that the schedule completion time of three algorithms is increased with task quantity increase. Meanwhile, the schedule completion time that should be completed by QoS Min - Min algorithm should be less than Min - Min algorithm because Deadline is considered during schedule decision-making by QoS Min - Min algorithm compared with Min - Min algorithms, all tasks can be scheduled before Deadline as far as possible, thereby effectively reducing task quantities exceeding deadline. Therefore, the performance of QoS Min-Min algorithm is better than Min-Min algorithm. Figure 4 shows that schedule completion time of BTSAGA is the shortest since BTSAGA aims at maximizing unit cost benefits. Factors in all aspects are comprehensively considered, it tends to assign the task to efficient and cheap computing resources. Min-Min algorithm and QoS Min-Min algorithm tend to assign tasks to the computing resource with the fastest processing speed, thereby leading to uneven load, increasing waiting time of some tasks, and increasing schedule completion time as a whole.

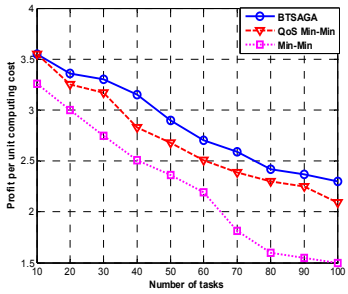

Figure 5. Unit cost benefit vs number vs. task quantity

Figure 5 shows the change condition of unit cost benefits with task quantity. It can be seen from figure 5 that BTSAGA unit cost benefit is higher than Min-Min algorithm and QoS Min-Min algorithm because BTSAGA task scheduling is more reasonable with more balanced load. Waiting time of task scheduling is effectively reduced. Figure 5 shows that unit cost benefits of three algorithms are also decreased with gradual increase of task quantity since resource competition can be aggravated due to increase of task quantity under the condition of fixed computing resources, thereby increasing service timeout phenomena and resulting in reduced unit cost benefit.

Conclusion

The paper starts from the perspective of cloud service provider. Benefit-driven task scheduling algorithm based on genetic algorithm is proposed. The algorithm aims at maximizing unit computing cost benefits, thereby effectively reducing schedule completion time, realizing load balance, and improving unit computing cost benefits.

Acknowledgement

This research was financially supported by Scientific Research Project of Heilongjiang Province Office of Education: A Study of the Establishment and Sharing of Digital Education Resources Based

on Cloud Computing (12531191), The Project of National Twelfth Five Years Plan for Researching Education Information Technology in 2014: A Study of the Technical Support Service System of Education Informatization in Mudanjiang (146231583) and The Specific Project of Twelfth Five Years Plan for Educational Science in Heilongjiang Province in 2014: A Study of Deep Integration of Information Technology and Teaching of Elementary Education

References

[1] LI Qiao, ZHENG Xiao. Research Survey of Cloud Computing, Computer Science, Vol.38, pp.32-37, (2011).

[2] Wang H T, Liu B. Analysis on Cloud Computing and Key Technology Problems. Telecommunications for Electric Power System, Vol.9, pp.56-62, (2011).

[3] LU, Hui, CHEN Ming. Parallel Test Task Scheduling with Constraints Based on Hybrid Particle Swarm Optimization and Taboo Search. Journal of Electronics, Vol.21, pp.615-618, (2012).

[4] Braun T D,Siegel H J,Beck N,et al.A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed Computing, Vol. 61, pp.810-837, (2001).

[5] Bhoi U, Ramanuj P N. Enhanced Max-min task scheduling algorithm in cloud computing.International Journal of Application or Innovation in Engineering and Management, Vol. 2, pp.259-264, (2013).

[6] CHEN Hai-yan. Task Scheduling in Cloud Computing Based on Swarm Intelligence Algorithm, Computer Science, Vol. 41, pp.83-86,(2014).

[7] YANG Jianyou, TANG Yan. Investigation of Overall Architecture, Applications and Business Models of Cloud Computing, Digital Communication, Vol.39, pp.3-6, (2012).

[8] Wang P. Research on Task Scheduling Strategy in Cloud Computing Environment. Computer & Modernization, Vol.9, pp.67-73,(2013).

[9] Dasgupta K, Mandal B, Dutta P, et al. A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. Procedia Technology, Vol. 10, pp.340-347, (2013).