Survey on auto elasticity techniques for microservices in cloud computing

Mary M Dsouza

Assistant Professor,ISE Department,AIT
Research Scholar,CSE Dept, *JSSATE*
Bangalore, India

marymdsouza@gmail.com

Dr.Nagamani N Purohit

HOD,CSE Dept, *JSSATE*
Bangalore, India

nagamaninpurohit@jssateb.ac.in

***Abstract*— Microservices have become a popular way to build scalable and flexible cloud-native applications. However, the rapid growth of cloud services, edge computing platforms, and container-based deployments has created big challenges in managing computing resources efficiently. The distributed and loosely connected nature of microservices makes it hard to choose the best service placement, allocate resources dynamically, and scale automatically while keeping performance and service level agreements (SLAs) intact. Recent research has looked into smart resource management methods that use artificial intelligence (AI), machine learning (ML), and meta-heuristic optimization techniques to support self-managing and adaptable system behavior. These methods aim to improve quality of service (QoS), lower operational costs and energy use, and allow flexible resource provisioning in cloud-edge environments. This paper offers a thorough survey of smart and adaptable resource management strategies for microservice-based cloud and edge systems. The study reviews current solutions related to auto-scaling, service placement and migration, predictive container orchestration, and multi-objective optimization frameworks based on the Monitor-Analyze-Plan-Execute (MAPE) model. Additionally, the paper points out existing research challenges and discusses potential paths for creating fully self-managing cloud-native resource management systems.**

Keywords— Cloud Computing, Microservices, Edge Computing, Resource Management, Auto-Scaling, Artificial Intelligence, Autonomic Systems

Introduction

The Microservices architecture has been widely accepted and has emerged as a popular solution for designing distributed and scalable cloud-native applications. Microservices architecture is different from traditional monolithic architecture, where applications are divided into small services, and these services communicate with each other using small interfaces. This architecture allows each microservice to be developed, deployed, and scaled independently, thus providing greater flexibility and continuous deployment support [1].

However, with the advent of microservice architecture, new challenges arise in managing resources. In a microservice architecture, applications are divided into small services, and each service is a container. Managing these services in a distributed environment is a complex task. As the use of cloud computing, edge computing, and containerization is becoming very popular, the management of computing resources in a distributed environment is becoming more complex. New applications, such as Internet of Things, smart healthcare, smart transportation, and mobile services, are emerging, and these applications require ultra-low latency and high reliability. With the advent of 5G networks and edge data centers, resource management is becoming a complex task. Therefore, new solutions for resource management are required in a distributed environment. Microservice architecture requires effective resource management solutions for effective application performance and to meet application requirements, such as service-level agreements. Therefore, effective solutions are required for resource management in a distributed environment. Threshold-based auto-scaling is a traditional solution for resource management, but it is becoming very difficult to handle unpredictable workloads in a distributed environment. Therefore, recent research has been focused on developing intelligent resource management frameworks, including the application of artificial intelligence, machine learning, reinforcement learning, and meta-heuristic optimization techniques. These techniques have the potential to predict resource workload, optimize resource allocation, and enable autonomous scaling using adaptive resource orchestration techniques, such as the MAPE model.

However, there are still some challenges, including cross-layer resource orchestration, energy-aware resource allocation, and edge-aware service migration. Therefore, a comprehensive survey on the application of intelligent and elastic resource management

strategies is required to understand recent developments and identify potential research areas in cloud-native microservice systems.

PROBLEM STATEMENT

The recent trend in developing modern cloud-native applications is based on a microservices architecture pattern implemented in a variety of cloud and edge infrastructures. Although the microservices-based architecture pattern ensures high scalability, modularity, and flexibility in developing applications, it poses various resource management and service orchestration problems.

The distributed nature of microservices in heterogeneous infrastructures involving various cloud servers, edge nodes, and container platforms complicates resource allocation and service orchestration problems. Traditionally, various resource allocation techniques have been used in microservices-based applications. However, these techniques are found ineffective in managing dynamic workloads.

Additionally, applications in a heterogeneous cloud-edge infrastructure need to meet various QoS constraints, including low-latency service provision, high service availability, and service provision in accordance with service level agreements. Inefficient resource allocation in a heterogeneous infrastructure may cause either over-provisioning of resources, resulting in increased operational costs and energy consumption, or under-provisioning of resources, leading to decreased performance.

