Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.033916

Article

Optimizing Resource Allocation Framework for Multi-Cloud Environment

Tahir Alyas[1], Taher M. Ghazal[2,3], Badria Sulaiman Alfurhood[4], Ghassan F. Issa[2], Osama Ali Thawabeh[5] and Qaiser Abbas[6,*]

[1]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
[2]School of Information Technology, Skyline University College, University City Sharjah, 1797, Sharjah, UAE
[3]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebansaan Malaysia (UKM), 43600, Bangi, Selangor, Malaysia
[4]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, 11564, Saudi Arabia
[5]Skyline University College, University City Sharjah, 1797, Sharjah, UAE
[6]Faculty of Computer and Information Systems Islamic University of Madinah, Madinah, 42351, Saudi Arabia
*Corresponding Author: Qaiser Abbas. Email: qabbas@iu.edu.sa

Received: 01 July 2022; Accepted: 28 October 2022

Abstract: Cloud computing makes dynamic resource provisioning more accessible. Monitoring a functioning service is crucial, and changes are made when particular criteria are surpassed. This research explores the decentralized multi-cloud environment for allocating resources and ensuring the Quality of Service (QoS), estimating the required resources, and modifying allotted resources depending on workload and parallelism due to resources. Resource allocation is a complex challenge due to the versatile service providers and resource providers. The engagement of different service and resource providers needs a cooperation strategy for a sustainable quality of service. The objective of a coherent and rational resource allocation is to attain the quality of service. It also includes identifying critical parameters to develop a resource allocation mechanism. A framework is proposed based on the specified parameters to formulate a resource allocation process in a decentralized multi-cloud environment. The three main parameters of the proposed framework are data accessibility, optimization, and collaboration. Using an optimization technique, these three segments are further divided into subsets for resource allocation and long-term service quality. The CloudSim simulator has been used to validate the suggested framework. Several experiments have been conducted to find the best configurations suited for enhancing collaboration and resource allocation to achieve sustained QoS. The results support the suggested structure for a decentralized multi-cloud environment and the parameters that have been determined.

Keywords: Multi-cloud; query optimization; cloud resources allocation; modelling; virtualization

1 Introduction

Cloud computing is the global platform for highly distributed online processing, communication, and data management functionality. Cloud computing provides operational facilities online; therefore, the internet is the medium to access cloud computing platforms. The organizations managing the cloud platforms are called Cloud Service Providers (CSPs). CSPs provide services ranging from software applications and middleware to highly sophisticated infrastructure to the end-users [1].

Nowadays, many users use cloud computing, so many users are taking advantage of its more and more services and applications wherever they are. Cloud computing is also a backup system, so any files with important information can have backup files on the cloud network. Cloud computing can share files anywhere the person is worldwide; users can share files quickly and safely without any hesitation or failure. Cloud computing also has the backup of application software so that we can entrust it to the failure of any software because we have its backup files [2]. Cloud computing can be both public and private. The public cloud is open for all users, and public cloud service providers provide it free on the internet. Private Cloud Providers give access to a private cloud to a specific number of individuals [3]. These services are a system of networks that supply hosted services.

With the rise of cloud computing, big data, and artificial intelligence, cloud resource demands are expanding quickly, particularly certain unclear and emergent resource demands. The emergent mode cannot be supported by conventional cloud resource allocation techniques for timely and optimal resource allocation [4].

In multi-cloud environment services, one of the critical challenges in security is authorization and authentication. In authorization scenarios, identify the user as having the authority to see this part of the application service, and authentication scenarios determine whether the cloud user is authentic. Different applications have various life cycles and policies to utilize authentication and authorization techniques. The most security challenges ensured that correct and authentic users could access your cloud-based application [5].

In a multi-cloud environment, monitoring is a critical challenge for security and management. When customer data is stored in the cloud, monitoring is the technique that ensures customer data is safe. Cloud-based services offer many benefits for business usage, and web Application Programming Interface (API) increases the security risk of deploying cloud-based services. Thus, most organizations are nervous about storing private data in the cloud. Monitoring enables organizations to balance risks, take advantage of the cloud, and reduce critical challenges' complexity [6].

In a multi-cloud environment, handling storage is always difficult. When designing a cloud-based system, challenges must be in mind that security is essential, which allows for storage with multiple clouds. Most users store their data in the cloud instead of saving it on local devices or hard drives. Data centres take care of the user's data and ensure it is safe [7].

System hardening is the technique used to reduce security risks and eliminate security attacks and the systems attack surface. Application hardening is the type of system hardening. Application hardening provides the facility to restrict the user's access to the application based on role management. In application hardening, remove any useless function. Application passwords are also easily managed through the tasks of updating and resetting. Most organizations use the best techniques to misuse the app's Internet Protocol (IP) cheating and prevent attackers. Application hardening methods include code obfuscation, anti-debugging, binary packaging, and white-box cryptography [8].

Multi-cloud computing is primarily related to profitability. Institutions with different silicon stations (e.g., development, testing, production, and support) are recommended to be used in separate cloud applications to avoid downtime. A multi-Cloud approach allows organizations to clearly, cover the risks of service disruption. There is a need to increase confidence in the association's dissemination plans. Recently, companies have developed many native cloud applications, the main driving force behind cloud evaluation, and choose to host them on multiple clouds [9].

QoS is the major component of cloud applications. The resources should also be relevant to the Service Level Agreement (SLA) defined by the end-user and the cloud providers. The proper utilization of resources concerning end-users needs increases the profit of cloud service providers. QoS management consists of a resource allocation problem with services such as cloud application performance, availability, and reliability. The cloud application QoS management method automates cloud computing's hardware and software resource allocation programming [10].

In cloud computing, resource allocation strategies are guaranteed virtual machines or physical resource allocation to the cloud user with minimal resources. Other parameters play an essential role when SLA between the service provider and cloud user, as shown in Fig. 1.

