International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: 6.614
Available Online at www.journalijcar.org
Volume 9; Issue 01 (A); January 2020; Page No.20905-20909
DOI: http://dx.doi.org/10.24327/ijcar.2020.20909.4095

Research Article

ENERGY AWARE VIRTUAL MACHINE SCHEDULING FOR CLOUD USING HEURISTIC MIGRATION

Dr.G.Ganesh Kumar[1] and Dr.M.Ramanan[2]

[1]Department of Computer Science and Engineering PARK College of Engineering and Technology Coimbatore, India
[2]Department of Physical Sciences and IT Tamilnadu Agricultural University Coimbatore, India

A R T I C L E I N F O

Article History:
Received 6th October, 2019
Received in revised form 15th November, 2019
Accepted 12th December, 2019
Published online 28th January, 2020

Key words:
The servers needs more energy consumption and money cost.

A B S T R A C T

Cloud computing provides resources as services into customers by using virtualization technology. As virtual machine (VM) is hosted on physical server, great energy is consumed by maintaining the servers in cloud data center. The servers needs more energy consumption and money cost. The energy efficiency is that challenge and our method will provide a promising approach for cloud hardware and also will be able to run on a similar physical server even while having different resources. The scheduling policy helps in proper and efficient utilization of Virtual Machine's (VMs). Both the Heuristic and the metaheuristic-based techniques have proven to have achieved some near-optimal solutions in a reasonable time frame. We analyse the current situation of cloud computing and introduce SFLA in resource allocation. To aiming that shuffled frog algorithm is easy to fall into local optimum with fast convergence speed into the subgroups of shuffled frog leaping algorithm. the number of migration and the consumption of energy than that of the previous work that is based on the particle swarm optimization (PSO) algorithm The results that shows the total simulation time (s) taken by the cloud data center when the actual number of VMs is 100 using the SFLA is less and it achieves much improved performance than the mechanism using PSO by about 18.6%.

INTRODUCTION

Cloud computing is a recent tread topic and is developing quickly. The applications delivered as services over the Internet and the hardware and software both are refer in the data centres that provide those services. Cloud computing provides computation and storage as services, which are made available as subscription-based services in pay-as-you-go mode to customers. The three major type of services are divided to: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS).

The virtualization technology further makes cloud computing different from the traditional grid computing or cluster computing paradigms, and also makes cloud computing more suitable in the commercialization. Virtualization provides promising approach to re-divide the hardware and software resources of one or more physical servers into multiple parts and each part runs in independent environment. Through creating isolated virtual machines (VMs) for different applications, multiple applications are able to run on the same physical server even with different resources. That is, the size of CPU, memory, and, other resources in VMs can be configured according to the time-varying demand of customers. This way, the cloud resources are more efficiently used.

Corresponding author:* **Dr.G.Ganesh Kumar
Biochemistry, Mahatma Gandhi Medical College & Hospital, Jaipur

As virtualization is an important technology in cloud computing, the VM placement (VMP) problem has become a significant research topic in cloud computing. VMP is to find an optimal map to place the VMs to physical servers so as to make the cloud resources used efficiently. As VMs are running on physical servers, great energy is consumed by maintaining the servers in data center and much cost is paid for cooling down the facilities. In this sense, more physical servers means more energy consumption and more money cost. Therefore, a promising scheduling purpose in the VMP problem is to effectively use the physical resources to host the virtual resources, so as to reduce the number of running physical servers.

Problem Statement

Bin packing is that classical combinatorial problem in optimization that has been extensively investigated. The problem of determining optimal VMs allocations is Non-deterministic Polynomial (NP)-complete, and getting optimal resolution of VMs allocations is often computationally infeasible when the cloud computing has multiple hosts and customers. A lot of researches have been done in energy efficient VMs allocations in cloud data center. The heuristic algorithms are dependent on the problem and they try to find the solutions by applying problem features in a complete way. Their solution is based on learning and exploration in which a comprehensive and scientific search for finding an optimal response and speeding to response process is applied.

However, they are very greedy and they are usually trapped in local optimal, moreover, they may defeat in getting widespread optimal solution. This work proposes a hybrid optimized with PSO and SFLA for energy efficient cloud computing.