Another challenge in developing applications in heterogeneous infrastructures arises from the integration of container orchestration platforms with various distributed edge infrastructures. Resource orchestration in a heterogeneous infrastructure involving various edge nodes and a central cloud controller requires sophisticated orchestration mechanisms capable of dealing with multiple optimization objectives.

Although various research studies have explored AI-based techniques in resource management in a heterogeneous infrastructure involving a variety of edge nodes and a central cloud controller, most of these techniques address only isolated problems in microservices-based applications. Hence, a unified framework capable of dealing with various problems in microservices-based applications using a single platform involving predictive workload modeling, intelligent orchestration, and elastic resource allocation techniques in a heterogeneous infrastructure involving a variety of edge nodes and a central cloud controller is still a research gap.

I. BACKGROUND AND MOTIVATION

Currently, new applications in the cloud environment are built using a microservices architecture paradigm in order to support scalability and deployment. In a microservices architecture paradigm, an application is divided into smaller services that can be built and deployed separately. It provides flexibility in continuous integration and continuous deployment.

With the advent of various cloud computing platforms, containerization using Docker and container orchestration using Kubernetes have greatly increased the use of microservices in building applications. These tools help developers in deploying applications in a distributed environment using a distributed computing infrastructure.

However, with an increase in the number of microservices in an application, it becomes more challenging to handle these services. Since each microservice may have different characteristics in terms of resource utilization, communication, and scalability, it becomes a challenge in terms of coordinating these resources and ensuring the consistent performance of these services.

In addition to the use of cloud computing platforms, edge computing has also emerged as an extension of the traditional cloud environment. It helps in reducing network latency in a distributed environment. It brings the resources of the infrastructure closer to the user or device in a network environment. It can be used in various applications such as Internet of Things (IoT), smart healthcare systems, autonomous vehicles, and next-generation wireless networks.

The use of microservices in a cloud-edge environment poses various challenges in terms of service placement, mobility-aware service migration, heterogeneous resource constraints, and energy-efficient scheduling. It requires an advanced resource management technique in order to adapt itself to a dynamic environment.

A. Autonomic and Elastic Resource Management

The ability of a system to perform self-management without human intervention is called autonomic computing. The system is developed with self-management characteristics like self-configuration, self-healing, self-optimization, and self-protection. The autonomic management system is important for cloud computing environments.

The second important feature of cloud computing is elasticity. The scalability of computer resources is achieved by means of elasticity. The resources are dynamically scaled according to the demand of the workload.

The scalability of microservice cloud computing resources occurs at various infrastructure levels:

• Container scaling: The resources are scaled by creating more copies of a container.

• Virtual machine scaling: The resources are scaled by creating more copies of a virtual machine.

• Edge node migration: The services are migrated closer to users.

• Service replication: The services are replicated.

The traditional approach of scaling resources using a rule is not effective in managing unpredictable workloads. The new research is focused on artificial intelligence and machine learning for predictive management

B. Motivation

The rising demand for low-latency and high-performance applications has created a significant challenge for resource management in cloud-edge environments. New technologies like 5G networks, Internet of Things (IoT) devices, and real-time data analytics require highly responsive and scalable computing resources.

Applications developed using microservice architectures running across distributed cloud-edge environments require effective management of resources. However, traditional resource management techniques are often unable to handle the dynamic nature of cloud-edge environments.

Therefore, there is a great need for developing effective resource management solutions that utilize intelligent and automated resource management techniques along with auto-scaling capabilities, service placement strategies, optimization techniques, and AI orchestration.

Using such resource management techniques is effective in managing quality of service (QoS), cost efficiency, and service level agreements (SLAs) for cloud-edge environments.

II. INTELLIGENT RESOURCE MANAGEMENT TAXONOMY

This section presents a structured taxonomy of intelligent and elastic resource management approaches used in cloud and edge-based microservice architectures. The taxonomy categorizes existing solutions based on architectural models, resource management functions, optimization objectives, and AI-driven strategies.

