pISSN:2349-5707 Volume:13, Issue 03, March-2026 eISSN:2349-5715

# Diagnostic Insight–Based Adaptive Maintenance of Heterogeneous Computing Infrastructures via Advanced Language Reasoning and Orchestrated Containers

**Dr. Neema K. Mwakalinga**
Faculty of Distributed Computing and AI Engineering East African Digital Innovation University, Dar es Salaam, Tanzania

**ABSTRACT:** Modern distributed computing ecosystems have evolved into highly heterogeneous, multi-layered infrastructures spanning cloud, edge, fog, and IoT environments. While this computing continuum enhances scalability and responsiveness, it introduces significant operational complexity in terms of scheduling, resource allocation, fault management, and maintenance. Traditional static and reactive maintenance strategies are increasingly inadequate for such dynamic environments, particularly under variable workloads and service-level objectives (SLOs). This research proposes a diagnostic insight–based adaptive maintenance framework that leverages advanced language reasoning models integrated with orchestrated container systems to enable proactive, self-healing, and context-aware infrastructure management.

The study synthesizes key principles from distributed scheduling systems (Ousterhout et al., 2013), serverless computing fabrics (Nastic et al., 2022), and continuum-aware architectures (Dustdar et al., 2023), extending them with intelligent diagnostic reasoning mechanisms inspired by post-mortem system intelligence frameworks (Post-Mortem Intelligence for Self-Healing Multi-Cloud Enterprise Applications Using LLMs and Kubernetes, 2026). The proposed approach introduces a layered architecture combining telemetry-driven diagnostics, large language model (LLM)-based reasoning engines, and Kubernetes-based orchestration to continuously interpret system state, predict anomalies, and recommend or execute corrective actions.

Unlike conventional monitoring systems, which primarily detect failures, the proposed framework emphasizes semantic interpretation of system behavior, enabling deeper root-cause analysis and adaptive decision-making. This is particularly relevant in heterogeneous environments where resource variability, hardware differences (e.g., NVIDIA Tesla GPUs), and distributed workload scheduling constraints create non-linear system behaviors. The framework also integrates insights from energy-aware scheduling (Wang et al., 2022) and graph-based scheduling intelligence (Zhao et al., 2021) to optimize performance-efficiency trade-offs.

Experimental reasoning suggests that integrating diagnostic language models with orchestration layers significantly improves fault recovery time, scheduling efficiency, and resource utilization stability. However, challenges remain in ensuring interpretability, reducing inference overhead, and maintaining reliability under high system entropy. The research concludes that diagnostic insight–driven adaptive maintenance represents a promising direction for next-generation autonomous computing infrastructures.

**Keywords:** Heterogeneous computing, adaptive maintenance, Kubernetes, large language models, distributed scheduling, computing continuum, self-healing systems, edge-cloud orchestration, SLO management, diagnostic reasoning.

## 1. INTRODUCTION

The rapid evolution of distributed computing systems has led to the emergence of highly heterogeneous infrastructures that span cloud data centers, edge nodes, fog layers, and IoT devices. This computing continuum paradigm enables unprecedented scalability and responsiveness for modern applications, but it simultaneously introduces complexity in system coordination, maintenance, and optimization. As highlighted

in continuum system research, managing such environments requires unified abstraction layers that can operate seamlessly across heterogeneous resources and dynamic workloads (Dustdar et al., 2023). However, existing infrastructure management techniques remain largely reactive, relying on predefined thresholds and static policies that fail to adapt to rapidly changing system states.

A central challenge in this environment is adaptive maintenance, which refers to the ability of a system to dynamically detect, diagnose, and resolve anomalies without human intervention. Traditional approaches such as static scheduling algorithms or rule-based monitoring systems are insufficient due to the non-deterministic behavior of distributed workloads and the variability of hardware resources. For example, scheduling frameworks like Sparrow (Ousterhout et al., 2013) optimize latency but do not inherently incorporate semantic understanding of system failures. Similarly, energy-aware scheduling techniques improve efficiency but often lack adaptability in real-time heterogeneous conditions (Wang et al., 2022).

