Research on the Resource Scheduling of the Improved SFLA in Cloud Computing

Yue Miao, Fu Rao and Luo Yu

China West Normal University, School of Computer, Sichuan Nanchong 637002

Abstract

It has always been a key topic in the current research how to make rational resource scheduling in the cloud computing environment. In this paper, the status of cloud computing resources is first analyzed, to point out the existing problems, and then combined with the characteristics of resource scheduling in cloud computing, the Shuffled Frog Leaping Algorithm is introduced. First, in its stage of subgroups classification, the chaos strategy is introduced and in the internal search the positive learning strategy is introduced, which makes the improved frog leaping algorithm gain good convergence, reduces the time of global search and optimization. Through the CloudSim platform, it shows that this algorithm can improve the efficiency of task processing and make the resource scheduling in cloud computing rational and effective.

Keywords: *Improved Frog Leaping Algorithm, resource scheduling, cloud computing*

1.Introduction

Cloud computing is a computing pattern developing along with distributed processing, parallel processing and grid processing, and it is the result of integrating the concepts such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). As the number of tasks in cloud computing rises exponentially and the cloud computing servers have to timely process large numbers of tasks. Besides, the structure of the network in cloud computing can neither guarantee the reasonable application of resources, nor guarantee the service quality of every cloud computing task [1-2]. Reference [3] describes that it is effective to some extent to introduce Swarm Intelligence Algorithm in virtual scheduling of cloud computing to carry out resource scheduling. Reference [4] applies the improved genetic algorithm in the task scheduling in cloud computing, the algorithm makes a certain effect. Reference [5] states that with the combination of the ACO algorithm and K-medoid algorithm, ACO-K-medoid resource allocation optimized algorithm is proposed based on the cloud computing environment, which gains optimal computing resources and improves the efficiency of cloud computing. Reference [6] proposes a cloud computing resource scheduling model based on Chaos Particle Swarm Algorithm, which improves the efficiency of resource utilization and is better in practicality and feasibility. Reference [7] uses the Discrete Particle Swarm Algorithm to solve the problem. With regard to the characteristics of the model, combined with the real situation of cloud computing service operation, computing samples were designed to carry out stimulation test. Reference [8] improves the starter strategy of resource scheduling in the environment of cloud computing in order to effectively enhance the overall system processing capability in this environment. Through simulation experiments, the results show that this method can effectively avoid the unbalanced load handling, and improve the overall processing capability of the system.

In this paper, in order to better solve the problem of resource scheduling in cloud computing, the Shuffled Frog Leaping Algorithm (SFLA) is introduced. Considering the problems in the original algorithm such as premature convergence and the improvement of global optimization, we introduce the chaos strategy is in the subgroup classification

stage of population, and the concept of positive learning in the internal searching stage. Through experiments, it shows that the algorithm performs well in convergence speed and accuracy, and by the application on the Cloudsim platform, the results show that the efficiency of resource scheduling has been significantly improved.

2. The Main Issues of Cloud Computing Resources

2.1 The Current Status of Cloud Computing Resources

In the framework of cloud computing, there are thousands of access requirements from cloud clients, so the cloud computing servers have to fully consider the quantity of cloud resources each cloud user can use, and the duration of using the resources. Currently, most cloud computing resource scheduling algorithms are mainly concerned about the total time spent on the completion of tasks among the cloud clients. This algorithm can not cover all of the cloud clients, and may lose the cloud client who spends the least time in completing the task.

2.2 The Service Quality in Cloud Computing

In the environment of cloud computing, the service quality of resource allocation is related to the satisfaction of cloud users. Because the cloud users have different requirements on the resource allocation and scheduling capabilities of the cloud computing system, the calculation of cost, mission time and user waiting time has to be considered in the service quality of cloud computing. In this paper, we mainly take the completion time and resources into account in the service quality of cloud computing:

Question 1: representation of task execution time. In this paper, time is defined as $Timer(Task[i], Source[j])$, which represents the time needed for allocating task $Task[i]$ to the resource $Source[j]$. From the requests of virtual machine of the cloud client server, the completion time of any resource scheduling program is indicated as:

$$Timer(x) = \max_{j=1}^{l}(\sum_{i=1}^{k}(Task[i], Source[j])) \tag{1}$$

In formula (1), l represents the number of virtual machines in the system, the quantity of resources needed for task k.

