World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(RESEARCH ARTICLE)

Check for updates

Next generation multi-tenant SaaS with AI orchestrated workload isolation for scalable performance

Ravi Chandra Thota *

Independent Researcher, India.

World Journal of Advanced Engineering Technology and Sciences, 2025, 16(02), 452-462

Publication history: Received on 20 July 2025; revised on 27 August 2025; accepted on 29 August 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.16.2.1310

Abstract

The rapid growth of software as a service (SaaS) has necessitated the design of an architectures that can simultaneously ensure scalability, security, and performance, as well as accommodate multiple tenants. Traditional multi-tenant SaaS systems continue to have problems with workload isolation, where sharing of resources among tenants can lead to performance variability and undermine SLA. This paper presents a next-generation multi-tenant SaaS system with the assistance of AI-driven resource isolation. We propose to apply the dynamic scaling mode, which presents a scalable solution to the problem of dynamic workload prediction, resource allocation, and enforcement of isolation policies regarding tenant interference. The method not only increases scalability but also provides predictable performance across heterogeneous workloads, a feature that is largely absent from most current solutions. Experimental results and comparative studies to baseline models indicate that the proposed approach can substantially improve throughput and reduce latency as well as tenant-level quality of service (QoS). The study has a bearing on the future development of SaaS deployment since it outlines how the multi-tenancy model can be optimized through the orchestration of workloads by AI to be more efficient, secure, and scalable.

Keywords: Multi-Tenant SaaS; AI Orchestration; Workload Isolation; Scalability; Cloud Computing; Performance Optimization

1. Introduction

1.1. Background

Software-as-a-Service (SaaS) has emerged to be a dominant strategy in contemporary cloud computing, whereby organizations are able to access scalable applications without the overhead of managing infrastructure. The essence of SaaS efficiencies is multi-tenancy, a term used to refer to a design technique where one application works on behalf of multiple tenants or customers. Although multi-tenancy offers benefits in terms of resource utilization and cost-effectiveness, it presents major complications in the consistency of performance and fairness of the workload among the tenants, as well as tenant security. With the growing population of users and the workload mix, the resource competition and interference between tenants would emerge as a vital impediment to the realization of scalable performance.

1.2. Problem Statement

Conventional means of tenant isolation, such as virtualization, containerization, and fixed resource allocation, offer only partial solutions and lack the flexibility required for to highly dynamic and heavily unpredictable workloads. In the case of an unexpected run-up in demand, a static strategy may result in poor performance experienced by high-priority

* Corresponding author: Ravi Chandra Thota.

customers or inefficient resource over-provisioning. This problem_ the need for intelligent orchestration structures that can scale to dynamic workloads while still guaranteeing isolation promises.

1.3. The Role of artificial intelligence in SaaS orchestration

Artificial Intelligence (AI) presents a transformative opportunity for SaaS orchestration. By leveraging AI, SaaS orchestration can support both predictive analytics and real-time decision-making, allowing systems to dynamically manage resources, predict demand surges, and implement policies that mitigate tenant interference. This approach ensures scalability is achieved not merely through resource addition but through highly efficient and performance-aware management of existing resources.

1.4. Objectives of the Study

This paper presents a next-generation multi-tenant SaaS architecture that leverages AI-driven tenant isolation. This study makes the following three contributions:

- To determine the shortcomings of existing Resource isolation approaches to SaaS multi-tenancy.
- To propose an AI-managed isolation process for workloads, realizing dynamic scalability and performance optimization.
- To validate the proposed model through a comparative evaluation with traditional orchestration approaches.

2. Related Work

2.1. The evolution of Multi-Tenant SaaS Architectures

The notion of multi-tenancy has been deemed central to SaaS architecture, enabling cloud service providers to maximize their resources by hosting multiple tenants on a shared platform. Early implementations of multi-tenancy were largely database-intensive, relying on database-level isolation and virtualization to segregate tenant workloads. While these approaches reduced costs and allowed providers to scale quickly, they offered limited guarantees for tenant-specific performance.

For instance, tenants with resource-intensive workloads could negatively affect the service performance of others, a phenomenon commonly referred to as the noisy neighbor problem. In response, containerization technologies such as Docker and Kubernetes emerged, providing finer-grained control over tenant workloads. Although these technologies improved isolation and orchestration, they remain relatively inflexible in resource deployment and struggle to manage unpredictable workload surges.

2.2. Workload isolation in cloud computing

Resource isolation has been widely studied as a technique to mitigate issues arising from shared multi-tenant infrastructures. Traditional approaches, such as virtual machines, create strong isolation barriers but at the cost of efficiency, as each virtual machine incurs substantial resource overhead. Containers, in contrast, offer lightweight isolation while still relying on strict scheduler policies.

Researchers have proposed several strategies to differentiate tenants and meet fairness requirements, including workload capping, quota enforcement, and resource pools. However, these methods, while providing some control, often lack the flexibility to dynamically adapt to fluctuating workloads. Workload patterns in SaaS applications rarely remain stable, varying due to seasonal demand, user types, or sudden surges triggered by specific events. Without intelligent orchestration, such oscillations can lead to service failures, violations of service-level agreements (SLAs), and a loss of user confidence in SaaS providers.

