



# Resource utilization enhancement through live virtual machine migration in cloud using ant colony optimization algorithm

Sandeep G. Sutar<sup>1</sup> · Pallavi J. Mali<sup>1</sup> · Amruta Y. More<sup>2</sup>

Received: 18 October 2019 / Accepted: 31 January 2020 / Published online: 11 February 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Cloud computing offers unlimited computational resources which are ready to use from anywhere, anytime on request. The achievement of maximized utilization of computational resources (physical and virtual) and minimized energy consumption of resources are goals of proposed system. The proposed system provides dynamic and energy efficient live VM (virtual machine) migration approach. This system reduces wastage of power by initiating sleep mode of idle physical machines results into energy saving. We propose a system consist with seven modules. (1) Resource monitor analyses energy consumption of resources. (2) Capacity distributor distributes maximum and minimum capacity for the physical machines. (3) Task allocator determines overloaded servers. (4) Optimizer analyses load on physical machine using ant colony optimization algorithm (5) Local Migration Agent calculates load of VMs to be migrated and select appropriate physical server. (6) Migration Orchestrator migrates the VM considering load. (7) Energy Manager initiates sleep mode for idle physical machine(PM)

**Keywords** Cloud computing · Energy management · Virtualization · Live virtual machine migration · Ant colony optimization algorithm

## 1 Introduction

Cloud computing offers unlimited computational resources (hardware or software) which are ready to use from anywhere, anytime on request. Cloud computing provides an significant chance of progress for industry to give a high quality of services like scalability and serviceability of computational resources. It is benefit of cloud computing that user can pay only for those resources which are accessible by them. These benefits of cloud encourage service providers to enlarge computational resources in data centre to sustain user requests. It includes significant increase in resources such as processor unit, storage

and etc. Therefore, proper utilization of cloud resources become major concern. There is increasing number of service providers who started to offer services of cloud. These services satisfy users requirement but consumes a great amount of resources energy. To achieve maximized utilization of computational resources (physical as well as virtual resources) and minimized energy consumption of resources are goals of proposed system. Virtual Machines using live migration helps to reduce the number of active physical machines (Cardoso et al. 2015). Virtualization introduces the layer between hardware and operating system. Without utilizing a physical version for any computing infrastructure or resources accessing virtual version of computing resources such as OS, storage device or network components is termed as virtualization (Cardoso et al. 2015). With help of virtualization software physical host machine can create and manage one or many virtual machines on it. Virtual Machines are able to run its own operating systems and applications same as a physical computer. Migrating a virtual machine is nothing but transferring running state of VM from source physical host (source node) to another destination physical host (target node). The active state of VM is transferred from source host to destination host without interrupting network connections. Live migration

✉ Sandeep G. Sutar  
sutarsandeep07@gmail.com

Pallavi J. Mali  
malipallavi07@gmail.com

Amruta Y. More  
amrutaym@gmail.com

<sup>1</sup> Department of Computer Science & Engineering,  
Annasaheb Dange College of Engineering & Technology,  
Ashta, India

<sup>2</sup> OOPRA IT Solutions PVT Ltd, Pune, India

of VM is termed when the original VM resides in running state, while the migration process takes place. Live VM migration proposes small amount of migration time which is beneficial for fast migration. The objectives of proposed system is to lively migrate the virtual machine from overloaded physical machine to under loaded physical machine within minimum time. To generate physical machine's resource utilization profile. To identify overloaded physical server using ant colony optimization algorithm. To migrate the virtual machines lively in minimum down time. To minimize the energy consumption of physical resources.

**Problem statement:** To develop an energy saving mechanism for shared cloud resources using live virtual machine migration technique using algorithmic steps of Ant Colony Optimization in cloud computing environment.

Following are the objectives of proposed work:

- To generate physical machine's resource utilization profile.
- To identify overloaded and under loaded physical server using theory in Ant Colony Optimization.
- To migrate virtual machines lively within minimum down time.
- The minimization of energy required for physical resources.

