Microservices in Hybrid Cloud Environments: A Survey

www.surveyx.cn

Abstract

This survey paper examines the integration of microservices architecture within hybrid cloud environments, emphasizing their collective impact on modern IT infrastructure. Microservices, characterized by modularity and independence, enable the decomposition of monolithic systems into smaller, manageable units, enhancing scalability and adaptability. This architectural style is pivotal for developing cloud-native applications, offering significant advantages in cost efficiency and flexibility by leveraging both on-premises and public cloud resources. The paper highlights the critical role of service orchestration in managing distributed microservices, providing a cohesive framework for integrating and coordinating services across diverse infrastructures. Advanced orchestration techniques are essential for optimizing performance and resource utilization. The survey also underscores the importance of resource allocation strategies in maintaining high service quality and reliability under varying workloads. Effective deployment methodologies demonstrate the successful integration of edge-cloud experiences in traditional environments. The paper identifies a gap between theoretical approaches and practical tool usability, advocating for further development to bridge this divide. Future research should focus on refining architectural partitioning, deployment modeling, and exploring AI techniques to enhance operational efficiency in the microservices lifecycle. By addressing these areas, the study aims to advance the efficiency and adaptability of IT infrastructures, ensuring they remain agile and competitive in an evolving technological landscape.

1 Introduction

1.1 Overview of Microservices in Hybrid Cloud

Microservices architecture (MSA) has become a leading paradigm in software design, emphasizing modularity, scalability, and maintainability, which are vital in today's software landscape [1]. This architecture decomposes complex applications into smaller, independent services that can be developed, deployed, and scaled autonomously, effectively overcoming the limitations of traditional monolithic and service-oriented architectures (SOA) [2]. The prominence of microservices is further highlighted by their adoption as the standard architecture for cloud-native applications, despite the complexities arising from intricate service interactions [3].

In hybrid cloud environments, microservices enable seamless integration of private and public cloud resources, allowing organizations to capitalize on both on-premises and cloud infrastructures [4]. This integration fosters cost-effective deployment strategies while maintaining control over sensitive data and applications. Moreover, the adaptability of microservices aligns with the demands of Industry 4.0, where IoT technologies and edge computing are increasingly integrated into manufacturing systems [2]. The rise of edge computing necessitates efficient scheduling techniques for microservices, underscoring the importance of resource optimization in these environments [5].

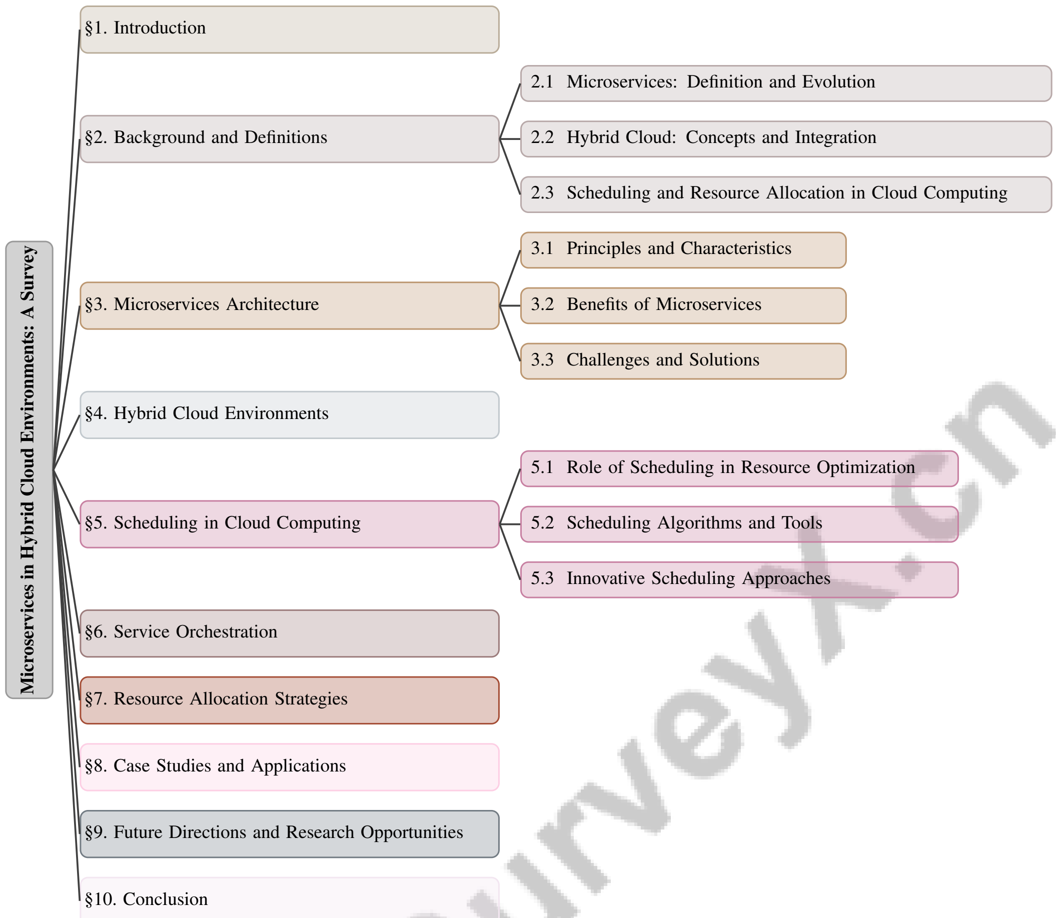

Figure 1: chapter structure

The application of AI techniques in microservices systems is gaining traction, effectively addressing challenges in development, optimization, and operational phases [6]. Collaborative Model-driven Software Engineering (CMDSE) enhances the development process by enabling multiple teams to manage the complexities associated with distributed service development [7]. Additionally, microservices streamline software development in scientific computing, bridging gaps in contemporary software engineering practices [8].

Despite their advantages, microservices introduce significant security and privacy concerns that must be addressed to maintain the integrity and confidentiality of applications in cloud computing environments. As the landscape of cloud computing evolves, microservices remain integral to developing efficient and scalable solutions in hybrid cloud environments, driving innovation across various sectors by optimizing resource allocation and addressing inherent challenges [4].

1.2 Structure of the Survey

This survey is systematically organized into ten sections, each addressing critical aspects of microservices in hybrid cloud environments. The introductory section outlines foundational concepts, emphasizing the relevance of microservices architecture (MSA) within modern IT infrastructure, particularly in hybrid cloud settings. The background and definitions section provides a comprehensive explanation of key terms and concepts, including microservices, hybrid cloud, scheduling, service orchestration, and resource allocation, while tracing their evolution in contemporary cloud computing.

The core components of the survey are explored in subsequent sections. Section three focuses on microservices architecture, detailing its principles, benefits, and challenges, including the complexities introduced by its adoption. Section four examines hybrid cloud environments, discussing the

AI-generated, for reference only.

integration of on-premises and public cloud resources, along with advantages and challenges such as cost efficiency, security, and performance [4].

In section five, the role of scheduling in optimizing resource allocation and performance within cloud computing environments is analyzed, reviewing various scheduling algorithms and their applications in hybrid cloud contexts [5]. Section six delves into service orchestration, highlighting its significance in managing microservices and exploring tools and frameworks that facilitate efficient service management.

Section seven addresses resource allocation strategies, focusing on scalability, dynamic allocation, and optimization techniques to maximize efficiency and reduce costs. The survey presents case studies and applications in section eight, showcasing real-world implementations of microservices in hybrid cloud environments and the lessons learned.

The penultimate section identifies future directions and research opportunities, discussing emerging trends and potential areas for further exploration in microservices and hybrid clouds. Finally, the conclusion synthesizes the survey's key findings, reflecting on the impact of microservices, hybrid cloud, scheduling, service orchestration, and resource allocation on modern IT infrastructure, and emphasizing the necessity for ongoing research and development in this dynamic domain.The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Microservices: Definition and Evolution

Microservices architecture signifies a transformative approach in software development, decomposing monolithic systems into small, independent services that enhance scalability, reusability, and maintainability [9, 10]. This shift addresses traditional system limitations by promoting agility and responsiveness [11]. However, transitioning from monolithic to microservices can introduce challenges, including the risk of adopting anti-patterns due to inexperience [12]. The distributed nature of Microservices Architecture (MSA) demands careful selection of patterns and strategies [13], with automation playing a crucial role in mitigating migration risks [10].

In cloud computing, microservices adeptly manage distributed, large-scale applications, optimizing resources across the edge-cloud continuum [14]. This capability is vital for IoT-integrated environments requiring rapid market adaptability [2]. Nevertheless, the distributed architecture introduces complexity, necessitating collaborative efforts for effective service management [7]. Migrating legacy systems to microservices requires detailed planning to mitigate costs and risks [15], especially within hybrid cloud-edge infrastructures characterized by diverse deployment and resource capabilities [4]. Understanding and optimizing microservices implementation is crucial for leveraging their potential in modern IT infrastructures, marking a significant shift towards scalable, efficient, and sustainable cloud computing solutions.

2.2 Hybrid Cloud: Concepts and Integration

Hybrid cloud environments integrate on-premises infrastructure with public cloud services, creating a cohesive IT ecosystem that enhances resource allocation, cost efficiency, and scalability [16, 17]. This model supports serverless function orchestration via a Hybrid Cloud Scheduler (HCS), optimizing operations and reducing costs [18]. Managing distributed databases in these environments demands careful design to ensure efficient data distribution and performance [19]. Robust frameworks prioritizing data-flow modeling facilitate seamless microservices integration [20]. The differences between Service-Oriented Architecture (SOA) and Microservices Architecture (MSA) necessitate specialized approaches for hybrid cloud integration [21].

Containerization technologies, such as Docker, are pivotal in hybrid cloud setups, enabling lightweight application deployment within microservices architectures [22]. These technologies, along with software-defined networking and encryption, significantly impact network performance [23]. DevOps practices are essential in hybrid cloud environments, supporting continuous integration and deployment in alignment with cloud-native paradigms [24]. Security and privacy challenges, particularly concerning container, data, and network security, are critical [25]. A tailored migration method addressing these constraints ensures secure operations in hybrid clouds [26]. As hybrid cloud environ-

AI-generated, for reference only.

ments evolve, integrating microservices and serverless computing within disaggregated architectures promises enhanced flexibility and efficiency [27]. This evolution underscores the hybrid cloud's role in modern IT infrastructure, driving innovation and transformation while managing complex, distributed systems [28].

2.3 Scheduling and Resource Allocation in Cloud Computing

Scheduling and resource allocation are critical in managing cloud computing environments, especially within hybrid clouds, where they optimize resource utilization and maintain Quality of Service (QoS) [29]. The shift from monolithic to microservices architectures introduces complexities due to intricate service interdependencies, requiring advanced scheduling strategies to manage the dynamic, distributed nature of microservices [30]. In hybrid clouds, characterized by integrating on-premises and public resources, scheduling must address environmental heterogeneity and dynamism to avoid Service Level Agreement (SLA) violations and inefficiencies [31].

Resource allocation challenges include efficiently managing container placements for microservices to minimize infrastructure costs while maintaining performance [32]. Dynamic resource allocation techniques are vital for optimizing cost and performance, ensuring service-level objectives (SLOs) compliance [33]. Independent autoscaling of microservices, often neglecting interdependencies, can result in inefficient resource allocation and increased costs, necessitating collective autoscaling approaches [34].

