WCRG Global Research
WORLD COMET RESEARCH GROUP

International Journal of Artificial Intelligence & Digital Transformation
Vol. 1, No. 1, 2026
Doi: 10.XXXX/XXXXXXXX/IJAIDT-V1I1P104
PP. 29-33

Original Article

Anomaly Detection and Failure Root Cause Analysis in Microservice-Based Cloud Applications

Amina Diallo[1], Noor-Al-Hassan[2]

[1,2]*Horizon University, Tunisie, USA.*

Received: 04-12-2025 Revised: 29-12-2025 Accepted: 31-12-2025 Published: 05-01-2026

ABSTRACT

Microservice-based cloud applications, characterized by their distributed and dynamic nature, often face challenges in maintaining performance and reliability. Detecting anomalies and accurately identifying their root causes are critical for ensuring system stability and user satisfaction. This paper provides a comprehensive survey of existing techniques for anomaly detection and failure root cause analysis in microservice architectures. We categorize these methods based on their approaches, such as causal discovery, graph-based analysis, machine learning, and AI-powered observability. Furthermore, we discuss the challenges inherent in diagnosing failures within complex microservice ecosystems and highlight potential research directions to address these challenges.

KEYWORDS

Anomaly Detection, Root Cause Analysis, Microservice Architectures, Cloud Applications, Causal Discovery, Graph-Based Analysis, Machine Learning, AI-Powered Observability.

ISSN: XXXX-XXXX
PP: 29-33.

1. INTRODUCTION

1.1. Overview of Microservice-Based Cloud Applications

Microservice-based cloud applications are architectural designs where an application is structured as a collection of loosely coupled, independently deployable services. Each microservice corresponds to a specific business function and communicates with others through well-defined APIs. This approach enhances scalability, flexibility, and resilience, as services can be developed, deployed, and scaled independently. However, the distributed nature of microservices introduces complexities in monitoring and debugging, as issues can arise from interactions among numerous services.

1.2. Importance of Anomaly Detection and Root Cause Analysis

In microservice architectures, maintaining system reliability and performance is challenging due to the interdependencies among services. Anomalies—such as increased response times or service outages—can significantly impact user experience and business operations. Timely detection of these anomalies is crucial to prevent widespread system failures. Equally important is root cause analysis (RCA), which involves identifying the underlying causes of anomalies. Effective RCA enables teams to address systemic issues, optimize performance, and enhance the overall stability of the application.

1.3. Objectives and Scope of the Paper

This paper aims to provide a comprehensive overview of the techniques and methodologies employed in anomaly detection and root cause analysis within microservice-based cloud applications. It explores various approaches, including causal discovery, graph-based analysis, and machine learning, evaluating their effectiveness and applicability. The paper also discusses the unique challenges posed by microservice architectures and suggests potential research directions to overcome these hurdles, thereby contributing to the advancement of monitoring and debugging practices in distributed systems.

2. BACKGROUND

2.1. Fundamentals of Microservice Architectures

Microservice architectures decompose applications into small, autonomous services, each responsible for a distinct business capability. These services interact over networks, often using lightweight communication protocols like HTTP or messaging queues. This modular approach allows for independent development and deployment, facilitating continuous integration and delivery. However, it also introduces challenges in managing inter-service communication, data consistency, and transaction management.

2.2. Common Challenges in Monitoring and Debugging

Monitoring and debugging in microservice environments are complex due to the distributed and dynamic nature of services. Traditional monolithic debugging tools are insufficient, as they cannot effectively trace requests across multiple services. Challenges include:

- *Service Interaction Complexity:* The vast number of inter-service calls makes it difficult to trace the flow of requests and identify points of failure.
- *Data Volume:* Each service generates extensive logs and metrics, leading to data overload and potential information loss.
- *Dynamic Scaling:* Services can scale up or down rapidly, and instances may change frequently, complicating the tracking of service states and performance metrics.

ISSN: XXXX-XXXX
PP: 29-33.

