International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4S2, December 2019

# A Two-Stage Heuristic Hybrid Evolutionary Algorithm for Virtual Machine Migration and Placement in Cloud Bursting

**Srividhya V, S. Balamurali**

***Abstract—The platform for cloud computing offers virtualization and a dynamic pool of resources to the consumers of the cloud. The acceptance and demand of cloud is growing on a regular basis. Cloud computing offers utility-based consumer services across the globe on a pay as you go strategy. Live Virtual Machine (VM) migration sets the basic foundation for cloud management. It hasakey role in reducing operating expenditure and revising quality of service without disruption of cloud services running in the VM. A lot of research has been done to yield better performance in live VM migration and has seen noteworthy development and accomplishment. However, some crucial problems require results and enhancement. With the growth of new cloud computing models like Mobile Edge Computing, certain problems related to optimization need to be addressed. The primary aim of this research work is to emphasize on optimum functioning of live migration. A migration algorithm to consolidate the computational resources, storage resources and network resources dynamically with a two-stage heuristic hybrid evolutionary algorithm is discussed. The resources are consolidated to cut down the energy and cost utilization depending upon the evolutionary Particle Swarm Optimization and the Ant Colony Optimization algorithms. These algorithms can rapidly identify the migrating virtual machines and locate their positions respectively.***

***Keywords— Cloud bursting, Cloud computing, VM Migration.***

## I. INTRODUCTION

This Most establishments have financed profoundly on information technology (IT) data centers that encompass compute as well as storage systems. In spite of this heavy investment incurred on the IT infrastructure, it quite often becomes insufficient to the computational needs especially when there is a planned or unexpected workload peak in the enterprise applications. Instead of investing on additional server capacity to manage spikes in workload, a hybrid model of cloud has evolved where organizations make use of the private data centers for most of the requirements and complements with added cloud resources when the local resources are over-utilized. This hybrid model of cloud computing is called cloud bursting. It allows organizations to surge or lessen its capacity as and when required and also optimally utilizing the existing infrastructure. While commercial and open-source virtualization appliances are beginning to bear the primary cloud bursting practices [1-3], the crucial attention is on the methodologies that support migration of virtual machines from one location to another. In such systems, important decisions are taken by the system administrators, who manually identify the appropriate time for the invocation of cloud bursting functionality and also the right application to burst. Manual decision making may cause poor choices in reducing cloud expenses and minimising down time during migration. This is due to lack of knowledge of system administrators. Carefully choosing the applications for migration is troublesome mainly as there are a sizeable quantity of varied applications in the data centers with a wide range of pricing models.

Bursting within the cloud goes by the supposition that both the local and cloud data centers use virtualization. When an application is bursted into the cloud, it requires copying the image of virtual disk and data, if any. In most cases the disk state is huge and transfer on a need basis to cloud needs to replicate volumes of data which might consume hours. The proposed system auto-detects the overload in the local data centers and identifies applications that may be migrated to the cloud with minimal expense and do the transitions required to scale the capacity at run-time as effectively as possible. With this automation, the new system quickly responds to workload spikes.

With the advent of virtualization, a physical server may be partitioned into multiple execution settings by deploying a VM manager also known as the hypervisor mounted above the operating system (OS). Each VM has come with its own set of applications and an OS. Using the cloud computing paradigm, enterprises can share their available resources by renting them to the consumers as a utility model in a pay per use style. This feature popularly known as multi-tenancy where users work on a VM and cutting down on the expenses which include both hardware acquisition and its maintenance. As we witness the sprawling demand where users opt cloud data centers to host their applications, it has become more important to manage the data center VMs in an efficient manner. As we see users requesting for resources from the cloud service provider, they also leave once their job is done. Accordingly, the cloud

**Revised Manuscript Received on December 05, 2019.**

*** Srividhya V**

**Srividhya V**, Department of Computer Applications, Kalasalingam Academy of Research and Education, Krishnankoil, Virudhunagar, India -626126. Email: srividhya.vp@gmail.com

