Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 13, no. 7, November 2017, pp. 1070-1076
DOI: 10.23940/ijpe.17.07.p9.10701076

Cloud Task Scheduling Algorithm based on Improved Genetic Algorithm

Hu Yao, Xueliang Fu*, Honghui Li, Gaifang Dong, Jianrong Li

College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot, 010018, China

Abstract

Cloud computing is a new type of business computing model. It is connected through the network and can obtain various applications, data and IT services. The core of cloud computing is task scheduling, and the application of genetic algorithm (GA) in cloud computing task scheduling is also a hot topic in recent years. In this paper, the "three-stage selection method" and the genetic strategy of "total-division-total" are put forward to improve genetic algorithm. Using simulation experiments in cloud computing simulation software named Cloudsim, the experimental results show that comparing with the simple genetic algorithm (SGA), the improved genetic algorithm (IGA) is better than the simple genetic algorithm on completion time, and it is an effective task scheduling algorithm in cloud computing environment.

Keywords: cloud computing; genetic algorithm; selection operation; crossover operation; task scheduling

(Submitted on July 25, 2017; Revised on August 30, 2017; Accepted on September 15, 2017)

(This paper was presented at the Third International Symposium on System and Software Reliability.)

1. Introduction

Cloud computing [1,2] is based on the mode that increases, uses and delivers related services of the Internet. It usually involves the provision of dynamic, scalable, and often virtualized resources over the Internet, and it is the product of the integration of traditional computer and network technologies, such as Distributed Computing, Parallel Computing, and Virtualization.

The amount of task that cloud computing needs to deal with and how to make use of the existing resources to allocate the cloud task reasonably is enormous. So to improve the efficiency of resource utilization and task completion efficiency and save time, it is the key point and difficulty in the research of cloud computing.

In this paper, by making a study of how to efficiently schedule cloud tasks and shorten the total completion time of cloud tasks, a series of improvements have been made on the basis of simple genetic algorithm (SGA), and the improved genetic algorithm (IGA) has been researched out. The results of simulation experiment show the effectiveness of the improved algorithm.

2. Description of task scheduling

For the cloud task scheduling [7,9], this paper aims at minimizing the total scheduling time. In order to solve the scheduling problem, the following assumptions are given:

- The input of the task is the minimum child task, which means that large tasks have been split into small tasks, and the length of the minimal task is random.
- All cloud computing resources are mapped to virtual machines, and the performance of all virtual machines is random.

* Corresponding author.
E-mail address:fuxl@imau.edu.cn .

- The number of tasks is much larger than the number of virtual machines.

Cloud task scheduling can be described as: N cloud tasks of different length assign to the M virtual machines of different performance, how to make reasonable allocation, so that the total completion time of all tasks is the shortest.

3. Task scheduling algorithm based on improved genetic algorithm

3.1. Genetic Algorithm

Genetic algorithm [11] was first proposed by Professor Holland, and it is a random search algorithm that draws on natural selection and natural genetic mechanisms in the biological world. The algorithm searches the global optimal solution by simulating biological evolution process, and is a collective, intelligent and heuristic algorithm. Simple genetic algorithm has some problems such as premature convergence and slower search speed, so the improvement of genetic algorithm has always been the focus of research [5,6,10].

There are three main operations of simple genetic algorithm for population：Selection, Crossover, and Mutation.

- Selection operation: in each generation, individuals who adapt to the environment (meet certain requirements) are selected by the fitness function, and the next generation is produced through these individuals.
- Crossover operation: two different individuals in a population exchange genes in the same position to produce new individuals.
- Mutation operation: the genes of certain individuals in the population may change, which may make individuals more adaptable and, of course, weaker.

Through these three operations, the population evolves in a more adaptive direction, thus finding the solution that meets our requirements.

Because of its parallelism and global solution space search, genetic algorithm has been introduced into the resource scheduling of large scale cluster systems. In particular, many researchers have studied the improved genetic algorithm and applied it to the task scheduling problem of cloud computing. In [8], a dual fitness genetic algorithm (DFGA) is proposed, which takes into account the total task completion time and the average task completion time, that improves the global search ability of the algorithm. In [4], the local population is optimized by simulated annealing algorithm in the mutation operation, and the quality of the population is improved. In [3], in order to improve the quality of the initial population，Min-Min and Max-Min algorithm are introduced to initialize the population.