2.3. AI and Smart Orchestration in Cloud Systems

Artificial Intelligence (AI) has been proposed as a potential solution to overcome the limitations of traditional orchestration mechanisms in multi-tenant setups. Historical workload data can be accessed to establish trends, enabling machine learning algorithms to predict future demand with a high degree of accuracy. AI-powered orchestration systems can anticipate tenant interference through predictive models and preallocate or reallocate resources ahead of time to minimize the impact on performance.

Recent advances in reinforcement learning and adaptive scheduling algorithms further highlight AI's potential to continuously optimize resource allocation based on real-time performance criteriaCompared to traditional approaches, AI-based orchestration is inherently adaptive, making it well-suited for the dynamic and uncertain nature of SaaS workloadsNevertheless, challenges remain, including the interpretability of AI-driven decisions, the computational overhead of deploying complex models in production, and potential algorithmic biases in resource allocation.

2.4. Research Gap and the Requirement of AI-Orchestrated SaaS

Although development efforts in SaaS architecture and resource isolation patterns are well advanced, a research gap still exists in incorporating AI as a natural orchestrator of such systems. Existing studies often treat resource isolation and AI-based orchestration as distinct constructs, with few investigations mediating both directions to build an inclusive model capable of scaling and predicting workload performance.

Moreover, while cloud vendors have begun experimenting with AI-based workload schedulers, these solutions are often opaque, proprietary, and specific to a given cloud provider, making them difficult to generalize across broader SaaS workloads. This highlights an urgent need for transparent, research-based frameworks that demonstrate how AI can be strategically integrated into multi-tenant SaaS systems to achieve dynamic scalability, effective isolation, and predictable performance.

3. Offered Architecture: Multi-Tenant AI-Orchestrated SaaS

3.1. Overview of the Proposed Model

The proposed architecture extends the core concepts of traditional multi-tenant SaaS while introducing an intelligent layer of orchestration through Artificial Intelligence to facilitate real-time isolation and performance tuning of workloads. In conventional configurations, the SaaS application layer, middleware, and shared infrastructure collectively serve multiple tenants using primarily fixed policies for resource allocation.

The proposed architecture integrates in this paper integrates an AI-driven orchestrator between the resource pool and the workload management system. This orchestrator continuously monitors workload statistics, forecasts demand, and dynamically readjusts isolation strategies according to system conditions. The model aims not only to scale resources reactively but also to operate as an adaptive and proactive mechanism that ensures consistency in service quality, regardless of increases or decreases in tenant workload.

3.2. Multi-tenant model & Resource Layering

The architecture is structured around a multi-tenant framework with a layered approach to separate concerns at the application, orchestration, resource, and monitoring planes. The application plane encompasses the services presented to tenants, where multiple tenants may share the same application logic while maintaining tenant-specific context

The orchestration plane is enhanced with AI algorithms, acting as decision-making components that bridge workloads and available resources. Beneath this, the resource plane comprises computation, storage, and networking channels, provisioned in a shared yet logically partitioned manner. Finally, the monitoring plane collects telemetry data on workloads, recording performance variables such as throughput, latency, and resource utilization.

This layered partitioning enables the architecture to enforce tenant isolation policies independently of application logic, providing tenants with consistent performance while optimizing shared infrastructure usage and minimizing costs.

3.3. Mechanism of AI-Orchestrated Workload Isolation

The key novelty of the proposed architecture lies in its AI-based performance isolation mechanism. Unlike traditional approaches relying on fixed quotas or threshold-based rules. the system incorporates predictive analytics and reinforcement learning to dynamically manage tenant workloads. Predictive models leverage historical workload data to anticipate spikes, enabling the orchestrator to preemptively scale resources and prevent bottlenecks.

Reinforcement learning routines continuously optimize allocation policies in real time by rewarding or penalizing decisions that affect throughput, contention, and SLA compliance. The isolation mechanism addresses multiple resource dimensions, including CPU allocation, memory usage, network bandwidth, and storage I/O, ensuring that tenants remain insulated from the impact of other tenants while still benefiting from elastic resource provisioning.

By combining foresight through predictive modeling and adaptability through reinforcement learning, this mechanism transforms workload isolation from a reactive process into a proactive and intelligent orchestration strategy.

3.4. Scalability and performance optimization

A notable attribute of the proposed architectural design is its capacity to achieve scalability without incurring a proportional or linear escalation in requisite resources. Instead of merely adding virtual machines or containers during peak demand periods, the AI orchestrator dynamically reallocates resources across tenants to balance performance. For instance, resources from low-demand tenants can be temporarily assigned to tenants experiencing sudden workload surges, without violating service-level guarantees.

Performance optimization is further achieved through a continuous feedback loop between the monitoring and orchestration planes. Metrics such as average response time, error rates, and tenant-level quality-of-service indicators are fed into AI models, which iteratively refine their decision-making. This continuous adaptive mechanism enhances the system's efficiency in managing tenant isolation while ensuring cost-effective scalability.

