Resource Scheduling Scheme Based on Improved Frog Leaping Algorithm in Cloud Environment

Senbo Chen[1, 2]

[1]*School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China*
[2]*School of Computer Science and Technology, Nantong University, Nantong 226019, Jiangsu, China*

Abstract
This article aims to optimize resource scheduling in cloud environment. Firstly, the proposed scheme takes into consideration the limitations of original shuffled frog leaping algorithm while using its improved version to optimize resource scheduling in cloud environment. Secondly, by introducing the idea of particle updating in local research, the improved scheme carries on well in chaotic perturbations of optimal individuals when the convergence speed is accelerating at the same time. Thus, the probability of local optimal emergence is gradually reduced. Finally, the scheme embodies CloudSinm platform to carry out the simulation experiment well. The experimental results show that the scheme has reduced the time for cloud computing to complete the task more reasonable load distribution with more reasonable resource allocation.
Key words: CLOUD COMPUTING, FROG LEAPING ALGORITHM, RESOURCE SCHEDULING, SCHEDULING SCHEME

1. Introduction

The task scale in the cloud computing environment has been always in a growing state recently while the resources also gradually show heterogeneity. However, the resource management service quality of cloud computing directly affects the performance indexes of the computer system. Currently, how to get the optimal choice from cloud computing resource scheduling strategies is always one of the hotspots in the study of modern cloud computing.

In the research literature, foreign scholars mainly put forward a combination of static and dynamic algorithms for cloud computing resource scheduling. With the static scheduling algorithm, on the one hand, M tasks are allocated to N resources. On the other hand, this kind of resource scheduling based on cloud computing, to a certain extent, is often required to prepare for the release of resource dynamic application [1]. As far as its nature is concerned, cloud computing is often the process of providing external services, which often requires to embody the basic cost of resource consumption. In fact, the static algorithm is difficult to meet full requirements of cloud computing resource scheduling fundamentally, which leads to a waste of resources while the dynamic scheduling algorithm has relatively more applications, wherein cloud computing resources are always in a multiple-constrained state. However, the applications of genetic algorithm and particle swarm optimization (PSO) algorithm, to a certain extent, often need to fundamentally allow for resource scheduling features using frog leaping algorithm in cloud environment before further simulating the biological behavior in nature. The proposed scheme by coping with the problems existing in the above-mentioned algorithms ultimately reflects that it has both the advantage of fast speed and a good performance. Thus this cloud computing resource scheduling scheme is relatively better than the original one [2].

In terms of optimal cloud computing resource scheduling, a better result can be achieved in simulation test and analysis when the proposed strategy is employed based on improved shuffled frog leaping algorithm [3].

2. Analysis of cloud computing resource scheduling

The so-called cloud computing system often provides high quality service, and for maximizing the use of resources, continuously reduces waste. Cloud computing system is generally characteristic of heterogeneity, and it is most suitable for operating on cloud computing resources in the state of resource competition [4].

In current cloud computing, resources are often combined with a virtual method and provided for cloud computing users. Any resource is represented by VM_i, as is shown in formula (1):

$$VM_i = \{CPU, Memory, BandWidth, MIPS, Cost, \cdots\} \quad (1)$$

The size of computer memory is denoted by $Memory$, running cost by $Cost$, computer network bandwidth by $BandWidth$ and the number of instructions executed by CPU by $MIPS$. Cloud computing users in the submission of the task is supposed to well prepare for of specific descriptions, as formula (2) shows below:

$$Task_i = \{Length, PesNumber, FileSize, OutputSize, CPU, Memory, BandWidth \cdots\} \quad (2)$$

The size of output and input file is expressed by $OutputSize$ and $FileSize$ respectively while length of task by $Length$. The application of cloud computing often needs to combine running cost and the task is effectively completed in the analysis of user requirements[5]. In the simplified computer complexity, completion time of cloud computing users is denoted expressed by QOS.

In cloud computing system, task i is indicated by $Task[i]$ while resource j by $VM[j]$. The execution time concerning $Task[i]$ based on $VM[j]$ is thus recorded as $time(Task[i] \to VM[j])$. However, X as cloud computing resource scheduling scheme, its completion time is expressed by formula (3) accordingly:

$$Time(X) = \max_{j=1}^{t} \sum_{i=1}^{k} Time(Task[i] \to VM[j]) \quad (3)$$

Where in task number is denoted by k and quantity of cloud computing resource by l.

