IJCTA, 9(31), 2016, pp. 369-375
© International Science Press
ISSN: 0974-5572

Independent Task Scheduling Using Self Organizing Migrating Algorithm on Cloud Environment

Gomathi B.* and Karthikeyan Krishnasamy**

ABSTRACT

Cloud computing aspires to share a pool of virtualized resources. Task scheduling is an ineluctable issue in cloud computing. To map the tasks of users to virtualized resources, this paper proposes SOMA(Self Organizing Migrating Algorithm) algorithm to balance the load of the virtualized resources in the process of task scheduling. Based on simulation results produced by cloudsim, this algorithm can significantly reduce the makespan of task scheduling as well as perform the good load balancing compared to other scheduling algorithms like PSO,GA etc.

Keywords: Energy Consumption, Data delivery ratio, mobility, time, Congestion, Load Balancing, throughput and link quality, multipath routing and WSNs.

1. INTRODUCTION

Cloud computing system[1, 2] is a new paradigm of parallel and distributed system and it affords virtualized computing and storage technologies as a service to the end users by low cost on a demand basis. One of the main challenging issues in cloud environment is dispatching the tasks of users to virtualized resources in an efficient manner which belongs to task scheduling. Task scheduling is one of the problem in cloud computing. Since the task scheduling in cloud environment is hard to find optimal solution in polynomial time, it is said to be NP-Complete problem. Now-a day's numerous heuristic algorithms like GA, PSO, ACOetc are widely proposed by many researchers to resolve NP-Completeness of the scheduling problems. In order to utilize the cloud resources effectively, SOMA(Self Organising Migration Algorithm) is proposed for task scheduling in cloud environment. Further, the parameters of SOMA are redefined to make this algorithm more suitable for task scheduling problem. In the scheduling process, each task is allocated to suitable resource with minimum completion time over all available resources. So SOMA algorithm minimizes the makespan of assigned tasks and improves the performance by balancing the load among the virtual machines in cloud environment. To evaluate this proposed algorithm, an extensive cloud simulation platform was developed based on cloudsim[3] toolkit package. The experimental results show that the proposed SOMA significantly outperforms when compared to PSO and GA in terms of performance.The rest of the paper is organized as follows. Section 2 gives the related work. Section 3 presents the overview of the task scheduling model. Section 4 covers detailed description of the SOMA algorithm. Section 5 provides a simulation experiment and experiment analysis of the proposed algorithm. Section 6 gives the conclusion.

2. TASK SCHEDULING MODEL

In order to conform to the parallel and distributed cloud environment, we take over the dynamic batching mode, where tasks are collected in a set that is analyzed for mapping instead of mapping tasks into resources

* Assistant Professor, Department of IT, Hindusthan College of Engineering and Technology, Coimbatore, India, *Email: gomathi.babu@gmail.com*

** Professor, Department of IT, Karpagam College of Engineering and Technology, Coimbatore, India, *Email: karthiaish1966@gmail.com*

as they arrive. According to the collected information such as real status of resources and task details, we can design more reasonable scheduling strategy to balance the load across resources. In this paper, SOMA scheduling algorithm is proposed to improve the resource utilization in cloud environment.To formulate the task scheduling problem in the cloud environment, the set of n mutually independent tasks are represented as *Tj*, where *j* = {0, 1 … n-1} and set of m heterogeneous resources are represented as *Ri*, where *i* = {0, 1,…*m*-1}. Assume that the expected execution time *Pi*,*j* for task *j* on resource i is known. Furthermore, tasks are considered as non-preemptive. Each job's workload is measured by millions of instructions and the capacity of each resource is measured by MIPS. The following constraints give the assurance that only one task can be executed by processing resource at a time(constraint 2) and each task is allocated to exactly one processing resource(constraint 1). The permutation matrix entry is defined as,

$$X_{i,j} = \begin{Bmatrix} 1, \; if \;\; task \; j \; is \; assigned \; to \; resource \; i \\ 0, \; otherwise \end{Bmatrix} \tag{1}$$

