Performance degradation assessment and VM placement policy in cloud

Susmita J. A. Nair[1], T. R. Gopalakrishnan Nair[2]
[1]Bharathiar University, Coimbatore, India Networks and Security Research Group, R and D Centre, India
[1,2]Rajarajeswari Group of Institutions, India

Article Info

Article history:

Received Dec 29, 2018
Revised May 1, 2019
Accepted Jun 27, 2019

Keywords:

Downtime
Migration time
Performance degradation
Virtualization
VM migration

ABSTRACT

In virtualized servers, with live migration technique pages are copied from one physical machine to another while the virtual machine (VM) is running. The dynamic migration of virtual machines encumbers the data center which in turn reduces the performance of applications running on that particular physical machine. A considerable number of studies have been carried out in the area of performance evaluation during live VM migration. However, all the aspects related to the migration process have not been examined for the performance assessment. In this paper, we propose a novel approach to evaluate the performance during migration process in different types of coupled machine environment. It is presented here that the state of art VM migration technology requires further improvement in realizing effective migration by monitoring comprehensive performance value. We introduced the parameter, θ, to compare performance value which can be used for controlling and halting unsuccessful migration and save significant amount of time in migration operation. Our model is capable of analyzing real time scenario of cloud performance assessment targeting VM migration strategies. It also offers the possibility of further expanding to universal models for analyzing the performance variations that occurs as a result of VM migration.

Corresponding Author:

Susmita J. A. Nair,
Bharathiar University, Coimbatore,
Networks and Security Research Group, R and D Centre, RRGI,
Bengaluru, India.
Email: susmitajanair@gmail.com

1. INTRODUCTION

The distributed architecture of datacenters supports an enterprise with enhanced computational capacity, storage and advanced applications. Virtualization is the technology used for enhancement of capabilities of datacenter. Usually the data center is transformed into fully fledged cloud architecture mainly through the implementation of successful virtualization of machines in processing communication and storage domains [1]. The applications and services available in cloud computing are stored in data centers that are distributed across several geographic locations. The even distribution of system load to various data centers help to achieve better performance, reduce response time and to deal with the fault tolerance. In order to optimize the energy efficiency of their data centers, the tasks running on the under-utilized physical machines are mapped onto other Physical Machines (PMs) of the data center and then the under-utilized ones are shut down [2].

This can be accomplished by the migration of the virtual machines (VM) from the overloaded servers to other servers with reasonable migration cost. Placing the VM in a distant location away from the data centers affects the application performance. So, the network aspects between data centers and the VM have to be considered for placing the application in a VM [3]. VMs are chosen from the overloaded host that will be migrated to an underutilized physical machine (PM). While migrating the VMs, the cloud providers have to ensure the quality of service and they have to comply with the SLA (Service Level Agreement) also [4-6].

The traditional process migration was a complicated technique that involved relocating a process from one machine to another, whereas the Live VM Migration technique involves transferring of VM from source server to destination server with a minimal downtime [7-9]. The live migration process has six stages- (a) The pre-migration stage in which resources are allocated to a selected remote machine. (b) The reservation stage where setup is done for deploying the VM. (c) The iterative pre-copy stage where the pages are repeatedly copied until a minimum number of dirty pages is remaining. (d) The stop-and-copy stage where the VM is stopped at the source and the last dirty pages are transferred to the destination which results in downtime. (e) The commitment stage – the VM no longer exist at the source and the resources held by the VM are released. (f) The activation stage- the VM is resumed at the destination [10].

The two important factors involved in various live migration techniques are the total migration time and the downtime. Total migration time is the total time required to migrate the VM from source to destination machine. The downtime is the amount of time when the VM is not running [11]. The total migration time and the downtime are the key performance factors to be considered during VM migration. The purpose of live virtual machine migration is to ensure an uninterrupted service provisioning to the hosted applications during the migration process [12]. The two approaches in live VM migration are the push phase and the stop and copy phase. In the push phase, the memory pages are transferred from source server to the new location. During the transfer, as the VM is running, some of the memory pages may be modified which is called dirtying of pages. The dirtied pages have to be rectified and transferred again. The process is repeated iteratively, and it takes several rounds for the migration process to complete. At some point, the time taken by the rounds and the number of pages to be transferred will be very low. At this stage, the stop and copy phase takes place. The entire process of live VM migration affects the performance of the applications running in those VMs [7, 13, 14].

