International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 12 | Dec 2020 www.irjet.net p-ISSN: 2395-0072

CS-ANN BASED FALSE ALLOCATION DETECTION SYSTEM FOR CLOUD DATA CENTRE

Ajit Mehta[1], Harpreet Kaur[2]

[1]*Ajit Mehta, Dept. of Computer Science &Engg.*
[2]*Harpreet Kaur, Head of CSE Dept., Rayat Bahra Group of Institutions, Patiala 147001, Punjab, India*

---***---

Abstract - *Modern Cloud Data Center (CDC) uses virtualization technology for effective resource management to minimize costs and energy budgets of cloud computing system. Virtualization technology has been widely used to compute resources in CDC. Virtualization enhances the resource utilization by reducing the use of more hardware. VM-based infrastructure overhaul program is necessary to recover the affected services in an appropriate manner. The VM selection plays an essential role to minimize VM migrations that in turn helps to minimize power consumption. Power consumption varies directly with respect to the CPU utilization. But with the existing Ant Colony Optimization (ACO) algorithm a number of migrations appear that led to degrade performance of the cloud system. The proposed Cuckoo Search (CS) algorithm along with Neural Network as a classification approach is applied for VM allocation to select an appropriate host that can results in small migration by allocating VMs to the best host. The performance of the designed Cloud system is tested in PYTHON in terms of SLA violation, number of migrations and energy consumption.*

***Key Words*: Cloud Data Centre, VM migration, Cuckoo Search, Artificial Neural network.**

1. INTRODUCTION

In the last couple of years, IT infrastructure is growing rapidly with the demand for computing power created by modern computing work and scientific applications. This large scale infrastructure leads to consume high electrical power and hence increase the cost of operation as well as its infrastructure [1]. The high power consumption of IT infrastructure results in low system reliability as well as the lifetime of the system due to overheating. It is very important for a mobile cloud provider to deliver high Quality of Service (QoS) for their users with negotiable Service Level Agreements (SLA) [2]. To assure effective resource management, cloud providers such as Amazon EC2 [3] has to maintain power consumption. It has been observed that in the cloud data centers, the average power utilized by resources is estimated to be less than 30% and the remaining 70 % of power is consumed by the ideal nodes (doing nothing).This results in increased infrastructure cost. To obtain high energy efficiency cloud service provider is now becoming a great challenge for the researchers [4].

In order to fully understand the capabilities of cloud computing, cloud providers need to ensure that they can adapt to their virtual machine (VM) transport to meet different buyer prerequisites while keeping customers away from the basic data center [2]. Virtualization applications dispense many comforts, including consolidation, migration. In this workspace, data centre has becomes a main hub of interchangeable computer resources which are leveraged to run the virtual machine as needed. A general technique for enhancing data centre energy proficiency is to place VMs by co-coordinating the number of dynamic servers to meet the current needs of VMs and using SLA violations to place the remaining servers in low-control standby mode [3].

Cloud allows multiple services to be hosted on globally shared resource pools, where resources are allocated to on-demand services [4]. It uses a virtualized environment to run the service, because there is no virtualized computing being inefficient and inflexible. But, it has some service performance degradation, in terms of energy costs and a lot of power consumption. In the past, many researchers have worked on energy-saving algorithms to reduce energy consumption. Many algorithms have been implemented to conserve data center power by shutting down or placing idle servers in the server's sleep mode [5]. However, these technologies are not as effective as service performance degradation and inappropriate resource utilization. Some of the previous work includes the idea of developing energy-efficient algorithms for data centers and put forward a virtual machine placement algorithm for minimizing the migration (MM) by using host CPU utilization [6]. The algorithm outperforms other placement algorithms, but they do not take SLA parameters into account when selecting virtual machines for migration, which may be achieved by real-time migration. Most violations occur during real-time migration of virtual machines that affects SLA parameters such as availability, response time, throughput, network bandwidth, and so on. Therefore, it is necessary to develop a new method for SLA-aware energy-efficient algorithms for resource allocation in the data center [7].