3.2. Individual genes coding and decoding

All the operations of the genetic algorithm are based on individual genes, and all the results are obtained through individual genes. During task scheduling, cloud tasks are assigned to each virtual machine, and the following assumptions can be made [3]:

- The individual genes length is equal to the number of cloud tasks.
- The individual genes types are equal to the serial number of virtual machines.

An example is shown in Figure 1.

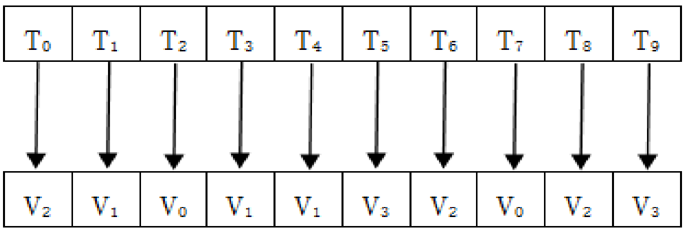

Figure 1. Relationship between virtual machine and cloud task assignment

The top line of the instance represents the cloud tasks, and the following line represents the individual genes, the number of cloud tasks is 10, and the number of virtual machines is 4，the corresponding relation : task 0 is assigned to virtual machine 2, task 1 is assigned to virtual machine 1, task 2 is assigned to virtual machine 0,..., task 9 is assigned to virtual machine 3. The individual genes can be decoded as:

$$V_0: \{T_2, T_7\}$$
$$V_1: \{T_1, T_3, T_4\}$$
$$V_2: \{T_0, T_6, T_8\}$$
$$V_3: \{T_5, T_9\}$$

After decoding, the cloud tasks are assigned to each virtual machine. Because the execution of tasks between the virtual machines is parallel, and an individual represents a scheduling scheme, the following design can be done:

- Basing on the length of completion time of all cloud computing tasks, the shorter the time, the better the individual (scheduling scheme) is.
- Single task completion time = the length of cloud task / VM performance.
- The time of all tasks completed = the time consumed by the longest running virtual machine.

3.3. Selection of fitness function

In genetic algorithm, the selection of fitness function is very important, and it is directly related to the convergence of the algorithm and the search of the optimal solution.

In this paper, combining with task scheduling, the completion time of all cloud tasks is used as fitness function in the improved genetic scheduling algorithm. That is, the lower the fitness of the individual in the contemporary population is, the shorter the completion time of all tasks and the less likely it is to be eliminated.

$$f = max(T_{Vi}) \quad i \in \{0,1,2,\ldots,M-1\} \tag{1}$$

Of which, T_{Vi} represents the time that the virtual machine *i* execute all the cloud tasks assigned to it, and *M* represents the number of virtual machines.

3.4. Selection operation

The selection operation is to select the good individuals and eliminate the bad individuals. The average fitness f_a and the random fitness f_r are introduced into the selection operation.

$$sum = \sum f(i) \quad i \in \{1,2,3,\ldots,n\} \tag{2}$$

$$f_a = sum / n \tag{3}$$

$$f_r = f_a \cdot Math.random() \tag{4}$$

In these equations, *sum* represents the total finishing time, including all individuals in this population, *f(i)* is the fitness of individual *i*, and *n* is population size, by formula (3) and (4) we can see $f_a > f_r$.

The three-stage selection method is adopted in this paper; each selection retains the best individual of the contemporary population and makes different grades for other individuals. If the individual fitness is $f < f_r$, the individual is defined as a excellent individual and retained; Individuals with fitness $f_r < f < f_a$ are defined as sub excellent individuals and replaced by variants of the optimal individual; Individuals with fitness $f > f_a$ are poor individuals, these individuals will be eliminated and replaced by new individuals. This not only ensures the stability and diversity of the population, but also helps to find the global optimal solution.

3.5. Crossover operation

Crossover operation is also the main means of searching global optimal solution by genetic algorithm; it simulates sexual reproduction in nature and passes good genes to the next generation. Some individuals may mutate during this process, and have a chance to get better individuals. In simple genetic algorithms, crossover operation does not take into account the similarity of two individual crossover regions; if the gene similarity of crossover region is too large, crossover operation will become meaningless. In this paper, crossover region similarity and asexual reproduction are introduced, and the specific operations are as follows:

Suppose the individual A crosses with the individual B, and the crossover region length is L, in this region, the number of same genes of two individuals is K, and the crossover region similarity is:

$$S = K/L \tag{5}$$

If the crossover region length is 9, the genes L_A=022130111, L_B=012112031, then K=4, crossover region similarity S=0.44, the crossover region similarity threshold μ=0.5 is specified in this paper.

