A PSO-Based Algorithm for Load Balancing in Virtual Machines of Cloud Computing Environment

Zhanghui Liu and Xiaoli Wang

College of Mathematics and Computer Sciences, Fuzhou University. Fuzhou, China
yuhcaolong@126.com, fzugwz@163.com

Abstract. It is possible for IT service providers to provide computing resources in an pay-per-use way in Cloud Computing environments. At the same time, terminal users can also get satisfying services conveniently. But if we take only execution time into consideration when scheduling the cloud resources, it may occur serious load imbalance problem between Virtual Machines (VMs) in Cloud Computing environments. In addition to solve this problem, a new task scheduling model is proposed in this paper. In the model, we optimize the task execution time in view of both the task running time and the system resource utilization. Based on the model, a Particle Swarm Optimization (PSO) – based algorithm is proposed. In our algorithm, we improved the standard PSO, and introduce a simple mutation 突变 mechanism and a self-adapting inertia 惯性 weight method by classifying the fitness values. In the end of this paper, the global search performance and convergence rate of our adaptive algorithm are validated by the results of the comparative experiments.

Keywords: Cloud Computing, VMs, Load Balancing, Task Scheduling, PSO.

1 Introduction

Nowadays, Cloud Computing has become a very popular commercial computing paradigm. Then the whole Cloud Computing system can provide services to users with virtual machine as the resources unit [1-2]. For users, all of the bottom resources are transparent. In theory, every job submitted by terminal users owned an independent virtual machine. The computing cells and memory cells for the jobs are in the situations of mutual isolation [3]. In Cloud Computing environment, each physical host can load one or more virtual machines, so that you can ensure users' applications run independently. So, the task scheduling in Cloud Computing happens between the virtual machines actually. Keep load balances of the VMs is the ongoing work in Cloud Computing systems.

NP难度的算法

It is an NP-hard combinational optimization problem to establish the mappings between jobs submitted by terminal users and dynamical resources encapsulated in the virtual machines. For such problem, researchers have put forward a variety of static, dynamic and mixed scheduling strategies [4-9]. Static scheduling algorithms are: ISH algorithm [10], MCP algorithm and ETF algorithm [10]. All of these algorithms are based on BNP (Bounded Number Processors), and very suitable for

Y. Tan, Y. Shi, and Z. Ji (Eds.): ICSI 2012, Part I, LNCS 7331, pp. 142–147, 2012.
© Springer-Verlag Berlin Heidelberg 2012

high performance networks in distributed environment. But the application requirements in Cloud Computing VMs are more complicated, and service costs are also required according to usages amount, those algorithms can not pay key roles. Currently, the swarm intelligence algorithms are well used for resolving these kinds of problems[11-12]. PSO is a global search optimization technique proposed by Kennedy and Eberhart in 1995[13]. But when the problem continues to expand there scales, Simple heuristic algorithm seem to be not so effective [14]. In this paper, a kind of improved PSO algorithm to solve the problem of virtual machines load balance is proposed, so as to establish corresponding relations between the tasks and the virtual machines effectively. Aiming at finding an optimal or nearly optimal scheduling solution, not only makes the task execution time the shortest, and can make the virtual machines of resource utilization the highest. The simulation results show that the algorithm has fast convergence speed, high efficiency, and has practical application significance. 收敛

2 Problem Description and Task Scheduling Model

In VMs of Cloud Computing environments, the jobs submitted by terminal users can be classified two kinds, which are independent ones and interrelated ones. The interrelated jobs can be divided into small separate tasks that can run without interferences, so we just study how to balance the workload of VMs with independent tasks. The objectives of our model are to achieve the minimum execution time of tasks and the maximum VMs resource utilization.

For simplicity, we assume that the virtual machine is resource unit of Cloud Computing environment. And for different virtual machines, the resource demands of any task are the same. The models can be represented as follows.

There are m virtual machines which are interconnected by network. The VMs can be represented by the set $V = (v_1, v_2, \cdots, v_m)$, in which v_i means the maximum resource capacity that virtual machine i can provide, where $i \in [1, m]$.

2.1 Task Model

There is a task sequence $T = (t_1, t_2, \cdots, t_n)$, in which t_j means the task which is number j, where $j \in [1, n]$, and n is the length of this sequence. Task model is defined as $t_j(timeNeed, resourceNeed)$, in which $timeNeed$ denotes the task's 表示 execution time, and $resourceNeed$ denotes the resource requirements of the task.

2.2 Objective Model

We use one-zero matrix A to represent the mapping relationships between tasks and VMs. There has $\sum_{i=1}^{m} a_{i,j} = 1$, and $i \in [1, m], j \in [1, n]$.