The whole work presented here results in the estimation of performance and cost based on the following four aspects: (1) the limit set for maximum iterations; (2) the volume of pages transferred with respect to the memory size; (3) the dirty rate comparison of successive iterations and the bandwidth usage; (4) the threshold set for dirty memory.

The primary contributions of this paper are:

a. An improved model for computing the number of iterations for VM migration by considering all the above-mentioned factors.
b. The model incorporates a very valuable parameter θ, a factor, which enable the datacenters to take a decision whether to continue with highly inefficient and under-performing migration operation which may be on continuation. Halting such faulty migrations bring rewards into the system through better resource utilization improving the efficiency of successful migration and achieving better performance for allowed operations. After the implementation of θ the performance value of migration jobs has improved up to 15%.
c. The simulations carried out according to the methodology discussed here estimated the performance and its variations successfully indicating the decline in the performance whenever occurred. This approach can enable the data centers to take necessary measures to reorganize the resources to contain the declined performance successfully.

The rest of the paper is organized as follows: Section 2 provides the related research method. Section 3 explains the performance evaluation and results. Section 4 concludes the paper.

2. RESEARCH METHOD

A guest VM is placed into the PM that has the least completion time. If such a PM is lacking the required resources, then either direct placement or migration-based placement technique can be adopted. In [15], the authors have examined that in on-line VM Placement, a VM whose migration overhead added with the completion time is minimum is chosen for migration. The number of VM request will not be known in advance. Hence, when the PMs are completely loaded, further VM requests should not be accepted. The migration cost from the overused PM to the active PM might be constrained by factors like bandwidth [16].

The VM to be migrated must communicate with the device drivers and network cards which are available in Domain 0. The device drivers are hosted in domain 0. It controls the physical network cards [17]. The performance of migration is depended on VM Memory size, memory dirtying rate, network transmission rate and the algorithm used for migration. The workload has to be distributed among all the host machines to enhance efficiency [18].

The migration process manages the state of resources, i.e., processor, memory, storage and I/O, during the transfer of VM. This includes memory allocated and used by VM, memory requested by application, virtual disk sizes and blocks used by VM [19]. In [13, 20] the number of iterations is calculated as the minimum value of threshold point and the maximum number of iterations set for migration. The threshold point is computed as a function of the ratio between threshold and memory allocated to the migrating VM. In [17], the authors have identified dirty rate and the frequency of its occurrence as the important factor affecting the iteration time and downtime.

2.1. The migration process

A server (host) with resource utilization of 80%, is considered to be as an over-utilized server. The workload of such an over-utilized host has to be distributed to the hosts which are under-utilized. A server with a utilization of 20% or below is assumed to be an underutilized server. The migration process evenly redistributes the workload among all the hosts with certain overhead. The process is pictorially depicted in Figure 1 and Figure 2. The migration cost is determined in terms of performance and energy.

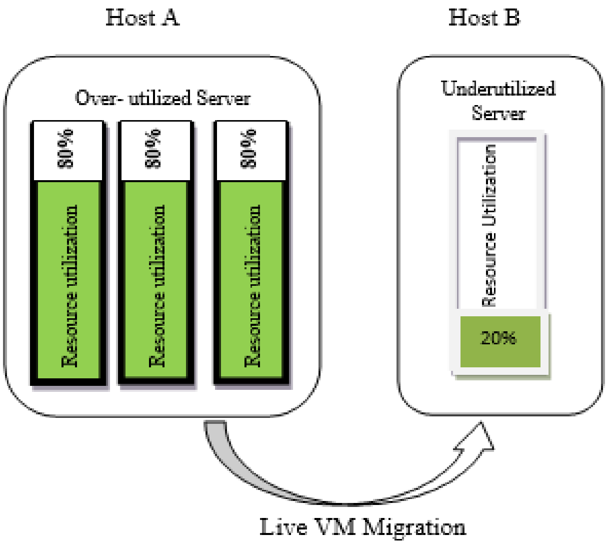

Figure 1. Before Migration Host A has a set of overutilized server with a utilization of up to 80% and the server at Host B is underutilized with 20% resource utilization

