Computers, Materials & Continua
DOI: 10.32604/cmc.2023.035161
Article

Performance Framework for Virtual Machine Migration in Cloud Computing

Tahir Alyas[1], Taher M. Ghazal[2,3], Badria Sulaiman Alfurhood[4], Munir Ahmad[5], Ossma Ali Thawabeh[6], Khalid Alissa[7] and Qaiser Abbas[8,*]

[1]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
[2]School of Information Technology, Skyline University College, University City Sharjah, Sharjah, 1797, UAE
[3]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebansaan Malaysia (UKM), Bangi, 43600, Selangor, Malaysia
[4]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Saudi Arabia
[5]School of Computer Science, National College of Business Administration & Economics, Lahore, 54000, Pakistan
[6]Dean of Student Affairs, Skyline University College, University City Sharjah, Sharjah, 1797, UAE
[7]Networks and Communications Department, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, 31441, Saudi Arabia
[8]Faculty of Computer and Information Systems Islamic University Madinah, Madinah, 42351, Saudi Arabia
*Corresponding Author: Qaiser Abbas. Email: qabbas@iu.edu.sa
Received: 09 August 2022; Accepted: 20 October 2022

Abstract: In the cloud environment, the transfer of data from one cloud server to another cloud server is called migration. Data can be delivered in various ways, from one data centre to another. This research aims to increase the migration performance of the virtual machine (VM) in the cloud environment. VMs allow cloud customers to store essential data and resources. However, server usage has grown dramatically due to the virtualization of computer systems, resulting in higher data centre power consumption, storage needs, and operating expenses. Multiple VMs on one data centre manage share resources like central processing unit (CPU) cache, network bandwidth, memory, and application bandwidth. In multi-cloud, VM migration addresses the performance degradation due to cloud server configuration, unbalanced traffic load, resource load management, and fault situations during data transfer. VM migration speed is influenced by the size of the VM, the dirty rate of the running application, and the latency of migration iterations. As a result, evaluating VM migration performance while considering all of these factors becomes a difficult task. The main effort of this research is to assess migration problems on performance. The simulation results in Matlab show that if the VM size grows, the migration time of VMs and the downtime can be impacted by three orders of magnitude. The dirty page rate decreases, the migration time and the downtime grow, and the latency time decreases as network bandwidth increases during the migration time and post-migration overhead calculation when the VM transfer is completed. All the simulated cases of VMs migration were performed in a fuzzy inference system with performance graphs.

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Keywords: Latency; cloud computing; dirty page ratio; storage; migration performance

1 Introduction

VM allows a virtualized service to seamlessly relocates between servers, allowing it to respond to changes in the environment quickly. However, there is a shortage of knowledge about its performance despite extensive investigation. Live migration is a solution that tries to move an application between physical hosts with nearly little downtime, making it ideal for huge, heterogeneous, and flexible environments since it allows the application to run in the best possible node at all times [1].

The cross-cloud platform is primarily intended for organizations with diverse users. Typically, the end user will submit a ticket to the 3rd level team to seek a service for purposes such as deploying new containers in the VM for development or testing and migrating containers from one VM to another VM within the same cloud or in a different cloud for various objectives. As a result, an organization's 3rd level team can use this management platform effectively to provide multiple services to end consumers. The third-level team can be regarded as a service provider that provides services to end users like developers and testers. There is a lack of migration information about how it performs in real-time migration. This work proposes a demonstrator that migrates several apps using Docker and KVM under various manually configurable parameters [2].

Live migration is a technique for migrating an application that is supported by Software Defined Networks and Network Function Virtualization. There is a shortage of support for moving VMs between multiple service providers and private and public cloud services. The primary issues emerge from data bandwidth and storage expenses during migration, which might negate cost savings.

The current generation of communication networks is developing toward a vast, heterogeneous, and flexible environment where nodes come and go, network services move around, and the network architecture changes regularly. Because of the dynamic and virtualized nature of the environment, the best node for running an application now may not be the best node in the future. This research suggests and analyses an approach for efficient and effective transmission and storage of high-duplication VM images for both instance and volume-based cloud storage [3].

Live migration is a service migration in which a service is seamlessly moved from one physical host to another. The goal of live migration is to minimize the time the service is unavailable during the move. As a result, the subsequent downtime is undetectable by the end-user and unaware that the server has been transferred (e.g., by detecting a new internet protocol (IP) address. The authors in [4] suggest that the time the copy of the basic image occurs during the preparation phase; thus, it is excluded from the migration time.

VM migration refers to moving VMs from one cloud system to another without interrupting the running applications. It has evolved into a running process in large-scale computing resources to overcome the fault tolerance system, traffic load balancing, and system recovery with zero downtime. Virtualization can increase the high throughput computing system, as the VMs are treated as jobs to be completed. However, there has been little research on exploiting live migration in the data center approach to avert failures [5].

A managing fault tolerance strategy is used to deal with failures caused by user interruption. Proactive migration is a well-known method for preventing failures affecting the system performance in virtualized settings. The system monitors its nodes and anticipates which ones are about to fail, allowing jobs to be migrated to different and more stable nodes. Experiments to perform VM is the task of an ailing node. The VMs were created and migrated using the Xen hypervisor. The hardware is also monitored and managed using different hardware control systems in the data center [6].

Platform as a Service is a popular trend to allow quick application deployment. The use of cloud resources has rapidly increased in recent years, and many businesses are migrating their applications to the cloud. Cloud platforms eliminate many complex configurations necessary to construct scalable applications by offering a controlled environment. On the other hand, migrations to and from clouds incur development costs and introduce additional vendor lock-in concerns. This process is problematic because repeated migrations may be required in the dynamic and fast-changing cloud industry [7].

The main goal of the research is to allocate resources better, resulting in more efficient services. Live VM migration has acquired a lot of transactions as data center efficiency by utilizing load and reducing system downtime during recovery and resiliency. Less power usage, fewer units-replicate the VM's batch, RAM, and zero downtime memory to ensure less friction in the VM's services are all part of the data center efficiency. Improved resource allocation, more efficient services, VM health, unused memory utilization, and over failure contribute to data center efficiency. The most important aspect is data transfer [8].

VM migration metrics can be calculated by measuring the VM downtime, link time, number of pages, dirty page rate, and total migration time as shown in Fig. 1. During migration, application degrades time, and consumer power is also significant.

Cloud-to-Cloud (C2C) migration allows businesses to transition between cloud environments without redoing their whole system. During the C2C migration, several VMs are migrated from one cloud computing environment to another. Because traditional migration strategies halt the VMs for a period, services operating on many VMs become unavailable during the C2C migration. Inappropriate C2C migration operations may raise the migration cost owing to the higher downtime of the service installed on the VMs [9].

