

Research on Cloud Computing Resource Scheduling Based on PSO-MC Algorithm

Xu Zhe-jun

(ZheJiang Technical College of posts & telecom, ShaoXing, Zhejiang, 312000)
zhejiangxzj@126.com

Abstract

The reasonable resource scheduling under the cloud computing environment has become the current research focus. Based on the disadvantages of the traditional particle swarm optimization, this paper introduces a membrane computing theory, and proposes a particle swarm optimization based on the membrane computing (PSO-MC). The paper first analyzes the problems of cloud resource scheduling, takes the membrane system as a framework, conducts the particle's local optimization for intramembrane and the particle swarm's global search and optimization for intermembrane, finds the optimal solution of cloud resource scheduling, and finally performs the algorithm simulation experiment via the CloudSim platform. According to the simulation experiment, the PSO-MC algorithm could reduce the average time to complete the task, improve the efficiency of task processing, and enable the cloud computing resource scheduling to be more reasonable.

Keywords: *Cloud computing, Membrane computing, resource scheduling, swarm optimization*

1. Introduction

Cloud computing is a kind of service mode based on the Internet and developed in recent years, which combines network technology with traditional computers like the distributed computing, parallel computing and grid computing, etc. It assigns access to computing resources of shared configurations like network, the server and storage, etc. and provides quick access platform between cloud users and providers. Currently, cloud computing services are divided into three layers, and they are infrastructure service layer, platform service layer and software service layer. Currently, the most used in cloud computing is the Map/Reduce framework, the main idea of which is that users put forward inquiries and other requirements to the cloud computing providers, and the cloud computing server analyzes and searches the requirements for resources through a high computer processing procedure, and finally returns the results to the users. As there are more and more Internet users and their requirements increase, business of high real-time demand have appeared, such as e-commerce, data mining and data storage, etc. It is especially true in cloud computing, because there are a large number of cloud users and the system needs to handle an increasing number of tasks, the scale of which is becoming larger. Therefore, it has become the key point in the research of reasonable allocating resources as how the system can quickly deal with requirements of quickly processing service requirements.

Cloud computing architecture is a focus of current research [1-2]. Resource allocation of cloud computing under different architectures is studied in literature [3-4] and a kind of resource allocation mechanism of self-protection is proposed and elaborated in literature [5]. Resource allocation system suitable for distributed computing is proposed in literature [6] and it effectively improves the service quality of cloud computing system. Currently, it is the main direction of research to introduce intelligent algorithm to resource allocation of cloud computing. The resource allocation of cloud computing is an NP problem, and

genetic algorithm, ant colony algorithm and particle swarm optimization are all for solving the NP problem. Algorithm based on particle swarm optimization is proposed in literature [7], and it has optimized the processing time. However, the particle swarm algorithm converges quickly at the beginning and later the search capabilities of some parts are inefficient and slow. In literature [8], resource allocation of cloud computing by using ant swarm algorithm is proposed to minimize the completion time of the total task and average completion time of every single task. However, at the beginning, the ant colony algorithm has scarce pheromone, and converges slowly. Gheorghe proposes the concept of membrane computing [9], and intra-membrane structure and inter-membrane communication are adopted in membrane computing so that membrane calculation can be easily used in parallel, distributed or grid computing system. Because it is easy to implement parallel computing in membrane computing, traditional intelligent algorithm and membrane calculation are combined and traditional intelligent algorithm is adopted in some local parts so as to gradually search the optimal solution in inter-membrane communication.

2. Cloud Computing Resources

2.1. Present Situation of Cloud Computing Resources

In cloud computing architecture, faced with thousands of access requests from cloud users, the cloud computing server needs to fully consider the amount of cloud resources each cloud user could use, the completion time and light. At present, most cloud computing resource allocation algorithms are mainly focused on cloud users' total completion time of tasks, so this algorithm can not involve all the cloud customers and it may lose cloud customers with the least completion time.

2.2. Description of Cloud Computing Resource Allocation

In order to better illustrate the superiority of the algorithm in the aspect of resource allocation, here only the circumstances of independent sub-tasks are considered and make the following description:

- (1) Operation time of these sub-tasks is almost the same.
- (2) All sub-tasks need fewer resources than the system.
- (3) Define resource-using time of each sub-task.