**Dr. S. Balamurali**, Department of Computer Applications, Kalasalingam Academy of Research and Education, Krishnankoil, Virudhunagar, India-626126. Email: sbmurali@rediffmail.com

service continues to produce and abolish VMs within data centers. We notice that cloud service providers running a multitude of data centers across the globe. For the provision of state-of-the-art services, collaboration of these numerous data centers is mandatory. It is always ideal to have a VM which is nearest to the consumers' current location. This helps in diminishing network latency. If the data centers are not managed efficiently, there might be servers running with minimal workload while there could be other data centers that are heavily loaded. These glitches are very well managed with the help of this key technology known as live VM migration

VMs may be shifted from server to server, starting at a given data center and moving on to another, without disrupting the VM applications under execution. Sizeable amount of operations pertaining to cloud management are viable as a result of VM migration – live. Even as VM migration research has seen major developments, there still exist challenges waiting for results and enhancements. It may be noticed that challenges exist in three different kinds of VM migration namely, migration in Local Area Network (LAN), Wide Area Network (WAN) or Mobile Edge Computing (MEC).

Live migration is crucial for efficient cloud management. VM consolidation within data center cuts down on operational expenses. VMs that interconnect with each other recurrently may be shifted to the same host. This enables efficient communication along with decreased network traffic inside a data center. With the arrival of excellent migration environments in data centers including high network bandwidth and shared storage, VM relocations within LAN is at its best with non-existing downtime and minimal migration time as a result of a wide range of optimizations. In spite of these advancements, problems still persist. The challenges include uncontrollable and transparent migration process both to the cloud managers and the consumers. Undue importance is given to migration performance while the requirements of the user is given least priority. Also, migration convergence problem is a by-product of the wide usage of Pre-copy [4].

Cloud management is elevated to greater heights with the migration of VMs across data centers. This demands cooperation among cloud service providers. Again, the service provides running a multitude of data centers can implement load balancing amongst its data centers. Yet, because of the large size of virtual disk of a VM and the relatively much smaller network bandwidth, storage data migration remains a challenge. To overcome this challenge, a network file sharing system is deployed. But this leads to huge disk input/output delay. This model increases the network traffic heavily. In another technique, even as each and every data center uses its own storage system, storage data migration is improvised using multiple methods such as data deduplication.

Mobile Edge Computing (MEC) was introduced to offer low-latency cloud computing due to increased use of User Equipment (UE) like smart phones and the growth of Internet of Things (IoT). As UEs roam from one coverage area to another, its VM must be migrated to reduce latency. Storage migration which was an issue in WAN continues in MEC as well. Along with which, inconsistency in UE mobility and VM migration leads to a pause in the services running. Hence, coordination strategy is mandatory to link VM migration and UE mobility. Mobility sets in a huge network overhead to MEC due to persistent VM migration.

A typical data center includes multiple servers while each server holds many VMs. Consumers utilize resources of any data center and leave the same upon completing their task. The data center in turn creates and destroys VMs for the sake of users. Proper management is quintessential for the VMs of a data center. VMs need to be migrated to servers with relatively lesser load.

## II. RELATED WORK

The term 'Cloud Bursting' was first suggested by Amazon's Jeff Barr as a way to allow organizations that already own significant amounts of IT infrastructure to continue to make use of the cloud during peaks in workload [5]. Cloud bursting goes by the presumption that a private data center is connected to public cloud, creating what is popularly known as a hybrid-cloud. Hybrid clouds have become a popular service offering for hosting and data center companies, and also have been the subject of research [6, 7].

The VM placement problem has gained a lot of importance due to server virtualization. It helps to enhance utilization of resources and minimizes power consumption along with cost. The procedure that selects the ideal Physical Machine (PM) for running a VM. Designing suitable mapping of VM to PM is the goal of VM placement algorithms.

Cloud providers make use of varied VM placement techniques. Silva Filho et al. [8] give a completeresearch survey on placement of VM, optimization and relocation in cloud set up. They have deliberated the problem formulations, benefits, and inadequacies of relevant research. A variety of goals for VM placement have been well-thought-out in preceding research, which includes power saving, minimizing the cost, load balancing, decreasing SLA violations, network latency, congestion, and service down time[8]. As per the goals of placement, VM placement approaches can be differentiated into two categories [9]:

- Power-based method: Identification of power-efficient VM-PM mapping based onutilization of resources [10–12].
- QoS-based approach: Identification of a VM-PM mapping using the thorough fulfilment of the requirement of service quality [11–13]. As per the type of optimization techniques applied to obtain a VM-PM mapping, the methods of VM placement may be broadly divided into six categories [9,14]:
- Heuristic Bin Packing: The VM placement problem is expressed with vector bin packing. VMs are treated as small items that are closely packed into a least number of bins, each treated a PM. A variety of heuristic approaches are

established to estimate the optimal solution to this packing problem.