A. Architectural Classification

1. Centralized Resource Management
Centralized resource management relies on a global controller responsible for monitoring resource usage and allocating resources across the system. This approach can achieve globally optimized decisions but may face scalability limitations and potential single points of failure.

2. Distributed Resource Management
In distributed resource management systems, decision-making is performed across multiple nodes or clusters. This architecture improves scalability and fault tolerance, although it requires coordination mechanisms to maintain consistency across distributed resources.

3. Hybrid Architecture
Hybrid resource management combines centralized decision-making with decentralized execution. In this approach, high-level policies are determined by centralized controllers while local nodes implement resource management decisions. This architecture aims to balance scalability with efficient resource coordination.

B. Resource Management Functions

Resource management in microservice environments typically involves several key operations:

1. **Resource Provisioning**
   Dynamic allocation of computing resources such as CPU, memory, and storage based on workload demands.

2. **Resource Allocation**
   Mapping containers or services to suitable infrastructure nodes.

3. **Resource Scheduling**
   Optimizing execution order and service placement to satisfy performance and latency requirements.

4. **Resource Monitoring**
   Continuously observing system metrics to support adaptive resource scaling.

C. Optimization Objectives

Intelligent resource management techniques aim to optimize one or more of the following objectives:

• Quality of Service (QoS)
• Service Level Agreement (SLA) compliance
• Cost minimization
• Energy efficiency
• Latency reduction
• Load balancing

In many cases, **multi-objective optimization algorithms** are employed to balance conflicting requirements among these objectives.

D. AI-Driven Strategies

Recent research has increasingly explored the use of artificial intelligence techniques for intelligent resource management in cloud and edge environments. These approaches include:

• Reinforcement learning–based resource orchestration
• Deep learning models for workload prediction
• Transformer-based frameworks for implementing adaptive Monitor–Analyze–Plan–Execute (MAPE) loops
• Metaheuristic optimization algorithms for resource allocation
• Predictive container orchestration models designed to improve decision-making in highly dynamic environments

These AI-driven approaches enable systems to make more accurate and adaptive resource management decisions in complex distributed infrastructures.

IV. AI-BASED AUTO-SCALING TECHNIQUES

Auto-scaling techniques based on AI have been recently proposed as efficient solutions for handling fluctuating workloads in cloud and edge computing environments. Unlike traditional rule-based auto-scaling mechanisms, AI-based auto-scaling uses predictive analytics and ML algorithms to analyze workload behaviors and make decisions on resource allocation. AI-based auto-scaling techniques allow for effective scaling decisions that enhance the performance of the system and minimize the costs of operation while ensuring the satisfaction of QoS requirements in microservices-based distributed systems.

A. Reinforcement Learning-Based Scaling

Reinforcement learning enables the cloud system to learn the best way to scale by constantly interacting with the environment. In the context of resource management, RL uses an agent that observes the system's state and takes action by adding or deleting container replicas. The agent receives a reward depending on the satisfaction of the performance metrics and the cost of operation. With time, the system learns how to scale the number of containers, virtual machines, and nodes on the edges.

B. Predictive Workload Forecasting

The predictive workload forecasting techniques use past workload trends and machine learning techniques to determine the future resource requirements. These techniques use time series analysis, deep learning models, and statistical forecasting techniques to determine the future trends of workloads. These techniques help predict the future trends of workloads and enable resource orchestration platforms to allocate resources in advance, reducing the chances of service degradation and service level agreement breaches in dynamic microservices environments.

C. Meta-Heuristic Optimization

It is a popular solution for dealing with complex and difficult multi-objective resource management problems in distributed cloud-edge environments. For example, methods like Particle Swarm Optimization, Genetic Algorithms, and Ant Colony Optimization are often employed to determine the best placement, migration, and resource allocation strategies. This is particularly useful when traditional optimization techniques are too computationally expensive and impractical for use in microservice environments.

D. MAPE-Based Autonomic Frameworks

The Monitor-Analyze-Plan-Execute (MAPE) loop provides a clear framework for developing autonomic resource management systems. In the context of the MAPE loop, the monitoring component collects system performance metrics, the analysis component identifies any pattern in the workload and anomalies, the planning component determines the course of action, and the execution component implements the decisions through orchestration. With the incorporation of AI in the MAPE loop, the decisions on scaling up the distributed microservices are more intelligent.

