PROCEEDINGS OF SCIENCE

Research on Improved Particle Swarm Optimization based on Membrane System in Cloud Resource Scheduling

Jia Chen[a1]; Wenjie Gong[b]; Tingyu Liang[c]
University of Electronic Science and Technology of China
Chendu, 610054,China
Email: [a]`jchen@uestc.edu.cn`;[b]`gongwenjie1205@163.com`;[c] `tingyliang@163.com`

Jiali Xu
Chengdu university
ChengDu, 610106, China
E-mail: `lotussunnyx@163.com`

Shenpeng Wei
Shandong University of Science and Technology
QingDao, 266000, China
Email:`710485523@qq.com`

In this paper, an improved particle swarm optimization algorithm based on membrane system about resource scheduling in cloud computing is proposed. The resource scheduling in cloud computing mentioned combines improved particle swarm optimization algorithm and membrane system. The resource space of cloud is simulated as the whole membrane system and is treated as the entire optimization particle swarm search range. The membrane system is divided into the main membranes and the auxiliary membranes. The system optimizes the particles in the auxiliary membranes at the same time, and transfers the particles of high quality to the main membranes. The main membranes removes the inferior particles to achieve the optimal allocation of resources in cloud computing. It illustrates that the algorithm is more efficient in resource scheduling of large-scale tasks in cloud computing compared with other algorithms in the CloudSim platform.

ISCC2017
16-17 December 2017
Guangzhou, China

[1] The research work was supported by the Sichuan Science and Technology Project under Grant No. 2017GZ0318 and the Fundamental Research Funds for the Central Universities under Grant No. ZYGX2015J068.

1.Introduction

At present, domestic and foreign scholars have provided a large number of feasible algorithms for cloud computing resource scheduling. Traditional algorithms, e.g. Min-Min and Min-Max Algorithm, Sufferage algorithm etc, are simple and easy to realisze but peform bad when processing complex troubles.[1,2] Heuristic algorithms have good performance of finding a optimal solution, but they are easy to be ended by an early convergence and get partial optimal solutions, e.g. artificial neural network algorithm, particle swarm algorithm, genetic algorithm, ant colony algorithm, simulated annealing algorithm.This paper aims to combine the dynamic nature of resource scheduling in cloud computing and the adaptability of particle swarm optimization algorithm based on membrane system.A new hybrid algorithm based on improved particle swarm optimization algorithm is proposed in this paper(M-OPSO).

2. Resource Scheduling Problem Description

2.1Resource Scheduling Model in Cloud Computing

The mathematical model of resource scheduling for cloud computing can be described by a 5 tuple: $S = \{ T, V, D, M_{tv,} M_{vd}\}$ (2.1)

In formula（2.1），the $V = \{ v_{1,} v_{2,} \ldots, v_m\}$ is collection of resources,$T = \{ t_{1,} t_{2,} \ldots, t_m\}$ is collection of M tasks; $D = \{ d_{1,} d_{2,} \ldots, d_m\}$ is collection of physical devices; M_{vd} is the correspondence between resources and physical devices; Mtv is the Mapping strategies between tasks and resources.

M_{tv} is assigned by the computing center according to the user's task, and M_{vd} needs to dispatch the resource to the corresponding physical device by scheduler.

Suppose a task t_i is mapped by M_{vd} to resource v_j, tasks allocated on v_j is scheduled to physical device on d_k to execute , according to the relationship between resources and tasks, task T_i after resource transfer after the arrival of the expected execution time to perform physical devices on the d_k is ETC($t_{i,} d_k$), then the distribution matrix called ETC matrix about T for D :

$$ETCmn= ETC(t_iM_{tv,} d_k)\ 1 \le i \le m，1 \le k \le n' \quad (2.2)$$

Formula (2.2) is a execution time matrix corresponding to the execution time matrix of the m task on the n′ physical device after the resource is mapped by $t_i M_{tv}$.