Table 1. Sub-tasks and Resource List

Sub-task	Resource Amount	Operation Time
N_1	M_1	$T(N_1, M_1)$
N_2	M_2	$T(N_2, M_2)$
.....
N_n	M_n	$T(N_n, M_n)$

Herein, $T(N_i, M_j)$ refers to the operation time of sub-task i on resource j , and the allocation is as follows:

$$T = \begin{bmatrix} T_{11}, T_{12}, \dots, T_{1n} \\ T_{21}, T_{22}, \dots, T_{2n} \\ \dots \dots \dots \\ T_{m1}, T_{m2}, \dots, T_{mn} \end{bmatrix} \quad (1)$$

It should be noted that $Begin_i$ is set as the start time of task N_i , T_{ij} is the expected completion time of task and the completion time of total task is $\max\{Begin_i + T_{ij}\}$.

3. Description of Basic Cloud Computing Resource Algorithm

3.1. Description of Membrane Computing

The concept of membrane computing comes from tissues of biological cells, and it is a computing model abstracted from the mechanism of chemical treatment of cells. From the perspective of micro cells, the cell nucleus and cyst are abstracted as a cell membrane; and from the perspective of macro cells, the whole cell or even the biological community is a cell membrane. The membrane structure is shown as Figure 1.

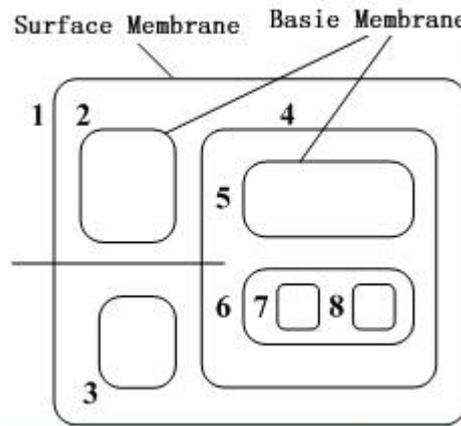


Figure 1. Membrane Structure Diagram

A membrane model is composed of the following three elements:

$$M = (C, V, I) \quad (2)$$

In the formula (2), V refers to a non-empty abstract set, consisting of characters in a character set. C refers to the structure, consisting of system structure, the objects and rules. I refer to the external environment. Membrane computing is described and analyzed in literature [9-10], and membrane algorithm has the following advantages:

- (1) Each section within the membrane runs separately and inter-membrane exchange only occurs in the adjacent sections. Membrane computing is segmented and it is more suitable for parallel, distributed computing systems and also cloud computing model.
- (2) A variety of intelligent optimization algorithms are used in membrane computing, such as the use of particle swarm optimization.
- (3) Local optimum can be avoided in membrane computing and the overall accuracy of optimal solution can be improved.

3.2. Description of Particle Swarm Algorithm

The basic idea [11] of elementary particle swarm optimization (PSO) is that an individual in the space is likened to a particle without quality or volume. Set the i particle in the group as $x_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$ and the position it has experienced is $p_i = \{p_{i1}, p_{i2}, \dots, p_{im}\}$, among which, the best location is p_{best} . The best location of all the particles in the group currently is p_{gbest} and velocity of the particle i is set as $v_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}$. Motion equation of particle is:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand() \times (p_{best} - x_{id}(t)) + c_2 \times rand() \times (p_{gbest} - x_{id}(t)) \quad (3)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (4)$$

Herein, w refers to inertia weight, c_1 and c_2 are constants and $rand()$ is random number, the value of which is between $(0, 1)$.

4. PSO-MC Algorithm Design

In terms of reasonable optimization of resources in cloud computing, make sure that reasonable resources can be obtained for sub-tasks while keep the completion time of total task minimum. Optimize it by adding membrane computing into particle swarm algorithm and each particle in the particle swarm algorithm represents an executable task scheduling scheme according to the characteristics of resource allocation in cloud computing. In the algorithm of this paper, divide the space of particles and each divided space is as a basic membrane, and local search strategy is used in the basic membrane to enhance particle's local search ability and convergence speed. In the basic membrane, send the optimal solution of local particle swarm to the surface membrane and find the global optimal solution by adopting the method of global search.