## 2 Related work

Veeravalli and He (2015), focus on various recent issues in cloud computing areas. Those issues are categorized into different sections i.e. cloud resource management, increase in cloud resource costing and marketing of resources as well as inclusion of Cloud within cloud. Mashayekhy et al. (2014) suggested most advantageous mechanism for managing resources in cloud. Kwak et al. (2015) proposed Dynamic Resource and Task Allocation algorithm for dynamically managing mobile cloud resources. Patel et al. (2014) recommend that an elastic management of cloud resources with green cloud mechanism. They propose DVFS concept for energy management. Cardoso et al. (2015) proposes mechanism that utilization of cloud resources done properly leads to save energy. For saving energy and utilizing resources in efficient way they focus on live VM migration. Also they mention Wake On Lan protocol to turn off and on idle physical machine to save power as per requirement. Agarwal and Raina (2012) proposes migration of active state virtual machines with essential phases of migration. Wen et al. (2015) described ant colony optimization algorithm to detect and balance load on cloud resources by using live VM migration. Biswas et al. (2016) have judged the work of

live migration of virtual machine using high network interfaces. The network topology is based on Openstack cloud environment. Yang et al. (2015) mentioned libvirt function to establish cloud infrastructure using KVM hypervisor. This paper proposes green cloud computing by shutting down idle cloud resources. Dhanoa and Khurmi (2015) have studied the effect of size of virtual machine and bandwidth of network on total migration time required to migrate VM. Live migration mechanism save energy and supports green cloud computing. Kumar and Prashar (2015) have done relative study of algorithms to equal load in cloud computing environment. Wang et al. (2015) mention curious mechanism to increase profit for placing VM in datacenter and diminish number of VM migrations.

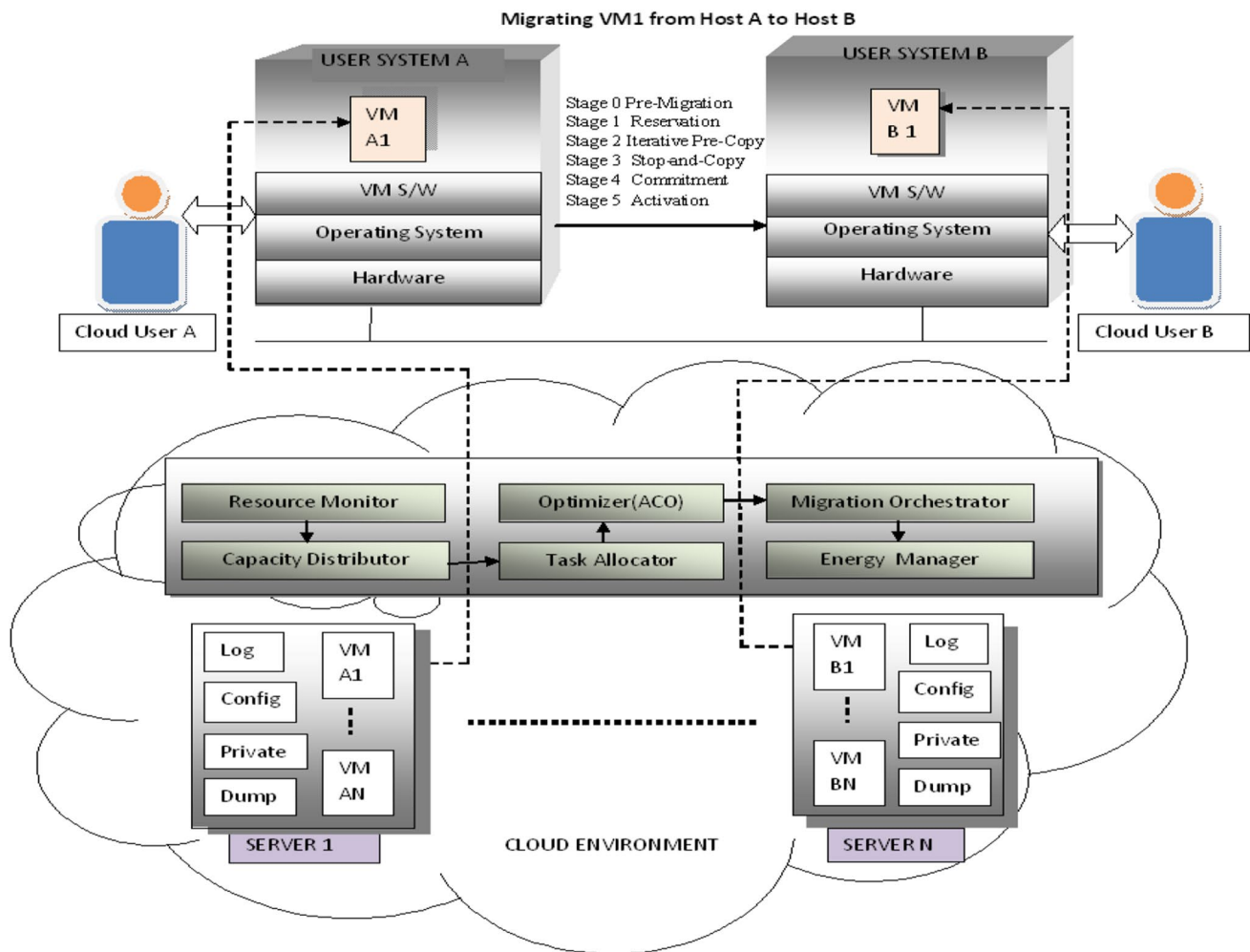
The remaining paper systemized in following way. In Sect. 3, recommend architectural design of proposed system. In Sect. 4 discusses the algorithms and the modules of the system. The Sect. 5 includes experimental results. Lastly, come up with conclusion in Sect. 6.

