Evaluation of cloud computing resource scheduling based on improved optimization algorithm

Huafeng Yu¹

Received: 26 February 2020 / Accepted: 30 May 2020 / Published online: 22 September 2020
© The Author(s) 2020

Abstract

Cloud computing, as a new computing mode in recent years, has been pursued by many users who have computational requirements, and the service quality of cloud computing depends largely on the efficiency of resource scheduling. In this study, an improved particle swarm optimization (IPSO) algorithm was proposed to improve the efficiency of resource scheduling, and simulation experiments were carried out on the IPSO algorithm and the traditional particle swarm optimization using CloudSim simulation platform. The phenomenon of premature appeared with the increase of the number of iterations, and the globally optimal solution was not found. The IPSO algorithm was more efficient in exploring the globally optimal solution, and the phenomenon of premature did not appear. As the number of tasks increased, the operation time of both algorithms increased, but the IPSO algorithm increased more slowly. The IPSO algorithm had more advantages when there were a large amount of tasks. Virtual machines in the two algorithms had different loads, and the load of the virtual machine in the IPSO algorithm was more balanced.

Keywords Cloud computing · Improved particle swarm algorithm · CloudSim · Resource scheduling

Introduction

As a new technology, cloud computing can accomplish the computing resource service using an easy extension method according to the user's need for computing [1]. Users can access various services provided by cloud computing through the Internet. Unlike traditional information technology patterns, cloud computing services do not require users to install and operate applications on their own, thus reducing the cost required by user services. Besides, the resources and operations required by users are on the cloud terminal of the Internet; as a result, the operation load on the user terminal is greatly reduced, which reduces the requirement on the configuration of computer and saves the cost of users. With the development of computer information technology, users' demands for cloud computing platform are also increasing. How to schedule resources based on demands has become a major problem. Bölöni et al. [2] made cloud

computing analysis of real estate investment opportunities based on scheduling information value and found that the operation can be greatly simplified. Chen et al. [3] proposed an improved ant colony system (IACS) method and carried out extensive experiments based on workflows with different sizes and different cloud resources. The experimental results showed that IACS could find better solutions and was more cost-effective than basic particle swarm algorithm and dynamic object genetic algorithm under different scheduling scales and time limits. Abdullahi et al. [4] proposed a symbiotic organisms search (SOS) algorithm based on simulated annealing (SA) to improve the convergence speed and quality of SOS solutions. The results of CloudSim simulation showed that SASOS was superior to SOS in convergence speed, response time, imbalance and MAK. Aiming at the cloud computing programming model, Zhang et al. [5] put forward an improved cross-generation heterogeneous recombination cataclysmic mutation (CHC) algorithm which inherited the advantages of standard genetic algorithm (SGA) and CHC algorithm. The improved CHC algorithm took into account both the resource allocation and the completion time of tasks, which improved the convergence speed of the algorithm. The experimental results demonstrated that the improved CHC algorithm had better

✉ Huafeng Yu
107885339@qq.com

¹ School of Digital Information Technology, Zhejiang Technical Institute of Economics, Hangzhou 310018, Zhejiang, China

efficiency and convergence, and the average time of task scheduling and completion time were short. In this study, an improved particle swarm optimization (IPSO) algorithm was put forward to enhance the efficiency of resource scheduling, and moreover, simulation experiments were carried out on the IPSO algorithm and the traditional PSO algorithm on CloudSim simulation platform.

Scheduling of cloud computing resources

The resource scheduling model in cloud computing is divided into three layers: service request, virtual resource pool, and physical resource pool [6]. The cloud computing resource scheduling is generally divided into two steps. First, tasks in service requests are assigned to the virtual machine in the virtual resource pool, and then the allocated virtual machines are deployed on the physical machine of the physical pool, as shown in Fig. 1.

Its mathematical model [7] is as follows. Tasks in service requests are divided into n subtasks which are independent from each other, and the set of the subsets is $M = \{m_1, m_2, m_3 \dots m_n\}$. The i -th task was denoted as m_i . One subtask can only run on one virtual node. Suppose there are b virtual nodes ($b < n$), then the set of virtual nodes is $V = \{v_1, v_2, v_3 \dots v_b\}$, where v_j stands for the j -th virtual node. Then the allocation situation can be expressed as matrix A :

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{b1} & a_{b2} & \dots & a_{bn} \end{pmatrix}. \quad (1)$$

In matrix A , a_{ji} stands for the relationship between subtask m_i and virtual node v_j . When the i -th task runs on the j -th node, a_{ji} is 1; otherwise it is 0. For convenience, suppose

Fig. 1 The scheduling of cloud computing resources

one node only runs one task once, and one task only runs on one node. Then, the time consumed in completing task M is:

$$\text{time} = \max \left(\sum_{i=1}^n a_{ji} t_{ji} \right) (1 \leq i \leq n, 1 \leq j \leq b).. \quad (2)$$

An excellent resource scheduling algorithm should complete a task with the least time, i.e.,

$$\text{Cost} = \min(\text{time}). \quad (3)$$

Equations (1), (2) and (3) are the mathematical models of cloud computing resource scheduling.