Scheduling serverless computing instances, like Amazon Lambda functions, involves managing constraints such as token-bucket mechanisms to maintain QoS [35]. The Service Placement Problem in Microservice Systems (SPPMS) involves deploying services to accommodate dependencies and maintain QoS [11]. Inadequate scheduling in edge environments, due to insufficient network QoS considerations, can lead to SLO violations, highlighting the need for robust scheduling frameworks [5].

AI-driven approaches for dynamic resource allocation are being explored to meet the demand for scalable, efficient resource management in hybrid clouds. These methods aim to multiplex shared resources among microservices efficiently, addressing latency issues that threaten SLOs [36]. As hybrid cloud environments evolve, leveraging dynamic resource allocation and innovative scheduling methods will be essential for navigating these infrastructures' complexities, ensuring seamless microservices operation and SLA fulfillment.

In recent years, Microservices Architecture has gained significant attention for its potential to enhance software development and deployment processes. This architectural style promotes a modular approach, allowing for independent development, deployment, and scaling of services. To better understand the complexities and advantages of this architecture, Figure 2 illustrates the hierarchical structure of Microservices Architecture. This figure highlights its core principles and characteristics, while also categorizing the benefits, such as improved scalability and operational efficiency, alongside the challenges associated with transitioning to and managing a microservices-based environment. By providing a comprehensive overview, the figure enhances our understanding of microservices in the context of modern IT infrastructure.

3 Microservices Architecture

3.1 Principles and Characteristics

Microservices architecture is founded on modularity, decentralization, and adaptability, facilitating the management of complex software systems by decomposing monolithic applications into smaller, autonomous services, each representing a distinct business capability [9]. This decomposition enhances operational independence, enabling services to evolve, scale, and deploy autonomously, thus boosting system resilience and scalability [29].

A key principle is the bounded context, ensuring each service operates within a defined scope, minimizing dependencies and enhancing maintainability [37]. This is crucial for decentralized data management, as frameworks categorize microservices data practices [38]. Containerization technologies, such as Docker, provide lightweight deployment environments, enhancing scalability and operational efficiency [32].

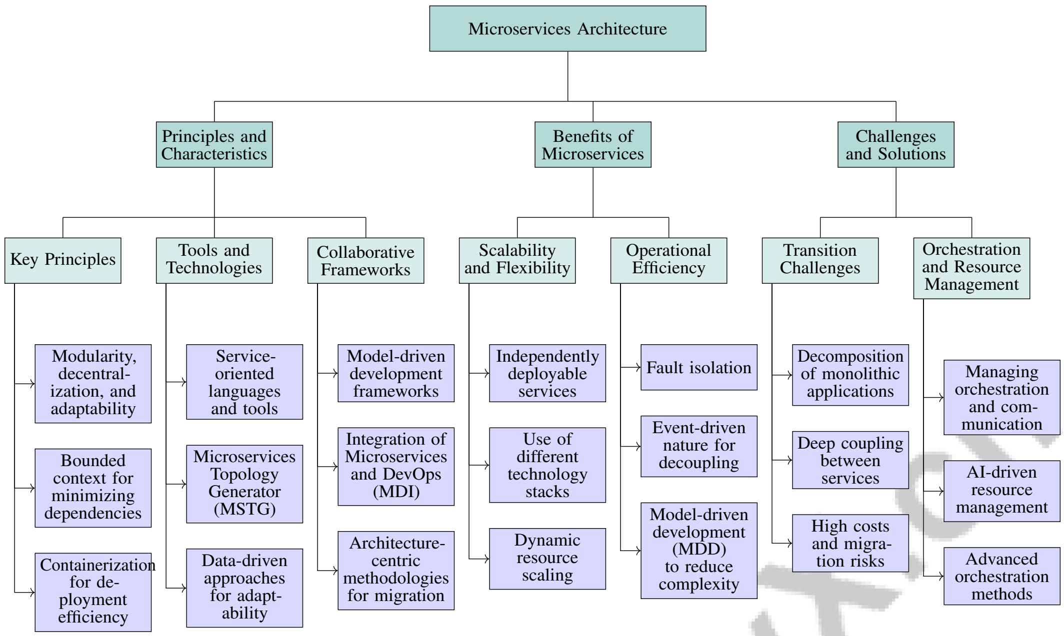

Figure 2: This figure illustrates the hierarchical structure of Microservices Architecture, highlighting its principles, benefits, and challenges. It categorizes the architecture into key principles and characteristics, benefits focusing on scalability and operational efficiency, and challenges related to transition and management, providing a comprehensive overview of microservices in modern IT infrastructure.

Service-oriented languages and tools facilitate communication and coordination, essential for modular development [17]. Tools like the Microservices Topology Generator (MSTG) enable the creation of microservices topologies that incorporate network components for realistic performance evaluations [39]. Linked-microservices decomposition emphasizes breaking down services into smaller components to optimize resource utilization, particularly in fog computing architectures [30].

Adaptability is enhanced through data-driven approaches using curated datasets for performance optimization [37]. Dynamic adaptability involves varying internal functions to balance execution time and quality [34]. This adaptability is vital for optimizing IT service management in hybrid cloud environments [29].

Collaborative model-driven frameworks emphasize internal and external collaboration, enhancing development through structured modeling [7]. Integrating Microservices and DevOps (MDI) streamlines scientific computing development, fostering collaboration and efficiency [8]. Architecture-centric methodologies guide migration, ensuring a smooth transition from monolithic to microservices architectures [15].

As illustrated in Figure 3, microservices architecture is characterized by modularity, independence, and adaptability, enabling the development of large, complex applications as cohesive, fault-tolerant services. This figure emphasizes the key principles of microservices architecture, focusing on modular decomposition, bounded context, operational independence, containerization, data-driven adaptability, and service-oriented tools as essential components in enhancing microservices architecture. Collaborative frameworks, DevOps integration, and architecture-centric methodologies further facilitate effective microservices development and management. This approach addresses challenges posed by monolithic systems, allowing efficient maintenance and scalability. The use of containerization and data-driven strategies supports effective microservice implementation, reflecting organizational structures in architectural patterns. Research highlights identifying suitable microservices candidates within monolithic systems, offering datasets of open-source projects exemplifying effective patterns, thus supporting further investigation in this evolving field [9, 40]. These elements enable efficient operations in dynamic cloud environments, underscoring the transformative potential of microservices in modern IT infrastructure.

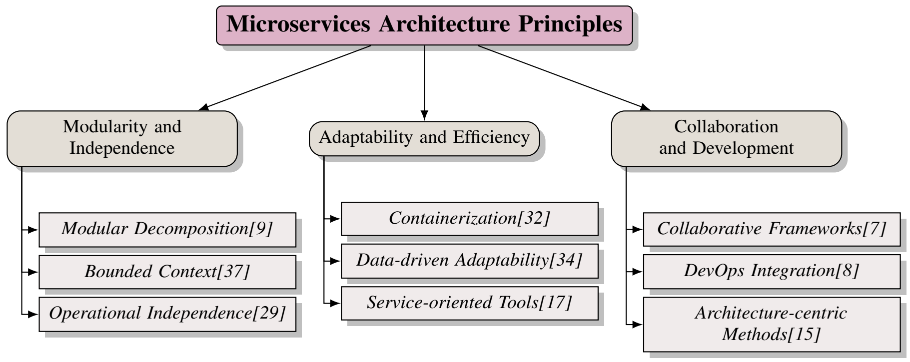

Figure 3: This figure illustrates the key principles of microservices architecture, focusing on modularity and independence, adaptability and efficiency, and collaboration in development. It highlights the use of modular decomposition, bounded context, operational independence, containerization, data-driven adaptability, and service-oriented tools as essential components in enhancing microservices architecture. Collaborative frameworks, DevOps integration, and architecture-centric methodologies further facilitate effective microservices development and management.

3.2 Benefits of Microservices

Microservices architecture offers significant benefits, notably in scalability and flexibility, essential for modern software development and maintenance. By decomposing applications into smaller, independently deployable services, organizations can allocate resources effectively across diverse cloud infrastructures, enhancing scalability and accelerating deployment and maintenance processes [41].

A major advantage is the ability to use different technology stacks for individual services, enhancing resilience and adaptability. This flexibility allows seamless integration with advanced technologies, such as artificial intelligence, improving personalization and operational efficiency [1]. The architecture supports fault isolation and resource efficiency through its event-driven nature, decoupling services to enhance scalability [9]. Dynamic resource scaling ensures efficient utilization without compromising service quality [34]. Collective autoscaling approaches manage resources for interdependent microservices, improving latency and reducing costs [34].

Microservices enhance component isolation and packaging, improving scalability and deployment speed across domains [41]. However, addressing security concerns and developing comprehensive testing frameworks are crucial to overcoming practical limitations [42].

Research indicates that model-driven development (MDD) mitigates the accidental complexity in microservices architecture, enabling efficient service design [7]. Integrating DevOps practices within microservices frameworks enhances collaboration between scientists and engineers, improving software quality and streamlining development [8].

Adopting microservices architecture provides substantial benefits, including enhanced scalability, flexibility, and operational efficiency, supporting the rapid evolution of software systems. This ensures organizations remain agile and competitive, with the capacity to transform monolithic systems into microservices, highlighting the architecture's transformative potential [9].

3.3 Challenges and Solutions

Transitioning to microservices architecture entails several challenges, primarily due to its decentralized and distributed nature. Decomposing monolithic applications into microservices requires careful analysis to ensure each service maintains a cohesive and independent business function [12]. Deep coupling between services can undermine modularity and flexibility [12].

Managing orchestration and communication between microservices also presents challenges. Central orchestrators can create bottlenecks and single points of failure, compromising stability [14]. Porting choreographed microservice benchmarks to orchestrated frameworks simplifies debugging and enhances reliability [3].

Resource allocation and workload management are further challenges, as microservices environments require adaptive systems to respond to real-time workload and resource changes [4]. Managing multiple dynamic workloads can lead to suboptimal resource utilization and increased costs [33]. AI-driven frameworks optimize resource usage and enhance performance through intelligent decision-making [4].

The migration process itself is fraught with difficulties, including high costs and risk of not achieving anticipated benefits. An architecture-centric methodology provides a structured pathway, integrating established techniques to address unique challenges during migration to microservices [15]. This methodology mitigates risks and ensures a smooth transition.

To navigate microservices complexities, organizations should implement structured decomposition techniques, advanced orchestration methods, and AI-driven resource management to optimize performance and scalability. This approach transforms large applications into cohesive services while leveraging architectural innovations like edge computing to enhance responsiveness and throughput [9, 43]. By adopting these solutions, organizations can ensure efficient, scalable, and resilient operations in cloud environments.

4 Hybrid Cloud Environments

4.1 Integration of On-Premises and Public Cloud Resources

Integrating on-premises resources with public cloud services in hybrid cloud environments enhances enterprise IT infrastructure by optimizing flexibility, scalability, and efficiency. This integration leverages both private and public clouds, facilitated by transitioning from monolithic systems to microservices architectures. The modular monolith approach acts as a transitional phase, focusing on loosely coupled modules that can evolve into microservices [44]. Service meshes are crucial for managing microservices interactions, ensuring efficient communication and connectivity in distributed systems [45]. Moreover, semantically interoperable microservices enhance data analytics, especially in IoT applications, by enabling seamless service integration [43].

