International Journal of Grid and Distributed Computing
Vol. 9, No. 3 (2016), pp.191-198
http://dx.doi.org/10.14257/ijgdc.2016.9.3.20

Research based on Effective Resource Allocation of Improved SFLA in Cloud Computing

Bin Ning, Qiong Gu and Yi Wang

(School of Mathematics and Computer Science,Hubei University of Arts and Science, Xiangyang, Hubei, 441053, China)
hubeiningbin@163.com

Abstract

It is the focus of current cloud research as how to reasonably use resources in cloud computing. In this paper, shuffled frog leaping algorithm is introduced into the cloud computing resource allocation, and aiming at its slow speed in getting solution and being easy to fall into local optimum, improvement is made. First, reverse learning algorithm is used to initialize the algorithm; then, cross factor is introduced into the local research; finally, evolutionary strategy is adopted in selecting the optimal solution combining the global search and local search to improve the performance of improved algorithm. It has be proven through the Cloudsim platform that algorithm in this paper effectively solves the problem of resource allocation and improves the efficiency of resource allocation.

Keywords: *cloud computing; shuffled frog leaping algorithm; reverse learning; cross f actor; evolutionary strategy*

1. Introduction

Cloud computing is a kind of collaborative and sharing compute mode with the fastest growing speed in internet in recent years. Cloud computing can help to realize sharing of software and hardware resources and avoid defects like information redundancy of distributed network [1]. Due to unbalanced distribution of resources, some requirements of cloud users cannot be fulfilled, thus efficiency of cloud computing is reduced. Given this reason, it is important to find out ways to distribute cloud computing resources in a reasonable manner. Researches showed that distribution of cloud computing resources is a target optimization problem. Foreign and domestic scholars introduced artificial intelligence algorithm into resource allocation, which had excellent effects to some extent. Reference [2] put forward a kind of resource scheduling method based on task delays, which method took full considerations of how to allocate local computing resources and latency time reasonably when immediate scheduling was unable to be carried out. Experimental results indicated that the method could improve efficiencies of global resource scheduling and local resource scheduling, but the algorithm convergence speed was slow. Reference [3] put forward a kind of genetic algorithm based on users' satisfaction, which method, putting users' fairness first, scheduled tasks into computational nodes where input data were, thus reducing network transmission expenditures. Simulated experiment showed that this algorithm was more adaptable to cloud computing environment in aspects of response time, fairness and users' satisfaction, but node consumption space was large. Reference [4] put forward a method of converting binary coding of quantum bit in genetic algorithm to real-number encoding and adopting rotating strategy and mutation operators to guarantee convergence of algorithm. Simulated experiment indicated that this algorithm could realize less service costs for cloud computing. Reference [5] introduced particle swarm optimization into cloud computing environment to optimize cloud computing targets. Simulated experiment showed that through this method, processing and transmission time and corresponding

ISSN: 2005-4262 IJGDC

transmission expenses could be optimized. Reference [6] came up with an idea of improving artificial bee colony algorithm under cloud computing resources, and through changing adjacent factors to enable modified algorithm enhance local search capability effectively and reduce average time in completing tasks. References [7] and [8] applied modified genetic algorithm and improved particle swarm optimization into cloud resource distribution respectively, which were effective and indeed reduced task completion time.

Based on researches of the above mentioned intelligence algorithm, this article puts forward a kind of cloud computing scheduling algorithm modified from shuffled frog leaping algorithm. Firstly, describing cloud computing resource scheduling and building cloud computing model; then, improving shuffled frog leaping algorithm. Wherein, adopting backward learning algorithm to initialize algorithm, introducing crossover factor into update of local search, and applying evolutionary strategy into the optimal solution in combination of global search and local search, thus improving performance of modified algorithm. Simulation platform testified this algorithm can effectively improve efficiency of cloud computing resources.

2. Description of Cloud Computing Resource Allocation

At preset, Map/Reduce Model is mainly applied in cloud computing. This model mainly breaks down large tasks into several small subtasks which are allocated to several virtual resource nodes and then results are returned. From the point of view of total consumed time to total cost, this article assumes that all subtasks are mutually independent, and through reasonably allocating resources for each subtask so as to minimize time for finishing general tasks and costs of consumed resources.

