



A HYBRID G-ACO APPROACH ON LOAD BALANCING IN CLOUD COMPUTING

Parmjeet Kaur

Computer Science Engineering
Punjab Technical University
Jalandhar, India

Abstract: Cloud computing permits facilitating of various administrations on various datacenters where assets are designated to clients on request. It utilizes the concept of virtualization for working online, on the grounds that without virtualization cloud computing is inefficient and not adaptable. Load balancing is a method that distribute the workload among different nodes in the given environment such that it no node in the system is overloaded. The different conventional load balancing techniques not performed well and they doesn't consider SLA quality of service parameters while choosing virtual machine for relocation. Many other issues are likewise associated with migration process like number of migration, utilization of resources, response time and memory. So there is have to grow new approach for load balancing in data centers using VM algorithms that beat the issues in customary methodologies and enhance their execution. So the hybrid approach utilizing different strategies like Ant Colony Optimization (ACO) and Genetic algorithm (GA) is proposed in this paper. This paper defeat the issue of stagnation in ACO-VMM system. The outcomes are mimicked in cloudsim environment.

Keywords: Overload balance, cloud computing, ACO, Qos, virtualization, migration strategy, Genetic method.

I. INTRODUCTION

Cloud computing is a new concept where services and resources are made available to user as a pay per use model. It represents the cooperation between multiple host machines and the services by network that provides powerful services to users [1] [2]. The basic idea behind cloud computing is distributed computing and grid computing. Due to usage of large data over internet it has become very important to manage the datacenters as well as computing resources [2]. Virtualization is the technology behind the cloud which realize the vision of utility computing. Cloud provides computing resources in the form of virtual machine, which is an abstract machine runs on physical machine. The requester using the virtual machine will have the feel of working on physical machine [3]. According to demand of user the datacenters are unevenly loaded of workload. So there are some issues like load balancing, power management, resource utilization, SLA violations that has to be take care of Virtual machine migration helps a lot in managing of the load in cloud computing by live migrations [4]. Virtual machine migration is the idea of moving the virtual machines starting with one physical machine then onto the next inside same datacenters or other and the resource utilization among all servers.

Idea of virtual machine migration in cloud computing has been appeared beneath in fig.1. Live migration is the main highlight of VMM, it basically exchange the entire state and condition of working VM starting with one host to another host. The live migration is used in cloud computing for load balancing and scheduling on host, adaptation to failure, power consumption and green figuring [5]. Some approaches are used in live migration, they are precopy, postcopy and hybrid live VMM. This work proposed a hybridization of Ant colony optimization and Genetic (G-ACO) algorithm to reduce the workload on host by reduction in VM migration.

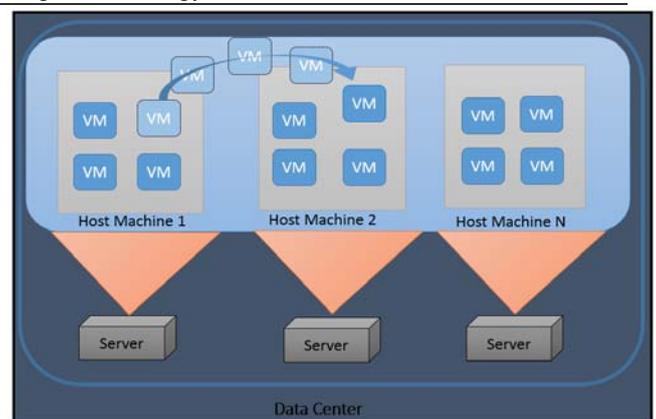


Figure. 1 Virtual Machine migration in cloud datacenters

II. RELATED WORK

Load balancing has dependably been a research area whose goal is to guarantee that each computing resource is distributed effectively and reasonably and at last improves utility. In traditional computing environments of parallel computing distributed computing, and grid computing, researchers in have proposed a number of static and dynamic and mixed scheduling strategies [6]. **Wei-Tao Wen et al.** [6] proposed a method for live migration which plays an important method for load balancing, in this proposed paper, Ant Colony Optimization algorithm is used to design a novel distributed Virtual Machine Migration strategy. In the ACO-VMM technique, local migration agent independently monitors the utilization of resources and launch the migration. Experiments show that ACO-VMM reduce the number of migrations and SLA violations as compare to traditional techniques. **Ramezani et al.** [7] have developed a TBSLB-PSO method that improves utilization of resources. VMM is been proposed for reducing the downtime for overloaded virtual machines, but this techniques still consume time, cost and large amount of memory while migration. To resolve these drawbacks, a

