Grenze International Journal of Engineering and Technology, Jul 2019

Virtual Machine Migration Techniques in a Cloud Computing Environment: A Survey

Jyoti M. Hurakadli [1] and B. Sathish Babu [2]
[1] Department of Computer Science & Engineering, Basaveshwar Engineering College, Bagalkot, India
Email: jyotimh2010@gmail.com, jyotimh2002@yahoo.com
[2] Department of Computer Science & Engineering, R. V. College of Engineering, Bangalore, India
Email: bsbabu@rvce.edu.in

***Abstract*—The study of different kinds of computing has been fascinating with the addition of another computing model called cloud computing. The unique features of cloud computing makes it very different from the other computing models. The datacenters in cloud use virtualization for efficient resource utilization and management to reduce infrastructural and operational costs. Virtualization empowered by virtual machine migration meets the increasing demands of dynamic workload by relocating virtual machines on different physical machines. The paper presents different virtual machine migration techniques in cloud datacenter along with some observations and challenges.**

***Index Terms*— Cloud, virtualization, virtual machine, migration, load balancing, consolidation.**

I. Introduction

Cloud utilizes virtualization to provide a uniform interface to dynamically scalable underlying resources, with the intention that the virtualization layer conceals physical heterogeneity, geographical distribution and faults [1]. The purpose of virtualization is to perform resource sharing, utilization and distribution to their maximum potential, reduce infrastructure costs and operational costs. A typical virtual machine (VM) comprises of guest OS, guest application(s), with shared resources - like CPU cores, memory, network interface card and disk drives - over a physical machines (PM) comprising its own layers of hardware and software. VMs reside independently, in isolation on PM and executes over Virtual Machine Monitor or Hypervisor.

Some of the issues related to VMs are VM scheduling, VM management and VM migration. VM scheduling addresses the placement of VMs within the cloud infrastructure while minimizing the operating costs of the cloud itself. Different system resources are taken into consideration for VM scheduling, like CPU, memory, network bandwidth, etc. The works in [2, 3, 4, 5] discuss some of the VM placement techniques. Once the VMs are allocated and scheduled, frequent management of these VMs are required to control and manipulate the size and placement of VM images in various ways to conserve resources and remove unnecessary bloats, thus the VMs are managed through *live migrations.*

A. VM Migration and the Related Issues

VM migration takes place often in context of load balancing, server consolidation, server management, disaster recoveries, etc. and can be initiated either reactively or proactively [6]. More specifically, the reasons that justify VM migration in a production system include: i) the need to balance system load, which can be accomplished by

Grenze ID: 01.GIJET.5.2.3

migrating VMs out of overloaded servers; and ii) the need to selectively bring servers down for maintenance after migrating their workload to other servers.

Popular hypervisors, such as Xen and VMWare, allow live/hot migration, as opposed to pure stop-and-copy/cold migration. Live migration of VMs allows workload movement with minimal application downtime which is important for serving live services. Live migrations are implemented using either *precopy / postcopy* [7] / *hybrid* [8] approach. In [9], William Voorsluys et al. evaluated the effects of live migration of VMs on the performance of applications running inside Xen VMs and showed that, in most cases, migration overhead is acceptable but cannot be disregarded, especially in systems where availability and responsiveness are governed by strict service level agreements (SLA). Also, the works in [10, 11, 12] present the different ways to enhance the performance of a single VM migration.

Live migrations of multiple VMs at once in the datacenter may increase the number of VMs in transit at any given time, thereby increasing the burden on the system infrastructure. Also, the gains from agile resource management may diminish as the resource management overheads begin to dominate. Thus it is beneficial to improve the VM migration efficiency either by improving the mechanisms of VM migration or by improving the physical server or network configurations.