Related Work

Particle Swarm Optimization (PSO) Algorithm

PSO is a self-adaptive global search optimisation technique introduced by Kennedy and Eberhart. The algorithm is similar to other population-based algorithms like Genetic Algorithms (GA) but, there is no direct combination of individuals of the population. PSO is a swarm-based intelligence algorithm influenced by the social behaviour of animals such as a flock of birds finding a food source or a school of fish protecting them from a predator. A particle in PSO is analogous to a bird or fish flying through a search (problem) space. The movement of each particle is coordinated by a velocity which has both magnitude and direction. Each particle position at any instance of time is influenced by its best position and the position of the best particle in a problem space. The performance of a particle is measured by a fitness value, which is problem specific.

The PSO algorithm is similar to other evolutionary algorithms. In PSO, the population is the number of particles in a problem space. Particles are initialized randomly. Each particle will have a fitness value, which will be evaluated by a fitness function to be optimized in each generation. Each particle knows its best position pbest and the best position so far among the entire group of particles gbest. The pbest of a particle is the best result (fitness value) so far reached by the particle, whereas gbest is the best particle in terms of fitness in an entire population. The particle will have velocity, which directs the flying of the particle. In each generation the velocity and the position of particles will be updated as (1 & 2):

$$v_i^{k+1} = wv_i^k + c_1 rand_1 \times (pbest_i - x_i^k) + c_2 rand_2 \times (gbest - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

Where v_i^k velocity of particle i at iteration k, v_i^{k+1} velocity of particle i at iteration k + 1, W inertia weight, c_j acceleration coefficients; j = 1, 2, $rand_i$ random number between 0 and 1; i = 1, 2, x_i^k current position of particle i at iteration k, $pbest_i$ best position of particle i, gbest position of best particle in a population, x_i^{k+1} position of the particle i at iteration k + 1.

Meta-Heuristic VM-Migration

The VM migration is defined as the movement of the VM from one host to that of another. The time taken for this includes the time taken for migrating among the virtual machines in cloud computing. Therefore, in order to be able to achieve an ideal load balancing, a cloud data center has to migrate dynamically, and also deploy a virtual machine in order to meet the needs of the users without a disruption to service. There are many more techniques in migration available for migrating the virtual machine such as the pre-copy and the post-copy, the adaptive compression, checkpoint recovery, replay method and the Least Recent Used (LRU).

For the process of bin packing, there may be a bin that is generally *opened* at the time it actually receives its first item. As soon as all the items in the bin depart, it gets *closed*. During this time, the total size of the active items that are in the open bin is referred to as the *bin level*. Every time there has been a new item added to this, either one or more open bins will accommodate this item. A First-fit places the item in the bin that is opened first and if no bin is able to accommodate, a new bin is opened.

This scheme of the MS is outlined as follows

Step 1: Begin with the initial (the current) solution x that is known as a configuration
Step 2: Evaluate the function criterion for the solution.
Step 3: After this, follow another set of candidate moves known as the neighbourhood N (x) of a current solution which is x.

In case the best among these moves if not the Migration (not in a ML) or in case the best is the ML, it can satisfy the criterion of aspiration. After this, the move is picked and considered to be a new current solution At the time the length of the ML actually reaches the size, the first solution will be freed from the MS and there is a new solution entering the list. This process will continue and the ML will act as its short-term memory. By means of recording the search history, the MS will control the actual direction that follows the searches.

Let S_b denote the best solution obtained in ML¸. The algorithmic description of the VM-Migration search can be summarized as follows:

1. *Initialize.* Generate an initial solution x. And let S_b = x. k = 1, ML = ϕ.
2. *Generate candidate set.* Randomly pick out a certain number of solutions from the neighborhood of x to form the candidate set N (x).
3. *Move.* (a) If N (x) = ϕ, Go back to step 2 to regenerate the candidate set. Otherwise, find out the best solution y in N(x). (b) If y $\in$ ML, i.e., it is VM-Migration search, and y does not satisfy the aspiration criterion, let N (x) = N (x) – {y}. Then go to 3(a). Otherwise, let x = y. And let S_b = y if y is better than S_b.
4. *Output.* If termination condition is satisfied, stop and output the S_b. Otherwise, let ML = ML $\cup$ {x}. (Add the new solution to the tail of ML. And if the length of ML exceeds a predefined size, remove the head item of the list.). Let k =k + 1 and go back to step 2.

In this section, it have simulated a data center comprising 100, 300, 500 VM. Each VM is modeled to have one CPU core with the performance equivalent to 1000, 2000 or 3000 MIPS, 4 GB of RAM and 512 Mb storage. Tables 1 to 3 and figures 2 to 4 shows the total simulation time (sec), energy consumption (KWH) and the number of migration respectively.

