Optimal Virtual Machine Resources Scheduling Based on Improved Particle Swarm Optimization in Cloud Computing

Hao Yuan, Changbing Li, Maokang Du
Key Lab of Electronic Commerce and Modern Logistics of Chongqing University of Posts and Telecommunications, Chongqing, P. R. China
Email: yln2446@21cn.com

***Abstract*—This paper presents virtual machines resources scheduling algorithm taking into computing capacity of processing elements and consideration their computational complexity. We apply the improved particle swarm optimization to solve virtual machines resources scheduling problem. The experiments show that the improved algorithms can provide effective solutions that the original algorithm can not provide on cloud systems.**

***Index Terms*—Particle swarm optimization; resources scheduling; cloudSim; cloud computing**

I. Introduction

With the development of High Performance Computing, there are many computing paradigms including Peer-to-Peer(P2P) computing, Cluster computing, Services computing,Grid computing and Cloud computing[1-2]. Efficient virtual machines resources scheduling mechanism can improve the resource utilization and meet users' requirements[3-4]..

This virtual machines resources scheduling problem can be classified as a combinatorial optimization problem and is NP-complete[5-6].Yee Ming Chen[7] presented a discrete particle swarm optimization (DPSO) approach for this problem. They constructed application Amazon EC2 as an example and simulation with Cloud based compute and transmission resources. The method was more efficient and surpasses those of mathematical programming and reflecting the actual benefit of saving with the total cost as well as tasks allocation. Xin Lu[8]proposed an ant colony algorithm for load-adaptive cloud resource scheduling model. Jianhua Gu[9]presented genetic algorithm for scheduling strategy on load balancing of VM resources. Wuqi Gao[10] put forward multi- dimension QoS for cloud simulation scheduling algorithm.

In this paper, we proposed an improved particle swarm optimization to solve virtual machines resources scheduling problem. and according to the special requirements of cloud by limiting the speed of particle in the cloud computing environment.

Manuscript received June 1, 2013; revised August 1, 2013; accepted September 1, 2011.
Copyright credit, the National Natural Science Foundation of China under Grant (No.60703035), the Natural Science Foundation of Chongqing University of Posts and Telecommunications(No. A2011-21) corresponding author: Hao Yuan.

II. Backgrounds

The typical virtual machine resource scheduling model is showed in **Figure 1**:

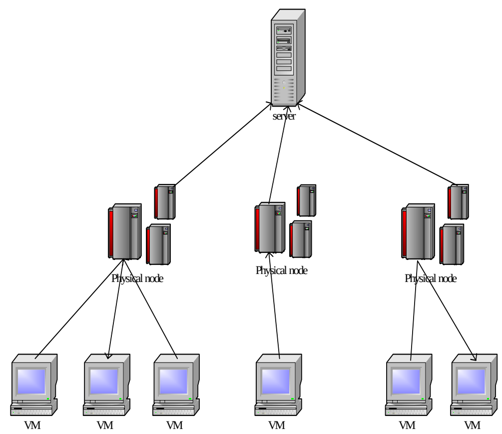

Figure 1 Virtual machine resource scheduling model

Virtual machine resource scheduling problem can be formalized as follows[11-14]: cloud users need physical node to provide a large number of VM, defined virtual machine cluster $V=\{V_1,V_2,...,V_n\}$, where n is the number of VM in the cluster. Physical node cluster $P=\{P_1,P_2,...P_m\}$, where m is the number of physical nodes in the cluster. i and j are used to represent the performance of VM and physical nodes. The four basic resources of are CPU, memory, network bandwidth, disk, assume that the resource type from the physical node in the data center are only four.

III. Improved Particle Swarm Optimization for Virtual Machines Resources Scheduling

A. Algorithm Design

(1) Encoding

$T=\{t_1,t_2,...,t_n\}$ represents the tasks waiting to be scheduled per unit of time, n is the number of tasks.

$N=\{n_1,n_2,...,n_m\}$ represents the set of nodes in the cloud system, assuming that the cloud system has

doi:10.4304/jsw.9.3.705-708

m nodes. For cloud computing system, n_i represents the computing resources on the n_i ; For cloud storage system, n_i represents the data on a n_i .

$V = [v_1, v_2, ..., v_n]$ represents the task scheduling vectors or a scheduling scheme. For cloud storage system, v_i represents the i-th task of data is provided by resources nodes that represented by v_i value, and the length of the vector is the total amount of scheduling tasks per unit time.

For example, a task scheduling vector [5, 1, 3, 2, 1, 6], the length of this vector is 6, and represents needs to schedule task number is 6 per unit of time. The value based on the position of No. 1 is 5, represent the data of the task 1 is provided by the system node 5.