A primary challenge is developing a conceptual model that supports scalability and adaptability from early development phases. The absence of such models can impede effective microservices deployment and management, highlighting the need for frameworks to bridge these gaps. Evaluations of existing implementations emphasize aligning integration strategies with organizational objectives and stakeholder needs [16]. Security, testing, and organizational structure are critical considerations for the successful deployment and operation of distributed systems [17]. Effective organizational structures aid in coordinating and managing complex microservices deployments. The evolution of service-oriented computing to microservices necessitates architectural shifts to integrate on-premises and cloud resources [41].

Successful integration requires defining migration goals aligned with stakeholder needs, analyzing organizational constraints, and establishing a structured migration pathway [46]. Advanced frameworks and methodologies can effectively manage hybrid cloud complexities, ensuring seamless operation and optimal resource utilization. Technologies like Docker and Kubernetes provide infrastructure for deploying and managing microservices, while integrating edge computing enhances data processing capabilities [30]. Tools like the INDIGO-DataCloud Orchestrator facilitate cloud resource management across multiple infrastructures [41].

Frameworks such as DAPP-ECC support hybrid cloud integration by enabling autonomous service placement and migration among datacenters [14]. Microservices and DevOps Integration (MDI) allows independent microservices development by experts, integrated through a DevOps pipeline, enhancing scalability and adaptability [8]. However, challenges like deployment environment heterogeneity, real-time monitoring needs, and microservices scalability across nodes must be addressed for effective on-premises and public cloud integration [4].

Figure 4 illustrates the key strategies, challenges, and technological tools involved in integrating on-premises and public cloud resources. It highlights the use of modular monoliths, service meshes, and semantic interoperability as integration strategies; conceptual models, security, and organizational structure as challenges; and Docker, Kubernetes, and DAPP-ECC as technological tools. This visual representation reinforces the complex interplay of these elements in achieving successful hybrid cloud integration.

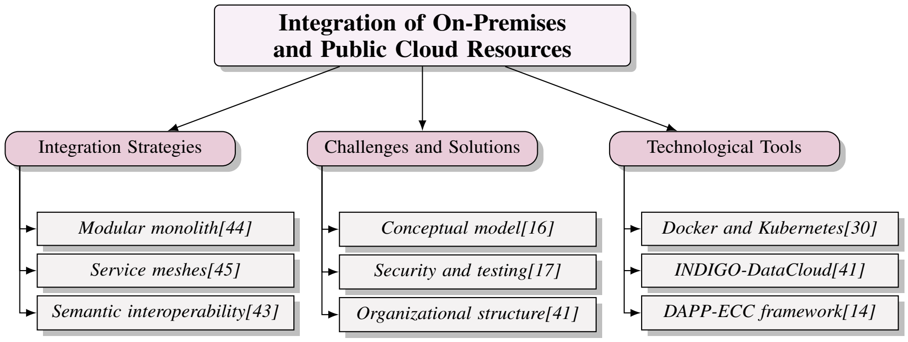

Figure 4: This figure illustrates the key strategies, challenges, and technological tools involved in integrating on-premises and public cloud resources, highlighting the use of modular monoliths, service meshes, and semantic interoperability as integration strategies; conceptual models, security, and organizational structure as challenges; and Docker, Kubernetes, and DAPP-ECC as technological tools.

4.2 Advantages of Hybrid Cloud Solutions

Hybrid cloud solutions offer significant advantages, particularly in cost efficiency and flexibility, crucial for optimizing IT infrastructure. By combining on-premises resources with public cloud services, organizations achieve optimal resource allocation and operational efficiency [16]. Hybrid cloud architectures support independent deployment and maintainability of microservices, enhancing agility and responsiveness in software development [47].

A key benefit is improved availability outcomes, achieved through configuration adjustments and redundancy strategies, enhancing reliability and continuous service delivery [48]. This is vital for maintaining high availability across diverse cloud infrastructures, supporting dynamic workloads and user demands [49]. Hybrid cloud environments facilitate seamless microservices integration, reducing distributed transaction complexity and optimizing performance. Domain-driven design effectively addresses legacy systems' complexities, aiding the transition to cloud-native architectures [49]. This approach supports microservices' modular nature, enabling efficient resource management and scalable operations.

Hybrid cloud solutions also enhance environmental sustainability by optimizing energy and resource efficiency, reducing IT operations' carbon footprint. Dynamic feedback control methods improve query performance and minimize packet loss, ensuring high-quality service delivery while mitigating environmental impact [47]. The adoption of hybrid cloud solutions offers a compelling blend of cost efficiency, flexibility, scalability, and sustainability, making them ideal for modernizing IT infrastructures. Integrating on-premises systems with public and private clouds balances scalable resource needs, enhances collaboration, and complies with regulatory standards while optimizing performance and minimizing latency. Successful implementation requires careful planning, robust security protocols, and continuous monitoring to manage distributed environments effectively. Hybrid cloud strategies provide actionable insights and best practices, empowering IT leaders and developers to maximize cloud infrastructure benefits [16, 50].

4.3 Challenges in Hybrid Cloud Management

Hybrid cloud management presents challenges, particularly in security, performance, and resource management. Orchestrating and integrating independent microservices is complex due to dynamic workloads and dependencies, risking Quality of Service (QoS) violations if resources are not proactively allocated [51]. The distributed nature of hybrid clouds complicates seamless operation and integration, especially concerning data security and latency [16].

Security is a paramount concern, with current studies often overlooking infrastructure-based attacks and recovery strategies, indicating a gap in comprehensive microservice security coverage [52]. The high data volume during migration, particularly in memory-intensive applications, poses challenges exacerbated by network bandwidth and latency limitations [53]. Infrastructure-level adaptations and reactive strategies dominate current research, limiting proactive approaches and application-level

AI-generated, for reference only.

adaptations crucial for maintaining energy efficiency and reducing carbon footprints while ensuring QoS.

Resource management is further complicated by the HA State Controller's 2N redundancy model, leading to high resource usage [54]. The Bag-of-Tasks focus by Toosi et al. may not generalize to applications with complex dependencies, highlighting current approaches' limitations [31]. Static analysis reliance in automated migration methods may not capture dynamic runtime interactions, affecting microservices identification accuracy [10].

Organizational challenges include limited focus on collaborative aspects of microservices architecture (MSA) development, particularly team interactions and dependency management [7]. The specificity of current studies restricts broader applicability, often concentrating on particular applications [9]. Managing shared database tables during migration further complicates organizational challenges [9].

Addressing hybrid cloud management challenges requires a comprehensive approach encompassing security, performance, and organizational strategies to ensure effective hybrid cloud infrastructure operations. Frameworks like MS2M, minimizing downtime and utilizing existing messaging infrastructure for state synchronization, enhance migration performance [55]. Integrating technical and organizational dimensions helps organizations navigate hybrid cloud complexities and achieve seamless microservices architecture transitions.

5 Scheduling in Cloud Computing

5.1 Role of Scheduling in Resource Optimization

Scheduling is pivotal for optimizing resource utilization in cloud environments, particularly within microservices and hybrid cloud architectures. It ensures efficient management of dynamic workloads and resource allocation, maintaining high service quality and performance. The complexity of cloud systems demands sophisticated scheduling mechanisms to navigate microservices deployment and operation intricacies [5]. In hybrid cloud settings, scheduling is crucial for maintaining service continuity during runtime migrations. Techniques like iterative progression emphasize stakeholder involvement, underscoring the need for adaptive scheduling to optimize resource allocation and performance across diverse cloud infrastructures [56]. This adaptability is vital for seamless microservices operation in dynamic environments [57].

Advanced strategies, such as those in the Reclaimer framework, utilize real-time data to predict potential Quality of Service (QoS) violations and proactively adjust allocations, minimizing idle resources and enhancing efficiency [58]. Frameworks like FIRM employ machine learning to detect and mitigate Service Level Objective (SLO) violations through dynamic reprovisioning, highlighting scheduling's strategic role in resource optimization [36]. AI-driven frameworks, as proposed by Barua et al., adapt resource allocations based on historical performance, preventing under- and over-provisioning [59]. Model-driven approaches enhance this by integrating performance monitoring to establish a continuous performance engineering loop, facilitating proactive identification and remediation of performance issues [60].

The INDIGO-DataCloud Orchestrator exemplifies how scheduling can abstract cloud resource management complexity, providing a flexible environment for deploying scientific applications [20]. In edge computing, scheduling is pivotal for optimizing resource use, as seen in frameworks like Humas, which generate capacity adjustment plans based on CPU usage patterns and predicted workloads. The method proposed by Ali et al. optimally places microservices and databases across physical nodes in a cloud-edge continuum, minimizing costs and latency, underscoring strategic scheduling's importance in resource optimization [61]. While Kubernetes supports high availability, it can encounter significant service outages under certain failure scenarios, emphasizing the need for improved scheduling and management strategies. The Adaptable TeaStore demonstrates dynamic scheduling's role in optimizing resource use by switching between local and external services based on real-time conditions [62].

The Distributed Asynchronous Placement Protocol for the Edge-Cloud Continuum (DAPP-ECC) optimizes resource use by enabling asynchronous deployment and migration of microservices, enhancing overall system performance [14]. The Polaris Scheduler, an SLO-aware framework for edge computing, optimizes microservice placement based on network QoS and resource requirements [5]. The Overbooking Lambda Functions (OLM) method allows dynamic resource allocation based on estimated execution times, improving utilization efficiency [35]. Effective scheduling is crucial for optimizing resource allocation in cloud environments, ensuring efficient operation and performance of microservices across hybrid infrastructures. By adopting advanced scheduling techniques, organizations can enhance resource utilization and cost efficiency while maintaining high service quality and reliability. These approaches, particularly when integrated with AI and microservices, facilitate real-time resource allocation optimization based on demand patterns, enhancing decision-making and operational agility. This evolution minimizes system downtime and latency, allowing for dynamic workload adjustments and improved performance metrics, especially in complex environments like travel reservation systems and multi-cloud infrastructures [63, 64, 65, 66].

5.2 Scheduling Algorithms and Tools

Method Name	Resource Management	Scaling Strategies	Optimization Techniques
ASA[66]	Resource Management Functionalities	-	Custom Scheduling Algorithms
SHPA[67]	Resource-efficient Heuristics	Hierarchical Architecture	Resource Exchange
Pogonip[68]	Resource Utilization	Elastic Scaling	Heuristic Approach
SS[69]	Resource Allocation Framework	Elastic Scaling Framework	Load Status Detector
SKD[70]	Dynamic Scheduling Algorithm	Dynamically Adjust Scale	Greedy Scheduling Algorithm
PEMA[71]	Feedback-based Tuning	Opportunistic Resource Reduction	Feedback-based Approach
SPMA[72]	Cloud Resource Management	Elasticity And Heterogeneity	Particle Swarm Optimization
BS[73]	Resource Provisioning	Dynamically Adjusts Scale	Optimize Resource Provisioning
MV[74]	-	-	Static Analysis Tool
N/A[75]	Dynamic Resource Allocation	Dynamically Adjust Resources	Machine Learning-based
PEC[76]	Dynamic Resource Allocation	Parsimonious Scaling Strategies	Lightweight Architecture
HCS[18]	Dynamic Resource Allocation	Dynamic Resource Allocation	Efficient Scheduling
MSO[77]	Amazon Aws Ec2	-	Grid Search
SACS[33]	Dynamic Selection Scaling	Dynamically Adjust Scale	Simulated Annealing Optimization
DPA[31]	Resource Provisioning	Dynamically Adjusts Resources	Data-aware Provisioning
PS[5]	Plugin-based Approach	-	Slo-aware Scheduling
OLM[35]	Dynamic Allocation	Overbooking Lambda Functions	Statistical Modeling

Table 1: This table presents a comprehensive comparison of various scheduling algorithms and tools utilized in hybrid cloud environments, focusing on their resource management, scaling strategies, and optimization techniques. The methods highlighted illustrate diverse approaches to enhancing microservices performance and resource allocation efficiency, showcasing both traditional and innovative solutions.