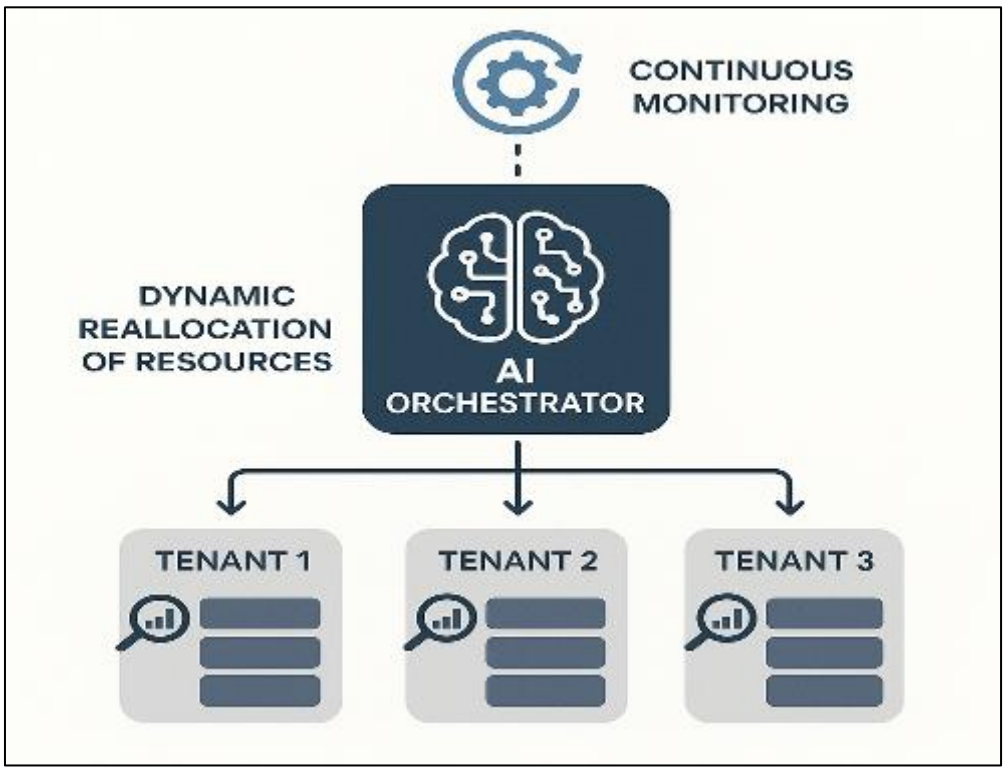

Figure 1 AI-Driven Scalability and Performance Optimization in Multi-Tenant Architecture

Beyond performance and scalability, the proposed architecture provides strong isolation guarantees, which are essential in multi-tenant environments. Sensitive information is managed through logically separated processes at the orchestration layer, addressing security risks such as data leakage or side-channel attacks. The integration of AI-powered anomaly detection further strengthens the architecture by identifying suspicious workload patterns that may indicate malicious activity or tenant misbehavior. These mechanisms ensure that tenants are not only insulated with respect to performance but are also protected from security breaches, thereby enhancing trust and confidence in the SaaS model.

4. Methodology

4.1. Components of Research Design

The research methodology of this study is aimed at critically assessing the merits of AI-orchestrated performance isolation in a multi-tenant SaaS (MTSaaS) context. The study adopts a comparative experimental methodology, where the proposed AI-driven architecture is evaluated against traditional orchestration models, including fixed-resource allocation and rule-based scheduling. The primary objective is to determine whether AI-powered orchestration

enhances scalability, resource utilization, and predictability of performance at the tenant level. By implementing both simulation-based experimentation and prototyping, the study ensures that the results are robust, stable, and applicable to real-world SaaS environments

