IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

THE EVOLUTIONARY PATH OF SERVERLESS COMPUTING TOWARDS EFFICIENT CLOUD SOLUTIONS

[1]Vidhya shree C, [2]Shrinidhi Hegde , [3]Ashwini K B, [4]Mamatha G S

[1,2]M.tech Students, [3]Associate Professor, [4]Professor

Information Science and Engineering Women in Cloud CoE in India

R V College of Engineering, Bangalore, India

Abstract: Serverless computing represents a significant shift in cloud computing, offering developers the ability to focus on application logic without worrying about underlying infrastructure. This paper provides a comprehensive exploration of serverless computing by examining its benefits, challenges, and diverse use cases. By analysing foundational and contemporary research, it highlights the cost efficiency, scalability, and operational simplicity provided by serverless architectures. The study also addresses key challenges such as resource management, vendor lock-in, and performance variability. Furthermore, it explores practical applications in areas like multimedia services, dynamic web hosting, and edge computing. The findings emphasize the transformative potential of serverless computing while exploring opportunities for future research and innovation.

***Index Terms* -** Serverless computing, Cloud computing, Application logic, Cost efficiency, Scalability, Operational simplicity, Resource management, Vendor lock-in, Performance variability, Multimedia services, Dynamic web hosting, Edge computing

1. INTRODUCTION

Cloud computing has revolutionized the way software applications are created, implemented, and expanded. Over the years, various service models such as Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) have emerged to simplify development processes. However, the introduction of serverless computing represents a paradigm shift by abstracting infrastructure management entirely, allowing developers to focus solely on application functionality [1].Serverless computing, often referred to as Function-as-a- Service (Faas), enables automatic scaling of applications and optimizes costs through a pay-as-you-go pricing model[2]. Unlike traditional cloud models, serverless platforms dynamically allocate resources based on the actual workload, ensuring efficiency and cost-effectiveness. This model is particularly advantageous for applications with variable workloads and event-driven architectures [3].

Despite its numerous benefits, serverless computing presents challenges such as vendor lock-in, performance variability, and resource management complexities [4], [5].Addressing these issues is critical for ensuring the long-term success and adoption of serverless architectures. This paper explores the key aspects of serverless computing, including its benefits, challenges, and practical applications across various domains. It delves into the underlying architecture of serverless platforms, highlighting how they facilitate automatic resource allocation and event- driven execution. Additionally, the paper examines performance considerations, such as cold starts and latency issues, along with strategies for optimizing execution efficiency. Security implications, including data privacy, access control, and compliance challenges, are also discussed. Furthermore, the study evaluates real-world use cases in industries like healthcare, finance, and IoT, showcasing how serverless computing enhances scalability, cost-effectiveness, and operational flexibility. By synthesizing insights from research and industry practices, this paper provides a comprehensive understanding of the evolving role of serverless architectures in modern computing environments.

Serverless Architecture Diagram

Figure1 Serverless Architecture Diagram showcases how the Serverless computing revolutionizes the way applications are built and deployed by enabling event-driven execution where user or API requests are processed dynamically, without the need to manage underlying server infrastructure. When a request is made, the API Gateway routes it to a serverless function, such as AWS Lambda or Azure Functions, which executes the necessary logic in a cloud-managed, isolated environment. These serverless functions can interact with various event sources like databases, storage solutions (e.g., S3, DynamoDB), and messaging services (e.g., Kafka, SNS) to retrieve or store data, enabling real-time data processing and seamless integrations. Cloud providers automatically provision and scale these execution environments based on demand, ensuring both cost efficiency and resource optimization, as users only pay for the compute resources they actually use. Nevertheless, certain challenges persist, including cold starts, performance variability, and the potential risk associated with vendor dependency lock-in, which necessitate efficient monitoring and troubleshooting through tools like AWS CloudWatch and Azure Monitor to maintain optimal performance. Studies have shown that serverless architectures are particularly effective in handling workloads from emerging technologies such as IoT, AI/ML, and data-intensive applications, offering enhanced scalability and flexibility. Nevertheless, concerns about latency and energy consumption persist, as the serverless model may introduce delays in response times during high traffic loads or require energy-intensive execution. Additionally, while serverless platforms reduce the overhead of infrastructure management, ensuring smooth integration with legacy systems and maintaining consistent performance across unpredictable traffic patterns remain crucial considerations. As organizations increasingly adopt serverless solutions, ongoing innovation and optimization will be essential to fully realize its potential in diverse application scenarios.

