

Live Migration of Virtual Machines Using a Mamdani Fuzzy Inference System

Tahir Alyas¹, Iqra Javed¹, Abdallah Namoun², Ali Tufail², Sami Alshmrany² and Nadia Tabassum^{3,*}

¹Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan

²Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, 42351, Saudi Arabia

³Department of Computer Science, Virtual University of Pakistan, Lahore, 54000, Pakistan

*Corresponding Author: Nadia Tabassum. Email: nadiatabassum@vu.edu.pk

Received: 27 April 2021; Accepted: 17 August 2021

Abstract: Efforts were exerted to enhance the live virtual machines (VMs) migration, including performance improvements of the live migration of services to the cloud. The VMs empower the cloud users to store relevant data and resources. However, the utilization of servers has increased significantly because of the virtualization of computer systems, leading to a rise in power consumption and storage requirements by data centers, and thereby the running costs. Data center migration technologies are used to reduce risk, minimize downtime, and streamline and accelerate the data center move process. Indeed, several parameters, such as non-network overheads and downtime adjustment, may impact the live migration time and server downtime to a large extent. By virtualizing the network resources, the infrastructure as a service (IaaS) can be used dynamically to allocate the bandwidth to services and monitor the network flow routing. Due to the large amount of filthy retransmission, existing live migration systems still suffer from extensive downtime and significant performance degradation in cross-data-center situations. This study aims to minimize the energy consumption by restricting the VMs migration and switching off the guests depending on a threshold, thereby boosting the residual network bandwidth in the data center with a minimal breach of the service level agreement (SLA). In this research, we analyzed and evaluated the findings observed through simulating different parameters, like availability, downtime, and outage of VMs in data center processes. This new paradigm is composed of two forms of detection strategies in the live migration approach from the source host to the destination source machine.

Keywords: Cloud computing; IaaS; data centre; storage; performance analysis; live migration



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.

1 Introduction

Infrastructure as a service (IaaS) is a network service that enables access to high-level APIs in order to substructure physical resources, such as computing resources, security, data partitioning, scaling, and backup. System virtualization is the basic methodology in the Infrastructure as a Service cloud computing model, where the virtual machines are executed within a hypervisor, for example, Oracle VirtualBox and VMware ESX/ESXi. The hypervisor pool of resources can take the form of an operating system or a computing resource that supports many virtual machines. It can be based on different customer needs and reduce the overhead of services. IaaS typically involves the use of cloud computing technologies, such as “OpenStack and Apache CloudStack”. Multiple virtual machines (VMs) can share the underlying physical machines to virtualization tools. Multiple virtual machines (VMs) operating on the same physical disk, known as VM consolidation, will increase cloud data center resource usage. Since the cloud infrastructure has become a popular solution, it is more critical than ever to take advantage of success opportunities and boost the cloud platform productivity [1].

The choice of major drivers is Linux containers that run on separate partitions in a single Linux kernel, running directly on the physical hardware. Linux kernels and namespaces are the core technologies of Linux kernels used to separate, protect, and manage containers. The included offers better performance than virtualization, as there is no cost. In addition, the container capacity increases dramatically and automatically with the calculation method, eliminates the problem of oversupply, and achieves usage-based payment. IaaS clouds typically provide a myriad of resources, for example VM disk-image library, VLANs, firewalls, resources storage, load balancers, and software bundles. Other services include basic processing, storage, network and computing resources to enable users to run any software, including operating systems. Although the users do not control or manage the underlying cloud structures, they can control the operating system, distribution applications and limited management of some network components [2].

According to the “Internet Engineering Task Force (IETF)”, the basic model of a cloud service is a model that provides subscribers with an IT infrastructure (virtual machines and other resources) as a service provider. Cloud service providers offer the infrastructure services as computing resources that are needed from the high capacity of Linux containers. For longer connections, customers can use the Internet or a cloud operator (i.e., dedicated virtual network). To deploy applications, cloud users install their operating system images and application software on the cloud infrastructures. In the IaaS service model, cloud users can rent and maintain an operating system, virtual servers, and various types of networking software. Cloud service providers usually charge for IaaS services based on utility calculations where the costs reflect the amount of capital allocated and resources used. Infrastructure as a service (IaaS) is a form of cloud computing that provides users with basic computing, networking, and storage resources on-demand, over the Internet, and for a fee. IaaS enables end users to expand their resources as needed, reducing the need for high capital expenditures or unnecessary “clean” infrastructure, especially during peak hours. Compared to the PaaS and SaaS models (even newer computer models such as containers and without a server), IaaS offers the lowest level of cloud resource management [3].

IaaS is a major part of modern cloud computing and has become the mainstream abstract model for many types of loads. With the advancement of new technologies such as containers and the Internet and the proliferation of small-scale applications, IaaS remains the foundation, but more prevalent than ever. Businesses choose IaaS because it is usually easier, faster, and cost-effective to implement workloads without the need to purchase, manage, and support the hardware infrastructure. With IaaS, companies can rent or lease the required infrastructure from other companies [4].