Task scheduling in cloud computing can be described as assigning n mutually independent subtasks to m resources ($m<n$), wherein, task is expressed as $Task=\{task_1,task_2,\cdots task_n\}$, $task_i$ represents No. i subtask, resource number is $\mathrm{Re}source=\{resource_1,resource_2,\cdots resource_n\}$, and $resource_i$ represents No. i resource. Therefore, each subtask can only be executed on one virtual resource, and the relation between $Task$ and $\mathrm{Re}source$ can be represented by the following matrix:

$$X=\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \tag{1}$$

In Formula (1), x_{ij} represents the relation between task i and No. j resource, and its value may be 0 or 1. x_{ij} equaling 0 means subtask i has not used No. j resource, otherwise, it has used. ETC_{ij} represents the time needed theoretically for subtask i in No. j recourse, therefore, ETC matrix is as follows:

$$ETC=\begin{pmatrix} ETC_{11} & \cdots & ETC_{1n} \\ \vdots & \ddots & \vdots \\ ETC_{m1} & \cdots & ETC_{mn} \end{pmatrix} \tag{2}$$

Similarly, RCU_{ij} can be used to represent the executory costs needed theoretically for subtask i in No. j recourse, therefore, RCU matrix is as follows:

$$RCU = \begin{pmatrix} RCU_{11} & \dots & RCU_{1n} \\ \vdots & \ddots & \vdots \\ RCU_{m1} & \cdots & RCU_{mn} \end{pmatrix} \tag{3}$$

According to *ETC* matrix and RCU matrix, time and costs for accomplishing assigned tasks in each resource node can be calculated.

In this formula, $T(i, j)$ represents time needed by No. j subtask in subtask i, and $resource(i, j)$ represents costs needed by No. j resource in subtask i. Formula (4) and formula (5) respectively represent total time and costs for finishing the task. Ultimate aim of cloud computing resource scheduling is to obtain maximum execution time and minimum total costs.

$$sumTime(i) = \max \sum_{i=1}^{resouce} \sum_{j=1}^{n} T(i, j) \times ETC(i, j) \tag{4}$$

$$sumCost(i, j) = \sum_{i=1}^{resource} \sum_{j=1}^{n} resource(i, j) \times RCU(i, j) \tag{5}$$

3. Research on Resource of Modified Shuffled Frog Leaping Algorithm in Cloud Computing

3.1 Modifications of Shuffled Frog Leaping Algorithm

Modifications of SFLA are in three aspects. The first one is modification of initialization aimed at frog leaping group, because the former random method of initialization had bad influences on optimizing effects and quality of ultimate optimal solution; the second aspect is update specific to local search, due to integer coding adopted in resource scheduling of cloud computing, modification to update operation of shuffled frog leaping algorithm is needed so as to better adapting to generation of locally optimal solution; the other is applying evolutionary strategy into the optimal solution in combination of global search and local search, so as to find out globally optimal solution.

(1) Backward Learning Strategy

Based on reference [10], this article adopts differential evolution algorithm based on backward learning to initialize population of frog population algorithm, thus guaranteeing diversity o population as mush as possible so as to accelerate rate of convergence of algorithm. Firstly, N initialized solution are generated randomly, and corresponding reverse solutions are made to each initialized solution according to formula (6):

$$op_{ij} = rand \times (\min_j + \max_j) - x_{ij} \tag{6}$$

Wherein, x_{ij} represents No. i initial solution in j-dimensional space, op_{ij} represents No. i reverse solution in j-dimensional space, $\min_j$ and $\max_j$ respectively represent the minimum value and maximum value in j-dimensional space. $rand$ represents random number whose value is between 0 to 1.

Composing initial solution and solution of backward learning to an aggregation from which selecting a better solution as the initial population through sequencing.

(2) Local Search Update

Local search update is based on the situation that all resource scheduling strategies in cloud computing are vectors with encoding length of d, and in combination of update mode to redefine SFLA through interlace operation in genetic algorithm. Firstly, generating a vector with encoding length of d randomly which is called crossover factor, and setting its value as 0 or 1. Taking this crossover factor as judgment, when value of corresponding crossover factor is 1, position of the worst frog shall be replaced by position of optimal frog in the sub-group, when value of crossover is 0, the position will not be changed.