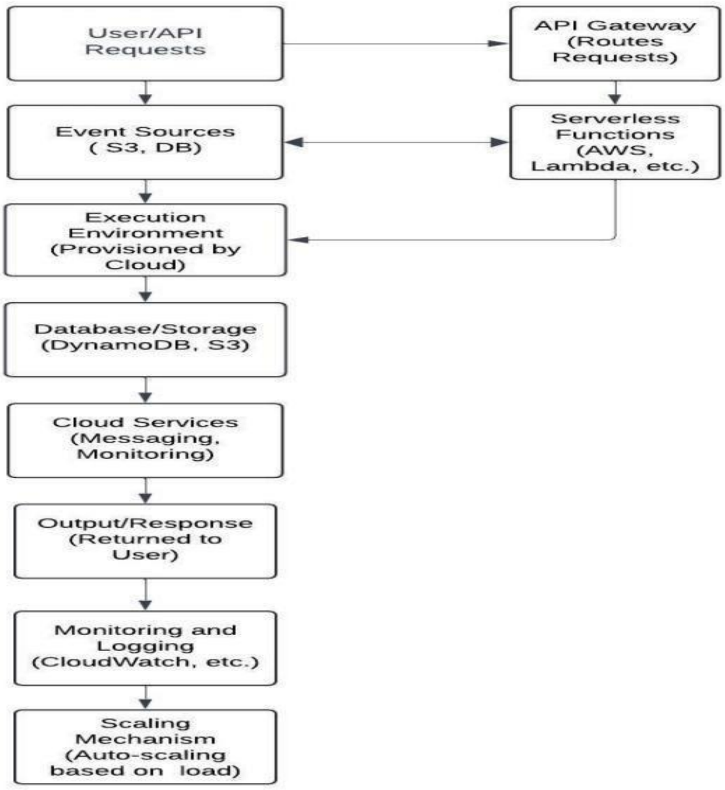

figure 1 : serverless architecture diagram

2. BENEFITS OF SERVERLESS COMPUTING

2.1 Cost Efficiency

The pay-as-you-go pricing model represents one of the most significant benefits of serverless computing. Unlike traditional hosting models, where resources are often over- provisioned to handle peak loads, serverless architectures charge users based on the exact execution time of their functions [3]. The cost for executing a function can be modelled as:

$C_{execution} = T_{execution} \, X \, C_{per\ time\ unit}$

Where $T_{execution}$ is the execution time of the function, and $C_{per\ time\ unit}$ is the cost per time unit of execution (e.g., per millisecond). This removes expenses related to idle resources, making serverless computing particularly appealing for startups and small businesses with limited budgets. Studies show that this model can reduce infrastructure costs by up to 80% for certain applications [4].

Moreover, serverless computing minimizes the need for upfront investments in hardware and software, allowing organizations to reallocate funds towards innovation and growth. As workloads fluctuate, serverless platforms dynamically adjust resource allocation. The total cost overtime for serverless computing can be expressed as:

$C_{Total} = \sum_{=1} \quad C_{execution}$

where N is the number of function executions during the period of use. This dynamic allocation ensures cost efficiency without compromising performance [5], [6].

2.2 Scalability

Serverless computing offers unparalleled scalability, automatically scaling resources to meet application demands in real time. This elasticity is a key differentiator, enabling applications to manage unexpected surges in traffic without manual intervention [7]. For instance, e-commerce platforms during holiday sales or streaming services during live events can seamlessly scale their operations without downtime or performance degradation. The scalability of serverless platforms can be represented mathematically by the following formula:

$S = \frac{Rd}{c}$

Where:

S represents the scalability of the system. R_d is the rate of dynamic resource provisioning (resources allocated to meet demand).

T_c is the time taken to scale the resources (response time for scaling). This formula illustrates how serverless platforms efficiently scale resources based on the time required for adjustment , ensuring optimal performance during fluctuating workloads. This dynamic scaling is made possible by the underlying infrastructure's ability to provision and decommission resources as needed. Serverless platforms enable developers to concentrate on feature development by separating application logic from resource management, eliminating the need to handle capacity planning [8].

2.3 Simplified Operations

Serverless architectures significantly reduce the operational burden on developers and IT teams. By abstracting infrastructure management, serverless platforms allow organizations to redirect their focus towards core competencies, such as product development and customer engagement [9].With serverless computing, routine tasks like patch management, server provisioning, and monitoring are handled by the cloud provider. This approach speeds up development processes while minimizing the chances of human errors during configuration and deployment [10].

In addition, serverless platforms enable faster iteration and deployment of applications. Developers can deploy individual functions independently, enabling workflows for seamless continuous integration and continuous deployment (CI/CD) [11]. This agility is crucial for organizations operating in competitive and fast-paced industries.

2.4 Enhanced Security

Serverless platforms improve security by isolating each function and minimizing the attack surface. Cloud providers ensure strong protection by implementing measures such as automated updates and patching to defend applications against vulnerabilities [12]. In addition, serverless architectures remove direct server access, reducing the risks of unauthorized entry and configuration errors. Developers can also leverage built-insecurity features, such as encryption and role-based access control (RBAC), To guarantee adherence to industry standards and regulations [13].