## 3 System architecture design

As shown in Fig. 1 the proposed system functions in following steps: In this architecture design there are number of physical host and they can include any number of virtual machines running on it. Therefore when a VM is migrating from one physical machine to another PM, the mutual connection is preserved between physical host and virtual machine. Migration of VM takes place in accordance with the proposed modules. The output of proposed module initiates the live migration of VM. The implementation of the proposed modules done at server physical machine in the cloud environment. Above diagram represents the system architecture design of proposed system. Following are the logical phases (Agarwal and Raina 2012) that are executed when migration of VM takes place. While migration of VM it is necessary to safely migrate VM and handle failure if occurs. To achieve this, following phases needs to take place during migration.

**Phase 1. Pre-migration** This phase start with running instance of virtual machine on physical machine A. In this phase the destination physical machine is determined and selected. This selection of destination physical machine helps to reduce total migration time.

**Phase 2. Reservation** During reservation phase the migration request is initiated from physical machine A to physical machine B. It confirms the required resources for successful implementation of VM are available on physical machine B. Then it reserves the resources if available, otherwise VM continues running on physical machine A.



**Fig. 1** System architecture design

**Phase 3. Iterative pre-copy** In this phase while first iteration all the memory pages are transferred from physical machine A to physical machine B. From next iterations remaining pages from previous iterations are transferred.

**Phase 4. Stop-and-copy** This phase stops the VM on source physical machine A and move its network traffic to destination physical machine B. The CPU state and remaining memory pages are redirected to host B. At this phase VM is suspended from host A still it is considered its primary copy in failure cases.

**Phase 5. Commitment** This is commitment phase where the host B sends an acknowledgment to host A that it has successfully received VM. This phase guarantees the successful migration process. After receiving successful acknowledgment from host B, the host A remove VMs copy on it. Host B is now primary machine for VM.

**Phase 6. Activation** In this phase the VM is successfully running on host B.

With the help of above defined phases and modules of the proposed system we migrate the virtual machines lively from overloaded PM to underloaded PM.

## 4 Algorithm and modules

The proposed system has seven modules. The resource monitor collects CPU and Memory usage of physical and virtual resources. Capacity distribution leads to specify threshold values for resources. Task allocation helps to determine overloaded servers. The optimizer helps to reduce physical machines which are in active state. Local Migration Agent helps provides best suited PM to VM to be migrated. Migration Orchestration perform live migration of virtual machine. The energy manager leads to save power consumption of resources.