Question 2: the bandwidth needed for the implementation of tasks. In this paper, the gap value between the bandwidth needed and resource bandwidth is indicated as $BandWidth(Task[i], Source[j]))$, which represents the expected bandwidth needed to allocate and implement task $Task[i]$ on $Source[j]$. The resources required for any resource scheduling plan are represented as:

$$BandWidth(x) = \max(\sum_{j=1}^{l}\sum_{i=1}^{k}BandWidth(Task[i], Source[j]))) \tag{2}$$

In formula (2), l represents the number of virtual machines in the system, the quantity of resources needed for task k.

3. Shuffled Frog Leaping Algorithm

Its basic idea is: a group of initialized solutions (population) is generated from the space of random solutions, and then the entire population is divided into multiple sub-groups, and the frogs in the sub-groups make internal search in accordance with a certain search strategy. After a specified number of internal searches in subgroups, mix the frogs and then classify them into subgroups again by sequencing, and thus guarantee that the information is fully exchanged among all the subgroups.

(1)Subgroup classification

The number of frogs in the population is set as M, the initial population as S, the number of subgroups as k, and the number of candidate solutions in the subgroups is set as n。 $x_i = (x_{i1}, x_{i2}, ... x_{iD})$ is used to represent the number i candidate solution, and here D represents the dimension of candidate solutions, $S = (x_1, x_2, ... x_M)$.

(2)Search in subgroups

It is assumed that X_{zbest} is the candidate solution in the whole population with the best fitness, X_{best} the candidate solution in the subgroups with the best fitness, and X_{worst} the candidate solution in the subgroups with the worst fitness. When searching within the subgroups, it is mainly to update the worst candidate solution X_{worst}. The process is described below:

$while(n \leq T)$

$$X_i^{'} = X_{worst} + t \times (X_{best} - X_{worst})$$

$if\ X_i^{'} > X_{worst}$

$$X_i^{'} = X_{worst}$$

$else$

$$X_{random} = X_{worst}$$

$endwhile$

Here, $X_i^{'}$ represents a new solution, and t is the random number within the range of (0,1). If the new solution generated is superior to X_{worst}, the former will replace the latter and at the same time the searching continues; or else, the random solution is used for replacement.

$$X_i^{'} = X_{worst} + t \times (X_{best} - X_{worst}) \qquad (4)$$

(3)Global information exchange

When the internal update of all the subgroups is complete, the subgroup classification and internal searching are implemented again, which is repeated until the final condition is satisfied.

4. Improved Shuffled Frog Leaping Algorithm

4.1 The Content Improved

The way to classify the subgroups more rationally is an important part in the leapfrog algorithm, and the quality of classification will have a direct impact on the quality of the overall speed of convergence of the algorithm and the corresponding solution. The way commonly used is to generate the initial solution of the algorithm with random initialization. In the population initialization stage, a positive learning mechanism is introduced, to generate a positive solution for each candidate solution. By comparing the candidate solutions and positive solutions, a solution of a short distance is selected as the solution of the initial population. In the internal search stage of subgroups, the chaos mechanism is introduced.

(1)Positive learning stage

Some good effects have been achieved by applying positive learning strategy in the process of iteration, used in the solution process for high dimension problems and according to the definition of positive learning strategy. Set $X = (x_1, x_2 .... x_D)$ as a point in space, and $[\min_j, \max_j]$ represents the minimum and maximum values,

meeting $x_j \in [\min_j, \max_j]$, so the positive solution of X is: $X^* = (x_1^*, x_2^* \ldots \ldots x_D^*)$. In order to better prevent the information loss in the searching space of convergence, a positive value can be introduced in the dynamic searching space, and the formula f(a,b) can be introduced as a result to guarantee the midpoints of the variable are in normal distribution. In this way, the solution of the space can be ensured to be effective.

$$\begin{cases} X_{i,j}^* = k \times (a_j(i) + b_j(i)) - X_{i,j} \\ X_{i,j}^* = f(a_j(i), b_j(i)) \quad X_{i,j}^* \notin [X_{\min}, X_{\max}] \end{cases} \quad (5)$$

The positive solution generated from formula(5)can be compared with the solution generated from formula(4), and thus the nearer one can be selected as the initial population, so as to accelerate the convergence.

