ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Simulated Annealing Load Balancing Algorithm for Cloud Data Centres through Ant Colony Optimization

[1] Anu V.R., [2] Elizabeth Sherly
[1] Research scholar, M.G.University, [2] Professor, IIITMK

Abstract: - **Load balancing optimization strategies on cloud data centers are active research area in cloud environment. Through energy efficient load balancing techniques, VMs can packed in to fewer number of servers and reduce the power dissipation and CO2 foot print. However allocating too many VMs into a physical serer can cause interference issues, hotspot and SLA violations. Here we present a simulated annealing load balancing strategy through ant colony optimization to keep a strong tradeoff between these above mentioned issues and provide an energy efficient, SLA guaranteed load balancing algorithm to map VMs effectively between physical servers. Experimental results show that the proposed algorithm simulated annealing through ant colony optimization performs well and achieve better load balancing results as compared with other algorithms.**

***Keywords*: Ant colony Optimization, energy efficiency, load balancing, simulated annealing algorithm, virtual machine.**

I. INTRODUCTION

Cloud computing and data centers have been an inevitable part of our daily life because of huge use of internet services and online applications. As per United States environmental protection agency, energy used by datacenters and federated servers are about 100 billion kWh and more. This electric energy demand will increase by more than 66% on the period 2015- 2035[1].This high energy consumption by data centers and CO2 footprint forced the research community to think and develop in terms of energy efficiency systems for modern cloud data centers. Energy favorable designs would help large energy savings in servers and potentially improve their efficiency in real time use. Energy efficiency has become a significant paradigm for server and data center applications focusing on the reduction of all energy related operational cost as well as capital cost. Virtualization helps workloads to be dynamically allocate resources in the form of virtual machines. These VMs can be relocated between set of physical servers through migration. As far as cloud data center are considered the lowest energy efficiency region corresponds to their load balancing mechanisms including migration process. Besides the server's routine operational cost, total cost of ownership (TCO) includes other energy dependent components such as cost of energy requited for cooling infrastructure and cost of mechanisms for reduction of CO2 footprint[15,16]. The unique factors affecting the performance of virtual machine migration and thus energy efficiency of these operations are migration link bandwidth, total migration time, down time and rate at which dirty pages are created and transferred during migration process. The amount of extra energy required to migrate a VM changes based on the kind of workload executed in that VM. Sometimes the naïve migration strategies may result increased power consumption while performing migration at cloud datacenters. Such kind of migrations should be avoided. Enhancing energy efficiency in data centers can be resolved by applying proper VM allocation strategies and through suitable VM load balancing methods at the same time. Moreover, turning off idle PMs by performing server consolidation can also constitute a solution to the energy efficiency issue. As per [2] all algorithms which are trying to achieve energy efficiency must find out optimum solution or a better tradeoff for the following queries.1) determine when the PM is overloaded and procedure selected to manage the overloading situation.2) determine when a host is considered as under loaded and strategy selected to manage under loaded condition. 3) Selection of VMs from PM to overcome overload condition.4) Selection of migration strategy to move VMs from overloaded PMs to under loaded PMs.

