CHEMICAL ENGINEERING TRANSACTIONS

VOL. 46, 2015

A publication of

AIDIC

The Italian Association
of Chemical Engineering
Online at www.aidic.it/cet

Guest Editors: Peiyu Ren, Yancang Li, Huiping Song
Copyright © 2015, AIDIC Servizi S.r.l.,
ISBN 978-88-95608-37-2; **ISSN** 2283-9216

DOI: 10.3303/CET1546066

Optimization of Resource Schedule Based on Improved Particle Swarm Algorithm in Cloud Computing Environment

Zhao Hongwei*[a], Shen Hongye[b]

[a]school of Information Engineering, Shenyang University, Shenyang, Liaoning, China
[b]Information Engineering School, Shenyang radio and television university, Shenyang, China
Zhw30@163.com

In order to optimize resources scheduling of cloud computing and to improve the utilization ratio of the resource as well as guarantee application quality of service, the dynamic resources scheduling algorithm basing on improved PSO has been designed and implemented after the study on the Resources Scheduling system of cloud computing . Firstly, the main idea of improved PSO is to extend Particle Swarm Algorithm to the interacting swarms model by cooperative Models. Secondly, a comprehensive model has been designed and implemented in consideration of the amount of resources, each join points' performance and current load distribution. Finally, the result of the experiment indicates that the improved PSO can effectively reduce the number of virtual machine migration and the number of physical nodes use, the scheduling system can improve the efficiency of dispatching resource in the Cloud Computing system.

1. Introduction

Cloud Computing is a computing model, providing resource to users by internet, of which the infrastructure cloud of these resources need not be understood, acknowledged or controlled by users. Also, Cloud Computing is a business model that distributes task to computer resource pool, so that various application systems can obtain necessary computing power, storage space and a variety of software resources accordingly. (Cheng, 2009).

In cloud computing system, because of the complexity and uncertainty of cloud computing, resources are dynamic changes. In order to improve the resource utilization, how to schedule resource has become the core mechanism in cloud computing system. (Fan and Wang 2009), Early cloud resource scheduling research mainly focused on improving the resource utilization rate. The current research is mainly energy saving aspects in the cloud computing system, it is mainly through the virtual machine migration and turn off unnecessary servers to reduce the system energy consumption. Therefore, in Cloud Computing Center, on the one hand to improve resource utilization, on the other hand, to ensure the quality of services and improve the application performance of the cloud applications. However, multiple targets are conflict in each other (He and Qu, 2009).

As an innovative artificial in intelligence skill, Swarm intelligence 1 fits for the solution of complicated problems of optimization, which is enlightened by aggregated activities of animals , such as bird flocks, ant colonies, and fish schools, and so on. Recently, in order to solve actual problems, a number of methods of algorithmic Swarm intelligence have been worked out, the most effective of which is Particle Swarm Optimization (PSO). Enlightened by the biological swarming's activities, such as school of fish, in flocks of birds, and even that of mankinds (Clerc, 2005). PSO is an optimization instrument, carried out to settle a variety of function optimization problems. Being as a skill to solve problems, the main advantage of PSO is its high speed of convergence, which exceeds that of Evolutionary Algorithms (EA) and the rest global optimization algorithms. Therefore, how to obtain the load information of each node and how to use PSO measurement and evaluation of the local agent to schedule resource will be the main research directions in clouding system.(Hayes,2003)

Please cite this article as: Zhao H.W., Shen H.Y., 2015, Optimization of resource schedule based on improved particle swarm algorithm in cloud computing environment, Chemical Engineering Transactions, 46, 391-396 DOI:10.3303/CET1546066