B. Challenges in Designing an Efficient VM Migration Mechanism in the Cloud

Currently, the cloud service providers commit only to availability of their services, through SLAs with their customers. However, QoS guarantees are not provided, mainly due to various factors that introduce uncertainty regarding the actual performance of VMs. Efficient VM migration mechanisms can improve utilization of the available cloud resources, reduce execution time and mitigate the effects of uncertainty but efficient utilization of the cloud computing infrastructure faces various fundamental challenges like workload uncertainty, unpredictability of multi-tenancy effects, unknown evolution of accompanying data volume, time-varying network link/server processing capacity, and partial availability of cloud-related information. [13]

Overall, the challenge lies in that a migration has contradicting consequences due to the reasons as stated: i) by reducing the number of active servers, the load is concentrated on fewer servers which are thus loaded more; ii) by performing load balancing, the heavy load of some servers is alleviated, but at the same time it is likely that more servers would need to be activated to accommodate it. Thus, the efficient VM migration mechanism depends on the relative number of VM migrations to be performed when servers are idle and loaded, as well as on the VM migration time and VM downtime. *An associated challenge would therefore be to derive analytical model for efficient VM migration scheme taking into account the possible uncertainties of the cloud.*

The rest of the paper is organized as follows. Section II includes the generic model of datacenter management using VM migrations and the related issues. Some of the existing approaches are reviewed and compared in section III. Section IV concludes the paper.

II. LIVE VM MIGRATIONS IN A CLOUD DATACENTER

A. A Generic Model for a Datacenter Management Using VM Migrations

Typically, a cloud environment comprises of a cloud datacenter and the cloud clients. Cloud datacenter in turn consists of many interconnected servers/PMs in which the VMs are deployed. Every PM will then set two predefined thresholds - upper and lower - based on resource utilization [18, 21, 22, 25] and the datacenter is usually managed as given in steps 1 and 2.

1. Place the incoming VM in an appropriate PM that will optimize the overall performance of the datacenter.
2. As the resource requirements of the VMs change dynamically over time, the performance degrades. To overcome this, the details of resource utilization / load on every PM is collected and conveyed to centralized *Decision Maker* to decide on the following.
   a. If load on a specific PM exceeds its upper threshold, reduce its load by migrating an appropriately selected VM to another PM, thus performing *load balancing*. [14]
   b. On the other hand, if load on a specific PM reduces below its lower threshold, migrate all the VMs to the available PMs and switch-off that PM to reduce the power consumption, thus performing *dynamic server consolidation*. [14]

B. Issues Involved in VM Migrations

To achieve each of the three goals - load balancing, server consolidation, and hotspot mitigation, Mayank Mishra et al. in [14] suggest that VM migration-based heuristics need to address three important questions: when to migrate, which VMs to migrate, where to migrate among the set of destination host machines. Migration heuristics address each of these questions based on several constraints: overhead of the migration process, impact on applications during migration, degree of improvement in intended goals of performance of resource utilization, and soon. The various issues / policies involved in VM migrations are listed below.

1. Monitoring the resources / Information policy. In a cloud environment, monitoring is performed for two main reasons: i) to ensure that VMs get the capacity stipulated in the SLAs; ii) to collect data for system-wide accounting of the resources required for billing. There are two complimentary types of monitoring that must be carried out by the system: i) infrastructure measurements, including low-level measurements of the resources used by the VM; ii) Key Performance Indicators specific to the application. [6]
2. When to migrate? / Transfer policy. Generally, the VM migrations are triggered either reactively using a predefined threshold or proactively after analyzing workload behavior for a specified time period. The paper [15] explicitly specifies the migration triggering point attribute that labels the condition to migrate a VM and these include heuristic-based, historical data-based, intelligent-learned QoS, or scheduled adoption. Deciding on migration triggering based on heuristics lead to unnecessary migrations if not managed properly. To avoid such multiple migrations and frequent overheads triggered by small transient spikes, migrations are triggered only if thresholds exceed a sustained time, termed as *conservative* approach. Else, the approach is said to be *aggressive* [20].
3. Which VM to migrate? / Selection policy. The VMs to migrate may include VMs with resource constraints / un-steady workload behavior/ minimum migration time/bandwidth consumption etc. [15]
4. Where to migrate? / Target policy. The selected VMs are migrated to the target PM based on the co-location criteria attribute [15] that defines the criterion opted to co-host multiple VMs within a PM. VM co-location criteria can be defined in terms of affinity towards shared memory / communication bandwidth between VMs, power efficiency, sufficient resource availability, etc.