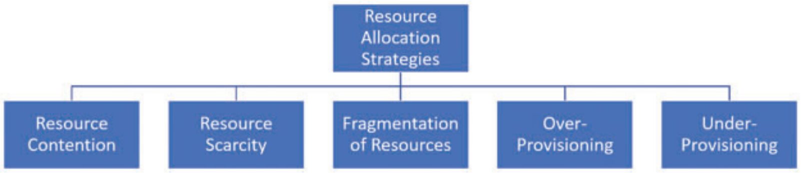


Figure 1: Resource allocation strategy

Resource contention occurs when two applications hosted on a server utilize the same resources simultaneously. Similarly, resource scarcity is related to the shortage of resources that arises when limited resources are available. As mentioned in the above diagram, resource fragmentation is another critical parameter that occurs when the resources available are limited but not intelligent enough to allocate the resources allocation of the desired application. Over and under-provisioning are the last two parameters; over-provisioning arises when the resources task is free of the actual demand for resources with need. Under-provisioning occurs when the user task is allocated with few quantities of resources than the actual need [11].

Cloud computing offers dynamically provided resources displayed as one or more integrated computer resources based on limits. The application provisioner must determine the appropriate software and hardware configuration throughout the cloud provisioning process to guarantee that the Quality of Services (QoS) aim of the application services is met without sacrificing system utilization and efficiency [12].

2 Related Work

Identifying the core players in a multi-cloud environment for resource allocation and management [13]. In cloud computing, one cloud service provider with a certain number of services to the end-user is a successful centralized setup that provides good performance. The resource sharing in a multi-cloud environment is configurable in a centralized or decentralized manner. Multi-cloud is split into different types based on these configurations and becomes part of the gradually developed multi-cloud architecture.

The multi-cloud platform has emerged in search of optimum benefits. It can be classified into two broad possible configurations: centralized and decentralized. A multi-cloud can be configured in a centralized manner that can have all the resources controlled by a central entity. The centralization certainly provides more control and requires tangible investment in developing the central infrastructure to serve all the cloud users and their respective requests. In the decentralized configuration of the multi-cloud, the control is distributed among multiple entities within the multi-cloud based on the capacity and availability of resources with different stakeholders who can manage their resources according to their localized requirements and pool the resources along with the administration services. Based on the strategy and policy development in a decentralized configuration, participants will either have a single policy regarding services and price slab, or participants will have their business transaction structure, as this configuration is not making limits in any manner [14].

In practice, end-users also map the required services among all participants and analyze the latency based on location proximity to get optimum results. Any participant with a better latency rate and performance is responsible for service provisioning [15]. The concept of virtualization is very much applicable to formulating a virtual cloud framework that can provide compute, storage, and network resources to the end-user to activate services from the constituent member clouds in a virtual execution environment. The most workable and sustainable multi-cloud architecture is the combination of multiple SDC (small data centres) representing the localized resources with every participant. The scalability of such an arrangement is also possible in an encapsulated environment without disturbing the end-user [16]. The perspective and needs of the end-user decide which centralized or decentralized configuration is more suitable and productive for the end-user [17].

The decentralized cloud combines the advantages of cloud computing and Peer-to-Peer (P2P) computing paradigms [18]. An immediate benefit of decentralization is improved privacy and security since there is no central control over the data. By participating in the decentralized cloud, small cloud players will get a channel to form business groups and support each other. Cloud end-users will gain more confidence in using this model because of its new features.

The participation of SDCs is more stable in decentralized clouds than the continuously joining and leaving of peers in the P2P system [19]. However, there are three key differences between the decentralized cloud and the P2P model. First, the peers in the decentralized cloud are mainly small data centres that are legal and economic social entities rather than individuals. Thus, cloud providers' incentives to cooperate are not merely for revenue maximization but also for other social dynamics such as legal constraints, location proximity, trust, and loyalty. Second, compared to a large population of peers, the number of cloud providers is relatively small. Third, the computing environment is not as highly dynamic as in the P2P model [20].

It is worth mentioning that the decentralized cloud will not replace the centralized cloud, but they complement each other. Centralized clouds have already demonstrated their power in web hosting and compute-intensive applications. However, the decentralized cloud is the best choice if applications need high information privacy or better control over the data [21].

3 Proposed Methodology

The proposed framework focuses on three segments in multi-cloud for resource allocation and management. The previous sections show in detail that the multi-cloud environment is expanding in the corporate sector and at the CSP level to provide specialized services relevant to the multi-cloud environment. The three-factor dealing with this process is the service provider mainly providing services in any specific configuration, i.e., private, public, hybrid, or community cloud. At the same time, the second actor is the resource provider with the physical structure and extends the facility to other resource providers. The third important factor in this equation is the end-user working in a multi-cloud environment. The resource provider has an SDC or some computing, storage, and network at the operational level. This capacity and resources are provided to the end-user through virtual machines, dockers, containers, and other means that the cloud service providers own. The services are provided on the resources following certain pricing and QoS strategy framework in different cases.

There are nine-step for QoS parameters for resource allocation in a cloud environment. In Step 1, the client sends a request to the server through a query in a virtual environment. App configuration and testing are done with QoS goals in a virtual environment. In Step 2, the required request is sent to cluster management for clustering services in multi-cloud environments through different configurations with QoS parameters, as shown in Fig. 2. QoS parameters include threshold configuration, test logic controllers, and a stepping plan. In Step 4, Step 5, and step 6, the test state model and model analysis are analyzed on different configurations and checked to see whether the QoS is satisfied. In step 7, if the QoS parameter is satisfied, resource provision from the resource pool is performed. In Step 8, the monitoring and business logic tiers are performed and communicated to the virtual environment. Finally, all the results and costs are transmitted to the client. The proposed framework is focused on three main parameters for the QoS and resource allocation in a multi-cloud environment, as shown in Table 1. These parameters are extracted from the literature and different cloud models, specifically the variants of multi-cloud environments. The structure of the proposed framework is based on these three parameters to extend the other processes and logic for the dynamic and swift resource allocation with improving the QoS in the multi-cloud platform.