Table 1 Total Simulation Time for VM Search

Total simulation time (second)	100 VM	300 VM	500 VM
VM Search	771	1076	1442
PSO	814	1233	1866

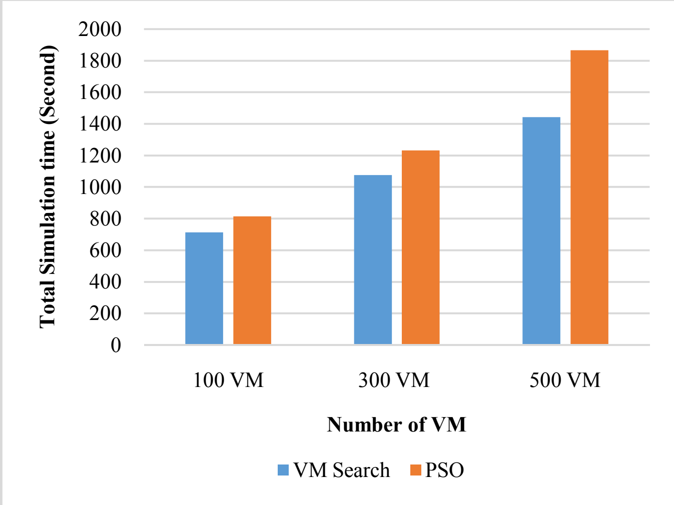

Figure 1 Total Simulation Time for VM Search

From the figure 1, it can be observed that the VM Migration Search has lower total simulation time by 7.59% for PSO when compared with 100 VM. The VM Migration Search has lower total simulation time by 11.96% for PSO when compared with 300 VM. The VM Migration Search has lower total simulation time by 4.39% for PSO when compared with 500 VM.

Table 2 Energy Consumption for VM Search

Energy Consumption KWH	100 VM	300 VM	500 VM
VM Search	3.2	4.9	7.5
PSO	3.8	5.5	8.9

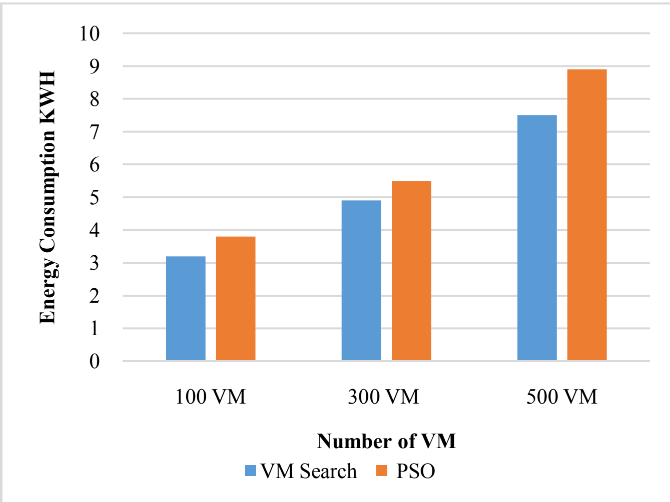

Figure 2 Energy Consumption for VM Search

From the figure 2, it can be observed that the VM Migration Search has lower energy consumption by 5.27% for PSO when compared with 100 VM. The VM Migration Search has lower energy consumption by 4.17% for PSO when compared with 300 VM. The VM Migration Search has higher energy consumption by 6.29% for PSO when compared with 500 VM.

Table 3 Number of Migrations for VM Search

Number of Migrations	100 VM	300 VM	500 VM
VM Search	46	78	109
PSO	65	94	136

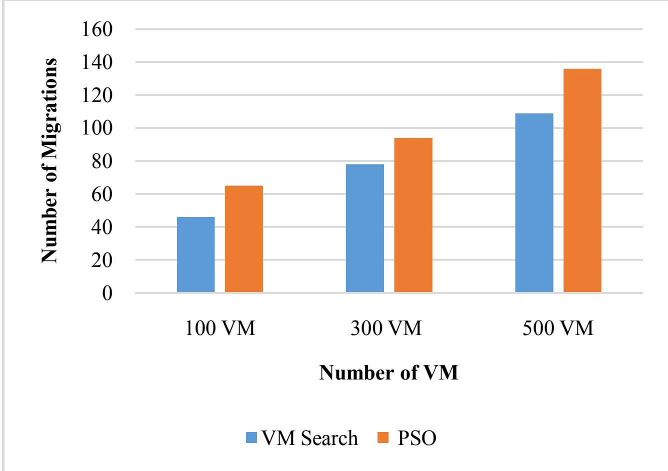

Figure 3 Number of Migrations for VM Search

From the figure 3, it can be observed that the VM Migration Search has lower number of migration by 8.19% for PSO when compared with 100 VM. The VM Migration Search has lower number of migration by 9.14% for PSO when compared with 300 VM. The VM Migration Search has lower number of migration by 7.39% for PSO when compared with 500 VM.