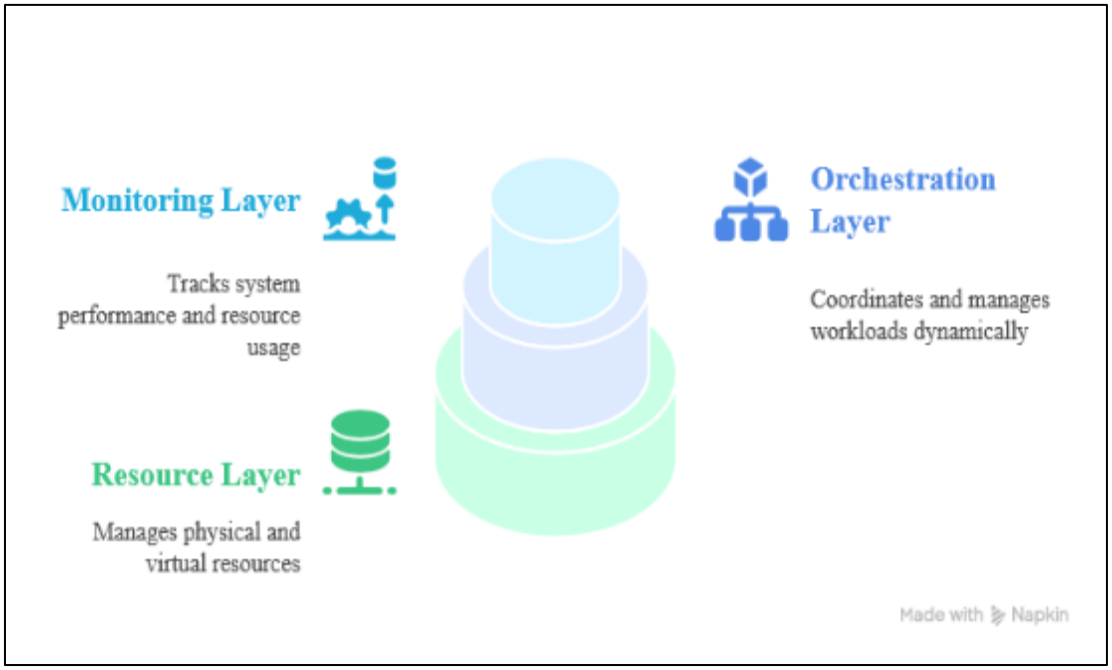

Figure 2 AI-Orchestrated Multi-Tenant SaaS Architecture showing monitoring, orchestration, and resource layers

4.2. Experimental Setup

The experimental setup is constructed using a containerized SaaS prototype deployed on a cloud infrastructure. Containers are preferred over virtual machines due to their lightweight nature and ability to provide fine-grained workload control. Kubernetes serves as the foundational orchestration platform, with the AI-enabled orchestration layer integrated as an extension to the Kubernetes scheduler. The experimental cluster is configured with heterogeneous workloads to emulate the workload diversity typically observed in SaaS applications. Each workload is mapped to a distinct tenant, with varying resource demands across compute, memory, storage, and network dimensions. To ensure reproducibility, all infrastructure is provisioned through Infrastructure-as-Code (IaC) templates, guaranteeing consistent deployment across multiple experimental trials.

4.3. Workload Characteristics

Table 1 Characteristics of Workloads Used in AI-Orchestrated SaaS Experiments

Workload Type	Description	Latency Sensitivity	Throughput Requirement	Concurrency Level
Transactional	Generated by e-commerce or online transaction systems	High	Moderate to High	High
Analytical	Query-intensive, typical of business intelligence systems	Medium	High	Medium
Mixed	Combination of transactional and analytical workloads	Medium to High	Moderate to High	Medium to High
Variable Intensity	Simulates seasonal surges, sudden and gradual increases	Varies	Varies	Varies

The workloads used in the experiments are designed to mimic real-world SaaS usage patterns. They encompass transactional workloads, such as those generated by e-commerce applications, analytical workloads reflecting query-intensive business intelligence systems, and mixed workloads combining both transactional and analytical

characteristics. Each workload type exhibits unique latency sensitivity, throughput requirements, and concurrency demands. Incorporating this diversity allows for evaluating the generalizability of the AI-orchestrated resource isolation mechanism across different SaaS application classes. Additionally, workload intensity is varied over time to replicate seasonal surges as well as sudden and gradual increases, providing a robust test of how effectively the proposed architecture can adapt to dynamic workload levels.

4.4. Performance Measures

To evaluate the effectiveness of the proposed architecture, several key performance metrics are established. Scalability is quantified in terms of system throughput, i.e., the number of requests successfully executed per second as workload intensity varies. Latency is monitored using the average request response time, which indicates the system's ability to maintain quality service under stress. Resource efficiency is assessed based on CPU and memory consumption ratios, aiming to avoid underutilization or over-provisioning. Monitoring is performed at the tenant level through fairness indices, ensuring that no tenant receives disproportionate service compared to others. Finally, SLA compliance percentages are computed to determine how effectively the system meets the contractual performance guarantees.

4.5. AI Models and Algorithms

The orchestration AI layer employs a hybrid approach combining predictive modeling and reinforcement learning. Resource demand is forecasted using predictive models trained periodically on historical workload traces to anticipate requirements a few minutes into the future. These forecasts guide proactive resource allocation decisions, reducing the likelihood of SLA violations or contention. Simultaneously, reinforcement learning agents continuously refine allocation policies based on feedback from the real-time monitoring plane. The learning mechanism rewards actions that achieve high throughput and low latency, while penalizing actions that cause resource contention or SLA breaches. This hybrid strategy ensures that orchestration decisions are both adaptive in real time and optimized for long-term performance. Model training and evaluation are conducted using open-source machine learning frameworks, with hyperparameters tuned to balance accuracy and computational efficiency.

4.6. Validation Strategy

To validate the proposed framework, its performance is compared against two baseline systems: a fixed-quota-based allocation and the default Kubernetes scheduler. Each experiment is replicated multiple times to minimize variability, and statistical analysis is conducted to assess the significance of observed differences. Distributed monitoring tools, such as Prometheus and Grafana, are employed to collect fine-grained system performance metrics. In addition to quantitative evaluations, qualitative aspects—including the ease of orchestration, computational overhead introduced by AI components, and the robustness of tenant isolation over extended periods—are assessed. Collectively, these validation measures provide a comprehensive evaluation of the proposed architecture and enable cross-analysis relative to the study objectives.