3. Resource scheduling scheme based on improved frog leaping algorithm in cloud environment

3.1. Traditional shuffled frog leaping algorithm

Traditional shuffled frog leaping algorithm was first proposed to mainly embody the simulation process of frog foraging behavior and integrated in particle swarm optimization algorithm. With the advantage of memetic algorithms, it has gained wide range application.

This algorithm first generates F frog particles with the division of population, and then combines the initial population in portfolio, as is expressed by $P = \{X_1, X_2, \cdots, X_F\}$, wherein the potential solution to a problem is expressed with a frog[6]. For frog k, its position is $X_k = \{X_{k1}, X_{k2}, \cdots, X_{kn}\}$. The fitness value of each frog is calculated with $f[X_k]$ and arranged in a descending order in preparation for basic permutation application before it is distributed equally in M populations. For population j, it is expressed by E_j and tis value is calculated with the formula below:

$$E_j = \{S_{j+m(t-1)} \in P_t \mid 1 \le l \le N\} \quad (4)$$

The number of frogs in the population is denoted by N and $F = M \times N$.

Secondly, for local search, the worst frog in the population is labeled as X_w, the location of which updates in a way expressed by formula (5)[7].

$$\begin{cases} X_{new} = X_w + D \\ D = r \times [X_b - X_w] \end{cases} \quad (5)$$

The location of the best frog is labeled as X_b and its random number is expressed by r.

Once $X_{new} > X_w$, then $X_{neww} = X_w$; otherwise, $X_b = X_g$. After re-updating, once $X_{new} < X_w$, a frog is generated at random and labeled as X_w. In the execution of the above process by the population, the number of iterations will be in a terminated state [8].

Finally, for the global information exchange, after local search is done combinations will be carried out to generate new populations, which are to be updated with reference to the arrangement of their fitness value for searching operations till optimal termination conditions are met eventually[9].

3.2. Improvement on shuffled frog leaping algorithm

Generally speaking, improvement on shuffled frog leaping algorithm often requires an update policy in analysis of the fundamentals. However, the local search performed in a restricted area tends to be confronted with relatively slow convergence speed. Thus, optimal probabilities will emerge once the search is stuck in a local area. The updating of individual frog is shown by formula (6):

$$D = r \times c \times (X_B - X_w) + W \quad (6)$$

$$W = [r_1 \times w_{1,\max}, r_2 w_{2,\max}, \cdots, w_{n,\max}]^T \quad (7)$$

$$X_{new} = \begin{cases} X_w + D & \|D\| \le D_{\max} \\ X_w + \frac{D}{\sqrt{D^T D}} D_{\max} & \|D\| > D_{\max} \end{cases} \quad (8)$$

r_i is any number located in [1,1]; the range of c is between 1 and 2; and the maximum uncertainty is expressed by $w_{i,\max}$.

The optimization of particle swarms is related to the location of particles. It is aimed to achieve speed update primarily, to find the optimal solution fundamentally and leads to effective update of each frog's location and its due application in applying particle update mechanism[11]. In the longest coherent expression, a frog's mobile step is expressed by formula (9):

$$D = R[r_2 D' + r_3(x_g' - x_w + r_c(X_b - X_w) + W \quad (9)$$

The weighting factor is denoted by R .In terms of the best optimizing process of frog strategy, the first step is to assume

$$X_g = \{X_{g1}, X_{g2}, \cdots, X_{gd}\} \quad (10)$$

Where in X_{gi} is mapped and improvement on shuffled frog leaping algorithm is done. The value range is calculated with formula (11):

$$z_i = x_L \times (x_u - x_L) \times x_{gi} \quad (11)$$

The minimum value of the variable is denoted by x_L while its maximum value by x_U.

The second step is to carry out iterative processing for formula (10). After multiple iterations, point chaos sequences amount to m, i.e., $z_i^{(m)} (m = 1, 2, \cdots)$ The third step is obtain a feasible solution $x_i^{(m)} = (x_{i1}^{(m)}, x_{i2}^{(m)}, \cdots, x_{id}^{(m)})$, as shown in formula (12):

$$x_{gi}^{(m)} = x_L + (x_u - x_L) z_i^{(m)} \quad (12)$$

The final step is to base substitution on individuals after chaotic mapping and to replace some individuals in terms of probability P_g. The calculating process of P_g is shown in formulas (13) and (14):

$$P_g = \begin{cases} 0 & CR > q \\ 1 & CR \le q \end{cases} \quad (13)$$

$$q = \ln t / (1 + \ln t) \quad (14)$$

3.2.1. Layout procedure

The first step in the layout procedures is to identify the encoding scheme, which is represented in formula (15).

$$X = [x_1, x_2, \cdots, x_p] \quad (15)$$

$$x_i \in [1, l \pm 1)$$

The special encoding way in combination of the search of optimal solution is expressed by (16).

$$X = [[x_1], [x_2], \cdots, [x_p]] \quad (16)$$

$[x_i]$ is mainly a resource sequence of x_i rounding down to integer.

