http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (23): 7090-7095, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.7090.7095

Optimized Task Scheduling and Resource Allocation in Cloud Computing Using PSO based Fitness Function

[1,2]Zhongxue Yang, [1]Xiaolin Qin, [2]Wenrui Li and [3]Yingjie Yang
[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, 210016, Nanjing, China
[2]School of Mathematics and Information Technology, Nanjing Xiaozhuang University, 211171, Nanjing, China
[3]Centre for Computational Intelligence, De Montfort University, Leicester, LE19BH, UK

Abstract: Cloud computing environment can offer dynamic and elastic virtual resources to cloud users on demand. It becomes an attractive challenge that how task scheduling satisfy the dynamic requirements of users and utilize the virtual resources efficiently in cloud environment. Particle Swarm Optimization (PSO) is a global metaheuristic method to solve optimization issues. The processing capacity (cost) and makespan associated with the task schedule and the resources allocated are taken into account to measure the performance of optimization algorithm in this study. PSO based fitness function scheduling heuristic to balance the load across the entire system is introduced while trying to minimize the makespan and increase the processing capacity. For comparison, ant colony algorithm is presented to simulate on the same datasets. The experiments results show that PSO based fitness function is more effective and efficient with shorter completion time and lower cost.

Key words: Task scheduling, resource allocation, particle swarm optimization, cloud computing, fitness function

INTRODUCTION

Cloud Computing is a distributed architecture that centralizes server resources on a scalable platform so as to provide on demand computing resources and services which is a model for enabling convenient and on demand accessing of the resources. This model of cloud offers a shared pool of resources that are available on customer's demand and can be accessed at anytime from anywhere (Bhupendra and Kapoor, 2013). The concepts behind cloud computing incorporate Software-as-a- Service (SaaS), Platform-as-a-Service(PaaS) and Infrastructure-as-a-Service(IaaS), respectively. Cloud computing offers the prospect of facilitating the development of large scale, flexible computing infrastructures and provides an extensible and powerful environment for growing amounts of services and data by means of on-demand self-service. The collaboration of technologies has allowed most talked and popular term to cloud computing in recent years. Most of the organizations have been shifted their business to the cloud computing environment and had a great impact on their services. Thus the growth of the IT technologies with cloud computing has reached to the customer in an efficient manner. As cloud offers a package of services that enables a whole new way of using IT and can be accessed on demand.

In cloud computing, resources, such as storage, computing capacities in virtual machines, are provided as a service and cloud service providers authorize their users to allocate, access, or control the resources. Therefore, resources allocation and task scheduling in cloud computing becomes more significant while a large amount of tasks being sent to cloud environment at the same time. Thus, task scheduling should satisfy the dynamic requirements of users and need to utilize the virtual resources efficiently in cloud environment. In cloud computing environment, machines are located in different regions and have disparate processing abilities, characteristics. In these situations, the cost and makespan associated with the task schedule and the resources allocated should be taken into account. Therefore, task scheduling and resource allocation should be carefully coordinated and optimized jointly in order to achieve an overall cost or processing capacity and time-effective schedule. That is, to find optimal task schedules by minimizing cost and makespan.

Corresponding Author: Zhongxue Yang, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, 210016, Nanjing, China

The cloud user utilizes IT resources like storage and server as a service and pays for the service. A cloud provider constructs a computing system which consists of several virtual machines interconnected and makes profits by processing tasks from users on the computing system. Therefore, how to schedule tasks and allocate resources efficiently is an important and challenging issue in cloud computing.

In this study, a novel PSO based fitness function algorithm to solve task scheduling and resource allocation in cloud computing is proposed. The rest of this study is organized as follows. In section 2, the problem is defined. Section 3 introduces PSO algorithm and fitness function, Simulation experiment is given in 4th section and the performance of the PSO based fitness function algorithm is evaluated as well. Finally, this study is concluded in section 5.

PROBLEM DESCRIPTION

Task scheduling is considered as one of the most famous combinatorial optimization problems. The main goal is to determine a proper sequence where tasks are executed while obeying to some (transaction logic) constraints. Implementations are labeled either as centralized or decentralized, static or dynamic, or a hybrid. All with their own strengths and limitations. In general, dynamic load-balancing mechanisms provide better performance than static once, but do encounter higher overhead due to the systems information that have to be updated on-the-fly (redistribution of tasks due to overloaded CPU's may occur as well). With a static setup, task scheduling is not affected by the state of the system, it is based on a-priori knowledge. The scheduling of tasks on the cloud resources have several objectives.