Proposed Work for Meta Heuristic SFLA Algorithm

The SFLA is a meta-heuristic optimization method which is based on observing, imitating, and modeling the behavior of a group of frogs when searching for the location that has the maximum amount of available food. The most distinguished benefit of SFLA is its fast convergence speed. The SFLA combines the benefits of the both the genetic-based Memetic Algorithm (MA) and the social behavior-based PSO algorithm. SFLA is a population based random search algorithm inspired by nature memetics. In the SFLA, a population of possible solution defined by a group of frogs that is partitioned into several communities referred to as memeplexes. Each frog in the memeplexes is performing a local search. Within each memeplex, the individual frog's behavior can be influenced by behaviors of other frogs, and it will evolve through a process of memetic evolution. After a certain number of memetics evolution steps, the memeplexes are forced to mix together and new memeplexes are formed through a shuffling process. The local search and the shuffling processes continue until convergence criteria are satisfied. The flowchart of SFLA is illustrated in varies steps are as follows:

Step 1: the SFLA involves a population 'P' of possible solution, defined by a group of virtual frogs (n).
Step 2: frogs are sorted in descending order according to their fitness and then partitioned into subsets called as memeplexes (m).
Step 3: frogs i is expressed as $X_i = (X_{i1}, X_{i2}, ..., X_{is})$ where S represents number of variables.
Step 4: within each memeplex, the frog with worst and best fitness are identified as X_w and X_b.
Step 5: frog with global best fitness is identified as X_g.
Step 6: the frog with worst fitness is improved according to the following equation (3 & 4).

$$D_i = rand()(X_b - X_w) \qquad (3)$$

$$X_{neww} = X_{oldw} + D_i(-D_{\max} \le D_i \le D_{\max}) \qquad (4)$$

Where rand is a random number in the range of [0, 1].

D_i is the frog leaping step size of the ith frog and D_{max} is the maximum step allowed change in a frog's position. If the fitness value of new X_w is better than the current one, X_w will be accepted. If it isn't improved, then the calculated (3) and (4) are repeated with X_b replaced by X_g. If no improvement becomes possible in the case, a new X_w will be generated randomly. Repeat the update operation for a specific number of iterations. Therefore, SFLA simultaneously performs an independent local search in each memeplex using a process similar to the PSO algorithm.

In this section, it have simulated a data center comprising 100, 300, 500 VM. Each VM is modeled to have one CPU core with the performance equivalent to 1000, 2000 or 3000 MIPS, 4 GB of RAM and 512 Mb storage. Tables 1 to 3 and figures 2 to 4 shows the total simulation time (sec), energy consumption (KWH) and the number of migration respectively.

Table 2 Total Simulation Time for SFLA

Total simulation time (second)	100 VM	300 VM	500 VM
VM Search	771	1076	1442
PSO	814	1233	1866
SFLA	690	980	1356

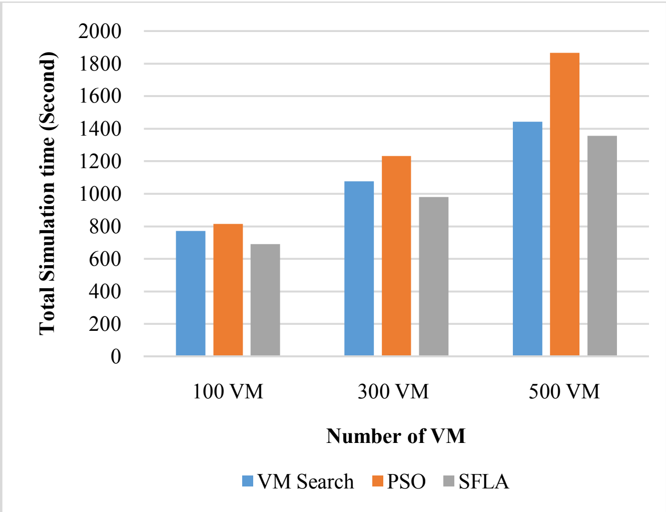

Figure 1 Total Simulation Time for SFLA

From the figure 1, it can be observed that the PSO has higher total simulation time by 3.59% for VM Search & by 11.62% for SFLA when compared with 100 VM. The TS has higher total simulation time by 5.16% for VM Search & by 9.15% for SFLA when compared with 300 VM. The PSO has higher total simulation by 14.39% for VM Search & by 13.57% for SFLA when compared with 500 VM.