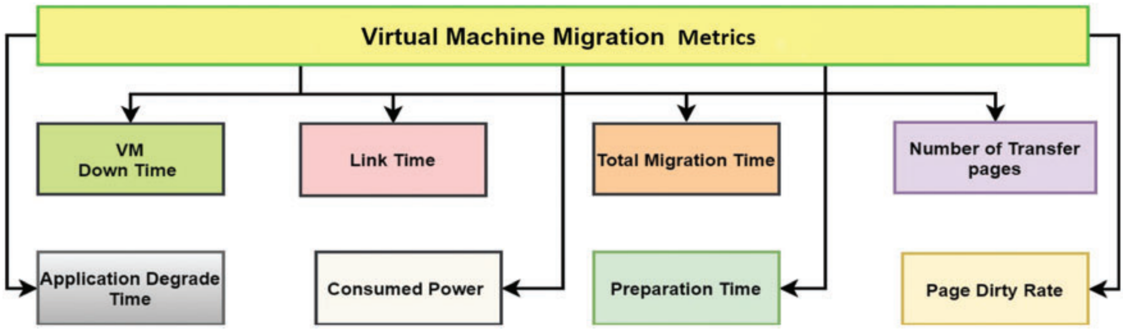


Figure 1: VM migration metrics

The key factors used to determine the migration performance operation are the source to destination time and transferred data. The amount of data moved is a critical component in the success of the migration process because it directly impacts the migration time and the downtime. Execution time is the time it takes when the migration iterations start. The process begins on the source host when the VM is entirely moved and operational ends. The VM is turned off, and its CPU state is normally transferred. The time difference in suspending the VM is known as downtime. During the downtime of VM, programs are interrupted and should be kept to a minimum [10].

Cloud computing is a method of handling applications that depends on sharing diverse computer resources rather than local servers. Virtualization technology is at the heart of cloud computing. Virtual machines must be moved from one host to another because of a fault, overcrowding, or slowness in the current host computer. The transfer of an operating VM from one host to another without interrupting the current job is known as live VM migration [11].

Downtime is one of the essential aspects that must be evaluated and reviewed throughout this live VM transfer. VMs are the most common virtualization technologies for hosting cloud services. VMs allow users to share computing and networking resources. VM placement must be constantly optimized in a dynamic environment to avoid SLA (Service level agreement) violations and satisfy QoS commitments. Live VM migration is the leading technology for moving operating VMs between physical hosts without affecting their availability [12].

VM migration is essential for attaining scalability in modern data centres by maximizing resource consumption. Most earlier research focused on transferring virtual machines with high traffic demands to the cloud. It was primarily concerned with the energy consumed by virtual computers and network traffic between them. None of the previous algorithms considered the client's perspective, such as processing time for their services based on the goal and procedures utilized by researchers. Depending on the workload and environment requirements, single or many VMs are moved whenever the migration is conducted. Fig. 2 shows a generic model for each of the three groups of migration expansion.

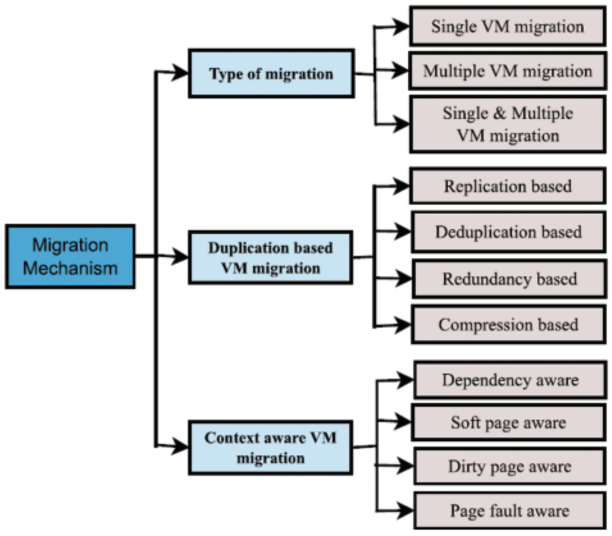


Figure 2: Migration mechanism techniques

Migration mechanisms depend on migration type, duplication, and context-aware migration. Migration types can be single, multiple, or both. Moreover, duplication base migration can be replicated, deduplicated, and compression-based. Context-aware VM migration is known for dirty page faults and dependencies based [13].

Live migration, a key component of virtualization, enables the transfer of virtual machines between physical hosts. However, for migration to be effective, a deployable feature, We must forecast migration timeframes with huge (datacenter) scale accuracy. This study describes the factors that impact live migration, focusing on virtualization in a particular platform. We go over the connections between the crucial factors that influence migration and demonstrate how migration the workload greatly impacts performance. We presented a Matlab simulation that can forecast a VM performance during migration times for both synthetic and actual benchmarks.

2 Related Works

The authors in [14] discovered specific characteristics that impact real-time migration performance in the system and network settings utilizing the OpenStack platform. The live VM migration was initially offered compared to the simple stop-and-copy strategy. Rapid dirtying pages that often change, known as a writable working set, were developed during the iterative memory copy phase of live migration employed in the Xen virtualization architecture. To reduce total migration time and sent data, certain pages will not be forwarded to the target site during the iteration round.

For technical reasons, the snapshots of a VM taken on one host system with its hypervisor may be transferred to another host machine; VM migration occurs hypervisor faces the delayed, interrupted, relocated, and relocated for a new VM. Suppose the earlier backup sync with regular intervals. This procedure may be relatively quick, allowing the VM to uninterrupted services to the data center link [15].

A working operating system can be moved to another data center with minimal downtime with live migration. Performance migration is complicated in bare-metal clouds, renting physical computers rather than virtual machines to provide the highest potential hardware performance. Live migration is difficult to implement in bare-metal clouds due to the lack of virtualization software. Previous research has explored OS-level, live migration should be OS-independent to avoid user participation and expand OS possibilities. Furthermore, live migration operations can result in little overhead as possible [16].

Containers are isolated and essentially self-contained, allowing for checkpoints and migration. OS-level containers can also support live migration. However, the number of available kernels is limited because the implementation strongly depends on the underlying kernel [17]. Virtualization overhead is also low with containers.

The Internet of Things (IoT) is a promising concept that allows numerous applications to connect via the internet. Such IoT applications generate a large amount of data that are processed, stored, and analyzed using cloud computing infrastructure and platform as a service. The IoT application placement and execution on the cloud is a difficult task. Sudden changes in the sensing environment produce spikes in data pouring into the cloud in cloud-based IoT applications, which causes the VM to run out of resources, forcing it to migrate from one physical server to another. On the other hand, unplanned migration significantly degrades a cloud-based application's performance. Choosing the correct destination server for the VM during migration is critical [18].