IaaS is an effective model for dealing with temporary, experimental, or unexpectedly altered workloads. For example, if a company develops a new software product, it may be more cost-effective if an IaaS provider is hired to maintain and test the applications. Once the new software is thoroughly tested and optimized, organizations may remove it from the IaaS environment for a more traditional use. Moreover, companies can opt for the long-term use of IaaS of their software, especially that long-term hosting costs are usually lower. Normally, IaaS customers pay for each use, usually on an hourly, weekly, or monthly basis. Some IaaS providers also charge customers based on the number of hours spent using the virtual machines. Licenses as such exclude the cost of financing the internal use of hardware and software. A VM's memory operations are divided into four categories: disk write, disk read, memory write, and memory read. In the pre-copy and post-copy models, we summarize the performance impacts of various processes during migration [5].

We have witnessed a constant increase in the adaptation of the cloud computing environments to host a diverse range of software applications, including web files, virtual reality, big data, and scientific computing [6]. Subsequently, the provision of cloud services that satisfy a certain level of quality (i.e., QoS) has become a critical matter with time. Recent works focused on optimizing energy use and reducing service latency dynamically to ensure maximum benefits are delivered to the cloud computing providers and users. In this research, we aim to design an architecture that can strengthen and enhance the performance of cloud computing during the migration of data in data centres [7]. Cloud service providers managing and maintaining data that has been migrated to the one data center to another data center. In the cloud system, storage services are available on demand, with capacity increasing and reducing as needed. Cloud storage eliminates the need for businesses to purchase, operate, and maintain their own storage infrastructure. In Fig. 1 depicts a memory representation in virtualization, where the guest OS and hypervisor are hosted on a host machine, which shares the guest virtual memory and guest physical memory.

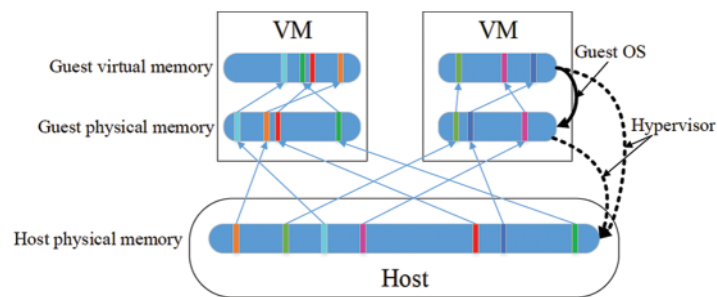


Figure 1: A memory representation in the cloud environment

A society cloud in figuring is a communication effort in which the structure is shared between a couple of relationships from a specific gathering with fundamental concerns (e.g., security, consistency, ward, etc.), paying little respect to whether regulated inside or by a pariah and encouraged inside or remotely. This is controlled and used by a social occasion of affiliations that have shared interest, the costs are spread over fewer customers than an open cloud (however more than a private cloud), so only a segment of the cost venture stores capacity of dispersed registering make sense of it [8].

From a migration point of view, a VM can be separated into three components namely: 1) operating states (including memory data, CPU states, all external system states, and so on), 2) storage data (e.g., disk data), and 3) network links with the VM and its users. In essence, the live VM migration endeavours to move the afore-named components from a source to a destination. Data and real-time

status transfer from the memory of a VM to the new host must be accomplished to prevent any interruption to the services running in the migrated VM. These data include CPU states, memory data, external system buffer data, and so on. Memory storage migration is the term used to describe the transition of operating states. Data transfer from storage involves moving a VM's disk image to a new host. This task is required when the source and destination hosts do not share a storage pool. When a virtual machine (VM) is relocated to a different site, a plan is required to redirect the users' network connections to the new location and maintain the network link [9].

The live migration in a cloud environment from a source host to a destination host is highly reliant on the network connection continuity between the user end and virtual machine. The data migration process shifts the data from the source host to the destination host as shown in Fig. 2. The main dependency is connectivity and complete migration of data from the source host to the destination host. When a server is overloaded, its lifetime is shortened and quality of its services (QoS) deteriorates as a result. On the other hand, servers that are underutilised cause a waste of resources. Live VM migration means that all servers in a data center run at the same time with no degradation to the QoS. When a live VM replication over a WAN is carried out, load balancing can also be achieved between many geo-distributed data centers [10].

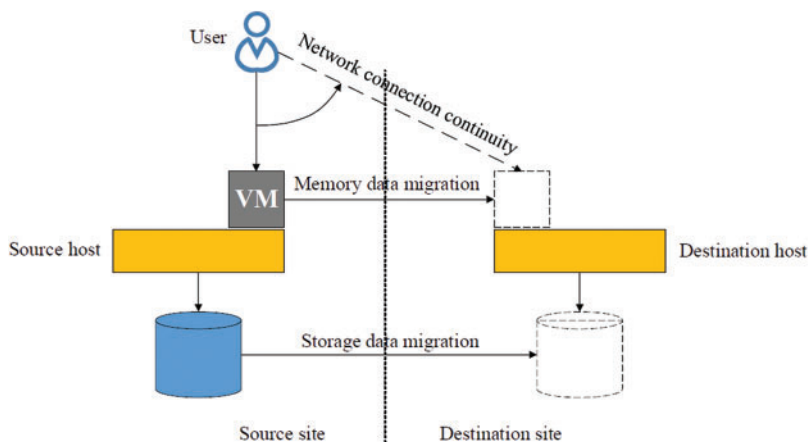


Figure 2: Cloud live migration from a source host to a destination host

In a data center, virtual machines are built and dismantled on a regular basis. Furthermore, any of the VMs may be inactive or suspended. If the servers in a data center are not sufficiently consolidated, the VMs would be in a mess. VMs are live migrated during server consolidation for energy efficiency (using as few servers as possible) or connectivity (locating the VMs communicating heavily with each other on the same server to reduce the network traffic) [11].