To allocate resources and schedule tasks effectively, the better way is to use a dynamic scheduling algorithm among virtual machines or resources themselves to get the real-time states. If overload or idleness occurred, tasks could be readjusted and redistributed among virtual machines or resources. By dynamic scheduling in virtual machines or resources, the scheduling algorithm can allocate tasks and resources flexibly and reduce the efficiency impact caused by the synchronization among virtual machines.

To formulate the problem, cloud user tasks can be denoted the set of n independent jobs as J_i, where i = 0, 1,…, n and set of m resources as R_j, where j = 0, 1,…, m. Each task can be split into many subtasks as J_{ik}, where k = 0, 1,…, q and each subtask is allowed to be processed on any given available resource. A subtask is processed on one resource at a time and the given resources are available continuously.

For an instance, 4 tasks can be described as (J_1, J_2, J_3, J_4) and each task has 3 subtasks, also 5 resources are denoted as (R_1, R_2, R_3, R_4, R_5). A given schedule is {(J_{11}, R_0), (J_{31}, R_2), (J_{32}, R_2), (J_{41}, R_2), J_{42}, R_1), (J_{33}, R_1), (J_{21}, R_1), (J_{22}, R_4), (J_{12}, R_4), (J_{43}, R_5), (J_{13}, R_5), (J_{23}, R_3)}. The framework and Gantt chart of task scheduling resource allocation are shown in Fig. 1 and 2, respectively. However, each subtask is dependent on the resources while it is performed and each subtask must be completed without interruption once starts. Similarly, resources can not perform more than one subtask at the same time.

The problem is to assign each subtask to an appropriate resource (routing problem) and to sequence the subtasks on the resources (sequencing problem) in order to minimize the total cost and makespan. The makespan is the total length of the schedule (that is, when

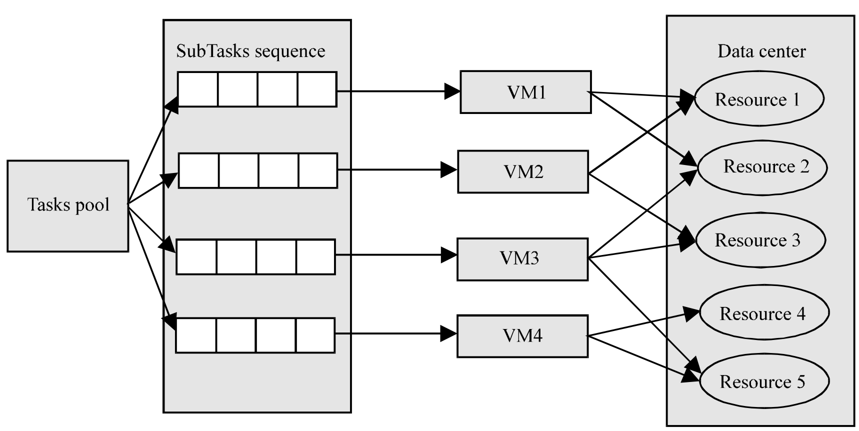

Fig. 1: Framework of task scheduling and resource allocation

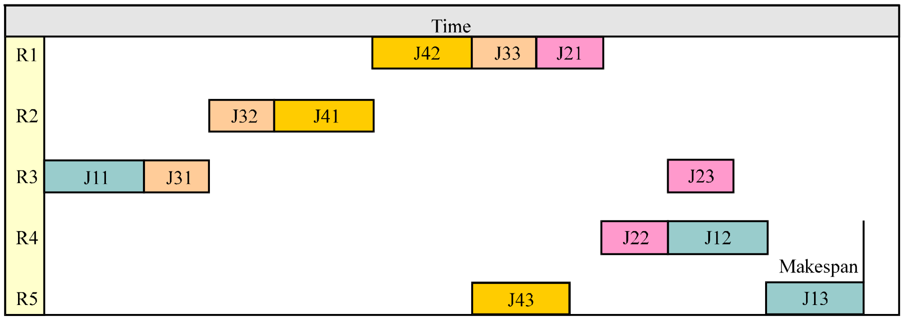

Fig. 2: Gantt chart of task scheduling and resource allocation

all the jobs have finished processing). The challenge is that task scheduling and resource allocation should be carefully coordinated and optimized jointly in order to achieve an overall cost and time-effective schedule.