VM live migration is essential for Cloud deployment because it allows for quick results in high migration performance and throughput of VMs. However, the CPU health and allocated resources will affect VM live storage migration performance. The study in [19] presents the technique to simultaneously increase VM and migration performance.

The migration thread will read data from the VM's virtual disc images from start to finish and sync the modified data to the destination server most of the time. The transformation process is an ideal sequential task from the standpoint of the Sequential IO attribute since it will read data solely sequentially within the space of the virtual disc images and deserves good performance; however, this is not the case [20].

Customers' needs are driving the construction of more data centers. IT service providers should pick cloud-based services for flexibility, stability, and scalability to fulfill the growing demand for processing capacity. On the other hand, the data centers consume a lot of energy, which attracts criticism. To solve these concerns, researchers offer energy-efficient algorithms that can reduce energy usage while maintaining a reasonable quality of service (QoS). One strategy for ensuring energy-QoS balance is VM consolidation. The authors in [21] investigate a fuzzy logic and heuristic-based VM consolidation strategy to achieve an energy-QoS balance.

The process of relocating a running application to a different physical location with minimal downtime is known as live migration. Live migration has several benefits to current cloud-based systems. Furthermore, various cloud providers offer new resources for flexibility in transformation and shift their workloads to enhance the performance or commercial goals with interoperability. The few proposed solutions and other techniques are available for improvement. We focus on designing and implementing live container migration between cloud providers as containerized applications become more prevalent [21].

Users' constantly increasing computing demands to prevent SLA violations call for efficient cloud resource management. In order to efficiently manage resources, virtualization co-locates numerous virtual machines (VMs) on a single physical server. However, The choice of "what" and "where" to place workloads significantly impacts the performance of hosted applications workloads. Cloud schedulers assume RAM as the sole resource for co-locating workloads. a SLA breach as a result of improper VM placement [22].

Due to dynamic resource supply and a pay-as-you-go business model, cloud computing has become one of the most popular computational paradigms. The service providers host numerous services in large data centres. Massive energy consumption by these data centres raises operating expenses and carbon footprints. Green cloud computing is therefore essential, as it decreases energy usage and has a good environmental impact. The workload consolidation strategy, which consolidates the jobs in the fewest number of servers possible, is used to reduce energy usage. However, workload consolidation may result in SLA violations due to the server's resources being unavailable. Therefore, the issue mentioned above should be taken into account by workload consolidation approaches [23].

3 Proposed Methodology

Cloud computing has recently become the most widely used computing paradigm. Following the widespread use of Cloud Computing, many developers are considering moving their apps to the cloud to take advantage of the benefits of this new environment. Because cloud users move around so much, maintaining the requisite quality of service (QoS) at a reasonable cost is difficult. The present cloud systems leverage some of the existing service replication and migration mechanisms to maintain the appropriate QoS for transferring data from one source to another. However, when data transfers between locations regularly, those policies become ineffective due to many factors, like migration parameters. The failure of existing migration solutions is a critical criterion related to page dirty rate context during service relocation or replication. The following proposed system model consists of visualization of two data centers: source machine (data center-1) and designation machine (data center 2), which covers the different migration operations like rapid development, snapshotting, and migration.

In broader terms, the proposed system model is divided into four major modules.

- Source data center

- Destination data center
- Operations of migration
- Migration module

Migration governance management, VM security management, service management, and audit datastore play a vital role in the migration process. They maintain the system functioning by providing services to the other data center. The monitoring parameter module is responsible for all service-related functions, such as server downtime, VM monitor, and datastore instances. The Audit datastore enables log access, general system monitoring, and event record if any problem occurs during migration. Timing and scheduling are other aspects of the migration process for activities like rapid development, VM snapshotting, and migration. A poorly scheduled snapshotting or migration process may have a negative impact and increase the page dirty rate ratio, not just for the VM but also for the collocated VMs. The performance of the host operating system has an impact on the virtual machine. The factors influenced by the host OS include disc speed, cache, and data bus rate. The operating system extensively uses virtual memory when a server runs out of memory. The frequent paging operations strain disc I/O, resulting in many discs I/O.

Fig. 3 the proposed migration module consists of four parts.

- Migration performance
- Monitoring Parameters
- Migration parameter
- Shared resources

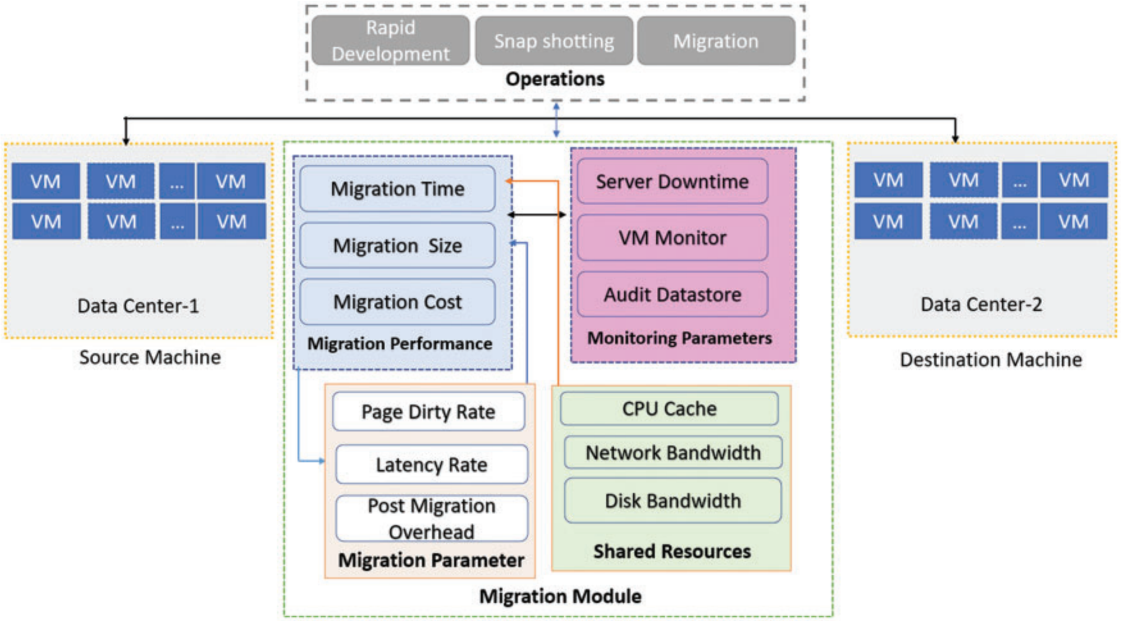


Figure 3: Our proposed migration performance model

The migration performance module is divided into custom migration file size and migration time to cater for the migration performance with coordination of monitoring parameters. The migration performance is linked with dirty page rate, latency, and post-migration overhead parameters. Additionally, the migration performance module is connected with the monitoring parameters module and is further connected with the migration parameters modules. The migration performance module is associated with the shared resources during the migration process. Shared resources may be CPU cache, network bandwidth, and disk bandwidth.