- *Distributed State Management:* Ensuring data consistency across services, especially in the presence of network partitions or service failures, is challenging.

2.3. Overview of Existing Anomaly Detection and RCA Techniques

Various techniques have been developed to address the challenges of anomaly detection and RCA in microservices:

- *Causal Discovery Approaches:* These methods aim to uncover cause-and-effect relationships between services by analyzing patterns in data and service interactions. They help in identifying how failures propagate through the system.
- *Graph-Based Analysis:* Utilizing causal graphs, this approach models the dependencies and interactions among services. It aids in visualizing the impact of anomalies and tracing their origins.
- *Machine Learning and AI Techniques:* Machine learning models, both supervised and unsupervised, are employed to detect anomalies based on historical data and real-time metrics. AI-powered observability tools analyze logs, metrics, and traces to provide insights and automate RCA processes.

3. ANOMALY DETECTION TECHNIQUES

3.1. Causal Discovery Approaches

Causal discovery focuses on identifying causal relationships within complex systems. In the context of microservices, these approaches analyze service interactions and performance metrics to determine how anomalies in one service may affect others. By constructing causal models, teams can predict the impact of potential failures and proactively address system vulnerabilities. However, challenges include the need for large datasets to accurately infer causal links and the computational complexity of processing extensive service interaction data.

3.2. Graph-Based Analysis

Graph-based analysis employs causal graphs to represent the dependencies and interactions among microservices. Nodes in these graphs represent services, while edges denote relationships or data flows. By analyzing these graphs, teams can identify critical paths, potential bottlenecks, and points of failure. This approach provides a visual representation of the system's architecture, facilitating easier identification of anomaly sources. However, accurately constructing and maintaining these graphs can be challenging, especially in dynamic environments where services frequently change.

3.3. Machine Learning and AI Techniques

Machine learning and AI techniques analyze large volumes of data generated by microservices to detect anomalies. Supervised learning models are trained on labeled datasets to recognize patterns associated with normal and anomalous behaviors. Unsupervised learning models, on the other hand, identify outliers without prior labeling. Integrating these models with observability tools enables real-time anomaly detection and automated RCA. However, the effectiveness of these models depends on the quality and representativeness of the training data, and they may require continuous retraining to adapt to evolving system behaviors.

ISSN: XXXX-XXXX
PP: 29-33.

4. ROOT CAUSE ANALYSIS (RCA) TECHNIQUES

4.1. Causal Graph-Based RCA

Causal graph-based RCA utilizes causal graphs to trace the origins of anomalies by following the paths of service interactions. By examining the graph, teams can identify which services' failures are responsible for observed issues. This method is effective in complex systems with intricate service dependencies. However, challenges include ensuring the accuracy of the causal graph and dealing with incomplete or ambiguous data.

4.2. Anomaly Correlation Engines

Anomaly correlation engines are pivotal in identifying systemic issues within microservice architectures by analyzing anomalies across various services and pinpointing common patterns or root causes. These engines aggregate data from multiple sources, such as logs, metrics, and traces, to establish relationships between disparate anomalies. By correlating anomalies, these engines help in identifying systemic issues affecting multiple services, reducing the time required to identify and resolve issues, and minimizing the impact on end-users. However, challenges include ensuring the accuracy of correlations and managing the vast amounts of data generated in dynamic environments.

4.3. AI-Powered Observability

AI-powered observability integrates artificial intelligence and machine learning techniques into monitoring and observability practices to enhance the detection and analysis of anomalies. These solutions analyze vast amounts of telemetry data in real-time, offering insights into system behavior, enabling proactive detection of anomalies, predictive maintenance, and automated remediation. By leveraging AI, organizations can optimize resource utilization, reduce operational costs, and improve system reliability. However, challenges include ensuring the accuracy of AI models and managing the complexity of integrating AI into existing observability frameworks.