PARTICLE SWARE OPTIMIZATION

Particle Swarm Optimization (PSO) is a global metaheuristic proposed by Kennedy and Eberhart (1995) and Eberhart and Kennedy (1995). The idea of PSO was based on simplified models simulating some social behaviors of animals, such as bird flocking and fish schooling. Due to its simplicity and its effectiveness, the nature of fast convergence and strong global optimization capability, PSO has received increasing attention and has been successfully employed in wide range of application to solve unconstrained optimization problems (Cai and Pan, 2012; Du and Li, 2008; Zhang *et al*., 2012) with lowest computational cost, such as reactive power optimization (Chuanwen and Bompard, 2005; Li *et al*., 2009; Li *et al*., 2009), neural network training (Eberhart and Shi, 1998; Ma and Zhang, 2007) etc. In fact, PSO is a metaheuristic as it makes few assumptions about the problem and can search very large spaces of candidate solutions. PSO does not use the gradient of the problem being optimized which means PSO does not require the optimization problem differentiable. Therefore, PSO are effective on problems that are partially irregular, noisy, change over time as well.

With regard to a given measure of quality, PSO optimizes a problem by iteratively trying to improve a population of candidate solutions, so-called particles and moving these particles around in the search-space according to a few of mathematical formula over the particle's position and velocity. Each particle's movement is influenced by its individual best known position but, is also guided toward the global best known positions in the search-space which are updated as improved positions are found by other particles. This process performs repeatedly and by doing so it is expected to move the swarm toward the best solutions.

PSO enhances its global or individual optimization capability by updating inertia weight ω, velocity v and position x of particle is updated as follows:

$$v_{id}^{t+1} = wv_{id}^{t} + \varphi_1 r_1 (p_{id}^{t} - x_{id}^{t}) + \varphi_2 r_2 (p_{gd}^{t} - x_{id}^{t})$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}$$

where, $x_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t)$ and $v_i^t = (v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t)$ indicates the position and the velocity of particle i, respectively, here $1 \le d \le D$, $1 \le i \le N$, D is the dimension of search-space and N is the number of particles in the swarm; p_{id}^t (pbest) denotes the individual best known position of particle i and p_{id}^t (gbest) denotes the global best known position in the dth dimension at t iteration. φ_1, $\varphi_2 \in \Re$ are acceleration coefficients and r_1, r_2 are random numbers in the range of [0,1] which are used to control the behavior and efficacy of the PSO method.

The inertia weight is adapted over the search iteration t as follows:

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \times t}{N}$$

where, ω_{max} and ω_{min} are the initial and final values of the inertia weight, respectively and N is the maximum iteration number. Typically, parameters ω_{max} and ω_{min} are set to 0.9 and 0.4, respectively.

A PSO algorithm is initialized with a group of random particles x_i. Each particle x_i has a corresponding fitness function value which is evaluated by the observation model $f(x_i)$. After the tth iteration, the fitness value of each particle is evaluated by a predefined observation model as follows:

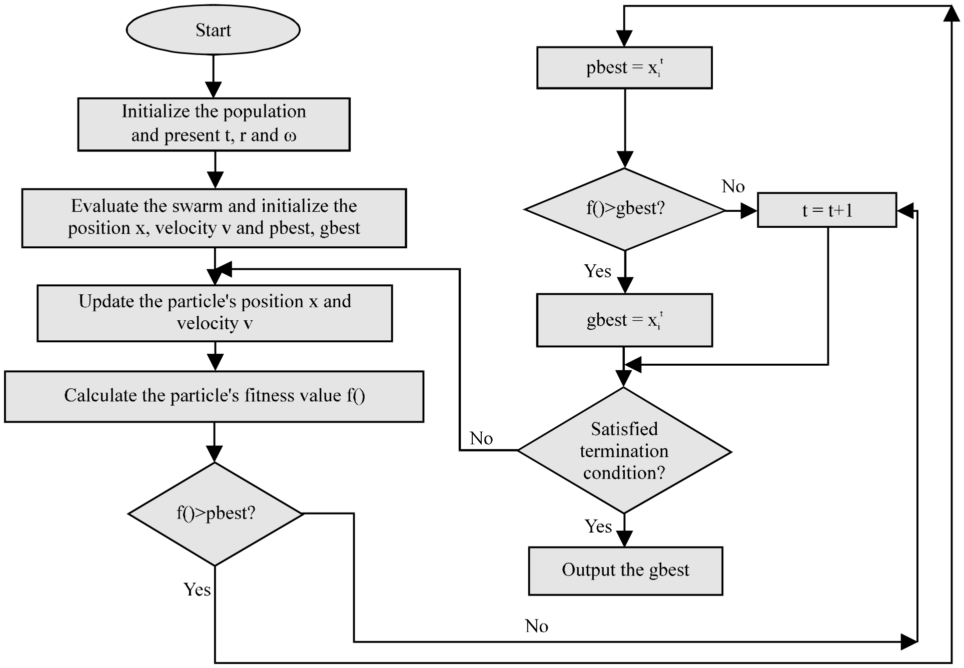

Fig. 3: Flowchart of fitness function based-PSO

$$f(x_i^t) = p(o_i^t \mid x_i^t)$$

where, o_i^t is the observation corresponding to the state x_i^t. Then the individual best and neighboring best of particles are updated in the following equations:

$$p_{id} = \begin{cases} x_{id}^t, & \text{if } f(x_{id}^t) > f(p_{id}) \\ p_{id}, & \text{otherwise} \end{cases}$$

Initialize $p_{gd} = \max(p_{id})$ firstly, For each particle k is the neighborhood of i, if $f(x_{kd}^t)>f(p_{gd})$ then $p_{gd} = x_{id}^t$. In this way, the particles search for the optima through the above iterations until convergence.

Due to problems in PSO like premature convergence in later generations, PSO by using fitness function is proposed for task scheduling and resource allocation. This PSO based fitness function is used to improve the resource utilization while try to find optimal schedule for given task set in the cloud environment.

SIMULATION

The parameters of total completion time and total processing capacity are taken into account to evaluate the performance of task scheduling and resource allocation in cloud computing.

The completion time (from the task submitted to the completion) serves as the evaluation criterion. Suppose the total completion time T_{total_i} on a resource in cloud computing includes total receiving time ΣT_{race}, total processing time ΣT_{exec} and total waiting time ΣT_{wait} of each subtasks and the total completion time can be denoted as equation $T_{total\_i} = \Sigma T_{rece}+\Sigma T_{exec}+\Sigma T_{wait}$ The makespan is denoted as the maximum value of total completion time of each task which is shown as follow: Makespan = $\text{Max}(T_{total\_i})$, where i = 1, 2,..., m and m is the number of given available resources.

The performance index of cost, in general, refers to the rent money per-time in business transactions, also is equivalent to the index of processing capacity while evaluating the performance of algorithm. The total processing capacity on cloud resource defined as the processing data volume each unit time, also denoted as C_{total}. Simulation program is adapted to CloudSim platform (Buyya *et al*., 2009) which is a simulation software of cloud computing. Simulation is performed on 4-task and 5-resource task scheduling and resource allocation in CloudSim, as described before section. In this work, iteration t, acceleration φ_1, φ_2 and the particles in swarm are preset as 20, 1.3, 1.3 and 300, respectively. For comparison, ant colony algorithm is also simulated on the same tasks.

Figure 4 shows that the total completion time, i.e., the makespan of PSO based fitness function algorithm is