The migration performance module is connected with the monitoring module to cater to the downtime of computing and server unavailability during data transfer. The audit datastore is responsible for keeping track of all types of failures during migration.

The live migration performance can be achieved through the migration page dirty rate of VM, VM latency, and VM post-migration overhead.

4 Simulation and Results

In Matlab [24], the three input variables, page dirty rate, latency, and post-migration overhead, are used to determine the VM migration performance of VMs in a cloud environment. You can use these migration parameters to migrate control to choose which VMs to migrate from source to destination data center. Avoiding constant resource-intensive VM movement can improve dynamic VM consolidation performance. However, VMs that utilize a substantial amount of resources regularly should not be transferred. As a result, migration control can be used on three input categories of VMs: those that consume a consistent amount of resources and those that consume a large number of resources. These phenomena should be considered when developing a VM selection strategy during the whole transfer time of VMs migrations. Fig. 4 shows the migration performance measurement fuzzy inference system with input and output variables.

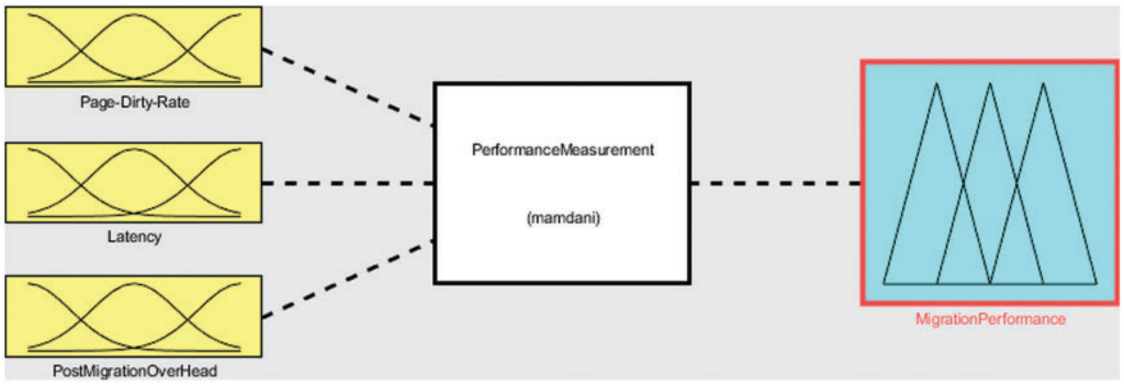


Figure 4: Migration performance measurement

This migration input will help to find out the performance of migration. Each input variable is divided into three linguistic signs: low, medium, and high. The range values of the input variables are set according to the dirty page rate of VM, and if the dirty page rate is high, then migrations performance is low. Page dirty rate calculated, the post-migration overhead of migrated VMs calculated. Similarly, the latency is mapped with linguistic values low, medium-high, and post-migration overhead that calculates the share resources overhead. Latency time, the delay in the network, is calculated in linguistic values none, low and high.

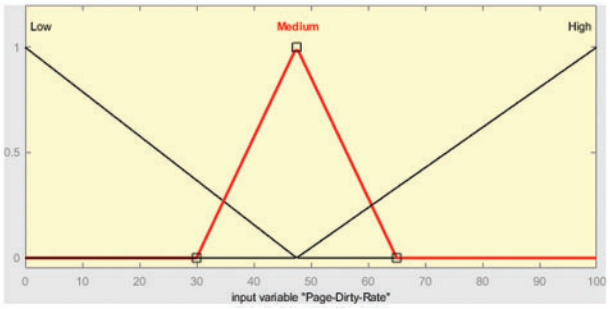
Table 1 shows the input variables, ranges, and the fuzzy set linguistic sign. The plot graph of membership output value is mapped between 1 and 100. In our proposed Fuzzy inference engine, values for calculations for input variables like; dp = dirty page, L = latency, Ov = Post migration overhead, and P = performance are used.

Table 1: Input variables

Input variables	Ranges	Fuzzy set linguistic sign
Page dirty rate	[0 20 35]	Low
	[35 47.7 65]	Medium
	[55 78 100]	High
Latency	[0 0 1]	None
	[1 27 55]	Low
	[40 70 100]	High
Post migration overhead	[0 1 1]	Low
	[1 3.5 6]	Medium
	[5 7.5 10]	High

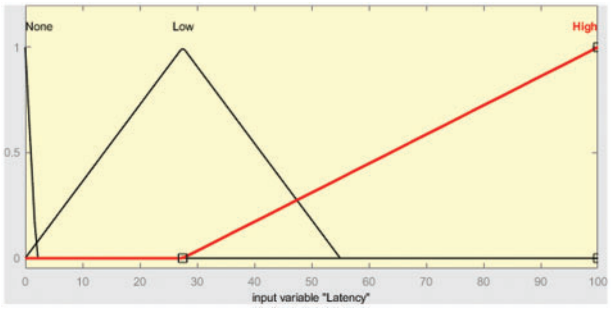
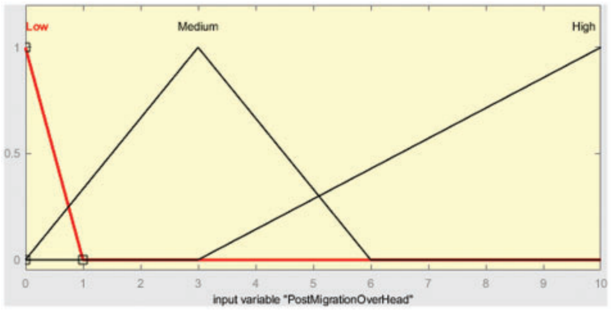
Table 2 shows the membership functions and the graphical representation to map the input variables to a particular set.