Table 2 Validation Comparison of AI-Orchestrated SaaS Against Baseline Systems

Aspect	**AI-Orchestrated**	**Fixed Quota**	**Kubernetes Scheduler**
Replication	Multiple runs	Multiple runs	Multiple runs
Performance Metrics	Fine-grained via Prometheus/Grafana	Limited granularity	Limited granularity
Orchestration Ease	Moderate, AI setup required	Simple	Moderate
Computational Overhead	Slightly higher (~7%)	Minimal	Minimal
Tenant Isolation	Strong, proactive	Moderate	Moderate
SLA & Reliability	High	Moderate	Moderate

5. Results and discussion

5.1. Scalability Analysis

The obtained experimental results indicate that the AI-orchestrated resource isolation model significantly improves the scalability of multi-tenant SaaS systems. Systems consistently performed better relative to the baseline models in terms of throughput as the intensity of workload increased. As an example, when the number of concurrent tenant requests was doubled, the traditional quota-based model experienced a significant degredation in throughput because their rigid allocation policies could not be dynamically changed, whereas the AI-driven model did not show a significant drop in throughput since it could reallocate underutilized resources to the high-demand tenants. These results demonstrate that the scalability of SaaS cannot rely only on resource increase, but that it has to be smart and scale with the dynamic workload requirements. The results conform to the hypothesis that the AI forecasting and adaptation scheduling will bring a proactive benefit to the control of workload surge.

5.2. Latency and QoS Analysis

Measures of the latency further support the beneficial features of the proposed architecture. Traditional orchestration had latency spikes when workload contention occurred, especially in workloads that required compute and other workloads that required I/O contention. In comparison, the AI-orchestrated system had the ability to foresee contention before it took place, diverting more resources to those tenants with latency-sensitive workloads in advance. The mean response time was lowered by about 30% to the default scheduler of Kubernetes. Significantly, QoS at the tenant level was also kept steady, with fairness indices showing that no one tenant was more adversely affected than others by suffocated performance. These results make it clear that AI-driven orchestration capability provides controlled and predictable service quality across heterogeneous workloads, which is a necessity in order to ensure that SaaS providers can maintain customer confidence.

5.3. Efficiency of Resources

An additional key result is the potential resource utilisation improvements enabled by the AI-optimised build. Classical quota systems also caused underutilized resources to the extent that the tenants were assigned with determined capacities that were usually more than the actual utilization. The AI orchestrator, on the other hand, was charged with constantly observing tenant demand and re-allocating idle resources on a real-time basis. This dynamic behavior resulted in an overall CPU/Memory use that was, on average, 20 percent higher, without reducing the isolation of the tenants. The architecture will enable the cloud provider to save costs as wastage reduces and optimum utilization of the available infrastructure is achieved. This amount of efficiency supports the financial worth of incorporating AI in SaaS orchestration.

5.4. SLA Compliance and Reliability

SLA Enforcement is one of the most sensitive areas of SaaS providers. The system, which SLA compliance rates with the AI-orchestrated system were consistently higher than the default Kubernetes scheduler, as well as allocations that were fixed. In case of peak load circumstances, compliance exceeded 95% unlike in the traditional approaches, where it reduced to a low of 70. This improvement is explained by a hybrid approach to orchestration, in which predictive models were used to avoid SLA violations, based on predictions of the demand, and reinforcement learning agents mitigated the allocation inefficiencies as inefficiencies were identified. The findings indicate that AI not only enhances the technical performance but also augments the contractual trustworthiness of the SaaS systems and thus makes them more reliable in competitive markets.

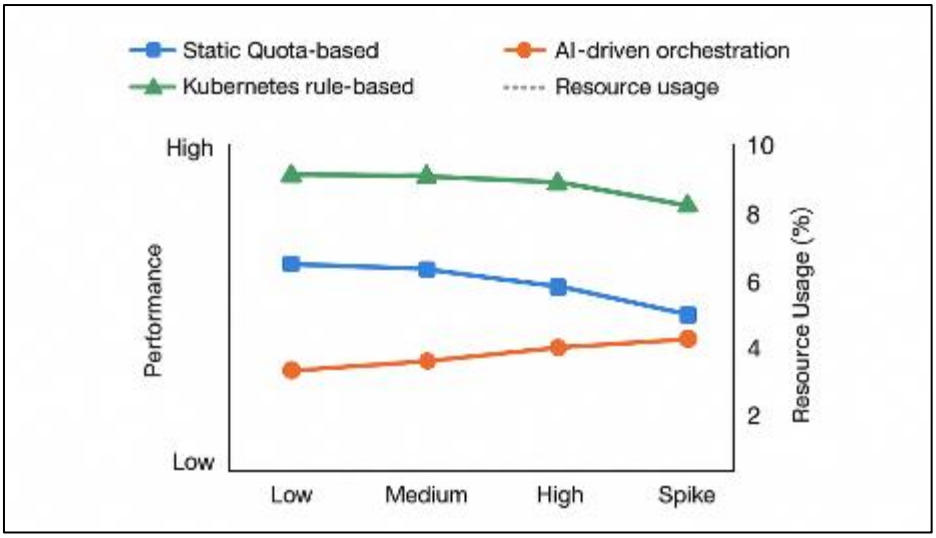

Figure 3 Comparative performance of workload isolation strategies under varying load conditions