2. Related works

This paper focuses on the improvement on the retrieval speed of resource and the construction of Cloud Computing system with high efficiency by studying the scheduling methods and load balancing features under the Cloud Computing environment (Gu, 2007).
GFS, the abbreviation of Google File System (Google File System) launched by Google, can meet the processing data requirements with rapid growth. Google File System is a scalable distributed file system, suitable for the large-scale, distributed, large amount of data access applications, running on ordinary low-cost hardware devices, while providing resource of higher overall performance and fault tolerance functions with a large number of users. However, a GFS cluster always handles the Master scheduling request by a global agency to, so that the global resource agency will become the system bottleneck under the large-scale Cloud Computing environment with more nodes.
In cloud computing environment, cloud resources can be divided into virtual resources and physical resources. Therefore, the scheduling of cloud resources can be divided into two levels, they are physical layer and virtual layer respectively. The meaning of virtual layer is submitted by user jobs to be divided into multiple tasks, as the core of the scheduling is to the task allocation appropriate virtual resources.The physical layer refers to the meaning reflects the relations between virtual and physical resources.
At present, many research work has been carried out for cloud computing resource scheduling method, presented the cloud utility maximization model, and compared with the traditional scheduling model. Objective function is no longer to minimize the maximum completion time, but to reach the effect of using maximum scheduling goals. The cloud utility maximization model is relatively simple, it cannot meet the expanding cloud computing scale. According to the new features of cloud computing, proposed the adaptive management model of virtual machine placement in cloud computing environment. However, the model of the cloud computing is not perfect, and the optimization cannot reach the desired effect. With the development of biological heuristic algorithms, many scholars proposed cloud resource scheduling algorithm based on ABC(artificial bee colony)algorithm, fish, PSO(particle swarm optimization) , these biological heuristic algorithm has characteristics of simple structure, strong search capability and can quickly find the optimal solution of the cloud resource scheduling, improve the efficiency of cloud resource scheduling. However, biological heuristic algorithm have some shortcomings in practical application. These shortcomings mainly include the slow convergence speed, easily falling into local optima, and thus cannot obtain optimal cloud resource scheduling scheme. Therefore, we need to improve the biological heuristic algorithm (Eberchart and Kennedy, 1997).
In the scheduling strategy of Cloud Computing system, we need to coordinately use the distributed resource. In order to implement the load balancing and improve resource utilization, how to schedule resource becomes a central mechanism in Cloud Computing system, while the scheduling strategy depends on the load information acquisition and the technology processing computing node information (Liu, 2011).

3. Resource Scheduling Base Particle Swarm Optimization

3.1 Particle Swarm Optimization

Particle swarm optimization algorithm simulates the behavior of a bird of prey. In the particle swarm optimization algorithm, each solution is considered to be a Particle in space. Each one has its own position and speed and there is the fitness value that is determined by an optimal function.5,6 In addition, each particle knows best position and current location so far in the entire swarm, which using the following information to change their current position:
(a) The current position;
(b) The current speed;
(c) The previous best position;
(d) The best position of the entire swarm.
Particle swarm can be thought of as a particle in D dimensional space and it can transfer information according to certain rule. The ith particle is represented as vectors of $x_i = (x_{i1}, x_{i2}, ...x_{iD})$ in the D dimensional space, the location of the ith particle in the D dimension space is X. the ith particle velocity is also a D dimensional vector, denoted as $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$. In the formula (1) and (2), i=1,2... m, m is the total number of particles in the swarm; V is the ith particle flight velocity vector; p_{id} is individual best position of particle; p_{gd} is best position in the swarm; c1, c2: weight; R1,R2 is random valus of [0, l]; By formula (1), (2) the ith particle decision the next position (Cheng and Zheng, 2009).

$$v_{id}(t) = \chi(v_{id}(t-1) + R_1 c_1(p_{id} - x_{id}(t-1)) \\ +R_2 c_2(p_{gd} - x_{id}(t-1))) \tag{1}$$

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t) \tag{2}$$

where c1, c2 are learning rates,R1 ,R2 are random values, the best position of the ith particle is p_{id}, p_{gd} is the best position of the ith particle in its neighborhood, and the constriction factor is χ :

$$\chi = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|} \tag{3}$$

3.2 Improved particle swarm optimization algorithm DPSO

The PSO algorithm has a good convergence and easy to achieve local optimum. The improved particle swarm optimization algorithm (DPSO) is improved by the dynamic multi-group collaboration. Each iteration improves the convergence rate of the algorithm, and the maximum speed $V_d^{\max}$ can be used to select the mutation particles. Through the mutation particle flight, the disadvantage of local optimization is reduced, and the diversity of the population is maintained, so as to improve the performance of the algorithm and the accuracy. $P = \{S_1, S_2, ..., S_M\}$ indicates M swarms, $S_k = \{X_1^k, X_2^k, ..., X_N^k\}$ is N particle, formula is as follows.

$$v_{id}^k(t) = \chi(v_{id}^k(t-1) + R_1 c_1(p_{id}^k - x_{id}^k(t-1)) \\ +R_2 c_2(p_{gd}^k - x_{id}^k(t-1)) + R_3 c_3(p_{gd}^\theta - x_{id}^k(t-1))) \tag{4}$$

$$x_{id}^k(t) = x_{id}^k(t-1) + v_{id}^k(t) \tag{5}$$

$$v_{id}^t = \begin{cases} v_d^{\max}, v_{id}^t > v_d^{\max} \\ -v_d^{\max}, v_{id}^t < -v_d^{\max} \end{cases} \tag{6}$$

The pseudo code of the DPSO algorithm is as follows:

```
Set  $t$ == 0;   Initialize. Source N swarms
each swarm M particles;
WHILE (not the conditions of termination )
     FOR (each swarm $k$)
          Calculate the  $k^{th}$ swarm neighborhood,
Find the best fitness of point;Set this point as  $p_{gd}^\theta$ ;
FOR (each particle)
Find in the particle neighborhood, the point with; Set this
point as  $p_{gd}^k$ ;the best fitness is replaced;
Update  velocity  of particle by (4);
Update  position of particle by (5);
           END FOR
    END FOR
    Set  $t$ := $t$ + 1;
END WHILE
```