Resource allocation is a mechanism that has been implemented in many computing regions, for example data centre management, operating systems, and grid computing [8]. Resource allocation encompasses scheduling of activities as well as allocating the accessible resources in a quite cost-effective way and also it implemented on optimal processes to proficiently assign physical and/or virtual resources to various applications of developers', therefore, by reducing the operational price of the cloud-environment. The consumption of energy associated with the resources allocation should be taken into account [9]. Computers can be made as the device that consumes less energy by using lower power processor, using cooling devices as well as using spinning SSD (Solid State Drive) of small size rather than large size. Intel has developed a process that is going to use less power. It utilizes resources like data centres, heat, light, power etc. in an optimal manner. Indian IT industry has witnessed big growth in the world [10]. However, this huge growth leads to the high consumption of energy as well as power that results in high cost consumption.

In cloud computing paradigm, the main challenge is the allocation of several accessible resources between various end-users having varying requests of resources dependent upon their patterns of application usage. The random as well as varying requests need to run on data-centre resources through Internet. The goal of resource allocation for any specific cloud provider could be either enhances the applications of Quality of Service or increase utilization of resource along with energy proficiency. The key objective is to augment Quality of Service parameters (i.e. response time) which measures the competence of resource-allocation irrespective of the category of ICT resources assigned to any specific end-users. The enhanced Quality of Service parameters could be any measure for example budget, communication delay, space, and time [11]. The aim of this paper is to allocate resource in an efficient way to end users by utilizing optimization approach in addition to artificial intelligence approach.

The rest of the paper is organized as: section 2 presented the related work. The proposed work along with detail description and algorithm is presented in section 3. Section 4 presented the result and discussion section. Conclusion is presented in section 5.

1. Related Work

The present allocation of VM can be optimized through the VM migration process that has been performed on the basis of pre-described constraints. The entire migration procedure is performed in three phases: selection of best VM, destination host selection as well as selection of VM migration. The VM placement is an NP-hard problem that can be deliberated as a bin packing dilemma with different bin sizes in which bin represents the mobile cloud servers and VM is used as object. A review of different VM placements techniques has been provided by Rathod et.al [12]. The algorithms such as First Fit, next fit and random fit are the simplest VM allocation algorithms and have not been taken into account the capacity of resources while placing VMs. These techniques have also been not considered the capacity of physical machine's resource consumption. Therefore, in [13] the power- aware algorithm has been initially presented, in which, the problem of dynamic VM placement has been resolved with less power requirement and with smaller SLA violation. A P mapper algorithm has been presented by Beloglazov et al; to resolve the power consumption and SLA violation problem. This technique mainly consists of three controlling units such as power management unit, performance and migration management unit. This technique performed both VM placement as well as VM reallocation to achieve an optimized allocation. The test results show that about 25 % of power has been saved using p-mapper algorithm compared to individual first fit and random fir placement algorithm [14].

There are a number of schemes that have been utilized by number of scholars to minimize energy consumption by the mobile cloud data servers in the mobile cloud environment. To resolve this problem [15] have proposed an energy-aware VM placement scheme in combination to swarm inspired Particle Swarm Optimization (PSO) algorithm to minimize the total energy consumption rate and hence enhance the server usages and provide satisfaction to mobile cloud users. The parameters, as well as the operators, have been redefined to resolve the discrete optimization issues. The parameters have been readjusted as the traditional PSO technique has failed to solve the problem of VM placement since it is mostly utilized for solving only continuous optimization complexities. The PSO algorithm helps to minimized the searching time and hence save energy with better server utilization. In [16] the concept of optimization algorithm namely; genetic Algorithm (GA) has been integrated with load balancing mechanism of hosts. A number of researchers have used GA for the placement of VMs on host, but sometimes this optimization algorithm tends to increase the migration unnecessarily such as migrated VM immediately after the placement. The problem of immediate migration has been resolved in this paper by using VM placement algorithm which utilized the past behaviour to take decision for the VM placement. The dynamic workload problem has been resolved using 70 % of threshold value. At last, the comparison of the proposed approach with the traditional Round Robin, Non Power-aware algorithm has been provided, which indicated that the proposed approach performs well in terms of energy-saving with optimal COU utilization. (Basu et al; 2019) have resolved the problem of workload using nature-inspired GA along with the improved scheduling of VMs. Every chromosome of GA has been defined as nodes and the VM is allocated to a node that is related to the chromosome's genes. Thereafter based on the crossover and mutation operator the allocation of VM has been performed. The results have indicated that the proposed approach has performed well in terms of resource utilization and load balancing [17].