Scheduling algorithms and tools are integral to optimizing resource allocation and enhancing microservices performance in hybrid cloud environments. The Aneka Scheduling API (ASA) provides a flexible framework for creating and integrating custom scheduling algorithms tailored to diverse application requirements [66]. This adaptability is crucial for addressing the dynamic nature of microservices, ensuring efficient resource management and operational performance. Innovative approaches like Smart HPA introduce resource-efficient heuristics for resource exchange among microservices, enabling scaling decisions that exceed predefined resource limits [67]. This method addresses traditional Horizontal Pod Autoscalers' (HPAs) limitations, enhancing resource utilization and system performance. Similarly, Pogonip formulates the placement problem as an Integer Linear Programming (ILP) optimization challenge, employing heuristics to approximate real-world execution scenarios for optimizing microservice deployment [68].

To better understand the landscape of these scheduling algorithms and tools, Figure 5 categorizes them into three primary areas: Resource Management, Scaling Strategies, and Optimization Techniques, highlighting key methods in each category. Table 1 provides a detailed classification of scheduling algorithms and tools, emphasizing their roles in resource management, scaling strategies, and optimization techniques within hybrid cloud environments. StatuScale evaluates workload stability through a load status detector, selecting appropriate scaling strategies to ensure elastic scaling that adapts to varying demands [69]. In contrast, Skedulix utilizes a greedy algorithm to allocate serverless function executions between public and private clouds, optimizing resource distribution across hybrid infrastructures [70]. In autoscaling, PEMA initially allocates sufficient resources to all microservices and opportunistically reduces resources based on performance feedback, optimizing resource allocation without compromising service quality [71]. The integration of Particle Swarm Optimization (PSO) and Memetic Algorithms (MA) in SPMA represents a hybrid approach to multi-objective workflow scheduling, dynamically adapting to cloud resources' performance to optimize scheduling outcomes [72].

BurScale evaluates cost savings and performance through experiments with stateless and stateful applications, demonstrating its efficacy in optimizing resource allocation in cloud environments [73]. Microvision employs static code analysis to reconstruct microservice architectures, presenting them in a 3D augmented reality environment to enhance understanding of complex scheduling interactions [74]. Machine learning models are utilized by Sinan to predict performance and optimize resource allocation for microservices, showcasing data-driven approaches' potential to enhance scheduling efficiency and performance [75]. PEC, a dynamic resource allocation method, employs a PID controller to adjust the number of microservices based on service request queue length, further highlighting advanced algorithms' role in resource management [76].

The HCS orchestrates serverless functions' execution across heterogeneous infrastructures, enabling efficient management of batch-processing pipelines [18]. Additionally, Dinh Tuan et al.'s method employs Grid search and Random search techniques to dynamically optimize microservices configurations, enhancing resource allocation strategies' adaptability [77]. SACS employs simulated annealing to dynamically adjust container resource allocation in response to varying workloads while minimizing execution time and costs [33]. The Data-aware Provisioning Algorithm calculates the additional resources needed to meet deadlines for data-intensive applications, considering both data transfer and execution times on public cloud resources [31].

The effectiveness of scheduling algorithms is further exemplified by the Polaris Scheduler, which accounts for resource availability and network QoS requirements, ensuring microservices can fulfill their SLOs over time [5]. The OLM method facilitates dynamic resource allocation based on estimated execution times, thereby enhancing utilization efficiency [35]. The development and implementation of advanced scheduling algorithms and tools are vital for optimizing resource allocation in cloud computing, ensuring efficient operation of microservices in hybrid cloud infrastructures. These advancements enhance resource utilization efficiency, reduce operational costs, and improve service quality, significantly advancing hybrid cloud computing development and functionality. By integrating edge computing and cloud resources, organizations can optimize workload distribution, minimize latency, and ensure rapid response times, particularly in data-intensive fields like scientific research and the Internet of Things (IoT). This evolution is supported by best practices in performance engineering that address the complexities of managing diverse systems within hybrid architectures, ultimately facilitating novel application deployment and improving overall system efficiency [41, 50, 27].

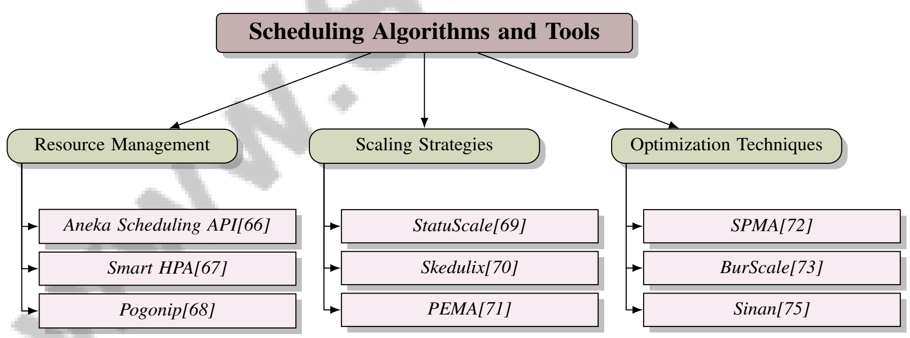

Figure 5: This figure illustrates the categorization of scheduling algorithms and tools into three primary areas: Resource Management, Scaling Strategies, and Optimization Techniques, highlighting key methods in each category.

5.3 Innovative Scheduling Approaches

Innovative scheduling approaches in cloud computing increasingly focus on enhancing performance and resource utilization through advanced techniques that address modern IT infrastructures' complexities. The integration of AI and machine learning into scheduling frameworks represents a significant advancement. For instance, FIRM leverages online telemetry data for fine-grained resource management, surpassing traditional static heuristics with dynamic adjustments that significantly enhance performance and resource efficiency [36]. The effectiveness of these innovative scheduling methods lies in their ability to adapt quickly to changes in workload and service offerings, allowing broad exploration of resource configurations before refining to optimal solutions [33]. The OLM method exemplifies this adaptability by improving resource utilization, reducing blocking probabilities, and responding to varying demand conditions [35].

Chadha et al. introduced a benchmark that provides a structured approach to evaluating performance and cost across multiple deployment strategies, offering insights often overlooked in existing benchmarks [78]. This structured evaluation is crucial for understanding trade-offs in different scheduling strategies and optimizing resource allocation. PerfSim, a performance simulator for cloud-native microservices, offers a fast, accurate, and user-friendly method for evaluating microservice performance, achieving significant speed improvements over traditional methods [79]. This tool is invaluable for testing and refining scheduling algorithms, providing a reliable platform for performance assessment.

The Polaris Scheduler, an SLO-aware scheduling framework, optimizes microservice placement based on network QoS and resource requirements, ensuring microservices can meet their SLOs over time [5]. Future research will explore hypergraphs for Cluster Topology Graph, additional SLOs, and distributed scheduling frameworks to enhance performance in larger clusters [5]. Toosi et al.'s method continuously updates resource provisioning decisions based on real-time data transfer rates and task execution times, showcasing an innovative approach to dynamic resource management [31]. This continuous updating mechanism is essential for maintaining optimal performance in rapidly changing cloud environments.

These innovative scheduling approaches underscore the importance of adaptability, real-time feedback, and automation in optimizing cloud resource management. By leveraging these advanced techniques, cloud environments can achieve enhanced performance, sustainability, and cost-effectiveness, driving the evolution of cloud computing infrastructure. Future research could explore hybrid approaches that combine static and dynamic analysis and refine clustering algorithms to improve microservices identification [10].

6 Service Orchestration

6.1 Importance of Service Orchestration

Service orchestration is crucial in hybrid cloud environments, enabling seamless integration and management of microservices across various infrastructures. The complexity of these environments necessitates sophisticated orchestration solutions that automate deployment and optimize distribution, thus improving resource utilization and reducing operational costs. The Polaris Scheduler exemplifies this by ensuring microservices in edge environments meet network Service Level Objectives (SLOs), thereby enhancing application performance and reliability [5].

In hybrid clouds, orchestration provides a unified framework allowing newly deployed microservices to function as cohesive units within complex architectures. This integration is vital for maintaining performance and reliability amidst dynamic workloads and distributed resources. Decision models for selecting orchestration patterns enhance decision-making in microservices architecture, ensuring the use of appropriate strategies to meet specific application demands [13].

Service orchestration also simplifies debugging by offering enhanced visibility and control over service interactions, which is essential for identifying and resolving performance bottlenecks. However, integrating multiple microservices increases complexity, requiring careful orchestration and monitoring to manage these interactions effectively [8]. Addressing these challenges is key to advancing orchestration capabilities in hybrid cloud settings, ensuring efficient resource management, enhanced security, and consistent service quality across distributed infrastructures.

6.2 Service Orchestration: Tools and Frameworks

Service orchestration in hybrid cloud environments employs various tools and frameworks to manage and coordinate microservices, ensuring optimal performance and resource utilization. Kubernetes is a prominent framework, widely used for its robust container orchestration capabilities, enabling automated deployment, scaling, and management of containerized applications [80]. It provides a comprehensive platform for orchestrating microservices, particularly within Docker environments, facilitating seamless integration and management of distributed services [45].

The Dynamic Executive Microservices Architecture Model (DEMAM) uses a BPMN-based workflow engine to enhance microservices orchestration, improving component interaction management and

AI-generated, for reference only.

enabling dynamic adaptation to changing conditions [81]. This model underscores the importance of structured workflows in orchestrating complex microservice architectures, ensuring efficient execution and coordination across diverse cloud environments.

The Simple Service Management and Monitoring Protocol (SSMMP) automates execution, scaling, and reconfiguration of cloud-native applications, facilitating direct communication between microservices and management agents, thus streamlining orchestration and enhancing responsiveness in cloud-native environments [82].

Security remains a critical concern, addressed by the Cloud Deployment Infrastructure (CDI), which allows orchestrators to validate the provenance of microservices and ensure compliance with security policies before deployment [83]. This capability is crucial for maintaining the integrity and security of microservices in distributed cloud environments.

Moreover, Augmented Reality (AR) visualization methods, such as AR-MV, offer a user-centric, context-aware approach to exploring microservices architectures. AR-MV facilitates interactive exploration, providing insights into orchestration and operation, thereby enhancing the management of complex cloud infrastructures [84].

The tools and frameworks available for service orchestration are essential for managing the complexities of microservices in hybrid cloud environments. By leveraging advanced solutions, including hybrid cloud models, policy-based resource management, and edge computing, organizations can optimize resource utilization, enhance security through centralized control and real-time monitoring, and ensure seamless operation of distributed services across various cloud infrastructures. These strategies promote efficient resource management, cost-effectiveness, and improved performance while ensuring compliance with regulatory standards, thereby meeting modern computing's increasing demands [27, 85, 86].