$$\textstyle\sum_{i=1}^{m} X_{i,j} = 1, \; 1 \le j \le n \tag{2}$$

In this paper, we deem that cloud environment has heterogeneous resources with diverseprocessing capability. The processing time of task may show a discrepancy according to the task scheduling on different resources. The following metrics are used to measure the performance of proposed algorithm. The completion time of assigned tasks on resource *Rj* is,

$$C_j = \textstyle\sum_{i \in \{1 \ldots n_j\}} P_{i,j} * X_{i,j} \tag{3}$$

The makespan [11] of a schedule is the time for completion of the last task in the schedule.

$$\text{Minimize MS} = \max_{1 \le j \le m} C_j \tag{4}$$

The smallermakespan shows the better feat of the algorithm. Another metric is Load Balancing Index (β) which is computed to gauge the deviations of load on VMs as follows[7].

$$\beta = \sqrt{\frac{\sum_{i=1}^{m} Li - \bar{L}}{m}} \tag{5}$$

Where *m* is the number of virtual machines, *Li* is the load of the virtual machine *i* and $\bar{L}$ is the average load of all virtual machines. The smaller β shows the better load balancing in cloud environment. To augment the performance of task scheduling in cloud, we hub on balancing the load across VMs to minimize makespan in cloud environment.

3. SELF ORGANIZING MIGRATING ALGORITHM(SOMA)

SOMA[4] is a population based stochastic optimization algorithm which is modeled on self-organizing, competitive and co-operative social behavior of animals searching together to find the food source. By the haphazardinitialization, population is arbitrarily distributed over the search space and this algorithm has latent to converge towards the global optimum. Instead of engendering new individuals in search, positions of the individuals are changed based on two operations such as perturbation and migration. At each iteration, the population is weighed up and the individual with highest fitness is considered as Leader. The other individuals will traverse towards the leader in n steps of defined length.PRT (expanded as Perturbation) factor is used to perturb this path randomly. Priorto an individual proceeds towards leader, PRT vector whose range is (0, 1), is created. An individual moves towards a leader using the following equation:

$$X_{i,j}^{MLnew} = X_{i,j}^{ML} + \left(X_{L,j}^{ML} - X_{i,j}^{ML} \right) tPRTVector_j \tag{6}$$

Where,

t < 0, by Stepto, Pathlength >

ML is the actual migration loop

$X_{i,j}^{MLnew}$ is the new position of an individual

$X_{i,j}^{ML}$ is the position of an active individual

$X_{L,j}^{ML}$ is the position of leader

4. TASK SCHEDULING IN CLOUD USING SOMA:

This section imparts a SOMA which is having competitive-cooperative behavior of intelligent creatures that helps to solve the NP-complete problem. Each individual moves towards the best solution in the hyperspace of the optimized model. This kind of behavior of the individuals leads to successful search and maintain the diversity of the population. While applying SOMA algorithm to task scheduling problem, each individual in the population should map with one of the solution in the problem. For the crisis in *n* tasks, the dimension of the individual is n and content of each cell in the individual represents resource allocation to that task. After initializing the population randomly, SOMA is sustain the preliminary population throughout the migration process and only the positions of the individual are changed.Pseudo code for SOMA is given in Figure 1.

```
1.  Input the scheduling problem with n tasks and m VMs
2.  Output an optimized VM allocation to the specified task set
3.  Generate initial population randomly
4.  Evaluate individuals in the population using fitness function
5.  Select best individual as leader
6.  Create PRT vector for all individuals
7.  All individual except leader migrate towards the leader using equation(6) until individual reaches the final
    position given by pathlength
8.  For each step, fitness is evaluated at each position and move individual to the new position if it gives best fitness
    value
9.  If max iteration is reached or stopping criteria is satisfied, then stop. Otherwise repeat from step4
10. Report the best individual solution as optimal solution.
```