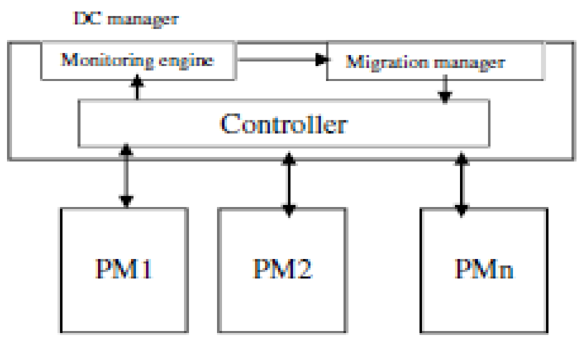

Fig. 1: Data center resource allocation model

According to [1] the PM load balancing strategy can be categorized into different groups such as a) based on time of decision making b) based on parameters considered c) based on optimization method used d) selection of objective function and e) as per evaluation method. Here we presented an energy efficient load balancing system based on optimization model. The optimization model can again be classified as per the nature of algorithm selected such as exact method, heuristic method and meta heuristic method. The algorithm we consider here is simulated annealing and it comes under meta heuristic optimization model [17, 18]. Compared with other optimization models meta heuristic models are problem independent techniques. These algorithms effectively guide the space search process to find optimum solution. But sometimes these algorithms took more time than heuristic solutions. In order to overcome that issue here we combine simulated annealing technique with ant colony optimization. As compared with other meta heuristic algorithms such as genetic algorithms and particle swarm optimization algorithm, simulated annealing with ant colony optimization algorithm performs well with load balancing strategy [19, 20]. The rest of paper is organized as follows, section 2 narrates the related literature connected with our proposed algorithm. Section 3 explains the system model of simulated annealing and Ant colony Algorithm with proposed SAACO algorithm. Section 4 presents the experiment setup and analyze the result through various graph. Section 5 concludes the paper.

II. LITERATURE SURVEY

Even though many algorithms are developed, load balancing in cloud environment is still a challenging research area. Ankita Chaudhary et.al [2] provided a critical analysis on energy efficient dynamic allocation of VMs in cloud data centers. They also proposed a technique for VM allocation using dynamic threshold values ensuring a deadlock free resource allocation focusing on multidimensional resources thus reducing energy consumption of data center. In [1] Varasteh et.al narrated a detail survey on server consolidation in data centers for reducing energy consumption.

Zonggin Fan et.al [3] presented a simulated annealing algorithm for solving resource mapping and scheduling problem in cloud environment. But in this, finding out neighboring solution is not in an optimized way. In our proposed algorithm we overcome that issue by introduction of ACO. Leonardo P.Cardoso et.al [4] proposed an elastic and energy aware virtual machine migration technique that reduce power consumption by turning off idle physical machine and turn them on as per demand. Here they use FITS (future internet test bed with security) as management mechanism. Here we planned to introduce an efficient load balancing strategy which will reduce the number of active PMs by balancing the load efficiently. The idle PMs can then turned off and this will improve energy efficiency of the entire system.

Tian et.al propose a simulation and modelling tool kit for scheduling virtual machines on cloud datacenters [5].In their work they have introduced a predictor of VM workloads and implemented it in simulated environment. The key idea is to provide a tool kit to help administrators on the decision making process on resource allocation for VM demands. In [6] Ilksen Caglar et.al introduced an efficient resource management policy for virtualized data centers which optimizes the number of resources required to meet dynamic workloads without any migration. Here they used an experiment based prediction module. Energy performance trade off can be achieved by comparing predicted and active number of servers periodically. Dilawaer Duolikun et.al proposed a migration approach which will reduce the electric energy consumption in data center [7]. The key idea of this mechanism is that VMs migrate from source PM to destination PM only when the guest server expected to consume smaller electric energy than the host server. They proposed the energy aware migration of virtual machine algorithm to implement the concept.

Debabrota Basu et.al [8] presented a reinforcement learning algorithm called Megh, for live migration of VMs that reduces the cost of energy consumption and enhance the performance. The algorithm analyses the uncertain dynamic workloads and a user dimensionality reduction scheme to project the space-action space to a polynomial dimensional space.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Jinhin Yang [9] introduce extended ACO method to solve GTSP problem. They introduced a mutation process to avoid locking into local minima. They also added a local searching technique namely 2opt search. Guan et.al [10] presented a meta heuristic algorithm based on ACO for embedding VMs into PMs in an energy efficient way. To reduce the space complexity they incorporated special methods that tracks and updates pheromone trails in this work. In [11] Plakunov et.al proposed a data center resource mapping algorithm through ACO optimization approach. In this algorithm they map VMs, virtual storages and virtual channels through ACO method and not bound to a certain data center network topology.

III. SYSTEM MODEL

In this work we present a simulated annealing (SA) strategy to manage the physical servers in a data centre with optimum load and energy efficiency [3]. Whether it is static or dynamic load balancing, traditionally a centralized management module is used to distribute resources through virtualization in cloud environments. In simulated annealing method, the key idea is that it curtails the standard deviation of hosts' load and keep entire system load in harmony [12]. One major characteristic for resource scheduling and load balancing in cloud computing environment is its target selection and how effectively it has to be reached from source server. Here in this work we use simulated annealing (SA) technique for target selection and ant colony optimization (ACO) technique to optimize the load balancing process with least energy and cost. Generally, VM load balancing mechanisms end up with large granularity and VM migration process of load balancing consume a major bandwidth for data communication [13, 14].