6.3 Advanced Orchestration Techniques

Advanced orchestration techniques in hybrid cloud environments focus on enhancing efficiency, security, and interoperability of microservices deployments. As cloud infrastructures grow in complexity, sophisticated orchestration strategies are imperative to integrate diverse services while optimizing resource utilization. Future research should prioritize enhancing orchestration techniques by exploring novel deployment strategies and validating these in real-world scenarios [45].

A critical area of advancement involves developing enhanced security measures to protect microservices from vulnerabilities inherent in distributed systems, including robust authentication protocols and compliance with security standards across various cloud platforms [17]. Establishing standards for cloud interoperability is also essential, facilitating seamless communication and data exchange between heterogeneous cloud services, thereby improving service orchestration efficiency.

Innovative pricing strategies are vital for optimizing cloud service utilization. By investigating dynamic pricing models that reflect real-time demand and resource availability, organizations can achieve cost-effective management of cloud resources, further enhancing orchestration efficiency [17]. These strategies improve resource allocation and ensure optimal utilization of cloud services, minimizing waste and maximizing performance.

Advancing orchestration techniques requires a comprehensive strategy that integrates security measures to ensure tenant isolation in multi-tenant environments, enhances interoperability across heterogeneous cloud infrastructures, and implements dynamic resource management to optimize microservices deployment and performance across diverse computing and networking tiers. This multifaceted approach addresses high-level application security requirements and leverages existing open-source solutions while optimizing resource allocation based on compute and network usage relationships, thereby improving overall application efficiency and meeting complex user demands [83, 9, 87, 20].

As illustrated in Figure 6, advanced orchestration techniques in hybrid cloud environments encompass various strategies aimed at optimizing efficiency, enhancing security, and establishing interoperability standards. Each category within the figure highlights key strategies and technologies that contribute to the improvement of microservices deployment and management. By focusing on these areas, future research can significantly enhance service orchestration capabilities, driving innovation and efficiency in hybrid cloud environments.

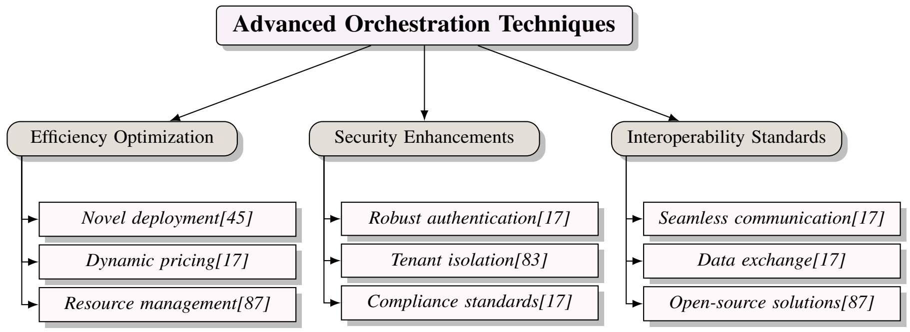

Figure 6: This figure illustrates advanced orchestration techniques in hybrid cloud environments, focusing on efficiency optimization, security enhancements, and interoperability standards. Each category highlights key strategies and technologies for improving microservices deployment and management.

7 Resource Allocation Strategies

7.1 Scalability and Resource Allocation

Effective resource allocation strategies are pivotal for scalability in hybrid cloud environments, maintaining performance across diverse workloads. The Ali-Pro approach exemplifies this by dynamically adapting resources based on historical data and real-time metrics, optimizing utilization [88]. Such adaptability ensures swift responses to demand fluctuations. Integrating edge and cloud resources is crucial for scalability, as demonstrated by Houmani et al., who balance analysis speed and accuracy through a hybrid approach [56]. Frameworks like Humas adjust to changing performance patterns in microservices, keeping resource allocation responsive [89].

Figure 7 illustrates the hierarchical structure of scalability and resource allocation strategies in hybrid cloud environments, highlighting key approaches in resource allocation, microservice deployment, and the integration of edge and cloud resources. Reducing delays and overheads in microservice deployments is essential for scalability. Liu et al. focus on minimizing image pull delays and communication overhead, critical for microservices' efficiency [90]. The CaMP-INC approach optimizes resource distribution, balancing execution time and cost in microservice architectures [61]. The Timed SmartDeployer method dynamically manages VM speeds and startup times, sustaining system performance under variable workloads [91]. This is crucial for preventing bottlenecks and maintaining service quality.

Effective resource allocation strategies, as highlighted by Barua et al., reduce latency and enhance user experience [92]. Armani et al. propose a heuristic for cost management, adapting to workloads and minimizing public cloud usage, achieving cost reductions [93]. Transitioning from monolithic architectures to microservices and leveraging edge computing alongside cloud resources allows organizations to operate efficiently, addressing modern IT infrastructure demands. This approach enhances responsiveness and resource utilization by processing data closer to its source, improving global service quality and availability [9, 27].

7.2 Dynamic Resource Allocation

Dynamic resource allocation in hybrid clouds involves real-time adjustments to meet fluctuating workloads, ensuring optimal performance and cost efficiency. This process leverages real-time analytics, policy-driven automation, and advanced algorithms to allocate resources dynamically [65]. The OLM method employs statistical models for overbooking serverless instances, optimizing resource utilization while ensuring service availability [35].

Frameworks like Ursa and ROMA highlight dynamic allocation's effectiveness, offering real-time adjustments tailored to user load and SLA requirements. Ursa accelerates resource allocation, enhancing control plane responsiveness significantly compared to traditional methods. ROMA optimizes resource orchestration in complex environments, achieving savings in compute and network bandwidth while maintaining performance [59, 94, 65, 95, 87].

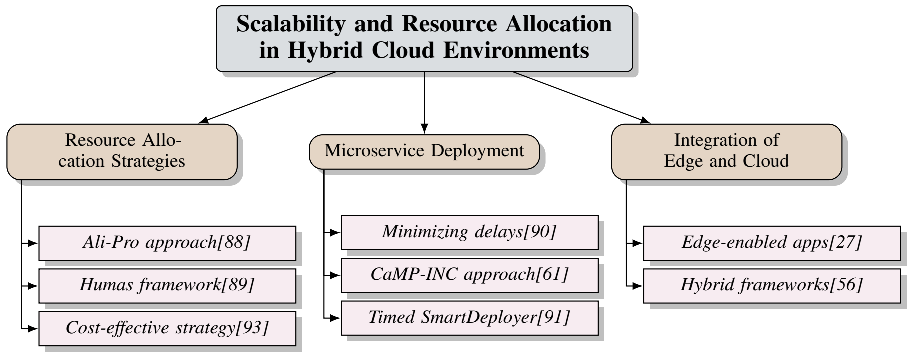

Figure 7: This figure illustrates the hierarchical structure of scalability and resource allocation strategies in hybrid cloud environments, highlighting key approaches in resource allocation, microservice deployment, and the integration of edge and cloud resources.

Innovative tools like the SWITCH workbench support dynamic allocation, enabling organizations to adapt to changing workloads [96]. The Reclaimer framework uses reinforcement learning to adjust CPU allocations, optimizing resource use and meeting QoS requirements [51]. AI-driven frameworks proposed by Barua et al. enhance resource utilization and minimize costs, ensuring efficient allocation [59]. The INDIGO-DataCloud Orchestrator exemplifies dynamic allocation by managing configurations and scaling based on user needs [20].

Turin et al. discuss a sampling strategy for service loads to derive cost tables, underscoring dynamic allocation's benefits in hybrid clouds [97]. Future research could expand frameworks to include dynamic tolerance for deadline violations and optimize dual cost/makespan objectives, as shown by Skedulix [70]. Dynamic resource allocation is vital for effective management in hybrid clouds, enabling automatic adjustments based on demand and usage patterns. This adaptability optimizes cost and performance, enhancing reliability, scalability, and agility in IT operations across multiple cloud providers [86, 85, 65, 98]. Leveraging real-time data, AI-driven frameworks, and reinforcement learning optimizes resource use, enhances performance, and reduces costs, achieving a more agile IT infrastructure.

7.3 Optimization Techniques

Optimization techniques in hybrid clouds focus on enhancing resource allocation efficiency and minimizing costs through dynamic and intelligent strategies. Central to these efforts are dynamic allocation techniques, adjusting resources based on real-time demand and usage patterns [65]. These techniques maintain optimal performance while minimizing unnecessary expenditures.

The Reclaimer framework exemplifies effective strategies by using reinforcement learning to enhance resource allocation for cloud microservices, achieving 100% QoS compliance while significantly reducing CPU allocations compared to traditional methods [51]. This not only ensures compliance with service-level objectives but also highlights AI-driven approaches' potential in optimizing resource management.

Integrating predictive analytics into frameworks allows proactive resource adjustments, optimizing performance and reducing costs. Leveraging historical data and real-time analytics, microservices-based frameworks predict demand fluctuations and dynamically allocate resources, minimizing over-provisioning and underutilization risks. Advanced machine learning models optimize management, ensuring each microservice is provisioned appropriately to maintain high QoS while adhering to SLAs. This proactive approach enhances efficiency, responsiveness, and reliability, improving customer satisfaction and performance across applications, including travel reservation systems and 5G technologies [99, 95, 87, 56, 63]. This predictive capability is vital for sustaining operations in dynamic environments with significant workload variability.

Advanced optimization techniques enhance resource allocation efficiency and achieve cost reductions, enabling effective workload distribution, minimizing latency, and streamlining management. Policy-based management and cross-cloud load balancing help navigate hybrid infrastructures' complexities while maintaining security and compliance, maximizing performance and operational efficiency

[50, 86, 98]. Utilizing dynamic allocation, reinforcement learning, and predictive analytics enhances strategies, ensuring peak efficiency in cloud infrastructures while minimizing expenses.

8 Case Studies and Applications

8.1 Case Studies and Real-World Applications

The deployment and management of microservices in hybrid cloud environments have been extensively examined through various real-world applications, revealing both advantages and challenges. A significant case study demonstrates the deployment of microservices on Kubernetes clusters, where experiments in a Docker/Kubernetes setting maintained a target queue size of 25 requests with exponentially distributed request arrival and processing times, showcasing the efficiency of microservices in managing dynamic workloads [76]. Another notable example involves orchestrating serverless functions within a simulated hybrid cloud environment, which processed batch jobs utilizing data from a storage system. This study highlighted microservices' potential to optimize resource allocation and performance in hybrid cloud settings [18].

The DeathStarBench benchmark's application of microservices in Social Network and Hotel Reservation systems further illustrates their practical deployment. Conducted on a five-node Kubernetes cluster, these experiments demonstrated the scalability and resource optimization capabilities inherent to microservices architectures [33]. In industrial contexts, case studies on a liqueur plant and the Gregor chair assembly system employed a combination of primitive and composite cyber-physical microservices, showcasing the integration of IoT technologies in managing complex industrial processes [2]. Additionally, a migration study from monolithic to microservices architectures utilized a dataset capturing interactions between domain entities, emphasizing the importance of understanding functional accesses during execution for a smooth transition [100]. A simulated deployment of an e-commerce application on a Kubernetes cluster with a workload of 100 virtual users further exemplifies microservices' adaptability and efficiency in high-demand scenarios [1].