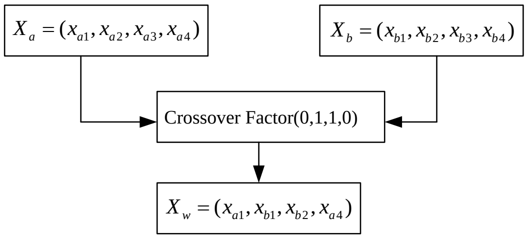

Figure 1. Schematic Diagram of Crossover Factor

In Figure 1, X_a represents individual sequence of the worst frog, and if X_w □ X_a, optimal frog in sub-group will substitute X_b to execute evolutionary operation, otherwise, X_a will not be changed.

(3) Evolutionary Strategy

In SFLA, the worst frog can only learn from the optimal frog in the same sub-group, and only when individual fitness value of the optimal frog is less than individual fitness value of other updated frogs in the sub-group, can the optimal frog be updated. This leads to slow rate of convergence, and only local optimum is found while it is hard to find globally optimal solution. Therefore, setting up matrix of information of optimal frogs in all sub-groups through introducing evolutionary strategy can help exchanges between update information of the worst frogs and information of optimal frogs of all sub-groups, thus keeping update and evolution of optimal frogs in the sub-group.

Given that in the No. n iteration, aggregation of optimal frogs in No. m is $X_m(t) = \{x_{m1}, x_{m2}, \cdots x_{mj}\}$, $m \in [1, W]$, and value of each dimension is $[1, M]$, so after analyzing statistics of times of optimal frogs in all sub-groups appearing in each dimensional space, a matrix is listed as follows:

$$P(n) = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1k} \\ P_{21} & P_{22} & \cdots & P_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ P_{m1} & P_{m2} & \cdots & P_{mk} \end{bmatrix} \tag{7}$$

Wherein, P_{ij} represents value of probability in i-dimensional space when value of No. t iteration is k. During the process of iteration, update of probability matrix is based on variable marginal distribution model.

$$\begin{cases} P(n) = dP(n-1) + (1-d)Q \\ d = e^{\frac{t_{\max} - t}{t_{\max}}} \end{cases}$$

(8)

In this formula, $d \in [0,1]$ represents increment factor, $P(n-1)$ represents probability set of last iteration, and Q represents new probability set in current population. Global and local search capabilities of frog population are balanced through increment factor, and when increment factor is small, global search capability mainly relied on the last search results. Along with gradually increase of increment factor, local search capability is strengthened, thus new global search capability is integrated with search results of last time and local search results of this time, which makes this algorithm possess a sound global search capability. $t_{\max}$ represents times of local search in sub-group, this can ensure good search capability when increment factor is small and improve search speed. Besides, when increment factor gradually increases, local convergence precision can be improved.

3.2 Steps of Algorithm

According to the foregoing descriptions, steps of algorithm in this article are described as follows:

Step 1: Initializing scale of population, and setting number of iterations.

Step 2: calculating individual fitness function value of each frog, composing initial solution and backward learning solution into an aggregation according to formula (6), and then selecting optimal solution as the initial population based on sequencing from big to small.

Step 3: carrying out local search on each frog sub-group, selecting the optimal individual frogs to form a frog group, and building probability matrix according to formula (7).

Step 4: selecting part of new frogs from the matrix according to sampling prescription, and putting these frogs in each sub-group, if individual fitness of a new frog is better than what of the optimal frog in the sub-group, the former optimal frog will be replaced by the new frog.

Step 5: Updating the worst frog in the sub-group according to local update strategy.

Step 6: Selecting the optimal frogs in all sub-groups to update probability matrix according to formula (8).

Step 7: Judging whether number of times of local search are finished, if they are, then exiting local search, otherwise, going back to step 3.

Step 8: judging whether number of times of iterations are reached, if they are, then stopping, and output optimal frog individual is the corresponding optimal decision of cloud computing.

4. Simulation Experiment

4.1 Simulation Environment

Simulation experiment is carried out with Core 3.0GHz CPU, 4GBDDR3 memory bank, 500G hard disk and Windows 7 operating system on Cloudsim Simulation platform. Frog population scale is set as 500, number of sub-group is 20, number of frogs in sub-group is 25, and under this environment, local search is 10, $rand$ is 0.5 and d is 0.8. In this article, comparisons on time and costs are made based on other modified intelligence algorithm.

3.2 Comparison with Other Intelligence Algorithm in Cloud Computing

In this article, references [11], [12] and [13] are compared, because these three algorithms are new comparatively and possess certain reference value.