IPSO algorithm

The traditional PSO algorithm

The traditional PSO algorithm is derived from behaviors of foraging and migration of birds; hence, it is also called bird foraging algorithm [8]. Its principle is to obtain globally optimal solution through tracing the current optimal solution. The traditional swarm particle algorithm is one of the evolutionary algorithms [9]. Similar to other evolutionary algorithms, it solves the set of solutions in a space based on population and then performs iterative operation through randomly initializing population and setting termination condition to obtain optimal solution. In the process of operation, the exploration direction of optimal solution is determined by fitness value, and whether the solution is good or not is determined by the size of fitness; globally optimal solution is obtained by comparing the previously optimal solution with the currently searched optimal solution. Compared to other evolutionary algorithms, PSO algorithm is simpler in coding and involves less parameters which need to be maintained. Only one core formula is needed in PSO algorithm. In PSO algorithm, individual updating is affected by historical particles rather than random. Suppose the number of particles in PSO algorithm as n , the vector position of the i -th particle as P_i in a M -dimensional space, i.e., $\vec{P}_i = (p_{i1}, p_{i2}, p_{i3} \dots p_{iM})$, and the speed of particles as S_i , i.e., $\vec{V}_i = (v_{i1}, v_{i2}, v_{i3} \dots v_{iM})$. The optimal position of the i -th particle was set as $pbest_i$, and the optimal position of particle swarm as $gbest_i$. Then, the formula of velocity change of the i -th particle [10] is:

$$V_{i+1} = \mu \cdot V_i + a_1 \cdot x_1 \cdot (pbest_i - P_i) + a_2 \cdot x_2 \cdot (gbest_i - P_i), \quad (4)$$

where μ is the numerical value of element, a_1 and a_2 are learning factors, x_1 and x_2 are random numbers which evenly distributes between 0 and 1, $a_1 \cdot x_1 \cdot (pbest_i - P_i)$ is a cognitive term, and $a_2 \cdot x_2 \cdot (gbest_i - P_i)$ is a social term.

The expression of the position change of the particle [11] is

$$P_{i+1} = P_i + V_i. \quad (5)$$

The optimal solution with the largest fitness was found out by repeatedly updating Eqs. (4) and (5) and analyzing the fitness of the particle.

Improved PSO algorithm

Currently, there are many ways to improve the traditional PSO, but the ultimate goal is to make up for the two shortcomings of the traditional PSO algorithm. One is that when the test object is a complex function, the algorithm is very likely to fall into the local extremum with the increase of the number of iterations, and it is difficult to get the real optimal under such a condition. The other is the selection of algorithm parameters; inertia factor and learning factor have the largest influence on the change of algorithm capability. In the traditional PSO algorithm, cognitive term coefficient a_1 and social term coefficient a_2 are kept constant; the two coefficients are learning factors. Therefore, the algorithm is greatly influenced by individual cognition when the number of iterations is small, and the social influence is large when the number of iterations is large, and it can not reflect the above changes, which will make the algorithm get local optimal solution and induce premature phenomenon. To solve the above problems, the traditional PSO algorithm was improved, and the expression after improvement [12] is:

$$S_{i+1} = \mu \cdot V_i + b \cdot \frac{1}{n} \cdot x_1 \cdot (pbest_i - P_i) + c \cdot n^2 \cdot x_2 \cdot (gbest_i - P_i). \quad (6)$$

It was found from the comparison between Eqs. (4) and (6) that the improvement was changing cognitive term coefficient a_1 into $b \cdot \frac{1}{n}$ to make the proportion of individual cognition decrease with the increase of number of iteration and changing social term coefficient a_2 into $c \cdot n^2$ to make the proportion of mass society term increase with the increase of number of iteration. Then, b was assigned with a larger value, c with a smaller value to satisfy the fact that individual cognition dominated in the initial stage and the proportion of the social term gradually became the dominant factor with the increase of number of iteration. The improved algorithm was more in accordance with the objective law of solution of optimal solution and more practical and could avoid local optimal solution.

The flow of the improved particle swarm algorithm [13] is shown in Fig. 2.