2 Related Works

In [12], authors identified specific variables that affect real-time migration performance using the OpenStack platform in the system and network environments, such as real-time static adaptation metric comparisons, parallel migration performance comparisons, and SDN sequencing. From a QoS perspective, response time and server response patterns, duplication, hybridization, and self-convergence-based migration need to be evaluated. A real-time mathematical model of the final transmission can be created to understand the static adjustment algorithm in OpenStack and the parallel and continuous transmission costs on the same network path.

In order to adjust the working time, the idle settings (maximum idle time, adjustment steps, and delay) need to be adjusted. This can help to get the best transmission results. If off-network costs (such as pre and post-transmission workload) account for a large proportion of the total transmission time, simultaneous transmission should be chosen to reduce response time and idle time. In cloud-based software centers with a software-defined network (SDN), real-time VM transmission is a key technology for promoting resource management and tolerance. Although many studies focus on the transmission of a network-sensitive real-time virtual machine to cloud computing, however, some variables such as service level objective for backup storage data, increase resource usage, load balancing of VMs that affect real-time transmission performance are largely ignored. In addition, while SDN offers more traffic flexibility, SDN delays directly affect live broadcasts [13].

Authors in [14] argued that live migration of VMs and virtual network functions (VNFs) in the SDN enabled cloud data centres is the most widely used approach. However, they claimed that there was a risk of performance degradation if this live migration was performed in a random order. Therefore, they suggested a group of algorithms to come up with a scheduled live migration.

In another article [15], authors reviewed real-time migration literature and analyzed various guidelines. Firstly, they classified real-time VM migration types (single, multi, and hybrid). Next, they classified VM migration technology based on copy methods (copying, duplication, aggregation, and compression) and contextual awareness (dependency, software pages, dirty pages, and default pages), and lastly, they evaluated different VM real-time migration technologies. Moreover, they discussed various performance metrics, such as application service downtime, total transmission time, and the total amount of data sent. A similar study was conducted in [16] where authors proposed an auto-scaling technique which they called Robust Hybrid Auto-Scaler (RHAS). They argued that RHAS could predict the future load on cloud infrastructure that in turn could help manage the resources optimally and could have positive impact on cost, response time, SLA violations and CPU utilization.

Authors in [17] briefly introduced the security threats associated with real-time migration and divided them into three different categories i.e., surveillance aircraft, data aircraft and costume units. They also explained some crucial security requirements and possible solutions to reduce the potential attacks. To improve the performance of real-time migration, specific gaps and research challenges were identified. The importance of this study is that it creates a context for real-time migration technology and conducts an in-depth insights that helps cloud professionals and researchers to further explore the challenges and provide the best solutions [18]. Similar efforts were seen in [19] where authors highlight the vulnerability of VM against co-resident attack. Authors in [20] proposed a probabilistic model to defend against these attacks with the help of optimal early warning mechanisms. Whereas in [21] authors proposed optimal data partition and protection policy which is based on probabilistic models. They claimed that their proposed model could provide protection against data theft in cloud environment.

Load balancing is major difficulties in cloud computing. In this paper, author mainly target on the impact of load balancing on emerging technologies which depend on cloud computing. Further, the current methods of reliable and dependable cloud computing can be achieved by efficient load balancing on cloud with IoT, Big Data & self-learning systems. It has been proved that DBLA & NDLBA algorithms are efficient and useful for succeeding generation cloud computing. The important factor of this paper is to improve the overall attainment & to maximize re virtualization for allotment of tasks on VMs [22].

Compared to the changes that occurred during transmission from different VM memory events, the idle experience changes relatively significantly as the application network latency changes. The

apparent total transport time is closely related to the size of the migrant worker's status. Virtualization offers a new way to cloud computing, allowing both small and large business setups to maintain their applications by renting available resources. Real-time VM forwarding allows virtual machines to be transferred from one host to another while the virtual machine is active and running. The main challenge for real-time WAN migration is to maintain an Internet connection during migration [23].

Moving a large number of virtual machines to high-performance servers and overloading workloads typically leads to a lack of space and reduced performance. Therefore, smaller recovery techniques can be used to perform the transfer in an acceptable manner. Before the VM is transferred to the powerful server, the entire state of the VM's memory is stored on the small host server, and after the VM is fully transferred, the rest of the powerful server is transferred. The main goal of this research [24] is to find and offer suitable solutions to improve the transmission capacity of virtual data centres.

This effectively improves real-time transmission and reduces both transmission and idle time. To implement the design of energy-efficient integration of servers, a virtual machine with less recycling technology is used in the cloud centre. Cloud computing can provide end-users with all IT technologies, from IT quality to infrastructure, applications, business processes and personal collaboration. A cloud is a set of hardware, software, networks, services, storage, and interfaces that are connected to provide computing functions and other devices as needed [25].

Similarly, in [26] authors highlight the importance of virtual machines placement with multiple resources. They proposed a unique approximation algorithm to solve this NP hard problem. They conducted real time tests with multiple cloud providers such as Azure and Google and proved that their algorithm performed better than the existing ones.

Although a number of other cloud studies have been conducted in the literature on economics, privacy, and security, there is a clear lack of research on migration, server integration, and network virtualization technology. To fill this gap, we have proposed different plans for classifying researchers' common and different views based on evidence-based literature [27].