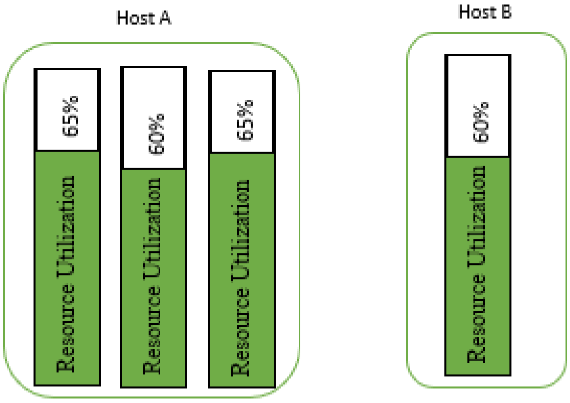

Figure 2. After Migration the servers at Host A and Host B have an even distribution of workload and resource utilization

2.2. Performance assessment and VM coupling

The performance of a system is defined as the amount of work done/unit time. The cost is the expense incurred due to the unutilized capability of the resources. The cost of migration is the amount of performance lost during migration. Performance has to be calculated (a) Before migration (b) During migration (c) After migration. Comparison between (a) and (b) gives the cost. Comparison between (a) and (c) gives the enhancement. The difference between (b) and (c) gives the cost of implementation of migration. In this paper, we have made the comparison between (a) and (b) to determine the cost before migration and the cost during migration. The virtual machines are migrated from one physical machine to another in the same domain or in a different domain. In any case, the processor will be associated with the memory, I/O and Storage.

Consider the VMs, say, VM0, VM1, …VMo, that are hosted at datacenter, DC1, with *m* resources. VMo represents the optimal number of VMs. When the number of VMs is greater than the optimal number (VMo), additional *r* resources are required to serve the extra VMs that exceeds the VMo. The processor coupling with resources can be achieved with migration and additional *r* resources can be acquired from other data centers to support all the VMs. Figure 3 pictorially represents the processor coupling with resources.

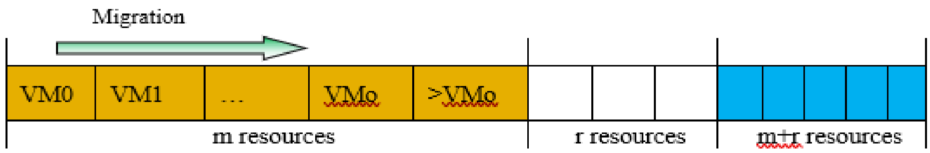

Figure 3. Processor coupling with resources will help to redistribute the VMs, that needs additional resources, to other servers

Table 1 shows the processing units for the memory, I/O and storage assuming that the number of processing units is same for all the three domains. All processing units are assumed to be homogeneous and homotopic with that of other processors. They can be mutually replaceable. Each VM has the structure of PM_{11} to PM_{1n}, PIO_{21} to PIO_{2n} and PS_{31} to PS_{3n} supporting equipotential capability in all parameters including capacity. The VM to be migrated can be coupled with any of these main processing domains, I/O processing domains and the Communication domains. Figure 4 shows the general image of such a coupling of processing units represented as PR.