Table 2 Energy Consumption for SFLA

Energy Consumption KWH	100 VM	300 VM	500 VM
VM Search	3.2	4.9	7.5
PSO	3.8	5.5	8.9
SFLA	3.0	4.6	7.2

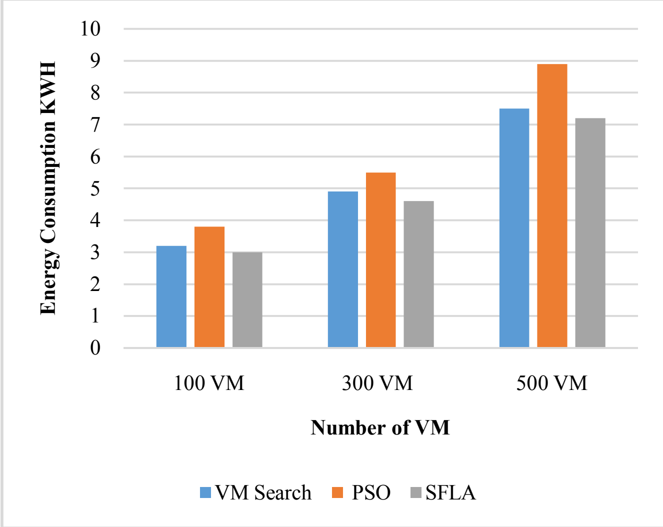

Figure 2 Energy Consumption for SFLA

From the figure 2, it can be observed that the PSO has higher energy consumption by 3.29% for VM Search & by 7.12% for SFLA when compared with 100 VM. The PSO has higher energy consumption by 3.16% for VM Search & by 4.14% for SFLA when compared with 300 VM. The PSO has higher energy consumption by 11.39% for VM Search & by 13.57% for SFLA when compared with 500 VM.

Table 3 Number of Migrations for VM Search

Number of Migrations	100 VM	300 VM	500 VM
VM Search	56	78	109
PSO	65	94	136
SFLA	52	71	98

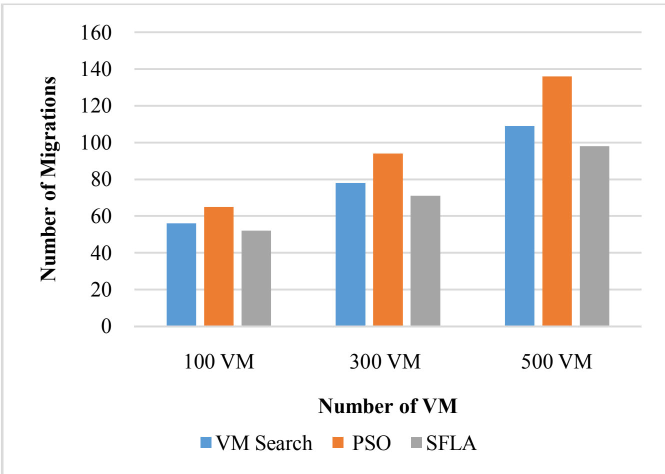

Figure 3 Number of Migrations for SLFA

From the figure 3, it can be observed that the PSO has higher number of migration by 3.29% for VM Search & by 7.12% for SFLA when compared with 100 VM. The PSO has higher number of migration by 3.16% for VM Search & by 4.14% for SFLA when compared with 300 VM. The PSO has higher number of migration by 11.39% for VM Search & by 13.57% for SFLA when compared with 500 VM.

CONCLUSION

Cloud computing technology is expected to grow and provide large services and computational power to end users. In this context energy efficiency is more important for virtualized data centers. In this work, the VM migration, first fit algorithm and TS algorithm is proposed. VM migration has proven to be a powerful technique for achieving a number of objectives, including workload consolidation, load balancing, reducing energy consumption, facilitating maintenance activities as well as supporting mobile applications. First-fit algorithm potentially assigns a VM to one of PMs with smaller identifiers. Therefore, the PMs with larger identifiers could be shut down and then the number of running PMs can be minimized. Results show that the MS has lower total simulation time by 2.55% for 100 VM, by 2.52% for 300 VM and by 2.75% for 500 VM when compared with first fit method.