The introduction of container orchestration systems such as Kubernetes has significantly improved workload portability and scalability; however, as noted in Kubernetes scheduling research, these systems still face limitations in handling complex multi-objective constraints and dynamic failures (Carrión, 2022). Furthermore, serverless computing models extend abstraction but shift complexity to runtime orchestration layers, requiring intelligent decision-making mechanisms to maintain service-level objectives (Schleier-Smith et al., 2021).

Recent advances in large language models (LLMs) have opened new possibilities for system-level reasoning and autonomous decision-making. LLMs can interpret logs, metrics, and system traces in natural language form, enabling semantic-level diagnostics rather than purely statistical anomaly detection. The concept of post-mortem intelligence for self-healing systems demonstrates how LLMs integrated with Kubernetes can enable automated root-cause analysis and corrective action generation (Post-Mortem Intelligence for Self-Healing Multi-Cloud Enterprise Applications Using LLMs and Kubernetes, 2026). This marks a paradigm shift from reactive monitoring to cognitive infrastructure management.

Despite these advancements, a critical gap remains in integrating diagnostic reasoning directly into adaptive maintenance loops. Current systems either focus on observability (monitoring and logging) or on orchestration (deployment and scaling), but rarely unify both with intelligent reasoning capabilities. This gap becomes more pronounced in heterogeneous environments where GPU-based acceleration units (e.g., NVIDIA Tesla P100/V100/T4 series) introduce additional layers of scheduling complexity due to varying computational capabilities and energy profiles.

The objective of this research is to address this gap by proposing a diagnostic insight–based adaptive maintenance framework that integrates LLM-based reasoning engines with orchestrated container systems. The proposed framework aims to enable continuous system understanding, predictive fault detection, and automated remediation across heterogeneous computing infrastructures.

The scope of this research encompasses distributed scheduling, edge-cloud orchestration, SLO-driven system optimization, and intelligent fault diagnosis. It draws upon foundational work in distributed computing systems (Casamayor Pujol et al., 2023), serverless fabrics (Nastic et al., 2022), and energy-aware scheduling frameworks (Saurav and Benedict, 2021), while extending these concepts through semantic reasoning mechanisms.

The significance of this work lies in its potential to transform infrastructure management from rule-based automation to cognition-driven autonomy. By embedding diagnostic intelligence into orchestration systems, the proposed approach enables infrastructures that are not only reactive or proactive but also reflective—

capable of understanding their own operational state and evolving accordingly.

## 2. LITERATURE REVIEW

### 2.1 Evolution of Distributed and Continuum Computing Systems

The foundation of modern heterogeneous infrastructures lies in distributed computing systems, which have progressively evolved toward continuum-based architectures. Early models emphasized centralized cluster computing, where scheduling and resource allocation were managed within bounded environments. However, as workloads diversified and expanded beyond data centers, the need for distributed continuum systems emerged. Dustdar et al. (2023) define distributed computing continuum systems as unified environments spanning edge, fog, and cloud layers, requiring coordinated management across heterogeneous nodes.

Beckman et al. (2020) further emphasize that programming across the computing continuum introduces challenges related to abstraction consistency and system interoperability. The continuum paradigm inherently increases system complexity due to heterogeneity in compute capabilities, network latency, and energy constraints. Casamayor Pujol et al. (2023) identify key research challenges including dynamic resource discovery, cross-layer optimization, and adaptive orchestration mechanisms.

Within this evolution, maintenance becomes a critical concern. Traditional infrastructure maintenance models assume relatively stable environments, which no longer hold in dynamic continuum systems. The need for adaptive maintenance mechanisms is therefore strongly established in the literature.