C. Classification of VM Migration Techniques

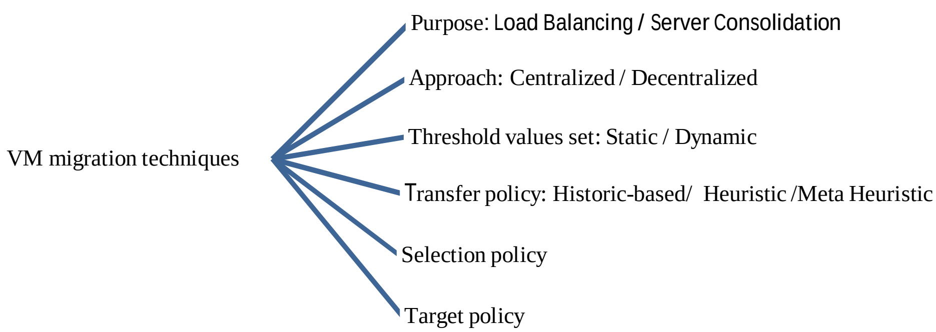

Figure 1. Classification of VM migration techniques

Broadly speaking, the datacenters in the cloud are managed in a *centralized /decentralized* manner, either to *balance / consolidate* the load or both. The thresholds can be set either *statically or dynamically* based on *single or multiple* VM resource types). The selected VMs are migrated *reactively or proactively* based on *heuristics, historical data, intelligent-learned QoS,* and are migrated to the target hosts so as to optimize the objective functions. The classification of the VM migration techniques is shown in Fig. 1.

III. Literature Survey

A. Different Live VM Migration Techniques in Cloud Datacenter

Zhiyong Yang et al. in [16] proposed a load balancing algorithm based on live migration of VM. This approach periodically collects the CPU load of all the PMs and stores it as a historical data. It then predicts the future CPU load based on the historical data and triggers migration only when predicted CPU load exceeds upper threshold for specified times. Fractal method is used to predict the CPU load as it has self-similarity. This approach limits its scope to migration triggering rule and do not consider which VM to migrate and where to migrate.

Xin Lu and Zhuanzhuan Zhang in [17] presented Modified Best Fit Decreasing (MBFD) algorithm for VM dynamic migration. To mitigate the hot-spot hosts, this selects the VM with maximum load from the hot-spot host to migrate to the target host selected using MBFD. MBFD selects a non-hot-spot host whose difference between the current load of that non-hot-spot host on VM migration and the maximum threshold is minimum.

Rajyashree et al. in [18] proposed a double threshold based load balancing approach, where the thresholds are set dynamically based on the load in the last 20 sec. For load balancing, it selects the VM - whose utilization is greater than or equal to the difference between the upper threshold and the host current utilization - to migrate to a target host – where i) current utilization lies between its upper and lower threshold and ii) the utilization do not exceed its upper threshold on migration. If no such target host is found, then VM is placed on new host. It also performs server consolidation.

Khushba Maurya et al. in [19] present an energy conscious, power aware load balancing strategy based on adaptive migration of VMs. This approach sets the two thresholds dynamically as in [18] but differs in VM selection and target selection. It selects multiple VMs - with minimum migration time - iteratively from overloaded host to migrate until its utilization reduces below its upper threshold. Also, the target host is selected using Power Aware Best Fit Decreasing (PABFD) algorithm which selects the host that has a minimum increase in power utilization on VM migration.

Wei Zhang et al. in [20] proposed a load balancing technique called Live VM Migration with less Cost and application Interference (LVMCI) by migrating the VMs conservatively. It estimates the migration cost of every VM of the overloaded PM in terms of its CPU utilization during the memory iteration time and downtime and selects the VM with minimum migration cost to migrate. Later, it estimates the performance interference of a migrating VM with the VMs of other hosts and selects the host with minimum performance interference as target host.

In [21], the authors proposed novel adaptive heuristics for dynamic consolidation and load balancing of VMs based on statistical analysis of historical data collected from the VMs resource usage. The upper threshold is set dynamically using Median Absolute Deviation (MAD)/Inter Quartile Range (IQR)/Local Regression (LR)/Robust Local Regression (LRR) and overloaded host is detected. The VM from overloaded host is selected based on three policies, Minimum Migration Time (MMT), Random Choice (RC), Maximum Correlation (MC) and the target host is selected using PABFD algorithm. Of all the combinations, LR+MMT algorithm is found to reduce energy consumption and ensure a high level of adherence to prescribed SLAs.