The virtual migration engine (VM) is widely used in and between domain controllers to meet the needs of different virtual cloud environments. For example, server compression requires the transfer of virtual machines for power management. In addition, real-time migration is required to balance workload, fault tolerance, system maintenance, and to reduce SLAs. The VM transfer process is very resource-intensive and requires ingenious methods to avoid network-saturated bandwidth and to reduce server downtime. Compared to traditional computing methods, cloud computing is a promising computing paradigm. Along with the traditional computing methods, numerous organizations prefer to utilize flexible cloud services to reduce cost significantly. Virtualization is a technology widely used in today's data centres to maximize utilization of resources and to reduce greenhouse gas emissions and costs.

3 Proposed Methodology

The proposed architecture is specifically designed to cater to the aims and objectives of the study of live migration in cloud environments. Our methodology is divided into two main parts: 1) block live migration and 2) sequential and parallel migrations. Considering the implications of hardware maintenance, energy-saving policy, load balancing, or encountering devastating incidents, there will be a requirement to abandon different parts of the VMs from different physical hosts to other hosts by using the live VM migration at the earliest. The models used in our study will share a similar network

traffic path. For instance, if there are five live migrations sharing the same path of the network along with all other sources and host's destination. As each migration completely utilizes the bandwidth path, part of the network transmission is significantly small in contrary to the parallel migration.

The proposed architecture consist of source and target machine to monitor and process the migration of VMs to achieve the cloud orchestration. Source machine and target machine have its own hardware and hypervisors running different VMs on it. The migration module is further divided into four major sub modules.

- Data Mapping Module
- Log Management Module
- QoS Parameter Module
- Hardware Performance Module

To cater the performance of VMs the role of migration module is important for migration of VMs services to achieve the cloud orchestration. Data mapping module is further divided into data rate measurement, performance measurement and migration decision.

In Fig. 3 organizes the lifecycle of the VMs in a cloud computing environment from source to the target machine. The migration module is responsible for scheduling, migrating, decommissioning, and spawning VMs on request. The migration module in IaaS is facilitated through the Log management, hardware management, and data mapping module. In the data center environment source machine having own host operating system, hypervisor that running different virtual machines which needs to secure and complete the migration process. By keeping the migration track changes the role of log management is important to take migration decisions. All types of the log are recorded and helpful for quality migration and hardware performance. The hardware performance module is divided into network performance, storage performance and input/output performance. this storage continues if the VM is present and is removed after terminating the VM.

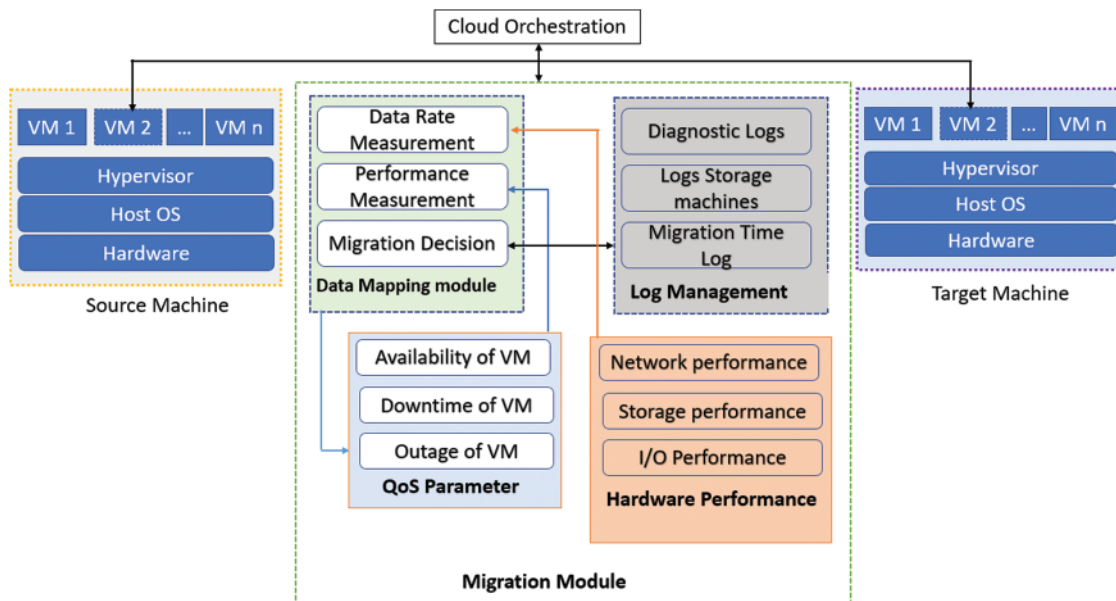


Figure 3: Our proposed IaaS cloud model for live storage migration

The CPU, storage devices, memory, and energy supply often decide the energy usage of the network resources in a data center. A linear relationship between energy consumption and CPU usage will adequately characterize a server's energy usage.

The proposed model is connected to two storages, e.g., source storage and destination storage that are linked with a network service and images service. The network performance will keep a record the VM connected to over the migration period. The Diagnostic log will keep the session identification of the source and destination service through migration. The performance of the live migration can be achieved through QoS parameters like availability of VM, downtime of VM, and outage of VM. The objective of live migration in IaaS will achieve cloud orchestration and will happen through the dashboard of two providers who will monitor the end-to-end data transfer through log management.