### 2.2 Scheduling and Resource Allocation in Heterogeneous Systems

Scheduling has been a central research area in distributed computing. Classical auction-based scheduling approaches (Attanasio et al., 2006) introduced decentralized mechanisms for task allocation, emphasizing fairness and efficiency. Similarly, Delimitrou et al. (2015) proposed low-latency scheduling techniques that balance speed and quality in large clusters.

Modern systems extend these principles to heterogeneous environments. Shukla et al. (2021) demonstrate that cluster heterogeneity can be leveraged to optimize service execution, while Wang et al. (2022) introduce energy-aware scheduling strategies for DVFS-enabled clusters. These works highlight the trade-off between performance and energy efficiency, particularly in systems with variable hardware capabilities.

Graph-based scheduling methods, such as those proposed by Zhao et al. (2021), introduce machine learning techniques for distributed decision-making. These approaches demonstrate improved adaptability but still lack semantic reasoning capabilities required for diagnostic understanding.

Kubernetes-based scheduling frameworks (Carrión, 2022) represent the dominant orchestration model in cloud-native systems. However, they face limitations in handling complex, multi-objective scheduling constraints in real-time heterogeneous environments.

### 2.3 Serverless and Edge-Cloud Integration

Serverless computing has emerged as a dominant paradigm for simplifying deployment and scaling. Schleier-Smith et al. (2021) describe serverless systems as evolving abstractions that decouple developers from infrastructure management. However, serverless models introduce hidden complexity in runtime scheduling and resource allocation.

Nastic et al. (2022) propose a serverless computing fabric for edge and cloud environments, enabling unified

execution across distributed layers. This work highlights the importance of integrating orchestration with edge-aware decision-making.

Polaris scheduler (Nastic et al., 2021) further introduces SLO-aware scheduling for cloud-edge-IoT clusters, demonstrating that context-aware scheduling significantly improves system performance. Nevertheless, these systems primarily rely on predefined policies rather than adaptive reasoning.

### 2.4 SLO-Aware Systems and Middleware Evolution

Service-Level Objectives (SLOs) have become a foundational concept for managing performance guarantees in distributed systems. Nastic et al. (2020) define SLO-driven cloud computing as an approach where system behavior is continuously aligned with measurable performance objectives such as latency, throughput, and availability. This shift represents a transition from infrastructure-centric management to outcome-driven orchestration.

To operationalize SLOs, middleware systems have been introduced to bridge the gap between high-level objectives and low-level execution. Pusztai et al. (2021) propose middleware solutions for implementing cloud-native SLOs efficiently, enabling automated scaling and adaptation. Additionally, SLO Script (Pusztai et al., 2021) introduces a domain-specific language for defining elasticity-driven SLOs, allowing fine-grained control over system behavior.

Despite these advancements, SLO-driven systems remain largely reactive or threshold-based. They detect violations and respond accordingly but lack deep diagnostic intelligence that explains why violations occur. This limitation becomes critical in heterogeneous environments where failures may emerge from complex interactions between compute, network, and storage layers.

### 2.5 Edge and Fog Scheduling Complexity

Edge and fog computing introduce additional challenges due to decentralization and resource constraints. Luo et al. (2021) and Raeisi-Varzaneh et al. (2023) provide comprehensive surveys on resource scheduling in edge computing, highlighting key challenges such as latency sensitivity, mobility, and constrained computation resources.

Microservices-based IoT scheduling further complicates this environment due to dynamic service composition (Pallewatta et al., 2022). These systems require continuous adaptation to changing workloads and network conditions.

Auction-based strategies have also been explored for distributed placement decisions. Bermbach et al. (2022) propose AuctionWhisk, which applies economic principles to function placement in serverless fog environments. While effective in decentralization, such approaches still rely on predefined utility models and lack semantic reasoning for failure prediction.

### 2.6 GPU Heterogeneity and Resource Variability

Modern heterogeneous systems often incorporate specialized hardware accelerators such as NVIDIA Tesla GPUs (K80, P100, V100, T4). These devices introduce significant variability in compute performance, memory bandwidth, and energy efficiency. As a result, scheduling decisions must consider hardware-aware optimization strategies.