(2)Chaos stage

$$for\ i = 1\ to\ N\ do$$

$$\quad for\ k = 1\ to\ K\ do$$

$$\quad\quad Ch_{k,j} = \int_i^N \ln(\pi Ch_{k,j}) \quad (6)$$

$$\quad endfor$$

$$\quad X_i^{'} = X_{worst} + Ch_{k,j} \times t \times (X_{best} - X_{worst})$$

$$endfor$$

4.2 Improved Algorithm Flow

```
begin
for(i=1;i<=Max;i++)
    {sequence and classify the candidate solutions xm
     if(f(Xm)<T)
       break;
       for(h=0;h<k;j++)// traverse the subgroups of the frog swarm algorithm
         for(j=1;j<=P;j++)// search and solve all the frog solutions in the
subgroups
           {Use (6), to solve X'
           if(f(X')<f(Xm))
             Xm=X'
           else{
               Xb =Xm;
               if(f(X')<f(Xm))
               Xm=X'
           According to the positive learning strategy, generate Xm
               }
           Classify and optimize the results in the subgroups according to chaos
algorithm, and introduce formula(6)
```

$$X_i^{'} = X_{worst} + Ch_{k,j} \times t \times (X_{best} - X_{worst})$$

```
                Endif
         Endif
           Endfor
       Endfor
     Endif
   Endfor
```

5. Experimental Simulation

To better embody the superiority of the algorithm in cloud computing, the performance of the algorithm is first tested in this paper, and the resource scheduling is tested on the simulation platform of cloud computing.

5.1 Performance Test for the Algorithm

The basis functions are adopted to make comparative test. Through the comparison with Elbeltagi's [11] improved frog leaping algorithm and the basic frog leaping algorithm, the effectiveness of the proposed algorithm is verified.

(1)Sphere function

$$f_1(x) = \sum_{i=1}^{N}(-x_i \sin(\sqrt{|x_i|})), -500 \leq x_i \leq 500$$

(2)Ackley function

$$f_2(x) = -20\exp[-0.2\sqrt{\frac{\sum_{i=1}^{N}x_i^2}{N}}] - \exp[\frac{\sum_{i=1}^{N}\cos(2\pi x_i)}{N}]$$
$$+20+\exp(1) \; -32.768 \leq x \leq 32.768$$

(3)Rosenbrock function

$$f_3(x) = \sum_{i=1}^{N-1}[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] -5 \leq x \leq 5$$

Table 1 The Optimization Results of the Two Algorithms

Function	Algorithm	Average Maximum	Average Minimum
f1	Basic frog leaping algorithm	0.032514	0.000114
	Algorithm of this paper	0.009725	0.0000008
f2	Basic frog leaping algorithm	0.072412	0.009754
	Algorithm of this paper	0.011312	0.004312
f3	Basic frog leaping algorithm	0.082513	0.007652
	Algorithm of this paper	0.048716	0.006127

5.2 Cloud Computing Simulation Platform

In the experiment, the cloud computing platform Cloudsim is used to simulate the cloud computing environment. There are 100 experimental tasks, and each task varies in the requirements of task length and bandwidth, as shown in Table 2. At the same time, 10 virtual machines are set, and the number of virtual CPUs and the corresponding resource requirements of each virtual machine are shown in Table 3.

Table 2. Parameters of Task Operation

Task No.	Task Length	Expected Bandwidth
Task[1]-Task[10]	100+10×t	20+10×t
Task[11]-Task[20]	200+10×t	30+10×t
Task[21]-Task[30]	300+10×t	40+10×t

Task[31]-Task[40]	400+10×t	50+10×t
Task[41]-Task[50]	500+10×t	60+10×t
Task[51]-Task[60]	600+10×t	70+10×t
Task[61]-Task[70]	700+10×t	80+10×t
Task[71]-Task[80]	800+10×t	90+10×t
Task[81]-Task[90]	900+10×t	100+10×t
Task[91]-Task[100]	1000+10×t	110+10×t

Table 3. Resource Parameters

CPU No.	CPU quantity	Bandwidth
Cpu[1]	1	100
Cpu[2]	2	90
Cpu[3]	2	80
Cpu[4]	3	90
Cpu[5]	2	120
Cpu[6]	4	90
Cpu[7]	2	100
Cpu[8]	3	80
Cpu[9]	2	100
Cpu[10]	4	90