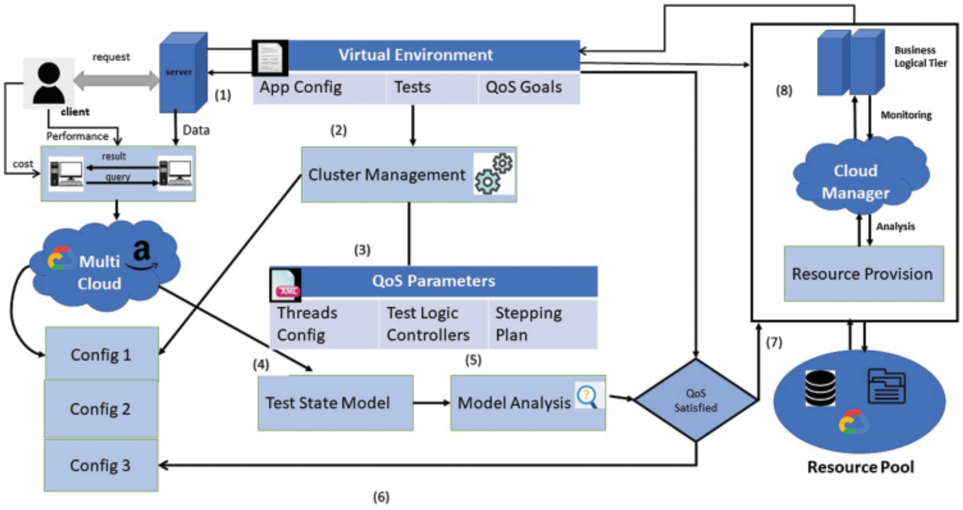


Figure 2: Workflow of proposed framework

It is important to note that the descriptive framework has three core parameters essential in developing any type of multi-cloud. The first parameter, cooperation, is the core element in developing the basis for optimization and sharing. As discussed, multi-cloud is the amalgamation of different services from various cloud service providers, the high level of heterogeneity from structural to service level makes this parameter fundamental and crucial. The subset of this parameter includes the parameters like incentives and strategies. Further detailing of these parameters has provided the variables related to financials, social factors, and SLA performance. Moreover, these parameters are vital in allocating resources with an evident QoS.

Table 1: Parameter for QoS

Multi-cloud	Cooperation	Incentives	Financial
			SLA performance
		Strategies	Workload base
			SLA base
	Optimization	Objective	Revenue
			Improve QoS
		Model	Constraint base
			Game theory
		Evaluation	Process cost
			SLA base
	Data sharing	Inter cloud	Storage
			Optimization
		Intra cloud	Duplication
			Summarisation

The proposed framework depicts that cooperation is not attainable without incentives and strategy. A good strategy may lead to minimal incentives and maximum outcomes. However, the incentives are not necessarily fiscal. These incentives may address social factors or may lead toward performance-based SLA. Therefore, resource providers need cooperation based on incentives and a carefully designed strategy. The resource allocation strategies are linked with incentives, especially in a multi-cloud scenario where resource providers have their assets and want to maximize their incentives, as shown in Fig. 3.

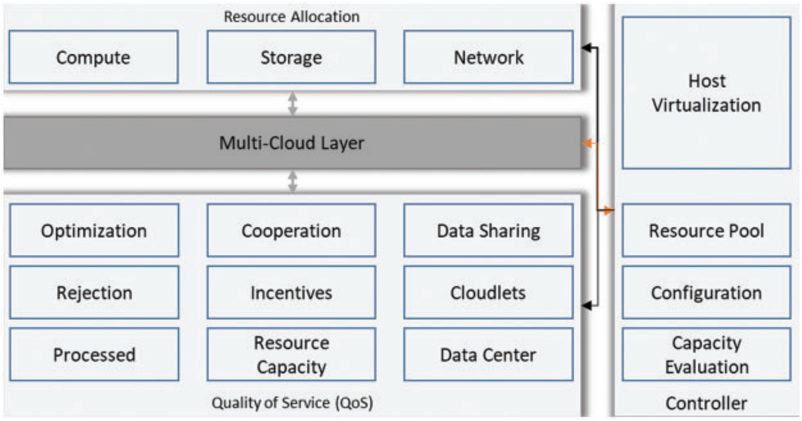


Figure 3: Parameters for resource allocation in multi-cloud

The development of this cooperation is challenging due to the heterogenetic nature of multi-cloud. The incentives can be different with different goals for a service provider. A stable bandwidth and lower latency can be a goal linked with incentives. Another factor under consideration is the relation between optimization and efficiency of cooperation that leads to the attainment of service and performance goals by the resource provider and service provider. It is, therefore, proposed as part of the framework to have a dynamic perspective following the use of this framework.

There are two explicit scenarios for the cooperation to allocate resources in a multi-cloud environment. The first scenario is in the presence of a central entity, while the second scenario is the decentralized multi-cloud configuration. The first scenario is simple regarding resource allocation as all the participants are providing services and resources to the central entity to accomplish specific tasks–the accomplishment of tasks is the primary concern of all the participants so they can move to the next assignment. The central entity has performed the main coordination, and none of the resource providers seek individual gains. Therefore, incentives become performance recognition, ranking, and a competitive environment always exists in the centralized multi-cloud environment. In other words, cooperation is simple and easy to establish in a centralized multi-cloud without having an in-depth strategy to elaborate the conditions of cooperation and incentive levels, as shown in Fig. 3.

As no central organisation supervises and organizes the cooperation among all stakeholders, resource allocation collaboration is very complicated in a decentralized multi-cloud setting. Therefore, the key issue is whether it is possible to create a cooperative environment for resource suppliers. In a decentralized multi-cloud environment, examining the potential scenarios and variables on which resource providers would base their cooperative resource allocation decisions is crucial. Due to the two different types of incentives and tactics used to categorize the processes, there are two subsets of cooperation.

The multi-cloud platform is an environment that provides all the opportunities to test technology and business theories. Cloud resource allocation with the first layer of cooperation is the primary parameter for successful multi-cloud environments. Keeping the interest or incentive of the resource provider intact means a longer and more stable service arrangement. Once the cooperation is attained using incentives and strategy with all of the subsets, the next logical step is optimization. The pre-requisite of optimization to be workable is the complete set of information related to resources and the multi-cloud structure. The way to use optimization with or without full information is to keep optimization as a centralized process (with comprehensive information related to all resource providers, resources and cloud types, and configurations). In case of an insufficient information set, optimization can be a localized process at each resource provider. However, it will only be effective at the local level and will not deal with multi-cloud as a platform. We have divided the optimization into the proposed model's objective function, constraints, model, and evaluation criteria.