So, the data of task 2 and 5 are provided by the node 1; the data of task 3 is provided by the node 3; the data of task 4 is provided by the node 2; the data of task 6 is provided by the node 6. For cloud computing system, it is on behalf of a task placed in a node.

(2) Fitness function design

B represents the bandwidth matrix of the cloud system. According to the assumption, the m-th node in the cloud system, then B is a matrix of $m \times m$, b_{ij} represents bandwidth between system node i and node j . Path between nodes is likely more than one; the bandwidth may have more than one value. Now generally max value. Based on the above assumptions, the fitness function is defined as:

$$F(i, j) = \max(\sum_i \frac{b_{ij}}{t_i}) \qquad (5)$$

where i represents the task number, j represents the resource node number, b_{ij} is bandwidth between the node i and node j . The function is in fact an evaluation of good and bad scheduling results.

(3) Iteration of particle position and velocity

To ensure the diversity of the scheduling scheme, the calculation in the iteration following conditions are met:

(a) Position vector of particles in initial particle group is defined within a certain range; in the iteration process if it exceeds this range, the need to apply rules revalue..

(b) The velocity vector of initial particle is limited within a certain range, in the iteration process if it exceeds this range, the need to apply rules revalue.

Set the local optimum particle in the current particle swarm expressed as X_p ,Global optimal particle is represented as X_g ,Updating the particle velocity and position according to the equation (2) and (3) respectively. Particle swarm optimization is by following the evolution equation velocity and position update.

$$V_{i+1} = \omega \times V_i + c_1 \times D_1 + c_2 \times D_2 \quad (6)$$

$$X_{i+1} = X_i + V_{i+1} \quad (7)$$

Where r_1, r_2 are random number between (0,1)； $D_1 = r(X_p - X_i)$, $D_{\min} \le D_1 \le D_{\max}$; $D_2 = r(X_g - X_i)$, $D_{\min} \le D_2 \le D_{\max}$ ； c_1 and c_2 are Learning factor, c_1 Reflects the cognitive ability of particles, and adjusts the particle to fly its own best position direction, c_2 Reflects the particle groups ability to learn, and adjusts the particle to fly the global best position direction；

ω indicates a coefficient weight, so that the particles maintain movement inertia, ω reflects the influence of the speed of the last iteration to the current speed, weight expression is:

$$\omega = (\frac{D_{\max} - D_{\min}}{D_{\max} + D_{\min}}) \times i \div L \qquad (8)$$

Where i the current number of iterations, L is is the total number of iterations, if ω is the greater value, the particle swarm algorithm has strong global search capability; if ω is the smaller value, the particle swarm algorithm has been tending to the local search.

(4) Particle speed limit

To limit the velocity of the particle, the maximum speed of the r-dimensional particle $v_r^{\max} = \mu \times (\frac{u_r - l_r}{u_r + l_r}), \mu \in [0,1]$, v_d is the velocity of the r-th dimension particles, u_r and l_r are the upper and lower limits of the r-dimensional space。

The purpose of the speed limit updated is: When the difference between the current position, the individual optimum position, and optimum position is large, especially after release of the particle. While excessive speed can cause the particle converge to the global optimal quickly is difficult. Speed limit can also avoid such situation, and increase the convergence rate, also retain the release of particles search capabilities so that it can converge to the global optimum near when being released again, and have the opportunity to update the population optimal.

B. The Procedure of IPSO

The IPSO algorithm steps are as follows:

Step1: Determining the size of the particle swarm, random initialization of the position and velocity of the particle, where in $x_r \in [l_d, u_d], |v_r| \le v_r^{\max}$;

Step2:Calculated ω according to formula (8),updated the velocity of the particle according to formula (6),if $v_r > v_r^{\max}$,then $v_r = v_r^{\max}$,if $v_r < -v_r^{\max}$,then $v_r = -v_r^{\max}$; updated the velocity of the particle according to formula (7) again, if $x_r < l_r$,then $x_r = l_r$,if $x_r > u_r$,then $x_r = u_r$;

Step3:Calculate the fitness of each particle and update the personal best of each particle and populations globally optimal;

Step4: If algorithm reaches the maximum number of iterations, return a global optimal solution gbest; otherwise, skip to Step2.