A. Simulated Annealing Load balancing strategy:

A typical distribution strategy of Virtual Machines (VMs) in physical servers of a cloud data centre are described in fig: 1.Each physical server can host one or more VMs and each VM can implement one or more application along with OS. Let$P = \{P_1\ P_2\ \dots\dots P_x\}$ be set of physical machines and Let$V = \{V_1\ V_2\ \dots\dots V_y\}$ be the set of virtual machines are required to be balanced in x number of PMs. Following are the assumptions to simplify the presentation such as the datacentre under consideration is homogeneous in nature and all PMS are of same computational as well as storage capacity .Through resource monitoring modules we can read the resource utilization of each VMs. Here we use a_{ij} to indicate the deployment of VMs and represented as,

$$a_{ij} = \begin{cases} 1 & \text{the } i^{th} \text{ VM is deployed in } j^{th} \text{ PM} \\ 0 & \text{Otherwise.} \end{cases} \quad (1)$$

Total workload of each PM can be calculated by adding the workloads of each VMs running in it .Let w_j be the load of jth machine belongs to P,

$$w_j = \sum_{i=1}^{x} V_i\, a_{ij} \quad (2)$$

To balance the resource utilization of physical machines and avoid SLA violation proper load balancing strategies has to be used. Let the residual capacity of a PM $P_j \in P$ be rc_j and rc′ denote the average residual capacities of all physical machines under consideration.

$$rc_j = ch - w_j, \quad rc' = \frac{1}{x}\sum_{i=1}^{x} V_i\, a_{ij} \quad (3)$$

$$\sigma\,(m) = \sqrt{1/x \sum_{i=1}^{x} (rc_j - rc')^2} \quad (4)$$

Where m is the mapping solutions and ch is the capacity of PM,
The objective of our approach is to figure out best mapping solution m so that the system will achieve best load balancing strategy.

$$\min\ C(m) = \sqrt{1/x \sum_{i=1}^{x} (rc_j - rc')^2} \quad (5)$$

Equation (2) follows two constraints such as each VM can be deployed only on a PM

$$\text{ie,} \quad \sum_{i=1}^{x} a_{ij} = 1 \quad (6)$$

and workload on a PM cannot exceed its capacity

$$w_j = \sum_{i=1}^{x} V_i\, a_{ij} \le ch \quad (7)$$

$$rc_j = ch - w_j, \quad rc' = \frac{1}{x}\sum_{i=1}^{x} V_i\, a_{ij}$$

$$a_{ij} = \{0,1\} \quad (8)$$
$$i \in x, j \in y$$

The SA scheduling algorithm is used when the search space is discrete. Initially an annealing table is constructed, such as (T_0, α, C, R), which contains the parameters to manage and control SA process. T_0 is the initial temperature, α is the temperature reduction parameter, C is increment counter and R is end of rule. This table assures the algorithm to return an acceptable optimal solution in stipulated time and to ensure global convergence and efficiency. The initial temperature T_0 has great influence on load balancing algorithm. It influences global convergence of SA directly, such as too low temperature can cause easy convergence to local minimum and too high temperature gives global solution

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

easily but took much time for calculation. The values of variables are set according to practical situation and given conditions.

Generally in SA algorithm the initial solution is selected at random .Here we use ant colony optimization strategy (ACO) to allocate VMs to physical machines in best possible way in less amount of time with no co located interference in PMs. While using this method the largest workload VM will be mapped to PM which has highest residual capacity. Allocation step continues until all VMs have been allocated in to the suitable PMs. At last we earned an initial solution m0, which can be used as input to the simulated annealing load balancing algorithm. Through this initial allocation, along with a reasonable initial solution the standard deviation of residual capacity in each PM σ (m0) is also small value. When a migration of VM occurs, a suitable neighbouring solution is developed from the simulated annealing neighbouring function. As the next step on current solution m, we choose to move the VM with highest resource utilization from the physical machine with least residual capacity to PM with most residual capacity to balance the workload among the PM. Ant colony optimization algorithm (ACO) is used to optimize the process of placing VMs into new assigned PM. This will give the new solution m'. At transfer the residual capacity of PM is recalculated and reordered the list.