task-based load balancing system is introduced using PSO (Particular Swarm Optimization). In this proposed work only the extra tasks from any overloaded VM is transferred while in previous techniques whole VM is migrated. **K R Ramesh Babu et al.** [8] have developed a heuristic algorithm based on honey bees to avoid load on virtual machines. The load can be distinguished as CPU, ram or network load. This paper proposed a bee colony algorithm that is based on the behavior of honey bee for foraging the food. The overloaded VMs are treated as bees and under loaded treated as food sources. In the waiting queues of VMs, this method also consider the priorities of tasks. The experimental results has been shown that QoS (Quality of Service) is improved, response time is minimized and number of migration of tasks are reduced. **Kun-Ting et al.** [9] have developed a network aware algorithm for parallel migration. A network aware topology for parallel migration is designed to enhance of speed of load balancing in datacenters. By using weighted bipartite graph the multi resource scheduling problem is transformed into minimum weight match problem. The experimental results prove that the algorithm is compatible for multi-resource migration and the balanced time is also improved. The 10% of throughput is gained by assuming a large number of applications on various PMs. **Peng Lu et al.** [10] developed an Adaptive Live Migration technique to balance the load among VMs in datacenters. This paper proposed a workload adaptive model to reduce the VM downtime and improve customer experience. This technique records the historical mappings of VMs to predict the future scheduling decisions. The results prove that this technique is faster than previous techniques and reduction in latency by 79% almost. The proposed work also reduces the minimal downtime by 73%. **Gutierrez et. al.** [11] have developed an Agent based Load balancing Algorithm for efficiently working of load balancing. The datacenters of clouds are composed of number of PMs that consider the number of VMs with various specifications and resources. According to user requests, the resources are imbalanced within data center that cause degradation in performance level and violate the Service Level Agreements. The working process is taken by agents (1) determine which VMs should migrated to their destination by migration heuristics. (2) The various migration policies that decide when the virtual machines are migrated (3) virtual machines acceptance policy to define which VMs are hosted (4) load balancing heuristics for front end. **Bhaskar et. al.** [12] proposed a mechanism implemented in two phases. Firstly it finds the cpu utilization and memory needed for each instance and also figure out the remaining memory for each virtual machine. Secondly ,it compare the available resource with required resource, if required resource are available then move further otherwise discard the request. **Kwang Mong Sim et al.** [13] proposed ACO algorithm for Routing and Load Balancing. The ACO (Ant Colony Optimization) is the technique in which ants are considered as VMs and transformed into powerful optimization methodology that find tasks in computer network. In this work, the NP-hard problem solving paradigm for ACO is explicated and this technique is compared to traditional algorithms based on the issues of routing knowledge, overhead and adaptivity. This survey includes the approaches for mitigating stagnation (major problem of ACO), apply ACO in load balancing and

routing to find better results, identify the open problems and discuss new directions. The terms that pay effect on mitigating stagnation are: aging, evaporation, pheromone heuristic control, privileged pheromone laying and pheromone smoothing. The comparison is done with ant based control algorithm, Ant Net algorithm, and ASGA and Synth ECA algorithms. **Ghumman et al.** [14] proposed a combinational approach of Max Min and Ant Colony Optimization. The various algorithms are introduced to solve these problems but still time and cost is the biggest issue. The improved Max-Min Ant Colony algorithm is depends upon the execution time rather than completion time. The main objective of this proposed work is to balance the load and minimize the makespan. The simulation is taken under CloudSim toolkit. The comparison between improved and traditional algorithm proves that the total processing cost and time is reduced. **Mayur S. et al.** [15] introduced the improved Genetic algorithm for better performance. The genetic algorithm work on large number of solutions well and it uses random selection process as input. The traditional Genetic algorithm takes the processors and jobs for selection process. To improve the efficiency of genetic algorithm, Logarithmic Least Square Matrix is used. The starvation and problem of being idle the VMs is solved in the experiment.

From the literature survey it has been observed that there are some drawback such as static nature of load balancing algorithm, less reliability and scalability.it being observed that artificial intelligent concept such as honeybee algorithm, ant colony optimization, particle swam optimization. Genetic and intelligent agent had been employed for load reduction in cloud computing environment. Thus there is a need for an effective algorithm in cloud computing.

