International Journal of Business Quantitative Economics and Applied Management Research

Volume-7, Issue-7, 2023 ISSN No: 2349-5677

PUBLIC CLOUD IN MULTI-CLOUD STRATEGIES INTEGRATION AND MANAGEMENT

Premkumar Ganesan
Technology Leader in Digital Transformation for Government and Public Sector
Baltimore, USA

Abstract

The terminology surrounding cloud computing—such as "public cloud," "private cloud," and "hybrid cloud"—is often used interchangeably, though their definitions remain contested. Ensuring system integrity involves implementing protocol authentication between the broker and its administration. A broker's accreditation serves as an indicator of the quality of service they provide, with reputation often gauged by threshold values estimated based on the provider's resources. The increasing demand for computing resources driven by greater market activity presents challenges for individual Cloud Service Providers (CSPs) in dynamically meeting these demands while sustaining the promised quality of service (QoS). Federated cloud computing, where CSPs collaborate to share underutilized resources, offers a solution by enhancing both availability and reliability. Effective resource management within the Infrastructure as a Service (IaaS) model is essential to maintaining QoS and maximizing resource efficiency. This paper introduces an innovative IaaS cloud architecture that reimagines traditional cloud computing by emphasizing virtualization, virtual machine migration, and resource consolidation to bolster service reliability and security. Additionally, it suggests the integration of a Trust Manager with a Broker Manager to improve service level agreement (SLA) oversight and trust assessment. The use of user profiling and advanced ranking algorithms, such as the Deep Q-based Algorithm and the Service Measurement Index (SMI), further facilitates the evaluation and selection of cloud service providers within the IaaS framework.

IndexTerms—Federated cloud computing VM migration Infrastructure as a Service Quality of service Trust Manager (TM) Broker Manager (BM) Cloud User (CU).

I. INTRODUCTION

A new paradigm is taking shape thanks to cloud computing, which enables customers to access on-demand, cost-effective outsourcing services and pay only for the resources they really use [1]. These services are provided by means of the cloud. Online services, data-driven apps, and the exponential growth of user-generated content have propelled the ever-expanding digital world into an unprecedented demand for computer resources in cloud computing environments [2]. To meet the incessant demand for data storage, computing power, and networking capabilities, cloud service providers (CSPs) have emerged as the digital age's backbone with their Infrastructure as a Service (IaaS) solutions. As the market for cloud services grows, it is become more and more difficult for cloud service providers (CSPs) to meet their quality-of-service promises while also effectively managing their customers' varied and dynamic needs for computing resources [3]. These shifting dynamics are largely responsible for the federated cloud computing paradigm's meteoric rise to prominence.

In this federated model, cloud service providers work together to create an environment in which their idle computing resources are shared and pooled. Distributed cloud computing, the resulting federation, has several advantages, such as higher availability and reliability [4]. By working together, the federation can circumvent the limitations that individual CSPs face in maintaining QoS, especially during times of low usage or strong demand for resources. To supply cloud-related infrastructures, the Federated Cloud Architecture aggregates numerous IaaS providers in a distributed and heterogeneous fashion. A federated cloud is another name for this setup [5]. Finding and implementing the best cost-effective cloud service provider for the services is an interesting challenge under current conditions. The capacity to ensure optimum utilization of computer resources even when total demand is minimal is one of the most remarkable properties of federated cloud computing [6]. Infrastructure as a service (IaaS) offers from cloud federation CSPs are made more apparent by this distinctive feature, which emphasizes the requirement of implementing appropriate strategies for managing resources.

These solutions are essential for maximizing the potential of idle computer resources and preserving the integrity, availability, and dependability of quality of service [7]. In Figure 1 we can see the big picture of the federated cloud architecture. The federation broker is responsible for distributing client requests among participating CSPs so that cloud computing's multitenancy feature can be utilized. To get there, we look at a bunch of objective functions, including several related to statistical multiplexing. Companies relying on the cloud run the risk of having their reputations tarnished due to management issues, providers' incompetence in protecting customer data, and providers' lack of transparency [8]. Which is why it's crucial to build trust so that people regard the cloud as reliable. Research into federated clouds has increased in recent years as a potential solution to widespread problems that are data-and computation-intensive. The intricacy of service delivery methods necessitates trust management for decentralized cloud services. Establishing trust amongst clients, cloud service providers, and cloud providers is crucial for a successful deployment in a federated cloud environment, which is open, dynamic, and unpredictable [9]. New protocols and technologies are constantly in demand because they may be used to analyze and enhance cloud computing services, brokers, and providers' levels of security.

Authentication, authorization, data protection, and other similar concerns must all be considered while designing a federated cloud service's security architecture. As we move towards the cloud, these core security objectives—which constitute the security principles—become vitally necessary. Cloud environments necessitate the use of risk assessment and evaluation tools to properly handle these security concerns with cloud services before selection. Any level of evaluation can be applied to trust as it pertains to achieving architectural safety [11].