Figure 1: SOMA scheduling algorithm

5. EXPERIMENTAL EVALUATION

Cloudsim is a cloud simulation tool which is used to estimate the projected task scheduling algorithm. The pragmatic analysis is conducted on machine which runscloudsim, with intel Pentium dual-core processor 2.66 GHz, 4GB memory configuration and Cloudsim can be used to construct VMs as resources to persuade the user prerequisite by data center and cloudlets are crafted from the workload of high performance computing center called HPC2N[8]. Then, the cloudlets are premeditated with help of the proposed V-SOMA scheduling algorithm which is employed as part of cloud broker in cloudsim and judged against with well-known algorithms like PSO and GA algorithm using the following metrics: makespan and load balancing indexing. The Table 1 shows the parameter settings of the above three algorithms. Here, PSO[9] used constraint random inertia weight to boost the global searching knack.

This simulation experiment is used to scrutiny the above metrics for SOMA, PSO and GA algorithms with 50 to 300 tasks in 20 or 50 virtual machines respectively.

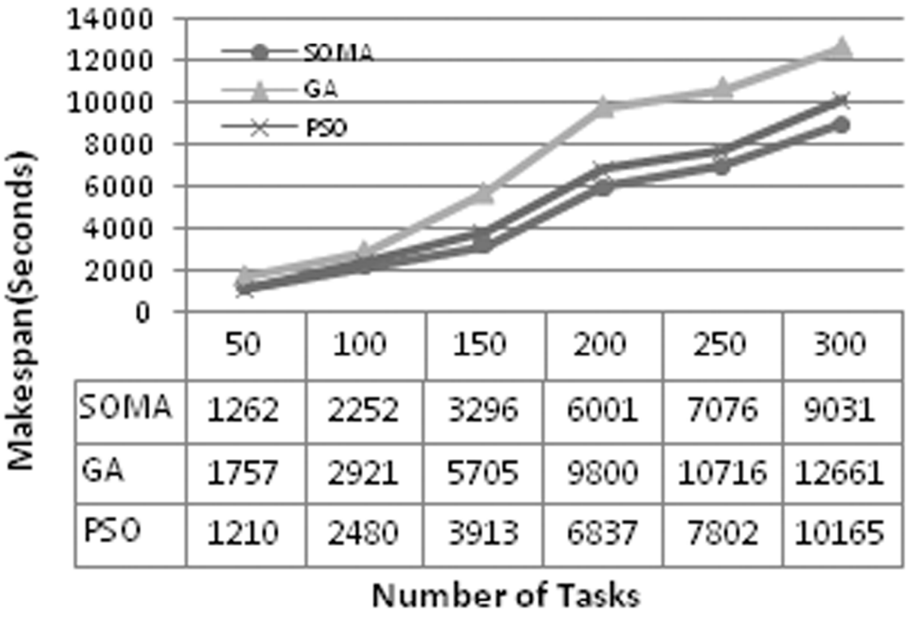
Table 1
Parameter settings for GA, PSO and SOMA algorithms

Algorithm	*Parameters*
GA	Population Size = 100Crossover probability = 0.7 Mutation probability = 0.2
PSO	Number of particles = 100acceleration coefficient c1 = c2 = 1.49
V-SOMA	Number of individuals = 100 Path length = 3 Step size = 0.23 PRT = 0.1

(i) Makespan:

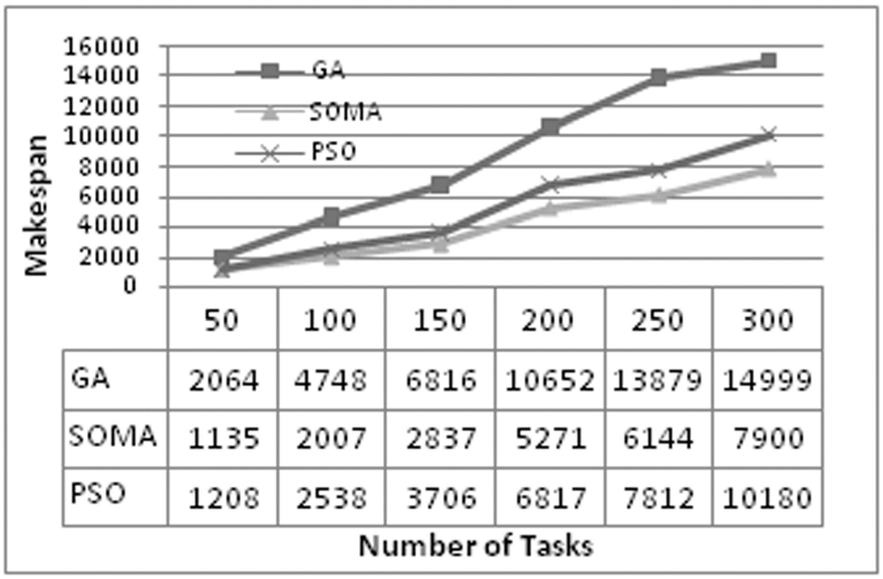
We ordeal the average of makespan of three algorithms by running simulation for 10 times with 50 to 300 tasks in 20 or 50 virtual machines as shown in Figure 2.

Figure 2shows that the makespanobtained from SOMA is lower than other two algorithms because competitive-cooperative behaviour of SOMA improves the individuals in the population. This leads to



	50	100	150	200	250	300
SOMA	1262	2252	3296	6001	7076	9031
GA	1757	2921	5705	9800	10716	12661
PSO	1210	2480	3913	6837	7802	10165

Figure 2(a): Average Makespan using 20VM



	50	100	150	200	250	300
GA	2064	4748	6816	10652	13879	14999
SOMA	1135	2007	2837	5271	6144	7900
PSO	1208	2538	3706	6817	7812	10180

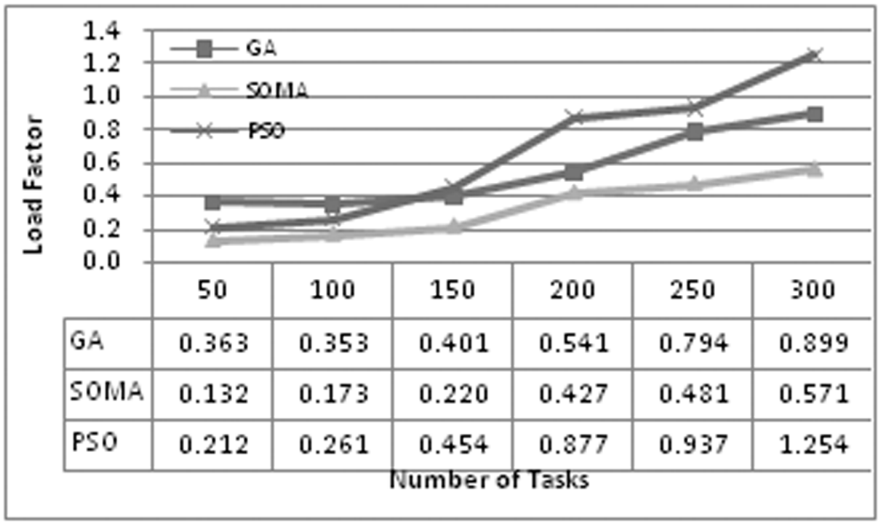
Figure 2(b): Average Makespan using 50VM

optimal resource allocation for all the tasks. Based on the observations, we can easily know their performance. For example, for small datasets, the convergence speed of SOMA is quite closer to other algorithms. However, as the number of tasks increases, competitive-cooperative behavior of SOMA has higher chance to find a better result than other algorithms because SOMA is able to avoid premature convergence by maintaining the diversity of the population.. Also, it can be shown that the Makespan is reducing with increase of VMs for the same number of tasks.