Table 1. Processing units representation for different domains

Domain	Processing Unit			
Main Processing domain	PM_{11}	PM_{12}	…	PM_{1n}
I/O Processing domain	PIO_{21}	PIO_{22}	…	PIO_{2n}
Storage domain	PS_{31}	PS_{32}	…	PS_{3n}

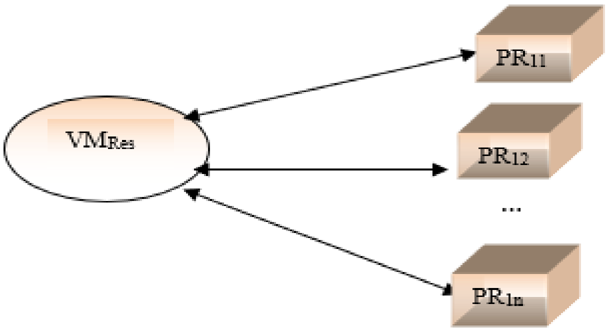

Figure 4. The general image of coupling of processing units to resources

2.3. The cost model

We assume that M number of VMs are to be migrated from PM1 (Physical Machine 1) to PM2 (Physical Machine 2). The pages get modified during migration and hence the recopying of modified pages and the migration process transpires in several iterations [17, 21].

Based on [13], in iteration 0, the whole memory of the VM gets copied to the destination machine. Let $Amt_{0,j}$ represents the amount of memory copied during 0th iteration of migration process of jth VM, VM_j, PM_{bwj} represents the physical memory bandwidth allotted for VM_j and $V_{memsize}$ shows the memory size of VM_j. Thus, if the 0th iteration which occurs between the time t_0 and time t_1, then $Amt_{0,j}$ can be represented as shown in (1).

$$Amt_{0,j} = \int_{t_0}^{t_1} \frac{V_{memsize}}{PM_{bwj}}(t)dt \qquad (1)$$

During the copy operation, some of the pages of the VM changes. This is called dirtying of pages. The dirtied pages have to be verified and copied in subsequent iterations. In (2), VM_{dirj} represents the portion of memory which is dirtied.

$$Amt_{i,j} = \int_{t_i}^{t_{i+1}} \frac{VM_{dirj}.Amt_{i-1,j}}{PM_{bwj}}(t)dt \qquad (2)$$

Once the copying process is over, the VM can be restarted at PM2. During the whole process the processor experiences a downtime, $T_{down,j,}$ which can be calculated as shown in (3). In this equation, P_{dir} is the page dirty rate, P_{size} is the size of the page size, $dur_{precopy}$ is the duration of the pre-copy and $VCPU_{context}$ is the time for context switch for the virtual CPU and $VM_{resumej}$ is the time taken to resume the VM_j at the new physical machine.

$$T_{down} = \frac{P_{dir}P_{size}dur_{precopy}}{PM_{bw}} + VCPU_{context} + VM_{resumej} \qquad (3)$$

The total down time for M can be calculated as

$$T_{down} = MT_{downj} \qquad (4)$$

A parameter,

$$x = \frac{P_{dir}P_{Size}}{PM_{bw}} \qquad (5)$$

is an important ratio, that decides the duration of the iteration. In the first iteration, the entire vm, $V_{memalloc,j}$ is copied. By substituting $dur_{i,j} = t_{i+1} - t_i$, for i=0,…,N, (1) and (2) can be represented as shown in (6) and (7) respectively.

$$Amt_{0,j} = V_{memalloc,j} = PM_{bw}.dur_{0,j} \qquad (6)$$

$$Amt_{i,j}=P_{size}.P_{dir}.Amt_{i-1,j}=PM_{bw}dur_{i,j}. \qquad (7)$$

The iteration may stop when it reaches the threshold, h, of dirty memory i.e.,

$$Amt_{i,j}=V_{memalloc,j}x^i \leq h. \qquad (8)$$

Another condition to stop the iterations is Max_{itr}, which is the maximum number of iterations set for pre-copy migration. The iteration stops when the volume of pages transferred is greater than the product of maximum multiple of memory size that migration should be terminated i.e., $Max_{memsize,}$ and the memory size of VM, $VM_{memsize}$. The iteration can stop in case the dirty rate of current iteration, *curdir* exceeds the previous iteration *predir* and the total bandwidth, Tot_{bw}, exceeds the maximum bandwidth, $Max_{bw.}$ This can be represented as $\theta = predir > curdir \ \& \ Tot_{bw} > Max_{bw}$.