III. PROPOSED WORK

Scheduling is a process or a job on a Physical Machine which is very common in this cutting edge period of innovative world. To diminish a weight of a Physical Machine, a Virtual Machine is created which intakes the memory of Physical Machine itself. Issues occurs when the demand of a job exceeds even through the virtual Machines. In such a case, an effective allocation algorithm is needed which can effectively shift the Virtual Machines to specific PM, in efficient manner but in a limited budget.

Proposed load balancing strategy depends on the ant colony and genetic algorithm concept.

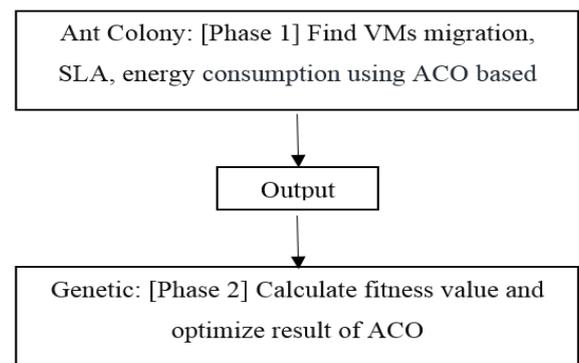


Fig. 2 Flow of Ant colony and Genetic combination

In our work ant colony use for the load balancing by vm migration to system and find vm on based of their criteria

but somewhere ant can't change the same resulted pheromone value for vm so not find the optimal result for it we used genetic fitness function for assuring to get optimal value of vms to migrate. The proposed work is divided into four sections as a source code but mainly work in two phases. The first section creates a random environment in which the user provides the number of Virtual Machines.

Algorithm 1:

Generation of Random Scheduling Environment:

```
generateEnvironment(host,VM)
// host= no. of host; //VM= no. of
VMs
for i=1: host
hostcpu(i) = xrandom(); //random generated cpu capacity
for PM
hostmemory(i) = xrandom(); //random generated memory
for PM
hostdisk(i) = xrandom(); //random generated disk for PM
end for;
for j=1:VM
vm_cpu(j) = xrandom();//random generated cpu capacity for
VM
vm_memory(j) = xrandom();//random generated memory
for VM
vm_disk(i) = xrandom(); //random generated disk for VM
end for ;
```

These steps created for dynamic runtime allocation and scheduling environment based on host machines and virtual machines. In above algorithm allocation of VMs to PMs machines has been done on random basis. Firstly initialization of no. VMs and hosts will be done.

Algorithm 2:

Allocation of VMs to PMs Algorithm:

```
PrimaryAllocation(hostcpu,hostmemory,hostdisk,vm_cpu,v
m_memory,vm_disk)
Allocation_Table = [ ] [ ];
count = 0;
for( i=0;host.length)
for( j=0:vm.length)
if(hostcpu[i]>vm_cpu[j]&&hostmemory[i]>=vm_memory[j]
&& hostdisk[i] >= vm_disk[j])
Allocation_Table[0] = vm_id;
Allocation_Table [1] = host_id;
count=count+1
end if
end for
end for
```

The second algorithm create an initial allocation probability based on the properties of the host and the VM machines. The algorithm takes number of Vms and PMs as Input and provides the allocation relationship of VMs and VMs as Output. If condition checks whether host have appropriate resources for the VM or not. A single VM could be allocate to more than one host at same time. E.g. Let us consider a VM which has cpu capacity required of 2.7 and memory required of 20 Mb.

We can see that 3 virtual machine fulfil the condition of algorithm so it has been placed into the allocation table three times on different hosts satisfied the condition of pre-allocation. After implementation of ACO and Genetic it would be decided that which Physical machine host VM 34. After that ACO-VMM algorithm is applied on it and a

pheromone solution is generated. SLA, number of vm migration and energy consumption is calculated.

Algorithm 3:

Implementation of ACO

- 1) If load condition if i-th PM > threshold then find the sorted list of VMs that should migrate.
- 2) While $i < N-1$
- 3) Do produce ants and Traversing according to positive traversing and update the amount of pheromone.
 $i=i+1$
- 4) Produce ants and traversing according to negative.
- 5) Obtain a list of low load condition PMs.
- 6) Match the PMs and VMs.
- 7) Get the new mapping relationship.
- 8) End if

Above algorithm give the new mapping relationship of migration we reduced the host load by migrating the vms of high cpu capacity from the sorted list. Also calculate the SLA and energy parameters.in above positive traversing is high load on host and negative traversing is low load on host.