The pseudo code of proposed SAACO algorithm is given below

```
1 begin
2       initialize:
3       T := T0;
4       m := m0;
Allocate VMs to Pi which has maximum residual capacity;
5 while T > Tf do
6       condition = false;
7             for C times do
8                   generate new solution m' from m;
        Call ACOextended();
9
        calculate Δf = rc(m') − rc(m) ;
10        if Δf < 0 or exp(−Δf/T) >   random(0, 1)
11
                  m=m';
12
            condition = true;
13                end if
14          end for
15      if condition==true
16                  T = T · α;
17      end if
18 end while
19end
```

```
ACOextended ()
  1. Begin
  2. find out maximum loaded VMs using,Vmax .
  3. Find out minimum loaded PMs using, Pmin;
  4. Call ACOmigration () to migrate Vmax toPmin;
  5. End
```

In pseudo code of SA, initialization steps from 2 to 4 construct the annealing initialization table with values for initial temperature T_0 and final temperature T_f . Also attain the initial solution m as random. Generate neighbouring function m' by calling the function ACOextended().This function maps VMs in descending order of workload to PMs with ascending order of available workload without interference issues . Meanwhile calculate rc() for m and m'. In step 9 calculate the metropolis condition. If$\mathrm{rc}(m') < \mathrm{rc}(m)$, m' can be set as initial condition for the successive simulations. On the other hand we select a random number ρ from (0,1) and if$\rho < (-(\mathrm{rc}(m') - \mathrm{rc}(m))/K.T)$, where K is a constant and T is current temperature .Then m' is set as the initial condition for coming simulations as per m=m' in line 11.Otherwise m' will be discarded and original solution is taken as initial solution for successive simulations. The above process will be repeated until the final value for the increment counter C under the current temperature. Now scale down the current temperature T by cooling ruler α and the new temperature is less than final temperatureT_f, Stop the process and return the optimum solutionm. Otherwise reset iteration counter and repeat the process from step 2.

Parameters	Values
T_0	100
T_f	0.01
A	.8
L	120

Table 1: Parameter values for SAACO

Pseudo code for Ant colony Optimization

ACOmigartion ()
1. Begin
2. Set timeC=0{ time counter is set}
   For every edge (i,j) between any two PM , set an initial τ_{ij} =c for trail density and $\Delta\tau ij = 0$
3. Set st=0{ travel set counter is set}
4. For a =1 to k do

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

a. Place ant(VM) a on a PM randomly and place that PM in the set called visited
b. Place the group of PMs in$tabu_a$.

5. Repeat until st<= m
   Set st=st+1
   For a=1 to k do
   Choose the next PM to be visited according to the possibility factor P_{xy} a given in equation ()
   Move the VM a to the next PM
   Insert the current selected PM to visited a
   Insert the group of selected PMs to $tabu_a$.
6. For a=1 to k do
   Move ant (VM) a from visited (n) to visited a (k).
   Compute the tour length la travelled by ant a.
   Update the shortest path length found.
   For every edge(x,y) do
   For a=1 to k do
   Update the pheromone trail density τ_{ij} according to equations () and ()
   Timec=timec+1
7. If time <(MAX_time) then
   Empty all visited and tabu k
   Go to step 2
   Else
   Get the shortest path to reach destination PM
8. End

B. Extended Ant colony optimization strategy

Here we use ACO optimization method to get next solution in simulated annealing method while mapping virtual machines to physical machines. Based on the observations from data centres it is clear that, the workload on VMs are totally unpredictable in nature. So a meta heuristic solutions more suited to this mapping optimization issues.