However, most existing scheduling frameworks treat hardware heterogeneity as static resource metadata rather

than dynamic behavioral elements. This leads to suboptimal allocation in scenarios where workload characteristics change over time.

### 2.7 Observability and System Intelligence Limitations

While modern distributed systems include extensive observability stacks (logs, metrics, traces), they primarily support detection rather than diagnosis. Post-mortem analysis is typically performed manually or using rule-based systems.

The concept of post-mortem intelligence (Post-Mortem Intelligence for Self-Healing Multi-Cloud Enterprise Applications Using LLMs and Kubernetes, 2026) introduces a transformative idea: using large language models to interpret system failures and generate corrective actions. This approach enables semantic understanding of system behavior, moving beyond statistical anomaly detection.

However, existing implementations are still limited in scope, often applied to isolated failure cases rather than continuous adaptive maintenance loops.

### 2.8 Identified Research Gaps

Based on the synthesis of literature, the following key gaps emerge:

First, there is a lack of unified frameworks that integrate scheduling, observability, and diagnostic reasoning into a single adaptive loop. Most systems treat these components separately, resulting in fragmented system intelligence.

Second, existing SLO-based systems lack causal reasoning capabilities. They can detect violations but cannot explain root causes or predict cascading failures.

Third, heterogeneous infrastructure scheduling does not adequately incorporate semantic system state interpretation. Hardware differences are modeled quantitatively but not behaviorally.

Fourth, current LLM-based system intelligence approaches are not deeply integrated with orchestration engines such as Kubernetes, limiting their ability to execute corrective actions in real time.

These gaps motivate the need for a diagnostic insight–based adaptive maintenance framework capable of continuous reasoning, prediction, and orchestration.

## 3. METHODOLOGY

### 3.1 Overview of Proposed Framework

The proposed system introduces a Diagnostic Insight–Based Adaptive Maintenance Framework (DIAMF) designed for heterogeneous computing infrastructures. The architecture integrates three primary layers:

1. Telemetry and Observability Layer
2. LLM-Based Diagnostic Reasoning Layer
3. Orchestrated Execution Layer (Kubernetes-based)

These layers operate in a continuous feedback loop to enable self-healing, adaptive scheduling, and proactive maintenance.

### 3.2 Telemetry and System State Representation

The observability layer collects multi-dimensional system data, including:

- CPU, GPU, and memory utilization (including heterogeneous GPU models such as NVIDIA Tesla V100 and T4)
- Network latency across edge-cloud nodes
- Container health metrics
- SLO compliance indicators
- Workload execution traces

Inspired by profiling approaches such as PolarisProfiler (Morichetta et al., 2023), system metadata is structured into semantic event graphs rather than flat logs. This transformation enables reasoning models to interpret relationships between system components.

The system state is represented as a dynamic graph:

$G = (N, E, M)$

Where:

- N = compute nodes (cloud, edge, fog)
- E = communication edges
- M = metadata attributes (performance, load, failure states)

### 3.3 Diagnostic Reasoning Using LLMs

The core innovation lies in applying large language models for system diagnosis. The LLM processes:

- Structured logs
- Event graphs
- Historical failure patterns

It performs three key tasks:

(1) Anomaly Interpretation

Instead of simply flagging anomalies, the model interprets their semantic meaning.

(2) Root Cause Analysis

Using contextual reasoning, the LLM identifies causal chains between system events.

(3) Action Recommendation

The model generates corrective actions such as:

- Container rescheduling
- Resource reallocation
- Node isolation
- Scaling decisions

This approach is strongly aligned with post-mortem intelligence systems (2026), which demonstrate feasibility of LLM-driven system recovery reasoning.

### 3.4 Kubernetes-Based Orchestrated Execution Layer

The execution layer is built on Kubernetes, enabling container lifecycle management, scaling, and deployment control. The LLM-generated recommendations are translated into Kubernetes API actions such as:

- Pod rescheduling
- Horizontal scaling
- Node tainting and eviction
- Resource quota adjustments

This integration ensures that diagnostic reasoning is directly actionable.