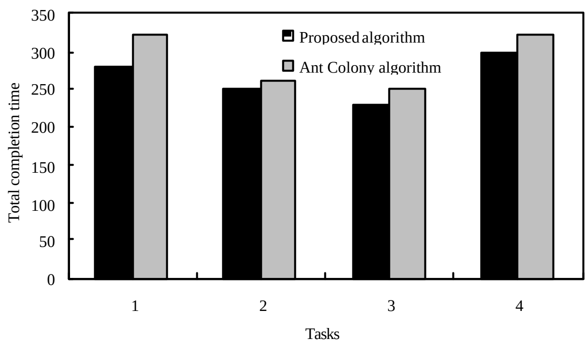

Fig. 4: Comparison of total completion time on each task

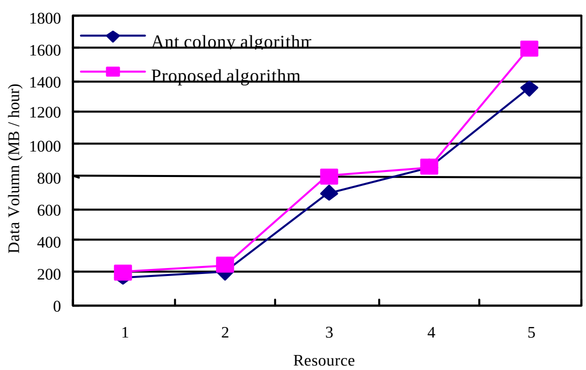

Fig. 5: Comparison of the processing capacity

lower than ant colony algorithm while performing on the same 4-task and 5-resource scheduling problem. Also, from Fig. 5, data volume per hour is more bigger while comparison with ant colony algorithm. This shows that the processing capacity of PSO based fitness function performs more effective. Thus, PSO based fitness function is more excellent performance with shorter makespan, greater computing capacity and lower cost.

CONCLUSION

In this study, PSO based fitness function is proposed to solve task scheduling and resource allocation in cloud computing. Cloud computing environment can offer dynamic and elastic virtual resources to the end users on demand basis. Task scheduling should satisfy the dynamic requirements of users and also need to utilize the virtual resources efficiently in cloud environment, so that task scheduling in cloud is an NP-Complete problem. Particle Swarm Optimization (PSO) based fitness function scheduling heuristic to balance the load across the entire system while trying to minimize the makespan and increase the processing capacity of a given task sets. The framework and Gantt chart for task scheduling and resource allocation is present to described problem. Finally, for comparison, ant colony algorithm is employed on the same tasks. Experimentation results show that PSO based fitness function is more effective and efficient.

ACKNOWLEDGMENTS

The authors would like to thank the support by the National Natural Science Foundation of China under Grant No. 61202136 and the Natural Science Foundation of Jiangsu Education Department under Grant No. 09KJD520011 and No. 12KJD520008, respectively. The work is also supported by Jiangsu Province Scholarships of Overseas Studies.

REFERENCES

Bhupendra, P. and R.K. Kapoor, 2013. Dynamic VM allocation algorithm using clustering in cloud computing. Int. J. Adv. Res. Comput. Sci. Software Eng., 3: 143-150.

Buyya, R., R. Ranjan and R.N. Calheiros, 2009. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. Proceedings of the 7th High Performance Computing and Simulation Conference, June 21-24, 2009, Leipzig, Germany, pp: 1-11.

Cai, J. and W.D. Pan, 2012. On fast and accurate block-based motion estimation algorithms using particle swarm optimization. Inform. Sci., 197: 53-64.

Chuanwen, J. and E. Bompard, 2005. A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimisation. Math. Comput. Simul., 68: 57-65.

Du, W. and B. Li, 2008. Multi-strategy ensemble particle swarm optimization for dynamic optimization. Inform. Sci., 178: 3096-3109.

Eberhart, R.C. and J. Kennedy, 1995. A new optimizer using particle swarm theory. Proceedings of the 6th International Symposium on Micro Machine and Human Science, October 4-6, 1995, Nagoya, Japan, pp: 39-43.

Eberhart, R.C. and Y. Shi, 1998. Evolving arti?cial neural networks. Proceedings of the International Conference on Neural Networks and Brain, October 27-30, 1998, Beijing, China, pp: 5-13.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proc. IEEE Int. Conf. Neural Networks, 4: 1942-1948.

Li, Y., P.P. Jing, D.F. Hu, B.H. Zhang and C.X. Mao *et al*., 2009. Optimal reactive power dispatch using particle swarms optimization algorithm based Pareto optimal set. Proceedings of the 6th International Symposium on Neural Networks, May 26-29, 2009, Wuhan, China, pp: 152-161.

Li, Y., Y. Cao, Z. Liu, Y. Liu and Q. Jiang, 2009. Dynamic optimal reactive power dispatch based on parallel particle swarm optimization algorithm. Comput. Math. Appl., 57: 1835-1842.

Ma, M. and L.B. Zhang, 2007. Particle Swarm Optimization Algorithm Design for Fuzzy Neural Network. In: Fuzzy Information and Engineering, Cao, B.Y. (Ed.). Springer, Berlin Heidelberg, pp: 309-314.

Zhang, Y., D.W. Gong and Z. Ding, 2012. A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. Inform. Sci., 192: 213-227.