These case studies collectively underscore the transformative potential of microservices in enhancing the flexibility, scalability, and efficiency of cloud-based applications across various industries. They offer valuable insights into best practices for implementation and highlight the importance of real-world evaluations in understanding microservices' impact on modern IT infrastructures [101].

8.2 Industry-Specific Applications and Insights

Microservices architecture in hybrid cloud environments provides significant advantages across various industries by enhancing scalability, flexibility, and operational efficiency. In the financial sector, microservices facilitate rapid deployment of new services, enabling institutions to swiftly respond to market changes and customer demands. Their modular nature allows for independent scaling, crucial for managing high transaction volumes typical in financial operations [51]. In healthcare, microservices support the integration of diverse applications and data sources, promoting interoperability and efficient data management. This architecture empowers healthcare providers to deliver personalized, real-time services, thereby improving patient care and operational efficiency. Deploying microservices across hybrid cloud environments ensures sensitive healthcare data remains secure while leveraging public cloud computational power for data processing and analytics [2].

Retailers benefit from microservices by enhancing the scalability and responsiveness of e-commerce platforms. They enable features like personalized recommendations and dynamic pricing, improving customer engagement and sales. The flexibility of hybrid cloud environments allows retailers to efficiently manage peak loads, ensuring a seamless shopping experience [1]. In manufacturing, microservices facilitate IoT device integration and edge computing, enabling real-time monitoring and control of production processes. This integration supports predictive maintenance and optimizes operations, reducing downtime and increasing efficiency. The use of microservices in hybrid cloud environments allows manufacturers to process data close to the source while maintaining centralized control and analytics capabilities [2].

The implementation of microservices across diverse industries highlights their transformative potential in improving operational efficiency, scalability, and innovation. This architecture enables organizations to decompose complex monolithic applications into smaller, independent services, facilitating easier maintenance and deployment. Furthermore, integrating microservices with edge

AI-generated, for reference only.

computing leverages low-latency capabilities, while cloud-based architectures optimize performance by addressing challenges such as response time and reliability. As companies transition from traditional legacy systems, they encounter opportunities and complexities, including the need for effective data management and extensive monitoring to ensure consistency and performance. The strategic adoption of microservices is reshaping business operations, driving modernization and competitive advantage [9, 22, 101, 43]. These architectures enable industries to remain agile and competitive in a rapidly evolving technological landscape, driving growth and improving service delivery.

8.3 Performance Evaluation in Hybrid Cloud Setups

Benchmark	Size	Domain	Task Format	Metric
DSB[102]	1,000,000	Cloud Computing	Microservices Performance Evaluation	Tail Latency, Throughput
MiMMo[103]	30	Software Engineering	Maturity Assessment	Maturity Level
K8s-AB[48]	300	Microservice Availability	Availability Evaluation	Outage Time, Recovery Time
TrainTicket[3]	30	Microservices	Debugging	Debugging Time

Table 2: This table provides an overview of representative benchmarks used in evaluating microservices performance, focusing on various domains such as cloud computing, software engineering, and microservice availability. It includes details on the size of each benchmark, the domain it pertains to, the task format, and the performance metrics used, offering a comprehensive perspective on the tools available for performance evaluation in hybrid cloud setups.

Evaluating microservices performance in hybrid cloud environments necessitates a comprehensive analysis of various deployment configurations, focusing on metrics such as system efficiency, response times, scalability, and cost-effectiveness. This assessment identifies advantages and challenges associated with cloud-based microservices architectures while exploring potential bottlenecks and optimization techniques. Utilizing frameworks like queuing models and benchmarking methods allows researchers to quantify the impacts of containerization and software-defined networking on performance, guiding system designers to achieve a balance between application performance, deployment costs, and sustainability goals [104, 58, 105, 101, 23]. The integration of microservices within hybrid cloud environments requires careful consideration of both on-premises and public cloud resources, as performance can vary significantly based on infrastructure and network conditions.

A comprehensive evaluation of database performance in hybrid cloud environments reveals that MongoDB and MySQL efficiently handle workloads while maintaining low latency levels. Conversely, databases such as Cassandra, Riak, CouchDB, and Redis encounter challenges when extensively leveraging public cloud resources, primarily due to increased latency issues [106]. This underscores the importance of selecting appropriate database technologies that align with the specific requirements of hybrid cloud architectures.

The dataset from Chadha et al. offers valuable insights into performance metrics and cost implications of various deployment strategies, including request response times, the number of successful requests, and cost per 1000 requests. These metrics are crucial for understanding the trade-offs between performance and cost in hybrid cloud setups, enabling organizations to optimize their deployment strategies for maximum efficiency [78]. PerfSim, a performance simulator for cloud-native microservices, provides a practical tool for evaluating performance in hybrid cloud environments. Tested on a real Kubernetes cluster with 104 prevalent scenarios, it offers a reliable comparison between simulation results and actual performance metrics [79]. This simulator assists in predicting performance outcomes and identifying potential bottlenecks, facilitating informed decision-making in microservices deployment.

The evaluation of microservices performance in hybrid cloud environments highlights the essential influence of infrastructure selection, deployment strategies, and simulation tools on enhancing system performance and cost-efficiency. This assessment reveals that careful planning and design are crucial for optimizing response time, throughput, scalability, and reliability in cloud-based microservices architectures. Additionally, the analysis emphasizes the benefits of transitioning from traditional monolithic systems to microservices, demonstrating improved service reusability and flexibility while addressing potential bottlenecks and monitoring challenges. Employing queuing models to evaluate efficiency indicates that decomposing large services into microservices can significantly reduce overall processing time, particularly under optimal conditions [104, 101]. By leveraging these insights, organizations can enhance their hybrid cloud implementations, ensuring robust and scalable microservices architectures. Table 2 presents a selection of benchmarks that are instrumental in assessing microservices performance within hybrid cloud environments, highlighting their relevance in different domains and the metrics used for evaluation.

9 Future Directions and Research Opportunities

Advancements in microservices architecture and hybrid cloud environments present both challenges and opportunities, setting the stage for future research. This section outlines critical directions and opportunities that will influence the evolution of this domain, providing a framework to guide researchers and practitioners in navigating microservices deployment and optimization complexities. The following subsection addresses anticipated research trends poised to influence microservices in the coming years.

9.1 Future Research and Trends

Future research in microservices and hybrid cloud environments will focus on scalability, security, and the integration of intelligent systems and IoT applications. Key areas include optimizing microservices integration at the field device level and exploring new IoT communication protocols [2]. Developing dynamic service distribution strategies and advanced feature extraction algorithms is crucial for enhancing data processing capabilities [30]. The integration of development tools and collaborative coding within cloud environments is vital for advancing cloud adoption, particularly concerning organizational culture [16]. Enhancements in algorithm efficiency through parallel processing and high-performance programming languages could significantly improve online deployment algorithms [11]. Additionally, refining reliability metrics, integrating predictive models for anticipatory scaling, and extending approaches to include more microservices are vital [1].

Research will also target granular decomposition tasks, including functions and data tables, influencing microservices evolution in hybrid cloud environments [107]. Improving frameworks like DAPP-ECC to adapt to varying network conditions and incorporating predictive algorithms for user mobility presents promising avenues [14]. Furthermore, porting benchmarks to Temporal and enhancing debugging and orchestration techniques will be key focuses [3]. Long-term investigations into harmful anti-patterns and the development of empirically-based guidelines are necessary for advancing microservices practices [12]. Refining statistical models for execution time estimation and integrating OLM with instance-placement strategies are critical for optimizing resource allocation [35].

Future research should also extend decision models, establish repositories, and develop automated pattern selection recommendation systems [13]. Evaluating existing CMDSE approaches for team-internal collaboration, extending models for distributed modeling, and investigating runtime models' roles in the MSA life cycle will be essential [7]. Additionally, refining integration processes, exploring automated dependency management solutions, and developing monitoring and debugging tools for distributed systems are crucial areas for exploration [8]. Enhancing management system adaptability, exploring edge computing trends, and developing comprehensive frameworks addressing security and reliability in hybrid infrastructures are key research priorities [4]. Future research will focus on refining methodologies based on user feedback and expanding tool support to enhance functionality and engagement [15].

These trends highlight the need for continued innovation in microservices and hybrid clouds. Addressing prevalent challenges, such as technical debt, continuous integration and delivery issues, and security vulnerabilities, will significantly enhance microservices' operational capabilities and efficiency, facilitating advancements in modern IT infrastructures as AI techniques and edge computing evolve [9, 6, 108, 43].

9.2 Emerging Trends and Future Directions

Emerging trends in microservices architecture are increasingly shaped by the rapid evolution of technology platforms, particularly in automation and tool support within the DevOps context [24]. This shift aims to enhance cloud application efficiency and scalability, emphasizing microservice granularity for performance optimization [109]. Organizations are showing growing interest in frameworks that provide benchmarks for assessing maturity and guiding architectural decisions [103].

The complexity of microservices deployments necessitates deeper exploration of anti-patterns and their impacts on system performance and maintainability. Despite existing taxonomies, the harmfulness of certain anti-patterns requires empirical validation to refine best practices [12]. Addressing these gaps is crucial for ensuring that microservices architectures remain robust and adaptable amid evolving technological demands.

The integration of edge computing with microservices significantly enhances data processing capabilities and reduces latency, as computations occur closer to data sources, minimizing cloud data transfer. This synergy leverages edge computing's low-latency advantages and microservices' high throughput, enabling novel applications and improving overall system responsiveness. A systematic literature review highlights various architectural approaches, microservice composition techniques, and offloading methods to maximize integration benefits, addressing research gaps and trends [27, 43]. This trend is likely to drive further research into resource allocation optimization and innovative orchestration techniques for seamless service integration across hybrid cloud environments, enhancing organizational agility and resilience.

9.3 Research Agenda and Best Practices

The future research agenda in microservices architecture and hybrid cloud environments should prioritize developing tools that enhance architectural quality assessments and gather developer feedback to better understand their needs [110]. This alignment with industry professionals' practical requirements will facilitate effective microservices architecture implementation.

Security remains a critical concern in microservices deployment, necessitating comprehensive ontologies that articulate relationships between security threats and mechanisms across architectural layers and platforms [111]. Such ontologies will serve as foundational frameworks for identifying and mitigating risks, ensuring the integrity of microservices in diverse cloud environments.

The integration of various orchestration technologies presents ongoing challenges, with significant questions about optimal implementation strategies across diverse cloud architectures [80]. Addressing these questions will require focused research on best practices for orchestration, including the development of standardized frameworks that promote interoperability and efficiency.

Best practices should include adopting agile methodologies that emphasize iterative development and continuous integration, enabling rapid adaptation to changing requirements and technological advancements. Cultivating a collaborative culture among developers, security experts, and operations teams is crucial for identifying and mitigating security threats. Systematic mappings of security mechanisms for microservices highlight the need for comprehensive strategies addressing both external and internal attacks across various architectural layers. By collaborating, these teams can enhance the scalability and maintainability of complex applications while effectively addressing evolving security challenges [9, 111].