- Evolutionary algorithm-based optimization: there exist many bio-inspired optimization methods like ant colony optimization method, the self-adaptive particle swarm optimization scheme, and genetic algorithms are implemented to pack VMs into a minimal count of PMs, for a given workload.
- Linear programming: The problem of placing VMs is built with the help of a linear programming problem which studies a number of limitations resulting from practical applications. Methods based on LP-relaxation are established for solving the expressed model.
- Constraint programming: Van et al. [15] have offered a resource management outline, which contains a dynamic utility-based manager for placing VMs and a dynamic manager providing VMs, to obtain an appropriate VM-PM mapping. Both these management tasks are viewed as problems involving fulfilment of constraints. More real-world aspects can be deliberated by extending the constraints in these problems.
- Stochastic integer programming: Due to the future demand of VM for providing network services is indeterminate, the stochastic integer programming technique is used to forecast an appropriate VM-PM mapping.
- Simulated annealing optimization: Liao et al. [16] have suggested an outline for mapping at the time of execution that acquires a simulated annealing optimization algorithm to map VMs to a minimal set of PMs to maximise without significant degradation in system performance.

Usmani & Singh [9] and Zheng et al. [14] have swotted advanced VM placement and consolidation approaches. The task with the VM placement problem is efficient identification of an optimal solution, particularly when the VM placement is NP-hard. A multitude of VM placement methods only reflect only upon the processor and storage resources and ignore networking restrictions. In any case, interaction between networking devices running on servers positioned in diverse data centers make the inadequate network bandwidth an issue of significance while allotting VMs to PMs. Few studies [17-19] have explored network requirements along with that of processor and storage requirements.

### A. A Two-stage hybrid evolutionary algorithm

Input required: list of VMs and PMs
Output generated: List for migration
Step 1. Rate every PM by evaluation of the PMs
Step 2. Categorize the PMs into over-utilized PMs or under-utilized PMs based on the rating
Step 3. Select the uppermost VM depending upon the given formula

$$\text{Distance} = \sqrt{(CPU\,\%)^2 + (MEM\,\%)^2} \qquad (1)$$

in the over-utilized PMs using the Particle Swarm Optimization Algorithm
Step 4. from the under-utilized PMs select all the VMs
Step 5. With Steps 3 and 4 generate the list for migration
Step 6. An improvised ACO procedure is used to put the identified VMs upon evaluation of the optimal location depending upon a multi-objective optimized model
Step 7. Transfer the control back to the rating model.

### B. Hybrid Algorithm

The algorithm is hybrid and it is an evolutionary algorithm with two stages. The first among the two phases evaluates the threshold. The VMs are then determined by the Particle Swarm Optimization procedure. Second phase solves the problem of placing VMs, which is done with the help of ACO algorithm.

## III. PROPOSED ALGORITHM

A Two-Stage Heuristic Hybrid Evolutionary Algorithm is proposed in this section. This intends to evade violations of SLA and minimises power utilization in the process of scheduling. The algorithm is divided into three parts.

Activation, Selection and location part. In activation part, hotspots are evaluated using a rating model. It explains the point in time at which the overloaded PMs are to be migrated. The VMs from the heavily utilized PMs which are relatively available as they are not used in the selection part. Apart from this, the VMs are selected from the hotspots with PSO. The migrated VMs are placed into the chosen slots with the help of Ant Colony Optimization (ACO) algorithm which proves to be better. The two-stage evolutionary algorithm which is hybrid in nature is given below.