4.1. Description of Spatial Segmentation Algorithm

Feasible solution space as described in literature [12] is adopted. Set $S = (s_1, s_2, \dots, s_n)$ and $W = (w_1, w_2, \dots, w_n)$, then (S, W) is a feasible solution. In particle swarm algorithm, choose the h dimension in the feasible solution space, then $x_i \in [S_h, W_h]$, divide into N sub-spaces on the h dimension in the feasible solution space (S, W) .

$$S_i(i) = S_i + (i - 1) \left(\frac{W_h - S_h}{N} \right) \quad (5)$$

$$W_i(i) = W_i + (N - i) \left(\frac{W_h - S_h}{N} \right) \quad (6)$$

4.2. Description of Local Search Algorithm

In the basic membrane of membrane computing, differential evolution algorithm is used for local search. And in the process of searching elements, self-adaptive differential mutation operators and selected operators are used to get the optimal particle in the particle swarm. The algorithm is described as follows:

m refers to the amount of particle swarms, t refers to the number of evolution, $\max(t)$ refers to the maximum number of evolution and $rand()$ refers to random function, the value of which is 0 and 1. The value of u is 1.2 and δ is 1.6.

Begin

for $i = 1$ to $size$ //set the size number of particle swarm as size

$$\text{Select individual species } x_k, x_j, x_z, r = \left[\begin{array}{c} \left(\frac{t - u}{\max(t)} \right)^2 \\ rand(1, N) * e \\ \delta^2 \end{array} \right]$$

Introduce formula (6) and get the following formula

$$x_i(t+1) = \begin{cases} x_k(t) + r \times (x_j(t) - x_z(t)) \\ x_{best}(t) + r \times (x_k(t) - x_z(t)) \\ x_i(t) + r \times (x_{best}(t) - x_z(t)) \end{cases} \quad (7)$$

End

4.3. Optimization of Membrane Computing in Particle Swarm Algorithm

In the optimization of particle swarm algorithm, through the optimization of membrane computing, the optimal solution of each basic membrane sector is sent to the surface membrane and it is searched through the global optimal solution of particle swarm. In the process of optimization, the most key factors are that whether the particle itself is a optimal solution and whether the optimal solution has been found in the species. In the particle swarm optimization, renewals are made mainly for the particle's speed and location, combined with local search algorithm. The settings are as follows: w refers to inertia weight, t refers to number of evolution, $\max(t)$ refers to the maximum number of evolution, c_1 and c_2 are constant, $\text{rand}()$ is random number and $\eta(t)$ is disturbance factor.

$$w = 0.5 + \frac{t}{\max(t)} + \eta(t) \quad (8)$$

$$c_1 = e^{\left(1 - \frac{t}{\max(t)}\right)} \quad (9)$$

$$c_2 = e^{\frac{t}{\max(t)}} \quad (10)$$

Put formula (8), (9) and (10) into formula (3), and then get formula (11)

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times \text{rand}() \times (p_{best} - x_{id}(t)) + c_2 \times \text{rand}() \times (p_{gbest} - x_{id}(t)) \quad (11)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (12)$$

In formula (9) and (10), c_1 changes according to c_2 , and in the process of evolution, c_1 becomes smaller gradually and c_2 becomes larger gradually. At the early stage of particle swarm, it depends mainly on the location of the particle swarm itself and at the later phase, it gradually turns to the global optimal position and improves the performance and speed through the disturbance factor. In the iteration process, when the particle achieves a position better than the historical position, inertia weight value is improved through disturbance factor $\eta(t)$, or it will be reduced.

Algorithm steps are as follows:

Step1: Initialize the limit value. Set evolutionary generation numerical.

Step2: Divide space. Separate the space of particle swarm into disjoint subspaces and each subspace is the basic region in the membrane system.

Step3: Particle swarm initialization. Initialize sub particle swarm everywhere in the basic membrane as in Step1.

Step4: Local search within membrane. In the basic membrane, use a differential algorithm for local search, and when the local search reaches the optimal solution, send it to the surface membrane for global search.

Step5: Global search within the membrane. In the particle swarm algorithm, formula (9) is adopted for global optimal search.

Step6: Meet the conditions and terminate the judgment. The algorithm ends when it has reached the evolutionary generation numerical or a satisfactory result. Otherwise, turn to Step3.