IV. Experimental Results and Analysis

In this paper, in order to demonstrate the effectiveness of the developed method, we designed experiments of GA [9]; PSO [11] and IPSO are compared with. The operating system is Windows XP SP3, MyEclipse version 8.5, the JDK version is jdk1.6.0. CloudSim version is CloudSim-3.0[15-16]; Ant build tool is Ant 1.8.1. In order to achieve IPSO algorithm, we write the inherited class in the basic class to be extended CloudSim. The network topology is shown in **Figure 2**, which is shown in the existing algorithms similar experiment. It consists of 10 nodes, numbered from 0-9. Value represents the bandwidth of the two directly connected between the points of the connecting line between the nodes, units trillion (m). Where in indirectly between the two nodes may exist multiple paths, the bandwidth is defined indirectly between two possible paths for the average bandwidth. Statistical comparisons data of the three algorithms are shown in **Table 1**. The three algorithms computing convergence curve are shown in **Figure 3,4**, Average SLA of three algorithms is shown in **Figure 5**. Average completion time of three algorithms is shown in **Figure 6**.

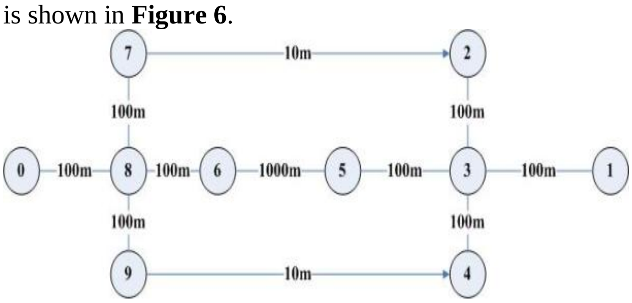

Figure 2. Experimental network topology

TABLE 1.
THREE ALGORITHMS FOR SCHEDULING（100 STATISTICS）

Algorithm	Fitness value	The number of iterations	Run time (ms)
GA	0.5757	370	3850
PSO	0.7834	293	3300
IPSO	0.8414	211	2044

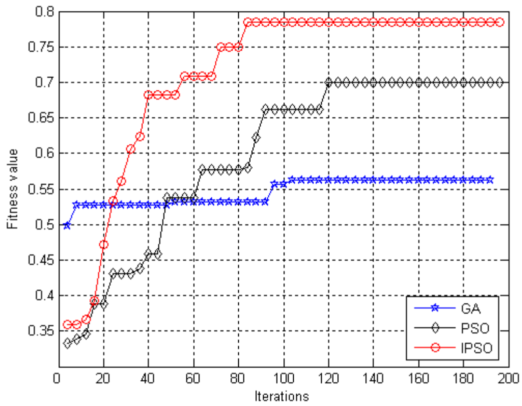

Figure 3 Convergence curve of three algorithms （200 iterations）

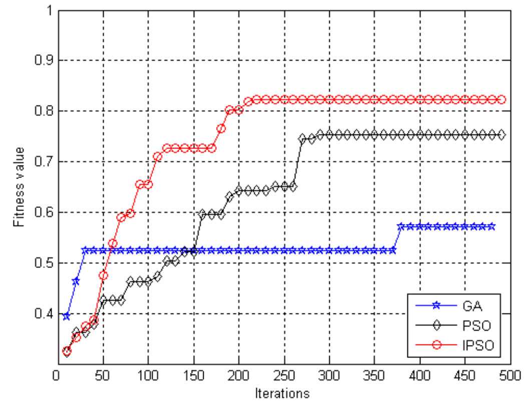

Figure 4 Convergence curve of three algorithms （500 iterations）

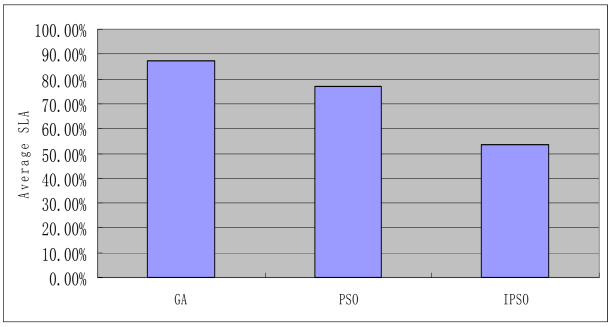

Figure 5 Average SLA of three algorithms

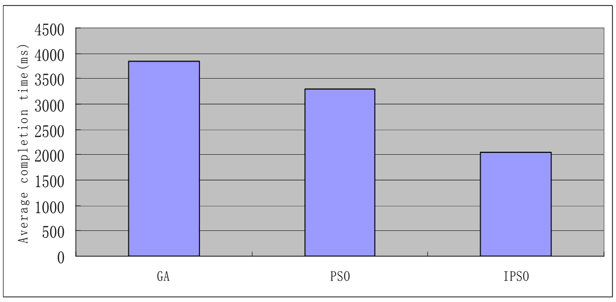

Figure 6 Average completion time of three algorithms

As can be seen from **Figure 3, 4** and **Table 1**, the IPSO algorithm solves the problem with good results and possible solutions are closer to the optimal solution. When PSO algorithm or GA algorithm solve the problem, in the generation of less than 300, were unable to reach the optimal solution that IPSO can achieve. So IPSO has better convergence, and can effectively avoid PSO algorithm is easy to fall into local optimal solution. From **Figure 5, 6**, we can see this IPSO algorithm is maintaining lower average SLA and shortest average completion time relative to PSO and GA algorithm.