In this work, focus on the VM with multi-resources allocation in cloud data center. A multi-resource energy efficiency VMs allocation model is proposed for PSO and SFLA. The SFLA is a meta-heuristic optimization method which is based on observing, imitating, and modeling the behavior of a group of frogs when searching for the location that has the maximum amount of available food. Results show that the SFLA has lower total simulation time by 2.87% & 17.8% for 100 VM, by 11.43% & 26.79% for 300 VM and by 4.44% & 29.33% for 500 VM when compared with TS and PSO method.

References

1. Zhang, H., & Sun, G. (2002). Feature selection using tabu search method. Pattern recognition, 35(3), 701-711.
2. Razali, R. A. M., Ab Rahman, R., Zaini, N., & Samad, M. (2014, June). Virtual machine migration implementation in load balancing for Cloud computing. In Intelligent and Advanced Systems (ICIAS), 2014 5th International Conference on (pp. 1-4). IEEE.
3. Zhang, Y., & Wu, L. (2012). Tabu Search Particle Swarm Optimization used in Cluster Analysis. *Journal of Science,* 1(1), 6-12.
4. Lin, M. J., Luo, F., Xu, Y. G., & Luo, L. (2013). A Novel Hybrid Optimization Method of Shuffled Frog Leaping Algorithm and Particle Swarm Optimization. In Advanced Materials Research (Vol. 717, pp. 433-438). Trans Tech Publications.
5. Chen, S. (2015). Resource Scheduling Scheme Based on Improved Frog Leaping Algorithm in Cloud Environment. METALLURGICAL AND MINING INDUSTRY, 7(8), 436-441.
6. Pandey, S., Wu, L., Guru, S. M., & Buyya, R. (2010, April). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In Advanced information networking and applications (AINA), 2010 24th IEEE international conference on (pp. 400-407). IEEE.
7. Tesfatsion, S. K. (2018). Energy-efficient cloud computing: autonomic resource provisioning for datacenters (Doctoral dissertation, Umeå University).
8. Ghribi, C. (2014). Energy efficient resource allocation in cloud computing environments (Doctoral dissertation, Institute National des Télécommunications).
9. Luo, J. P., Li, X., & Chen, M. R. (2014). Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers. Expert Systems with Applications, 41(13), 5804-5816.
10. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M.Zaharia, “A View of Cloud Computing”, in Proc. Communications of the ACM, vol. 53 no. 4, pp. 50-58 Apr. 2010.
11. R. Buyya, S. Pandey, and C. Vecchiola, “Cloudbus toolkit for market-oriented cloud computing,” in Proc. 1st International Conference on Cloud Computing, 2009, pp. 24–44.
12. Y. Q. Gao, H. B. Guan, Z. W. Qi, Y. Hou, and L. Liu, “A multiobjective ant colony system algorithm for virtual machine placement in cloud computing,” *Journal of Computer and System Sciences*, vol. 79, no. 8, pp. 1230-1242, Dec. 2013.
13. G. International. “Make it green: Cloud computing and its contribution to climate change”, 2010.
14. M. Stillwell, D. Schanzenbach, F. Vivien, and H. Casanova. “Resource allocation algorithms for virtualized service hosting platforms”, *Journal of Parallel and Distributed Computing*, vol.70, no. 9, pp. 962-974, September 2010.
15. S. Chaisiri, B. S. Lee, and D. Niyato, “Optimal virtual machine placement across multiple cloud providers”, in Proc. IEEE Asia-Pacific Services Computing Conference, 2009, pp. 103–110.
16. Y. Ajiro and A. Tanaka, “Improving packing algorithms for server consolidation”, in Proc. International Conference for the Computer Measurement Group (CMG), Computer Measurement Group, 2007.

How to cite this article:

Dr.G.Ganesh Kumar and Dr.M.Ramanan (2020) ' Energy Aware Virtual Machine Scheduling for Cloud Using Heuristic Migration', *International Journal of Current Advanced Research,* 09(01), pp. 20905-20909.
DOI: http://dx.doi.org/10.24327/ijcar.2020. 20909.4095

****