These four conditions (C1, C2, C3, and C4), to stop the iterations can be represented as-

$$\text{C1: } ceil(\log_x \frac{h}{v_{memalloc,j}}) \tag{9}$$

$$\text{C2: } Max_{itr} \tag{10}$$

$$\text{C3: } Vol_{page} > Max_{memsize}.VM_{memsize} \tag{11}$$

$$\text{C4: } \theta = 1 \tag{12}$$

The total number of iterations, N, is calculated as shown in (13). The value of N is initialized to 0. The value of N in incremented iteratively until any one of the four conditions become true.

$$\text{N}=\begin{cases} N+1, if\ (C1 \vee C2 \vee C3 \vee C4 = False) \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{13}$$

The migration cost for the jth VM can be calculated as shown below.

$$Mig_{time} = \frac{V_{memalloc,j}}{PM_{bwj}} \frac{1-x^{N-1}}{1-x} \tag{14}$$

The total migration time, *TotalMig_{time,}* for M is estimated as given below.

$$TotalMig_{time} = M.Mig_{time} \tag{15}$$

The cost before migration can be represented as-

$$Cost_{jbefmig} = \int_{t_0}^{t_1} (\omega.VM_j)(t)dt + \int_{t_0}^{t_1} (\rho.VM_j)(t)dt + \int_{t_0}^{t_1} (\alpha.VM_j)(t)dt \tag{16}$$

In the above equation ω represents the CPU factor for memory; α represents the CPU factor for I/O and ρ represents the CPU factor for storage. The cost during migration can be represented as

$$Cost_{jdurmig} = \int_{t_1}^{Mig_{time}} (\omega.VM_j)(t)dt + \int_{t_1}^{Mig_{time}} (\rho.VM_j)(t)dt + \int_{t_1}^{Mig_{time}} (\alpha.VM_j)(t)dt \tag{17}$$

The performance before migration, represented as $\text{Perf}_{\text{jbefmig,}}$ and the performance during migration, represented as $\text{Perf}_{\text{jdurmig,}}$ can be calculated as follows.

$$\boldsymbol{Perf_{jbefmig}} = \frac{\mathbf{1}}{\boldsymbol{Cost_{jbefmig}}} \tag{18}$$

$$\boldsymbol{Perf_{jdurmig}} = \frac{\mathbf{1}}{\boldsymbol{Cost_{jdurmig}}} \tag{19}$$

2.4. Algorithm design

The procedure for performance evaluation is described in the following algorithm.

```
Input: hostList, vms
Output: Performance variation
For each host in hostList
   Step 1  : Verify host utilization
   Step 2  : If hostutil >= 80%, identify the vm, j, to migrate
                a. Search the vm with Minimum Migration Time
                b. Calculate Cost before migration as in (16)
             endif
             goto step 5
   Step 3  : if hostutil < =20%, Migrate all the VMs to another PM
   Step 4  : Search the PM with
                a. Average resource utilization, i.e between 20% & 80% and
```

b. New VM resource requirement +Exising resource utilization<80%

Step 6 : repeat the iterations for migrating VM, j to the identified PM until (13) is satisfied.

Step 7 : Calculate Cost during migration with (17)

Step 8 : Compute Performance before and after migration with (18) and (19) respectively.

Step 9 : Calculate performance variation as the difference between performance before migration & performance after migration.

3. PERFORMANCE EVALUATION

There are several cloud computing environments supporting live migration of VM. VMWare vSphere migrates the virtual machine's state while the network related details are retained. Automatic optimization of virtual machines, hardware service without any interruption of normal operations is possible with live vm migration [22]. In Xen hypervisor, the process of memory dirtying and updating of memory continues till the estimated time for transferring the remaining pages equals the time the guest is paused for migration [23].