In the summary of completion time of the task, the basic definition of fitness function is represented by(17).

$$Fitness(X) = \frac{1}{1 + \frac{Time(X) - Time(X_t^b)}{Time(X) - Time(X_t^w)}} \quad (17)$$

x_t^w is used to denote a resource scheduling scheme that spends the longest time on completing the task while x_t^b the shortest[12].

3.2.2. Solution steps

The solution steps comprise in detail the following. Firstly, assume that the number of the initial population is N_{pop} and the population is *memeplex* with its size as N_m at the same time. Then the iterations of *memeplex* is *Iter* [13].

Thirdly, in encoding frog individuals, update each sub-population X_w based on formula (17). Fourthly, shuffle the populations and calculate the fitness value of frogs to achieve chaos optimization of frogs X_g in a descending order. Fifthly, when the termination conditions are met, the optimizing process gradually ends with selecting the optimal cloud computing resource scheduling scheme in combination with the optimal series.

4. Simulation experiment

4.1. Experiment environment

The simulation experiment is carried out on a computer with Window XP operating system and 2GB memory, and embodies Cloudsim simulation software to prepare for its application. The analysis is done in terms of ISFLA simulation results and in combination with the comparison between SFLA and genetic algorithm [14].

4.2. Comparison of convergence speed before and after improvement

The size of resources is set as 20 and the cloud computing systems amounts to 20000. The solution process of optimal scheduling scheme is shown in Figure 1.

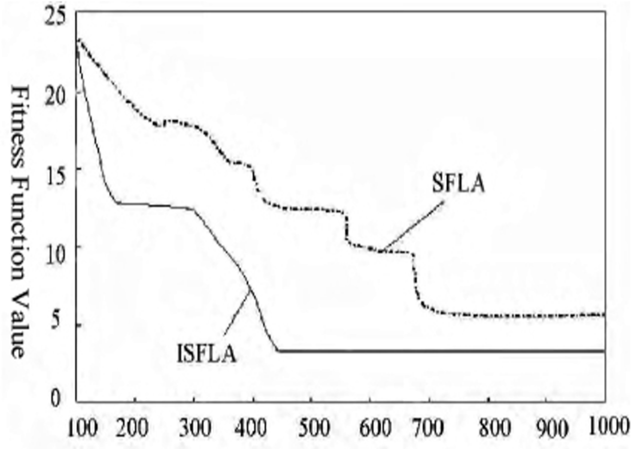

Figure 1. A comparison between iterations of optimal solution by chaos frog leaping algorithm before and after improvement

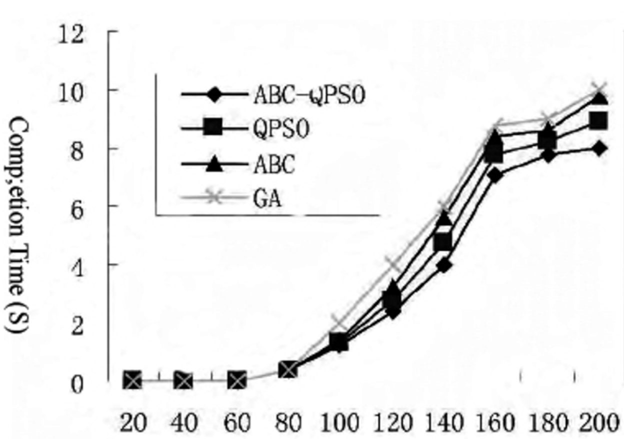

Figure 2. Task number

In the comparison, it can be seen that iterations of optimal solution by chaos frog leaping algorithm after improvement embodies the update policy and applies the idea of updating particles, during which process the inertia of the last update is extended and the simulation information of the optimal individual is introduced. Furthermore, convergence speed of the algorithm continues to accelerate. In the use of chaos optimization strategy for reference, direct optimization of optimal individual chaotic perturbations is realized and the population is reduced, which leads to the improvement of the speed of problem-solving in terms of resource scheduling.