There are two types of VM migration: 1) non-live migration and 2) live migration. Live migration is only possible if the operating systems are not disrupted, while non-live migration is not restricted in this way. A VM is first paused or shut down before a non-live migration can take place, depending on whether it can continue to operate the services after migration or not. Live VM migration has received substantial coverage as one of the most important techniques for increasing data center utilization by efficiently allocating the resources, leading to more reliable services. In data center, performance optimization includes hardware, data mapping and QoS parameters, but no limited to, reduced power demand and less unutilized space. Moreover, in the event of repairs or breakdowns, there will be fewer service interruptions.

The aim of a live VM migration is to copy the VM's state, memory, and disk with the least amount of downtime possible, ensuring non-disruption to the VM's services. Live VM transfer approaches, including Pre-download, Post-copy, and Hybrid-copy [11], make a copy of the VM's memory states to the destination host whilst the VM is still running on the source host. Modern cloud-based applications will take advantage of live relocation, which refers to the method of moving a running program to a new physical location with minimized downtime. Furthermore, the live migration between various cloud service providers gives cloud users a new degree of flexibility, allowing them to switch their workloads around to achieve their business goals without being bound to a single provider. While this vision is not new, there are just a few solutions and proof-of-concepts available. Live migration is a technique for moving a virtualized operating service between hosts in a smooth manner, allowing networks to respond to changes quickly. The idea behind this technology is to keep the downtime of the service to a minimum so that the end-user is unaware of the migration. Docker for containers and KVM for virtual machines are the two most common off-the-shelf virtualization technologies. However, there is a lack of knowledge about how these virtualization technologies perform during real-time migration.

4 Simulation and Results

In this section, the Mamdani Fuzzy Inference System (i.e., MFIS) is used to simulate the performance of cloud storage services. In short, a Mamdani fuzzy inference system is a method that uses a set of if-then fuzzy rules defined by domain-expert operators to make decisions about the outputs of the system's behavior. There are many benefits to using the MFIS, including its widespread acceptance, intuitiveness, suitability for human input, and ease of interpretability. The description of fuzzy inference system for predicting the cloud storage service is depicted in Fig. 4. Overall, the below Mamdani FIS takes three inputs, namely storage service availability, downtime, and outage. The output of the Mamdani system represents the performance of storage services.

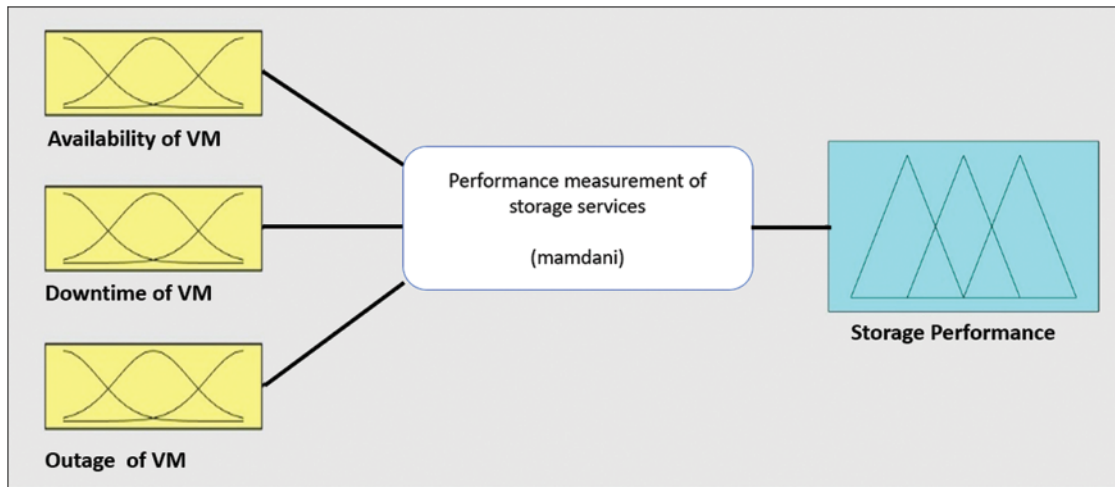


Figure 4: Performance prediction of cloud storage service for MFIS

In essence, the input parameter variables are statistical values that are used to calculate the storage performance of cloud services. [Tab. 1](#) presents the input variables. Through migration from source to target machine the total number of Available of VM on source machine which needs to migrate. Downtime of VM is calculated if the VMs become unavailable during migration process and outage of VM means the how many times the out of migrated VMs. The range values of the input variables is set according to the availability of VM, if the availability of VM is 100% then its means the virtual Machines are remained highly available during migrations. Similarly the Down time is mapped with linguistic value if no down time then the availability is 100% otherwise high downtime leads the poor performance. Outage of VMs means the total number of time the concerned VM remains out of connection during migration process. In cloud infrastructure.

Table 1: Input variables and their ranges; VMs = virtual machines

Input variables	Ranges	Linguistic variables of fuzzy sets
1. Availability of VMs	[0 20 40]	Low
	[30 47.7 65]	Medium
	[55 78 100]	High
2. Downtime of VMs	[0 0 1]	None
	[1 27 55]	Low
	[40 70 100]	High
3. Outage of VMs	[0 1 1]	Low
	[1 3.5 6]	Medium
	[5 7.5 10]	High

[Tab. 2](#) lists the membership functions that we used to map the input variables to a particular set, where the membership output value is always limited between 0 and 1. The below functions are used to fuzzify or defuzzify the inputs to/from some linguistic terms. As seen in [Tab. 2](#), our membership functions are triangular, typically limited by a lower and upper value for each fuzzy set. For each

input variable, our membership functions produce three fuzzy set outputs. We have used the MATLAB simulation software to implement the below triangular functions and visualize the depicted graphs.