Which is can be described:

$$A = \begin{bmatrix} a_{1,1}, a_{1,2}, \cdots, a_{1,n} \\ a_{2,1}, a_{2,2}, \cdots, a_{2,n} \\ \vdots \quad \vdots \quad \vdots \\ a_{m,1}, a_{m,2}, \cdots, a_{m,n} \end{bmatrix}.$$

Based on the above models, we have the equations as follows.

$$\begin{cases} VTime = \max\limits_{i=1}^{m} \left[\sum\limits_{j=1}^{n} (a_{i,j} \times t_j.timeNeed) \right] \\ VRutilization = \sum\limits_{i=1}^{m} \left(\dfrac{\sum\limits_{j=1}^{n} a_{i,j} \times t_j.resourceNeed}{v_i} \right) \end{cases}. \tag{1}$$

where we use $VTime$ to denote the execution time of VMs for executing all of the tasks, and $VRutilization$ to denote the resource utilization of VMs during the process of running the tasks. So, the objective function of the task scheduling model is:

$$\begin{cases} Min(VTime) \\ Max(VRutilization) \end{cases}. \tag{2}$$

3 Optimization Algorithm Description

Resource allocations and scheduling strategies are combined to realize the mappings from tasks to VMs. In this paper, we introduce mutation operator and self-adaptation of inertia weight to the standard PSO algorithm aiming at virtual machines assignment for the user tasks.

3.1 Fitness Function

In order to measure how well the particle's position, the fitness function can be defined:

$$f = Min(\tfrac{VTime}{VRutilization}). \tag{3}$$

3.2 Define the Positions and Velocities of Particles

Suppose there is a N-dimension particle $X = (x_1, x_2, \cdots, x_n)$, where $x_i (i \in [1, n])$ is the serial number of virtual machine on which the number i task is processed. At the same time, an N-dimension velocity $V = (v_1, v_2, \cdots, v_n)$ is defined, where $v_i (i \in [1, n])$ means the velocity of x_i. And

$$\begin{cases} x_i \in Z^+ \ \&\&1 \le x_i \le m \\ v_i \in [-v\max, v\max] \end{cases} . \tag{4}$$

where $v\max$ denotes the maximum velocity component of particle. In this paper, we set the $v\max = m$ [15]. But, it always leads to precocity. We introduce a simple mutation mechanism: When overflow occurs, the position will get random value from the solution space.

3.3 Self-adapting Inertia Weight and Updating Positions and Velocities

$$\begin{cases} w = \begin{cases} 0.2, (\left|\dfrac{fi - fg}{\max(fi, fg)}\right| < 0.2) \\ 0.8, (\left|\dfrac{fi - fg}{\max(fi, fg)}\right| > 0.8) \\ 1 - 0.2 \cdot e^{-(fi-fg)^2 \cdot rand()}, else \\ \end{cases} \\ vx_i^{k+1} = w \cdot vx_i^k + c_1 \cdot rand() \cdot (p_i^k - x_i^k) + c_2 \cdot rand() \cdot (fg^k - x_i^k) \\ vx_i^{k+1} = \begin{cases} vx_i^{k+1}, (\left|vx_i^{k+1}\right| \le v\max) \\ v\max, else \end{cases} \\ x_i^{k+1} = \begin{cases} \left\lfloor x_i^k + vx_i^{k+1} \right\rfloor, (1 \le x_i^k + vx_i^{k+1} \le m) \\ random\_value \in [1, m], else \end{cases} \end{cases} . \tag{5}$$

where, there has: w inertia weight, f_i the fitness value of particle i, p_i the previous best fitness value of particle i, fg the global best fitness value, x_i the position of particle i, vx_i the velocity of particle i, c_1, c_2 the coefficients which are set as 2.05 in this paper.

4 Experimental Results and Analysis

The optimization process is simulated by MATLAB in this paper, where has 80 virtual machines and 120 tasks, and with 200 iterations.

To validate the improvement of our algorithm(MAPSO), we compared it with non-adapting standard PSO which has the invariable inertia weight. Fig1 shows that both convergence speed and robustness of MAPSO algorithm are better than those of SPSO.