Xiong FU et al. in [22] proposed VM selection and VM placement algorithms for dynamic consolidation of VMs. This approach set the thresholds dynamically as in [35]. To balance the load, the VM selection algorithm *Meets Performance* (*MP*) selects the VM with minimum size whose migration from that host reduces its load lower than upper threshold. The VM placement policy *Minimum Correlation Coefficient (MCC)* selects the target host that has a minimum correlation coefficient between the migrating VM and the VMs of other hosts.

The authors in [23] address the load balancing problem by defining two parameters, *Integrated Overload Index* (*IOI*) for each PM, using the measured CPU load at a given time and *Overload Index (OI)* using some prescribed maximum capacity. If host's *IOI* exceeds *OI*, then the host is overloaded. To balance the load, VMs are selected using four different policies, random policy, volume-aware, volume-aware with setpoint normalization and migration likelihood with setpoint normalization.

VMs with strong utilization correlation should be scattered on different PMs to reduce the risk of future overload. Xiangliang Zhang et al. in [24] proposed a load balancing algorithm *Scattered* and compared with *Largest VM First* (LVF) algorithm. The *Scattered* algorithm selects the VM - with strong correlation with other VM on that host and whose utilization has large variance during certain time - and migrates to a host with fewer numbers of correlated VMs, hosting VMs and large free capacity. On the other hand, LVF algorithm selects VM with largest CPU requirement to migrate to a host with largest free capacity among non-overloaded hosts.

Xiaoying Wang et al. in [25] proposed a decentralized VM migration approach to balance the load, save energy consumption and maintain QoS. Here, every PM periodically evaluates a parameter *load index* based on its CPU utilization and shares it with other PMs. Each PM will then add the received *load index* values into its load vector. On overload detection, a VM is migrated to the target host using *BestFit* (most light-loaded) strategy.

Ren Gao et al. in [26] present a novel approach to balance the load dynamically using ant colony optimization (ACO). Two strategies, forward-backward ant mechanism and max-min rules, are used to quickly find out the candidate nodes to balance the load. ACO is suitable for load balancing because it is a self-organizing algorithm based on the local information to make judgments and actions, that is, the system does not require a global control center. Also, the larger sized VMs are migrated from overloaded host to under-loaded hosts.

Mahdi Mollamotalebi et al. in [27] aims to effectively place the VMs, relocate/migrate the VMs to balance the load and also periodically consolidate the VMs for energy efficiency. It achieves these by considering two single-goal approaches, Energy-Reduction to reduce the energy consumption and SLAV-Reduction to reduce the SLA violation rates. It then presents a double-goal approach - Energy-reduction and SLAV-reduction dynamic run-time replacement (ESDR), achieving the above two goals simultaneously. With regard to changes of datacenter's state, ESDR switches between different policies at run-time if it is necessary. It checks datacenter metrics monitored during different executions in order to determine if current in-use approach needs changes. It uses SLA violation and power efficiency ratio metrics to evaluate whether the active approach should be switched.

Salam Ismaeel in [28] proposed the proactive dynamic provisioning framework in a data centre with a focus on energy conservation. The framework consists of an important subsystem namely, Workload Prediction Subsystem, that predicts the load based on prediction techniques. It also explains different techniques to detect host overload / underload, VM selection, VM placement and migration.

To reduce the power consumption, the VM placement algorithm of OpenStack Neat - Modified Best-Fit Decreasing (MBFD) - minimizes the number of servers but increases SLA violations and the VM migrations. To overcome the drawbacks of MBFD, the authors of [29] proposed VM placement algorithms based on both bin-packing heuristics and servers' power efficiency, namely Power Efficient FFD (PEFFD), Power Efficient BFD (PEBFD) and Medium Fit Power Efficient Decreasing (MFPED). MFPED improves SLA violation and number of VM migrations against MBFD by up-to 78% and 46%, respectively, depending on the cloud scenario.