(1) Comparisons of completion time and costs under different quantities of tasks

In this article, resource quantity is set as 50 and task quantity is 10000. Figure 2 and Figure 3 show that along with increase of tasks, total completion time and costs are continuously increasing. Values under algorithm in this article are less than values under algorithms in references [11] and [12], and algorithm in this article can still maintain well resource scheduling performance under the situation of continuously increasing tasks.

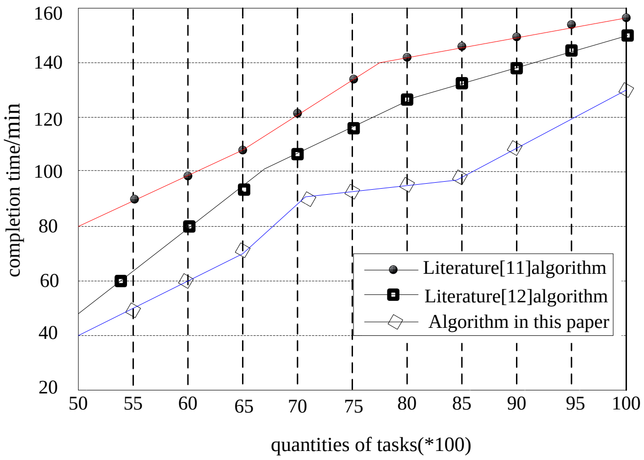

Figure 2. Comparison of Total Completion Time under different quantities of tasks

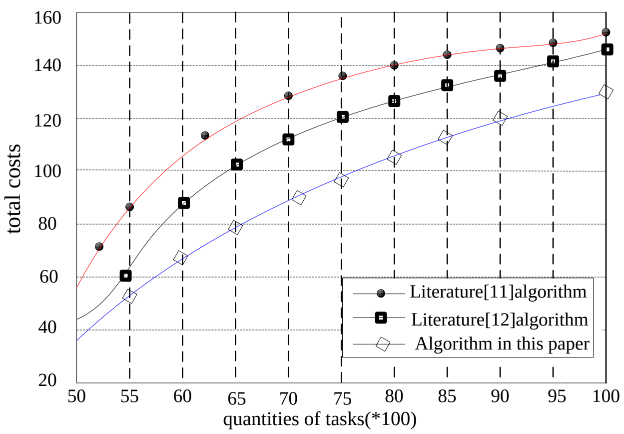

Figure 3. Comparison of Total Costs under Different Quantities of Tasks

(2) Comparisons of completion time and costs under different quantities of resources

In this article, resource number is set as 50 tasks and resource quantity is set as 100. Figure 4 and Figure 5 show that along with increase of resources, total completion time and costs are continuously decreasing. Values under algorithm in this article are less than values under algorithms in references [11] and [12], indicating algorithm in this article still maintains a well performance,

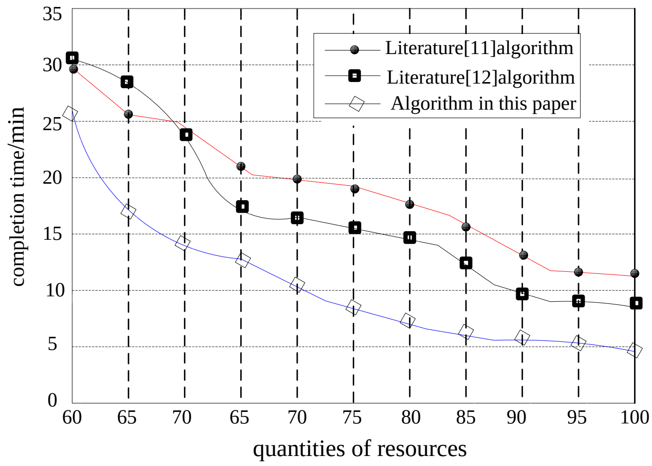

Figure 4 Comparison of Total Completion Time under Different Quantities of Resources