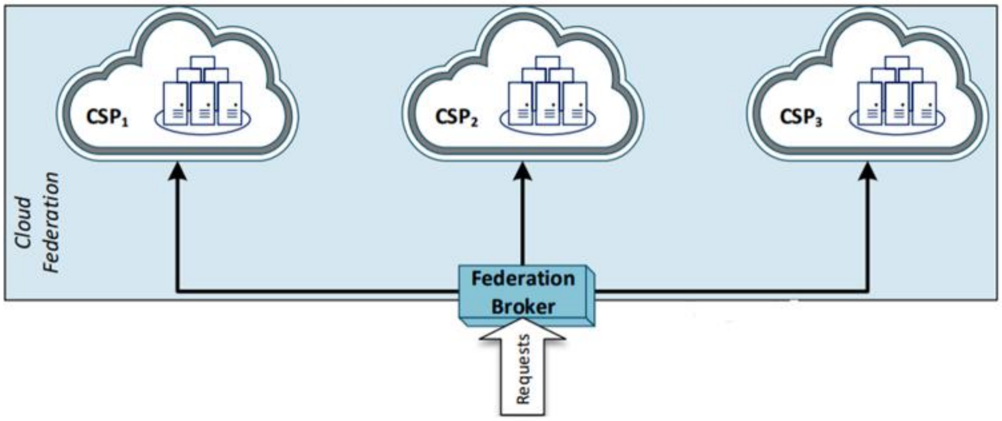

Fig. 1: An Introduction to Federated Cloud Concepts.

Comparing two entities establishes a level of trust between them; a reputation, however, is a more comprehensive assessment of that entity. To make a trust decision in the absence of defined criteria, it is essential in the practice to choose the supplier based on a variety of attributes. Federated clouds offer a larger pool of resources, which improves efficiency and quality while decreasing costs. There is potential for cost-effectiveness to be enhanced in this manner. Both the user and the supplier can profit from modifications that optimize resource utilization, increase the system's throughput, or decrease the time needed to execute a task for a given cost [12].

II. LITERATURE REVIEW

Over the course of several years, the industry has seen a widespread adoption of the use of containers in multi-cloud environments. [13] Containers are software packages that are both standalone and executable, and they contain everything that is required to run an application. The application's settings, libraries, code, and system tools are all part of this. A software and all of its dependencies can be packaged together using containers. This facilitates the easy and modification-free migration of the program to various environments [14]. Containers also allow for customization of the application. The multi-cloud environment, on the other hand, refers to the spread of cloud assets [15]. When organizations use containers in a multi-cloud context, they are able to achieve flexibility, agility, and cost efficiency. A uniform environment allows developers to construct applications, and it is simple to move those apps between many cloud platforms. This allows developers to take use of the distinct advantages that each cloud platform offers, such as large infrastructure, seamless integration, and comprehensive data analytics. Figure 2 provides an illustration of containerization in multi-cloud environments. An overview of apps that have been deployed inside different cloud topologies, such as public, private, and hybrid models, is presented in this image. The fact that every cloud hosts many programs in containers demonstrates the isolation and mobility that containerization offers. These apps execute different types of tasks using the binaries and libraries that are necessary for those tasks. A container platform layer, exemplified by technologies like Kubernetes or Docker, is responsible for managing and orchestrating these containers across different cloud environments [16]. Efficient application deployment and operations in a multi-cloud environment may be achieved using this architecture's flexible and scalable approach, all while ensuring consistency and robustness across platforms.