(ii) Load Balancing Index:

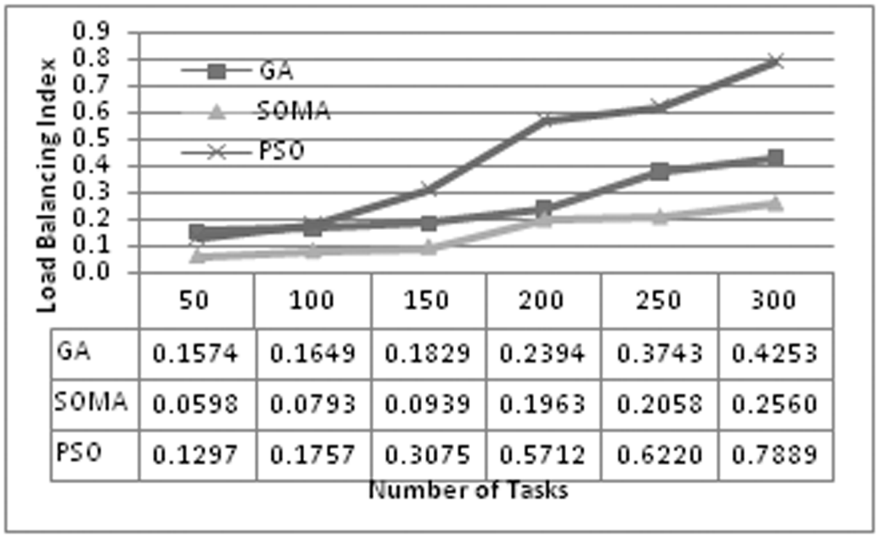
We examine the average of load balancing index of three algorithms by running simulation for 10 times with 50 to 300 tasks in 20 or 50 virtual machines as shown in Figure 4.

Figure 3demonstrates that load balancing index in PSO is highest because all the tasks were allocated to few optimal resources according to its scheduling strategy. This directs to load imbalance in the system. The load balancing index in the GA is quite closer to SOMA because they used modulus operator[10] to find resource allocation for each task which leads to proper load balancing amid set of resources. Also, it shows that load balancing index is reducing with the increase of VMs for the same number of tasks.



	50	100	150	200	250	300
GA	0.363	0.353	0.401	0.541	0.794	0.899
SOMA	0.132	0.173	0.220	0.427	0.481	0.571
PSO	0.212	0.261	0.454	0.877	0.937	1.254

Figure 3(a): Average of Load Balancing using 20VM



	50	100	150	200	250	300
GA	0.1574	0.1649	0.1829	0.2394	0.3743	0.4253
SOMA	0.0598	0.0793	0.0939	0.1963	0.2058	0.2560
PSO	0.1297	0.1757	0.3075	0.5712	0.6220	0.7889

Figure 3(b): Average of Load Balancing using 50VM

(iii) Convergence Rate:

We analyze the convergence rate of three algorithms by running simulation with 150 tasks in 20 virtual machines at 2000 iterations as shown in Figure 4.