Two individuals perform crossover operation, and when the crossover region similarity is greater than the threshold, we decide that the crossover operation is meaningless and the two individuals will reproduce in asexual form: individual recombine genes randomly in crossover region then add it to the next iteration.

4. Algorithm simulation and result analysis

In this paper, cloudsim [12,13] (a cloud simulation software) is used to simulate the task scheduling algorithm of simple genetic algorithm (SGA) and improved genetic algorithm (IGA), the sequential scheduling (similar to the FIFO scheduling strategy in Hadoop) is added to make a comparison.

The experimental parameters are as follows: the maximum number of iterations is 300, the population size is 1000, the number of virtual machines is 10, the mutation probability is 0.05, and the crossover region similarity threshold is 0.5.

Through a large number of simulation experiments, we get the following results：

In the results of the experiments, the vertical axis represents the time of completion of all cloud tasks, and the time scale is 1:1000. The time to complete a cloud task is equal to the length of the cloud task divided by the performance of the virtual machine that assigned to the cloud task. The time from the first cloud task to the last cloud task is completion time, but not the sum of time with each cloud task. The horizontal axis represents the number of submitted cloud tasks that span from several hundred to several thousand.

4.1. Sequential scheduling and SGA scheduling

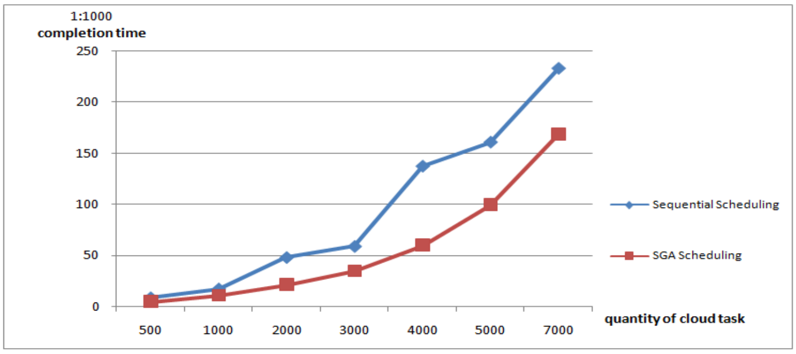

Figure 2. Comparison between sequential scheduling and SGA scheduling

Figure 2 shows that no matter how many tasks there are, SGA scheduling is more advantageous than sequential scheduling. Genetic algorithm is an intelligent swarm search algorithm, and each individual represents a solution. As the iterations go deeper, a lot of better individuals will emerge, so the resulting solution is certainly better than sequential scheduling.

4.2. SGA scheduling and IGA scheduling

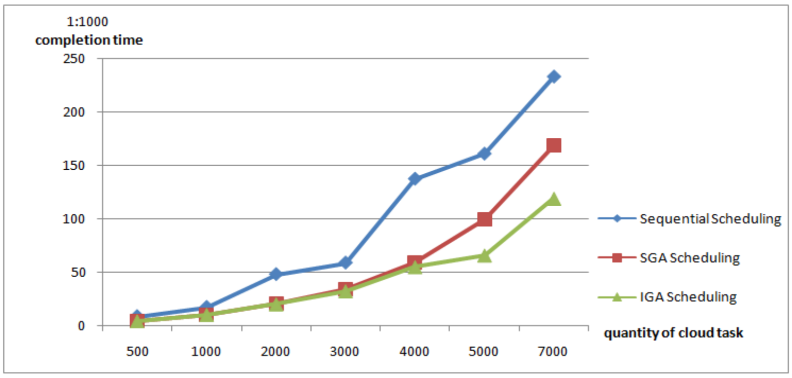

Figure 3. Sequential scheduling, SGA scheduling and IGA scheduling

As shown in Figure 3, when the number of cloud tasks is small，IGA scheduling has advantages over SGA scheduling, but not much. When the number of cloud tasks gradually increased, the advantage of IGA scheduling is obvious. Thus it can be seen that the IGA scheduling algorithm is effective, especially in practical application, the number of tasks needed to run in a cloud computing center is much more than that. Therefore, the IGA scheduling algorithm is very practical.