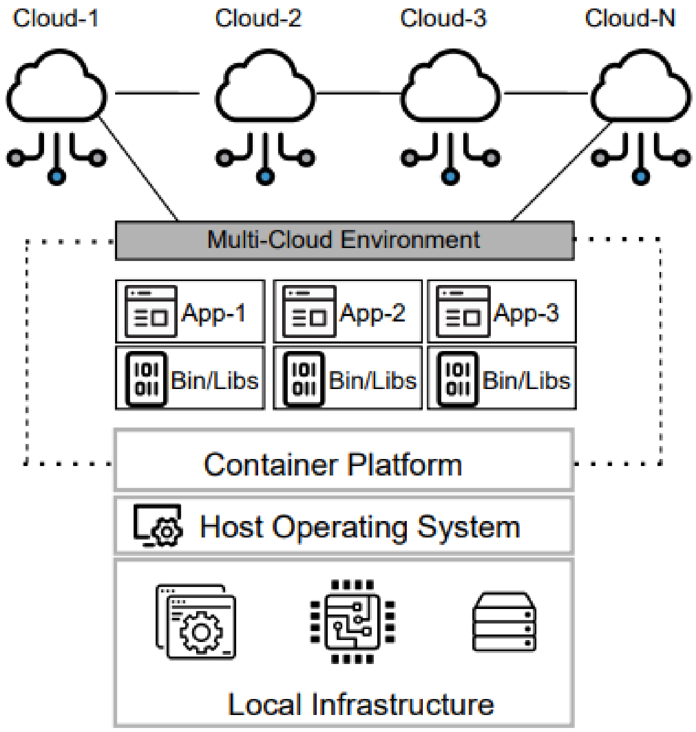

Fig. 2. The use of containers in a multi-cloud setting

Containerization in a multi-cloud context, on the other hand, is not without its obstacles, which organizations can encounter [17]. The architectural design phase, the system implementation phase, and the establishment of an automated development infrastructure are all examples of activities and phases that may present these issues during the development of container-based applications. It is also possible for the difficulties to manifest themselves during the process of testing the system, the process of coding, and the phase of deployment. Organizations may encounter difficulties during architectural design with tasks like choosing suitable patterns, strategies, and tools for container orchestration that can manage containers across different clouds and ensuring the smooth integration of containerized components with current systems [18]. The development of effective Continuous Integration/Continuous Deployment (CI/CD) pipelines that can manage deployment scenarios across multiple clouds while keeping performance and security principles consistent is another challenge that organizations may encounter when establishing an automated development infrastructure [19]. In a similar vein, during the period of system deployment, obstacles may include both the optimization of resource utilization to effectively control costs and the resolution of any compatibility issues that may arise between containers and the various cloud platforms. Additionally, the academic community faces issues because of the current state of affairs with containerization in several clouds. Whether it be a one-of-a-kind pattern, an unknown difficulty, or a creative solution, it is the responsibility of the researchers to navigate through this large array of information to discover the specific features that they are looking for.

A coherent and comprehensive grasp of the subject matter is delayed as a result of the dispersed nature of this knowledge, which further complicates circumstances. Recent research (for example, [20]) has brought to light the fact that the software development life cycle is intricately tied to the difficulties that arise in the design, development, monitoring, and testing of containerization applications in environments employing multiple clouds. The term "multi-cloud computing" describes a setup where multiple separate cloud environments are used, each of which does not rely on any one third party or cloud provider. Twenty-one, so Cloud is designed on a module-based platform with a primary focus on platform as a service (PaaS), which addresses and improves the features of the multi-cloud system, including portability, elasticity, resource provisioning, and availability [21]. SoCloud is based on the OASIS service components architecture. When implementing multi-cloud, ten different cloud providers are used, including DELL KACE, CloudBees, dotCloud, Heroku, Eucalyptus, Windows Azure, Amazon EC2, OpenShift, Jelastic, and Appfog private cloud. Load balancing, service development, node provisioning, constraints validator, monitoring, platform as a service (PaaS) deployment, workload management, controller components, and software as a service (SaaS) deployment are all factors in the launch of soCloud architecture. [22].

III. METHODOLOGY OF PROPOSED WORK

The three primary layers of the cloud operating system are depicted in Figure 3 in an Infrastructure as a Service (IaaS) cloud architecture. The Drivers Layer is concerned with hardware abstraction; it uses device drivers to convert OS requests into hardware commands and connects to physical components of data centers, such as servers and networking gear. Hypervisors and other resource management technologies enable the Core Components Layer to build and maintain virtual machines (VMs), which in turn optimize the allocation of system resources. To further facilitate VM access to these resources, this layer provides interfaces for networking and storage. Security tools, graphical user interfaces, orchestration frameworks, monitoring solutions, and automation capabilities make up the High-Level Tools Layer that concludes. Friendly user interfaces are also a part of it. Automation, security, performance tracking, and the deployment of virtual machines (VMs) are all made possible by these technologies. As a result, the IaaS architecture becomes a full cloud operating system.