### C. The proposed ACO algorithm

**Required Input**: the chosen VM = {$VM_i$ | i = 1,2,3,…..,n}
**Generated Output**: associate the chosen VMs to a host that is reasonable
Step 1. Set initial values for all of the ants
Step 2. The path is chosen arbitrarily
Step 3. While (*iteration < iteration maximum*)
Step 4. For each and every ant
Step 5. Gauge the fitness function by using the following equations

$$\tau_{iu} = (1-\rho)\times\tau_{iu} + \Delta\tau_{iu}^{best} \qquad (2)$$

where

$\tau_{iu}$ – *Pheromone*

$\rho$ – *Pheromone evaporation factor*

$\Delta\tau_{iu}^{best}$ – *Incremental Gain*

and

$$\Delta\tau_{iu}^{best} = \begin{cases} Fit_{best}\ \text{if}\ plan(VM_i, PM_u) \in path \\ 0 \quad otherwise \end{cases} \qquad (3)$$

where

$Fit_{best}$ – *Multiobjective Fitness Function*

Step 6. if value of fitness is greater,
Step 7. The fitness value is assumed to be the current value
Step 8. Update the pheromone
Step 9. End if

Step 10. Identify path of behavior chances with Equations

$$P_{iu}^{k}(t)=\begin{cases}\dfrac{\tau_{iu}^{\alpha}\times\eta_{iu}^{\beta}}{\sum_{s\in allow_k}\tau_{iu}^{\alpha}\times\eta_{iu}^{\beta}} & i\in allow_k\\ 0 & otherwise\end{cases} \quad (4)$$

and

$$\eta_{iu}(t)=\begin{cases}\sum_{s\in allow_k} Variance(i,u) & i\in allow_k\\ 0 & otherwise\end{cases} \quad (5)$$

where

$\eta_{iu}$ - heuristic information

Step 11. End For
Step 12. End iterations
Step 13. Compare all solutions of the algorithm upon various parameters
Step 14. Stop

### D. Investigation

A simulated investigation is carried out in this research paper. The experiment is implemented on the CloudStack platform to authenticate the soundness of the proposed algorithm. The outcomes reveal that the projected algorithm increases not only the utilization of the CPU and memory, it also brings down violation of SLA and consumption of energy.

### E. Investigation Settings

The suggested algorithm is implemented with seven physical machines. While one physical machine is mounted on CloudStack, six of the remaining physical machines run on top of XenServers. Eighteen virtual machines are generated in the cluster. The investigation set up is split into three sections. The first section is to assess the method, we put it under varying workloads. In the second section, performance is checked. Performance measures are defined which comprises ratio of SLA violation, ratio of energy consumption and ratio of resource wastage. In order to assess the proposed algorithm effectively, it is implemented using heuristic algorithms for comparison.
Investigation environment: to check the suggested algorithm, two divisions in the investigation are established. In the first investigation, workload generator memtester [20] is used to create the processor and storage workloads step by step. This is depicted in Fig. 1.

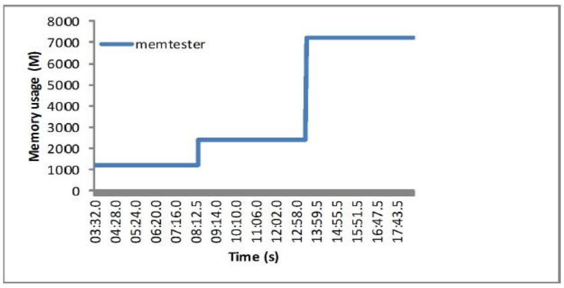


Fig 1. Load designed for memtester

TPC-W benchmark [21] is used in the second stated set of investigations to generate the application requests. Workload traces from web requests like EPA [22] and NASA [23] are implemented in the second set experiments. Jmeter is used to create the simulation and is seen in Fig 2. More parameters like processor and storage utilization is monitored by the Jmeter plugin.
Measures: The investigations make use of different assessors as measures. They are violation ration of SLA *slaratio*, utilization of resources *Ucpu*, ratio of energy consumption *pratio* and ratio of resource wastage *wratio*. The percentage of SLA violation is given in (6). If the utilization of CPU is more than 0.8, violation of SLA has the highest probability. The ratio of energy consumption is given in (7). The ratio of wastage of resources is given in (8).