After developing the objective function and its parameters, the model for the optimization and resource allocation of the multi-cloud platform is based on two main parameters, i.e., models linked with constraints or a rule base and the other parameter is, and the game theory parameters, i.e., benefits analysis for a win-win situation.

The capacity of the resources shall be between the total resources and total allocated resources, e.g., compute, storage and bandwidth. All the resources shall work within the allocated budget in terms of capacity, time, and availability. All resources allocated/available shall perform tasks following the schedule; delayed tasks can be skipped and re-scheduled. In multi-cloud, the reliability factor is part of the rule base; the failure instances in terms of accessibility, availability, replication, etc., directly impact reliability. The rule base follows the legal and regulatory requirements of the geological location where the cloud is being operated. This concept is suitable for all multi-cloud types in centralized and decentralized configurations. Many models tried to formulate resource allocation mechanisms using game theory. Mostly these models take price and QoS as key elements to develop respective modelling.

For the proposed framework, the evaluation criteria to analyze the optimization techniques and service allocation are based on the following points:

- The computation value of the model is an important segment to be considered.
- The process cost includes the total time required to perform specific tasks and the delays, the average expected time, and the actual time-consuming.
- Financial gains are also an important parameter to depict the performance of any resource allocation model. The costing factor is linked with computing, bandwidth and storage, and multi-cloud configuration adjustments like centralized or decentralized.
- Network capacity in terms of throughput and bandwidth is a standard parameter for evaluating the quality of service.
- Network capacity is linked with using that capacity in real-time by cloud services. The difference is depicting the latency ratio and performance.
- Data availability in replication over a spread of geological locations to reduce the processing cost.
- The workload balancing by managing the received requests and performing requests at various data centers.
- SLA are focused on evaluating the performance and optimization parameters; the instances to breach the SLA guidelines are the key parameters to be considered to define the quality of service.

The purpose is to schedule the data sharing and movement in a disciplined manner using different optimization tools, while the cloud service providers also provide few. Intercloud scheduling has a positive impact on delay management and bulk data transfer. In this method, the link capacity between two nodes is calculated with multiple protocols to increase the throughput and minimize the transaction time; it also considers the compute power and storage speed as it affects the bulk transfer. It is supported by an optimization algorithm to manage the transactions during the transfer and use other protocols to get maximum throughput and QoS [22].

This method helps optimize data at rest, i.e., before transferring data, it is better to maximize the data to ensure a swift and cost-effective transfer. The data is transformed into smaller chunks; the information lossless method uses compression, keeping all the metadata information in the original dataset. Another process that supports the optimization is called deduplication which ensures the data transfer once, and resources shall not be used again for the same data transfer. Therefore, the metadata contains the information related to all replications tagged as “transferred” to avoid wastage of resources. The optimization is also linked with protocol optimization; CloudOpt is a proven platform that provides multiple services for data compression, deduplication, optimization, throughput adjustment, and protocol optimization [23].

4 Results and Discussion

The proposed framework has been evaluated using CloudSim simulator, a toolkit for cloud computing simulation with various perspectives, as discussed in detail about the different parameters for the QoS and optimization and other relevant aspects. In the simulation, we have formulated the same in three main variables, i.e., a decentralized multi-cloud, a broker, and a load manager. A decentralized cloud aims to simulate SDCs, virtual machines, and sharing within the SDCs. The broker’s role is to submit load to virtual machines and request resource allocation to demand quality of service. The load manager depicts the requests and accomplishments as per the end-user [24].

Multiple cloud service providers charge users for different services, so hiring cloud services for experimentation is challenging. Therefore, the research, implementation, and development related to the cloud are mainly performed using cloud simulators. For validation purposes, in this research, Cloud Sim has been used to simulate large-scale cloud computing with a virtual server facility and customized virtual machine creation. CloudSim [25] is also used to evaluate models with simulated workloads and data centres. We have used the following setup for the simulation.

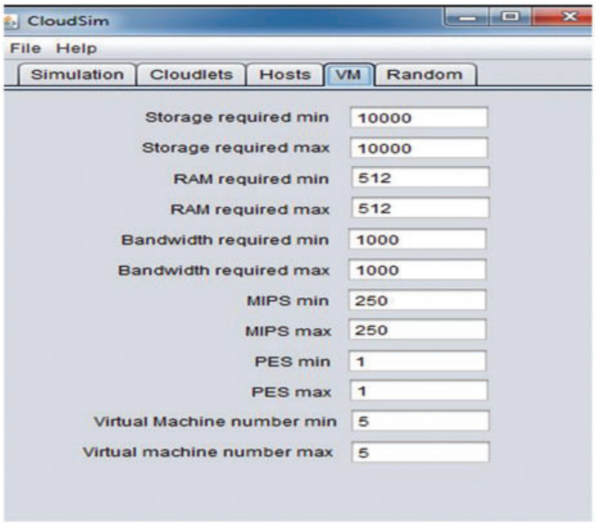
CloudSim does not have a graphical user interface (GUI); instead, it works with different integrated development environments (IDE)s like Eclipse and NetBeans. This simulation uses Eclipse with CloudSim. The configuration of the CloudSim is based on the cloudlets, hosts, virtual machines, and randomization to simulate various configurations at different virtual machines, as shown in Table 2. Google Cloud Services (GCP) is used as the host platform for the simulation, while Microsoft Azure is used for comparative validation.