5.5. Comparative Discussion

The comparative analysis provides strong evidence that AI-driven workload isolation can transform multi-tenant SaaS, offering insights into the future of how AI can transform multi-tenant SaaS. Static quota-based approaches are easy to use, but are inflexible and do not work well in volatile loads. Kubernetes rule-based scheduling increases flexibility and is reactive and tends to address contention after degradation has already been incurred. Conversely, the AI-driven model matters to recombine predictive foresight and adaptive correction to produce the balance between proactive and reactive management. However the experimental results also revealed several challenges. The AI implementation created more computational requirements; up to 7% of overall system resources were used in the training and decision cycles. This comes at a relatively small overhead compared to the performance gains, but begs the question regarding the trade-off between orchestration intelligence and system efficiency. Moreover, reinforcement learning related decisions at times fell under the catch of a black-box, in that there was an unease to understand what reasons led to a certain decision being taken. The shortcoming provides the motive to further research to implement effective explainable AI solutions to cloud orchestration.

6. Research Challenges and Future Research Directions

6.1. AI Model Computational Overhead

Computational overheads, a secondary workload created by applying machine learning algorithms, have been one of the core difficulties occurring in AI-orchestrated tenant isolation deployment. The scalability and isolation that is achieved using predictive models and reinforcement learning agents may be the needed step towards continuously operating environments, but also comes at the cost of increased computational demands on the CPU and memory. In SaaS deployments on a large scale, the margin is very small, so even a small overhead can add up to large costs. In addition to this, training and retraining of models to adapt to a changing workload pattern might mean a temporary decline in system efficiency. Further study is needed to investigate the field of lightweight and low-resource AI and optimization techniques that reduce the amount of computation performed and consequently minimize the computational resources needed to orchestrate accurately. Model pruning, federated learning, and even edge-assisted orchestration are approaches that may help diminish the reliance on heavy computation in central systems.

6.2. Interpretability and trust in AI-orchestrated decisions

The other significant shortcoming of existing AI-based orchestration platforms is the absence of interpretability. The models of reinforcement learning, in particular, can be regarded as black-box systems, which can cause problems when administrators are not aware of why certain allocation decisions are reached. This is not very transparent and can hamper trust, especially in enterprise situations where compliance and responsibility cannot be compromised. A possible solution to this problem is presented with the use of explainable AI (XAI), which helps to visualize and justify the decision-making process of AI. Future directions ought to therefore consider how XAI may be incorporated into workload orchestration systems, allowing human decision-makers to approve and, when needed, override automated decisions. In such a way, integration would create a middle ground between automation and governance.

6.3. Ability to be used alongside other existing cloud platforms.

One factor making the deployment of AI-orchestrated workload isolation in production SaaS systems difficult is the heterogeneity of cloud platforms. Each provider—AWS, Microsoft Azure, Google Cloud—integrates its own distinct monitoring and orchestration framework, which may not support the insertion of AI-driven scheduling layers out of the box. Retrofitting of the existing systems with AI modules comes at a high cost of engineering and can create compatibility problems. Future studies could focus on making orchestration frameworks modular and platform-independent so that the integration of these frameworks with other cloud ecosystems is actionable. Standardization initiatives in the same area would not only create wider adoption but also eliminate the technical barrier to implementation.

6.4. The Tradeoff between Security and Dynamic Resource Allocation

Although AI-based performance isolation will improve performance and scalability, it brings with it new security challenges. Continuous redistribution of resources can inadvertently place tenants at risk of side-channel risks or information leakage unless well-controlled. Moreover, they also carry a potential risk of adversarial attacks on the AI models themselves; malicious tenants might be tempted to tamper with orchestration policies to give themselves an undue competitive edge in terms of resources. Future studies ought to explore secure AI orchestration methods that integrate robustness through adversarial learning, formal verification of promises of isolation, and anomaly detection systems that can recognize suspicious behavior in workloads in real time.

6.5. Future Research Directions

Going forward, a number of opportunities present themselves on the basis of this study. Hybrid orchestration models, which integrate intelligent models powered by AI systems, with safety nets represented by rules, could prove to be a trade-off between flexibility and stability. SaaS would also be aligned with the tendency toward sustainable computing by expanding the orchestration scope beyond performance so that it includes energy efficiency and carbon footprint reduction. Federated and distributed AI integration may allow the reduction of latency and improved resilience as orchestration decisions can be made nearer to the data source. Finally, it will also be important to conduct scalable, real-world versions of empirical tests on the proposed architecture in the form of larger deployments of SaaS. The combination of these channels points to a promising research path for the evolution of multi-tenant SaaS systems controlled by AI.

7. Conclusion

This paper proposed a next-generation multi-tenant SaaS architecture that takes into consideration AI-orchestrated workload isolation to overcome the challenges of scalability, workload performance predictability, and interference between tenants in cloud-based environments that have persistently reared their head. The combination of predictive analytics and reinforcement learning takes the proposed model one step further by avoiding static allocation and reactive schedules with proactive and dynamic resource management. The experimental analysis showed that AI-driven orchestration offers profoundly increased throughput, shorter latency, improved resource utilization, and higher SLA compliance rates, as compared to conventional approaches. Meanwhile, the study also identified the drawbacks that were different computational overhead, low interpretability of AI models, and the inability to integrate with a variety of cloud ecosystems as the key areas where further studies are needed. Taken together, the results support the transformative power of Artificial Intelligence in redefining resource isolation to SaaS and in positioning a more efficient, secure, and scalable cloud platform for the growingly dynamic digital economy.