4. Benchmark Test

4.1. Test Function and Parameters

For any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class”13. To fully estimate the performance of DPSOBS algorithm, rather than coming to a conclusion with bias facing the above problems, 5 bench mark functions have been employed as follows. These benchmark functions can be classified from f1 to f5. Parameters of the test functions show in table 1 and the formulas of such functions are shown as below:

Table 1: Parameters of the test functions

	Dim	Initial Range	population Size
f1	20	[50,100]D	40
f2	20	[15,30]D	40
f3	20	[2.56, 5.12]D	40
f4	20	[100,300]D	40
f5	20	[15,30]D	40

The size of population of all algorithms tested was set as 100 in this experiment. 15 For canonical DPSOBS PSO, the learning rates c1=c2=2.0, the initial value of inertia weight ω is 0.9 and the end is 0.5.

The experiment runs for 100 times for respective algorithm on every benchmark function. The amount of generations was set to be 100 for the 5 benchmark functions.

f1=Sphere

$$f_1(x)=\sum_{i=1}^{D} x_i^2 \qquad (7)$$

f2=Rosenbrock

$$f_2(x)=\sum_{i=1}^{D} 100\times(x_{i+1}-x_i^2)^2+(1-x_i)^2 \qquad (8)$$

f3=Rastrigrin

$$f_3(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10 \qquad (9)$$

f4=Griewank

$$f_4(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1 \qquad (10)$$

f5=Ackley

$$f_5(x)=-20\exp(-0.2\sqrt{\tfrac{1}{30}\sum_{i=1}^{D}x_i^2})$$
$$-\exp(\tfrac{1}{30}\sum_{i=1}^{D}\cos 2\pi x_i)+20+e \qquad (11)$$

4.2 Simulation results for benchmark functions

The experimental results are showed in Table 1, such as the best, worst, average, and standard deviation of the function values, and all algorithms are terminated after 100,000 function evaluations:

From the results, we test the three DPSO and PSO algorithms. Table 2 lists the results of experiment for every algorithm on functions from f1 to f4.

As shown in Table 2, the DPSO converged rather rapidly to obviously better results than the rest of algorithms, which is the most rapid one for searching for good results within comparatively few generations. As illustrated in Figure 1, PSO is worst not only on its convergence speed, but also on its performance on all benchmark functions. On Sphere function, all algorithms perform very well. DPSO is superior to others especially on convergence speed, which can be seen in Figure 1.

From Table 2 and Figure 1, the convergence speed of DPSO is much higher than the others as to finding good results within relatively few generations, the DPSO is the fastest one. All algorithms were able to consistently find the minimum to functions f1, f2 and f3 within 100 generations.

After comparing DPSO with PSO algorithms, an obvious result can be seen that the performance of DPSO is significantly superior to others on continuous unimodal functions f1 ~ f5. From the rank values illustrated in Table 2, the order of each performance among the above 2 algorithms is DPSO > PSO

Table. 2 Results comparison of different optimal algorithms for 20D

30D		DPSO	PSO
Sphere	Average	**1.25E-18**	2.24E-04
	Best	**2.89E-18**	2.30E-07
	Worst	**4.35E-19**	4.33E-06
	Std	**1.95E-18**	7.00E-09
Rosenbrock	Average	**2.23E-01**	2.34E+01
	Best	**3.38E-02**	4.35E+02
	Worst	**2.34E+00**	3.35E+00
	Std	**1.29E-01**	3.42E+02
Quadric	Average	**3.23E-05**	3.20E+02
	Best	**2.35E-11**	2.42E+02
	Worst	**4.78E-04**	3.37E+02
	Std	**2.23E-06**	2.96E+01
Sum of different powers	Average	**1.35E-02**	7.23E+04
	Best	**3.82E-04**	8.35E+03
	Worst	**2.35E-03**	2.45E+03
	Std	**2.12E-06**	1.23E+01