Table 2: Configuration set up

Tools	Description
Apache Tomcat 9.0.48	Webserver for hosting DSpace
DSpace 5.10	Database Simulation host
JDK8	Development environment
PstgreSQL 9.6.22	Database Simulation
Dataset	KDD2020

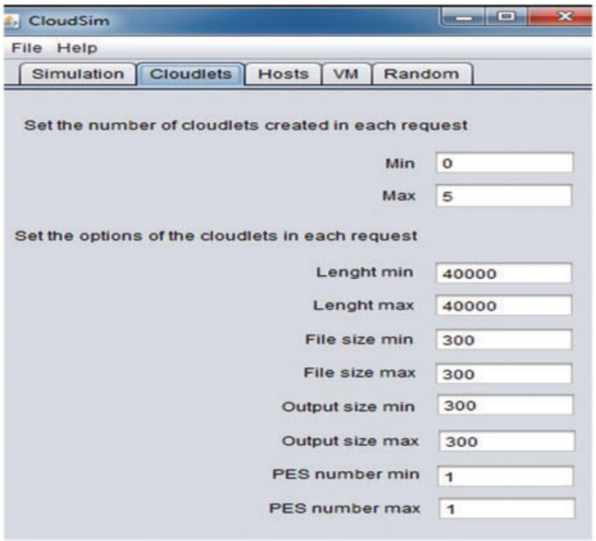
This simulation evaluates various configurations in multiple cloud setups to ensure the validity of the parameters mentioned in the proposed framework, i.e., optimization, cooperation, and sharing. All three parameters represent the quality of service. Virtual machines are randomized with five devices, as shown in the configuration screenshot. The simulation is activated with minimal configuration instead of abundant resources to understand the workload conditions and failures. The virtual machine has 1 GB of storage. Instead of using a physical dataset, DSpace is used to simulate the datasets ranging from 1000 to 10000 with concurrent query options to evaluate the capacity management of the proposed framework.

As shown in Fig. 4, the virtual machine configuration explains the storage at 10000 with ram at 512 while the rate of millions of instructions per second (MIPS) is fixed at 250. Maximum status and the number of virtual machines at five on both minimum and maximum levels; randomization uses these five machines to generate multiple episodic transactions for computing, network, and storage. At the same time, the cloudlets are configured to provide multiple simulated data centres for the virtual machines to simulate multiple resources and multiple instances of storage, as shown in Fig. 5. A maximum of 05 cloudlets are being used, which means each virtual machine will simulate 05 different cloudlets. Overall, 25 instances are simulated to evaluate the proposed framework using compute, storage, and network, as shown in Fig. 5.



Storage required min 10000
Storage required max 10000
RAM required min 512
RAM required max 512
Bandwidth required min 1000
Bandwidth required max 1000
MIPS min 250
MIPS max 250
PES min 1
PES max 1
Virtual Machine number min 5
Virtual machine number max 5

Figure 4: Virtual machine configuration

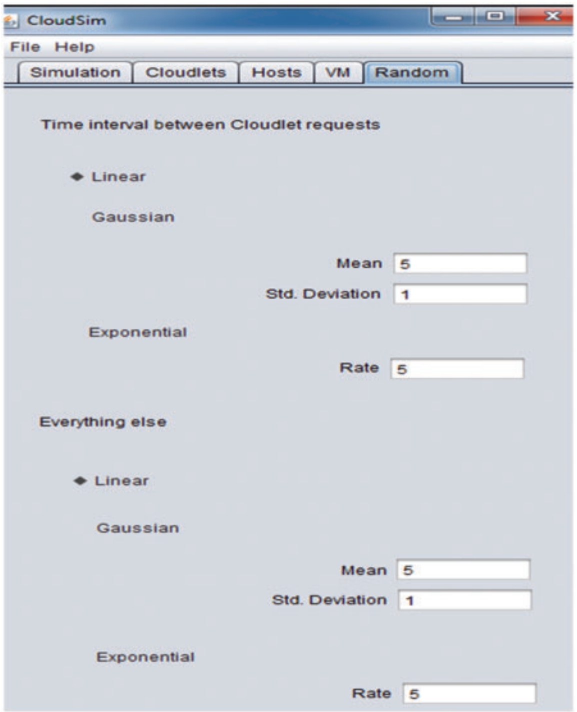


Set the number of cloudlets created in each request
Min 0
Max 5

Set the options of the cloudlets in each request
Lenght min 40000
Lenght max 40000
File size min 300
File size max 300
Output size min 300
Output size max 300
PES number min 1
PES number max 1

Figure 5: Cloudlets configuration

The allocation of resources with a versatile availability of cloud service providers plays a significant role, as mentioned in the problem statement and proposed framework in multi-cloud. Therefore, the configuration of virtual machines is defined with an average capacity but with processing challenge in terms of MIPS at 250 with minimum and maximum 05 virtual machines in one instance as shown in Fig. 6. The compute, network and storage are allocated in this unit. Further configuration of CloudSim will show the complex scenario for validating the proposed framework. The first experiment is based on the general resources, i.e., compute, storage, and network on SDC and workload in a certain peak hour, as shown in Fig. 7. Notably, the whole simulation is performed in a decentralized multi-cloud environment. The initial validation cycle is simulated without the proposed framework and the parameters we have identified in the framework. The result shows a decline in the resource capacity with the increasing load and time interval. The SDC and relevant resources were unable to optimize the resource to enhance the quality of service. Simulation Configuration, as shown in Fig. 8, shows the actual run with repeated simulation run is 30 with initial seed is 42.