Table 2: The membership functions

Input variables	Triangular membership functions	Graph of membership function
1.dp ($\mu_{dp}(a)$)	$\mu_{dp,low}(a) = \begin{Bmatrix} \frac{a}{35} & if\ a\epsilon\,[0,\ 35] \\ \frac{47.5 - a}{35} & if\ a\epsilon\,[35,\ 47.5] \end{Bmatrix}$ $\mu_{dp,medium}(a) = \begin{Bmatrix} \frac{a - 30}{17.5} & if\ a\epsilon\,[30,\ 47.5] \\ \frac{65 - a}{17.5} & if\ a\epsilon\,[47.5,\ 65] \end{Bmatrix}$ $\mu_{dp,high}(a) = \begin{Bmatrix} \frac{a - 55}{23} & if\ a\epsilon\,[55,\ 78] \\ \frac{100 - a}{22} & if\ a\epsilon\,[78,\ 100] \end{Bmatrix}$	

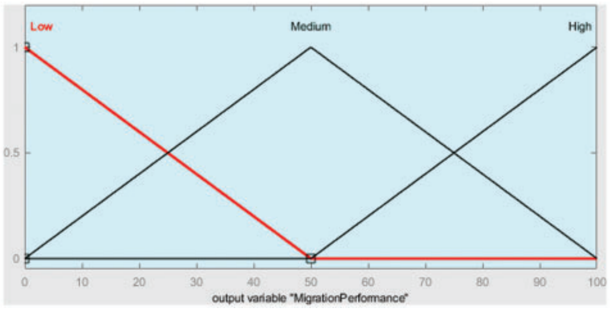
(Continued)

Table 2: Continued

Input variables	Triangular membership functions	Graph of membership function
2. L(μl(l))	$\mu l_{,none}(l) = \begin{Bmatrix} \frac{l}{1} & if\ l\epsilon\,[0,\ 1] \\ \frac{1-l}{1} & if\ l\epsilon\,[1,\ 2] \end{Bmatrix}$ $\mu_{l,low}(l) = \begin{Bmatrix} \frac{l-1}{26} & if\ l\epsilon\,[1,\ 27] \\ \frac{55-l}{28} & if\ l\epsilon\,[27,\ 55] \end{Bmatrix}$ $\mu_{l,high}(l) = \begin{Bmatrix} \frac{l-6}{2} & if\ l\epsilon\,[40,\ 70] \\ \frac{100-l}{30} & if\ l\epsilon\,[70,\ 100] \end{Bmatrix}$	
3. OV(μ_{ov}(o))	$\mu_{ov,low}(o) = \begin{Bmatrix} \frac{o}{1} & if\ o\epsilon\,[0,\ 1] \\ \frac{2-o}{1} & if\ o\epsilon\,[1,\ 2] \end{Bmatrix}$ $\mu_{ov,medium}(o) = \begin{Bmatrix} \frac{o-1}{1} & if\ o\epsilon\,[1, 3.5] \\ \frac{6-o}{2.5} & if\ o\epsilon\,[3.5,\ 6] \end{Bmatrix}$ $\mu_{ov,high}(o) = \begin{Bmatrix} \frac{o-5}{2.5} & if\ o\epsilon\,[5,\ 7.5] \\ \frac{10-o}{2.5} & if\ o\epsilon\,[7.5,\ 10] \end{Bmatrix}$	

(Continued)

Table 2: Continued

Input variables	Triangular membership functions	Graph of membership function
4. P(μ$_{per}$(p))	$\mu_{per,low}(p) = \left\{ \begin{array}{ll} \frac{p}{25} & if\ p\epsilon\,[0,\ 25] \\ \frac{50-p}{25} & if\ p\epsilon\,[25,\ 50] \end{array} \right\}$ $\mu_{per,medium}(p) = \left\{ \begin{array}{ll} \frac{p-50}{50} & if\ p\epsilon\,[0,\ 50] \\ \frac{50-p}{50} & if\ p\epsilon\,[50,\ 100] \end{array} \right\}$ $\mu_{per,high}(p) = \left\{ \begin{array}{ll} \frac{p-50}{50} & if\ p\epsilon\,[50,\ 80] \\ \frac{100-p}{20} & if\ p\epsilon\,[80,\ 100] \end{array} \right\}$	

Migration performance is calculated based on dirty page rate input variables, latency, and post-migration overhead as shown in Table 3.

Table 3: The migration performance for different output ranges

Output variable	Output ranges	Migration performance
Migration performance	[0 20 25]	Low
	[25 50 75]	Medium
	[50 75 100]	High

Eq. (1) shows the system model equation [25,26]

$$\text{Defuzzifier} = \frac{\sum_{i=0}^{n} c_i min\left\{\mu_{dp}(A), \mu_{l_i}(d), \mu_{ov_i}(o)\right\}}{\sum_{i=0}^{n} min\left\{\mu_{dp}(A), \mu_{l_i}(l), \mu_{ov_i}(o)\right\}} \tag{1}$$

The following presents the calculation of the membership function in Eq. (2) [27,28] for migration performance of input variables Pages dirty rate, latency, and post-migration overhead time.

$$\mu_{per}(Y) = Min(\mu_{dp}(X), \mu_l(X), [\mu_{ov}(X)] \tag{2}$$

$$\mu_{per}(Y) = \mu_{(dp\cap d\cap ov}(X) \tag{3}$$

$$\mu\,Per\,(dp) = \left\{ \begin{array}{ll} \frac{(1-a)\,(1-i)}{(0.65)^2} & if\ a, i\ \in [0.35,\ 1] \\ 0 & wise \end{array} \right.$$

Rules base for calculation of performance of migration of VMs is shown in Table 4.

Table 4: Performance calculation rules

IF (Page dirty rate) is	AND (Latency) Is	AND (MigrationOverhead) Is	Then migration performance
Low	None	Low	High
High	None	Low	Medium
Low	None	Medium	Medium
Medium	Low	High	Medium
High	Low	High	Low
Low	None	High	Medium
High	High	Low	Medium

Fig. 5 shows low migration performance, with values of the page dirty rate, latency, and post-migration overhead. The migration performance is overall low with a value of 16.6, while the VM page dirty rate is low with a value of 93.9, VM latency is high at 91.2, and the post-migration overhead time value is high at 9.268. Fig. 5, the output performance value is 16.6, and low migration performance is highlighted.