Algorithm 4:

- 1) Get VMs and Host from the resulted migration of ACO technology.
- 2) Calculate population and Population places from that results.
- 3) Find Fitness function on each host.
 $F = T/C$
Where T= total cpu of all VMs on that Host
 $C = \text{count of VMs}$
- 4) Find Best_Fit VMs from all the VMs
If (population [num] >= f)
Best=population [num]
- 5) Apply mutation and crossover
if (change >= .50)
 $my_avg = my_avg + \lambda$
else
 $my_avg = my_avg - \lambda$
- 6) If current_util > my_avg
Update Final Solution and calculate VMs from Host

The above algorithm takes the output of ACO-VMM algorithm as Input and applies the Genetic algorithm to it and find the fittest ant of the fit data. Only the optimal solution will be generated from large pool set of solutions. Algorithm create a minimal change in data to affect the allocation .only greater that my_avg allocation is applicable.

IV. RESULT AND ANALYSIS

This section explain the simulation work. The work focuses on Load Balancing on physical machine through VM migrations in Cloud Computing. Ant Colony optimization and Genetic algorithm are combined for balancing the load.

Three parameters are used for simulating the work that are: SLA violation, number of migrations and energy consumption that are described below.

A. SLA Violation

SLA (Service Level Agreement) parameter fulfil QOS requirements that has service level needs for maximum response time for the data centers. Buyya and Beloglazov

[16] proposed a metric called SLAV (SLA violations) which considers both the SLA Violations due to Over-utilization (SLATAH) and SLA Violations due to Performance Degradation in Migrations (PDM).

$$SLATAH = \frac{1}{N} \sum_{i=1}^N \frac{Ts_i}{Ta_i}$$

Where N is the number of PMs; Ts_i is the total time during which the i -th PM has experienced the utilization of 100% leading to an SLA violation; Ta_i is the total of the i -th PM being in the active state

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{Cd_j}{Cr_j}$$

Where M is the number of VMs; Cd_j is the estimate of the performance degradation of the j -th VM caused by migrations; Cr_j is the total CPU capacity requested by the j -th VM during its lifetime.

$$SLAV = SLATAH \times PDM$$

The SLA violation for each number of virtual machines is calculated as shown in Fig. 3 by different test cases for ACO-VMM and G-ACO.

Table 1: Comparison between ACO-VMM and G-ACO method for SLA violations

| SLA Violation for ACO and G-ACO | | | | |
|---------------------------------|--------------|-----|---------|-------|
| Test cases | Input values | | ACO-VMM | G-ACO |
| Test case 1 | VMs | 100 | 1.20 | 0.19 |
| | Host | 100 | | |
| Test case 2 | VMs | 120 | 1.14 | 0.20 |
| | Host | 100 | | |
| Test case 3 | VMs | 150 | 1.14 | 0.21 |
| | Host | 100 | | |
| Test case 4 | VMs | 200 | 0.99 | 0.20 |
| | Host | 100 | | |

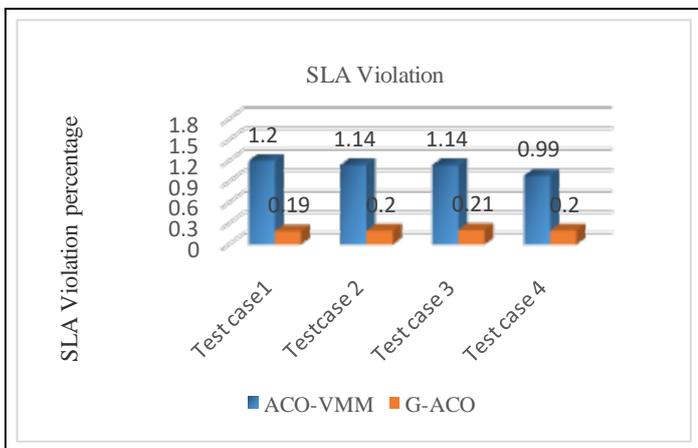


Figure. 3 Comparison between ACO-VMM and G-ACO method for SLA violations

B. Number of Migrations

Live VM migration is an expensive operation that includes some amount of CPU processing on the source PM, the link bandwidth between the source and destination PMs, the down time of the services on the migrating VM, and the total migration time [17]. One of our Objectives is to reduce the number of migrations. Figure 4 shows the number of VM migrations of both algorithms. Our algorithm has apparently reduced the number of migrations comparing to other ACO-VMM.