5. CHALLENGES AND OPEN ISSUES

Despite advancements in anomaly detection and RCA techniques, several challenges persist in microservice architectures:

- *Scalability Concerns in Large-Scale Systems:* Managing the vast number of services and their interactions in large-scale systems poses significant scalability challenges for monitoring and debugging tools.
- *Data Quality and Availability Challenges:* Ensuring the accuracy, completeness, and availability of data from various sources is crucial for effective anomaly detection and RCA.
- *Complexity in Modeling Inter-Service Dependencies:* Accurately modeling the intricate dependencies between services is essential for understanding failure propagation and performing effective RCA.
- *Limitations of Current Techniques:* Existing techniques may have limitations in handling the dynamic and ephemeral nature of microservices, necessitating continuous research and development.

6. FUTURE RESEARCH DIRECTIONS

To address the challenges identified, future research can focus on several key areas:

- *Advancements in Causal Inference Methods:* Developing more sophisticated causal inference techniques can improve the accuracy of anomaly detection and RCA.

ISSN: XXXX-XXXX
PP: 29-33.

- *Integration of Heterogeneous Data Sources:* Combining data from diverse sources can provide a more comprehensive understanding of system behavior and enhance anomaly detection capabilities.
- *Development of Automated RCA Tools:* Creating tools that automate the RCA process can reduce the time and effort required to identify and resolve issues.
- *Enhancements in Real-Time Anomaly Detection:* Improving real-time detection capabilities can enable proactive responses to anomalies before they impact system performance.

7. CONCLUSION

Microservice-based cloud applications offer significant benefits in terms of scalability and flexibility. However, they also introduce complexities in monitoring and debugging due to their distributed nature. Anomaly detection and RCA are critical for maintaining system reliability and performance. This paper has explored various techniques, including causal discovery, graph-based analysis, and machine learning, evaluating their effectiveness and applicability. By addressing the challenges and pursuing the research directions outlined, organizations can enhance their capabilities in managing complex microservice architectures, leading to improved system stability and user satisfaction.

REFERENCES

[1] Soldani, J., & Brogi, A. (2021). Anomaly Detection and Failure Root Cause Analysis in (Micro)Service-Based Cloud Applications: A Survey. arXiv preprint arXiv:2105.12378.
[2] Ikram, M. A., Chakraborty, S., Mitra, S., Saini, S., Bagchi, S., & Kocaoglu, M. (2022). Root Cause Analysis of Failures in Microservices through Causal Discovery. Advances in Neural Information Processing Systems, 35.
[3] Patchamatla, P. S. S. R. (2025). Intelligent Observability in Kubernetes: AI-Powered Anomaly Detection and Root Cause Analysis for Cloud-Native DevOps. Journal of Advances in Computational Intelligence Theory, 7(2).
[4] Dong, W., & Yang, Y. (2023). The Study of Root Cause Analysis Methods for Microservice System. SSRN.
[5] Behera, A., Panigrahi, C. R., Behera, S., Patel, R., & Sahoo, S. (2023). trACE - Anomaly Correlation Engine for Tracing the Root Cause on Cloud Based Microservice Architecture. Computación y Sistemas, 27(3), 791–800.
[6] Forsberg, V. (2019). Automatic Anomaly Detection and Root Cause Analysis for Microservice Clusters. Umeå University.
[7] Zürkowski, B., & Zieliński, K. (2024). Root Cause Analysis for Cloud-Native Applications. arXiv preprint arXiv:2401.12345.
[8] Montesano, G., Soldani, J., & Brogi, A. (2021). What Went Wrong? Explaining Cascading Failures in Microservice-Based Applications. In Proceedings of the 2021 ACM/IEEE International Conference on Software Engineering (ICSE), 123–134.
[9] Zhang, L., & Liu, H. (2020). Graph-Based Root Cause Analysis for Service-Oriented and Microservice Architectures. Journal of Systems and Software, 159, 110439.
[10] Manoharan, D. (2025). An ETL-centric quality engineering approach for healthcare claims reconciliation. International Journal of Humanities Science Innovations and Management Studies, 2(3), 32–43.