As the initial step of ACO we build a construction graph G (N, L)where N represents the physical nodes as vertices and L denotes the edge or link between any two nodes x and y in which the possibility of moving an artificial ant(VM) to the node y when it is in node x. This possibility or likelihood of mapping function is calculated based on heuristic and historical factor called pheromones. Pheromone content in the links would be updated depends on the energy usage of mapping solutions in previous iterations and ultimately evaporates with time.

A trace from nest ‘start‘ to food ‘end’ corresponds to an expedient solution to optimized mapping problem. In the graph representation artificial ant (VM) start its trail from source physical server nest start to destination server to the food end with minimum total cost towards energy consumption.

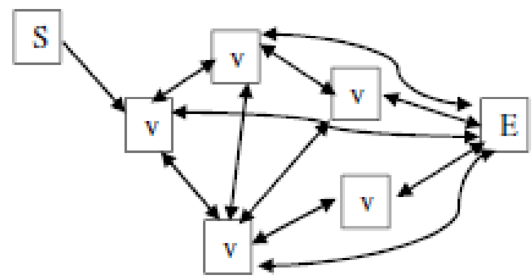

Fig. 2: Building graph for ACO model

Pheromone trails (τ_{xy}): The pheromone content between any two nodes x and y represented as the desirability of placing VM into corresponding PM. The initial value of pheromone (likelihood of mapping function) τ0 on each trail is very same while constructing a graph. With each iteration pheromone on each trail evaporate as,

$$\tau_{xy} = \max\tau_{0}\,(1-\xi)\tau_{xy} \tag{9}$$

Where ξ is the evaporation constant.

Heuristic function (τ_{xy}): It is the ratio between the minimum possible node energy consumption to the energy consumption for mapping the VM to current PM under consideration.

The possibility factor for mapping a VM to a particular PM is,

$$P_{xy} = \frac{(\tau_{xy}(t)^{\alpha}\acute{\eta}_{xy}(t)^{\beta})}{(\Sigma_{a\epsilon k}\,\tau_{xy}(t)^{\alpha}\acute{\eta}_{xy}(t)^{\beta})} \tag{10}$$