The earliest completion time of task t_i on the physical device d_k is:

$$Finish(t_iM_{tv}，d_k) = Start(d_k) + ETC(t_iM_{tv}，d_k) \quad (2.3)$$

In formula（2.3）, the Start(d_k) represents the earliest start time about physical device d_k ,so the total execution time of the assigned tasks on the physical device d_k is:

$$\mathrm{Sum}(d_k) = \sum_{i=1}^{m} cik\ Finish(t_i M_{tv}, d_k) \quad (2.4)$$

$$c_{ik} = \begin{cases} 1 & tiM_{tv} = d_k \\ 0 & others \end{cases} \tag{2.5}$$

In formula（2.5）,c_{ik}= 1 indicates that the task ti is finally mapped to the physical device d_k. the total time executes of all tasks T = { $t_1, t_2, \ldots, t_m$} is:

$$\text{total}(T) = \sum_{k=1}^{n} Sum(d_k) \tag{2.6}$$

The goal of cloud resource scheduling is to make formula (2.6) the minimum, and one of the objective functions of cloud resource scheduling is:

$$Goal(T) = \min \sum_{k=1}^{n} Sum(d_k) \tag{2.7}$$

This study also takes into account the unit execution cost of the default virtual machine,the task requires the exchange of virtual machines during data transfer and exchange, as follows:

$$\text{taskCost} = t_i/\text{VmMIP}*\text{VmCost} \tag{2.8}$$

$$\text{taskDataTransferCost} = t_i*\text{VmTran} \tag{2.9}$$

In formula（2.8）,taskCost represents the cost of the task, where VmMIP represents the virtual machine execution capability,VmCost represents the unit execution cost. In formula(2.9),taskDataTransferCost represents the cost of information exchange, where VmTran represents the virtual machine unit information exchange costs.

The aim is to make a balance between the minimum task execution cost and the minimum execution time.

3.Article Swarm Optimization Algorithm based on Membrane System on Cloud Resource Scheduling

3.1Introduction to Membrane Systems

Membrane computing is proposed by Paun in 1998. [3,4]The P system consists of three parts: the hierarchical structure of membrane, the multiple sets of objects and the rules of evolution. [5]It‘s formally structure is represented as:

$$\prod = \text{V}', T', \mu, w_1, w_2, \cdots, w_m, (R_1, \rho_1), \cdots, (R_7, \rho_7) \tag{3.1}$$

3.2 An Introduction to M-OPSO

Particle Swarm, also known as Particle Swarm Optimization (PSO).

Chaotic neighborhood particle swarm optimization initializes the velocity and position of particles by chaotic sequence .The Logistic map formula is:

Zn + 1 = Zn (1-Zn), n = 0, 1, 2, ..., Zn + 1 (3.2)

The improved particle swarm algorithm flows steps as follows:

Step 1,Initialize the algorithm termination condition and the maximum number of iterati-

-on, the inertial weight of the particle swarm algorithm and the learning factor.

Step 2, Chaos initializes the velocity and position of the particle. Randomly generate a n-dimensional vector e.g. z1=(z11,z12,z13...z1D) in which each of its value is between 0 and 1 and D is the number of variables in the objective function.

Step 3, If the particle fitness is better than the individual extremum pBest, then pBest is set to the new position.If the particle fitness is better than the global extremum gBest, then pBest is set to the new position.

Step 4, According to the formula (3.3), (3.4), update the speed and location of the particle. The optimal neighbor particle selected for the caculation formula below:.

$$v_{id}^{t+1} = \omega_{id}^{t} v_{id}^{t} + c_1 r_1（p_{id}^{t} - x_{id}^{t}）+ c_2 r_2 (p_{gd}^{t} - x_{id}^{t}) + c_3 r_3（n_{id}^{t} – x_{id}^{t}） \quad (3.3)$$

$$X_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (3.4)$$

Step 5, Commit chaos optimization on the optimal position Pg = (Pg1, Pg2, Pg3,..., PgD). The mapping domain Pgi (i = 1,2,3, ... D) is mapped to the domain [0,1] of the Logistic equation zi = (Pgi-ai) / (bi-ai), (i = (F = 1, 2, ...), then the chaotic variable sequence is generated by inverse mapping Pgi (F) = ai + (Pg1 (F), Pg2 (F), ..., PgD (F)), (F = 1, 2, ...) .). Calculate its fitness value to obtain the optimal feasible solution P*and to replace any one particle position.