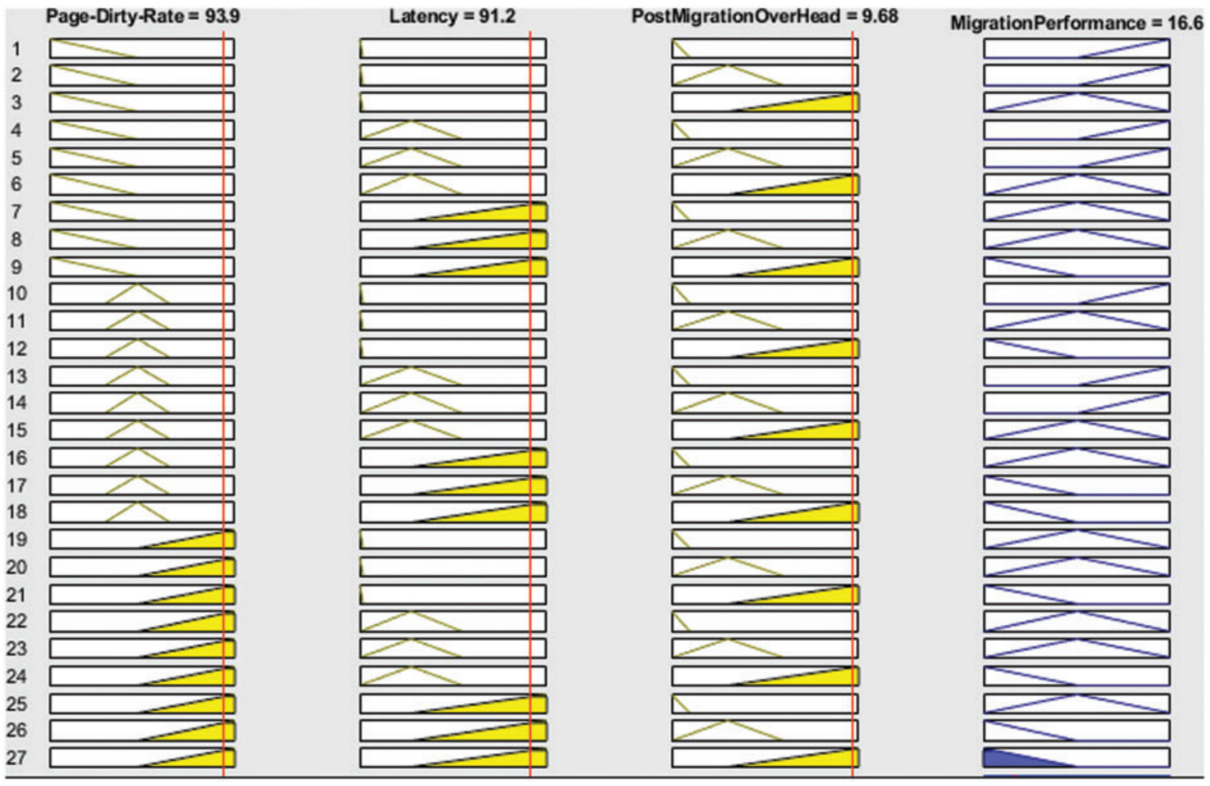


Figure 5: Low migration performance

Fig. 6 shows the medium migration performance system page, dirty rate value is 51.9, latency range is 90, and post-migration overhead value is 3.17. As a result, the output of the Migration Performance value is 50, which shows the medium migration performance.

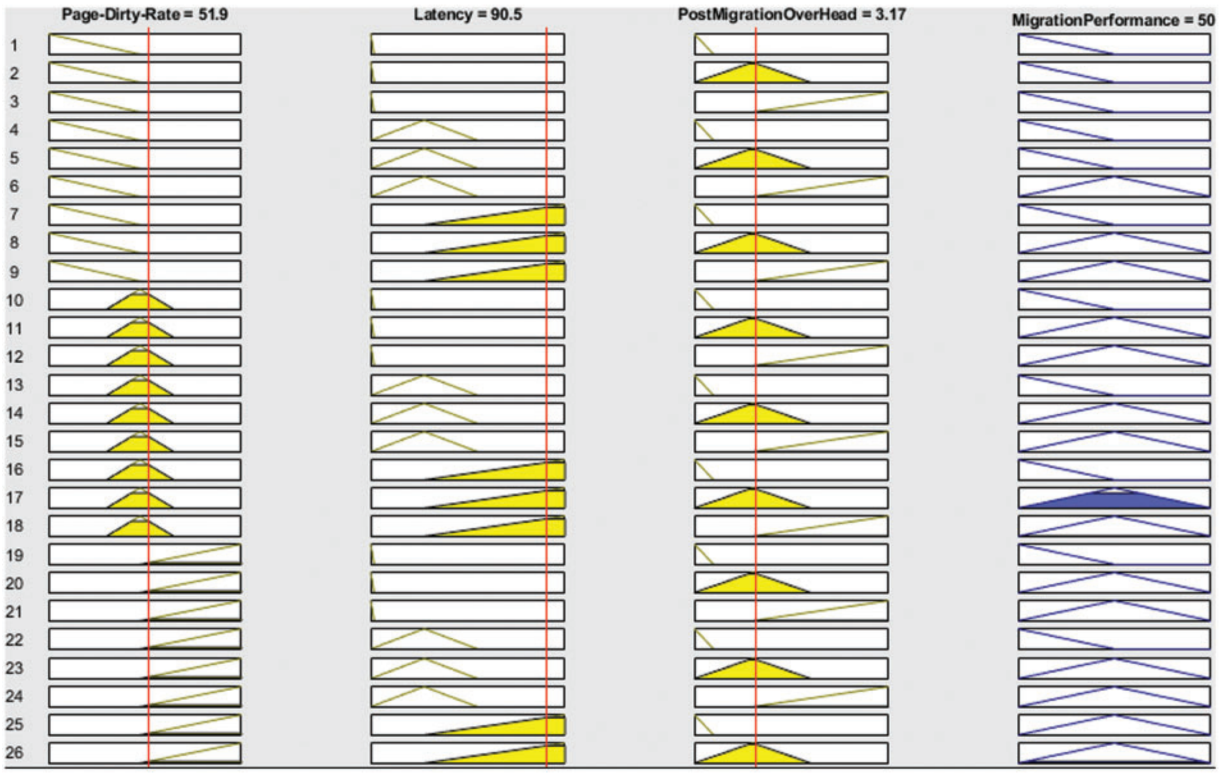


Figure 6: Medium migration performance

Fig. 7 shows the high migration performance when the VMs page dirty rate ranges are 1.15, the latency value is 24.5, and the post-migration overhead value is 0.30; the overall effect of 82.4 shows the high performance of the cloud migration.

Fig. 8 is the graphical representation of migration performance prediction. It clearly shows that the performance prediction is low if latency is 0–20 and the page dirty rate is 0–40. The migration performance prediction is medium if the latency is low and the page dirty rate is 50–80. The migration performance prediction is high if latency is 60–100 and the page dirty rate is 80–100.

Fig. 9 is the graphical representation of migration performance prediction. It clearly shows that the performance prediction is low if post migration overhead is 0–20 and page dirty rate 0–40. If post migration overhead is low and the page dirty rate is 50–80, the migration performance prediction is medium. If postmigration overhead is 60–100 and page dirty rate 80–100, the migration performance prediction is high.

The process of migration includes some necessary resources and a lot of bandwidth which is cost-effective and time-consuming. The extimated time complexity of performance migration is O (migration time x latency time).