## 4.1 Resource monitoring

The Resource Monitoring module collects CPU and memory usage profile of cloud resources. The psutil library (python system and process utilities) communicate with physical machine's hypervisor and collect resource usages. The CPU utilization of each virtual machine and physical machine is directly calculated by psutil.cpu\_percent(). The CPU usage profile for physical machine is calculated by aggregating CPU usages of its virtual machines. The process.memory\_info() function provides memory utilization of cloud resources. This resource utilization is saved into the database for future reference.

## 4.2 Capacity distribution

The Capacity distribution distributes the capacity among the physical machines. This is simply sets the upper threshold and lower threshold values for the physical machines. This will help in optimizer module to detect the load.

## 4.3 Task allocation

The Task allocation calculates the execution time and memory required for the coming task according to that the task will be assigned to the machines. According to calculations distribution of task will help us in migration phase. To minimize the migration time task allocator helps.

## 4.4 Optimizer

The Optimizer receive the resource usage profiles with help of Resource Monitor and perform an ant colony optimization algorithm to determine the load amount on physical machines which are in active state. After successful execution of ACO-Based VM Distribution List algorithm it gets a new list of allocation of virtual machines for physical machines. The resulting message of VM redistribution on PM is transmitted to the Migration Orchestration with help of local migration agent.

Algorithm A (ACO-Based VM Distribution List)	Algorithm B (VM Mapping with PM Algorithm)
Step I: <b>if</b> an instance of physical machine loaded it exceeds the threshold <b>then</b>	Step I: Input: $L_{VM}^+, L_{PM}^+$
Step II: Generate sorted list of VMs running on that instance of PM that should migrate	Step II: $Best\ Fit = -\infty$
Step III: <b>while</b> $i < M$ <b>do</b>	Step III: <b>for</b> $\forall vi \in L_{VM}^+$ <b>do</b>
Step IV: Generate ants	Step IV: <b>for</b> $\forall pj \in L_{PM}^+$ <b>do</b>
Step V: ants traverse with Positive Traversing Strategy	Step V: Calculate $Fitness(v_i, p_j)$
Step VI: $i = i + +$ ; $i$ is pheromone	Step VI: <b>if</b> $Fitness(v_i, p_j) > Best\ Fit$ <b>then</b>
Step VII: <b>end while</b>	Step VII: $Best\ Fit = Fitness(v_i, p_j)$
Step VIII: Generate ants	Step VIII: <b>end if</b>
Step IX: ants traverse Negative Traversing Strategy	Step IX: <b>end for</b>
Step X: Generate set of physical machines in low load condition	Step X: <b>if</b> $Best\ Fit > = 0$ <b>then</b>
Step XI: Match the relationship between the PMs with respect to VMs	Step XI: Select the instance of PM as the destination PM
Step XII: Recent mapping relationship is generated	Step XII: Migrate instance of VM
Step XIII: <b>end if</b>	Step XIII: $L_{VM}^+ - \{v_i\}$
	Step XIV: <b>end if</b>
	Step XV: <b>end for</b>
	Step XVI: <b>if</b> $L_{VM}^+ \neq \emptyset$ <b>then</b>
	Step XVII: Reschedule Algorithm 1
	Step XVIII: <b>end if</b>

Algorithm A (Wen et al. 2015) describes ACO-based VM distribution list algorithm. The  $M$  defines the iteration times. Physical machines in high load condition are specified by step III to VIII. Algorithm B (Wen et al. 2015) outputs the sorted list of VMs  $L_{VM}^+ = \{v_1, v_2, \dots, v_i\}$ . This list defines VM should be migrated. Also it generates the list of destination PMs  $L_{PM}^+ = \{p_1, p_2, \dots, p_k\}$ . This list defines possibly low load condition PM. With the help of these two sets, it initiates the matching of VMs with the PMs.