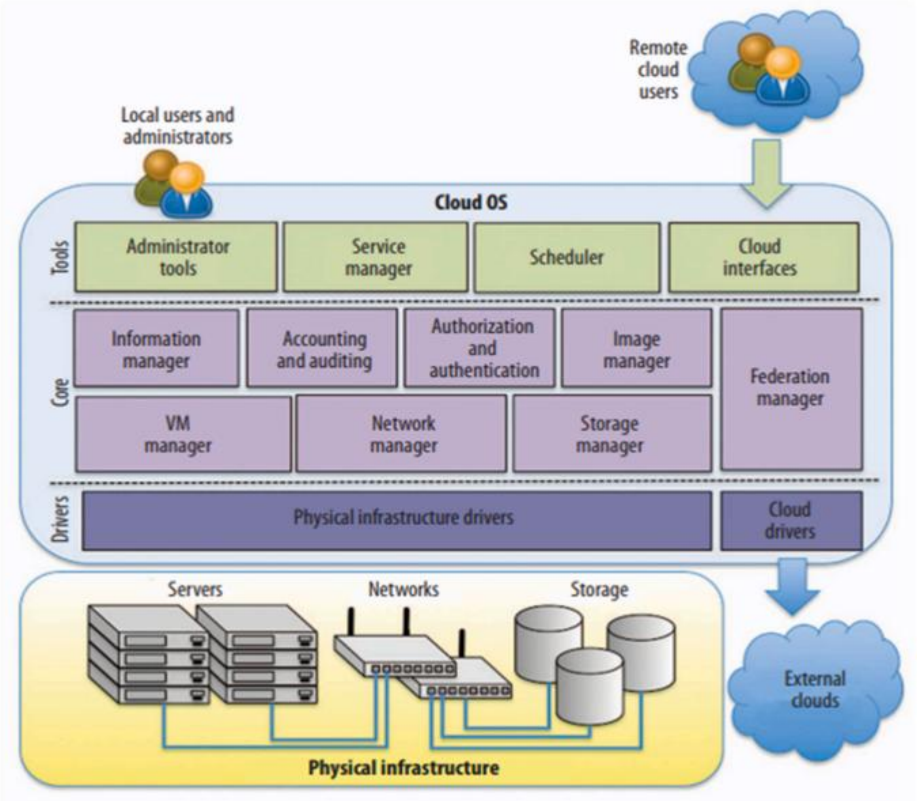

Fig. 3: Internet as a service (IaaS) cloud architecture component.

This study's recommended cloud architecture has several key features that set it apart from more conventional cloud designs. To ensure the sustainability of cloud computing in the long run, it places an emphasis on virtualization methods. Figure 4 shows the proposed federated cloud architecture for infrastructure as a service. As part of this approach, cutting-edge concepts like CU and Reputation Management are included. These ideas enhance the reliability and safety of cloud services by calculating trust factors and using specific algorithms for reputation scoring. Tracking SLAs and evaluating trust are both improved by integrating TM and BM. User Profiling is a novel feature of cloud user management that divides users into three distinct types of profiles: personal, social, and business. A combination of SMI features and ranking algorithms, such as the Deep Q-based Algorithm, allows for more accurate identification and evaluation of cloud service providers. Moreover, the Banker's algorithm and the SLA Management methodology as a whole lead to resource efficiency. The following sections offer a comprehensive breakdown of this cloud design's components.

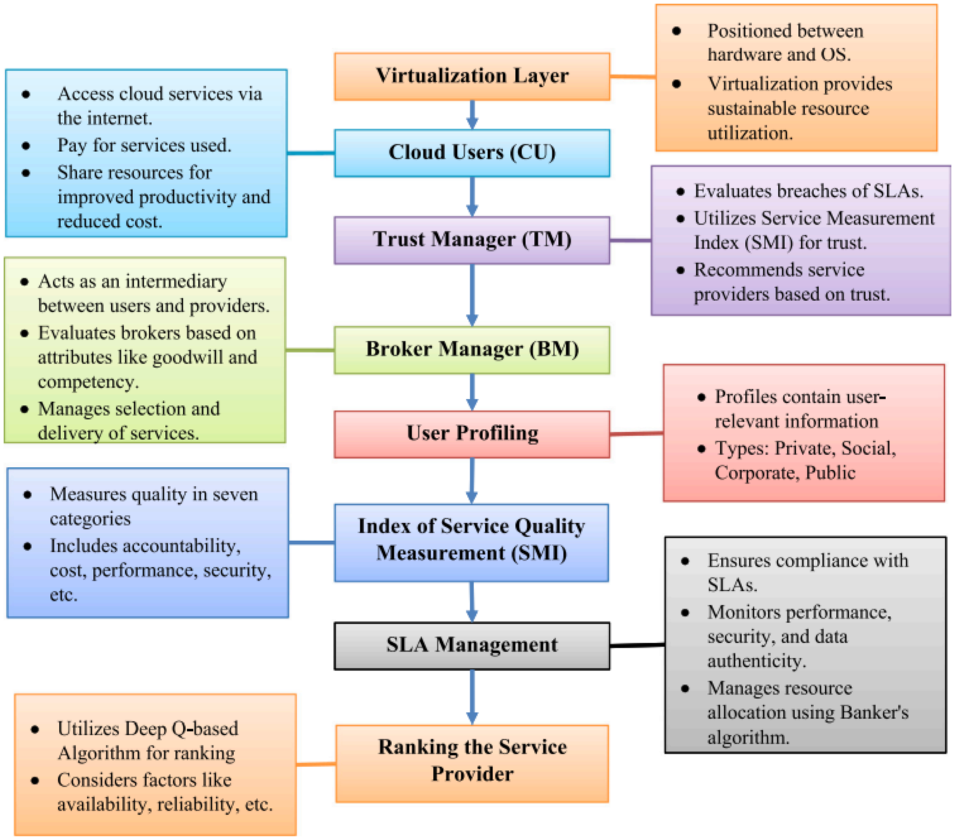
Virtualization Layer
Positioned between hardware and OS.
Virtualization provides sustainable resource utilization.
Access cloud services via the internet.
Pay for services used.
Share resources for improved productivity and reduced cost.
Cloud Users (CU)
Trust Manager (TM)
Evaluates breaches of SLAs.
Utilizes Service Measurement Index (SMI) for trust.
Recommends service providers based on trust.
Acts as an intermediary between users and providers.
Evaluates brokers based on attributes like goodwill and competency.
Manages selection and delivery of services.
Broker Manager (BM)
User Profiling
Profiles contain user-relevant information
Types: Private, Social, Corporate, Public
Measures quality in seven categories
Includes accountability, cost, performance, security, etc.
Index of Service Quality Measurement (SMI)
SLA Management
Ensures compliance with SLAs.
Monitors performance, security, and data authenticity.
Manages resource allocation using Banker's algorithm.
Utilizes Deep Q-based Algorithm for ranking
Considers factors like availability, reliability, etc.
Ranking the Service Provider

Fig. 4: Infrastructure as a Service (IaaS) Federated Cloud Architecture Overview.

