INTERNATIONAL JOURNAL OF NEXT-GENERATION ENGINEERING AND TECHNOLOGY (IJNGET)

eISSN: 3087-4319

Volume. 01, Issue. 01, pp. 37-41, December 2024"

Adaptive Latency-Aware Microservice Orchestration and Anomaly-Resilient Edge–Cloud Architectures for Mixed Reality and Time-Critical Applications

Dr. Elena Markovic
Department of Computer Science, University of Amsterdam, Netherlands

Article Received: 05/11/2024, Article Revised: 25/11/2024, Article Accepted: 10/12/2024, Article Published: 30/12/2024

© 2024 Authors retain the copyright of their manuscripts, and all Open Access articles are disseminated under the terms of the Creative Commons Attribution License 4.0 (CC-BY), which licenses unrestricted use, distribution, and reproduction in any medium, provided that the original work is appropriately cited.

ABSTRACT

The rapid proliferation of mixed and augmented reality applications, latency-sensitive IoT systems, and time-critical cloud-native services has intensified the need for adaptive, resilient, and latency-aware microservice orchestration across heterogeneous cloud–edge environments. Emerging immersive applications such as augmented reality commerce and mixed reality surgical systems demand ultra-low latency, high availability, and robust fault tolerance, while simultaneously confronting network instability and resource heterogeneity.

This study proposes a comprehensive architectural and methodological framework for adaptive microservice orchestration and anomaly-resilient deployment across distributed cloud–edge infrastructures. Drawing exclusively upon established literature in microservices scheduling, edge deployment robustness, anomaly detection, Bayesian optimization, and mixed reality systems, the study synthesizes a unified theoretical model addressing latency-awareness, robustness, monitoring scalability, and intelligent anomaly detection.

A multi-layer adaptive orchestration framework is designed, integrating optimal cloudlet placement, robustness-oriented edge deployment, QoS-aware multi-resource management, latency-aware accelerator utilization, decentralized adaptive cloud–edge–dew models, hierarchical monitoring architectures, and machine learning-based anomaly detection. The methodology incorporates Bayesian optimization for orchestration parameter tuning, synthetic minority oversampling for imbalance mitigation, and hybrid anomaly detection pipelines using Random Forest, Support Vector Machines, and log-based failure prediction.

The proposed framework demonstrates improved theoretical latency resilience, enhanced deployment robustness, reduced service failure propagation, and improved anomaly detection reliability in heterogeneous environments. Conceptual evaluation based on industrial surveys, scheduling models, and architectural comparisons indicates improved system adaptability and fault containment compared to centralized orchestration approaches.

Conclusion: The integration of latency-aware orchestration, robustness-oriented deployment, and adaptive anomaly detection forms a foundational architecture for next-generation immersive and time-critical systems. The study contributes a theoretically grounded and practically aligned blueprint for scalable, resilient microservice ecosystems supporting mixed reality and distributed intelligent applications.

KEYWORDS

Microservices orchestration, Edge computing, Latency-aware systems, Anomaly detection, Mixed reality, Cloud-edge architecture.

INTRODUCTION

The convergence of mixed reality technologies, cloud-native microservices, and distributed edge computing infrastructures has catalyzed a fundamental transformation in how digital services are designed, deployed, and consumed. Augmented reality and mixed reality applications are increasingly substituting physical experiences with holographic or virtual overlays, reshaping retail, education, and industrial training landscapes (Rauschnabel, 2021). In clinical contexts, mixed reality technologies have demonstrated promising applications in orthopedic surgery, where real-time visualization and precision guidance impose stringent latency and reliability requirements (Lu et al., 2022).

These emerging application domains demand computational infrastructures capable of delivering ultra-low latency, continuous availability, and adaptive scalability under dynamic network conditions.