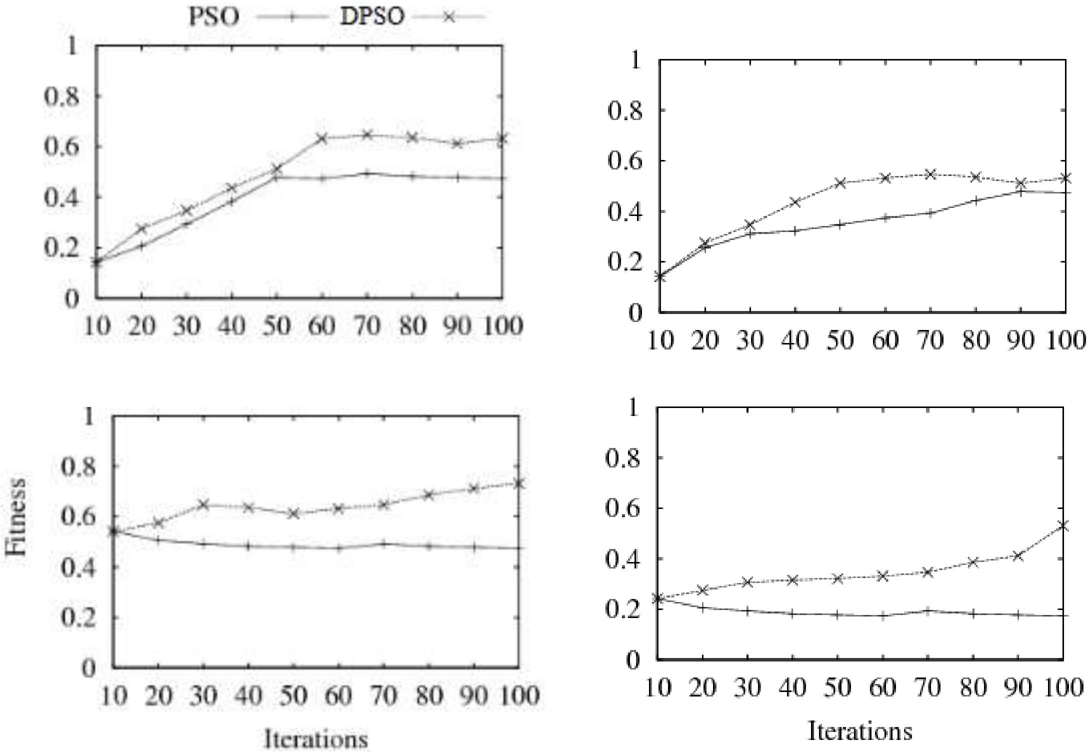

Figure 1: The median convergence results of 20D unimodal continuous functions.(a) Sphere function. (b) Rosenbrock function. (c) Quadric function. (d) Sum of different powers.

5. Conclusions

In this paper, a kind of recourse schedule method in Cloud Computing system is proposed in this paper, followed by the analysis of the specific model and technology related to Cloud Computing system, and then a resource distribution method base on DPSO has been given out. DPSO is a multi-objective optimization algorithm which considers resource utilization, cloud application service quality and performance. Finally, the result of the experiment indicates that the scheduling system can improve the efficiency of dispatching resource and the utilization ratio in the Cloud Computing environment.

Acknowledgments

The authors are supported financially by International cooperation project(Project No.S2012ZR0191) and the Natural Science Foundation of Liaoning Province (Project No.2013020011) and the Social Science Foundation of Liaoning Province (Project No.L14ASH001) and this work is supported in part by the International S&T Cooperation Program of China (ISTCP) under Grant 2011DFA91810-5 and Program for New Century Excellent Talents in University of Ministry of Education of China under Grant NCET-12-1012.

Reference

Cheng K., Zheng W.M., 2009, Cloud Computing: system instance and Research Journal of Software (05): 1337-1348.

Cheng Y.M., Jiang M.Y. and Yuan D.F., 2009, A Novel Clustering Algorithms based on Improved Artificial Fish Swarm Algorithm. Sixth International Conference on Fuzzy Systems and Knowledge Discovery. IEEE Press (3); 141-145.

Clerc M., 2005, Binary Particle Swarm Optimizers: Toolbox, Derivations, and Mathematical Insights, Available: http://clerc.maurice.free.fr/pso/.

Eberchart R.C. and Kennedy J., 1997, A new optimizer using par-ticle swarm theory, Proceeding of the 6th International Sympo-sium on Micromachine and Human Science, M.Dorigo, L. M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, pp. 53-66.

Fan Y.J., Wang D., Sun M.M., 2007, Improved Artificial Fish Swarm Algorithm. Journal of Chongqing Normal University (Natural Science Edition), 24(3):24-26.

Gu L.M., 2007, Micro-Computer Information 4, 20.

Hayes B., 2009, Cloud Computing Communications of the ACM. Vol. 51, p.9-11.

He X.D., Qu L.D., 2009, Artificial Fish Swarm Clustering Algorithm. Application Research of Computers, 26(10): 3666-3668.

Li D.M. and Shi H.H., 2003, A Hierarchical Load Balancing Scheduling Model Based on Rules Computer Science 30, 16.

Liu J., 2011, Improved Artificial Fish Swarm Algorithm and Its Applications in Function Optimization, Journal of Shijiazhuang Institute of Railway Technology, 10(3):33-36.