2. Proposed Work

The flowchart of proposed work is shown in Figure 1. Initially, a set of users and servers are created to develop a simulation environment. Then each server is allocated to requested users using MBFD algorithm. The VM allocation problem can be divided into two parts: (i) accept the request from the user, and placing VMs on an appropriate host; (ii) not optimize the current VM allocation. To resolve the first problem, modified Best-Fit Descending (MBFD) algorithm is used to allocate VM to host according to their CPU utilization. The algorithm for MBFD is written below.

Algorithm 1: MBFD

Input: H_{LIST} ← List of Host

VM_{LIST} ← List of VMs

Output: VM_{SORTED} ← List of Sorted VMs

Start
Apply sorting on VM_{LIST} based on their CPU utilization
H-Count = Size (H_{LIST})
VM-Count = Size (VM_{LIST})
For each H-Count
Calcuate power of VMs, Minpower ← max
Create a null matrix to store sorted VMs, VM ← [null]
For each VM-Count
Assign Power, VM_{POWER} ← Estimated Power (VM_{LIST}(VMs), H_{LIST})
  If VM_{POWER} < mean(VM_{POWER}) then
Allocate VM ← VM_{LIST}(VMs)
Assign, Min(VM_{POWER}) ← VM_{POWER}
  If Allocated H_{LIST} ≠ null then
Allocate VM
VM_{SORTED} ← Sort VMs according to VM_{POWER}
End – If
End – If
End – For
End – For
Return: VM_{SORTED} as a list of Sorted VMs
End – Algorithm

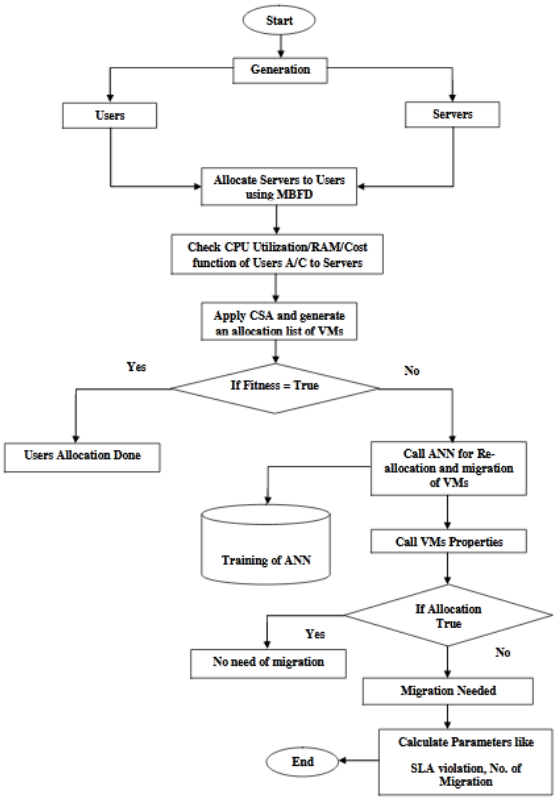

Figure 1: Flow Chart of proposed work

List of host is entered as input data to MBFD algorithm. Then, sorting of VMs is performed as per their CPU utilization level. For every host count and VM count, minimum and maximum power is estimated. In case of tie between host and CPU utilization, the algorithm sort VMs based on available RAM. This algorithm is highly efficient while consolidating VMs to a smaller number of servers. The energy is saved using this algorithm as it turns off those servers that are in sleep mode. But this algorithm also faces some drawbacks like placing VMs to the host with maximum CPU utilization led to overload the servers, which in turn increase SLA violation and number of migrations. To reduce this problem Cuckoo search (CS) algorithm is applied as an optimization approach. Using this algorithm a list of VMs is created. The aim of using CS is to select appropriate host to the VMs that need to be migrated. The description of CS is written below.