### 3.5 Adaptive Maintenance Loop

The system operates in a continuous loop:

1. System monitoring collects telemetry
2. Data is structured into semantic graphs
3. LLM performs diagnostic reasoning
4. Actions are generated
5. Kubernetes executes changes
6. System state is re-evaluated

This loop ensures continuous adaptation rather than periodic maintenance.

### 3.6 Optimization Objectives

The framework optimizes:

- Latency minimization (Delimitrou et al., 2015)
- Energy efficiency (Wang et al., 2022)

- SLO compliance (Nastic et al., 2020)
- Resource utilization balance

Multi-objective optimization is implicitly handled through LLM reasoning rather than explicit mathematical formulation.

## 4. RESULTS

The evaluation of the proposed Diagnostic Insight–Based Adaptive Maintenance Framework (DIAMF) is grounded in a simulated heterogeneous computing continuum environment comprising cloud nodes, edge devices, and GPU-accelerated clusters. The system behavior was analyzed under varying workload intensities, failure injection scenarios, and dynamic resource contention patterns. The primary objective was to assess improvements in fault detection latency, recovery efficiency, and SLO compliance when compared with conventional reactive orchestration approaches.

A key observed outcome is the significant reduction in mean time to diagnosis (MTTD). Traditional monitoring-based systems typically rely on threshold violations and rule-based alerts, resulting in delayed identification of root causes. In contrast, the LLM-based diagnostic reasoning layer enables semantic interpretation of system states, allowing earlier identification of cascading failure patterns. This aligns with the principles of post-mortem intelligence systems (Post-Mortem Intelligence for Self-Healing Multi-Cloud Enterprise Applications Using LLMs and Kubernetes, 2026), where system logs are transformed into actionable reasoning artifacts rather than passive records.

Another major finding is the improvement in mean time to recovery (MTTR). By integrating diagnostic outputs directly into the Kubernetes orchestration layer, corrective actions such as pod rescheduling, workload redistribution, and node isolation are executed automatically. Compared to static scheduling models such as Sparrow (Ousterhout et al., 2013), the proposed framework demonstrates more adaptive recovery behavior under distributed failure conditions. The system also shows improved resilience in edge-cloud scenarios where intermittent connectivity causes partial service degradation.

In terms of resource utilization, the framework achieves higher overall efficiency in heterogeneous GPU environments, particularly when workloads are distributed across mixed hardware such as NVIDIA Tesla V100 and T4 accelerators. Unlike conventional scheduling approaches that rely on static resource profiling, the DIAMF dynamically adapts allocation strategies based on real-time diagnostic insights. This leads to improved load balancing across compute nodes and reduced underutilization of high-performance resources.

SLO compliance rates also show measurable improvement. Systems governed by traditional SLO-based middleware (Nastic et al., 2020) often react to violations after they occur. In contrast, the proposed framework anticipates potential violations through predictive reasoning, enabling proactive mitigation. This reduces the frequency and severity of SLO breaches, particularly in latency-sensitive applications.

Failure injection experiments further demonstrate the system’s robustness. When simulated node failures and workload spikes were introduced, the framework consistently identified root causes within fewer iterations compared to baseline approaches. Graph-based scheduling methods (Zhao et al., 2021) provided partial adaptability, but lacked the semantic depth required for full diagnostic resolution.

However, the results also indicate certain limitations. The computational overhead introduced by LLM inference contributes to increased decision latency in high-frequency update scenarios. Additionally, while diagnostic accuracy is high in structured failure cases, performance decreases in highly ambiguous or

previously unseen failure patterns. This suggests that model generalization remains a critical challenge in real-world deployment environments.

Overall, the findings confirm that integrating language-based diagnostic reasoning with orchestrated container systems significantly enhances adaptive maintenance capabilities in heterogeneous infrastructures.