Table 2: The membership functions of our mamdani FIS; Av = availability, D = downtime, Ou = Outage, and P = performance

Input Variables	Triangular Membership Functions	Graph of Membership Function
1. Av ($\mu_{.}(a)$)	$\mu_{low}(a) = \begin{cases} \frac{a}{20} & \text{if } a \in [0, 20] \\ \frac{40-a}{20} & \text{if } a \in [20, 40] \end{cases}$ $\mu_{medium}(a) = \begin{cases} \frac{a-30}{17.5} & \text{if } a \in [30, 47.5] \\ \frac{65-a}{17.5} & \text{if } a \in [47.5, 65] \end{cases}$ $\mu_{high}(a) = \begin{cases} \frac{a-55}{23} & \text{if } a \in [55, 78] \\ \frac{100-a}{22} & \text{if } a \in [78, 100] \end{cases}$	
2. D ($\mu_{.}(d)$)	$\mu_{low}(d) = \begin{cases} \frac{d}{1} & \text{if } d \in [0, 1] \\ \frac{1-d}{1} & \text{if } d \in [1, 2] \end{cases}$ $\mu_{mid}(d) = \begin{cases} \frac{d-1}{26} & \text{if } d \in [1, 27] \\ \frac{55-d}{28} & \text{if } d \in [27, 55] \end{cases}$ $\mu_{high}(d) = \begin{cases} \frac{d-6}{2} & \text{if } d \in [40, 70] \\ \frac{100-d}{30} & \text{if } d \in [70, 100] \end{cases}$	
3. Ou ($\mu_{.}(o)$)	$\mu_{low}(o) = \begin{cases} \frac{o}{1} & \text{if } o \in [0, 1] \\ \frac{2-o}{1} & \text{if } o \in [1, 2] \end{cases}$ $\mu_{medium}(o) = \begin{cases} \frac{o-1}{1} & \text{if } o \in [1, 3.5] \\ \frac{6-o}{2.5} & \text{if } o \in [3.5, 6] \end{cases}$ $\mu_{high}(o) = \begin{cases} \frac{o-5}{2.5} & \text{if } o \in [5, 7.5] \\ \frac{10-o}{2.5} & \text{if } o \in [7.5, 10] \end{cases}$	
4. P ($\mu_{.}(p)$)	$\mu_{low}(p) = \begin{cases} \frac{p}{20} & \text{if } p \in [0, 20] \\ \frac{40-p}{20} & \text{if } p \in [20, 40] \end{cases}$ $\mu_{medium}(p) = \begin{cases} \frac{p-30}{20} & \text{if } p \in [30, 50] \\ \frac{70-p}{20} & \text{if } p \in [50, 70] \end{cases}$ $\mu_{high}(p) = \begin{cases} \frac{p-60}{20} & \text{if } p \in [60, 80] \\ \frac{100-p}{20} & \text{if } p \in [80, 100] \end{cases}$	

The fuzzy output variable (i.e., performance of VMs) is determined based on the values of input variables (i.e., availability, downtime, outing). [Tab. 3](#) details of output ranges and linguistic variables, mainly low, medium, and high.

Table 3: MFIS performance output ranges

MFIS final output	Output ranges	Semantic sign of fuzzy output (performance)
1. Performance of VMs	[0 20 40]	Low
	[30 50 70]	Medium
	[60 80 100]	High

[Eq. \(1\)](#) outlines the mathematical equation for our MFIS defuzzifier

$$\text{Defuzzifier} = \frac{\sum_{i=0}^n c_i \min\{\mu_{Av_i}(A), \mu_{d_i}(d), \mu_{ou_i}(o)\}}{\sum_{i=0}^n \min\{\mu_{Av_i}(A), \mu_{d_i}(d), \mu_{ou_i}(o)\}} \tag{1}$$

The membership function of performance depends on the minimum values of membership of availability, downtime, and outage time. The mathematical equation is shown in [Eq. \(2\)](#)

$$\mu_{per}(Y) = \text{Min}(\mu_{av}(X), \mu_d(X), [\mu_{ou}(X)]) \tag{2}$$

[Eq. \(2\)](#) can also be written as

$$\mu_{per}(p) = \mu_{av}(X) \text{ and } \mu_d(X) \text{ and } \mu_{ou}(X) \tag{3}$$

Similarly, [Eq. \(3\)](#) can be written as shown in [Eq. \(4\)](#)

$$\mu_{per}(Y) = \mu_{(av \cap d \cap ou)}(X) \tag{4}$$

MFIS must be designed on top of a clear rule-based engine. There are 81 rules were formulates by keeping the input and output variables. On the basis of our input three input variables, the VM performance output values were calculated using the assigned rule listed. Few rules are displayed in [Tab. 4](#).

The simulation results ([Figs. 5–7](#)) are produced using the MATLAB R2021a software. [Fig. 5](#) depicts the lookup diagram for the low storage performance, particularly the relationship between the system performance and low input values of availability, downtime, and outage. This diagram shows that the storage performance is low (e.g., 20), when the VM availability is low (e.g., 9), VM downtime is medium (e.g., 50), and outage time is low (e.g., 0.37). [Fig. 5](#) indicates that output performance values equal to or below 20 are perceived as a poor cloud storage performance.