Table 2: Comparison between ACO-VMM and G-ACO method for VM migration

| Number of VM Migrations | | | | |
|-------------------------|--------------|-----|---------|-------|
| Test cases | Input values | | ACO-VMM | G-ACO |
| Test case 1 | VMs | 100 | 2687 | 1262 |
| | Host | 100 | | |
| Test case 2 | VMs | 120 | 3205 | 1517 |
| | Host | 100 | | |
| Test case 3 | VMs | 150 | 4660 | 1581 |
| | Host | 100 | | |
| Test case 4 | VMs | 200 | 4824 | 1098 |
| | Host | 100 | | |

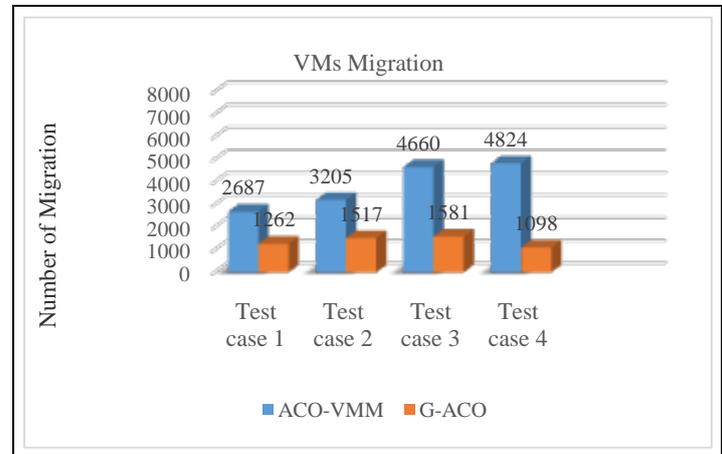


Figure. 4 Comparison between ACO-VMM and G-ACO method for VM migration

C. Energy Consumptions

Energy consumption by server in data center mainly consists of two power. Static power is always considered as a constant as long as the machine is switched on. Dynamic power is related with virtualized resource utilization in data center. Recent studies [18] show that server energy consumption is linear with CPU resource utilization. Moreover, it is identified that on average an idle server consumes approximately 70% of the power when it is fully utilized. Therefore, the power consumption as a function of CPU utilization ($P(u)$) is defined as follows:

$$P(u) = P_{static} + P_{dynamic} = 0.7P_{max} + 0.3P_{max} \cdot u = P_{max}(0.7 + 0.3u)$$

Where P_{max} could be obtained by statistic methods (for our experiment, P_{max} is set to be 250 W); u is the CPU utilization of a server.

Table 3: Comparison between ACO-VMM and G-ACO method for Energy Consumption

| Energy Consumption(Kw.h) for ACO and G-ACO | | | | |
|--|--------------|-----|---------|-------|
| Test cases | Input values | | ACO-VMM | G-ACO |
| Test case 1 | VMs | 100 | 7.58 | 6.73 |
| | Host | 100 | | |
| Test case 2 | VMs | 120 | 7.88 | 6.72 |
| | Host | 100 | | |
| Test case 3 | VMs | 150 | 6.67 | 5.51 |
| | Host | 100 | | |
| Test case 4 | VMs | 200 | 8.20 | 5.27 |
| | Host | 100 | | |

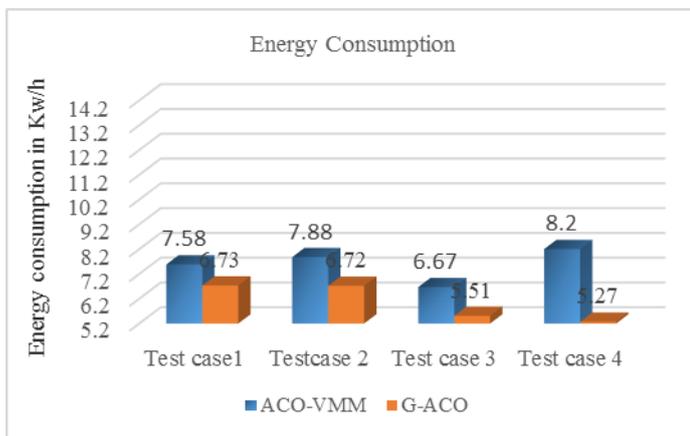


Figure. 5 Comparison between ACO-VMM and G-ACO method for energy consumption

V. CONCLUSION

To enhance the performance and overcome the problem of stagnation of traditional load balancing technique (ACO-VMM) we develop an efficient load balancing algorithm using hybridization of Ant colony optimization technique and Genetic algorithm. The various performance parameters like CPU cost, memory cost, configuration time and distance cost are used. From the simulation results, it is concluded that G-ACO (proposed work) is best from other policies, because it has least violate SLA (Service Level Agreement), number of migrations and energy consumption. In this paper, SLA is calculated, by considering no. of iterations as well as for no. of migrations of virtual machines over number of host machines.