IV. EXPERIMENT SETUP &ANALYSIS

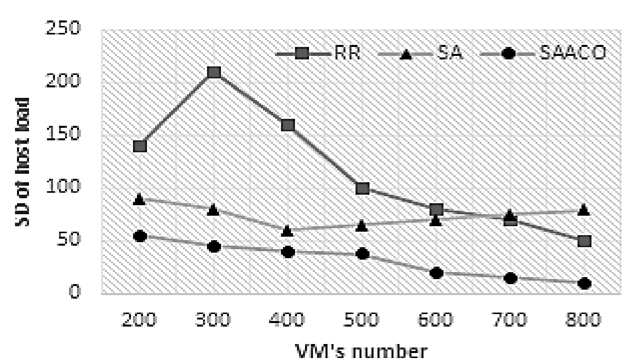

Fig 3: Average standard deviation of each algorithm

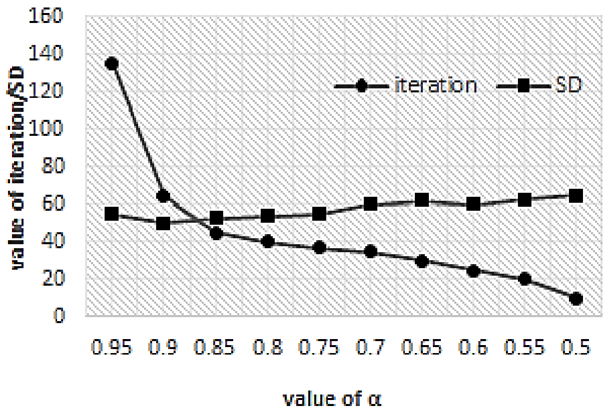

Fig 4: Iteration comparison between algorithms

We have used the extensible simulation tool kit called cloud-sim which enables to simulate the cloud computing environment and applications related to that.VM mapping is supported by cloud-sim at two levels namely host level and VM level. At host level it is possible to specify the amount of host processing power allotted to each VM and at VM level, a fixed amount of available processing power is assigned to VM for individual application services. The experiment is conducted with one datacenter and 200-800 Cloudlets. Each cloudlet is embedded to a VM, which is hosted on one of the 200 hosts under the simulation platform. The cloudlets' computation workload is considered from 100 MI (Million Instruction) to 200 MI and the hosts' capacity is 1500 MI. And time-shared manager types is selected for the datacenter Here we made the comparison between the proposed simulated annealing with ACO (SAACO) algorithm with round robin (RR) and simulated annealing (SA) for various load balancing loads. In RR method, normal RR policy is followed such as VMs are allocated from first to the last with a stipulated time quanta. In ordinary SA algorithm the initial solution is chosen randomly and stochastically generate a new solution. In the proposed algorithm SAACO, we incorporated ACO for generating neighborhood solutions and it improves effectiveness the mapping function. Here we compared the standard deviation caused by these three algorithms while performing VM mapping process [21, 22]. Fig. 3 exhibits the average degree of imbalance of each algorithms when the number of tasks varying from 200 to 800. As the increase of the number of the virtual machines occurs, the standard deviation of the host load decreases. The standard deviation shown by SA and RR are more than that of the SAACO because of randomness. From the figure it is also obvious that the average performance of the SAACO algorithm is better than the other two algorithms.

Fig.4 shows the number of iterations of SA and SAACO. Clearly, the iterations of SAACO is stable and less than the basic simulated annealing SA algorithm. This is because initial solution selection is done randomly with SA and gains the new solution stochastically. But in the proposed SAACO method we got a relatively better initial solution and obtained new neighbouring solution more effectively through ACO method, so it took less time to finish the search.

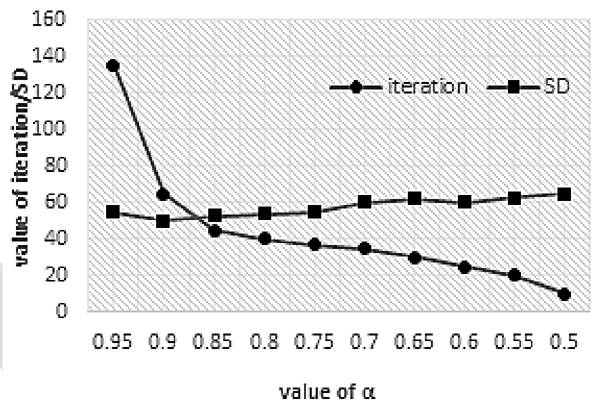

Fig 5: Analysis of parameter α

Fig.5 shows the influence of the parameter α in the temperature reduction function to the SAACO algorithm. If this parameter is close to 1 the temperature falls too slow, then the rate of convergence of the algorithm will be greatly reduced. From the figure it is clear that it reacts more as the iterations are more. If α is small the temperature falls too fast and the minimum point might be lost. So for our experiment we choose the value of α at 0.8.

V. CONCLUSION

In this paper we presented a meta heuristic approach for load balancing called Simulated Annealing with Ant Colony Optimization (SAACO). To improvise and get much better results for initial and neighbouring solutions for simulated annealing methodology, ant colony optimization method was used. The effectiveness of algorithm was evaluated by cloud-sim simulator and got better results for load balancing as compared with normal simulated annealing algorithm and round robin strategy. The optimization results show that our algorithm outperforms the greedy heuristics and decrease the number of physical machines up to 50% in simulated environment.