Virtualization layer

Under this model, it sits between the OS and the exposed hardware. This is the physical location where the virtualization layer resides. Operating systems encounter a simulated version of the hardware when installed in this way. "Have machine" describes the hypervisor's working environment, whereas "guest machine" describes the system component that makes use of the virtual machine. The two words mean the same thing. A hypervisor's principal function is to establish a connection between a host computer and a virtual machine (VM), facilitating communication between the two. The virtual machine will access the host computer's resources, which are partitioned by this process as well.
This is an additional feature of the virtual machine.

Cloud User (CU)

The services utilized by the CU in its operations are supplied by CSP. Hosting the services on the internet and making them available online is how the CU receives the services that are supplied to it. Users that make use of cloud computing can gain access to these services in a quick and easy manner whenever they like, and they are only required to pay for the services that they employ. Using virtualized systems, service providers can sell their offerings to a large number of clients, which guarantees that the available resources are utilized to their fullest capacity. Customers may have the idea that the resources they have access to were produced specifically for them if service providers attend to the demands of end users and supply them with the resources they require. Improved system productivity and reduced system cost are both result of user capacity to share resources such as processors, memory, storage, bus, and networking, among other things.

Trust Manager (TM)

In the event that a user or CSP violates a SLA, TM is a key component in the investigation. The CSPs' belief-related product and service offers to the CUs are thoroughly investigated by the TM. TM offers its services to credit unions that have trust. TM's operations are in line with the SLA's specifications and the application, device, and user's QoS functioning. The TM recommends a module to manage prediction and service selection after establishing a trust link with the provider based on the SMI's attributes. It is TM's responsibility to determine the degree of trust according to the user's specific SLA requirements. After Trust Manager determines the supplier's credibility, they update the Broker Registry accordingly.

Broker Manager (BM)

In order to analyze performance via service selection and delivery, BM, as an intermediary between Federated Service Providers (FSPs) and users, plays a crucial role. To determine the degree of trust in each broker associated with the provider, many factors are examined, including accreditations, policy compliance audits, self-assessment, reputation, and recommendation. Other criteria include broker attributes (goodwill, competency, and integrity).

The standard of the service (QoS)

Quality of service (QoS) is useful when choosing a service provider because it describes a set of non-functional attributes that make a service unique. When evaluating cloud service quality, it is common practice to compare several attributes that belong to the seven SMI categories. Here, qualitative and quantitative measures are integrated and taken into account simultaneously. Applicable monitoring tools, which may consist of either software or hardware, allow for the evaluation of quantitative features of the service, including its responsiveness, precision, accessibility, and expense. Usability, adaptability, appropriateness, and elasticity are some of the qualitative features that can be measured by looking at the user experience. The below formula calculates the overall deviation (R) between the expected Quality of Service (QoS) and the actual QoS provided by a cloud service provider. It does this by summing the absolute differences across multiple QoS attributes and normalizing them over the total number of attributes or time period (T). This helps in comparing and evaluating the performance of different service providers in a multi-cloud environment.

$$R = \frac{\sum_{i-1}^{N} |(S_i(U_c, SP) - S_i(A_{SP}, SP))|}{T}$$

SLA management

A "service level agreement," or SLA, is a framework that streamlines the process by which customers can find suitable service providers, identify the services they need, negotiate reasonable service levels, and receive service at the agreed upon level. The current method under consideration places the responsibility of monitoring the policies related to each user on the Broker Manager. It is their job to make sure that everyone follows the rules. One possible form for the SLA is a legally binding contract, while another is an official agreement between the CSP and CU. A clear and concise explanation of the provider's promised level of performance is provided in the SLA. Great service is guaranteed, the user's trust is enhanced, and the company's rules are prompted. It keeps tabs on a lot of different things, like service quality, performance, cloud storage security, and data authenticity. A service level agreement (SLA) can help consumers and CSPs reach an understanding. Ensuring enough availability for consumers is ensured by the continuous monitoring of service level agreements.

IV. EXPERIMENTAL RESULTS

BeforeOne toolkit for cloud settings, CloudSim, is used to accomplish the proposed method. It entails modelling the behavior and operation of cloud components like data centers, virtual machines, and resource allocation algorithms. Included in the parameters are the following: Cloud Service Provider ID, Status, Data Centre, Virtual Machine ID, and Start and End Times for Virtual Machines.