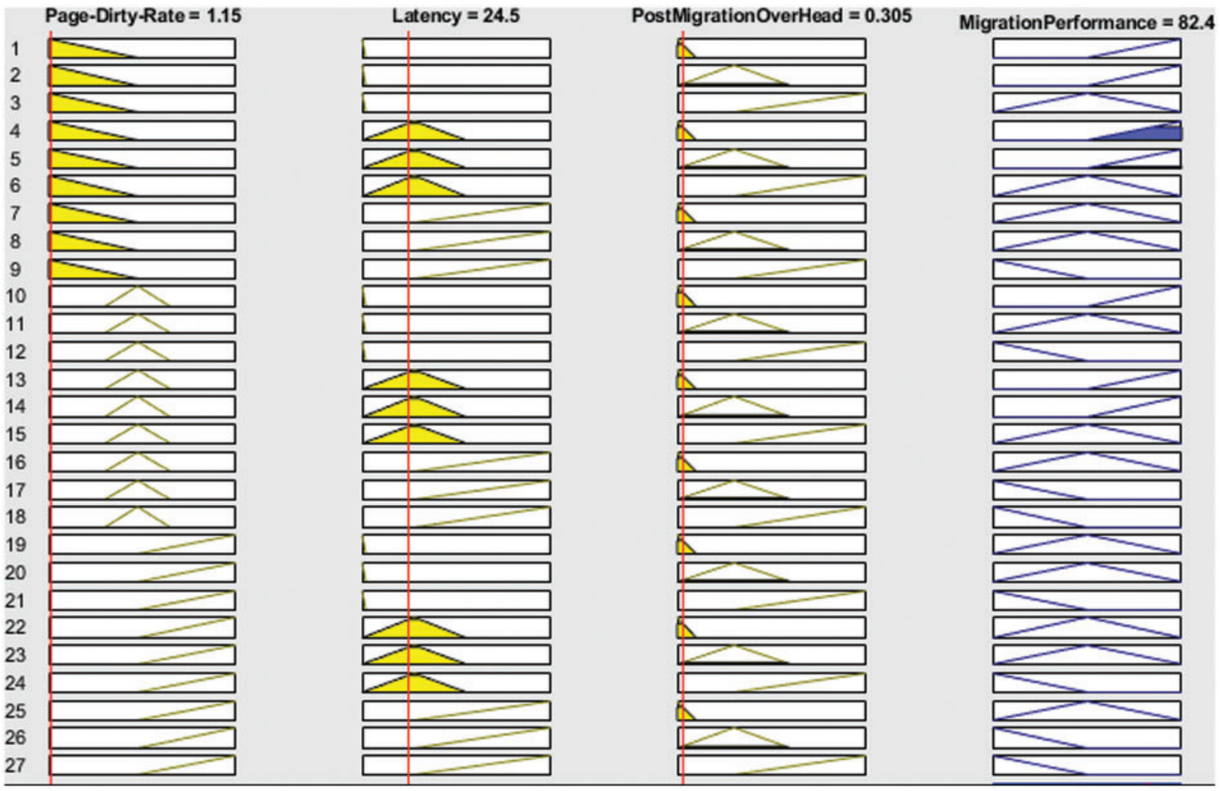


Figure 7: High migration performance

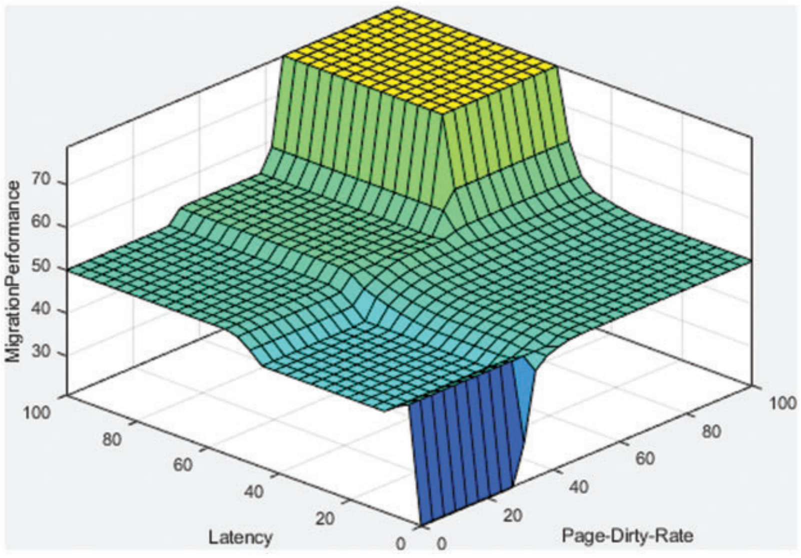


Figure 8: Rule surface of the proposed migration performance model

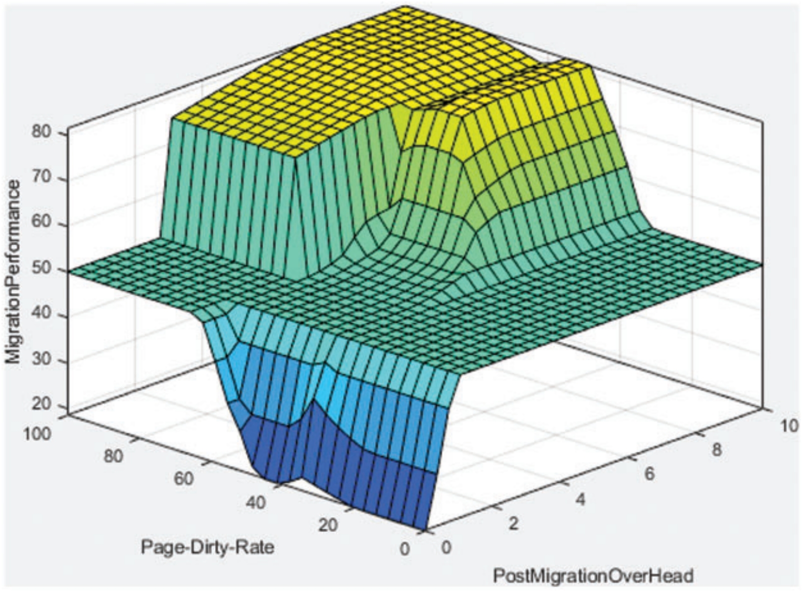


Figure 9: Rule surface of the proposed migration performance model

5 Limitations of Work

The emerging challenge to incorporating the situation of two separate data centers is due to the heterogeneous nature of cloud computing. Migration is a technique for moving an entire virtual computer from one physical system to another. Other characteristics, such as network speed, VM size, and dirty page rate, can also enhance it.

6 Conclusion

VM live migration is a critical feature of virtualization that allows VMs to be moved from one location to another without being suspended. Cloud computing service providers provide a cost-effective way to meet peak computational demands by utilizing live migration of VMs for effective and efficient workload migrations with little downtime. On the other hand, the cloud infrastructure reaches an astounding limit in terms of cost. As a result, the migration concept is used in cloud computing systems to improve data movement performance from one cloud to another with less dirty page ratio, low latency, and less overhead migration. The proposed framework enhances the VM migration in achieving better cloud data usage.

