Advances in Engineering Research (AER), volume 61
International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2017)

Research on Cloud Task Scheduling based on Multi-Objective Optimization

Xiaohong Hao [a], Yufang Han [b,*], Juan Cao [c], Yan Yan [d], Dongjiang Wang [e]

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

[a]haoxhlut@163.com, [b]cicyyufang@163.com, [c]angelhoodcj@163.com, [d]yanyan@lut.cn, [e]wangdj_xhz@163.com

*Corresponding author: Yufang Han

Keywords: Cloud computing, Task scheduling, Multi-objective optimization, Memetic algorithm.

Abstract. The efficient task scheduling in cloud environment has become the main research topic recently, the execution time, execution cost and load balancing for optimization in a cloud environment is significant. The scheduling of execution time and cost is a NP-hard multi-objective optimization problem, however, the current task scheduling under the cloud environment is generally the execution time or cost of single objective optimization with constraint conditions, incompletely meeting the complex cloud systems with load balancing. Given above motivations, in this paper, we propose a Memetic algorithm (MA, Memetic Algorithm) aiming at cloud task scheduling. Standardizing the objective function, the algorithm introduces the selection scheme based on the roulette, and Hill Climbing algorithm as local search. At last, we demonstrate the feasibility and efficiency of the proposed approach on the CloudSim simulator.

1. Introduction

Cloud computing is a computing model, in this model, users can access a lot of computing and storage resources, and don't need to care about where they are and how they are configured [1].

Recently, many studies have been designed to meet users' various constraint conditions [2-5]. However, most of cloud task scheduling algorithms generally concern a single service quality (QoS, Quality of Service) constraint, for example, a genetic algorithm of the task scheduling was proposed by Li [6], only considering the completion time. Wang [7] put forward a cloud task scheduling method based on parallel genetic algorithm, which reduces the overall execution time of scheduling tasks. Pandey proposed a particle swarm optimization algorithm, minimizing the execution cost [8]. A task scheduling was proposed based on the load balancing, but the needs of users' QoS in these papers are not fully considered.

Through in-depth study and analysis of the existing methods, it's easy to find that the traditional cloud task scheduling algorithm mainly focused on studying the task execution time or cost, not taking into account the multi-objective optimization.

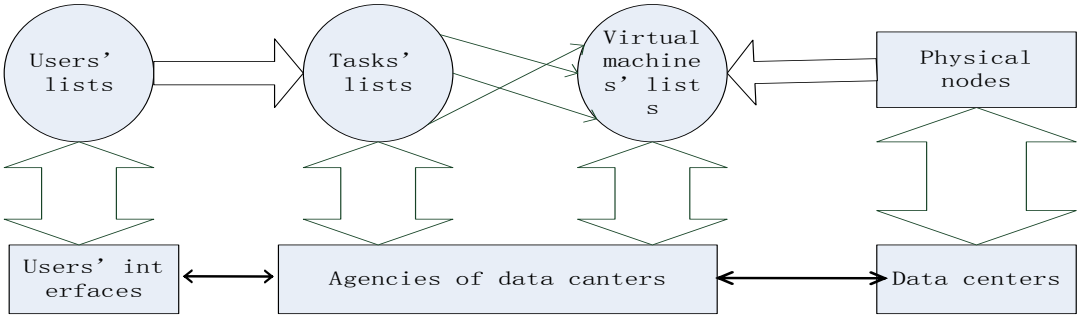

Fig. 1 Cloud task scheduling model

2. Cloud Task Scheduling Model

The cloud task scheduling model is shown below in Fig. 1.

The mathematical description of the parameters is shown below.

Tasks' lists: T= {t_1, t_2, t_3,... t_n}, The amount of data in each task is Numi .

Virtual machines' lists: VM= {m_1, m_2, m_3,... m_n}, the processing capacity in each virtual machine is MIPs, the price of each virtual machine is $Price_v$.