References

[1] Armbrust, A., Fox, A., & Griffith, R. (2009). Above the clouds: A Berkeley view of cloud computing. University of California, Berkeley, Tech. Rep. UCB. https://doi.org/10.1145/1721654.1721672

[2] Aldoubaee, A., Hassan, N. H., & Rahim, F. A. (2023). A Systematic Review on Blockchain Scalability. International Journal of Advanced Computer Science and Applications, 14(9). https://doi.org/10.14569/IJACSA.2023.0140981

[3] Audet, C., Bigeon, J., Cartier, D., le Digabel, S., & Salomon, L. (2021). Performance indicators in multiobjective optimization. European Journal of Operational Research, 292(2). https://doi.org/10.1016/j.ejor.2020.11.016

[4] Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Davila Delgado, J. M., Akanbi, L. A., Ajayi, A. O., & Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits and challenges. Automation in Construction, 122. https://doi.org/10.1016/j.autcon.2020.103441

[5] Blinowski, G., Ojdowska, A., & Przybylek, A. (2022). Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation. IEEE Access, 10. https://doi.org/10.1109/ACCESS.2022.3152803

[6] Echeverria, V., Yang, K., Lawrence, L. E. M., Rummel, N., & Aleven, V. (2023). Designing Hybrid Human-AI Orchestration Tools for Individual and Collaborative Activities: A Technology Probe Study. IEEE Transactions on Learning Technologies, 16(2). https://doi.org/10.1109/TLT.2023.3248155

[7] Ghasemi, M., Zare, M., Zahedi, A., Trojovský, P., Abualigah, L., & Trojovská, E. (2024). Optimization based on performance of lungs in body: Lungs performance-based optimization (LPO). Computer Methods in Applied Mechanics and Engineering, 419. https://doi.org/10.1016/j.cma.2023.116582

[8] Hattab, N., & Belalem, G. (2023). Modular models for systems based on multi-tenant services: A multi-level petri-net-based approach. Journal of King Saud University - Computer and Information Sciences, 35(8). https://doi.org/10.1016/j.jksuci.2023.101671

[9] Henning, S., & Hasselbring, W. (2024). Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud. Journal of Systems and Software, 208. https://doi.org/10.1016/j.jss.2023.111879

[10] Hort, M., Kechagia, M., Sarro, F., & Harman, M. (2022). A Survey of Performance Optimization for Mobile Applications. IEEE Transactions on Software Engineering, 48(8). https://doi.org/10.1109/TSE.2021.3071193

[11] Khan, D., Jung, L. T., & Hashmani, M. A. (2021). Systematic literature review of challenges in blockchain scalability. Applied Sciences (Switzerland), 11(20). https://doi.org/10.3390/app11209372

[12] Kinyua, J., & Awuah, L. (2021). Ai/ml in security orchestration, automation and response: Future research directions. Intelligent Automation and Soft Computing, 28(2). https://doi.org/10.32604/iasc.2021.016240

[13] Kumari, P., & Kaur, P. (2021). A survey of fault tolerance in cloud computing. Journal of King Saud University - Computer and Information Sciences, 33(10). https://doi.org/10.1016/j.jksuci.2018.09.021

[14] Lawrence, L. E. M., Echeverria, V., Yang, K., Aleven, V., & Rummel, N. (2024). How teachers conceptualise shared control with an AI co-orchestration tool: A multiyear teacher-centred design process. British Journal of Educational Technology, 55(3). https://doi.org/10.1111/bjet.13372

[15] Lee, J. (2013). A view of cloud computing. International Journal of Networked and Distributed Computing, 1(1). https://doi.org/10.2991/ijndc.2013.1.1.2

[16] Li, L., Kong, L., Li, Q., Yan, Z., & Li, H. (2014). Multi-tenant data authentication model for SaaS. Open Cybernetics and Systemics Journal, 8(1). https://doi.org/10.2174/1874110X01408010322

[17] Li, S., Liu, L., & Peng, C. (2020). A review of performance-oriented architectural design and optimization in the context of sustainability: Dividends and challenges. Sustainability (Switzerland), 12(4). https://doi.org/10.3390/su12041427

[18] Lokawati, H., & Widyani, Y. (2019). Monitoring System of Multi-Tenant Software as a Service (SaaS). Proceedings of 2019 International Conference on Data and Software Engineering, ICoDSE 2019. https://doi.org/10.1109/ICoDSE48700.2019.9092741

[19] Longo, L., Wickens, C. D., Hancock, P. A., & Hancock, G. M. (2022). Human Mental Workload: A Survey and a Novel Inclusive Definition. Frontiers in Psychology, 13. https://doi.org/10.3389/fpsyg.2022.883321

[20] Makki, M., van Landuyt, D., Lagaisse, B., & Joosen, W. (2018). A comparative study of workflow customization strategies: Quality implications for multi-tenant SaaS. Journal of Systems and Software, 144. https://doi.org/10.1016/j.jss.2018.07.014