For our experiment, the simulation set up has been done in Cloudsim 3.0.3 [24]. The features of the VM resemble the features of AmazonEC2 instance types, but with single core VM. The model uses the Minimum Migration Time (MMT) policy which selects the VM with minimum migration time requirement [24, 25].

The values entered for the page size, bandwidth and the threshold values decide the number of iterations, the migration time and the down time. The simulation is carried out by assigning certain values of the following parameters. The results are shown in the graph below.

a. The number of VMs are assumed to be: 5, 10, 15 and 20.
b. PM_{bw} – The bandwidth of the physical machine is 1 Gbps
c. The first iteration results in the dirtying of some pages, which has to be subsequently transferred in further iterations. The page dirty rate is proportional to the memory page transfer required which further adds up the downtime. On an average the page dirty rate Pdir=2500 pps.
d. The page size Psize is 4KB.
e. The threshold is the point beyond which the iterations does not occur and the VM migration fails. We are assuming the value of threshold 'h' as 100MB.

Figure 5 shows the migration time and downtime as a function of x varying the memory size ratio. The pre-copy algorithm will work effectively for x <1. There is a substantial decline in performance when the dirtying rate increases.

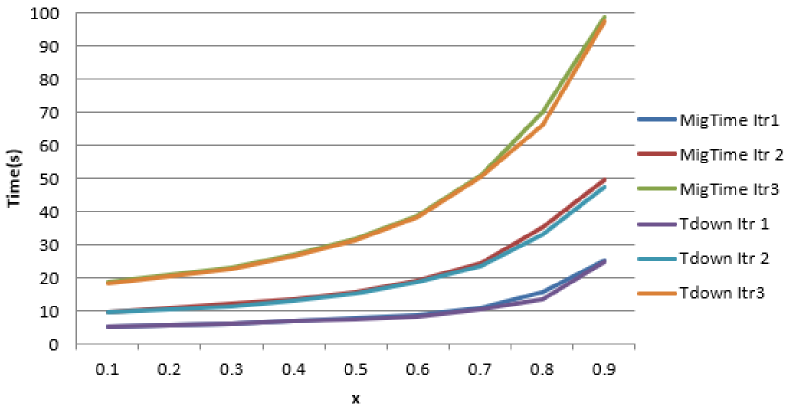

Figure 5. Migration and downtime with varying values of x

The CPU factor for memory was taken in the range of 100 to 1000 Mbits/s, I/O in the range 5 to 4200 Mbits/s and for storage 500 to 1000 Mbits/s. For a particular run of the algorithm, the cost and the performance obtained in terms of Mbits/sec is given in the Table 2.

The cost before migration using the different CPU factors were calculated using (12). The simulation was performed by varying the number of VMs. Figure 6 shows the percentage of performance degradation during VM Migration. The performance has degraded by 40% to 75% for different variation of ω, ρ and α. The improvement in performance by including the parameter θ, is shown in Figure 7.

Table 2. Cost and performance table

Iteration Number	ω	α	ρ	Number of VMs	Cost Before Migration	Performance Before Migration	Cost during Migration	Performance During Migration
1	533	5	1024	5	0.5469	1.8283	10.5560	0.9473
				10	0.5786	1.7283	11.6368	0.8593
				15	0.4924	2.0308	15.9552	0.6268
				20	0.4924	2.0308	15.9768	0.6259
2	200	41	600	5	0.2462	4.0623	5.8144	1.7199
				10	0.2424	4.1256	7.5992	1.3159
				15	0.2462	4.0623	9.5700	1.0449
				20	0.2424	4.1256	12.0588	0.8293
3	1024	512	512	5	0.4800	2.0835	11.3528	0.8808
				10	0.7093	1.4099	19.4112	0.5152
				15	0.9937	1.0064	20.8088	0.4806
				20	0.6606	1.5138	18.1492	0.5510