The execution time,cost and the load balancing are relatively equation(1), (3), and (4), with execution time in each virtual machine is equation (3).

$$totaltime = \max(timeVM1, timeVM2, ..., timeVMm) \tag{1}$$

$$timeVMi = \frac{\sum size}{Mips} \tag{2}$$

$$total\ cost = \sum_{i=1}^{m} Price_V * size \tag{3}$$

$$load = \sqrt{\sum_{i=1}^{m} (Numi - \frac{n}{m})^2} \tag{4}$$

We describe *n* as the amount of tasks and m as the number of virtual machines. In this paper, we adopts the standard deviation of virtual machines' load balancing, the smaller the value, the more balanceable the load balancing.

3. The Task scheduling based on Memetic Algorithm

Multi-objective Memetic Algorithm	
1: initialize parameters configuration, the entire population G and t←0	
2: calculate F(i) to G (t)	7: execute Hill Climbing()
3: if(t>n‖load>l_{max}), output results	8: return: G'(t)
4: else:	9: end else
5:execute the population crossover operation	10: end if
6: return: G (t)	11: perform mutation operation
12:carry out roulette wheel selection operation to G'(t) and save results to G (t)	
13:return step 3	

Hill Climbing:		
Input: G, i, i_v	6:Best_calculate(VM)	13: end if
Output: bestG	7: if(VM>v)	14: t←t+1;
1: initialize G	8: v←VM	15: if(v>bestG)
2: if(t!=Max)	9: else	16: bestG←v;
3: i←false	10: i←true	17: return bestG;
4: Rand_select(i_v)	11: end else	18: end if
5: Calculate(i_v)	12: end if	

3.1 Encoding and Decoding.

The algorithm adopts the resources-task indirect encoding. If there are 8 tasks and 4 virtual machines in the cloud environment, then the chromosome is x={1, 1, 2, 2, 3, 3, 4, 4}. It means that the first two tasks are in the first virtual machine, the 3rd and 4th tasks are in the second virtual machine..., and so on.

When decoding, Num1={1,2}; Num2={3,4}; Num3={5,6}; Num4={7,8}. It's easy to obtain the relevant task information of each virtual machine.

3.2 Design of Fitness Function.

A fitness function is the objective optimization function in this paper.

Because the execution time and cost are not in the same level, this paper adopts the approach in the literature [9] to normalize the two objective functions, and then uses the weight calculation to design the fitness function.

The execution time and cost functions are relatively equation (5) and (6).

$$f_T(i) = \frac{1/totaltime(i)}{1/totaltime(1)+1/totaltime(2)+...+1/totaltime(g)} \tag{5}$$

$$f_C(i) = \frac{1/total\cos t(i)}{1/total\cos t(1)+1/total\cos t(2)+...+1/total\cos t(g)} \tag{6}$$

Normalizing, each objective function value is [0, 1]. The fitness function is below.

$$F(i) = \omega f_T(i) + (1-\omega)\ f_C(i) \tag{7}$$

It's noted that $\omega \in [0, 1]$ and ω represents the users' dependence to time.

3.3 Genetic Operations

3.4.1 Selection Operator

Roulette method, in fact, is the proportion selection. Its main idea is that the individual selected probability is proportional to the size of the fitness functions. Due to the standardized processing to the objective function, the adaptive value in the group doesn't vary too much, as a result, this algorithm will not fall into a local optimum earlier and it's not difficult to find that the probability that the individual i inherit to next generation is as follows.

$$P_i = F(i) / \sum_{i=1}^{g} F(i) \tag{8}$$

3.4.2 Crossover and Mutation Operator

In this paper, crossover operator uses the single point crossover restructuring, while mutation operator employs the multipoint mutation. For the gene location that the individual needs to crossover and mutate is determined randomly. Only through the selection can the next generation come into being.

4. Simulation Experiment

This section mainly verify the efficiency and which place that the algorithm is suitable to by applying the Memetic algorithm to different cloud simulation scenes.