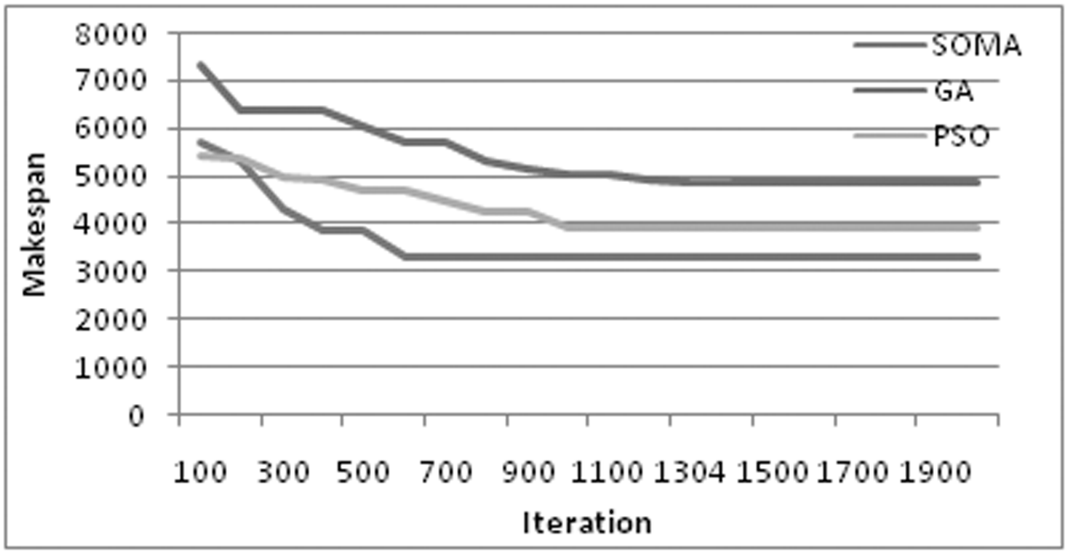


Figure 4: Convergence Analysis

The above figure illustrates that SOMA converges faster thanGA and PSO. However, GA and PSO were getting struck into local optimum, but mutation strategy in SOMA gives the better and quicker convergence as well as reduces the makespan as compare to other two algorithms.

In addition, competitive-cooperative behavior in SOMA used to avoid premature convergence as well as maintained the diversity of the population. It shows that the end results of SOMA are usually better than the other scheduling algorithms, especially for complex and large-scale problems.

6. CONCLUSION

The Self Organizing Migrating Algorithm (SOMA)is proposed to solve task scheduling problem in cloud environment. The task scheduling algorithm using SOMA considers two different objectives such as makespan and load balancing index. The mutation strategy and competitive-cooperative behavior in SOMA produces good trade-off between global exploration and local exploitation abilities that help to avoid premature convergence by maintaining the diversity of the population.The simulation results show that the proposed algorithm achieves better performance than GA and PSO for cloud computing systems.

REFERENCES

[1] G. Boss, P. Malladi, D. Quan, L. Legregni, and H. Hall, Cloud computing, *Technical Report, IBM* High Performance on Demand Solutions, 2007.

[2] Dikaiakos, M., katsaros, D., Mehra, P.,Vakali A., Cloud Computing: Distributed Internet Computing for IT and Scientific Research, *IEEE Transactions on Internet Computing*, pp. 10-13, 2009.

[3] Calheiros R.N., Ranjan R., Beloglazov A., De Rose CAF, Buyya R., CloudSim: a toolkit for modeling and simulation of Cloud Computing environments and evaluation of resource provisioning algorithms. Software: Pract Exp; 41(1): 23–50, 2011.

[4] I. Zelinka, *SOMA—self organizing migrating algorithm*, in New Optimization Techniques in Engineeri*ng*, B. V. Babuand G. Onwubolu, Eds., pp. 167–218, Springer, 2004.

[5] Mladenovic, N., and Hansen, P. :Variable Neighborhood Search. *Computers and Operations Research, pp.* 1097–1100, 1997.

[6] M. F. Tasgetiren, M. Sevkli, Y-C. Liang, G. Gencyilmaz, Particle swarm optimization algorithm for single-machine total weighted tardiness problem, Congress on Evolutionary Computation, 2004.

[7] B. Kruekaew and W. Kimpan, Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony, Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol I, 2014.

[8] HPC2N. The HPC2N Seth log; 2016. Available from: http://www.cs.huji.ac.il/labs/parallel/workload/l_hpc2n/index.html

[9] Hussein S. Al-Olimat, Robert C. Green, and MansoorAlam, Cloudlet Scheduling with Population Based Metaheuristics.

[10] L. Zhang, Y. Chen, R. Sun, S. Jing, and B. Yang. "A task scheduling algorithm based on PSO for grid computing". International Journal of Computational Intelligence Research, vol. 4, 2008.

[11] Gomathi B, KarthikeyanKrishnasamy. Task Scheduling Algorithm Based On Hybrid Particle Swarm Optimization in Cloud Computing Environment, Journal of Theoretical and Applied Information Technology, (2013), pp. 33-38.