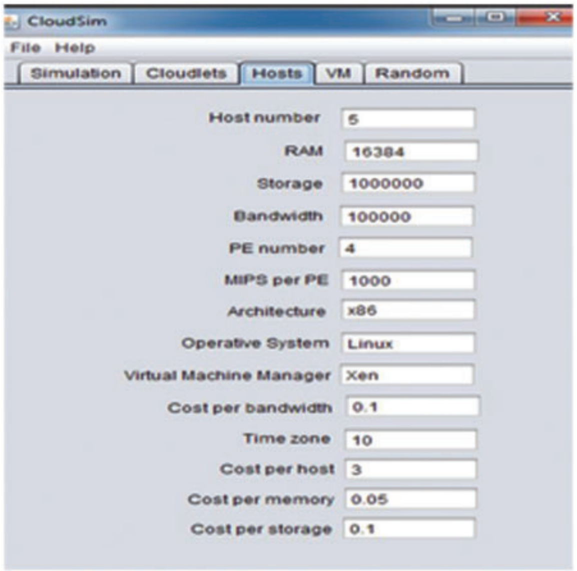
Time interval between Cloudlet requests

- ◆ Linear
- Gaussian
  - Mean: 5
  - Std. Deviation: 1
- Exponential
  - Rate: 5

Everything else

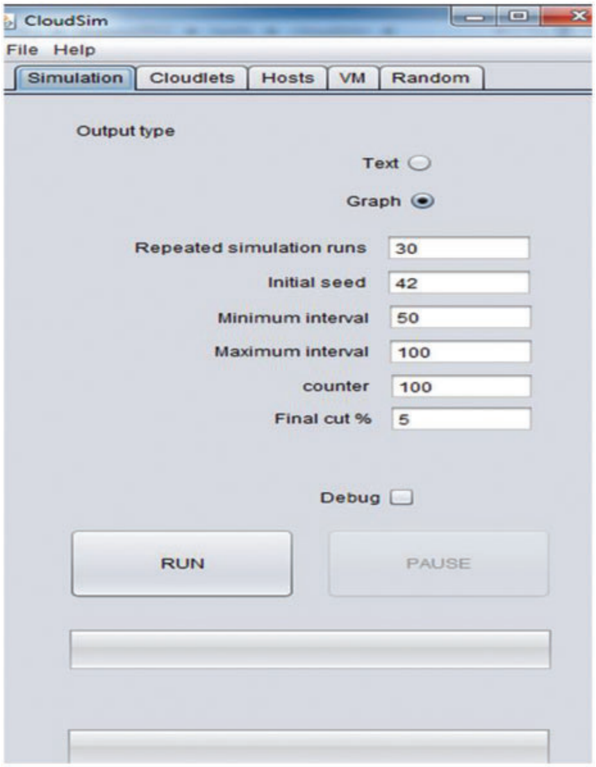
- ◆ Linear
- Gaussian
  - Mean: 5
  - Std. Deviation: 1
- Exponential
  - Rate: 5

Figure 6: Randomization configuration



Host number	5
RAM	16384
Storage	1000000
Bandwidth	100000
PE number	4
MIPS per PE	1000
Architecture	x86
Operative System	Linux
Virtual Machine Manager	Xen
Cost per bandwidth	0.1
Time zone	10
Cost per host	3
Cost per memory	0.05
Cost per storage	0.1

Figure 7: Host configuration



Output type

Text ○

Graph ●

Repeated simulation runs 30

Initial seed 42

Minimum interval 50

Maximum interval 100

counter 100

Final cut % 5

Debug □

Figure 8: Simulation configuration

The continuity of the simulation provides another spike in the QoS that reaches a maximum value and starts declining again, as shown in Fig. 9. Therefore, the decentralized multi-cloud with standard configuration fluctuates on quality of service and resource capacity. As the capacity starts exhausting, the quality of service starts declining, and there is no impact visible of any built-in optimization protocol. The simulator uses the parameters and values to establish a decentralized multi-cloud based on various proven scenarios.

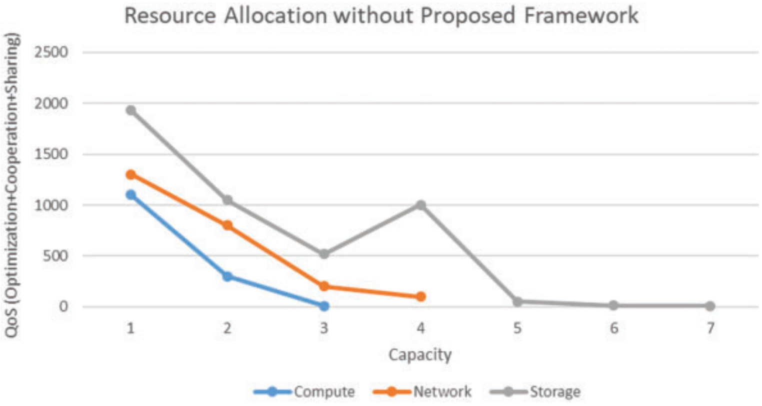


Figure 9: Resource allocation without optimization

The second simulation cycle focuses on the same experiment but incorporates the proposed framework and the respective parameters discussed in the methodology section. The simulation again provides the same three modules, i.e., decentralized multi-cloud, broker, and load manager, with the same parameters and SDCs setup as shown in Fig. 10. The same load is simulated with the identical timeframe. The results are as in the graph. A sharp increase is visible on all counts, i.e., compute, storage and network. The proposed framework uses the scenario to enhance service quality by optimizing allocated resources in the given timeframe. The cooperation, optimization, and data availability values are taken as variables to fluctuate between 1 to 10. The simulation results are very significant to support the objective and research questions of this framework, i.e., as the cooperation increases, the quality of service is also increasing; similarly, the mediating factors of cloud optimization instead of local optimization are performed, which is also showing a positive impact. Finally, the data is available with strategic replication to minimize resource wastage and latency, as shown in Fig. 11.