3. Cuckoo Search

The Cuckoo search algorithm is one of the swarm intelligence algorithms, developed by Xin-She Yang and Suash Deb in 2009. The inspiration for its creation was the nesting parasitism of some species of cuckoos that lay their eggs in the nests of other birds (other species of birds). Some of the owners of the nests may come into direct conflict with the cuckoos that rush into them. For example, if the owner of the nest discovers that the eggs are not his or her, then he or she will either discard these alien eggs or simply leave the nest and create a new one somewhere else. The algorithm for the same is written below.

Algorithm 2: Cuckoo Search Algorithm

Input: H_{LIST} ← List of Host

VM_{LIST} ← List of VMs

VM_{SORTED} ← List of Sorted VMs

Output: $VM_{ALLOCATED}$ ← List of Allocated VMs

Start
Initialize algorithm parameter – Iterations (I)
Number of Egg (N_{Egg})

Number of Variables (Nvar)

VM-Count = Size (VM_{LIST})
Set fitness function for cuckoo eggs, fit_{fun}

$$fit_{fun} = \begin{cases} 1; & if\ VM\ have\ allocation\ power\ (C_{Egg} > T_{Egg}) \\ 0; & otherwise \end{cases}$$

Where fs is the random change in egg and fth isthe threshold which should be achieved by host birds
For each I&VM-Count
$C_{Egg} = \sum_{I=1}^{VM-Count} VM\ Power = N_{Egg}$ // Current egg represent the current VM
$T_{Egg} = \frac{\sum_{I=1}^{VM-Count} Power\ (VM_{SORTED})}{VM-Count}$ // Threshold egg represents the current VM
fit_{fun} = fitness function // which is define above
$VM_{allocation} = CSA(N_{Egg}, N_{var}, fit_{fun})$
$VM_{ALLOCATED} = VM_{allocation}$
End – For
Return: $VM_{ALLOCATED}$ list of Allocated VMs
End – Algorithm

The inputs to CS algorithm are (i) list of Vms and (ii) list of hosts, and (iii) list of sorted VMs by MBFD algorithm. Initially, the parameters for CS, like number of eggs (VMs), number of variables (CPU, RAM), VM count and size has been initiated. If fitness function is 1 means satisfied the condition, and then added to VM allocated list. Otherwise call for ANN for Re-allocation and migration of VMs.

4. Artificial Neural Network

Neural networks are able to independently learn and develop, building by their experience on the mistakes made. Analyzing and processing information from a specific source or from the Internet as a whole, a self-organizing system is able to create new products, not only reproducing and structuring input data, but also forming a qualitatively different result previously inaccessible to artificial intelligence. ANN mainly comprised of three layers namely input, hidden and output layer. The output of CS algorithm that is list of allocated VMs is passed to the input layer along with the sorted VMs list. The VMs are processed in the hidden layer by passing weight function. A list of migrated VMs is obtained, which is decided by the neural network as per the properties of current VMs in cloudlet. If ANN finds that Allocation is true then there is no need to migrate VMs, whereas in case of false allocation ANN needs to migrate VMs. The algorithm for the same is written below.