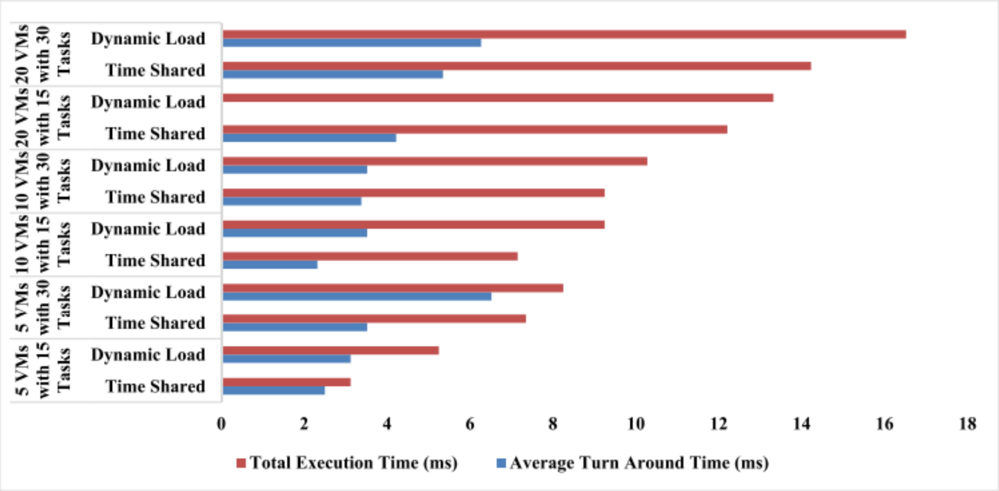

Fig. 5: Various virtual machine and task configurations were tested to compare Dynamic

Figure 5 indicates that the performance of the cloud system is greatly affected by the load management strategy, in addition to the specific virtual machines and task parameters. In situations with a high number of virtual machines (VMs) and tasks, "Dynamic Load" strategies may be more efficient in terms of total execution time, despite having somewhat longer average turnaround times compared to "Time Shared" strategies.

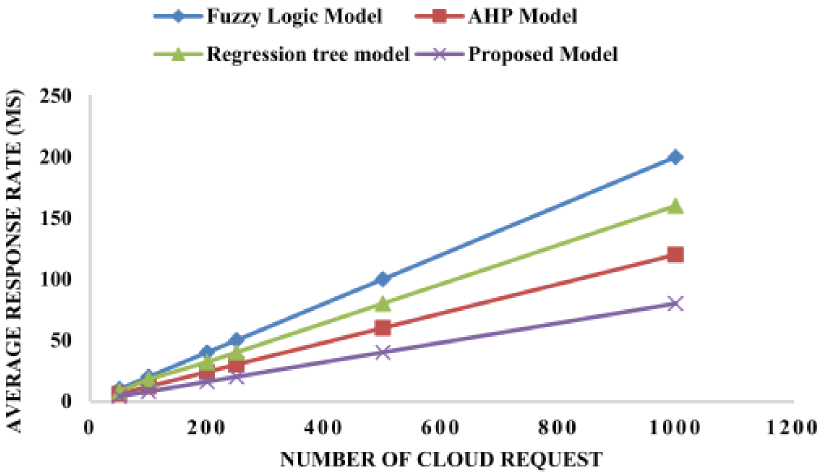

Fig. 6. Comparison of the Current Models with the Suggested Models in Terms of Average Response Time.

Figure 6 shows that the Proposed Model outperforms alternative models now in use, with several clear benefits. By consistently exhibiting significantly reduced reaction times in milliseconds across all tested scenarios, the Proposed Model positions itself as a provider of faster and more responsive cloud services. In addition, the performance gap expands from 50 to 1000 cloud requests, indicating its efficiency in handling larger workloads.