5. Emulation Experiment

5.1. Algorithm Performance Test

Use the three benchmark functions in literature [13] for comparison test so as to testify the validity of algorithm and analyze the performance of the algorithm.

$$f_1(x) = \sum_{i=1}^N (-x_i \sin(\sqrt{|x_i|})), -500 \leq x_i \leq 500 \quad (13)$$

This function is a continuous and convex peak function with the global minimum point as 0, that is $x_i = 0 (i = 1, 2, \dots, n)$, and there is no interaction between variables.

$$f_2(x) = -20 \exp\left[-0.2 \sqrt{\frac{\sum_{i=1}^N x_i^2}{N}}\right] - \exp\left[\frac{\sum_{i=1}^N \cos(2\pi x_i)}{N}\right] + 20 + \exp(1) - 32.768 \leq x \leq 32.768 \quad (14)$$

The function is a multimodal function with the global minimum point as 0, that is $x_i = 0 (i = 1, 2, \dots, n)$ and there is no interaction between variables.

$$f_3(x) = \sum_{i=1}^{N-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] - 5 \leq x \leq 5 \quad (15)$$

The function is a multimodal function with the global minimum point as 0, that is $x_i = 0 (i = 1, 2, \dots, n)$ and there is no interaction between variables.

Initially set the membrane system to be made up of 9 basic membranes and 1 surface membrane. Within the membrane, the value of initial particle swarm is 50, and the parameter of particle swarm algorithm in the surface membrane is $\eta(t) = 0.9$, r is 50 and $\max(t)$ is 100. In this paper, the three functions are tested for 30 times respectively so as to get the best result, the worst result and the average value. Through the comparison with basic particle swarm algorithm, the results are shown as Table 2, and Table 2-4 are the convergence curve comparison of algorithm in this paper with basic particles algorithm in the three functions.

Table 2. Comparison of Various Test Functions

Function	Algorithm	Optimal Results	Worst Results	Average Value
$f_1(x)$	Basic Particle Swarm Algorithm	0.014858	0.0198452	0.0275211
	Algorithm in this paper	0.014321	0.0125425	0.0225841
$f_2(x)$	Basic Particle Swarm Algorithm	3.012547	2.9562147	2.9925631
	Algorithm in this paper	2.941257	2.6584121	2.9841253
$f_3(x)$	Basic Particle Swarm Algorithm	3.256847	2.5681201	3.4285296
	Algorithm in this paper	3.1254413	2.4125631	3.3052176

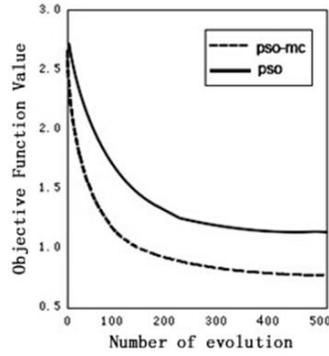


Figure 2. Convergence Curve of Function

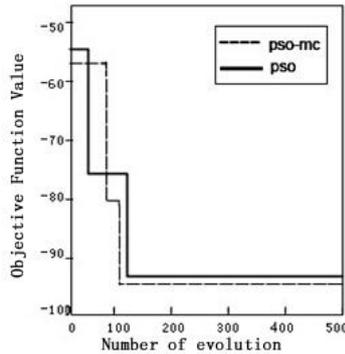


Figure 3. Convergence Curve of Function

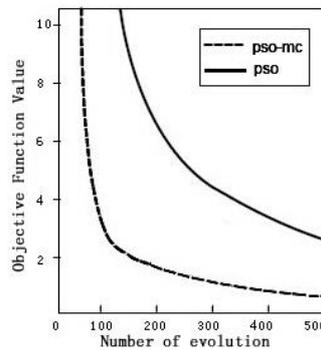


Figure 3. Convergence Curve of Function

As can be seen from Table2 no matter in terms of the optimal results, worst results or the average value, the optimized particle swarm algorithm is significantly superior to traditional particle swarm algorithm. It can be seen from Fig.2-4; the optimized particle swarm algorithm has relatively better stability and can reach a position near the global optimal position.

5.2. Resource Allocation of Cloud Computing

In this paper, Cloudsim platform is adopted to simulate the cloud computing environment and set the virtual resources for 10 and 500 sub-tasks. Set the processing capacity of virtual resources on the platform as {100MIPS, 200MIPS.....600MIPS} , and the length of sub-task is [100MIPS, 400MIPS]. Set the group size of particle swarm as size=100, the number of evolution is 50, the largest number of evolution is 500 and the number of iterations is 700. The results are shown in Fig.5.

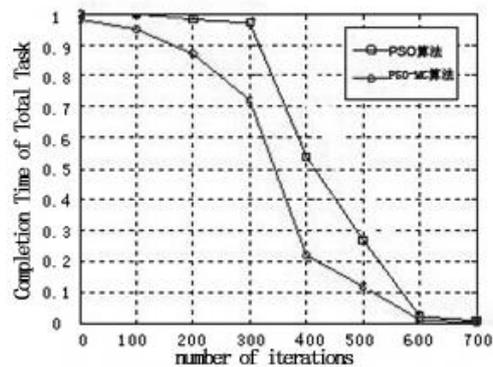


Figure 5. Completion Time of Total Task of Two Algorithms

As is shown in Fig.5, the optimization effect of the PSO-MC algorithm in this paper is significantly superior to PSO algorithm in terms of resource allocation, and the completion time of total task is the minimum, achieving relatively good effects on time and performance.