Algorithm 3: ANN

```
Input:  VM_ALLOCATED ← List of Allocated VMs as T-Data
N ← Number of neurons as carrier
        VM_SORTED ← List of Sorted VMs
Output: VM_MIGRATED ← List of Allocated VMs

Start
Call ANN and Set parameters using VM properties: List of Allocated VMs as a Training data (T-Data),
No. of VMs in VM_SORTED as a Target (T_R) and
Neurons (N)
Define aCloud Network, Cloud-Net = Newff (T-Data, T_R, N)
Cloud-Net.TrainParam.Epoch = 1000 // Total simulation rounds
Cloud-Net.Ratio.Training = 70% // Default division of T-Data as training ratio
Cloud-Net.Ratio.Testing = 15%// Default division of T-Data as testing ratio
Cloud-Net.Ratio.Validation = 15%// Default division of T-Data as validation ratio
Cloud-Net = Train (Cloud-Net, T-Data, T_R)
Current VM = Properties of current VM in Cloud-Net
VM_ALLOCATION = simulate (Cloud-Net, Current VM)
Check allocation
If VM_ALLOCATION = True
Not need to migrated the VMs
VM_MIGRATED = []
Else
Need of migration
VM_MIGRATED = VM_SORTED
End – If
Return: VM_MIGRATED as a list of migrated VMs
End – Algorithm
```

5. Result and Discussion

In this section, we describe the experimental results to check the performance of the designed CS-NN based cloud system. The test results are analyzed to show the feasibility of the proposed CS-NN system compared to exiting schemes. The performance is evaluated by analyzing parameters SLA violation, and VM migration. The CS-ANN based cloud system is implemented by considering the input parameters as listed in Table 1.

Table 1: Considered Parameter for Experiment

Number of hosts	300
Number of VMs	800
CPU capacity of VM	200,400,600,800
CPU capacity of Host	1500,2500,3500
Number of Cloud Data Center (CDC)	1
Memory capacity of host (MIPS)	1000
Memory capacity of VM (MIPS)	200

Based on above consideration parameters, that are 300 hosts and 800 VMs, the simulation has been performed 10 times so that accurate results with small error can be determined. The examined parameters with respect to number of simulations are presented in Table 2.

Table 2 Simulated Results

Number of iterations	SLA Violation (%)		Number of Migrations		Energy Consumption (KWh)	
	Proposed Work	Karda and Kalra (2019)	Proposed Work	Karda and Kalra (2019)	Proposed Work	Karda and Kalra (2019)
1	8	10	20	27	6.5	6.8
2	7.1	8	15	23	6.42	6.64
3	7.8	9.8	16	24	6.46	6.75
4	6.6	9	17	26	6.61	6.76
5	7	8	19	27	6.52	6.6
6	8	9.1	18	28	6.62	6.79
7	8.6	10	21	29	6.5	6.7
8	7.4	9.9	25	32	6.52	6.78
9	9.2	11	26	31	6.3	6.5
10	7.8	10	20	30	6.2	6.5

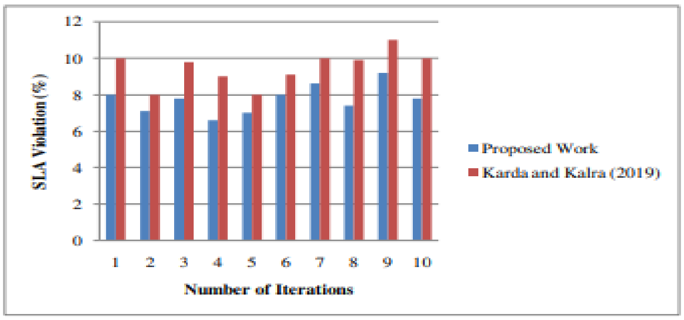

Figure 1: Comparison of SLA Violation

SLA violation is an important parameter that cloud providers must meet to ensure better delivery of cloud services to their users. In addition to energy-saving policies, SLA parameters must be met. It represents the agreement between the user and the cloud service provider. Lower SLA means that the cloud provider will provide better services to cloud users, conversely. The graph drawn in Figure 1, shows that the proposed cloud system provides a better services for its cloud users. The average values of SLA violations obtained by proposed CS-ANN scheme are 7.75%. Thus, there is an improvement of about 18.25 % has been observed compared to existing Karda and Kalra (2019).