Under the same initial condition, the previously mentioned algorithms are used to make simulation. By setting 200 times of iteration for the algorithms, experiments are made in terms of completion time and bandwidth consumption, the results of which are shown in Table 1-2.
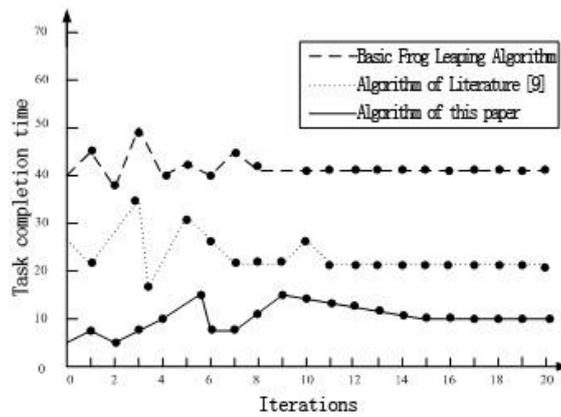

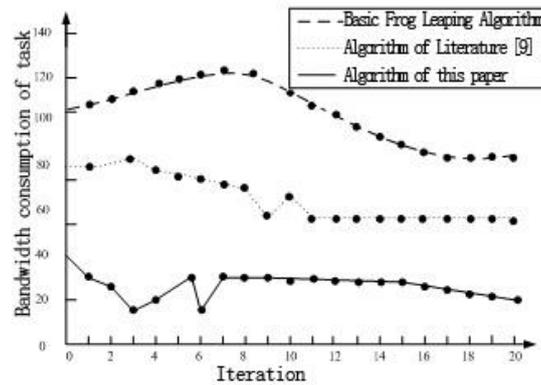**Figure 4. Comparison of Task Completion Time**

Figure 5 Comparison of Bandwidth Consumption

From figure 1, it can be found that the scheduling plan of the algorithm of this paper is superior to the algorithm of literature [9] and Basic Frog Leaping Algorithm. To make sure the broadband permits, the completion time of the algorithm presented in this paper is minimal, with the average completion time probably within 1000MS, better than the 3000MS of the algorithm of literature [9] and 4500MS of Basic Frog Leaping Algorithm. Figure 5 shows the comparison of consumption of different algorithms. The algorithm presented in this paper consumes the least bandwidth, superior to the algorithm of literature [9] and Basic Frog Leaping Algorithm. From the two points above, it can be proved that the proposed algorithm is more suitable to solve the problems concerning service quality and resource scheduling in the cloud computing environment.

6. Conclusion

How to make better resource scheduling in cloud computing has always been a focus of research. In this paper, an improved frog leaping algorithm is proposed to further guarantee the reasonable resource scheduling on the cloud computing platform. The results of simulation experiments show that the improved frog leaping algorithm has good resource optimization and scheduling ability. The research on cloud computing has not stopped, which needs further study, so as to optimize the resource performance in the cloud computing environment.

References

[1] M. Armbrust, A. Fox and R. Griffith, "A view of cloud computing", Communications of the ACM, vol. 53, no. 4, **(2009)**, pp. 50-58.
[2] X. Baomin, Z. Chunyan and H. Enzhao, "Job scheduling algorithm based on Berger model in cloud environment", Advances in Engineering Software, vol. 42, **(2011)**, pp. 419-425.
[3] D. Wei, "New energy application in the cloud computing data center: research status and tendency", Chinese Journal of Computers, vol. 36, no. 3, **(2013)**, pp. 582-598.
[4] Z. Zongbin and D. Zhongjun, "Cloud computing task scheduling algorithm based on improved GA", Computer Engineering and Applications, vol. 49, no. 5, **(2013)**, pp. 77-80.
[5] M. Ying, "ACO-K midpoint resource optimization algorithm based on cloud computing", Computer Engineering and Applications, vol. 49, no. 5, **(2013)**, pp. 103-107.
[6] Z. Yinghua, "Research on the resource scheduling optimization model under cloud computing environment", Computer Simulation, vol. 30, no. 5, **(2013)**, pp. 362-365.
[7] D. Yanyan, "PSO reliable resource scheduling under cloud computing environment", Computer Engineering and Applications, vol. 49, no. 18, **(2013)**, pp. 78-81.
[8] Z. Bo, "Improved load scheduling method based on starter strategy", Computer Engineering and Applications, vol. 49, no. 17, **(2013)**, pp. 73-77.

Authors

Yue Miao (1980～), MA, lecturer, research direction: Computer network, computer simulation.

Fu Rao (1982～), MA, lecturer, research direction: System application direction.

Luo Yu (1982～), MA, lecturer, research direction: Electronics application direction.