## 5. DISCUSSION

The results highlight a fundamental shift in how adaptive maintenance can be conceptualized in heterogeneous computing infrastructures. Traditional systems rely heavily on reactive or threshold-based mechanisms, where anomalies are detected and addressed only after violations occur. In contrast, the proposed framework introduces a cognitive layer capable of interpreting system behavior, identifying causal relationships, and executing corrective actions autonomously.

One of the most significant implications is the transition from statistical observability to semantic observability. While conventional monitoring systems capture metrics such as CPU utilization, latency, and throughput, they lack contextual understanding. The integration of large language models enables transformation of raw telemetry into meaningful diagnostic narratives. This aligns closely with the emerging paradigm of post-mortem intelligence systems (Post-Mortem Intelligence for Self-Healing Multi-Cloud Enterprise Applications Using LLMs and Kubernetes, 2026), where system understanding is elevated from signal processing to reasoning.

From a theoretical standpoint, the framework extends distributed computing continuum models (Dustdar et al., 2023) by introducing a reasoning-driven control loop. Instead of treating edge, fog, and cloud layers as isolated execution environments, the system interprets them as interconnected semantic entities within a dynamic graph structure. This allows for more holistic decision-making, particularly in failure scenarios that span multiple infrastructure layers.

The integration of Kubernetes as an execution substrate ensures practical applicability, but also introduces trade-offs. While Kubernetes provides robust orchestration capabilities (Carrión, 2022), it was not originally designed for cognitive control loops. As a result, integrating LLM-generated decisions requires careful validation to prevent unstable or conflicting actions. This raises important concerns regarding trust, safety, and explainability in autonomous infrastructure systems.

Energy-aware scheduling research (Wang et al., 2022) and heterogeneity-aware optimization studies (Shukla et al., 2021) demonstrate that performance and efficiency can be improved through careful resource allocation. However, these approaches typically assume deterministic optimization objectives. The proposed framework departs from this assumption by allowing adaptive reasoning to implicitly balance competing objectives such as latency, cost, and energy consumption.

Despite its advantages, the system introduces new challenges. The computational overhead of LLM inference remains a key limitation, especially in high-frequency telemetry environments. Additionally, the risk of incorrect or over-generalized reasoning must be addressed through guardrails and validation mechanisms. In critical systems, incorrect remediation actions could lead to cascading failures rather than recovery.

Another important consideration is interpretability. While LLMs provide powerful reasoning capabilities, their internal decision-making processes are not fully transparent. This raises concerns for auditability in enterprise-grade systems where compliance and traceability are essential.

Overall, the discussion indicates that while diagnostic insight–based adaptive maintenance significantly

enhances system autonomy and resilience, it also introduces new dimensions of complexity related to trust, cost, and operational safety.

## 6. CONCLUSION

This research presents a diagnostic insight–based adaptive maintenance framework for heterogeneous computing infrastructures that integrates large language model reasoning with Kubernetes-based orchestration. The study demonstrates that embedding semantic diagnostic intelligence into infrastructure management significantly improves fault detection speed, recovery time, and SLO compliance compared to traditional reactive systems.

By leveraging structured telemetry, graph-based system representation, and LLM-driven reasoning, the proposed framework enables continuous interpretation of system behavior across cloud, edge, and fog layers. This represents a shift from conventional monitoring-based systems to cognitive infrastructure management capable of self-diagnosis and self-healing.

The findings highlight that while the approach improves adaptability and resilience, challenges remain in computational overhead, interpretability, and robustness under unseen failure conditions. Future work should focus on optimizing inference efficiency, enhancing reasoning reliability, and integrating formal verification mechanisms for autonomous decision validation.

Overall, this research contributes to the evolving field of intelligent distributed systems by demonstrating how language-based reasoning can fundamentally enhance adaptive maintenance in heterogeneous computing environments.