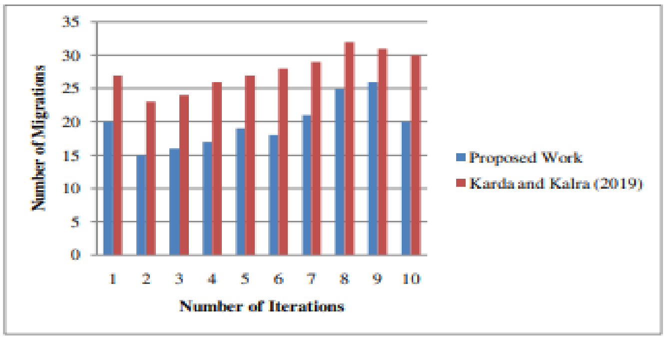

Figure 2: Comparison of Number of Migrations

Figure 2 represents the comparison graph drawn from a number of migrations with respect to number of iterations. The comparison has been done with the existing work performed by Karda and Kalra (2019) using Ant Colony as a VM selection approach. From the graph it is clear that the integrated technique of CS with ANN performs well compared to applying an individual optimization scheme.

Thus, there is an improvement of about 28.88 % has been observed while evaluating Number of migrations.

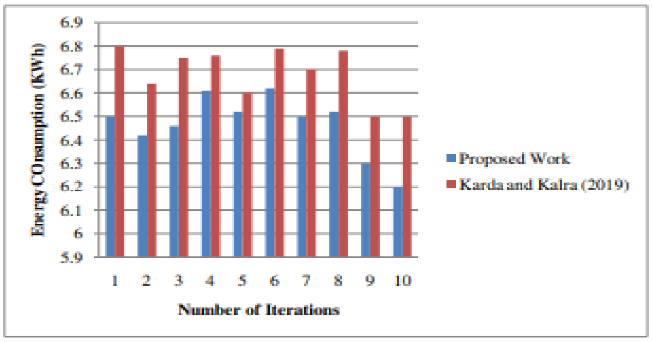

Figure 3: Comparison of Energy Consumption

Figure 3 shows the comparison of energy consumption with respect to number of iterations. From the graph it is observed that energy consumed while migrating VMs to host using CS with ANN approach is small compared to energy consumed by using ACO approach by Karda and Kalra (2019). The percentage reduction of 3.29 % has been attained while using CS with ANN approach compared to Karda and Kalra (2019).

6. Conclusion

This paper presented an energy-efficient approach for allocation of VM in cloud computing environment using the concept of CS along with the ANN. The process of VM allocation typically refers to allocation of VMs to physical hosts, which usually takes in consideration of their CPU utilization, RAM and Storage Capacities. However, in a cloud computing environment, we have more VMs that need to be allocated, so this is a complicated task for researchers to manage and allocates VMs with minimum number of migration and energy consumption. In this paper, we use the concept of CS along with the ANN to make a fast and robust mechanism for VM allocation and their migration if needed. Here, CS is used with a novel objective/fitness function which is used to optimize the allocation mechanism by considering the cloud environment as an entire population space based on the VM properties with traditional concept of MBFD. Here, MBFD and CS helps to allocate the VMs efficiently but in some case user faced the wrong allocation problem. So, we used the concept of ANN to verify the reallocate the VMs and if migration required then migrate then with maximum efficiency. From experiments, the percentage reduction in SLA violation, VM migration and energy consumption compared to ACO approach has been examined as 18.25 %, 28.88%, and 3.29% respectively.

References

[1] Jeba, J. A., Roy, S., Rashid, M. O., Atik, S. T., & Whaiduzzaman, M. (2019). Towards Green Mobile cloud Computing an Algorithmic Approach for Energy Minimization in Mobile cloud Data Centers. International Journal of Mobile cloud Applications and Computing (IJCAC), 9(1), 59-81.