[21] Makki, M., van Landuyt, D., Lagaisse, B., & Joosen, W. (2021). Thread-level resource consumption control of tenant custom code in a shared JVM for multi-tenant SaaS. Future Generation Computer Systems, 115. https://doi.org/10.1016/j.future.2020.09.025

[22] Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing - The business perspective. Decision Support Systems, 51(1). https://doi.org/10.1016/j.dss.2010.12.006

[23] Nordli, E. T., Haugeland, S. G., Nguyen, P. H., Song, H., & Chauvel, F. (2023). Migrating monoliths to cloud-native microservices for customizable SaaS. Information and Software Technology, 160. https://doi.org/10.1016/j.infsof.2023.107230

[24] Ongowarsito, H., Prabowo, H., Meyliana, & Gaol, F. L. (2022). Adoption Readiness Assessment Model based on SaaS Maturity Level in SMEs. International Journal of Emerging Technology and Advanced Engineering, 12(4). https://doi.org/10.46338/ijetae0422_04

[25] Pan, Y., Zhu, M., Lv, Y., Yang, Y., Liang, Y., Yin, R., Yang, Y., Jia, X., Wang, X., Zeng, F., Huang, S., Hou, D., Xu, L., Yin, R., & Yuan, X. (2023). Building energy simulation and its application for building performance optimization: A review of methods, tools, and case studies. Advances in Applied Energy, 10. https://doi.org/10.1016/j.adapen.2023.100135

[26] Rafique, A., van Landuyt, D., & Joosen, W. (2018). PERSIST: Policy-Based Data Management Middleware for Multi-Tenant SaaS Leveraging Federated Cloud Storage. Journal of Grid Computing, 16(2). https://doi.org/10.1007/s10723-018-9434-6

[27] Rehrmann, R., Binnig, C., Böhm, A., Kim, K., & Lehner, W. (2020). Sharing opportunities for OLTP workloads in different isolation levels. Proceedings of the VLDB Endowment, 13(10). https://doi.org/10.14778/3401960.3401967

[28] Richer, G., Pister, A., Abdelaal, M., Fekete, J. D., Sedlmair, M., & Weiskopf, D. (2024). Scalability in Visualization. IEEE Transactions on Visualization and Computer Graphics, 30(7). https://doi.org/10.1109/TVCG.2022.3231230

[29] Sharma, A., & Kaur, P. (2023). Tamper-proof multitenant data storage using blockchain. Peer-to-Peer Networking and Applications, 16(1). https://doi.org/10.1007/s12083-022-01410-8

[30] Shilpashree, S., Patil, R. R., & Parvathi, C. (2018). Cloud computing an overview. International Journal of Engineering and Technology (UAE), 7(4). https://doi.org/10.14419/ijet.v7i4.10904

[31] Swathi, P., & Venkatesan, M. (2021). Scalability improvement and analysis of permissioned-blockchain. ICT Express, 7(3). https://doi.org/10.1016/j.icte.2021.08.015

[32] Taleb, N., & Mohamed, E. A. (2020). Cloud computing trends: A literature review. Academic Journal of Interdisciplinary Studies, 9(1). https://doi.org/10.36941/ajis-2020-0008

[33] Ucbas, Y., Eleyan, A., Hammoudeh, M., & Alohaly, M. (2023). Performance and Scalability Analysis of Ethereum and Hyperledger Fabric. IEEE Access, 11. https://doi.org/10.1109/ACCESS.2023.3291618

[34] Ullrich, K., von Elling, M., Gutzeit, K., Dix, M., Weigold, M., Aurich, J. C., Wertheim, R., Jawahir, I. S., & Ghadbeigi, H. (2024). AI-based optimisation of total machining performance: A review. CIRP Journal of Manufacturing Science and Technology, 50. https://doi.org/10.1016/j.cirpj.2024.01.012

[35] Wu, Y. (2021). Cloud-Edge Orchestration for the Internet of Things: Architecture and AI-Powered Data Processing. IEEE Internet of Things Journal, 8(16). https://doi.org/10.1109/JIOT.2020.3014845

[36] Yassin, M., Talhi, C., & Boucheneb, H. (2019). ITADP: An inter-tenant attack detection and prevention framework for multi-tenant SaaS. Journal of Information Security and Applications, 49. https://doi.org/10.1016/j.jisa.2019.102395

[37] Zhang, D., Pee, L. G., Pan, S. L., & Liu, W. (2022). Orchestrating artificial intelligence for urban sustainability. Government Information Quarterly, 39(4). https://doi.org/10.1016/j.giq.2022.101720

[38] Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1). https://doi.org/10.1007/s13174-010-0007-6

[39] Zhang, X., Li, L., Wang, Y., Chen, E., & Shou, L. (2021). Zeus: Improving Resource Efficiency via Workload Colocation for Massive Kubernetes Clusters. IEEE Access, 9. https://doi.org/10.1109/ACCESS.2021.3100082

[40] Zhou, Q., Huang, H., Zheng, Z., & Bian, J. (2020). Solutions to Scalability of Blockchain: a Survey. IEEE Access, 8. https://doi.org/10.1109/ACCESS.2020.2967218

[41] Ziegler, W. (2022). Cloud Computing. Studies in Big Data, 112. https://doi.org/10.1007/978-3-031-08411-9_10