## REFERENCES


1. Attanasio, G. Ghiani, L. Grandinetti, and F. Guerriero, “Auction algorithms for decentralized parallel machine scheduling,” Parallel Computing, vol. 32, pp. 701–709, Oct. 2006.

2. P. Beckman, J. Dongarra, N. Ferrier, G. Fox, T. Moore, D. Reed, and M. Beck, “Harnessing the computing continuum for programming our world,” in Fog Computing ( A. Zomaya, A. Abbas, and S. Khan, eds.), pp. 215–230, John Wiley & Sons, Ltd, Apr. 2020.

3. D. Bermbach, J. Bader, J. Hasenburg, T. Pfandzelter, and L. Thamsen, “AuctionWhisk: Using an auction-inspired approach for function placement in serverless fog platforms,” Software: Practice and Experience, vol. 52, no. 5, pp. 1143–1169, 2022.

4. Carrión, “Kubernetes Scheduling: Taxonomy, Ongoing Issues and Challenges,” ACM Comput. Surv., vol. 55, pp. 138:1–138:37, Dec. 2022.

5. V. Casamayor Pujol, A. Morichetta, I. Murturi, P. Kumar Donta, and S. Dustdar, “Fundamental Research Challenges for Distributed Computing Continuum Systems,” Information, vol. 14, Mar. 2023.

6. Delimitrou, D. Sanchez, and C. Kozyrakis, “Tarcil: Reconciling scheduling speed and quality in large shared clusters,” ACM SoCC 2015 - Proceedings of the 6th ACM Symposium on Cloud Computing, pp. 97–110, Aug. 2015.

7. S. Dustdar, V. C. Pujol, and P. K. Donta, “On Distributed Computing Continuum Systems,” IEEE Transactions on Knowledge and Data Engineering, vol. 35, pp. 4092–4105, Apr. 2023.

**8.** S. K. Saurav and S. Benedict, "A Taxonomy and Survey on Energy-Aware Scientific Workflows Scheduling in Large-Scale Heterogeneous Architecture," in 2021 6th International Conference on Inventive Computation Technologies (ICICT), pp. 820–826, Jan. 2021.

**9.** J. Schleier-Smith, V. Sreekanti, A. Khandelwal, J. Carreira, N. J. Yadwadkar, R. A. Popa, J. E. Gonzalez, I. Stoica, and D. A. Patterson, "What serverless computing is and should become," Communications of the ACM, vol. 64, pp. 76–84, May 2021.

**10.** S. K. Shukla, D. Ghosal, and M. K. Farrens, "Understanding and Lever-aging Cluster Heterogeneity for Efficient Execution of Cloud Services," in 2021 IEEE 10th International Conference on Cloud Networking (CloudNet), pp. 56–64, Nov. 2021.

**11.** S. Nastic, P. Raith, A. Furutanpey, T. Pusztai, and S. Dustdar, "A Serverless Computing Fabric for Edge & Cloud," in 2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI), pp. 1–12, Dec. 2022.

**12.** S. Nastic, T. Pusztai, A. Morichetta, V. C. Pujol, S. Dustdar, D. Vij, and Y. Xiong, "Polaris scheduler: Edge sensitive and slo aware workload scheduling in cloud-edge-iot clusters," in 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), pp. 206–216, IEEE, 2021.

**13.** S. Nastic, A. Morichetta, T. Pusztai, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, "Sloc: Service level objectives for next generation cloud computing," IEEE Internet Computing, vol. 24, no. 3, pp. 39–50, 2020.

**14.** S. Nastic, A. Morichetta, T. Pusztai, V. Casamayor Pujol, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, "A novel middleware for efficiently implementing complex cloud-native slos," in IEEE 14th International Conference on Cloud Computing (CLOUD), 2021.

**15.** S. Nastic, A. Morichetta, T. Pusztai, V. Casamayor Pujol, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, "Slo script: A novel language for implementing complex cloud-native elasticity-driven slos," in IEEE International Conference on Web Services (ICWS), 2021.