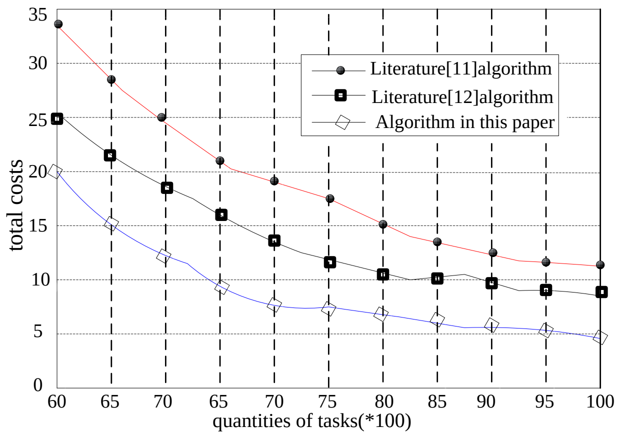

Figure 5. Comparison of Total Costs under Different Quantities of Resources

5. Conclusion

It is a focus topic of current research that how to use resources under cloud computing. This article introduces SFLA into cloud computing resource scheduling to solve problems of slow algorithm solution speed and tendency of falling into local optimum, adopts backward learning algorithm to initialize algorithm, introduces crossover factor into local search update, and adopts evolutionary strategy in selecting optimal solution in combination of global search and local search, thus improving algorithm performances through modifications of the above mentioned three aspects. Simulation platform testifies that algorithm in this article can effectively solve resource distribution problems, reduce total time, total costs and energy consumptions, and improve efficiency of algorithm resource allocation.

References

[1] Lin Wei-wei,Qi De-yu, “Survey of Resource Scheduling in Cloud Computing[J]”,Computer Science, vol. 39, no. 10, **(2012)**, pp. 1-5.

[2] Y zhao-feng,W qi-ming, L hai-lian, “Research on Resource Scheduling Algorithm of Cloud Computing Based on Improved DAG Diagram and Task Delay [J]”, Computer Measurement & Control , vol. 22, no. 2, **(2014)**, pp. 499-502.

[3] Z wei-ming,Y qiong, “Consumer satisfaction genetic algorithm in cloud computing [J]”, Application

Research of Computers , vol. 31, no. 1, **(2014)**, pp. 85-88.
[4] L wei-lin,L Hong-bin,L bo, "Cloud computing resource scheduling based on improved quantum genetic algorithm [J]",Journal of Computer Applications , vol. 33, no. 8, **(2013)**, pp. 2151-2153.
[5] G li-zhen, "Multi-objective optimization based on particle swarm optimization algorithm in cloud computing[J]", Computer Engineering and Design, vol. 34, no. 7, **(2013),** pp. 2358-2362.
[6] H hua, "The Study of the Allocation of Resources on an Improved Artificial BeeColony Algorithm in Cloud Computing [J]", Bulletin of Science and Technology, vol. 29, no. 5, **(2013),** pp. 142-146.
[7] Z zong-bin, D zhong-jun, "Improved GA-based task scheduling algorithm in cloud computing [J]", Computer Engineering and Applications, vol. 49, no. 5, **(2013)**, pp. 77-80.
[8] D yan-yan, "Creditable resource scheduling based on particle swarm algorithm in cloudcomputing [J]", Computer Engineering and Applications, vol. 49, no. 18, **(2013),** pp. 78-81.
[9] M M Eusaff, K E Lansey, "Water distribution network design using the shuffled frog leaping algorithm[C]", //Proc of Word Water Congree.**(2004)**, pp. 111-115.
[10] W Hui,W Zhi-jian, S Rahnamayan, "Enhan-ced opposition based differential evolution for solving high-dimensional continuous optimizati on problems[J]", Soft Computing, vol. 15, no. 11, **(2011)**, pp. 2127-2140.
[11] Z hui,G pin,Li qi, "On a Load Banlancing Algorithm Based on Bandwidth Constraintsin Cloud Computing [J]", Journal of Southwest China Normal University(Natural Science Edition) , vol. 39, no.. 7, **(2014)**, pp. 121-127.
[12] C song,Z jian-ming,C ji-ming,"Load balancing technology based on naive Bayes algorithm in cloud com puting environment[J]", Journal of Computer Applications, vol. 34, no. 2, **(2014),** pp. 360-364
[13] X yan-wen,Wjin-bao,L ya-ping.Energy Efficient Data Placement Algorithm and Node-Scheduling Polici es in Cloud Computing Systems[J],Journal of Computer Research and Development,**(2013)**,S1:342-351

Fund Project:

This work was supported by the project of the science and technology research of Educational Department of the Hubei provice(D20152601)

Authors

Bin Ning (1977.07-),Male, Master, Associate Professor, Research Direction:Software engineering, Algorithm design.

Qiong Gu (1973.03-),Female, Doctor,Associate Professor,Research Direction:Software engineerin g, Algorithm design.

Yi Wang (1980.03-),Male, Doctor,Lecturer,Research Direction:Software engineering.