5. Enhanced IGA scheduling algorithm

5.1. The thinking of enhanced IGA

On the basis of IGA scheduling algorithm, the genetic strategy of "total-division-total" is put forward, and the concept of "species" is introduced in this paper. Upgrading the original population to species, and the scale is the same as the original population. The evolutionary process follows:

- First, the species evolves several generations
- The species that have evolved several generations will be divided into several populations; each population evolves independently in different directions
- In each population, individuals whose fitness is less than average fitness are preserved and the others will be eliminated
- Assembling all retained individuals into a new species in order to ensure that the species size is the same as before; the insufficient part is replaced by the new individuals, which ensures the diversity of individuals in the species

The improvement is based on IGA scheduling algorithm, selection operation and crossover operation are the same as those of IGA, so the improvement is called the Enhanced IGA (EIGA).

5.2. Experimental simulation and result analysis

The experimental parameters are as follows: the number of iterations of initial species is 100, The number of iterations in intermediate population is 100, the number of iterations of final species is 100, the species size is 1000, the number of split population is 4, the number of virtual machines is 10, the mutation probability is 0.05, and the crossover region similarity threshold is 0.5.

The simulation results are shown in Table 1 below:

Table 1. IGA scheduling and EIGA scheduling（4 split population）

Quantity of Cloud Task and Completion Time	500	1000	2000	3000	4000	5000	7000
IGA scheduling	4410.3	10396.8	20232.7	32549.7	55132.3	66096.7	119384.1
EIGA scheduling	4406.4	10375.7	20224.3	32511.5	55024	65878.4	115688.7

Since the improvement range of EIGA relative to IGA is small, such a hypothesis can be made: as the number of split population is less, the result is not much improved compared with non segmentation. In the next simulation experiment, other conditions remain unchanged, and the number of split population is increased to 10.

Table 2. IGA scheduling and EIGA scheduling（10 split population）

Quantity of Cloud Task and Completion Time	500	1000	2000	3000	4000	5000	7000
IGA scheduling	5593.2	10937.5	24042.8	29867.8	49756.1	76587.2	126494.9
EIGA scheduling	5543	10317.5	23368.6	28817.9	49013.6	73245.5	119190.2

Because the cloud task length and the virtual machine performance are selected randomly in a certain range, the obtained data has a certain fluctuation.

As shown in Table 2，when the number of split population is 10, the EIGA scheduling is obviously improved than the IGA scheduling. It can be seen that it's better to divide the species by 10 than 4. Such a conclusion can be reached: as the number of split population increases, the evolutionary direction is more diverse; it is similar to the diversity of individuals in the population.

Besides, we can see that with the increase in the number of cloud tasks, EIGA scheduling saves more and more time than IGA scheduling. It can be speculated that when the number of cloud tasks is very large, the savings of time is considerable. Therefore, the improvement of EIGA scheduling has visible practical significance.

6. Conclusions

The "three-stage selection method" and the genetic strategy of "total-division-total" are put forward in this paper. Two improvements have been made to the SGA task scheduling algorithm. Simulation results show that comparing with IGA scheduling algorithm, SGA scheduling algorithm has obvious improvement, and comparing with IGA scheduling algorithm, EIGA scheduling algorithm also improves to some extent. On a large scale task scheduling problem, the two improved algorithms can effectively improve task scheduling efficiency. The next step is to integrate other algorithms to further improve task scheduling.

Acknowledgements

This research was financially supported by Chinese Natural Science Foundations (61363016, 61063004), Key Project of Inner Mongolia Advanced Science Research (NJZZ14100), Inner Mongolia Colleges and Universities Education Department Science Research (NJZC059), Natural Science Foundation of Inner Mongolia Autonomous Region of China (NO.2015MS0605, NO.2015MS0626 NO.2015MS0627, and NO.2017MS0605) and Ministry of Education Scientific research foundation for Study abroad personel[2014] 1685, Inner Mongolia Autonomous Region Science and Technology Project(Development of On-line Monitoring and Data Transmission System for GSM Network with Penetrating Precipitation).