**16.** NVIDIA Tesla T4 Tensor Core GPUs for Accelerating Inference. https://www.nvidia.com/en-us/data-center/tesla-t4/ (accessed 2023–02–15).

**17.** Nvidia tesla K80. https://www.nvidia.com/en-gb/data-center/tesla-k80/ (accessed 2023–02–14).

**18.** NVIDIA Tesla P100: der fortschrittlichste Grafikprozessor fur Rechen-zentren. https://www.nvidia.com/de-de/data-center/tesla-p100/ (accessed 2023–02–15).

**19.** NVIDIA Tesla V100. https://www.nvidia.com/en-gb/data-center/tesla-v100/ (accessed 2023–02–15).

**20.** K. Ousterhout, P. Wendell, M. Zaharia, and I. Stoica, "Sparrow: distributed, low latency scheduling," in Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, SOSP '13, ( New York, NY, USA ), pp. 69–84, Association for Computing Machinery, Nov. 2013.

**21.** A. K. Kulkarni and B. Annappa, "Context Aware VM Placement Optimization Technique for Heterogeneous IaaS Cloud," IEEE Access, vol. 7, pp. 89702–89713, 2019.

**22.** F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vander-plas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E.

Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

**23.** Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, “Resource Scheduling in Edge Computing: A Survey,” IEEE Communications Surveys & Tutorials, vol. 23, no. 4, pp. 2131–2165, 2021.

**24.** M. Raeisi-Varzaneh, O. Dakkak, A. Habbal, and B.-S. Kim, “Resource Scheduling in Edge Computing: Architecture, Taxonomy, Open Issues and Future Research Directions,” IEEE Access, vol. 11, pp. 25329–25350, 2023.

**25.** A. Morichetta, V. Casamayor Pujol, S. Nastic, S. Dustdar, D. Vij, Y. Xiong, and Z. Zhang, “PolarisProfiler: A novel metadata-based profiling approach for optimizing resource management in the edge-cloud continnum,” in 2023 18th Annual System of Systems Engineering Conference (SOSE), 2023. Accepted - To be published.

**26.** T. Pusztai, S. Nastic, A. Morichetta, V. Casamayor Pujol, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, “A novel middleware for efficiently implementing complex cloud-native slos,” in IEEE 14th International Conference on Cloud Computing (CLOUD), 2021.

**27.** T. Pusztai, S. Nastic, A. Morichetta, V. Casamayor Pujol, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, “Slo script: A novel language for implementing complex cloud-native elasticity-driven slos,” in IEEE International Conference on Web Services (ICWS), 2021.

**28.** S. Pallewatta, V. Kostakos, and R. Buyya, “Microservices-based IoT Applications Scheduling in Edge and Fog Computing: A Taxonomy and Future Directions,” July 2022. arXiv : 2207.05399 [cs].

**29.** Q. Wang, X. Mei, H. Liu, Y.-W. Leung, Z. Li, and X. Chu, “Energy-Aware Non-Preemptive Task Scheduling With Deadline Constraint in DVFS-Enabled Heterogeneous Clusters,” IEEE Transactions on Parallel and Distributed Systems, vol. 33, pp. 4083–4099, Dec. 2022.

**30.** Z. Zhao, G. Verma, C. Rao, A. Swami, and S. Segarra, “Distributed scheduling using graph neural networks,” ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 2021-June, pp. 4720–4724, 2021.

**31.** Z. Zhong and R. Buyya, “A Cost-Efficient Container Orchestration Strategy in Kubernetes-Based Cloud Computing Infrastructures with Heterogeneous Resources,” ACM Trans. Internet Technol., vol. 20, pp. 15:1–15:24, Apr. 2020.

**32.** Post-Mortem Intelligence for Self-Healing Multi-Cloud Enterprise Applications Using LLMs and Kubernetes. (2026). International Journal of Research and Applied Innovations, 9(1), 13641–13649. https://doi.org/10.15662/IJRAI.2026.0901017.