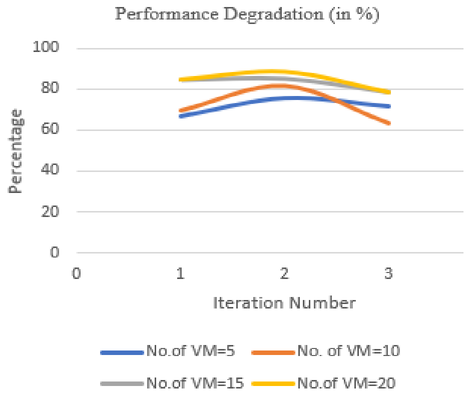

Figure 6. Performance Degradation in %. The performance varies with the number of VMs and the variation in CPU factors associated with the resources

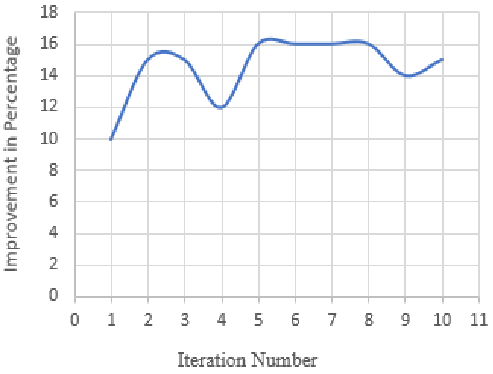

Figure 7. Improved Performance with θ. On an average the performance has improved upto 15%

4. CONCLUSION

In a cloud environment localized overloading of resources remains to be a challenge. Overloaded regions of resources like memory, storage and CPU facilities can be relieved from VM migration. Here we presented a detailed study of various aspects of VM migration through improved models and calculated the cost of migration which could be used for performance studies in cloud clusters. The analysis has shown the nature of degradation of performance which is a vital factor in improving the overall performance of current cloud architectures. The model can be improved to accommodate various other factors to detect further optimal values enabling cost reduction.

REFERENCES

[1] S. J. A. Nair and T. R. G. Nair, “Resource Extension Techniques in Enterprise Architectures through Cloud Connectivity,” *Int. Conf. Inv Comput Techn., Coimbatore, India*, 2016.

[2] V. De Maio, *et al.*, “An Improved Model for Live Migration in Data Centre Simulators,” *Proc. 16th IEEE/ACM Int. Symp. Cluster, Cloud, and Grid Comput*, pp. 527-530, 2016.

[3] J. Piao and J. Yan, “A network-aware virtual machine placement and migration approach in cloud computing,” *Proc. Int. Conf. on Grid and Cooper. Computing, Nanjing, China*, pp. 87-92, 2010.

[4] M. R. Chowdhury, *et al.*, “Implementation and performance analysis of various VM placement strategies in CloudSim,” *J. Cloud Comp*, 2015.

[5] M. Dabbagh, *et al.*, “Efficient Datacenter Resource Utilization Through Cloud Resource Over commitment,” *IEEE INFOCOM 2015 Workshop on Mobile Cloud and Virt.*, pp. 330-335, 2015.

[6] G. Sun, *et al.*, “A new technique for efficient live migration of multiple virtual machines,” *Future Generation Comp. Syst.*, pp. 74-86, 2016.

[7] H. Liu and B. He, “VMbuddies: Coordinating Live Migration of Multi-Tier Applications in Cloud Environments,” *IEEE Trans. Par. Distr. Systems*, vol. 26, pp. 1192-1205, 2015.

[8] W. Voorsluys, *et al.*, “Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation,” *Proc. 1st International Conference on Cloud Computing, Beijing*, pp. 254-265, 2009.

[9] S. Sotiriadis, *et al*., "Analysis of requirements for virtual machine migration in dynamic clouds," *12th Int. Symp. Par. and Distr. Computing*, pp. 116-123, 2013.