B. Review Summary of Different Live VM Migration Techniques in a Datacenter

The review summary of different live VM migration approaches are tabulated in Tables 1 and 2. First, the works discussed in [16-24] balance the load in a centralized manner but suffer from scalability and fault-tolerant issues. To overcome these drawbacks, the decentralized load balancing approaches are listed in [25, 26]. Also, the works in [18, 21, 22, 25, 27] perform server consolidation along with load balancing.

Second, to detect the hosts as overloaded/underloaded, the static threshold values are set in [16, 17, 20, 25]. It is observed that the greater upper threshold in load balancing increases the resource utilization and reduces the number of migrations whereas the greater lower threshold in server consolidation reduces the resource utilization and increases the number of migrations. Also, the static thresholds are not suitable for dynamic, adaptable and unpredictable workloads, if not set properly. To overcome these limitations, thresholds are set dynamically in [18, 19, 21, 22, 23, 24, 26, 27] to adapt to the changing workload patterns exhibited by the applications.

Third, the works in [16, 21, 22, 23, 24, 25, 27] considered only CPU to measure the VM load, whereas [17, 18, 19, 20, 26] considered combinations of CPU, memory, and bandwidth. Fourth, VM migrations are triggered by considering different scenarios. Analyzing the historic-based data and deciding on migrations is in [16, 21, 22], collecting the load related information and applying heuristics online is done in [17, 18, 19, 20, 23, 24, 25, 27] whereas meta-heuristic based migration decision is in [26]. Also, the aggressive migrations will increase the number of migrations, rather conservative migrations as in [16, 20] are preferred.

Finally, VMs are selected for migration based on their size, migration time, migration cost, correlation etc and destination hosts are selected based on either First Fit, Best Fit, First Fit Decreasing, Best Fit Decreasing, Modified BFD, PABFD, or Minimum correlation etc.

TABLE I. METHODOLOGY USED

P No.	Methodology used	Centralized / Decentralized, Static / Dynamic	Monitored Parameters	Optimization Objective	Simulator
16	Load balancing	Centralized, Static	CPU	Max. resource utilization in general	Testbed
17	VM Dynamic Migration Scheduling Model Based on MBFD	Centralized, Static	CPU, memory, bandwidth	Min. variance of CPU utilization	Cloudsim 3.0.3
18	Double Threshold Based Load Balancing Approach	Centralized, Dynamic	CPU, memory, bandwidth	Exact solution	Testbed
19	Energy Conscious Dynamic Provisioning	Centralized, Dynamic	CPU, memory, bandwidth	Min. variance of CPU utilization	Cloudsim
20	Live VM Migration with less Costs and application Interference	Centralized, Static	CPU and disk	Min. migration cost and performance interference.	Testbed based on Xen
21	Adaptive heuristics for dynamic consolidation of VM	Centralized, Dynamic	CPU	Max. Resource utilization, Min. energy consumption	Cloudsim 3.0

22	VM selection and VM placement for dynamic consolidation	Centralized, Dynamic	CPU	Max. Resource utilization, Min. energy consumption	Cloudsim 3.0
23	Virtual machine migration in cloud infrastructures	Centralized, Dynamic	CPU	Max. resource utilization	Testbed
24	Scattered algorithm	Centralized, Dynamic	CPU	Min. bandwidth / aggregate traffic rates	Testbed
25	A Decentralized VM Migration Approach for datacenter	Decentralised, Static	CPU	Min. energy consumption, SLAs, Max. QoS,	Testbed
26	Dynamic Load Balancing Strategy using Ant Colony Optimization	Decentralised, Dynamic	CPU, memory, bandwidth	Max. resource utilization	Cloudsim 3.0
27	Energy-reduction and SLAV-reduction dynamic run-time replacement	Centralised, Dynamic	CPU	Max. Energy-reduction and SLAV-reduction	DCSim Simulation
29	Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework	Centralised, Dynamic	CPU	Min. the power consumption	Cloudsim