The proposed research agenda focuses on enhancing the scalability, security, and efficiency of microservices architectures by addressing challenges such as technical debt, data consistency, and security threats, fostering innovation and resilience in modern IT infrastructures. This initiative draws upon a curated dataset of open-source microservices projects, systematic literature reviews on security mechanisms and edge computing integration, and empirical studies highlighting common developer issues, ultimately providing a comprehensive framework for robust microservices systems development and maintenance [40, 22, 111, 108, 43]. By focusing on these key areas, future research can significantly advance microservices and their integration within hybrid cloud environments.

10 Conclusion

This survey underscores the pivotal impact of microservices and hybrid cloud environments on contemporary IT infrastructures, particularly in enhancing scalability, flexibility, and operational efficiency. By disaggregating monolithic systems into modular, independent units, microservices architecture fosters adaptability and efficiency, which are essential for the development and deployment of cloud-native applications. This approach enables organizations to leverage both on-premises and public cloud resources, achieving significant cost savings and operational flexibility. The complexity inherent in managing distributed microservices is effectively addressed through advanced service orchestration, which integrates and coordinates services across diverse infrastructures, optimizing deployment and resource utilization. Despite advancements, there remains a gap between theoretical models and practical tool usability, highlighting the necessity for further empirical validation and tool refinement. Resource allocation strategies are critical for optimizing hybrid cloud performance, ensuring service quality and reliability amidst fluctuating workloads. The importance of strategic planning in migration processes is evident, with robust planning, security measures, and continuous monitoring being vital for successful hybrid cloud implementations. Continued research is essential to advance microservices and hybrid cloud capabilities, with a focus on refining architectural designs and exploring AI-driven enhancements to improve operational efficiency. Addressing these areas will significantly enhance the agility and competitiveness of IT infrastructures in an ever-evolving technological landscape.

AI-generated, for reference only.

References

[1] Nazanin Akhtarian, Hamzeh Khazaei, and Marin Litoiu. Employing software diversity in cloud microservices to engineer reliable and performant systems, 2024.

[2] Kleanthis Thramboulidis, Danai C. Vachtsevanou, and Alexandros Solanos. Cyber-physical microservices: An iot-based framework for manufacturing systems, 2018.

[3] Anas Nadeem and Muhammad Zubair Malik. A case for microservices orchestration using workflow engines, 2022.

[4] Pedro Nobre Oliveira dos Santos. Towards an automatic microservices manager for hybrid cloud edge environments. Master's thesis, Universidade NOVA de Lisboa (Portugal), 2022.

[5] Thomas Pusztai, Stefan Nastic, Andrea Morichetta, Víctor Casamayor Pujol, Philipp Raith, Schahram Dustdar, Deepak Vij, Ying Xiong, and Zhaobo Zhang. Polaris scheduler: Slo-and topology-aware microservices scheduling at the edge. In *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*, pages 61–70. IEEE, 2022.

[6] Sergio Moreschini, Shahrzad Pour, Ivan Lanese, Daniel Balouek-Thomert, Justus Bogner, Xiaozhou Li, Fabiano Pecorelli, Jacopo Soldani, Eddy Truyen, and Davide Taibi. Ai techniques in the microservices life-cycle: A systematic mapping study, 2025.

[7] Jonas Sorgalla, Florian Rademacher, Sabine Sachweh, and Albert Zündorf. On collaborative model-driven development of microservices, 2018.

[8] Maximillien de Bayser, Vinicius Segura, Leonardo Guerreiro Azevedo, Leonardo P. Tizzei, Raphael Melo Thiago, Elton Soares, and Renato Cerqueira. Devops and microservices in scientific system development, 2021.

[9] Alessandra Levcovitz, Ricardo Terra, and Marco Tulio Valente. Towards a technique for extracting microservices from monolithic enterprise systems, 2016.

[10] Meryam Chaieb and Mohamed Aymen Saied. Automate migration to microservices architecture using machine learning techniques, 2023.

[11] Xiang He, Zhiying Tu, Markus Wagner, Xiaofei Xu, and Zhongjie Wang. Online deployment algorithms for microservice systems with complex dependencies, 2021.

[12] Davide Taibi, Valentina Lenarduzzi, and Claus Pahl. Microservices anti patterns: A taxonomy, 2019.

[13] Muhammad Waseem, Peng Liang, Aakash Ahmad, Mojtaba Shahin, Arif Ali Khan, and Gastón Márquez. Decision models for selecting patterns and strategies in microservices systems and their evaluation by practitioners, 2022.

[14] Itamar Cohen, Paolo Giaccone, and Carla Fabiana Chiasserini. Asynchronous distributed protocol for service provisioning in the edge-cloud continuum, 2023.

[15] Jonas Fritzsch, Justus Bogner, Markus Haug, Stefan Wagner, and Alfred Zimmermann. Towards an architecture-centric methodology for migrating to microservices, 2022.

[16] Naimil Navnit Gadani. Hybrid cloud strategies for enterprise software development: A comparative study. *INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING AND TECHNOLOGY (IJARET)*, 14(07):91–103, 2023.

[17] Rajkumar Buyya and Karthik Sukumar. Platforms for building and deploying applications for cloud computing, 2011.

[18] Aristotelis Peri, Michail Tsenos, and Vana Kalogeraki. Orchestrating the execution of serverless functions in hybrid clouds, 2024.

[19] Evgeniy Pluzhnik, Oleg Lukyanchikov, Evgeny Nikulchev, and Simon Payain. Designing applications with distributed databases in a hybrid cloud, 2015.

[20] Miguel Caballer, Sahdev Zala, Álvaro López García, Germán Moltó, Pablo Orviz Fernández, and Mathieu Velten. Orchestrating complex application architectures in heterogeneous clouds, 2017.

[21] Florian Rademacher, Sabine Sachweh, and Albert Zündorf. Analysis of service-oriented modeling approaches for viewpoint-specific model-driven development of microservice architecture, 2018.

[22] Vinicius L. Nogueira, Fernando S. Felizardo, Aline M. M. M. Amaral, Wesley K. G. Assuncao, and Thelma E. Colanzi. Insights on microservice architecture through the eyes of industry practitioners, 2024.

[23] Nane Kratzke. About microservices, containers and their underestimated impact on network performance, 2017.

[24] Davide Taibi, Valentina Lenarduzzi, and Claus Pahl. Continuous architecting with microservices and devops: A systematic mapping study, 2019.

[25] Hemanth Gopal, Guanqun Song, and Ting Zhu. Security, privacy and challenges in microservices architecture and cloud computing- survey, 2022.

[26] MGW Blok. Private legacy to cloud: A tailored migration method for private cloud deployment of legacy software projects. Master's thesis, University of Twente, 2024.

[27] João Leitão, Pedro Ákos Costa, Maria Cecília Gomes, and Nuno Preguiça. Towards enabling novel edge-enabled applications, 2019.

[28] Manuel Mazzara, Kevin Khanda, Ruslan Mustafin, Victor Rivera, Larisa Safina, and Alberto Sillitti. Microservices science and engineering, 2017.

[29] Yu Gan and Christina Delimitrou. The architectural implications of microservices in the cloud, 2018.

[30] Badraddin Alturki, Stephan Reiff-Marganiec, Charith Perera, and Suparna De. Exploring the effectiveness of service decomposition in fog computing architecture for the internet of things, 2019.

[31] Adel Nadjaran Toosi, Richard O Sinnott, and Rajkumar Buyya. Resource provisioning for data-intensive applications with deadline constraints on hybrid clouds using aneka. *Future Generation Computer Systems*, 79:765–775, 2018.

[32] Volodymyr Chernetskyi. Container-based microservice placement optimization in cloud.

[33] Nader Alfares and George Kesidis. Container sizing for microservices with dynamic workload by online optimization. In *Proceedings of the 9th International Workshop on Container Technologies and Container Clouds*, pages 1–6, 2023.

[34] Vighnesh Sachidananda and Anirudh Sivaraman. Collective autoscaling for cloud microservices, 2022.

[35] George Kesidis. Overbooking microservices in the cloud, 2019.

[36] Haoran Qiu, Subho S. Banerjee, Saurabh Jha, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. Firm: An intelligent fine-grained resource management framework for slo-oriented microservices, 2020.

[37] Meryam chaieb and Mohamed Aymen Saied. Migration to microservices: A comparative study of decomposition strategies and analysis metrics, 2024.

[38] Rodrigo Laigner, Yongluan Zhou, Marcos Antonio Vaz Salles, Yijian Liu, and Marcos Kalinowski. Data management in microservices: State of the practice, challenges, and research directions, 2021.

[39] Emilien Wansart, Maxime Goffart, Justin Iurman, and Benoit Donnet. Mstg: A flexible and scalable microservices infrastructure generator, 2024.

[40] Mohammad Imranur, Rahman, Sebastiano Panichella, and Davide Taibi. A curated dataset of microservices-based systems, 2019.

[41] Marco Capuccini, Anders Larsson, Matteo Carone, Jon Ander Novella, Noureddin Sadawi, Jianliang Gao, Salman Toor, and Ola Spjuth. On-demand virtual research environments using microservices, 2019.

[42] Manuel Mazzara, Ruslan Mustafin, Larisa Safina, and Ivan Lanese. Towards microservices and beyond: An incoming paradigm shift in distributed computing, 2016.

[43] Lucas Fernando Souza de Castro and Sandro Rigo. Relating edge computing and microservices by means of architecture approaches and features, orchestration, choreography, and offloading: A systematic literature review, 2023.

[44] Ruoyu Su and Xiaozhou Li. Modular monolith: Is this the trend in software architecture?, 2024.

[45] Prathamesh Muzumdar, Amol Bhosale, Ganga Prasad Basyal, and George Kurian. Navigating the docker ecosystem: A comprehensive taxonomy and survey, 2024.

[46] Dimitrios Kallergis, Zacharenia Garofalaki, Georgios Katsikogiannis, and Christos Douligeris. Capodaz: A containerised authorisation and policy-driven architecture using microservices, 2020.

[47] Armin Balalaie, Abbas Heydarnoori, and Pooyan Jamshidi. Migrating to cloud-native architectures using microservices: An experience report, 2015.

[48] Leila Abdollahi Vayghan, Mohamed Aymen Saied, Maria Toeroe, and Ferhat Khendek. Kubernetes as an availability manager for microservice applications, 2019.

[49] Momil Seedat, Qaisar Abbas, and Nadeem Ahmad. Systematic mapping of monolithic applications to microservices architecture, 2023.

[50] Anshul Sharma. Optimizing hybrid cloud architectures: A comprehensive study of performance engineering best practices. *INTERNATIONAL JOURNAL OF ENGINEERING AND TECHNOLOGY RESEARCH (IJETR)*, 9(2):288–299, 2024.

[51] Quintin Fettes, Avinash Karanth, Razvan Bunescu, Brandon Beckwith, and Sreenivas Subramoney. Reclaimer: A reinforcement learning approach to dynamic resource allocation for cloud microservices, 2023.

[52] Priyanka Billawa, Anusha Bambhore Tukaram, Nicolás E. Díaz Ferreyra, Jan-Philipp Steghöfer, Riccardo Scandariato, and Georg Simhandl. Sok: Security of microservice applications: A practitioners' perspective on challenges and best practices, 2022.

[53] Mostafa Noshy, Abdelhameed Ibrahim, and Hesham Arafat Ali. Optimization of live virtual machine migration in cloud computing: A survey and future directions. *Journal of Network and Computer Applications*, 110:1–10, 2018.