## 4.5 Local migration agent

When the load condition of an instance of the physical server machine surpass the upper threshold value, the local migration agent will rearrange all VMs from instance of that PM according to their moderate load. This load is evaluated by the arithmetic Formula 1 given below. The higher loaded virtual machine has greater priority.

$$\overline{Vi(j, T)} = \frac{1}{T} \sum_{k=1}^n Vi(j, k)(tk - tk - 1) \quad (1)$$

$Vi(j, k)$  defines the resources acquired by the  $j$ -th instance of VM on the  $i$ -th instance of PM in time period  $k$ , is calculated by

$$Vi(j, k) = \frac{1}{k} \left( \overline{U}_{cpu}^2 + \overline{U}_{mem}^2 + \overline{U}_{net}^2 \right) \quad (2)$$

where,

$$K = \left( \bar{U}_{cpu}^2 + \bar{U}_{mem}^2 + \bar{U}_{net}^2 \right) \tag{3}$$

$$\bar{U}_{cpu} = \frac{1}{2} \left( U_{cpu_k} + U_{cpu_{k-1}} \right) \tag{4}$$

$$\bar{U}_{mem} = \frac{1}{2} \left( U_{mem_k} + U_{mem_{k-1}} \right) \tag{5}$$

$$\bar{U}_{net} = \frac{1}{2} \left( U_{net_k} + U_{net_{k-1}} \right) \tag{6}$$

- $U_{cpu_k}$ : CPU resource consumption by the  $j$ -th instance of VM on the  $i$ -th instance of PM in the time interval  $k$ , and it is similar for the terms  $U_{mem_k}$ , as well as  $U_{net_k}$ .
- $U_{cpu_k}$ ,  $U_{mem_k}$ , and  $U_{net_k}$ : constants.

The load on physical machine instance  $P_i$  is calculated by following formula,

$$P(i, T) = \sum_{j=1}^{m_i} \overline{Vi(j, T)} \tag{7}$$

where,

- $m_i$ : defines the count of virtual machines on the instance of physical machine.

Repeatedly, the Local Migration Agents observes the load on the physical machine. If the physical machine is overloaded then migration takes place. The upper and lower threshold values are set by capacity distributor (Wen et al. 2015).

### 4.6 Migration orchestration

The Migration Orchestration organize the migration of the virtual machines. The actual migration of virtual machines lively is carried out by migration orchestration. The Orchestration uses vMotion feature of vSphere Client and Vcenter Server Appliance. Following factors are responsible to affect the migration time: VM size, network traffic, physical machine resource usage. Virtual machines are migrated one by one to reduce overload of network traffic.

### 4.7 Energy management

After execution of all above modules the idle physical systems that are identified via previous module needs to shut down to save energy. The idle physical machine is one that has no VM running on it. But this module is unable to start the physical machines whenever there is need to satisfy increasing demands of clients for acquiring resources. So

it is suggested to turn physical machine into sleep mode whenever it is needed.

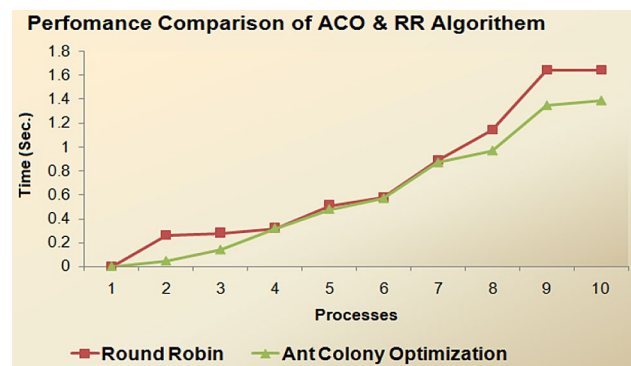
## 5 Experimental results

This section consists with experiments have been supervised to analyse the efficacy for proposed system. The proposed ant colony optimization algorithm (Wen et al. 2015) is compared with existing Load Balancing algorithm. Here its also examined proposed system in terms of migration time acquired to migrate a virtual machine. To evaluate our proposed system we set up virtualized cloud environment with Vmware workstations, Vmware vSphere Client, NAS storage and Vcenter server appliance. Our experiment is mainly based on random workload designed to evaluate the proposed system. In the experimental environment there are four physical machines involved. Those physical machines have 8 GB of RAM and Core i5-6200U. Three ESXi virtual machines of 4 GB RAM are created on these physical machines. One physical machine is worked as Vcenter Server Appliance. One virtual machine is installed for NAS storage. To demonstrate the superiority of proposed system, the consumption of CPU is generated stochastically.