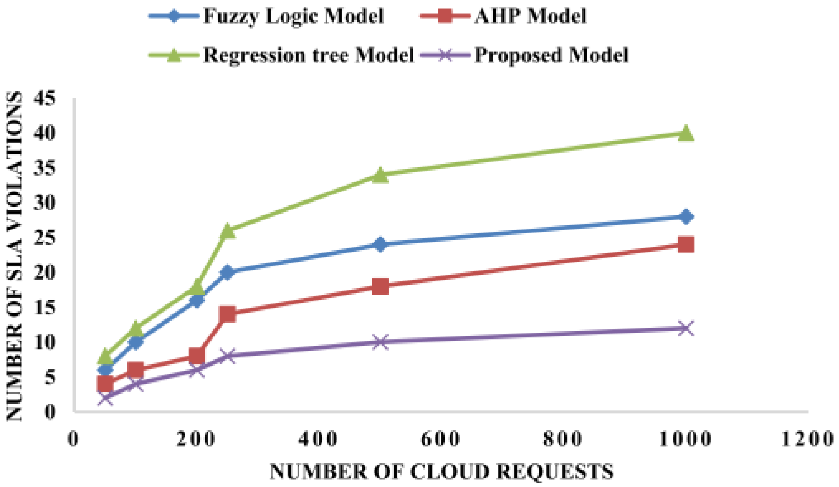

Fig. 7: Problems with the Current Models and the Proposed Ones with SLA

When compared to competing models, the Proposed Model consistently produces fewer SLA breaches (Fig. 7). When comparing the various models' ability to meet SLAs, the Proposed Model clearly stands out. The Proposed Model records two service level agreement (SLA) breaches out of five models that handle fifty cloud requests. The AHP Model records four, the Fuzzy Logic Model six, and the Regression Tree Model eight. This tendency is expected to continue as long as cloud searches continue to rise. The Proposed Model outperformed the competition with the fewest SLA violations (12) even when subjected to the most extreme demand of 1000 cloud requests. Cloud service providers looking to decrease SLA violations and ensure consistent service quality for their clients will find the Proposed Model attractive because it is reliable and efficient in achieving service level agreements.

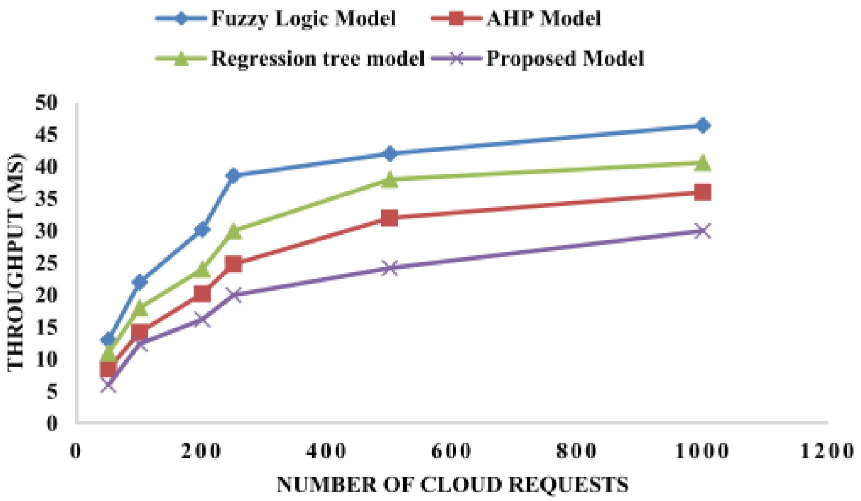

Fig. 8: Comparing the Current Models with the Suggested One: Average Throughput

The data is graphically shown in Fig. 8, which highlights the enhanced performance of the Proposed Model in terms of average throughput. Cloud providers and consumers seeking high-performance cloud computing solutions will find the Proposed Model an enticing alternative due to its enhanced throughput. The Proposed Model outperforms the other models in terms of throughput in every situation that was tested (Fig. 8). This suggests that the Proposed Model can handle cloud requests faster and provide results with less effort. As the volume of cloud requests increases, the Proposed Model's performance advantage becomes more apparent, indicating its scalability and ability to efficiently handle larger workloads.

V. CONCLUSION

The research presented here offers a novel and all-encompassing approach to the problem of resource management in IaaS-enabling federated cloud systems. The proposed infrastructure as a service (IaaS) cloud architecture has demonstrated its capacity to enhance the reliability, safety, and quality of cloud services; it is built on virtualization principles and incorporates novel ideas like Trust Manager and Reputation Management components. In terms of critical performance metrics such as throughput, average response time, and service level agreement violations, the study's thorough simulations and evaluations demonstrated that the Proposed Model performed better than the competition. A cloud simulation toolset called CloudSim was utilized to construct the proposed approach for managing resources. It made it possible to represent several parts of the cloud, including data centers, virtual computers, and systems for providing resources. The evaluation considered factors like the following: status, data center, Virtual Machine ID, Cloud Service Provider ID, and the start and finish times of the VM. After applying the model to CloudSim version 3.0.3, success rate, SLA violations, average response time, and throughput were utilized for evaluation. Using benchmarks such as AHP, regression trees, and fuzzy logic models, we compared the suggested model against the best of the best. By aggregating crucial input data from several CSPs, we can better understand the state and performance of virtual machines. This study examined Time Shared and Dynamic Load techniques across different virtual machine and task configurations to determine their impact on average turnaround time and overall execution time. These findings are vital for improving the implementation of load control mechanisms in cloud systems. The average response time, throughput, success rate, and SLA breaches were all included in the key performance indicators that were investigated in this study. The Proposed Model consistently beat competing models across a variety of cloud request volumes, demonstrating its superior performance in providing quicker and more responsive cloud services.