In future, the performance of the load balancing algorithms is improved by using improved ACO algorithms, such as the Rank based Ant System and the Recursive Ant System. Migration cost is also a valuable parameter for future work. There are lots of algorithms which have been proved work well with ACO algorithm like honeybee, PSO or simulated annealing algorithm as a hybrid technique.

VI. REFERENCE

- [1] M. Sajid and Z. Raza, "Cloud Computing: Issues & Challenges", International Conference on Cloud, Big Data and Trust, pp. 13-15, 2013.
- [2] L. Wang, J. Tao, M. Kunze, A. Castellanos, D. Kramer and W. Karl, "Scientific Cloud Computing: Early Definition and Experience", 2008 10th IEEE International Conference on High Performance Computing and Communications, 2008.
- [3] L. Malhotra, D. Agarwal and A. Jaiswal, "Virtualization in cloud computing", International Journal of Computer Science and Mobile Computing, vol. 3, no. 8, pp. 745-749, 2014.
- [4] Rimal, Bhaskar Prasad, Eunmi Choi, and Ian Lumb, "A taxonomy and survey of cloud computing systems," INC, IAMS and IDC, pp. 44- 51, 2009.
- [5] R. Singh, D. Kahlon and S. Singh, "Comparitive Study of Virtual machine migration techniques and challenges in Post Copy live virtual machine migration", International Journal of Science and Research, vol. 5, no. 3, pp. 117-121, 2016.
- [6] W. Wen, C. Wang, D. Wu and Y. Xie, "An ACO-based Scheduling Strategy on Load Balancing in Cloud Computing Environment", 2015 Ninth International Conference on Frontier of Computer Science and Technology, 2015.
- [7] F. Ramezani, J. Lu and F. Hussain, "Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization", International Journal of Parallel Programming, vol. 42, no. 5, pp. 739-754, 2014.
- [8] K. Babu, A. Joy and P. Samuel, "Load Balancing of Tasks in Cloud Computing Environment Based on Bee Colony Algorithm", 2015 Fifth International Conference on Advances in Computing and Communications (ICACC), 2015.
- [9] K. Chen, C. Chen and P. Wang, "Network aware load-balancing via parallel VM migration for data centers", 2014 23rd International Conference on Computer Communication and Networks (ICCCN), 2014.
- [10] P. Lu, A. Barbalace, R. Palmieri and B. Ravindran, "Adaptive Live Migration to Improve Load Balancing in Virtual Machine Environment", European Conference on Parallel Processing, pp. 116-125, 2017.
- [11] J. Gutierrez-Garcia and A. Ramirez-Nafarrate, "Agent-based load balancing in Cloud data centers", Cluster Computing, vol. 18, no. 3, pp. 1041-1062, 2015.
- [12] R. Bhaskar, S. Deepu and B. Silylaja, "Dydnamic allocation method for load balancing in virtual machines for cloud computing environment", Advanced Computing: An International Journal, vol. 3, no. 5, pp. 53-61, 2012.
- [13] Kwang Mong Sim and Weng Hong Sun, "Ant colony optimization for routing and load-balancing: survey and new directions", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 33, no. 5, pp. 560-572, 2003.
- [14] N. Ghumman and R. Kaur, "Dynamic combination of improved max-min and ant colony algorithm for load balancing in cloud system", 2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-5, 2015.
- [15] Mayur S. Pilavare, Amish Desai " A novel approach towards improving performance of load balancing using Genetic Algorithm in cloud computing" IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ,ICIIECS'15
- [16] Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and

- performance efficient dynamic consolidation of virtual machines in Cloud data centers", *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397-1420, 2011.
- [17] P. Lu, A. Barbalace, R. Palmieri and B. Ravindran, "Adaptive Live Migration to Improve Load Balancing in Virtual Machine Environment", *European Conference on Parallel Processing*, pp. 116-125, 2017.
- [18] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy and G. Jiang, "Power and Performance Management of Virtualized Computing Environments Via Lookahead Control", *2008 International Conference on Autonomic Computing*, pp. 3-12, 2008.