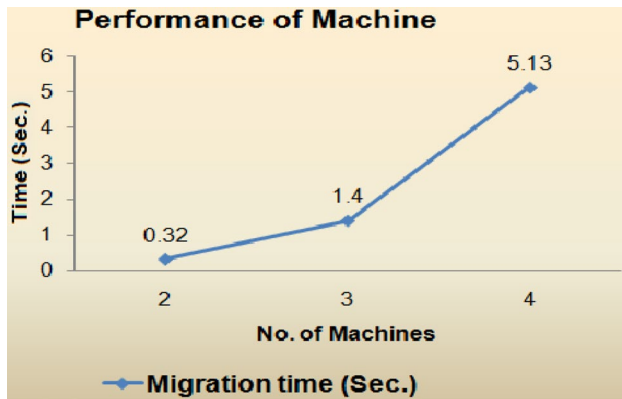
Comparison of Ant Colony Optimization (ACO) algorithm (Wen et al. 2015) with the Round Robin (RR) load balancing algorithm (Kumar and Prashar 2015) has been done. The functioning of these algorithms has examined with respect to time required to balance the same amount of inputted data.

Graph 1 shows that the ACO algorithm requires the minimum time to detect and balance processes. This specifies that ACO is better than Round Robin algorithm.

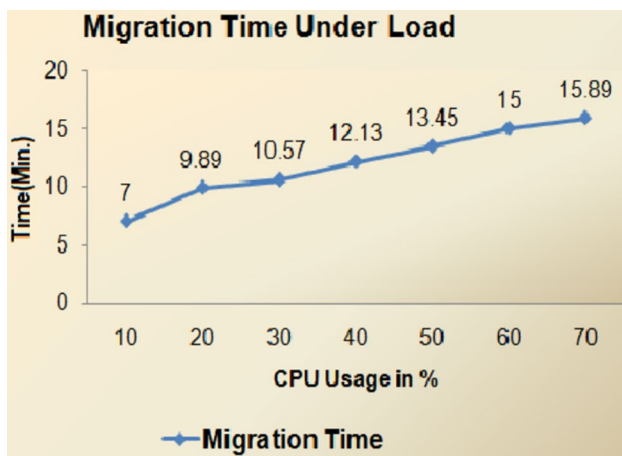
Performance of VM migration with respect to time required for migration is evaluated by considering following factors.



Graph 1 Performance comparison of RR and ACO algorithm



**Graph 2** Performance evaluation of VM migration when physical machines increases



**Graph 3** Performance evaluation of VM migration when CPU usage increases

### 5.1 Increase in number of physical server machines

The above graph says that as the number of physical servers in the network increases the migration time is increased. In this proposed system minimum 2 machines are required for migration of virtual machines so in the above graph minimum 2 physical machines are there. When the migration is carried out between 2 physical machines 0.32 s are required to migrate a virtual machine. As there is increase in physical machine i.e. 3 and 4 the migration time increases to 1 min 4 s and 5 min 13 s respectively. So this graph depicts that as there is increase in number of physical machines the migration time increases (Graph 2).

### 5.2 Increase in CPU usage of physical machine

Graph 3 shows that while migrating a virtual machine from source physical machine to destination physical machine the migration time is affected by the CPU usage of the source

**Table 1** Performance evaluation of CPU usage (before migration) in VM migration

Source machine type	PM name	VM migrated	CPU usage (%)
Source machine	ESXi_Server1	VM11	90
Destination machine	ESXi_Server2	VM11	45

**Table 2** Performance evaluation of CPU usage (after migration) in VM migration

Source machine type	PM name	VM migrated	CPU usage (%)
Source machine	ESXi_Server1	VM11	60
Destination machine	ESXi_Server2	VM11	75

physical machine. Consider CPU usage of source physical machine in terms of percentage and migration time in minutes. As the CPU usage of physical machine's are increased due to its own performance and applications the time required to migrate VM is increasing. When CPU usage of physical machine is 10% the VM migration time is 7 min and as per the CPU usage is increased upto 50% the migration time required is 13 min and 45 s. From the above graph it shows that as CPU usage are increases the migration time increases.