Step 6, If the stop condition is satisfied, the search is stopped and the global optimal position is output. Otherwise, it goes back to step 3.

4.Description of Membrane System Combined with Improved Particle Swarm Algorithm

4.1Improved Particle Swarm Optimization Algorithm based on Membrane System

Each particle position X_i = （X^t_{i1},X^t_{i2}…, X^t_{id}） will be treated as an object, and all the particles corresponding to each particle location X_i = （X^t_{i1},X^t_{i2}…, X^t_{id}） will be treated as an object, all the particles corresponding to the the set of solutions is seen as the object set of the membrane system. The multiple sets in each region can be expressed as: W_i = （$X_{1i}X_{2i}…X_{ni}$） , i = 1, 2, 3, 4, 5, ..., where n represents the number of particles in the film region i and X^t_{1i}， X^t_{2i}， …, X^t_{ni} denotes the solution corresponding to each particle in the region i, in which X^t_{ni} = (x^t_{ni1},x^t_{ni2},…, x^t_{nid}), and d is the dimension of the search space; initialize the relevant parameters. Membrane system structure in which master film and auxiliary film are defined,is shown in figure 1. Each film contains at least one particle structure abstractions. The structure of the initial membrane system is defined as follows: $[_1[_6]_6[_3[_4]_4[_5]_5]_3]_1$, wherein the structure contains m '= 5 layer film, including the surface film 1, the membrane 6, the auxiliary film 3, 4, 5. The film 1, namely the surface film does not carry out the specific fitness calculation, because it's only responsible for the recycle of the bad particles abandoned by the main film. The initial particle group is randomly allocated to the membrane system structure initialized in step 1 to ensure that each layer contains at least one particle, and the surface film 1 is empty. Details are as follows:

$\omega^0 = \lambda$; $\omega^1 = q_1q_2q_3 \ldots q_{n1}, n_1<n$; $\omega^2 = q_{n1+1}q_{n1+2} \ldots q_{n2}, n_1+n_2<n \ldots \ldots$

$\omega^m = q_{n（m-1）+1}q_{n(m-1)+2} \ldots q_{nm}, n_1+n_2+\ldots n_m<n$;

Where q_i (i = 1, 2, 3, ..., n) is the individual particles in the auxiliary film of each layer.

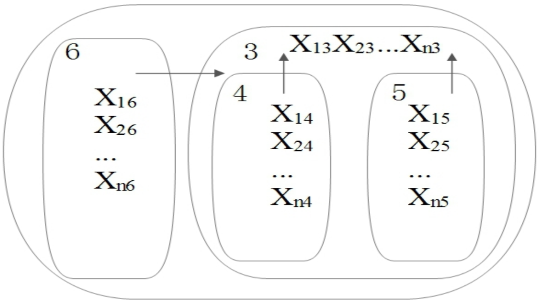

Figure 1: Membrane System Structure in Cloud Computing Resource Scheduling

4.2Detailed Steps of Algorithm

Step 1, The membrane system is initialized into the main film and the auxiliary film, and the velocity and the position of the particles are initialized by the chaos initialization respectively in the two layers. Each particle maintains its own neighbor particles and each layer contains at least one particle.

Step 2, Compute the fitness value of each particle, the determine the optimal value of the individual and the global optimal value; records the Pbest and Gbest.

Step 3, The velocity and position of the particle are updated according to the formula (3.3) and (3.4), Please refer to section 3.2 for details of the algorithm.

Step 4, The particles are sorted according to the fitness value according to the fitness value, and the selected dominant particles are fed into the main film.

Step 5,Judge whether the test reaches the optimal value or satisfies 200 iterations.