[2] Bhattacherjee, S., Das, R., Khatua, S., & Roy, S. (2019). Energy-efficient migration techniques for mobile cloud environment: a step toward green computing. The Journal of Supercomputing, 1-29.

[3] Xu, X., Zhang, Q., Maneas, S., Sotiriadis, S., Gavan, C., & Bessis, N. (2019). VMSAGE: A virtual machine scheduling algorithm based on the gravitational effect for green Mobile cloud computing. Simulation Modelling Practice and Theory, 93, 87-103.

[4] Patel, D., Gupta, R. K., & Pateriya, R. K. (2019). Energy-Aware Prediction-Based Load Balancing Approach with VM Migration for the Mobile cloud Environment. In Data, Engineering and Applications (pp. 59-74). Springer, Singapore.

[5] C.Lin, P.Liu, J. Wu, "Energy-Aware Virtual Machine Dynamic Provision and Scheduling for Cloud Computing", IEEE 4th International Conference onCloud Computing, 2011, pp.736-737.

[6] Y. Jadeja and K. Modi, "Cloud computing-concepts, architecture and challenges," in Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on, 2012, pp. 877– 880.

[7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype,and reality for delivering computing as the 5th utility," Futur. Gener. Comput. Syst., vol. 25, no. 6, pp. 599–616, 2009.

[8] Yamini, B.; Selvi, D.V., "Cloud virtualization: A potential way to reduce global warming," Recent Advances in

Space Technology Services and climate Change (RSTSCC), 13-15 Nov. 2010, pp.55-57.

[9] Xiaomin Zhu, Laurence T. Yang, Huangke Chen Ji Wang, Shu Yin and Xiao cheng Liu, "Real-Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds", IEEE 2014.

[10] Jinn-Tsong Tsai Jia-Cen Fang, Jyh-Horng Chou "Optimized task scheduling and resource allocation on cloud computing environment uses Improved Differential Evolution Algorithm (IDEA)", Computers & Operations Research Elseveir, vol. 40, issue 12, 2014, pp.3045-3055.

[11] Qiang Li, Qinfen Hao, Limin Xiao, and Zhoujun Li, "Adaptive Management of Virtualized Resources in Cloud Computing Using Feedback Control," in First International Conference on Information Science and Engineering, April 2009, pp. 99-102.

[12] Rathod, Suresh B., and V. Krishna Reddy. "NDynamic Framework for Secure VM Migration over Cloud Computing." Journal of Information Processing Systems 13, no. 3 (2017).

[13] Verma, A., Ahuja, P., & Neogi, A. (2008, December). pMapper: power and migration cost aware application placement in virtualized systems. In Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (pp. 243-264). Springer-Verlag New York, Inc..

[14] Beloglazov, A., & Buyya, R. (2010, May). Energy efficient resource management in virtualized mobile cloud data centers. In Proceedings of the 2010 10th IEEE/ACM international conference on cluster, mobile cloud and grid computing (pp. 826-831). IEEE Computer Society.

[15] Wang, S., Liu, Z., Zheng, Z., Sun, Q., & Yang, F. (2013, December). Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers. In 2013 International Conference on Parallel and Distributed Systems (pp. 102-109). IEEE.

[16] Madhusudhan, B., & Sekaran, K. C. (2013). A genetic algorithm approach for virtual machine placement in cloud. In Published in the proceeding of International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA2013).

[17] Basu, S., Kannayaram, G., Ramasubbareddy, S., & Venkatasubbaiah, C. (2019). Improved Genetic Algorithm for Monitoring of Virtual Machines in Cloud Environment. In Smart Intelligent Computing and Applications (pp. 319-326). Springer, Singapore.

[18] Karda, Raksha Kiran, and Mala Kalra. "Bio-inspired Threshold Based VM Migration for Green Cloud." In Advances in Data and Information Sciences, pp. 15-30. Springer, Singapore, 2019.