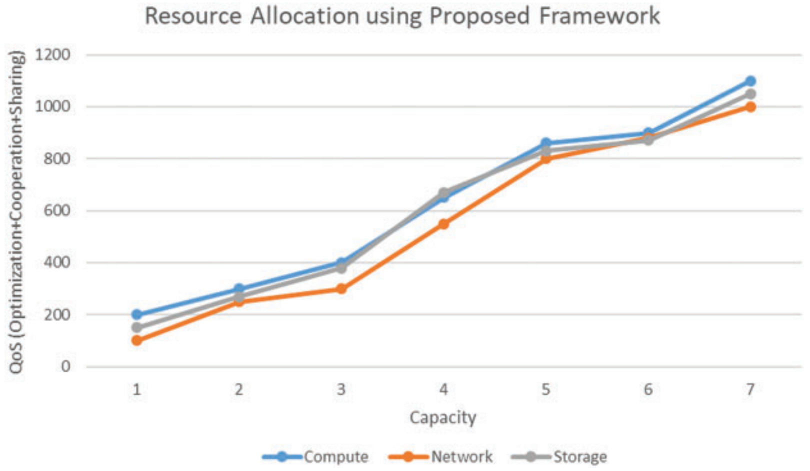


Figure 10: Resource allocation with optimization using the proposed framework

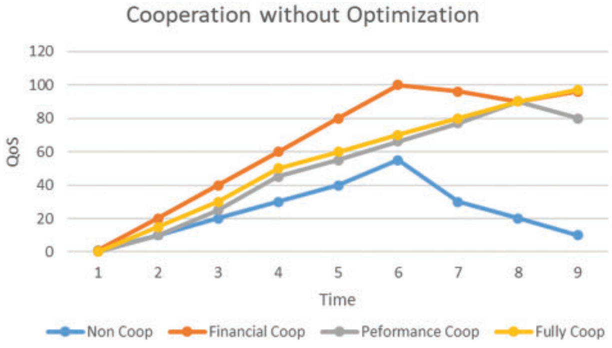


Figure 11: Cooperation impact analysis

The role of SDC and the loss of request is also observed. The results show that the loss of requests decreases after optimization while using the same without the proposed framework also reduces the loss of requests. Still, the impact increases by 67% with incorporating the proposed framework. It is also important to note that the simulation is set up with peak hours and a heavy workload. Therefore, it is safe to say that the proposed framework and the parameters identified in this framework are valid and workable in achieving the research objectives and solving the problem statement.

The impact of optimization on cooperation is also analyzed with parameters consisting of non-cooperative, financial collaboration, performance cooperation, and fully cooperated. As it is essential to evaluate whether the impact of optimization seen in the previous experiment on resource allocation impacts increasing cooperation, the non-cooperation is a variable here due to the lack of strategy, awareness, and communication that isolates the resource providers.

The results are shown in Fig. 12. The non-cooperation increased over time, but as soon as the maturity level was achieved, it started decreasing. The exciting result is that cooperation due to financial incentives spikes while the other parameters, i.e., performance-based cooperation and full cooperation, are almost identical. This also endorsed the previous section's research that financial incentives are the most common reason to engage resource providers for cooperation.

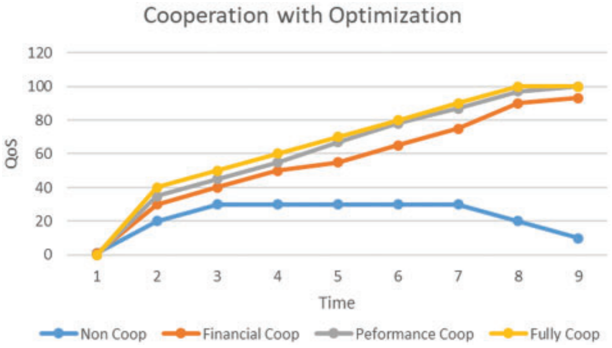


Figure 12: Cooperation impact analysis with optimization

It is important to analyze that an optimized decentralized multi-cloud is a more suitable and enhanced platform for resource providers in terms of financial gains, sharing, and performance improvement [26]. The same experiment is simulated again with the inclusion of identified parameters and a proposed framework to analyze the impact of optimization on the cooperation of resource providers. After the inclusion of the proposed framework and specified parameters, the results show a significant improvement on all counts, i.e., the non-cooperation is reducing. At the same time, the cooperation based on financial incentives is now more realistic, and performance-based or full is spiking high over the stipulated timeframe [27].

Further simulation is performed on query optimization with and without using the proposed framework. In query optimization, three elements are taken as core parameters, i.e., rejection of the queries based on incomplete, inconsistent, and redundant results. Processed are the queries which are successful in execution and provisioning of results, and finally, the same conditions are applied in simulation after optimization proposed in the framework. Results shown in Fig. 13 show the rejection significantly decreases with the increase in optimization and finally reaches a minimal value. Another critical point is the initial good performance of processed without optimization. Still, it is also visible that the process without optimization is performing well until reaching the record range of 1500. While the optimized range is low when datasets have fewer numbers, optimization improves with the increase in the data volume.

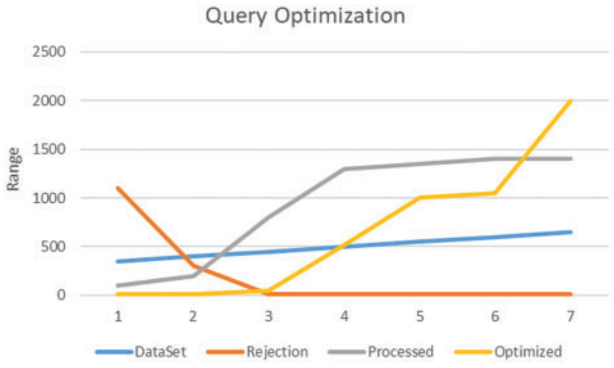


Figure 13: Query optimization-1

Results are promising in compute utilization with and without using the proposed framework. As the framework targets the cloud platform, it is desirable to validate the framework for the compute evaluation. In the simulation, the same parameters and range have been used. The compute-1 represents the simulation without query optimization, as shown in Fig. 14. As it is visible, compute starts increasing with the increment in the data range. Up to 600 cycles, the computed increase is slow, but after crossing the 600 range, there is a sharp increase in the compute, flattened out at the range of 1600.

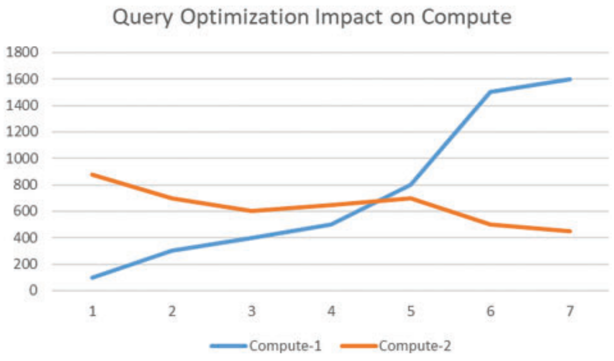


Figure 14: Query optimization-2

On the other hand, the higher value at the start of compute-2 represents the simulation after optimization. With the increment, the results show a stable decline in the compute requirement. The results flatten out and get stable at the range of 400.