Step 6, Decoded the optimal solution; palce the virtual machine;map the corresponding physical machine. The resources are collected, the results are output, and the request ended.

5.Evaluation

5.1 Simulation Environment and Contrast Algorithm

Simulated by CloudSim, the particle swarm algorithm (PSO), chaotic particle swarm optimization algorithm (CLSPSO) and membrane system particle swarm optimization algorithm (M-OPSO) are compared. PSO population size is 30, ω is 0.95, c1, c2 is 0.90,r1, r2 is random number between (0,1]; CLSPSO as same as PSO, the chaotic search factor is 0.01, the chaotic search step is 15, M-OPSO population is set up as same as PSO with neighborhood factor considered, c3 is 0.8,r3 is random number between (0,1]. The membrane system is divided into three auxiliary Membrane, a main Membrane.The number of iteration of the three algorithms is 200.

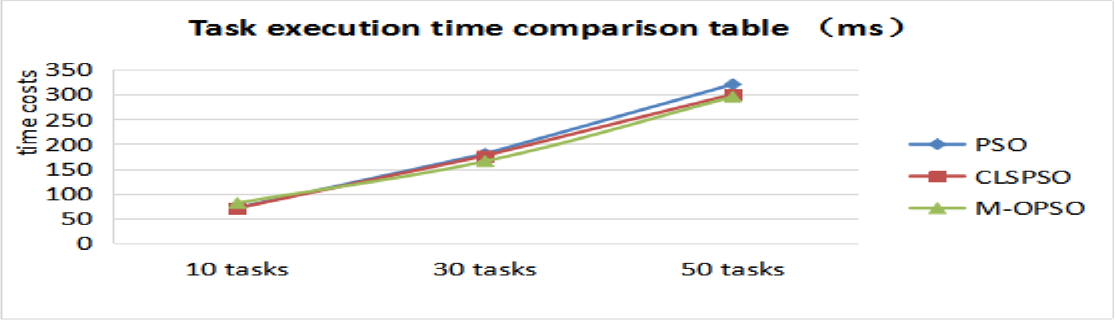

Figure 2: Task Execution Time Comparison Table

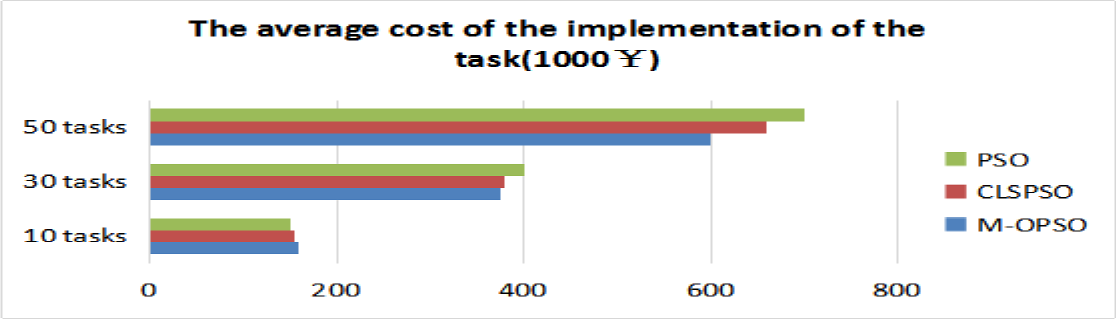

Figure 3: The Average Cost of the Excution of the Task

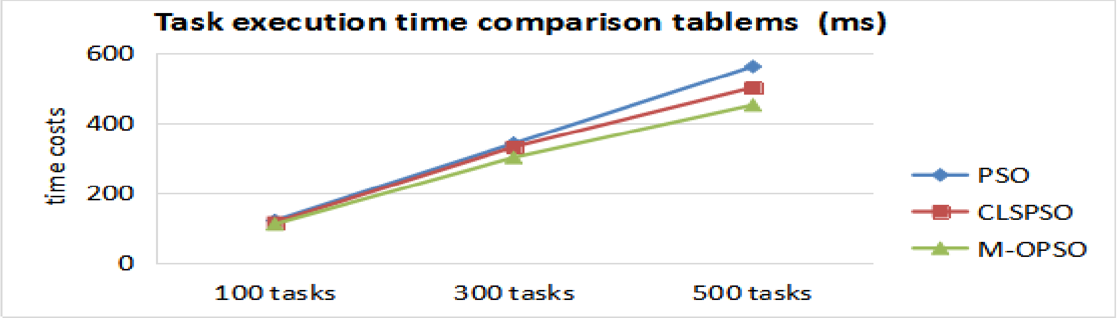

Figure 4: Task Execution Time Comparison Table

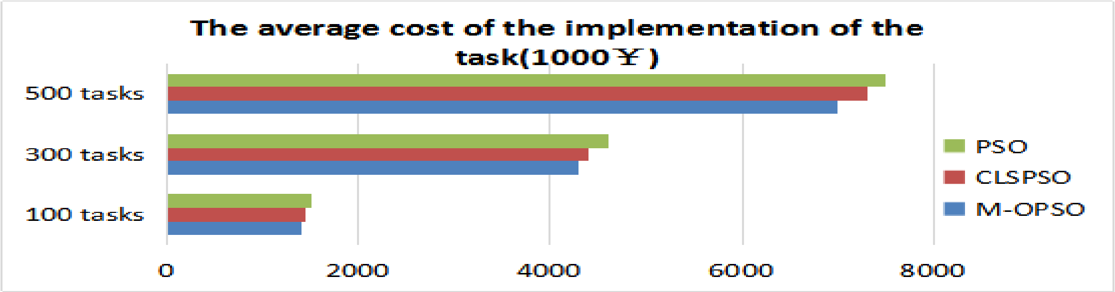

Figure 5 : The Average Cost of the Excution of the Task

1) 10 to 50 tasks are assigned to 4 cloud computing resource nodes with 8 virtual machines. As shown in figure 2, when process few tasks, PSO, CLSPSO, M-OPSO performance almost the same. The average execution of all tasks' total cost of PSO, CLSPSO, M-OPSO is shown in figure 3.