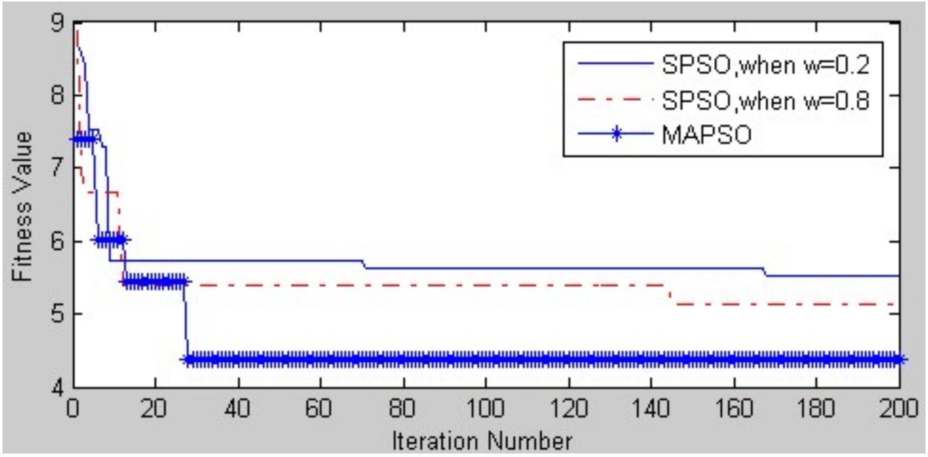

Fig. 1. Comparison of fitness values SPSO and MAPSO

5 Conclusions

Nowadays, from both scheduling flexibility and application scale, there is much work should be done on the past studies. this paper works to solve the load balancing problem in VMs of Cloud Computing environment. Based on the development of virtualization and distributed technology, we put forward MAPSO tasks scheduling algorithm by improving standard PSO. In addition, because of this experiment with simulation environment, some specific questions need to be overcome in specific actual cloud environment, such as restrictions from bandwidth, problems in job decomposition, energy costs of cloud datacenters etc.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grant No. 61103175, the Key Project Development Foundation of Education Committee of Fujian province under Grand No. JA11011 and JK2010001, the Technology Innovation Platform Project of Fujian Province under Grant No.2009J1007, the project development foundation of Fuzhou University under Grand No. 2010-XQ-21 and XRC-1037.

References

1. Virtualization and Cloud Computing Group.: Virtualization and Cloud Computing, pp.110–114. Publishing House of Electronics Industry, Beijing (2009) (in Chinese)
2. Hu, J., Gu, J., Sun, G., et al.: A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment. In: 3rd International Symposium on Parallel Architectures, Algorithms and Programming, Dalian, Liaoning, China, pp. 89–96 (2010)
3. Fang, Y., Wang, F., Ge, J.: A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. In: Wang, F.L., Gong, Z., Luo, X., Lei, J. (eds.) WISM 2010. LNCS, vol. 6318, pp. 271–277. Springer, Heidelberg (2010)
4. Paton, N.W., de Aragao, M.A.T., Lee, K., Fernandes, A.A.A.: Optimizing Utility in Cloud Computing through Automatic Workload Execution. IEEE Data Eng. Bull. 32, 51–58 (2009)
5. Li, L.: An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers. In: Third International Conference on Multimedia and Ubiquitous Engineering, Qingdao, China, pp. 295–299 (2009)
6. Wei, G., Athanasios, V.V., Yao, Z., et al.: A game-theoretic method of fair resource allocation for Cloud Computing Services. The Journal of SuperComputing 2, 252–269 (2009)
7. Martin, R., David, L., Taleb-Bendiab, A.: A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing. In: 2010 IEEE 24th International Conference on Advanced Information Netwoking and Applications Workshops, Perth, Australia, pp. 551–556 (2010)
8. Zhang, B., Gao, J., Ai, J.: Cloud Loading Balance Algorithm. In: 2nd International Conference on Information and Engineering, ICISE 2010, Hangzhou, China, pp. 5001–5004 (2010) (in Chinese)
9. Laura, G., David, I., Varun, M., et al.: Harnessing Virtual Machine Resource Control for Job Management. In: The 1st Workshop on System-level Virtualization for High Performance Computing, Lisbon, Portugal (2007)
10. Kwok, Y.-K., Ahmad, I.: Static scheduling algorithms for allocating directed task graphs to multiprocessors. ACM Computing Surveys 4, 406–471 (2009)
11. Ji, Y.-M., Wang, R.-C.: Study on PSO algorithm in solving grid task scheduling. Journal on Communications 10, 60–66 (2007) (in Chinese)
12. Pandey, S., Wu, L., Guru, S., et al.: A Particle Swarm Optimization (PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. In: 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia, pp. 400–407 (2010)
13. James, K., Russell, E.: Particle Swarm Optimization. In: Proceedings of Neural Networks 1995, Perth, Australia, pp. 1942–1948 (1995)
14. Zhou, H.-R., Zheng, P.-E.: Optimization for parrel multi-machine scheduling based on hierarchial genetic algorithm. Computer Applications, 2273–2275 (2007) (in Chinese)
15. Zhou, C., Gao, H.-B., Gao, L., et al: Particle Swarm Optimization (PSO) Algorithm. Application Research of Computers, pp. 7–11 (2003) (in Chinese)