Fig. 2 The flow of the IPSO algorithm

Simulation experiment

Experimental environment

The performance of the IPSO algorithm was tested by taking CloudSim3.0 [14] as the simulation platform of cloud computing resource scheduling, and it was compared with the

traditional PSO to verify that the improved PSO algorithm had improved cloud computing resource scheduling ability. In this study, experiments were carried out on Windows10 operating system with 16 G memory and 1000 G hard disk storage. CloudSim3.0, JDK1.7 and development tool MyEclipse were also used.

Experimental parameters

Relevant parameters of virtual machine resources are shown in Table 1. The executive force of virtual machines in the experiment was determined by the number of processor.

Relevant parameters of some tasks are shown in Table 2. The size of the task was randomly generated, and the unit is MI. In general, time is the main factor affecting the quality of service for the users; so, the weight of the completion time was set as the highest, 0.4, and the other factors were set as 0.2 in the experiment.

The values of relevant parameters of the PSO were as follows. The total particle swarm scale N was 60, the maximum and minimum values of inertia factor, $\mu_{\max}$ and $\mu_{\min}$, were 0.8 and 0.3, respectively; the maximum and minimum values

of learning factor, $a_{\max}$ and $a_{\min}$, were 2 and 1, respectively, and the maximum speed of particles, $\bar{V}_{\max}$, was 1.

Experimental results

Comparison of convergence speed

According to the parameters given above, the convergence performance of the traditional PSO algorithm and IPSO algorithm was tested, and the results are shown in Fig. 3. The parameters setting of the two algorithms was basically the same in the experiment; therefore, Fig. 3 could reflect the ability of the two algorithms in searching task solution.

It could be noted from Fig. 3 that both algorithms explored the optimal solution when the number of iterations was between 0 and 100, and the exploration speed of the IPSO algorithm was faster. It was because that the parameters of the IPSO algorithm were optimized, i.e., a larger inertia factor was set at the beginning of the operation, and the initial value of the IPSO algorithm was generated by the chaos process rather than randomly, which avoided the loss of solution direction. The convergence of the PSO algorithm fluctuated when the number of

Table 1 The setting of parameters of visual machine resources

No. of virtual machine	Number of processor/n	Virtual memory/MB	Bandwidth/Kbps
0	5	2048	400
1	5	512	700
2	3	1024	700
3	3	512	800
4	2	1024	900
5	3	2048	600
6	5	2048	600
7	2	1024	500
8	5	512	900
9	3	1024	1000

Fig. 3 Comparison of the convergence speed between the traditional PSO algorithm and IPSO algorithm

Table 2 The values of relevant parameters of the particle swarm algorithm

No. of task	0	1	2	3	4	5	6
Category of task	Compute mode	Storage mode	Transmission mode	Storage mode	Compute mode	Transmission mode	Compute mode
Size of task/MI	4000	2000	4000	5000	7000	3000	1000
Expected completion time/ms	450	200	250	350	500	250	300
Expected memory/MB	1024	2048	1024	1024	2048	1024	1024
Expected bandwidth/Kbps	450	550	550	750	350	450	650
Energy consumption/W	{40,55}	{45,55}	{35,50}	{30,40}	{25,45}	{45,55}	{35,55}

iterations was between 100 and 140. It could be found from Fig. 3 that there was a stationary phase in the operation of the two algorithms; the PSO algorithm entered a stationary phase when the number of iterations was 300, while the IPSO algorithm entered a stationary phase when the number of iterations was 240. In general, the stationary phase means that the optimal solution has been found and the calculation can be stopped. However, it could be seen from Fig. 3 that the IPSO algorithm began to explore the direction of the optimal solution when the number of iteration was larger than 360, which was because that the IPSO algorithm researched the solution space in the stationary phase. With the increase of the number of iterations, the PSO algorithm had not found the optimal solution in the exploration of the global solution, was less efficient than the IPSO algorithm, and entered the stationary phase earlier.

Simulation time

There were 10 virtual machines in this experiment, and the speed of the PSO and IPSO algorithms was tested with different numbers of tasks. The result is shown in Fig. 4. It could be noted from Fig. 4 that the increase of tasks resulted in longer computation time in both algorithms. When the number of tasks was 20, the time difference of the two algorithms was small, and the PSO algorithm completed tasks about 5 ms earlier. But as with the increase of tasks, the complexity of the scheduling increased, and the IPSO algorithm tended to be better in the operation time. As shown in Fig. 4, when the number of tasks was 100, the IPSO algorithm completed tasks about 8 ms earlier. Therefore, it could be concluded that the IPSO algorithm was superior in scheduling a large number of task resources.