Traditional centralized cloud architectures are insufficient to support such latency-sensitive and geographically distributed workloads. Consequently, cloud–edge computing paradigms have gained prominence. Cloudlet-based computation models, which strategically place compute resources closer to end-users, offer a pathway toward reduced latency and enhanced responsiveness (Jia et al., 2015). However, the optimal placement of cloudlets and allocation of users remains a nontrivial optimization problem involving network topology, user mobility, and resource constraints. Moreover, as applications increasingly adopt microservice-based architectures, orchestration complexity escalates. Microservices are inherently modular and independently deployable, yet their distributed interactions create intricate dependency graphs susceptible to latency bottlenecks and failure cascades (Zhou et al., 2018).

The rise of microservices has also intensified the importance of robust deployment strategies in heterogeneous environments. Edge application deployment must consider variability in resource capacity, network stability, and workload volatility (Li et al., 2020). Robustness-oriented edge application deployment frameworks emphasize resilience against node failures and resource contention, ensuring sustained quality of service even under adverse conditions. Similarly, microservices scheduling models across heterogeneous cloud–edge infrastructures must account for IoT-driven workloads characterized by intermittent connectivity and bursty traffic patterns (Filip et al., 2018).

Parallel to deployment challenges, resource management within data centers and edge nodes introduces another dimension of complexity. Multi-resource management systems such as Avalon aim to improve utilization and QoS awareness through coordinated CPU, memory, and network resource allocation (Chen et al., 2019). Spatial multitasking accelerators introduce additional latency considerations, requiring latency-aware scheduling to optimize throughput without degrading service responsiveness (Zhang et al., 2019). These considerations are particularly critical in immersive applications where latency directly influences user experience and system safety.

Time-critical applications further necessitate development frameworks capable of supporting deterministic performance guarantees. The SWITCH workbench proposes development methodologies tailored for time-critical microservice-based cloud-native applications, emphasizing deployment automation and runtime adaptability (Stefanič et al., 2019). Complementing this, decentralized adaptive latency-aware cloud–edge–dew architectures propose hierarchical distribution models to mitigate unreliable network conditions and reduce centralized bottlenecks (Tefera et al., 2019). Hierarchical monitoring architectures such as HSACMA offer scalable monitoring solutions for distributed cloud environments, enabling adaptive management across multiple abstraction layers (Wang et al., 2020).

While significant advances have been made in microservice scheduling, edge deployment robustness, and latency-aware resource management, the integration of anomaly detection and failure prediction remains insufficiently unified within orchestration frameworks. Microservice systems exhibit complex failure modes due to inter-service dependencies and dynamic scaling (Zhou et al., 2018). Log-based failure prediction approaches employing support vector machines and random indexing demonstrate the potential for early detection of system faults (Fronza et al., 2013). Random Forest-based anomaly detection methods further enhance classification performance in distributed environments (Primartha & Tama, 2017). Surveys of anomaly detection in networked systems emphasize the need for adaptive, data-driven techniques capable of handling evolving threat landscapes and workload patterns (Agrawal & Agrawal, 2015; Eltanbouly et al., 2020).

Imbalanced datasets pose additional challenges in anomaly detection, as rare failure events are underrepresented. Synthetic minority oversampling techniques have been shown to optimize classification tasks in intrusion detection and botnet datasets (Gonzalez-Cuautle et al., 2020). Integrating such imbalance mitigation techniques within microservice monitoring pipelines enhances detection reliability in operational environments.

Optimization of orchestration parameters introduces another research dimension. Bayesian optimization provides a principled framework for optimizing expensive cost functions under uncertainty, offering potential applications in adaptive orchestration tuning (Brochu et al., 2010). Despite its demonstrated efficacy in hierarchical reinforcement learning and active user modeling, its integration into cloud–edge orchestration remains underexplored.

A further challenge lies in modularizing legacy systems to transition toward microservice architectures. Machine learning-assisted service boundary detection facilitates the identification of modular boundaries within monolithic systems, thereby enabling systematic migration toward microservices (Hebbar, 2022). Such modularization is crucial for organizations seeking to modernize infrastructure while preserving operational continuity.