V. COMPARATIVE ANALYSIS

However, despite the significant advancements in intelligent resource management techniques, a few areas still need to be addressed. Some of these areas include:

- Lack of interoperability between cloud and edge infrastructures
- Scalability problems in central orchestration systems
- High computational complexities of deep learning models
- Lack of a standard framework in performance benchmarking
- Lack of incorporation of security-aware scaling

After reviewing the various techniques available in the literature, it can be noted that a tradeoff among optimization accuracy, computational complexities, scalability, and energy efficiency is a common problem in all techniques. Hence, finding a balance among these factors in resource management systems is a challenge in cloud-edge microservices.

VI. OPEN CHALLENGES AND FUTURE DIRECTIONS

However, the intelligent orchestration and resource management advancements still face a number of challenges in the cloud-edge microservices architecture.

1) **Cross Domain Orchestration:** Smooth resource orchestration across clouds, edges, and IoT is a knotty problem.

2) **Ultra Low Latency Support:** Next-gen applications like autonomous systems and real-time analytics require ultra-low latencies, especially considering the advent of 5G and 6G.

3) **Energy Sustainable Computing:** High performance without sacrificing power is the key to a sustainable cloud.

4) **Security Aware Resource Allocation:** Security aspects like trust, data security, and privacy need to be taken care of during resource allocation.

5) **Federated and Collaborative Learning**: DAI techniques may facilitate collaborative resource management along with data privacy.

PROPOSED FRAMEWORK

The framework consists of five functional layers:

A. Monitoring

The first layer continuously collects metrics from the system. These metrics include CPU and memory consumption, network delays, request rates, SLA breaches, power consumption, etc. It provides a clear and observable view of the entire system.

B. Prediction & Analytics

In this layer, AI-driven prediction models are employed. These include time series prediction and reinforcement learning. These prediction models are employed to predict the demand for the system.

C. Optimization & Decision

In the next layer, the problem of resource allocation is formulated as a multi-objective optimization problem. The objectives include the minimization of latency, reduction of operational expenses, maximization of energy efficiency, and SLA fulfilment. Meta-heuristics are employed to solve the problem and find the best possible solution. These include the placement of services and their scaling.

D. Orchestration & Control

The next layer employs the techniques of scaling, migration, and provisioning. These techniques are employed by container orchestration. To enhance the scalability of the system, a distributed local agent and a centralized global controller are employed. These are combined to enhance the scalability of the system.

E. Feedback & Autonomic Loop

The last layer employs the MAPE loop. This consists of Monitor, Analyze, Plan, and Execute. The system continuously adapts by feeding the performance metrics back into the system. This helps to reassess the decisions taken by the system. The framework provides a way to integrate cloud-native systems with next-generation networks. These include 5G and beyond. It employs a multi-objective optimization problem and AI-driven prediction. It provides a scalable system with low latency and is cost-effective and sustainable.

APPLICATION AREAS

The microservice architecture in the cloud-edge setup helps manage the distributed data centers intelligently. This provides a flexible IT system. It helps power a variety of applications. These applications are modern and require the system to scale, perform well, and be reliable. Here are the key areas of application:

A. Smart Health Care

The microservice cloud platform helps power scalable electronic health records, patient monitoring, and real-time medical diagnostics. Smart resource allocation helps maintain the reliability and availability of critical health care services. It provides the necessary speed required by health care services.

B. Internet of Things (IoT) and Smart Cities

IoT applications generate vast amounts of data from sensors and other connected devices. Efficient microservice deployment and mobility-aware resource allocation help scale the data processing speed. This helps power smart city applications efficiently.

C. 5G and Beyond Network Architectures

The next-generation network uses cloud-native microservice architecture and distributed edge nodes. These nodes provide ultra-reliable and low-latency communications. Elastic resource allocation helps deploy 5G services dynamically. It helps deploy 6G services as well in the future.

D. Edge Computing and Real-time Applications

Applications like self-driving cars, virtual reality games, and industrial automation require razor-thin latency. Smart service migration and resource allocation help enhance the performance of the applications deployed on the edges.