4.2. Result analysis

In the analysis of small scale tasks, the performance of this algorithm, compared with that of others, is often combined with the model of cloud computing system. In the allocation of resource nodes, this algorithm has done well in reasonable allocation of tasks, application of resource scheduling till optimal scheduling scheme is obtained. The change of the task time is mainly shown in Figure 2.

Once there are fewer tasks, the user will encounter a smaller probability of both resource competition and conflict.

For large-scale tasks, when for example, the total size of resources is 50, the comparison among different algorithms is displayed in Figure 3.

In application of these algorithms, when the number of tasks increases, there are both a more intense competition and an increased conflict probability. During the increase of task completion time, the improved shuffled frog leaping algorithm has a great advantage. This advantage, to a certain extent, can be used to speed up the convergence rate. In the practical analysis, the local optimal solution can be avoided effectively with the chaos optimization strategy introduced in time.

The effect of resource nodes on task completion time is shown in Figure 4.

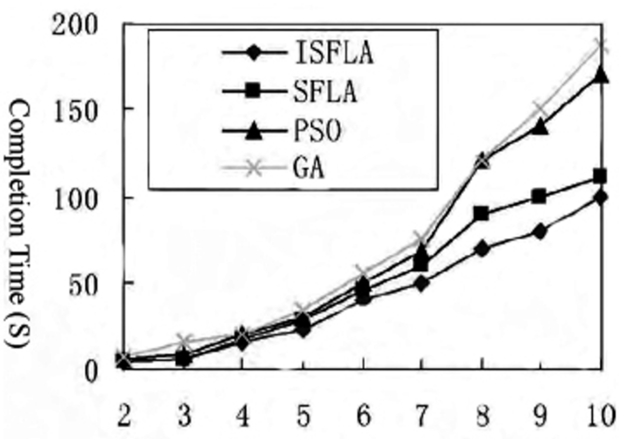

Figure 3. Comparison of large-scale task completion time among different algorithms

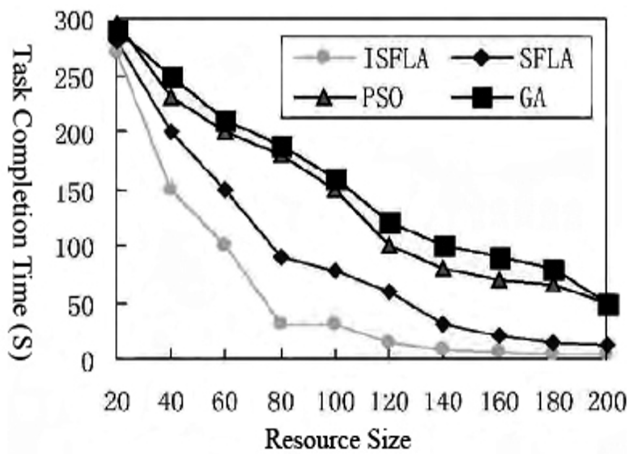

Figure 4. Changes in relationship between resource nodes and task completion time

When nodes are increasing, the completion time will be decreasing continuously with resource competition also in a gradual weakening of the state. On the contrary, once total resources exceed a certain node, this magnitude will gradually decrease and by increasing the number of nodes of cloud computing resources, the efficiency of problem solving is improved continuously in cloud computing task scheduling.

For a comparative analysis of load balance between resource nodes, 100000 tasks are, for example, allocated with resource nodes, as shown in Figure 5.

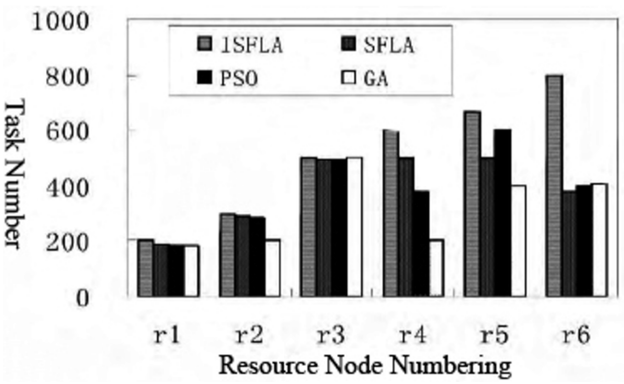

Figure 5. Resource load distribution of different algorithms

Resource scheduling based on improved SFLA tends to have a relatively strong processing capability in fair utilization of resources.