[10] T. Al-Said and O. Rana, "Implementing Migration-Aware Virtual Machines," *IEEE 2nd Int. Conf. on Cyber Sec and Cloud Computing*, 2015.

[11] S. Akoush, *et al*., "Predicting the Performance of Virtual Machine Migration," *Proc. 2010 IEEE Int Symp on Modeling, Analy and Sim. Of Comp and Tele Sys (MASCOTS'10), IEEE Computer Society, Washington DC, USA*, pp. 37-46, 2010.

[12] B. Hu, *et al*., "A Mobility-Oriented Scheme for Virtual Machine Migration in Cloud Data Center Network," *IEEE Access*, vol. 4, pp. 8327-8337, 2016.

[13] F. Callegati and W. Cerroni, "Live Migration of Virtualized Edge Networks: Analytical Modeling and Performance Evaluation," *IEEE SDN4FNS, Trento, Italy*, 2013.

[14] S. J. A. Nair and T. R. G. Nair, "Energy and Performance Evaluation in Cloud Live Vm Migration," *Int J. Res. Analytical Review*, vol. 6, pp. 1107-1112, 2019.

[15] K. Li, *et al*., "Virtual machine placement in cloud systems through migration process," *Int. J. Par., Emer Distr Systems*, pp. 393-410, 2014.

[16] X. Wang, *et al*., "Delay-cost tradeoff for virtual machine migration in cloud data centers," *J. Netw. Comput. Appl.*, vol. 78, pp. 62-72, 2017.

[17] W. Zhang, *et al*., "Performance degradation-aware virtual machine live migration in virtualized servers," *Proc. 13th IEEE Int. Conf. Parallel Distrib. Comput. Appl. Technol. (PDCAT)*, pp. 429-435, 2012.

[18] C. Yang, *et al*., "A Dynamic Resource Allocation Model for Virtual Machine Management on Cloud," *Commun. Comput. Inform. Science, Springer, Berlin, Heidelberg*, vol. 261, pp. 581-590, 2011.

[19] A. Choudhary, *et al*., "A critical survey of live virtual machine migration techniques," *J. Cloud Comput: Adv., Syst. Appl*, vol. 6, pp. 1-41, 2017.

[20] W. Cerroni and F. Esposito, "Optimizing Live Migration of Multiple Virtual Machines," *IEEE Trans. On Cloud Comput.*, vol. 6, pp. 1096-1109, 2018.

[21] X. Fu and C. Zhou, "Virtual machine selection and placement for dynamic consolidation in Cloud computing environment," *Frontiers Comput. Sci*., vol. 9, pp. 322-330, 2015.

[22] vSphere vMotion- VMware. Available: https://www.vmware.com/in/products/vsphere/vmotion.html.

[23] Red Hat, Inc, "The Red Hat Enterprise Linux Virtualization Guide," Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/virtualization/chap-virtualization-xen_live_migration

[24] A. 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, pp. 1397-1420, 2012.

[25] A. R. Mohazabiyeh and K. H. Amirizadeh, "Energy-Aware Adaptive Four Thresholds Technique for Optimal Virtual Machine Placement," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, pp. 3890-3901, 2018.

BIOGRAPHIES OF AUTHORS

Susmita J. A. Nair holds a graduate degree in computer science and a post graduate degree in Computer Applications. She is currently pursuing her Ph.D. in Computer Science from Bharathiar University, Coimbatore, India. Her area of interest is Cloud Computing.

Dr. T.R. Gopalakrishnan Nair, a fellow of Institution of Engineers, has 34 years of experience in professional field spread over Research, Industry and Education. Currently, he is the Rector for Rajarajeswari Group of Institutions in India. He was the ARAMCO Endowed Chair in Technology in PM University, KSA. He holds degrees M.Tech. (I.I.Sc., India) and a Ph.D. in Computer Science. His areas of interest include Advanced networks, Cognitive Systems and Multidisciplinary studies including Brain and physical systems. He is a senior member of IEEE, ACM and few other professional bodies.