E. High Performance Computing (HPC)

The microservice architecture deployed on the cloud platform helps scale the distributed HPC applications. Optimized communication protocols and elastic resource allocation help scale the applications efficiently.

RELATED WORK

Resource management is an area of research in both cloud and microservices-based environments. Traditionally, auto-scaling in the early days of cloud computing was mostly based on rule- or threshold-based approaches. However, these methods have proven to be time-consuming in scaling resources.

With the advent of cloud-native applications, various studies have proposed elastic resource management platforms suitable for microservices-based systems. These studies have aimed at improving scalability and flexibility in these systems without compromising the quality of service provided to users. Resource management studies have also aimed at reducing operation costs without compromising the use of resources. However, these methods have proven to have scalability limitations due to the use of centralized orchestration.

Edge and fog computing have gained popularity in service placement and migration in recent years. Multi-criteria optimization methods have also been used in service placement based on factors such as latency, energy consumption, and operation costs. Similarly, meta-heuristic methods have been used in optimizing resource management in IoT-based microservices systems based on factors such as user mobility. These methods have proven to be effective in improving the performance of systems. However, these methods have proven to have additional computational complexities.

In recent years, research in predictive orchestration using artificial intelligence techniques has gained popularity. These methods have proven effective in improving the performance of systems.

Additionally, microservices and cloud-native applications have proven effective in supporting next-generation networks based on architectures such as 5G and beyond. These architectures have proven effective in providing low-latency services in a distributed manner. Resource management in these systems has proven effective in supporting autonomic and elastic resource management systems. Various studies have proposed a taxonomy of autonomic and elastic resource management systems.

However, these methods have proven effective in optimizing a single objective at a time. There is a need to develop an integrated framework using AI-based auto-scaling methods, predictive orchestration methods, and multi-layered coordination in both cloud and edge infrastructures.

RESEARCH GAPS

In spite of the significant advances made in the intelligent and flexible resource management for cloud-edge microservice systems, a number of significant research challenges persist.

The first is that, while many researchers have shown that AI techniques like reinforcement learning and transformer predictors can be used for auto-scaling distributed systems, a unified framework for the integrated orchestration of resource provisioning, service placement, and auto-scaling is lacking.

The second is that, while AI techniques like reinforcement learning and transformer predictors have shown significant promise for auto-scaling distributed systems, deploying them for large microservice systems is still a challenge.

The third is that, while many researchers have focused on the optimization of resource provisioning for microservices, the importance of sustainability and green resource management is still largely ignored.

The fourth is that, while many researchers have proposed resource management techniques for microservices, the importance of distributed and federated resource management is still largely ignored.

The fifth is that, while many researchers have proposed resource management techniques for microservices, the importance of security and privacy-preserving resource management is still largely ignored.

The last is that, while many researchers have proposed resource management techniques for microservices, a standardized benchmarking and evaluation framework is still largely missing. Most researchers have used isolated simulations and small datasets for testing and evaluating their techniques, while a standardized benchmarking and evaluation framework is necessary for testing and evaluating the scalability and performance of cloud-edge microservice resource orchestration techniques.

FUTURE WORK

For intelligent resource management in microservice-based cloud-edge systems in the future, there is a need to focus on fully autonomous and self-adapting orchestration systems. This can be achieved through the development of integrated cross-layer optimization frameworks that bring together resource provisioning, service placement/migration, and predictive scaling

under a unified orchestration framework. Such frameworks should be based on hierarchical and hybrid orchestration to ensure a balance between scalability, performance, and optimization efficiency.

Another important area to focus on is to design lightweight intelligent orchestration mechanisms for edge computing systems. Federated learning and distributed reinforcement learning are some techniques that can be used to improve such orchestration systems.

Sustainability-aware resource management is another area that is likely to be important in future intelligent resource management systems. For future resource management systems, there is a need to ensure that energy-aware scheduling and carbon footprint estimation are integrated into auto-scaling and resource provisioning processes.

Future networks such as 5G networks today and 6G networks tomorrow will require ultra-low latency provisioning and high reliability provisioning. For such networks, there is a need to ensure predictive service migration and latency-constrained orchestration to ensure efficient support for real-time services.