2) This test allocates 100 to 500 tasks to 30 cloud computing resource nodes, with 60 virtual machines, and the three algorithms perform 20 times of average execution time and average cost for all tasks. It's shown in Figure 4 that the task time increased as the number of tasks increases for all algorithm, but M-OPSO accomplishes the transitory task time, especially when the tasks exceed 100, so M-OPSO works better in large-scale resource scheduling. Figure 5 indicates that the average excution cost of all tasks of M-OPSO is less than CLSPSO and PSO.

It is illustrated by the experiments above that the M-OPSO algorithm utilizes the advantages of optimizing the particle swarm optimization algorithm and the membrane system to overcome their shortcomings and form a better combinatorial algorithm to find the optimal global cloud computing resource scheduling scheme.

6.Conclusion

Considering the problems of resource scheduling in cloud computing and the shortcomings of standard particle swarm algorithm, a cloud computing resource scheduling method based on particle swarm optimization (PSO) is proposed.It’s more suitable for cloud computing environment with large-scale task of resource scheduling and its execution cost is less than that of the algorithms campared in this paper.

References

[1] Etminani K,Naghibzadeh M. *A Min-Min Max-Min selective algorihtm for grid task scheduling*[C].Internet, 2007. ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia on. IEEE,2007:2-7.

[2] Munir E U,Li J,Shi S. *QoS sufferage heuristic for independent task scheduling in grid*[J].Information Technology Journal,2007,6(8):1166-1170.

[3] P un G, Rozenberg G, Salomaa A. *Handbook of Membrane Computing.* Oxford: Oxford University Press, 2009

[4] P un G. Computing with membranes. *Journal of Computer and System Sciences*, 2000, 61(1): 108-143.

[5] P un G, Suzuki Y, Tanaka H. *On the power of membrane division in P systems. Theoretical Computer Science,* 2004,324(1): 61-85.