In CloudSim simulation platform, two cloud scenarios are set up. To make results more intuitive, some parameters of virtual machines are omitted, such as memory, so on, only selecting some parameters associated with optimization goal.

Scenario 1: Set population size is 20, the task number is 20, the virtual machine number 10, the crossover probability is 0.6, mutation probability is 0.1, the threshold of load balancing is 5 and evolution algebra is 2000. Tasks and attributes of virtual machine are shown in Table 1, while experimental results are shown in Fig. 2.

Table 1. The tasks virtual machines of scenario 1

T_{number}	D_{amount}	T_{number}	D_{amount}	VM_{number}	$VM_{capacity}$	Cost
1	18793	11	34577	1	345	4
2	50465	12	45633	2	254	3
3	30123	13	45678	3	143	2
4	44132	14	75688	4	398	4
5	14564	15	44655	5	143	2
6	20046	16	23941	6	543	6
7	35467	17	38791	7	345	4
8	30987	18	28971	8	231	3
9	25346	19	34516	9	341	4
10	12365	20	32416	10	253	3

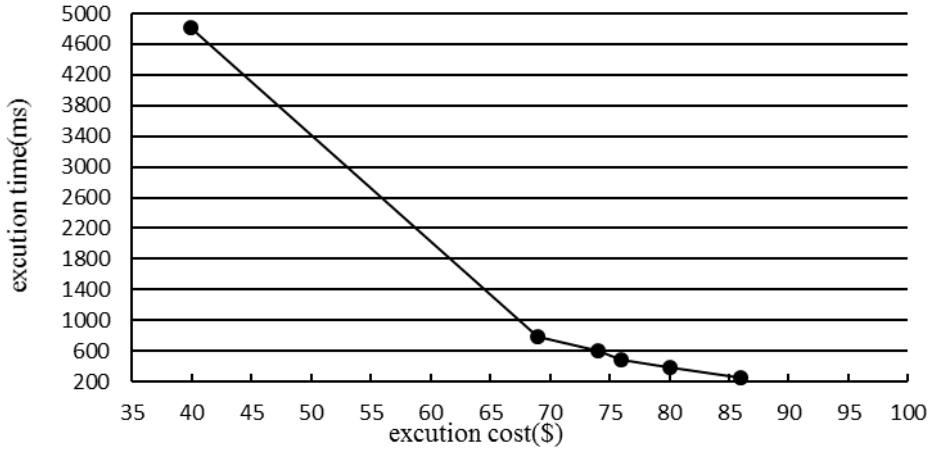

Fig. 2 The experimental result of scenario 1

Table 2. The tasks of scenario 2

T_{number}	D_{amount}	T_{number}	D_{amount}	T_{number}	D_{amount}	T_{number}	D_{amount}
1	18793	11	34577	21	187931	31	345778
2	50465	12	45633	22	504653	32	456336
3	30123	13	45678	23	301235	33	456783
4	44132	14	75688	24	441326	34	756881
5	14564	15	44655	25	145648	35	446556
6	20046	16	23941	26	200469	36	239419
7	35467	17	38791	27	354675	37	387916
8	30987	18	28971	28	309878	38	289718
9	25346	19	34516	29	253469	39	345163
10	12365	20	32416	30	123652	40	324162

Table 3. The virtual machines of scenario 2

VMnumber	VMcapacity	cost	VMnumber	VMcapacity	Cost
1	345	4	11	345	4
2	254	3	12	254	3
3	143	2	13	143	2
4	398	4	14	398	4
5	143	2	15	89	1
6	543	6	16	70	1
7	345	4	17	345	4
8	231	3	18	231	3
9	341	4	19	341	4
10	253	3	20	253	3