[Fig. 6](#) depicts the lookup diagram for the medium storage performance. The performance is in the medium range when the system availability is medium (e.g., 36), downtime is high (e.g., 76), and outage is low (e.g., 2.9). Performance values approaching 50 are considered to exhibit a medium cloud storage performance.

Table 4: The rule base engine for the VM performance measurement

IF (availability) is	AND (down time) is	AND (outage time) is	Then performance
Low	None	Low	Low
Low	None	Medium	Low
Low	None	High	Low
Low	Medium	Medium	Low
Medium	High	Low	Medium
Medium	None	Medium	Medium
High	Low	Low	High

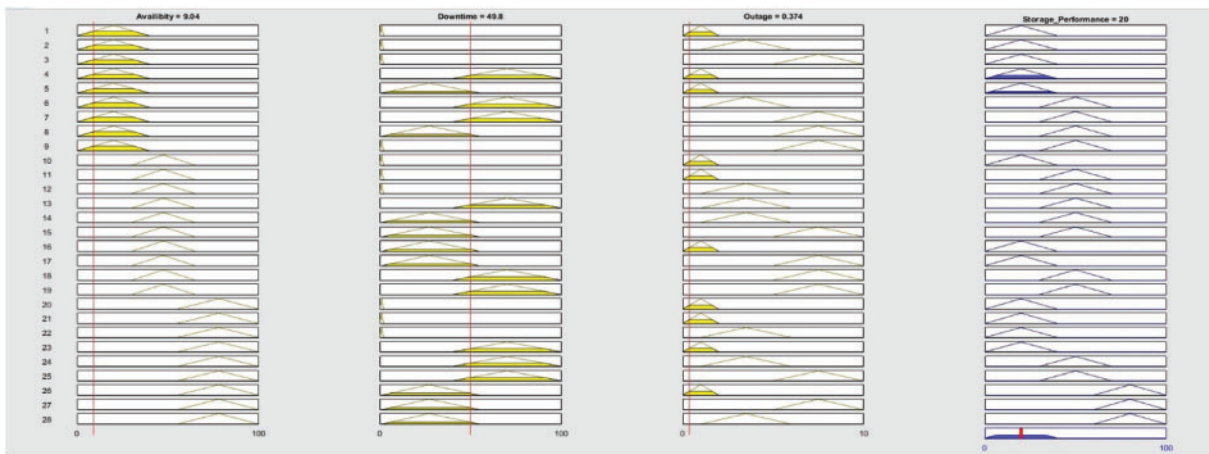


Figure 5: The lookup diagram for the low VM storage performance ranges

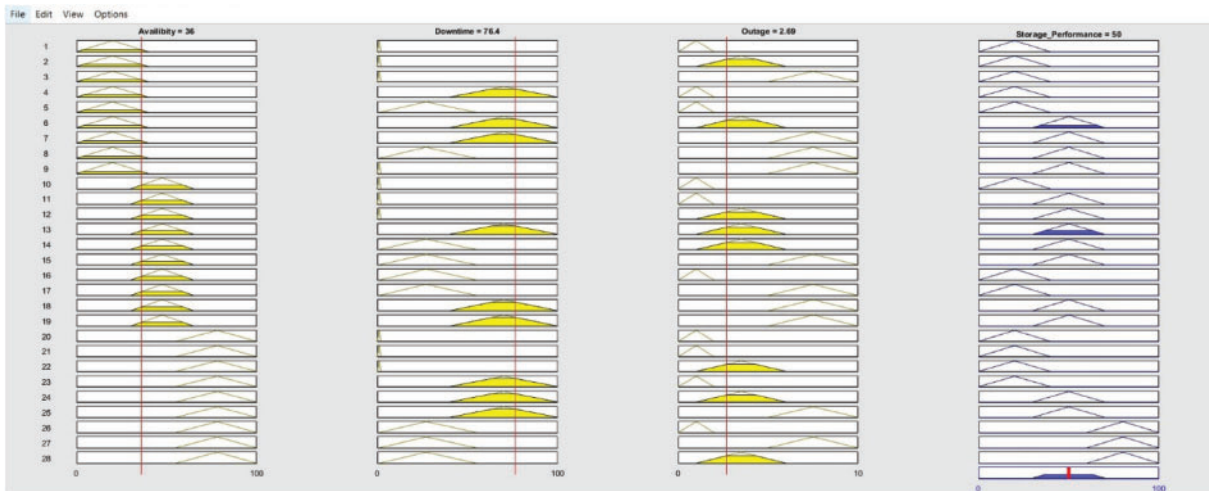


Figure 6: The lookup diagram for the medium storage performance ranges

Fig. 7 depicts the lookup diagram for the high storage performance. The system performance is considered high when the VMs availability ranges in the high values (e.g., 90), downtime is low (e.g., 16), and outage value is low (e.g., 0.5). Output values greater than 80 are considered to exhibit a high performance of the cloud storage.