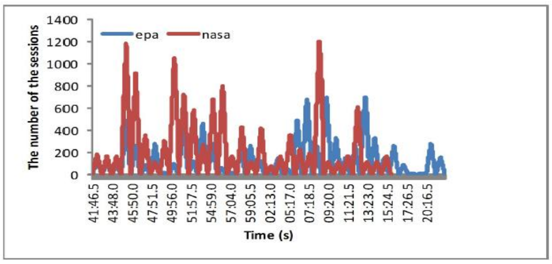


Fig 2. Simulated realistic workloads

$$sla=\frac{1}{1+e^{(U_{cpu}-0.8)}},\ sla_{ratio}=\frac{\sum_{i=1}^{k} sla_i}{total_{sla}} \quad (6)$$

$$p_{ratio}=\frac{\sum_{i=1}^{k} P_i}{total_P} \quad (7)$$

$$w_{ratio}=\frac{\sum_{i=1}^{k} W_i}{total_W} \quad (8)$$

*k* indicates the server count.
*Pi* and *Wi* are the proportion of wastage of power and resources.
$total_W$ indicates total wastage of resources
and $total_P$ indicates total energy consumption.

### F. Investigation results

The suggested algorithm is compared with the following algorithms

- Single objective algorithm (ACO-U)
- Double objective algorithm (ACO-UP) [24]
- Multi-objective algorithm (MACO) [25].

MPSO resembles MACO. They make use of similar fitness functions, however MPSO is implemented using PSO algorithm. ACO-U is similar to FF (first fit) algorithm. The suggested algorithm tries to parallel reduce wastage of resource and reducing SLA violation and consumption of energy.
The workload includes three divisions. The first division (Group 1) indicates the workload with lesser variation.

This is implemented with memtesters executing in two virtual machines at almost full utilization. The second division (Group 2) suggests workload with slightly higher variation with memtesters executing in four virtual machines at almost full utilization. The third division (Group 3) suggests workload with higher variation where memtesters are executed in eight virtual machines at almost full utilization. The suggested algorithm is assessed on four measures namely ratio of SLA violation, utilization of resources, consumption of energy and ratio of resource wastage.

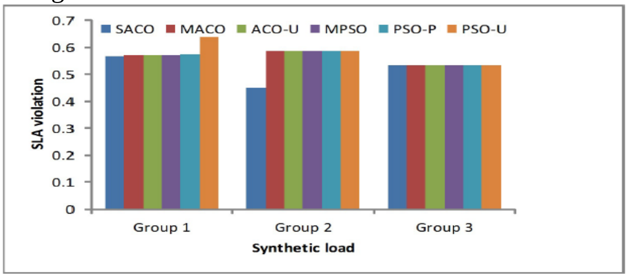


Fig 3. Synthetic loads – SLA violation rates

be the best among the other algorithms from the view of SLA violation the proposed algorithm has both merits as well as demerits. SLA violations are lesser in division 1 and division 2, however with higher workload, it is similar to the other algorithms.

Utilization of resources: reduction of tenant cost and increasing utilization is the major goal of resource utilization. Fig 4 proves that the proposed SACO algorithm is better in comparison with other algorithms.

With lighter workload consolidation results are higher compared to other algorithms. With medium workload, the suggested solution yields better outcomes compared to the other algorithms as it increases the utility of resources and minimizes tenancy cost.

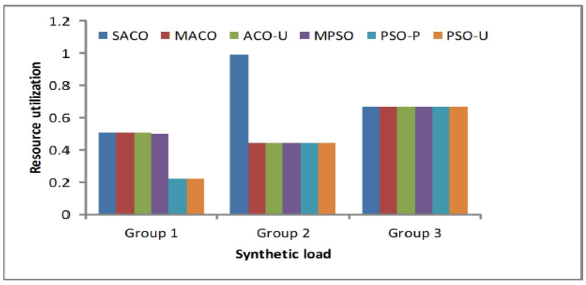


Fig 4. Synthetic loads – CPU resource utilization

With heavy work load the suggested algorithm is almost same as other algorithms. The outcomes of the three divisions are rational and exhibit the efficiency of the proposed algorithm.

Consumption of energy: As given in fig 5, the suggested algorithm achieves superior outcomes due to reduced consumption of energy. The algorithm is rational in case of synthetic workloads and gains superior results during the scheduling process.