Collectively, the literature reveals several critical gaps. First, while individual components such as cloudlet placement, robust edge deployment, QoS-aware resource management, and anomaly detection have been studied independently, there is limited integration into a cohesive, adaptive orchestration framework tailored for mixed reality and time-critical applications. Second, latency awareness is often treated as a static constraint rather than a dynamically monitored and optimized variable. Third, monitoring architectures and anomaly detection pipelines are frequently decoupled from orchestration decision-making processes. Finally, optimization techniques such as Bayesian optimization are rarely incorporated into real-time orchestration loops.

This study addresses these gaps by proposing a comprehensive adaptive orchestration framework that unifies latency-aware scheduling, robustness-oriented deployment, hierarchical monitoring, and intelligent anomaly detection across cloud–edge infrastructures. The framework is grounded exclusively in established research and synthesizes their theoretical contributions into an integrated architecture suitable for mixed reality, IoT, and time-critical applications.

METHODOLOGY

The methodological design of this study is conceptual yet rigorously grounded in existing empirical and theoretical research. The framework integrates multiple established components into a unified adaptive orchestration model. Each methodological layer is derived directly from the referenced literature and elaborated to ensure theoretical coherence.

The foundational layer involves cloudlet placement and user allocation strategies inspired by optimal cloudlet placement models (Jia et al., 2015). The methodology conceptualizes metropolitan-scale networks wherein edge nodes are positioned to minimize aggregate latency and network congestion. User-to-cloudlet allocation decisions consider proximity, workload intensity, and resource availability. Rather than static placement, the proposed model extends this concept by incorporating continuous monitoring feedback to adjust allocation dynamically in response to workload fluctuations.

Building upon this placement strategy, the second methodological layer integrates robustness-oriented edge application deployment principles (Li et al., 2020). Deployment decisions prioritize not only performance optimization but also resilience against node failures and resource contention. The robustness metric is conceptualized as the capacity of a deployment configuration to maintain service continuity under simulated failure scenarios. This aligns with the notion of robustness as a primary objective in heterogeneous edge computing environments.

Microservices scheduling across heterogeneous infrastructures follows the scheduling model proposed for IoT applications (Filip et al., 2018). Services are decomposed into fine-grained tasks, each with resource and latency requirements. Scheduling decisions consider device heterogeneity, including computational power and energy constraints. The model extends this by integrating latency thresholds derived from mixed reality application requirements (Lu et al., 2022).

Resource management within each node adopts principles from QoS-aware multi-resource management systems such as Avalon (Chen et al., 2019). CPU, memory, and network bandwidth allocations are dynamically adjusted based on service-level objectives. Additionally, latency-aware spatial multitasking principles (Zhang et al., 2019) are incorporated to ensure that accelerator resources are utilized without violating latency constraints.

To support time-critical application development, the framework integrates deployment automation concepts inspired by the SWITCH workbench (Stefanič et al., 2019). Continuous integration and deployment pipelines are configured to support deterministic performance testing prior to production deployment. Latency profiling and robustness evaluation are embedded within the development lifecycle.

The architecture further adopts a decentralized adaptive cloud–edge–dew model (Tefera et al., 2019). Instead of relying solely on centralized orchestration, local edge nodes possess limited autonomy to make latency-aware decisions when network reliability degrades. This hierarchical decision model enhances resilience against network partitions.

Monitoring infrastructure is structured according to hierarchical scalable adaptive cloud monitoring architecture principles (Wang et al., 2020). Monitoring agents operate at service, node, and cluster levels, aggregating metrics for latency, throughput, resource utilization, and anomaly indicators. Monitoring data feeds directly into orchestration engines.

Anomaly detection methodology integrates multiple techniques. Log-based failure prediction employing support vector machines and random indexing (Fronza et al., 2013) forms the basis for service-level anomaly detection. Random Forest classifiers provide complementary detection capabilities, particularly for structured performance metrics (Primartha & Tama, 2017). To address class imbalance, synthetic minority oversampling techniques are applied during model training (Gonzalez-Cuautle et al., 2020). Survey insights from anomaly detection research guide feature engineering and evaluation strategies (Agrawal & Agrawal, 2015; Eltanbouly et al., 2020).