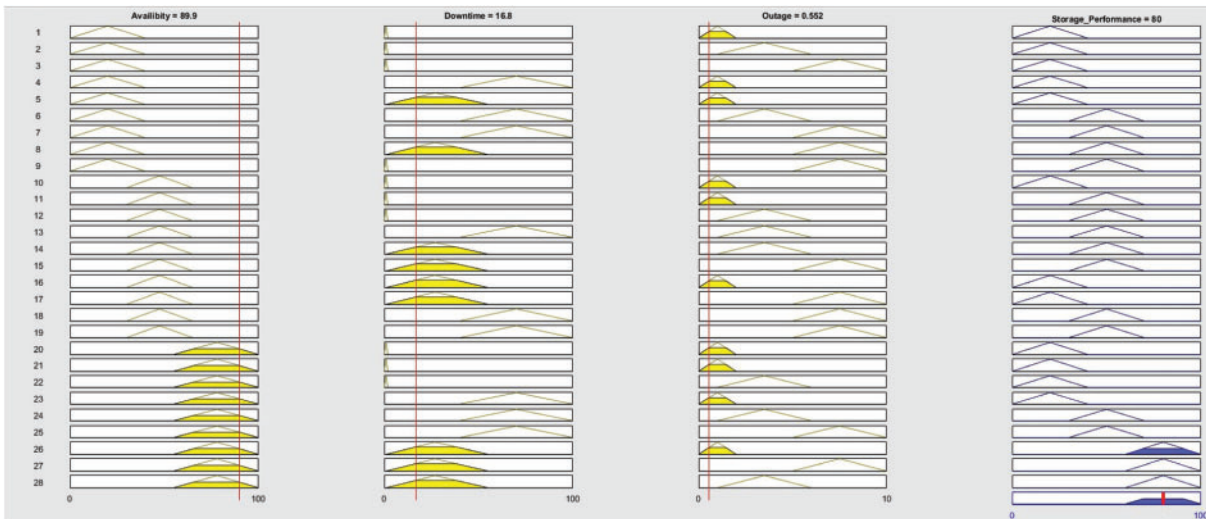


Figure 7: The lookup diagram for the high storage performance ranges

The performance of the cloud infrastructure as a service (IaaS) is checked through a Mamdani fuzzy inference system. The results were analyzed based on the network performance of the source and destination host. The cloud storage service offered by different cloud service providers, in terms of its availability time, downtime and outage time, is simulated and tested. Traditional virtualization systems are unable to effectively isolate shared micro-architectural resources among virtual machines (VMs). Different types of CPU and memory-intensive VMs competing for these common resources would result in varying degrees of performance loss, thus lowering device stability and QoS in the cloud.

5 Implications and Limitations

A live migration is a technology that allows a complete virtual machine to be relocated from one physical system to another while it is still running. Active memory and execution state are transmitted from the source to the destination when a VM is migrated as a whole. This provides for seamless online service migration without the need for clients to reconnect. Through fuzzification of the possible research problem is plotted with three input variables. However it can be improved with other variables like link speed, size of VM and page dirty rate. Due to heterogenous structure of cloud computing, it is difficult to map the real scenario of two different data center.

6 Conclusion and Future Works

Live migration mechanisms in cloud computing have many compatibility issues with pre-copy migration of the optimization of VM migration. Performance models for cloud storage during the migration procedure is designed, from which several performance features are tested and monitored the high performance. It not only considers users' requirements for migration efficiency but also solves

the storage convergence issue. By modifying the storage in live migration parameters like availability, downtime and outage from the source to destination host are tested. By adjusting the network rate of the migrated VM or/and the network bandwidth for migration, our proposed model demonstrated feasibility. This is beneficial in terms of improving cloud management in storage dashboard, are specific about the consistency they may expect before migrating their virtual machines. In the future, we aim to conduct further studies to improve the migration time migration and performance due to geographical limitations.

Acknowledgement: Thanks to our colleagues, who provided moral and technical support.

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 this study.

References

- [1] N. Tabassum, A. Ditta, T. Alyas, S. Abbas and M. A. Khan, "Prediction of cloud ranking in a hyperconverged cloud ecosystem using machine learning," *Computer Materials Continua*, vol. 67, no. 3, pp. 3129–3141, 2021.
- [2] B. Rad, B. Bhatti and H. Ahmadi, "An introduction to docker and analysis of its performance," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 17, no. 3, pp. 228, 2017.
- [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. Kozłowski, "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] A. Iancu, "A mamdani type fuzzy logic controller," *Fuzzy Logic-Controls, Concepts, Theories and Applications*, vol. 15, no. 2, pp. 325–350, 2012.
- [23] 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.
- [24] P. Singh, A. Kaur, P. Gupta and K. Jyoti, "RHAS: Robust hybrid auto-scaling for web applications in cloud computing," *Cluster Computing*, vol. 24, no. 2, pp. 717–737, 2021.
- [25] L. Gregory, X. Liudong and X. Yanping, "Optimal early warning defense of N-version programming service against co-resident attacks in cloud system," *Reliability Engineering & System Safety*, vol. 106969, no. 201, pp. 85–92, 2020.
- [26] H. Huang, G. Levitin and L. Xing, "Security of separated data in cloud systems with competing attack detection and data theft processes," *International Journal of Risk Analysis*, vol. 39, no. 4, pp. 846–858, 2019.
- [27] D. M. Shah, M. V. Murthy and A. Kumar, "A new approximation algorithm for virtual machine placement with resource constraints in multi cloud environment," in *2020 Third Int. Conf. on Advances in Electronics, Computers and Communications (ICAIECC)*, Bengaluru, India, no. 5, pp. 1–5, 2020.