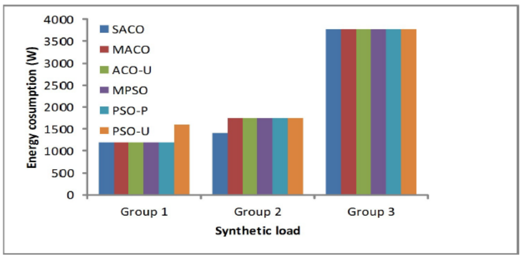


Fig 5. Synthetic workload – Energy Consumption

Resource wastage ratio: The suggested algorithm discusses about CPU and memory utilization. The algorithm yields minimal wastage of resources in Group 1. In second division (group 2) and the third division (group 3), it is similar to other algorithms. The proposed algorithm gains marginally better outcomes compared to other algorithms.

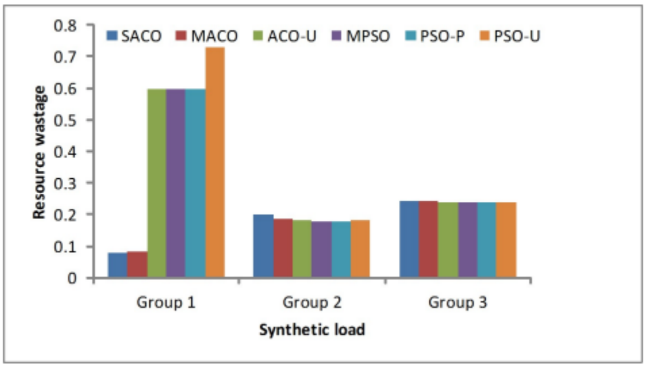


Fig 6. Synthetic workloads – Resource wastage rates

The projected algorithm is authenticated upon evaluation with different algorithms from various perspectives

- The single objective method combines the resources with the intent of decreasing energy consumption. The implemented is done using ACO algorithm. Yet another objective is to optimize utilization through bin-packing algorithm upon research allocation
- In Double-objective algorithm consolidation of the resources is dependent upon two goals, namely energy consumption and resource wastage. The suggested algorithm is executed by using the ACO algorithm.
- The suggested ACO algorithm to combine the resources based on various goals, like power utilization and violation of Service Level Agreements. An upgraded algorithm which is multi-objective and that depends on the projected ACO algorithm, which keeps track of SLA violations, wastage of resources and utilization of energy with diverse weighing parameters.

## IV. CONCLUSION

Existing scheduling methods concentrate on the energy-efficiency model to minimize the load. But features that influence the scheduling process are present. In this research paper, a two-stage hybrid evolutionary algorithm is presented to achieve reasonable cost and energy conservation. The first stage evaluates the hotspots by using a rating model and then shifts the VMs by using the PSO algorithm. The second stage discovers the sites by using the upgraded ACO algorithm, which concurrently attempts to reduce the rental charge and the power consumption. To augment the complexity of this study, more research in the future will focus on several aspects. To begin with, extra features influence dynamic scheduling problem, such as the temperature and CPU rate. The scheduling algorithm can be applied in intricate settings, like workflows in IaaS. Degradation of performance is another future research direction. For instance, with time, corruption of data and resource exhaustion can bring about degradation in performance.