Security-aware resource management frameworks are another area to focus on in future intelligent resource management systems. Such frameworks should be able to ensure reliable support for mission-critical applications.

Academia-industry collaboration is likely to be important in designing standardized benchmarking systems to assess intelligent orchestration systems in cloud-edge systems.

REFERENCES

[1] N. Kaushik, H. Kumar, and V. Raj, "Micro frontend based performance improvement and prediction for microservices using machine learning," *Cluster Computing*, 2024.

[2] M. Yang, X. Wang, B. Cai, Y. Wang, and Y. Guo, "Full stack optimization of microservice architecture: Systematic review and research opportunity," *Cluster Computing*, 2025.

[3] A. John, J. Kawash, and R. Alhajj, "Predictive container orchestration in the cloud using artificial intelligence techniques," *Cluster Computing*, 2025.

[4] M. Simić, I. Prokić, J. Dedeić, and M. Stojkov, "A hierarchical namespace approach for multi-tenancy in distributed clouds," *IEEE Access*, 2024.

[5] J. de la Croix K. Ki, B. Zerbo, and M. Soma, "Multi-criteria optimization for service placement and migration in collaborative edge computing," *Intelligent and Converged Networks*, vol. 6, no. 4, pp. 278–288, 2025.

[6] L. Shalev, H. Ayoub, N. Bshara, and E. Sabbag, "A cloud-optimized transport protocol for elastic and scalable HPC," *IEEE Micro*, vol. 40, no. 6, pp. 78–85, Nov./Dec. 2020.

[7] P. Song, H. Peng, and X. Zhang, "A micro-service approach to cloud native RAN for 5G and beyond," *IEEE Access*, vol. 11, 2023.

[8] P. Wang, L. Dong, Y. Xu, W. Liu, and N. Jing, "Clustering-based emotion recognition micro-service cloud framework for mobile computing," *IEEE Access*, vol. 8, 2020.

[9] S. M. Rajagopal, M. Supriya, and R. Buyya, "Resource provisioning using meta-heuristic methods for IoT microservices with mobility management," *IEEE Access*, vol. 11, 2023.

[10] B. Kumar, A. Verma, and P. Verma, "A multivariate transformer-based monitor-analyze-plan-execute (MAPE) autoscaling framework for dynamic resource allocation in cloud environment," *Cluster Computing*, 2025.

[11] M. Simić, I. Prokić, J. Dedeić, G. Sladić, and B. Milosavljević, "Towards edge computing as a service: Dynamic formation of the micro data-centers," *IEEE Access*, vol. 9, pp. 111600–111613, 2021.

[12] J. Zaki, M. Abdullah-Al-Wadud, S. M. R. Islam, N. S. Alghamdi, and K.-S. Kwak, "Introducing cloud-assisted micro-service-based software development framework for healthcare systems," *IEEE Access*, vol. 10, pp. 36429–36445, 2022.

[13] H. Ahmad, M. Wagner, C. Treude, and C. Szabo, "Resilient auto-scaling of microservice architectures with efficient resource management," *arXiv preprint arXiv:2506.05693*, 2025.

[14] Y. M. Dalal, S. Supreeth, S. Rohith, G. Shruthi, and B. J. Sowmya, "Towards smarter IoT through taxonomy and prospective directions for microservices placement in fog computing paradigms," *Discover Artificial Intelligence*, vol. 6, p. 89, 2026.

[15] J. Zaki, M. Abdullah-Al-Wadud, S. M. R. Islam, N. S. Alghamdi, and K.-S. Kwak, "Introducing cloud-assisted micro-service-based software development framework for healthcare systems," *IEEE Access*, vol. **10**, pp. **36429–36445**, 2022.

[16] H. Ahmad, M. Wagner, C. Treude, and C. Szabo, "Resilient auto-scaling of microservice architectures with efficient resource management," *arXiv preprint arXiv:2506.05693*, 2025.

[17] Y. M. Dalal, S. Supreeth, S. Rohith, G. Shruthi, and B. J. Sowmya, "Towards smarter IoT through taxonomy and prospective directions for microservices placement in fog computing paradigms," *Discover Artificial Intelligence*, vol. **6**, p. **89**, 2026.