References

1. H.W. Fan, “Value Creation and Its Mechanism of Cloud Computing,” *Bullentin of Chinese Academy of Sciences*, 2(2015)
2. Hayes, Brian, “Cloud Computing,” *Communications of the Acm*, 51.7(2008):9-11
3. Y.H. Hu and X.L. Tang, “A Task Scheduling Algorithm Based on Improved Genetic Algorithm in Cloud Computing Environment,” *Computer Technology and Development* , 26.10(2016):137-141
4. J. Jing, “Cloud Computing Task Scheduling Based on Genetic Algorithm,” *Telecom World*, 1(2016):33-34
5. Kumar, Pardeep, and A. Verma, “Scheduling Using Improved Genetic Algorithm in Cloud Computing for Independent Tasks,” *International Conference on Advances in Computing*, Communications and Informatics, 2012:137-142
6. Kaur, Shaminder, and A. Verma, “An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment,” *International Journal of Information Technology & Computer Science*, 4.10(2012):159-190
7. Lakshmi, R. Durga, and N. Srinivasu, “A Dynamic Approach to Task Scheduling in Cloud Computing Using Genetic Algorithm,” *Journal of Theoretical & Applied Information Technology* (2016)
8. J.F. Li and J. Peng, “Task Scheduling Algorithm Based on Improved Genetic Algorithm in Cloud Computing Environment,” *Journal of Computer Applications*, 31.1(2011):184-186
9. J.T. Ma, Y.E. Chen, G.J. Hu and L.L. Yan, “Research of Task Scheduling Based on Genetic Algorithm Technology in Cloud Computing Environment,” *Software Guide*, 15.1(2016):51-53

10. J. Ma, W. Li, T. Fu, L. Yan and G. Hu, "A Novel Dynamic Task Scheduling Algorithm Based on Improved Genetic Algorithm in Cloud Computing," Wireless Communications, Networking and Applications. *Springer India*, 2016:184-186
11. Y.J. Ma and W.X. Yun, "Research Progress of Genetic Algorithm," *Application Research of Computers*, 29.4(2012):1201-1206
12. X.J. Wang, "Study and Application of A Toolkit for Cloud Computing Simulation—CloudSim," *Microcomputer Applications* , 29.8(2013):59-61
13. Y.N. Wang and W.H. Wu, "The Analysis of Simulation Process of Cloudsim 3.0," *Computer Engineering & Software*, 6(2015):109-113

Hu Yao is a graduate student, graduated from Inner Mongolia Agricultural University in 2015 in China and received his bachelor's degree in network engineering. His research interest is task scheduling algorithm for cloud computing.

Xueliang Fu is a professor in algorithms of graph theory at Inner Mongolia Agricultural University in China. He received the MS and PhD in software engineering from DALIAN University of Technology in 2005 and 2008 in China. He received his BS in computer science in 1992. His research interests include Domination Number of Graphs，Intelligent Computation and Data Mining. He has published about 50 papers in international journals, conference proceedings. Now Xueliang Fu is the Vice Dean of the College of computer and information engineering of Inner Mongolia Agricultural University.

Honghui Li received the BS, MS and PhD degrees all in computer science from Electronic technology University, China, Inner Mongolia University, China and Concordia University, Canada in 1992, 2002, and 2012 respectively. She is currently a professor in Computer science & technology with School of Computer and Information Engineering, Inner Mongolia Agricultural University, China. Her research interests include network optimization and Network planning algorithms. She has published about 25 papers in international journals and conference proceedings.

Gaifang Dong received the BS and MS degrees both in computer science from Inner Mongolia Normal University, China and GuiZhou University, China, in 2002 and 2005 respectively. She is currently a Ph.D. student and an Associate Professor at the College of Computer and Information Engineering in the Inner Mongolia Agricultural University. Her research interests include computational intelligence, bioinformatics computing and parallel computing. She has published about 15 papers in international journals and conference proceedings.

Jianrong Li received the BS degree in electric automation in 1999 and the MS degree in control theory and control engineering in 2004 that both from Inner Mongolia Polytechnic University. She is currently a Ph.D. student and an Associate Professor at the College of Computer and Information Engineering in the Inner Mongolia Agricultural University. Her research interest is task scheduling algorithm water resources management for cloud computing.