In summary, the experiments show that the improved algorithms can provide effective solutions that the original algorithm can not provide on cloud systems , and save a lot of the execution time of the original algorithm.

V. CONCLUSIONS

This paper presents virtual machines resources scheduling algorithm. We apply the improved particle swarm optimization to obtain the optimal solution in reasonable time. The experiments show that the improved algorithms can provide effective solutions that the original algorithm can not provide on cloud systems.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China under Grant (No.60703035), the Natural Science Foundation of Chongqing University of Posts and Telecommunications (No. A2011-21)

REFERENCES

[1] SM. Armbrust, A. Fox, R. Griffith, A. D. Joseph, et al. A view of cloud computing [J]. *Communication of the ACM*, 2010, 53(4): 50-58

[2] Mark Stillwell, David Schanzenbach, Frederic Vivien, et al. Resource allocation algorithms for virtualized service hosting platforms [J].*Journal of Parallel and Distributed Computing*, 2010,70(9):962-974

[3] Baomin XU, Ning Wang, Chunyan Li. A Cloud Computing Infrastructure on Heterogeneous Computing Resources[J]. *Journal of Computers*, 2011,6(8): 1789-1796.

[4] S. Sindhu, Saswati Mukherjee. Efficient Task Scheduling Algorithms for Cloud Computing Environment [J].*Communications in Computer and Information Science*, 2011, 169(1):79-83

[5] Daniel Warneke, Odej Kao. Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2011, 22(6):1045-9219

[6] Daji Ergu, Gang Kou, Yi Peng, et al. The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment [J].*The Journal of Supercomputing*,2011,

[7] Yee Ming Chen, Shin-Ying Tsai.Optimal Provisioning of Resource in a Cloud Service[J].*International Journal of Computer Science*,2010,7(6):95-99

[8] Xin Lu; Zilong Gu. A load-adapative cloud resource scheduling model based on ant colony algorithm[C]. *IEEE International Conference on Cloud Computing and Intelligence Systems* (CCIS), 2011: 296 – 300

[9] Jianhua Gu, Jinhua Hu, Tianhai Zhao, et al. New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment [J].*Journal of Computer*, 2012,7(1):42-52

[10] Wuqi Gao, Fengju Kang. Cloud Simulation Resource Scheduling Algorithm Based on Multi-dimension Quality of Service [J]. *Information Technology Journal*, 2012,11(1): 94-101.

[11] R. Shakerian, S. H. Kamali, M. Hedayati, et al. Comparative Study of Ant Colony Optimization and Particle Swarm Optimization for Grid Scheduling[J].*The Journal of Mathematics and Computer Science*, 2011, 2(3):469-474

[12] Qiang Li, Yike Guo. Optimization of Resource Scheduling in Cloud Computing[C]. Proceedings of the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Washington, DC USA,IEEE Computer Society,2010:315-320

[13] Guojun Ding, Lide Wang, Peng Yang, Diagnosis Model Based on Least Squares Support Vector Machine Optimized by Multi-swarm Cooperative Chaos Particle Swarm Optimization and Its Application[J]. *Journal of Computers*,2013,8(4):975-982

[14] Bo Xu,Qing Guan,Ke Chen. Multi-Agent Coalition Formation Based on Quantum-behaved Particle Swarm Optimization [J]. *Journal of Information & Computational Science*, 2010, 7(5): 1059-1064

[15] CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services [R]. *Melbourne: University of Melbourne*, 2009

[16] Huaibin Wang, Haiyun Zhou, Chundong Wang. Virtual Machine-based Intrusion Detection System Framework in Cloud Computing Environment [J]. *Journal of Computers* 2012,7(10):2397-2403

Hao Yuan He is a master, who now works at Key Lab of Electronic Commerce and Modern Logistics of Chongqing University of Posts and Telecommunications. He was born in 1970. His main research interests include computational intelligence and the Internet of things.

Changbing Li He is a doctor, who now works at Key Lab of Electronic Commerce and Modern Logistics of Chongqing University of Posts and Telecommunications. He was born in 1970. His main research interests include computational intelligence and logistics planning.

Maokang Du He is a master, who now works at Key Lab of Electronic Commerce and Modern Logistics of Chongqing University of Posts and Telecommunications. He was born in 1969. His main research interests include computational intelligence and logistics planning.