6. Conclusions

In this paper, aiming at unreasonable allocation of resources in cloud computing, the author puts forward the concept of adding membrane computing to particle swarm algorithm. Simulation experiments have shown that PSO-MC is a relatively better resource allocation algorithm with a good capacity of optimizing resources and allocating resources reasonably. The resource allocation of cloud computing is rather complex, for it includes problems like calculating costs, and load costs. Therefore, we can carry out further research so as to optimize the performance of resources in a cloud computing environment.

References

- [1] I. Foster, Y. Zhao, I. Raicu, *et al.*, "Cloud computing and grid computing 360-degree compared", Proceeding of the 2008 Grid Computing Environments Workshop, Washington DC, IEEE Computer Society, (2008), pp. 1-10.
- [2] M. Armbrust, A. Fox, R. Griffith, *et al.*, "Above the clouds, A Berkeley view of cloud computing", (2010) January 25.
- [3] X. Hong-Wei, J. Tong, "Dynamic Load Balancing Model Based on NET Remoting", Computing Engineer, vol. 35, no. 21, (2009), pp. 2, 64, and 266.
- [4] R. S. C. Ho, W. C. -Li, F. C. Lau, "Lightweight process migration and memory prefetching in openMosix", Proceedings of 2008 IEEE International Symposium on Parallel and Distributed Processing, Piscataway, NJ:IEEE Press, (2008), pp. 1-12.
- [5] M. Randles, A. Taleb-bendlab, D. Lamb, "Scalable self-governance using service communities as ambient", Proceedings of SERVICES'09 Proceedings of the 2009 Congress on Services-I, Washington, DC IEEE Computer Society, (2009), pp. 813-820.
- [6] O. Abu-Rahmeh, P. Johnson, A. Taleb-Bendiab, "A dynamic biased random sampling scheme for scalable and reliable grid networks", Journal of Computer Science, vol. 7, no. 4, (2008), pp. 1-10.
- [7] G. Lizheng, Z. Shuguang, S. Shigen, *et al.*, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm", Journal of Networks, vol. 7, no. 3, (2012), pp. 547-553.
- [8] J. Li, P. Jian, C. Xiaoyang, *et al.*, "A Task Scheduling Algorithm Based on Improved Ant Colony Optimization in Cloud Computing Environment", Energy Procedia, no. 13, (2011), pp. 6833-6840.
- [9] Z. Ge-Xiang, "A Survey of Membrane Computing as a new branch of Natural Computing", Chinese Journal of Computing, vol. 33, no. 2, (2010), pp. 208-214.
- [10] Huangliang, "Research on the calculation method for the optimization of membrane", (2007), Hangzhou, Zhejiang university.
- [11] Z. Wen-ai, L. Li-Fang and L. Xiao-Rong, "Particle Swam optimization based on Particle evolution", Computer Engineer and Application, vol. 44, no. 7, (2008), pp. 51-53.
- [12] J. Zhongyang, C. Zixing and W. Yong, "Hybrid self-adaptive orthogonal genetic algorithm for solving global optimization problems", Institute of Software, vol. 21, no. 6, (2010), pp. 1296-1307.

- [13] K. Deep and J. C. Bansal, "Mean particle swarm optimization for function optimization", Computational Intelligence Studies, vol. 1, no. 1, (2009), pp. 72-91.
- [14] W. Jiyi, L. Wenjuan, J. P. Huang, J. L. Zhan and, D. R. Chen, "Key techniques for Mobile Internet", a survey, SCIENTIA SINICA Information, vol. 45, no. 1, (2015), pp. 45-69.
- [15] J. Zhang and J. Y. Wu, "Provable Secure Efficient Arbitrated Quantum Signature Scheme", Journal of Beijing University of Posts and Telecommunications, vol. 36, no. 2, (2013), pp. 113-116.
Source of project: 2012 Scientific Research Project of Education Department of Zhejiang Province (Y201225992)

Author

Xu Zhe-jun, (1980), male, lecturer, master, research direction: data mining and cloud computing