TABLE II. DETAILS OF VM MIGRATIONS AND RESULTS

P No.	Purpose	When to migrate? / Transfer Policy	Which VM to migrate? / Selection Policy	Where to migrate?/ Target Policy	Observations from results
16	Load balancing	Historic based data	--	--	Reduced frequent migration and overheads
17	Hot-spot mitigation	Heuristic	Maximum load VM	MBFD	Reduced energy consumption, number of migrations, SLA violations
18	Load balancing and consolidation	Heuristic	VM with utilization greater >= (upper threshold-host utilization)	PM, which will not be overloaded on VM migration	Reduced energy consumption, efficient resource utilization and minimum number of migration
19	Load balancing	Heuristic	Multiple VMs with minimum migration time	PABFD	Reduced energy consumption
20	Load balancing	Heuristic	VM with minimum migration cost	PM with minimum performance interference	Downtime is high for the workloads with high frequent dirty rates
21	Load balancing and consolidation	Historic based data	VM with MMT, RC, MC	PABFD	Reduced energy consumption, number of VM migrations, SLA violations
22	Load balancing and consolidation	Historic based data	MP	MCC	Reduced energy consumption, SLA violations
23	Load balancing	Heuristic	Random, VA, $VA_{x,}$ ML_x	PM with minimum load	Reduced number of migrations per VM, inter-arrival migration time
24	Load balancing	Heuristic	VM with strong correlation & large variance	PM with fewer number of correlated VMs, the hosting VMs and large free capacity	Reduced overload risk, number of migrations, minimal impact on communication cost
25	Load balancing and Consolidation	Heuristic	VMs whose migration reduces the load.	Most light-loaded host	Reduced SLA violations, number of VM migrations, energy consumption
26	Load balancing	Meta Heuristic	Larger sized VM	Underloaded host	high network performance under medium and heavily load
27	Load balancing and Consolidation	Heuristic	VM with max CPU utilization	PM with sufficient CPU resources (FFD)	Increased host utilization, Reduced power consumption, SLA violations, number of migrations
29	Load Balancing and Consolidation	Heuristic	VMs of lightly loaded PMs	PEFFD, PEBFD, MFPBD	Reduces the power consumption, SLA violation and number of migrations

IV. Conclusion And Future Challenges

The unpredictable workload nature, varying bandwidth capacity, inability to accurately predict application demands, and aggressive migration decisions call for dynamic, adaptive, lightweight, and optimal VM migration scheme so as to improve applications performance, resource utilization, energy consumption and fault tolerance. The lightweight VM migration design tends to reduce the overall development effort, adds to application performance, and speed-up the processing in the datacenter. Yet many challenges still exist in the datacenter concerning the accuracy in estimating the resource demand of cloud-hosted applications, the optimal distribution of VMs over hosts due to the dynamic resource demand of hosted applications, and the security while co-hosting VMs within servers.

References

[1] D. Garg, K. Kant and A. Bansal, "Review of Virtual Machine Migration in Datacenters", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, issue 6, pp. 204 – 208, June 2013

[2] U. Bellur, C. S. Rao and S. D. Madhu Kumar, "Optimal Placement Algorithms for Virtual Machines in Distributed, Parallel and Cluster Computing", ACM classes, 2010

[3] W. Li,, "Virtual Machine Placement in Cloud Environments", *www8.cs.umu.se/research/uminf/reports/ 2012/013/part1.pdf*

[4] C. Tang, M. Steinder, M. Spreitzer and G. Pacifici, *"Scalable Application Placement Controller for Enterprise Data Centers"*, International World Wide Web Conference Committee (IW3C2), pp. 331-340, 2007

[5] N. Quang-Hung, N. Thoai and N. T. Son, "EPOBF: Energy Efficient Allocation of Virtual Machines in High Performance Computing Cloud", J. of Science and Technology, vol. 51, issue 4B, 173 – 182, 2013

[6] E. Elmorth and L. Larson, "Interfaces for Placement, Migration, and Monitoring of Virtual Machine in Federated Clouds", 8th International Conference on Grid and Cooperative Computing, pp. 253-260, 2009

[7] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt and A. Warfield, "Live migration of virtual machines", NSDI'05: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation, Berkeley, CA, USA, USENIX Association pp. 273–286, 2005

[8] J. Song, W. Liu, F. Yin and C. Gao, "TSMC: A Novel Approach for Live Virtual Machine Migration", J. of Applied Mathematics, Article ID 297127, Hindawi Publishing Corporation, 2014