The network is next critical element to be examined in continuing this simulation process. As it is visible in the attached graph that Network-1 (without optimization) is increasing gradually up to 600. On the other hand, Network-2 (optimized) has fluctuations initially, but it contrasts with the without optimized values. The results are getting stabilized slowly and ultimately reaching a stable compute, as shown in Fig. 15.

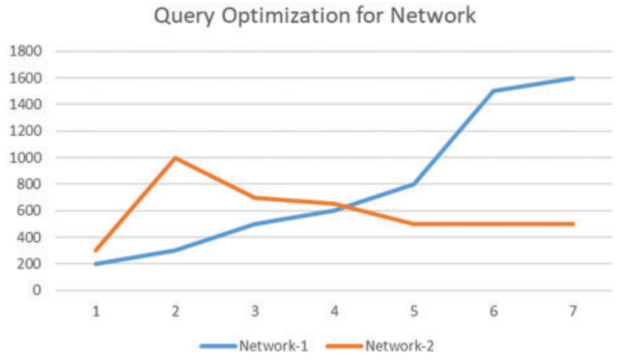


Figure 15: Query Optimization-3

The summary graph explains the complete simulation, i.e., there are three groups initially, i.e., Rejection, Processing, and Optimized. The second group contains compute-1 and compute-2, while the third group is about the networks in Fig. 16. The red lines show the optimization results, while the grey lines show the non-optimized readings. Compute and network both show signification results, i.e., minimizing the compute and network after optimization. The optimization is taking a short hike as the volume increases.

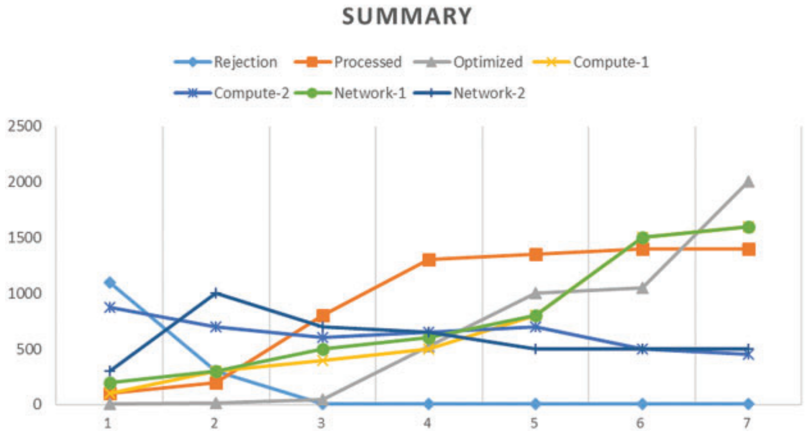


Figure 16: Resource allocation & performance

The proposed framework is used in simulation to identify the capacity and functionality of the same. Notably, the simulator used in this framework now requires real datasets. Instead, it develops the storage environment in space and uses the database to replicate virtually and provide accurate results without using any fixed datasets. The simulator configuration provides a clear picture of the setup with virtualization and capacity evaluation. The simulation outcome is significant and shows the stability and sustainability ranges for the proposed framework.

5 Conclusion

Scheduling and resource distribution are the main determinants of service quality in cloud computing. Since allocating resources inside a single cloud environment is complicated, cloud computing calls for an effective resource allocation module. At the same time, resource allocation in a multi-cloud scenario further increases the complexity of the allocation processes. Until now, task requirements were used to determine how to assign resources from the multi-cloud environment. The optimization process for resource allocation is based on optimal resource capacity utilization. However, the main parameters have been used without the subsets, i.e., compute, network, and storage for the simulation. The outcome of the simulation exposes the role of various variable and factors that plays a significant role in the quality of service. The cooperation amongst resource providers to create a functioning model that serves as the source of optimal quality of service is based on the very nature of decentralised multi-cloud, which is the second component of the optimization process. This simulation's outcome has also provided significant results on cooperation parameters and its impact on the quality of service. The experimental findings demonstrate that our algorithm can balance the consumption of all types of resources while swiftly and optimally allocating resources for unexpected demands.

Acknowledgement: Thank you to our coworkers for their moral 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," *Computers, 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, no. 4, 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, no. 3, 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, 2019.

[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, no. August (12), 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, no. 12, pp. 1043–1055, 2021.

[14] Z. Li, S. Guo, L. Yu and V. Chang, "Evidence-efficient affinity propagation scheme for virtual machine placement in the 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 the 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] L. Heilig, E. Lalla-Ruiz and S. Voß, "Modeling and solving cloud service purchasing in multi-cloud environments," *Expert System Application*, vol. 147, no. 3, pp. 113165, 2020.

[19] M. Alaluna, E. Vial, N. Neves and F. M. V. Ramos, "Secure multi-cloud network virtualization," *Computer Networks*, vol. 161, no. 4, pp. 45–60, 2019.

[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] B. Liu, X. Chang, Z. Han, K. Trivedi and R. J. Rodríguez, "Model-based sensitivity analysis of IaaS cloud availability," *Future Generation Computer System*, vol. 83, no. 7, pp. 1–13, 2018.

[22] 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.

[23] 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.

[24] 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.

[25] N. Tabassum, T. Alyas, M. Hamid, M. Saleem and S. Malik, "Hyper-convergence storage framework for ecocloud correlates," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 1573–1584, 2022.

[26] T. Alyas, K. Alissa, M. Alqahtani, T. Faiz, H. H. Naqvi *et al.,* "Multi-cloud integration security framework using honeypots," *Mobile Information Systems*, vol. 2022, pp. 1–13, 2022.

[27] S. Malik, N. Tabassum, M. Saleem, M. Hamid, U. Farooq *et al.,* "Cloud-IoT integration: Cloud service framework for m2m communication," *Intelligent Automation & Soft Computing*, vol. 31, no. 1, pp. 471–480, 2022.