[54] Leila Abdollahi Vayghan, Mohamed Aymen Saied, Maria Toeroe, and Ferhat Khendek. A kubernetes controller for managing the availability of elastic microservice based stateful applications, 2020.

[55] Hai Dinh-Tuan and Felix Beierle. Ms2m: A message-based approach for live stateful microservices migration, 2022.

[56] Zeina Houmani, Daniel Balouek-Thomert, Eddy Caron, and Manish Parashar. Enabling microservices management for deep learning applications across the edge-cloud continuum. In *2021 IEEE 33rd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 137–146. IEEE, 2021.

[57] Minxian Xu, Adel N. Toosi, and Rajkumar Buyya. A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing, 2020.

[58] Vittorio Cortellessa, Daniele Di Pompeo, and Michele Tucci. Exploring sustainable alternatives for the deployment of microservices architectures in the cloud, 2024.

[59] Biman Barua and M. Shamim Kaiser. Ai-driven resource allocation framework for microservices in hybrid cloud platforms, 2024.

[60] Vittorio Cortellessa, Daniele Di Pompeo, Romina Eramo, and Michele Tucci. A model-driven approach for continuous performance engineering in microservice-based systems, 2023.

[61] Soukaina Ouledsidi Ali, Halima Elbiaze, Roch Glitho, and Wessam Ajib. Camp-inc: Components-aware microservices placement for in-network computing cloud-edge continuum, 2023.

[62] Simon Bliudze, Giuseppe De Palma, Saverio Giallorenzo, Ivan Lanese, Gianluigi Zavattaro, and Brice Arleon Zemtsop Ndadji. Adaptable teastore, 2024.

[63] Biman Barua and M. Shamim Kaiser. Microservices-based framework for predictive analytics and real-time performance enhancement in travel reservation systems, 2024.

[64] Biman Barua and M. Shamim Kaiser. Real-time performance optimization of travel reservation systems using ai and microservices, 2024.

[65] Naseemuddin Mohammad. Dynamic resource allocation techniques for optimizing cost and performance in multi-cloud environments. *International Journal of Cloud Computing (IJCC)*, 1(1):1–12, 2023.

[66] Rajinder Sandhu, Adel Nadjaran Toosi, and Rajkumar Buyya. An api for development of user-defined scheduling algorithms in aneka paas cloud software: User defined schedulers in aneka paas cloud software, 2019.

[67] Hussain Ahmad, Christoph Treude, Markus Wagner, and Claudia Szabo. Smart hpa: A resource-efficient horizontal pod auto-scaler for microservice architectures, 2024.

[68] Thomas Pusztai, Fabiana Rossi, and Schahram Dustdar. Pogonip: Scheduling asynchronous applications on the edge. In *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, pages 660–670. IEEE, 2021.

[69] Linfeng Wen, Minxian Xu, Sukhpal Singh Gill, Muhammad Hafizhuddin Hilman, Satish Narayana Srirama, Kejiang Ye, and Chengzhong Xu. Statuscale: Status-aware and elastic scaling strategy for microservice applications, 2024.

[70] Anirban Das, Andrew Leaf, Carlos A Varela, and Stacy Patterson. Skedulix: Hybrid cloud scheduling for cost-efficient execution of serverless applications. In *2020 IEEE 13th international conference on cloud computing (CLOUD)*, pages 609–618. IEEE, 2020.

[71] Md Rajib Hossen, Mohammad A. Islam, and Kishwar Ahmed. Practical efficient microservice autoscaling with qos assurance, 2022.

[72] Padmaveni Krishnan and D John Aravindhar. Self-adaptive pso memetic algorithm for multi objective workflow scheduling in hybrid cloud. *Int. Arab J. Inf. Technol.*, 16(5):928–935, 2019.

[73] Ataollah Fatahi Baarzi. Efficient service deployment on public cloud: A cost, performance, and security perspective. 2022.

[74] Tomas Cerny, Amr S. Abdelfattah, Vincent Bushong, Abdullah Al Maruf, and Davide Taibi. Microvision: Static analysis-based approach to visualizing microservices in augmented reality, 2022.

[75] Yanqi Zhang, Weizhe Hua, Zhuangzhuang Zhou, Ed Suh, and Christina Delimitrou. Sinan: Data driven resource management for cloud microservices, 2021.

[76] Mathieu Simon, Alessandro Spallina, Loic Dubocquet, and Andrea Araldo. Parsimonious edge computing to reduce microservice resource usage, 2021.

[77] Hai Dinh-Tuan, Katerina Katsarou, and Patrick Herbke. Optimizing microservices with hyperparameter optimization, 2022.

[78] Mohak Chadha, Victor Pacyna, Anshul Jindal, Jianfeng Gu, and Michael Gerndt. Migrating from microservices to serverless: An iot platform case study, 2022.

[79] Michel Gokan Khan, Javid Taheri, Auday Al-Dulaimy, and Andreas Kassler. Perfsim: A performance simulator for cloud native microservice chains, 2021.

[80] Alberto Giaretta, Nicola Dragoni, and Manuel Mazzara. Joining jolie to docker - orchestration of microservices on a containers-as-a-service layer, 2017.

[81] Mahyar Karimi and Ahmad Abdollahzadeh Barfroush. Proposing a dynamic executive microservices architecture model for ai systems, 2023.

[82] Stanislaw Ambroszkiewicz and Waldemar Bartyna. A simple protocol to automate the executing, scaling, and reconfiguration of cloud-native apps, 2024.

[83] Marcela S. Melara and Mic Bowman. Enabling security-oriented orchestration of microservices, 2021.

[84] Amr S. Abdelfattah. Student research abstract: Microservices-based systems visualization, 2022.

[85] Hanan Shukur, Subhi Zeebaree, Rizgar Zebari, Diyar Zeebaree, Omar Ahmed, and Azar Salih. Cloud computing virtualization of resources allocation for distributed systems. *Journal of Applied Science and Technology Trends*, 1(2):98–105, 2020.

[86] Tummalachervu Chaitanya Kanth. Efficient strategies for seamless cloud migrations using advanced deployment automations. 2023.

[87] Anousheh Gholami, Kunal Rao, Wang-Pin Hsiung, Oliver Po, Murugan Sankaradas, and Srimat Chakradhar. Roma: Resource orchestration for microservices-based 5g applications, 2022.

[88] Minxian Xu, Lei Yang, Yang Wang, Chengxi Gao, Linfeng Wen, Guoyao Xu, Liping Zhang, Kejiang Ye, and Chengzhong Xu. Practice of alibaba cloud on elastic resource provisioning for large-scale microservices cluster, 2023.

[89] Qin Hua, Dingyu Yang, Shiyou Qian, Jian Cao, Guangtao Xue, and Minglu Li. Humas: A heterogeneity- and upgrade-aware microservice auto-scaling framework in large-scale data centers, 2024.

[90] Yuxiang Liu, Bo Yang, Yu Wu, Cailian Chen, and Xinping Guan. How to share: Balancing layer and chain sharing in industrial microservice deployment, 2022.

[91] Lorenzo Bacchiani, Mario Bravetti, Saverio Giallorenzo, Jacopo Mauro, Iacopo Talevi, and Gianluigi Zavattaro. Microservice dynamic architecture-level deployment orchestration (extended version), 2021.

[92] Biman Barua and M. Shamim Kaiser. Optimizing airline reservation systems with edge-enabled microservices: A framework for real-time data processing and enhanced user responsiveness, 2024.

[93] Valentino Armani, Francescomaria Faticanti, Silvio Cretti, Seungwoo Kum, and Domenico Siracusa. A cost-effective workload allocation strategy for cloud-native edge services, 2021.

[94] Kan Hu, Minxian Xu, Kejiang Ye, and Chengzhong Xu. Lsram: A lightweight autoscaling and slo resource allocation framework for microservices based on gradient descent, 2024.

[95] Yanqi Zhang, Zhuangzhuang Zhou, Sameh Elnikety, and Christina Delimitrou. Analytically-driven resource management for cloud-native microservices, 2024.

[96] Polona Štefanič, Matej Cigale, Andrew C Jones, Louise Knight, Ian Taylor, Cristiana Istrate, George Suciu, Alexandre Ulisses, Vlado Stankovski, Salman Taherizadeh, et al. Switch workbench: A novel approach for the development and deployment of time-critical microservice-based cloud-native applications. *Future Generation Computer Systems*, 99:197–212, 2019.

[97] Gianluca Turin, Andrea Borgarelli, Simone Donetti, Ferruccio Damiani, Einar Broch Johnsen, and Silvia Lizeth Tapia Tarifa. Predicting resource consumption of kubernetes container systems using resource models, 2023.

[98] Evgeniy Pluzhnik, Evgeniy Nikulchev, and Simon Payain. Optimal control of applications for hybrid cloud services, 2014.

[99] Amadou Ba, Pavithra Harsha, and Chitra Subramanian. Leveraging interpretability in the transformer to automate the proactive scaling of cloud resources, 2024.

[100] Bernardo Andrade, Samuel Santos, and António Rito Silva. From monolith to microservices: Static and dynamic analysis comparison, 2022.

[101] Ganesh Chowdary Desina. Evaluating the impact of cloud-based microservices architecture on application performance, 2023.

[102] Yu Gan, Yanqi Zhang, Dailun Cheng, Ankitha Shetty, Priyal Rathi, Nayan Katarki, Ariana Bruno, Justin Hu, Brian Ritchken, Brendon Jackson, Kelvin Hu, Meghna Pancholi, Yuan He, Brett Clancy, Chris Colen, Fukang Wen, Catherine Leung, Siyuan Wang, Leon Zaruvinsky, Mateo Espinosa, Rick Lin, Zhongling Liu, Jake Padilla, and Christina Delimitrou. An open-source benchmark suite for cloud and iot microservices, 2019.

[103] Jean-Philippe Gouigoux, Dalila Tamzalit, and Joost Noppen. Microservice maturity of organizations: towards an assessment framework, 2021.

[104] Abel C. H. Chen. Research on efficiency analysis of microservices, 2023.

[105] Marcelo Amaral, Jordà Polo, David Carrera, Iqbal Mohomed, Merve Unuvar, and Malgorzata Steinder. Performance evaluation of microservices architectures using containers, 2015.

[106] Yaser Mansouri, Victor Prokhorenko, and M. Ali Babar. An automated implementation of hybrid cloud for performance evaluation of distributed databases, 2020.

[107] Alex Mathai, Sambaran Bandyopadhyay, Utkarsh Desai, and Srikanth Tamilselvam. Monolith to microservices: Representing application software through heterogeneous graph neural network, 2022.

[108] Muhammad Waseem, Peng Liang, Aakash Ahmad, Arif Ali Khan, Mojtaba Shahin, Pekka Abrahamsson, Ali Rezaei Nasab, and Tommi Mikkonen. Understanding the issues, their causes and solutions in microservices systems: An empirical study, 2023.

[109] Dharmendra Shadija, Mo Rezai, and Richard Hill. Microservices: Granularity vs. performance, 2017.

[110] Tomas Cerny, Amr S. Abdelfattah, Vincent Bushong, Abdullah Al Maruf, and Davide Taibi. Microservice architecture reconstruction and visualization techniques: A review, 2022.

[111] Abdelhakim Hannousse and Salima Yahiouche. Securing microservices and microservice architectures: A systematic mapping study, 2020.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.