Acknowledgement: Thank you to our researchers for their collaborations and technical assistance.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Tabassum, A. Ditta, T. Alyas, S. Abbas and M. A. Khan, "Prediction of cloud ranking in a hyper-converged cloud ecosystem using machine learning," *Computer Materials Continua*, vol. 67, no. 3, pp. 3129–3141, 2021.

[2] A. Alzahrani, N. Tabassum, K. Alissa, Q. Abbas, Y. Alsaawy *et al.,* "Hybrid approach for improving the performance of data reliability," *Sensors*, vol. 22, no. 16, pp. 1–19, 2022.

[3] A. Martin, A. Raponi, S. Combe and D. Pietro, "Docker ecosystem–vulnerability analysis," *Computer Communications*, vol. 122, pp. 30–43, 2018.

[4] K. Ye, H. Shen, Y. Wang and C. Xu, "Multi-tier workload consolidations in the cloud: Profiling, modeling and optimization," *IEEE Transactions on Cloud Computing*, vol. 71, no. 3, pp. 1–9, 2020.

[5] M. Shifrin, R. Mitrany, E. Biton and O. Gurewitz, "VM scaling and load balancing via cost optimal MDP solution," *IEEE Transactions on Cloud Computing*, vol. 71, no. 3, pp. 41–44, 2020.

[6] M. Ciavotta, G. Gibilisco, D. Ardagna, E. Nitto, M. Lattuada *et al.,* "Architectural design of cloud applications: A performance-aware cost minimization approach," *IEEE Transactions on Cloud Computing*, vol. 71, no. 3, pp. 110–116, 2020.

[7] P. Kryszkiewicz, A. Kliks and H. Bogucka, "Small-scale spectrum aggregation and sharing," *IEEE Journal Selected Areas Communication*, vol. 34, no. 10, pp. 2630–2641, 2016.

[8] G. Levitin, L. Xing and Y. Xiang, "Reliability vs. vulnerability of N-version programming cloud service component with dynamic decision time under co-resident attacks," *IEEE Transaction Server Computing*, vol. 1374, no. 3, pp. 1–10, 2020.

[9] M. Aslanpour, M. Ghobaei and A. Nadjaran, "Auto-scaling web applications in clouds: A cost-aware approach," *Journal Network Computer Application*, vol. 95, pp. 26–41, 2017.

[10] T. He, A. N. Toosi and R. Buyya, "Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers," *Journal of Parallel Distribution Computing*, vol. 131, pp. 55–68, 2019.

[11] M. A. Altahat, A. Agarwal, N. Goel and J. Kozlowski, "Dynamic hybrid-copy live virtual machine migration: Analysis and comparison," *Procedia Computer Science*, vol. 171, no. 2019, pp. 1459–1468, 2020.

[12] O. Alrajeh, M. Forshaw and N. Thomas, "Using virtual machine live migration in trace-driven energy-aware simulation of high-throughput computing systems," *Sustainable Computing Informatics System*, vol. 29, pp. 100468, 2021.

[13] S. Padhy and J. Chou, "MIRAGE: A consolidation aware migration avoidance genetic job scheduling algorithm for virtualized data centers," *Journal of Parallel Distribution Computing*, vol. 3, pp. 1043–1055, 2021.

[14] Z. Li, S. Guo, L. Yu and V. Chang, "Evidence-efficient affinity propagation scheme for virtual machine placement in data center," *IEEE Access*, vol. 8, pp. 158356–158368, 2020.

[15] T. Fukai, T. Shinagawa and K. Kato, "Live migration in bare-metal clouds," *IEEE Transaction of Cloud Computing*, vol. 9, no. 1, pp. 226–239, 2021.

[16] Y. Chapala and B. E. Reddy, "An enhancement in restructured scatter-gather for live migration of virtual machine," in *Proc. of 6th Int. Conf. Invention Computing Technology ICICT 2021*, New York, USA, no. 6, pp. 90–96, 2021.

[17] N. Naz, S. Abbas, M. Adnan and M. Farrukh, "Efficient load balancing in cloud computing using multi-layered mamdani fuzzy inference expert system," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 3, pp. 569–577, 2019.

[18] N. Walia, H. Singh and A. Sharma, "ANFIS: Adaptive neuro-fuzzy inference system-a survey," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 123, no. 13, pp. 1–9, 2015.

[19] S. Rizvi, J. Mitchell, A. Razaque, M. Rizvi and I. Williams, "A fuzzy inference system (FIS) to evaluate the security readiness of cloud service providers," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–17, 2020.

[20] R. Rahim, "Comparative analysis of membership function on mamdani fuzzy inference system for decision making," *Journal of Physics*, vol. 930, no. 1, pp. 012029, 2017.

[21] A. Akgun, E. Sezer, H. Nefeslioglu, C. Gokceoglu and B. Pradhan, "An easy-to-use MATLAB program (MamLand) for the assessment of landslide susceptibility using a mamdani fuzzy algorithm," *Computers & Geosciences*, vol. 38, no. 1, pp. 23–34, 2012.

[22] M. Liaqat, A. Naveed, R. L. Ali, J. Shuja and K. -M. Ko, "Characterizing dynamic load balancing in cloud environments using virtual machine deployment models," *IEEE Access*, vol. 7, pp. 145767–145776, 2019.

[23] S. Mustafa, K. Sattar, J. Shuja, S. Sarwer, T. maqsood *et al.,* "SLA-aware best fit decreasing techniques for workload consolidation in clouds," *IEEE Access*, vol. 7, pp. 135256–135267, 2019.

[24] T. He, A. N. Toosi and R. Buyya, "SLA-aware multiple migration planning and scheduling in SDN-NFV-enabled clouds," *Journal of Systems and Software*, vol. 176, no. 4, pp. 110943–110950, 2021.

[25] N. Iqbal, S. Abbas, M. A. Khan, T. Alyas, A. Fatima *et al.,* "An RGB image cipher using chaotic systems, 15-puzzle problem, and DNA computing," *IEEE Access*, vol. 7, pp. 174051–174071, 2019.

[26] T. Alyas, I. Javed, A. Namoun, A. Tufail, S. Alshmrany *et al.,* "Live migration of virtual machines using a mamdani fuzzy inference system," *Computers Materials & Continua*, vol. 71, no. 2, pp. 3019–3033, 2022.

[27] T. Alyas, K. Alissa, M. Alqahtani, T. Faiz, S. A. Alsaif *et al.,* "Multi-cloud integration security framework using honeypots," *Mobile Information Systems*, vol. 2022, pp. 1–12, 2022.

[28] S. Ali, H. U. Khan, A. Samad, K. Alissa, M. A. Saleem *et al.,* "Container performance and vulnerability management for container security using docker engine," *Security and Communication Networks*, vol. 70, no. 1, pp. 1127–1140, 2022.