5. Conclusions

In general, rational utilization of resources in cloud environment is always one of the current concerns in the cloud computing area. This article through the analysis of the solving process of shuffled frog leaping algorithm solves the problem of slow solving speed and provides an effective solution to the problem of being trapped in local optima. In the application of improved shuffled frog leaping algorithm in resource scheduling, the results of the simulation test show that the optimal scheme for cloud computing resource scheduling can be obtained with the realization of solution finding for large-scale tasks.

Acknowledgements

This work was supported by 61300167.(National Natural Science Foundation of China, NSFC)

References

1. Luo J.P., Li X., Chen M.R. (2012) Guaranteed QoS resource scheduling based on improved shuffled frog leaping algorithm in cloud environment. *Computer Engineering and Application*, 48(29), p.p.67-72.
2. Klemm M.,Haueisen J.,Ivanova G. et al.(2009) Independent component analysis: comparison of algorithms for the investigation of surface electrical brain activity. *Medical and Biological Engineering and Computing: Journal of the International Federation for Medical and Biological Engineering*, 47(4), p.p.413-423.
3. Zhang M. (2015) Improved shuffled frog leaping algorithm in resource scheduling of cloud computing. *Computer Applications and Software*, No.4, p.p.330-333.
4. Takeda A.,Sanuki N.,Kunieda E. et al.(2009) Stereotactic body radiotherapy for primary lung cancer at a dose of 50 Gy total in five fractions to the periphery of the planning target volume calculated using a superposition algorithm. *International Journal of Radiation Oncology, Biology, Physics*, 73(2), p.p.442-448.
5. Ning F.F., Wang J.X.(2015) Resource scheduling based on estimation of distribution shuffled frog leaping algorithm in cloud environment. *Microcomputer Applications*, 31(7), p.p.59-61,65.
6. Milov A.D., Samoilova R.I., Tsvetkov Y.D. et al.(2009) Structure of self-aggregated alamethicin in ePC membranes detected by pulsed electron-electron double resonance and electron spin echo envelope modulation spectroscopies. *Biophysical Journal*, 96(8), p.p.3197-3209.
7. Liu W.J., Zhang M.H., Guo W.Y. (2011) Cloud computing resource schedule strategy based on MPSO algorithm. *Computer Engineering*, 37(11), p.p.43-44,48.
8. Lin W.W., Qi D.Y. (2012) Survey of resource scheduling in cloud computing. *Computer Science*,39(10), p.p.1-6.
9. MarkakiV.E.,AsvestasP.A.,MatsopoulosG.K. et al.(2009) An iterative point correspondence algorithm for automatic image registration: an application to dental subtraction radiography. *Computer Methods and Programs in Biomedicine*, 93(1), p.p.61-72.
10. Hao L., Cui G., Qu M.C., Zhang K.(2014) A cost constrained resource scheduling optimization algorithm for reduction of energy consumption in cloud computing. *Chinese High Technology Letters*, 24(5), p.p.458-464.
11. Lim G.H., Huber G. (2009) The tethered infinitesimal tori and spheres algorithm: a versatile calculator for axisymmetric problems in equilibrium membrane mechanics. *Biophysical Journal*, 96(6), p.p.2064-2081.
12. Zhang H.W., Wei B., Wang J.D., He J.J.(2014) Cloud resource scheduling method based on estimation of distribution shuffled frog leaping

algorithm. *Application Research of Computers*, No.11, p.p.3225-3228.
13. Hwang A.B., Bacharach S.L., Yom S.S. et al. (2009) Can positron emission tomography (PET) or PET/Computed Tomography (CT) acquired in a no treatment position be accurately registered to a head-and-neck radiotherapy planning CT?. *International Journal of Radiation Oncology, Biology, Physics*, 73(2), p.p.578-584.
14. He Z.M., Zhang Y., Gao L. (2014) Cloud computing resource schedule strategy based on QPSFLA algorithm. *Computer Knowledge and Technology*, No.2, p.p.311-314.

A Perturbation Bound of the Partitioned Linear Response Eigenvalue Problem

Zhongming Teng

College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou 350002, Fujian, China.
Corresponding Email: peter979@163.com

Xuansheng Wang

School of Software Engineering, Shenzhen Institute of Information Technology, Shenzhen 518000, Guangdong, China.

Jinfu Liu

College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou 350002, Fujian, China.