REFERENCES

1. Luca Acquaviva, Paolo Bellavista, Filippo Bosi, Antonio Corradi, Luca Foschini, Stefano Monti, and Andrea Sabbioni. 2017. NoMISHAP: A Novel Middleware Support for High Availability in Multicloud PaaS. IEEE Cloud Computing 4, 4 (2017), 60–72.
2. Mohammad Matar Al-shammari and Ali Amer Alwan. 2018. Disaster recovery and business continuity for database services in multi-cloud. In Proceedings of the 1st International Conference on Computer Applications & Information Security (ICCAIS). IEEE, 1–8.
3. Max Alaluna, Eric Vial, Nuno Neves, and Fernando MV Ramos. 2017. Secure and dependable multi-cloud network virtualization. In Proceedings of the 1st International Workshop on Security and Dependability of Multi-Domain Infrastructures. 1–6.
4. Khalid Alhamazani, Rajiv Ranjan, Karan Mitra, Prem Prakash Jayaraman, Zhiqiang Huang, Lizhe Wang, and FethiRabhi. 2014. Clams: Cross-layer multi-cloud application monitoring-as-a-service framework. In Proceedings of the 11th International Conference on Services Computing. IEEE, 283–290.
5. André Almeida, Everton Cavalcante, Thais Batista, Nelio Cacho, and Frederico Lopes. 2014. A component-based adaptation approach for multi-cloud applications. In Proceedings of the 7th International Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 49–54.

6. Juncal Alonso, LeireOrue-Echevarria, Valentina Casola, Ana Isabel Torre, MaiderHuarte, EnekoOsaba, and Jesus L Lobo. 2023. Understanding the challenges and novel architectural models of multi-cloud native applications–a systematic literature review. Journal of Cloud Computing 12, 1 (2023), 1–34.
7. Juncal Alonso, Kyriakos Stefanidis, LeireOrue-Echevarria, Lorenzo Blasi, Michael Walker, Marisa Escalante, María José López, and Simon Dutkowski. 2019. DECIDE: an extended devops framework for multi-cloud applications. In Proceedings of the 3rd International Conference on Cloud and Big Data Computing (ICCBDC). 43–48.
8. Luiz Fernando Altran, Guilherme Galante, and Marcio Seiji Oyamada. 2022. Label-affinity-Scheduler: Considering Business Requirements in Container Scheduling for Multi-Cloud and Multi-Tenant Environments. In Proceedings of the 12th Brazilian Symposium on Computing Systems Engineering (SBESC). IEEE, 1–8.
9. Atakan Aral, Rafael BrundoUriarte, Anthony Simonet-Boulogne, and IvonaBrandic. 2020. Reliability management for blockchain-based decentralized multi-cloud. In Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). IEEE, 21–30.
10. Greg Austin. 2018. Cybersecurity in China: The next wave. Springer.
11. UchechukwuAwada. 2018. Application-Container Orchestration Tools and Platform-as-a-Service Clouds: A Survey. International Journal of Advanced Computer Science and Applications (2018).
12. Kiran Baby and AnupriyaVysala. 2015. Multicloud architecture for augmenting security in clouds. In Proceedings of the 1st global conference on communication technologies (GCCT). IEEE, 474–478.
13. Naylor G Bachiega, Paulo SL Souza, Sarita M Bruschi, and Simone Do RS De Souza. 2018. Container-based performance evaluation: A survey and challenges. In Proceedings of the 6th IEEE International Conference on Cloud Engineering (IC2E). IEEE, 398–403.
14. Armin Balalaie, Abbas Heydarnoori, and PooyanJamshidi. 2016. Microservices architecture enables devops: Migration to a cloud-native architecture. IEEE Software 33, 3 (2016), 42–52.
15. Luciano Baresi, Sam Guinea, Giovanni Quattrocchi, and Damian A Tamburri. 2016. Microcloud: A container-based solution for efficient resource management in the cloud. In Proceedings of the 1st International Conference on Smart Cloud (SmartCloud). IEEE, 218–223.
16. Cornel Barna, HamzehKhazaei, MariosFokaefs, and Marin Litoiu. 2017. Delivering elastic containerized cloud applications to enable DevOps. In Proceedings of the 12th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). IEEE, 65–75.
17. Maxime Bélair, Sylvie Laniepce, and Jean-Marc Menaud. 2019. Leveraging kernel security mechanisms to improve container security: a survey. In Proceedings of the 14th international conference on availability, reliability and security. 1–6.

18. T. Dahlberg, “Longitudinal Study on the Expectations of Cloud Computing *Benefits and an Integrative Multilevel Model for Understanding Cloud Computing Performance,” pp. 4251–4260, 2017
19. Flouris, I., Manikaki, V., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Mock, M., Bothe, S., Skarbovsky, I., Fournier, F., Stajcer, M. and Krizan, T., 2016, June. Ferari: A prototype for complex event processing over streaming multi-cloud platforms. In Proceedings of the 2016 International Conference on Management of Data (pp. 2093-2096). ACM.
20. Hioual, O. and Hemam, S.M., 2015, November. Cost Minimization and Load Balancing Issues to Compose Web Services in a Multi Cloud Environment. In Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (p. 88). ACM.
21. Chen, M. and Zadok, E., 2019, May. Kurma: Secure geo-distributed multi-cloud storage gateways. In Proceedings of the 12th ACM International Conference on Systems and Storage (pp. 109-120). ACM