REFERENCES

1. Varasteh, A., & Goudarzi, M.,"Server consolidation techniques in virtualized data centres: A survey", IEEE Systems Journal, 11(2), 772-783, 2017.
2. Choudhary, A., Rana, S., & Matahai, K. J.," A critical analysis of energy efficient virtual machine placement techniques and its optimization in a cloud computing environment". Procedia Computer Science, 78, 132-138,2016.
3. Fan, Z., Shen, H., Wu, Y., & Li, Y., "Simulated-annealing load balancing for resource allocation in cloud environments". In Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2013 International Conference on (pp. 1-6). IEEE, December 2013.
4. Cardoso, L. P., Mattos, D. M., Ferraz, L. H. G., Duarte, O. C. M., & Pujolley, G., "An efficient energy-aware mechanism for virtual machine migration". In Global Information Infrastructure and Networking Symposium (GIIS), 2015 (pp. 1-6). IEEE, October 2015.
5. Tian, W., Zhao, Y., Xu, M., Zhong, Y., & Sun, X., "A toolkit for modeling and simulation of real-time virtual machine allocation in a cloud data center". IEEE Transactions on Automation Science and Engineering, 12(1), 153-161, 2015.
6. Çağlar, İ., & Altilar, D. T., "An energy efficient VM allocation approach for data centers", In Big Data Security on Cloud (Big Data Security), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference on (pp. 240-244), IEEE, April 2016.
7. Duolikun, D., Watanabe, R., Enokido, T., & Takizawa, M., "A model for migration of virtual machines to reduce electric energy consumption", In Complex, Intelligent, and Software Intensive Systems (CISIS), 2016 10th International Conference on (pp. 159-166), IEEE, July 2016.
8. Basu, D., Wang, X., Hong, Y., Chen, H., & Bressan, S, "Learn-as-you-go with megh: Efficient live migration of virtual machines", In Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on (pp. 2608-2609). IEEE, June 2017.
9. Yang, J., Shi, X., Marchese, M., & Liang, Y., "An ant colony optimization method for generalized TSP problem", Progress in Natural Science, 18(11), 1417-1422, 2008.
10. Guan, X., Wan, X., Choi, B. Y., & Song, S.,"Ant colony optimization based energy efficient virtual network embedding", In Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on (pp. 273-278), IEEE, October 2015.
11. Plakunov, A., & Kostenko, V. (2014, October),"Data center resource mapping algorithm based on the ant colony optimization", In Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), 2014 First International (pp. 1-6), IEEE, October 2014.
12. Chander, S. "SWOT Analysis of Tabu Search and Simulated Annealing Task Scheduling Algorithms in Distributive Environment".
13. Takouna, I., Dawoud, W., & Meinel, C. (2012). "Energy efficient scheduling of hpc-jobs on virtualize clusters using host and VM dynamic configuration", ACM SIGOPS Operating Systems Review, 46(2), 19-27, 2012.
14. Bagheri, Z., & Zamanifar, K.," Enhancing energy efficiency in resource allocation for real-time cloud services", In Telecommunications (IST), 2014 7th International Symposium on (pp. 701-706), IEEE, September 2014.
15. Chawarut, W., & Woraphon, L., "Energy-aware and real-time service management in cloud computing", In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on (pp. 1-5), IEEE, May 2013.
16. Challita, S., Paraiso, F., & Merle, P., A Study of Virtual Machine Placement Optimization in Data Centers. In 7th International Conference on Cloud Computing and Services Science, CLOSER 2017, April 2017.
17. DONG, J. K., WANG, H. B., LI, Y. Y., & CHENG, S. D., "Virtual machine placement optimizing to improve network performance in cloud data centers". The Journal of China Universities of Posts and Telecommunications, 21(3), 62-70, 2014.
18. Guan, X., Choi, B. Y., & Song, S., "Topology and migration-aware energy efficient virtual network embedding for green data centers", In Computer Communication and Networks

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

(ICCCN), 2014 23rd International Conference on (pp. 1-8). IEEE, August 2014.

19. Gutierrez-Garcia, J. O., & Ramirez-Navarrete, A." Collaborative agents for distributed load management in cloud data centers using live migration of virtual machines", IEEE transactions on services computing, 8(6), 916-929, 2015.
20. Akhter, N., & Othman, M., "Energy efficient virtual machine provisioning in cloud data centers", In Telecommunication Technologies (ISTT), 2014 IEEE 2nd International Symposium on (pp. 330-334). IEEE, November 2014.
21. Zheng, X., & Cai, Y, "Dynamic Virtual Machine Placement for Cloud Computing Environments", In Parallel Processing Workshops (ICCPW), 2014 43rd International Conference on (pp. 121-128). IEEE, September 2014.
22. Ye, K., Huang, D., Jiang, X., Chen, H., & Wu, S. "Virtual machine based energy-efficient data center architecture for cloud computing: a performance perspective", In Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing (pp. 171-178). IEEE Computer Society, December 2010.