Parameter optimization for orchestration policies is conducted conceptually through Bayesian optimization frameworks (Brochu et al., 2010). Hyperparameters such as scaling thresholds, resource allocation weights, and latency sensitivity coefficients are iteratively tuned to minimize a composite cost function balancing latency, robustness, and resource efficiency.

Finally, migration of legacy systems into this architecture leverages machine learning-assisted service boundary detection techniques (Hebbar, 2022). Dependency graphs and runtime logs are analyzed to identify modular service boundaries, enabling gradual transition toward microservices without service disruption.

RESULTS

The integration of latency-aware placement, robustness-oriented deployment, hierarchical monitoring, and anomaly detection produces several conceptual performance improvements.

First, dynamic cloudlet placement combined with decentralized orchestration reduces latency variability under fluctuating network conditions. The theoretical reduction in response time volatility aligns with findings on latency-aware architectures in unreliable networks (Tefera et al., 2019). Mixed reality applications benefit particularly from reduced jitter, enhancing user experience and operational safety (Lu et al., 2022).

Second, robustness-oriented deployment significantly improves fault containment. By evaluating deployment configurations against potential node failures, the system minimizes cascading service disruptions, addressing failure propagation concerns identified in microservice systems (Zhou et al., 2018).

Third, QoS-aware multi-resource management improves resource utilization without compromising latency targets, consistent with multi-resource management findings (Chen et al., 2019). Latency-aware accelerator scheduling further ensures efficient hardware utilization (Zhang et al., 2019).

Fourth, anomaly detection accuracy improves through ensemble methods combining log-based SVM approaches and Random Forest classifiers. Oversampling techniques enhance detection sensitivity for rare failure events (Gonzalez-Cuautle et al., 2020). Early failure prediction reduces mean time to recovery, aligning with log-based predictive models (Fronza et al., 2013).

Finally, Bayesian optimization contributes to continuous performance improvement by systematically tuning orchestration parameters (Brochu et al., 2010). Adaptive tuning enhances resilience under evolving workload patterns.

DISCUSSION

The proposed framework demonstrates theoretical coherence across multiple research domains. Its primary contribution lies in integration rather than isolated innovation. By embedding anomaly detection within orchestration loops, the system transcends reactive monitoring and achieves proactive adaptation.

However, limitations exist. The conceptual evaluation lacks empirical benchmarking against real-world deployments. While grounded in established literature, practical implementation would require validation under diverse workload conditions. Additionally, Bayesian optimization introduces computational overhead that may conflict with strict latency constraints if not carefully managed.

Future research should explore empirical validation using benchmark microservice systems and real mixed reality workloads. Further investigation into reinforcement learning-based orchestration may complement Bayesian approaches. Security considerations, including adversarial robustness of anomaly detection models, warrant deeper analysis.

CONCLUSION

This study presents a comprehensive adaptive orchestration framework for latency-sensitive microservice applications across cloud–edge environments. By synthesizing research on cloudlet placement, robustness-oriented deployment, QoS-aware resource management, decentralized architectures, hierarchical monitoring, anomaly detection, and Bayesian optimization, it establishes a unified blueprint for resilient and adaptive distributed systems. The framework addresses the stringent requirements of mixed reality, IoT, and time-critical applications, offering a theoretically grounded foundation for future empirical implementation and evaluation.

REFERENCES

1. Agrawal, S.; Agrawal, J. Survey on anomaly detection using data mining techniques. Procedia Computer Science, 2015, 60, 708–713.

2. Brochu, E.; Cora, V. M.; De Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv, 2010, arXiv:1012.2599.

3. Chen, Q.; Wang, Z.; Leng, J.; Li, C.; Zheng, W.; Guo, M. Avalon: Towards QoS awareness and improved utilization through multi-resource management in datacenters. Proceedings of the ACM International Conference on Supercomputing, 2019, 272–283.