[9] W. Voorsluys, J. Broberg, S. Venugopal and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation", CloudCom '09 Proceedings of the 1st International Conference on Cloud Computing, pp. 254-265, Springer-Verlag Berlin, Heidelberg, 2009

[10] S. Sahni and V. Varma, "A Hybrid Approach to Live Migration of Virtual Machines", IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2012

[11] R. K. Raj and G. J. Leelipushpam, "Improved live migration using compressed log files", International Journal of Engineering Trends and Technology, vol. 4, issue 3, 373-377, 2013

[12] M. Gahlawat and P. Sharma, "Reducing the cost of virtual machine migration in federated cloud environment using component based VM", J. of Information Systems and Communication, vol. 3, issue 1, 288-290, 2012

[13] L. Gkatzikis and I. Koutsopoulos, "Migrate or Not? Exploiting Dynamic Task Migration in Mobile Cloud Computing Systems", Wireless communications, vol. 20, issue 3, 24-32, 2013

[14] M. Mishra, A. Das, P. Kulkarni and A. Sahoo, "Dynamic Resource Management Using Virtual Machine Migrations", IEEE Communications Magazine, pp. 34-40, September 2012

[15] R. Ahmad, A. Gani, S. H. A. Hamid, M. Shiraz, A. Yousafzai and F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers", J. of Network and Computer Applications, vol. 52, pp. 11-25, 2015

[16] Z. Yang, C. Li, A. Yun and C. Liu, "A new trigger strategy based on live migration of the virtual machine", International Conference on Control Engineering and Communication Technology, pp. 677-680, 2012

[17] X. Lu and Z. Zhang, "A Virtual Machine Dynamic Migration Scheduling Model Based on MBFD Algorithm" International Journal of Computer Theory and Engineering, vol. 7, issue 4, 278-282, 2015

[18] Rajyashree and V. Richhariya, "Double Threshold Based Load Balancing Approach by Using VM Migration for the Cloud Computing Environment", International Journal of Computer Science and Engineering, vol. 04, issue 1, pp. 971-9976, 2015

[19] K. Maurya and R. Sinha, "Energy Conscious Dynamic Provisioning of Virtual Machines using Adaptive Migration Thresholds in Cloud Data Center", IJCSMC, vol. 2, issue 3, pp. 74 – 82, 2013

[20] W. Zhang, M. Zhu, Y. Mei, L. Xiao, L. Ruan, Y. Gao and Y. Sun, "LVMCI: Efficient and Effective VM Live Migration Selection Scheme in Virtualized Data Centers", IEEE 18th International Conference on Parallel and Distributed Systems, pp. 368-375, 2012

[21] 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, pp. 1397-1420, 2012

[22] X. FU and C. ZHOU, "Virtual machine selection and placement for dynamic consolidation in Cloud computing environment", J. of Frontiers of Computer Science, vol. 9, issue 2, pp.322-330, 2015
[23] A. V. PaPadopoulos and M. Maggio, "Virtual machine migration in cloud infrastructures: Problem Formulation and Policies Proposal" CDC, pp. 6698-6705, 2015
[24] X. Zhang, Z. Shae, S. Zheng and H. Jamjoom, "Virtual machine migration in an over-committed cloud", IEEE Network Operations and Management Symposium, 2012
[25] X. Wang, X. Liu, L. Fan and X. Jia, "A Decentralized Virtual Machine Migration Approach of Data Centers for Cloud Computing", Article ID 878542, Hindawi Publishing Corporation Mathematical Problems in Engineering, 2013
[26] R. Gao and J. Wu, "Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization", Future Internet, vol. 7, pp. 465-483, 2015
[27] M. Mollamotalebi and S. Hajireza, "Multi-objective dynamic management of virtual machines in cloud environments", Journal of Cloud Computing: Advances, Systems and Applications, vol. 6, issue 16, 2017
[28] S. Ismaeel, R. Karim and A. Miri, "Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres", Journal of Cloud Computing: Advances, Systems and Applications, vol. 7, issue 10, 2018
[29] F. F. Moges and S. L. Abebe, "Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework", Journal of Cloud Computing: Advances, Systems and Applications, vol. 8, issue 2, 2019.