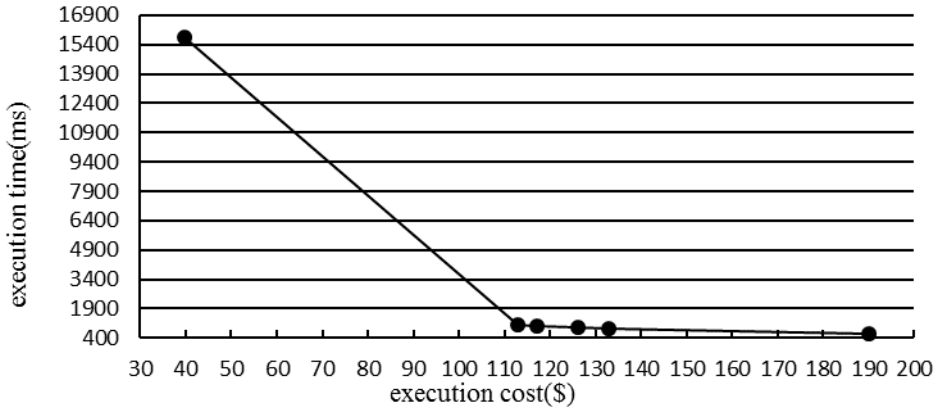

Fig. 3 The experimental result of scenario 2

The load balancing value is significantly higher in scene 2 than that is in scene 2, which means that the more the number of tasks in cloud computing, the more difficult to achieve load balance. Therefore, a certain tolerance requirement should be given under such circumstances.

In the scene 1, the curve is steep, and with the increasing of weight, the execution time and cost without increasing or decreasing obviously, which represents that two goals' optimization results are not very obvious; while the curve is smoother in the scene 2, which fully explains that optimization results in the scene 2 are relatively obvious.

As is shown above, the algorithm is more suitable for task scheduling based on multi-objective optimization under the complex cloud system. According to users' reliance on different goals, choose the appropriate weight to meet users' demands.

5. Summary

In this paper, we propose us a Memetic algorithm aiming at complex cloud task scheduling and verify the feasibility and efficiency of the proposed approach by simulating experiments. However, there are still some difference between simulation platform and the reality. With further improvements we hope to come up with more flexible scheduling method in the future to compare with the proposed approach in the paper and apply them to the real cloud environment.

References

[1] M. Armbrust, A. fox, R. Griffith, A view of cloud computing, Communications of the ACM, vol.53, no.4, pp.50-58, 2010.

[2] W. Chen. A set-based discrete PSO for cloud workflow scheduling with user-defined QoS constraints, Systems, Man, and Cybernetics (SMC), IEEE International Conference on. IEEE, pp.773-778, 2012.

[3] L. Singh , S. Singh, A genetic algorithm for scheduling workflow applications in unreliable cloud environment, Recent Trends in Computer Networks and Distributed Systems Security. Springer Berlin Heidelberg, pp.139-150, 2014.

[4] R. Tolosana, J. BanAres and C. Pham, Enforcing qos in scientific workflow systems enacted over cloud infrastructures, Journal of Computer and System Sciences, vol.78, no.5, pp.1300-1315, 2012.

[5] A. Klein, F. Ishikawa, S. Honiden, Efficient qos-aware service composition with a probabilistic service selection policy, Service-Oriented Computing. Springer Berlin Heidelberg, pp.182-196, 2010.

[6] L. jin, J. Peng, Task scheduling algorithm based on improved genetic algorithm in cloud computing environment, Comput.Appl, vol.31, no.1, pp.184–186, 2011.

[7] Z. Zhong, W. Rui, Z. Hai, An approach for cloud resource scheduling based on parallel genetic algorithm, Proc of the 3rd International Conference on Computer Research and Development, Washington DC: IEEE Computer Society, pp.444-447, 2013
[8] S. Pandey, L. Wu, S. M. Guru, A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments, Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. IEEE, pp.400-407, 2010.
[9] H. Chao, Y. Xing, H. Demin, Multi-objective genetic algorithm's application in the cloud computing task scheduling, Information technology, no.5, pp.130-134, 2014.