Fig. 4 Comparison of the operation time between the PSO and IPSO algorithms

Load at different nodes

The resource load of the two algorithms at 10 nodes of virtual machine is shown in Fig. 5. The loads on different virtual machines were different, which was attributed to the difference of processing ability. The comparison of the load between the two algorithms suggested that the load of the IPSO algorithm was more even, and the nodes with strong processing ability were allocated with more tasks; the load of the PSO algorithm was not even. Therefore, the IPSO algorithm was more excellent in resource optimal scheduling.

Conclusion

Aiming at the problem of virtual machine resource scheduling in cloud computing environment, an IPSO algorithm was proposed in this study. The algorithm was improved based on the traditional PSO algorithm. The improvement made was to change the constant coefficients of cognition and social items to coefficients that could change with the number of iterations in the formula of velocity variation. In this study, the simulation experiments of the IPSO and traditional PSO algorithms were carried out using Cloud-Sim. The following results were obtained. When the number of iterations was between 0 and 100, both algorithms were searching the optimal solution, and the IPSO algorithm was more efficient; but the convergence of the traditional PSO had a fluctuation when the number of iteration was between 100 and 140. When the number of iterations was 300, the traditional PSO algorithm had premature phenomenon and entered a stationary phase without finding the global optimal solution; the IPSO algorithm entered a stationary phase when the number of iterations was 240, but it began to explore the optimal solution when the number of iterations exceeded 360. In operation time, when the number of tasks was 20, the traditional PSO algorithm was slightly superior

Fig. 5 The load at nodes of different virtual machines

and was 5 ms faster; when the number of tasks increased, the operation time of the two algorithms also increased, but the improved PSO algorithm began to dominate; when the number of tasks was 100, the improved PSO algorithm was about 8 ms faster. The loads of different virtual machines were different when the two algorithms were used; the virtual machine load under the IPSO algorithm was more balanced, and the virtual machine with stronger processing ability was allocated with more tasks.

Acknowledgement This study was supported by the Teaching Reform Project of Higher Education in Zhejiang Province (No. jg20160374).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Tsai WT, Sun X, Balasooriya J (2010) Service-oriented cloud computing architecture. In: International Conference on Information Technology: New Generations. IEEE, 2010, pp 684–689
2. Bölöni L, Turgut D (2016) Value of information based scheduling of cloud computing resources. *Future Gener Comput Syst* 71:1–9
3. Chen ZG, Zhan ZH, Li HH, et al (2015) deadline constrained cloud computing resources scheduling through an ant colony system approach. In: International Conference on Cloud Computing Research and Innovation. IEEE, 2015, pp 112–119
4. Abdullahi M, Ngadi MA (2016) Hybrid Symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment. *PLoS One* 11(6):e0158229
5. Zhang L, Tong W, Lu S (2015) Task scheduling of cloud computing based on Improved CHC algorithm. In: International Conference on Audio, Language and Image Processing. IEEE, 2015, pp 574–577
6. Chen J, Xu J, Hui B (2017) Cloud Computing Resource Scheduling based on Improved Semantic Search Engine. In: International Conference on Intelligent Information Processing. ACM, 2017, pp 1–5
7. Zhou H, Deng S, Huang H (2016) Stability property of clouds and cooperative scheduling policies on multiple types of resources in cloud computing. *J Supercomput* 72(6):2417–2436
8. Zhang Y, Wang S, Ji G (2015) A comprehensive survey on particle swarm optimization algorithm and its applications. *Math Probl Eng* 2015 (1): 1–38
9. Akbar MF, Munir EU, Rafique MM, et al (2017) List-based task scheduling for cloud computing. In: IEEE International Conference on Internet of Things, IEEE, 2017, pp 652–659
10. Sun J, Wu X, Palade V et al (2015) Random drift particle swarm optimization algorithm: convergence analysis and parameter selection. *Mach Learn* 101(1–3):345–376
11. Mahi M, Ömer KB, Kodaz H (2015) A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem. *Appl Soft Comput J* 30(C):484–490
12. Section (2015) An improved particle swarm optimization algorithm based on centroid and exponential inertia weight. *Computer Eng Appl* (1): 1–14
13. Gou J, Guo WP, Wang C et al (2017) A multi-strategy improved particle swarm optimization algorithm and its application to identifying uncorrelated multi-source load in the frequency domain. *Neural Comput Appl* 28(7):1635–1656
14. Filho MCS, Oliveira RL, Monteiro CC, et al (2017) CloudSim Plus: a cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In: Integrated Network and Service Management, IEEE, pp 400–406

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.