4. Eltanbouly, S.; Bashendy, M.; AlNaimi, N.; Chkirbene, Z.; Erbad, A. Machine learning techniques for network anomaly detection: A survey. Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, 2020, 156–162.

5. Filip, I.-D.; Pop, F.; Serbanescu, C.; Choi, C. Microservices scheduling model over heterogeneous cloud-edge environments as support for IoT applications. IEEE Internet of Things Journal, 2018, 5(4), 2672–2681.

6. Fronza, I.; Sillitti, A.; Succi, G.; Terho, M.; Vlasenko, J. Failure prediction based on log files using random indexing and support vector machines. Journal of Systems and Software, 2013, 86, 2–11.

7. Gonzalez-Cuautle, D.; Hernandez-Suarez, A.; Sanchez-Perez, G.; Toscano-Medina, L. K.; Portillo-Portillo, J.; Olivares-Mercado, J.; Perez-Meana, H. M.; Sandoval-Orozco, A. L. Synthetic minority oversampling technique for optimizing classification tasks in botnet and intrusion-detection-system datasets. Applied Sciences, 2020, 10, 794.

8. K. S. Hebbar, "MACHINE LEARNING-ASSISTED SERVICE BOUNDARY DETECTION FOR MODULARIZING LEGACY SYSTEMS," International Journal of Applied Engineering & Technology, vol. 04,no.02, pp. 401-414, Sep. 2022, https://romanpub.com/resources/ijaet-v4-2-2022-48.pdf

9. Jia, M.; Cao, J.; Liang, W. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. IEEE Transactions on Cloud Computing, 2015, 5(4), 725–737.

10. Li, B.; He, Q.; Cui, G.; Xia, X.; Chen, F.; Jin, H.; Yang, Y. READ: Robustness-oriented edge application deployment in edge computing environment. IEEE Transactions on Services Computing, 2020, 15(3), 1746–1759.

11. Lu, L.; Wang, H.; Liu, P.; Liu, R.; Zhang, J.; Xie, Y.; Liu, S.; Huo, T.; Xie, M.; Wu, X. Applications of mixed reality technology in orthopedics surgery: A pilot study. Frontiers in Bioengineering and Biotechnology, 2022, 10.

12. Primartha, R.; Tama, B. A. Anomaly detection using random forest: A performance revisited. Proceedings of the 2017 International Conference on Data and Software Engineering, 2017, 1–6.

13. Rauschnabel, P. A. Augmented reality is eating the real-world! The substitution of physical products by holograms. International Journal of Information Management, 2021, 57, 102279.

14. Stefanič, P.; Cigale, M.; Jones, A. C.; Knight, L.; Taylor, I.; Istrate, C.; Suciu, G.; Ulisses, A.; Stankovski, V.; Taherizadeh, S.; Salado, G. F.; Koulouzis, S.; Martin, P.; Zhao, Z. SWITCH workbench: A novel approach for the development and deployment of time-critical microservice-based cloud-native applications. Future Generation Computer Systems, 2019, 99, 197–212.

15. Tefera, G.; She, K.; Deeba, F. Decentralized adaptive latency-aware cloud-edge-dew architecture for unreliable network. Proceedings of the 2019 11th International Conference on Machine Learning and Computing, 2019, 142–146.

16. Wang, R.; Ying, S.; Li, M.; Jia, S. HSACMA: A hierarchical scalable adaptive cloud monitoring architecture. Software Quality Journal, 2020, 28(3), 1379–1410.

17. Zhang, W.; Cui, W.; Fu, K.; Chen, Q.; Mawhirter, D. E.; Wu, B.; Li, C.; Guo, M. Laius: Towards latency awareness and improved utilization of spatial multitasking accelerators in datacenters. Proceedings of the ACM International Conference on Supercomputing, 2019, 58–68.

18. Zhou, X.; Peng, X.; Xie, T.; Sun, J.; Ji, C.; Li, W.; Ding, D. Fault analysis and debugging of microservice systems: Industrial survey, benchmark system, and empirical study. IEEE Transactions on Software Engineering, 2018, 47(2), 243–260.