### 5.3 Performance evaluation of CPU usage in VM migration

The performance of physical machines are analyze when CPU usage is increases while migration of virtual machine. With the help of ACO algorithm detect the load on physical server then it is decided to migrate the virtual machine to appropriate physical server. Hence, before migrating VM the CPU usage profile of source physical machine gets higher and after migration of virtual machine CPU usage profile of source physical machine gets reduced. Same way before migration of virtual machine on the destination physical machine's CPU usage is lower whereas after migration of virtual machine the destination machine's CPU usage is getting increased. This is shown in the following table.

As shown in the above Table 1 Virtual machine named VM11 is being migrated from source server machine ESXi\_Server1 to destination server machine ESXi\_Server2. Before migration of virtual machine VM11 the CPU usage of ESXi\_Server1 is 90% and is reduced after migration upto 60%. Where as the destination machine's CPU usage before the virtual machine is placed over ther 45% and after virtual machine VM11 is placed on that destination server machine CPU usage is 75%. In this way we can say that with the

help of virtual machine migration we can balance the load on the physical server machines. By analyzing from above Tables 1 and 2 it is said that CPU load is balanced after virtual machine is migrated.

## 6 Conclusion

Our system reduces the energy consumption by lively migrating virtual machines in cloud environment. It provides minimum migration time for virtual machine migration. This work results into proper utilization of available physical resources in cloud environment leads to save energy consumption. It is concluded that virtual machine are lively migrated successfully leads to save energy consumption.

As a future work we plan to focus on cleaning the VMs by removing unnecessary processes. It will reduce the size of VM before migration. This leads to reduce the migration time. Also it is intended to improve the threshold value for load detection from a predetermined value to an elastic value.

## References

- Agarwal, A., & Raina, S. (2012). Live migration of virtual machines in cloud. *International Journal of Scientific and Research Publications*, 2(6), 1–5.
- Biswas, M. I., Parr, G., McClean, S., Morrow, P., & Scotney, B. (2016, March). A practical evaluation in openstack live migration of vms using 10 gb/s interfaces. In *2016 IEEE symposium on service-oriented system engineering (SOSE)* (pp. 346–351). IEEE.
- Cardoso, L. P., Mattos, D. M., Ferraz, L. H. G., Duarte, O. C. M., & Pujolley, G. (2015). An efficient energy-aware mechanism for virtual machine migration. IEEE
- Dhanao, I. S., & Khurmi, S. S. (2015, May). Analyzing energy consumption during VM live migration. In *International conference on computing, communication & automation* (pp. 584–588). IEEE.
- Kumar, R., Prashar, T. (2015, June) Performance analysis of load balancing algorithms in cloud computing. *International Journal of Computer Applications* (0975–8887) 120(7)
- Kwak, J., Kim, Y., Lee, J., & Chong, S. (2015). DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems. *IEEE Journal on Selected Areas in Communications*, 33(12), 2510–2523.
- Mashayekhy, L., Nejad, M. M., & Grosu, D. (2014). Physical machine resource management in clouds: A mechanism design approach. *IEEE Transactions on Cloud Computing*, 3(3), 247–260.
- Patel, V. J., & Bheda, H. A. (2014). An advanced survey on research issues of energy management in cloud computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(1)
- Veeravalli, B., & He, B. (2015). Guest editors' introduction: Special issue on economics and market mechanisms for cloud computing. *IEEE Transactions on Cloud Computing*, 3(3), 245–246.
- Wang, X., Yuen, C., Hassan, N. U., Wang, W., & Chen, T. (2015, June). Migration-aware virtual machine placement for cloud data centers. In *IEEE ICC 2015—workshop on cloud computing systems, networks, and applications (CCSNA)*.
- Wen, W. T., Wang, C. D., Wu, D. S., & Xie, Y. Y. (2015). An ACO-based scheduling strategy on load balancing in cloud computing environment. IEEE
- Yang, C. T., Chuang, C. L., Liu, J. C., Chen, C. C., & Chu, W. C. (2015, March). Implementation of cloud infrastructure monitor platform with power saving method. In *2015 IEEE 29th international conference on advanced information networking and applications workshops* (pp. 223–228). IEEE.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.