## REFERENCES


1. openstack: Cloud Software. http://www.openstack.org.
2. Open Nebula: The Open Source Toolkit for Data Center Virtualization. http://www.opennebula.org.
3. VMware: Public & Hybrid Cloud Computing. http://www.vmware. com/ solutions/cloud-computing/public-cloud/products.html.
4. Cooper, Brian F., Silberstein, Adam, Tam, Erwin, Ramakrishnan, Raghu, and Sears, Russell. Benchmarking cloud serving systems with ycsb. In Proceedings of the 1st ACM symposium on Cloud computing (New York, NY, USA, 2010), SoCC '10, ACM, pp. 143–154.
5. Chen, Jin, Soundararajan, Gokul, and Amza, Cristiana. Autonomic Provisioning of Backend Databases in Dynamic Content Web Servers. In ICAC (June 2006), pp. 231–242.
6. Mateescu, Gabriel, Gentzsch, Wolfgang, and Ribbens, Calvin J. Hybrid computingwhere hpc meets grid and cloud computing. Future Gener. Comput. Syst. 27 (May 2011), 440–453.
7. Sotomayor, B., Montero, R.S., Llorente, I.M., and Foster, I. Virtual infrastructure management in private and hybrid clouds. Internet Computing, IEEE 13, 5 (sept.oct. 2009), 14 –22.
8. Silva Filho, M.C.; Monteiro, C.C.; Inácio, P.R.M.; Freire, M.M. Approaches for optimizing virtual machine placement and migration in cloud environments: A survey. J. Parallel Distrib. Comput. 2018, 111, 222–250. [CrossRef]
9. Usmani, Z.; Singh, S. A survey of virtual machine placement techniques in a cloud data center. Procedia Comput. Sci. 2016, 78, 491–498. [CrossRef]
10. Song, W.; Xiao, Z.; Chen, Q.; Luo, H. Adaptive resource provisioning for the cloud using online bin packing. IEEE Trans. Comput. 2014, 63, 2647–2660. [CrossRef]
11. Beloglazov, A.; Buyya, R. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. IEEE Trans. Parallel Distrib. Syst. 2013, 24, 1366–1379. [CrossRef]
12. Ghobaei-Arani, M.; Rahmanian, A.A.; Shamsi, M.; Rasouli-Kenari, A. A learning-based approach for virtual machine placement in cloud data centers. Int. J. Commun. Syst. 2018, 31. [CrossRef]
13. Bobroff, N.; Kochut, A.; Beaty, K. Dynamic placement of virtual machines for managing SLA violations. In Proceedings of the 10th IFIP/IEEE International Symposiumon Integrated Network Management, Munich, Germany, 21–25 May 2007; pp. 119–128.
14. Zheng,Q.; Li,R.; Li,X.; Shah,N.; Zhang,J.; Tian,F.; Chao,K.M.; Li,J.Virtual machine consolidated placement based on multi-objective biogeography-based optimization. Future Gener. Comput. Syst. 2016, 54, 95–122. [CrossRef]
15. Van, H.N.; Tran, F.D.; Menaud, J.-M. Performance and power management for cloud infrastructures. In Proceedings of the IEEE 3rd International Conference on Cloud Computing (CLOUD 2010), Miami, FL, USA, 5–10 July 2010; pp. 329–336.
16. Liao, X.; Jin, H.; Liu, H. Towards a green cluster through dynamic remapping of virtual machines. Future Gener. Comput. Syst. 2012, 28, 469–477. [CrossRef]
17. Guo,C.;Lu,G.; Wang,H.J.; Yang,S.; Kong,C.; Sun,P.; Wu,W.; Zhang,Y.SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees. In Proceedings of the 6th International Conference (Co-NEXT '10), Philadelphia, PA, USA, 30 November–3 December 2010. Article No. 15.
18. Ballani, H.; Costa, P.; Karagiannis, T.; Rowstron, A. Towards predictable datacenter networks. ACM SIGCOMM Comput. Commun. Rev. 2011, 41, 242–253. [CrossRef]
19. Xie, D.; Hu, Y.C. The only constant is change: Incorporating time-varying network reservations in data centers. ACM SIGCOMM Comput. Commun. Rev. 2012, 42, 199–210. [CrossRef]
20. Clark, C., Fraser, K., Hand, S., Hansen, J. G, Jul, E., Limpach, C., Pratt, I., and Warfield, A. Live migration of virtual machines. In Proceedings of NSDI (May 2005).
21. Menascé D. A. TPC-W: A benchmark for e-commerce. IEEE Internet Computing. 2002; (3): 83–87.
22. Traces in the Internet Traffic Archive [EB/OL]. Available from: http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html.
23. Traces in the Internet Traffic Archive [EB/OL]. Available from: http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html.
24. Gao Y., Guan H., Qi Z., Hou Y., & Liu L. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. Journal of Computer and System Sciences. 2013; 79(8): 1230–1242.
25. Ma F., Liu F., & Liu Z. Multi-objective optimization for initial virtual machine placement in cloud data center. Journal of Information &Computational Science. 2012; 9(16): 5029–5038.


## AUTHORS PROFILE

**Srividhya V** has completed MCA from Kongu Engineering College, Perundhurai, Erode, India. She is currently pursuing her Ph. D in Computer Applications from Kalasalingam Academy of Research & Education, Krishnankoil, Virudhunagar, India -626 126.

**S. Balamurali** is a Professor of Statistics and Director of Computer Applications at the Kalasalingam Academy of Research and Education. He received his undergraduate, postgraduate and doctoral degrees in Statistics from Bharathiar University, India. His research interests include applied statistics, data mining, network security and bioinformatics.