2.5 Environmental Sustainability

By optimizing resource utilization, serverless computing contributes to environmental sustainability. Resources are provisioned only when needed, reducing energy consumption and carbon emissions. Studies have highlighted the potential of serverless architectures to improve the energy efficiency of cloud data centers [14]. This aligns with the growing emphasis on green computing, where organizations aim to reduce their ecological footprint while maintaining operational efficiency. Serverless computing serves as a model for sustainable cloud practices, combining economic and environmental benefits [15].

3. CHALLENEGES IN SERVERLESS COMPUTING

3.1 Resource Management

Resource management in serverless computing is a critical challenge due to the dynamic and ephemeral nature of serverless functions. Unlike traditional environments, serverless platforms dynamically allocate resources based on workload demands, leading to potential inefficiencies in resource utilization [4]. Research emphasizes the need for advanced scheduling algorithms and adaptive scaling mechanisms to address these issues. For example, predictive models leveraging machine learning enhances efficiency by optimizing resource allocation by anticipating workload patterns and pre- emptively scaling resources [6]. The lack of control over underlying infrastructure can also lead to inconsistencies in performance. Developer often have limited visibility into resource allocation decisions made by cloud providers, which complicates debugging and optimization efforts [8]. Tackling these challenges necessitates cooperation among cloud providers and developers to create transparent and efficient resource management frameworks.

3.2 Vendor Lock-In

Vendor lock-in Represents among the most significant concerns in serverless computing. Organizations that adopt proprietary serverless platforms often find it challenging to migrate their applications to alternative providers due to platform-specific features and APIs [9]. This dependence on a single vendor not only limits flexibility but also increases costs in the long term.

To mitigate vendor lock-in, researchers advocate for standardized abstraction layers and open-source serverless frameworks. These solutions aim to provide portability and interoperability across multiple platforms, enabling organizations to switch providers without significant reengineering efforts [10]. For instance, open-source tools like OpenFaaS and Knative offer developers the flexibility to deploy serverless functions across diverse environments, reducing reliance on specific vendors [11].

3.3 Performance Variability

Performance variability, including issues like cold starts and inconsistent response times, poses a significant challenge in serverless computing. Cold starts happen when a serverless function is invoked after a span of inactivity, leading to delays as the platform initializes the execution environment [7]. This latency can be detrimental for real-time and latency-sensitive applications, such as IoT and financial services.

Solutions to mitigate cold starts include pre-warming strategies, where cloud providers keep a subset of execution environments ready to handle incoming requests. Additionally, optimizing function code and reducing dependencies can minimize initialization times [12]. Nevertheless, these solutions frequently come at the cost of increased operational expenses, requiring a careful balance between performance and cost efficiency. Performance variability is also influenced by the multitenancy model of serverless platforms, where multiple functions share the same infrastructure. This can lead to resource contention and unpredictable performance, particularly during peak usage periods [14]. Addressing these issues requires innovations in workload isolation and resource scheduling to ensure consistent and reliable performance.

3.4 Debugging and Monitoring

The distributed and event-driven nature of serverless applications complicates debugging and monitoring. Traditional debugging tools are often inadequate for tracing issues across multiple functions and services. Developers need specialized tools that provide end-to-end visibility into application workflows and dependencies [15]. Cloud providers have introduced monitoring solutions, such as AWS CloudWatch and Azure Monitor, to address these challenges. These tools offer real-time insights into function performance, error rates, and invocation patterns. However, achieving comprehensive observability requires integrating these tools with third-party solutions to gain deeper insights into application behavior [16].

3.5 Cost Management

While serverless computing is often touted for its cost efficiency, managing costs in large-scale deployments can be challenging. The pay-as-you-go model requires careful monitoring of function invocations and execution times to avoid unexpected expenses [17]. Organizations must implement cost optimization strategies, such as consolidating functions and optimizing code execution times, to maximize the financial benefits of serverless computing.

4. USE CASES OF SERVERLESS COMPUTING

4.1 Dynamic Web Hosting

Dynamic web hosting is one of the most prevalent use cases of serverless computing. Serverless platforms provide the scalability and flexibility required to handle fluctuating traffic demands without manual intervention [7]. For example, e-commerce websites experiencing high traffic during holiday sales can benefit from automatic scaling capabilities. Serverless architectures eliminate the need for over-provisioning, ensuring cost efficiency while maintaining performance.

In addition, serverless web hosting enables developers to focus on building features and improving user experiences. By abstracting server management, organizations can achieve faster time-to-market and lower operationaloverhead [8]. Real-time content delivery and personalized user experiences are also facilitated by the event-driven nature of serverless platforms.

4.2 Multimedia Services

Multimedia applications, such as video transcoding and content streaming, leverage serverless computing for its cost efficiency and scalability. Traditional transcoding workflows often require significant computational resources, leading to high costs and delays. Serverless platforms address these challenges by dynamically allocating resources based on workload demands, reducing costs and processing times [9].

For instance, video streaming platforms can use serverless functions to process user-generated content in real time. The elasticity of serverless architectures ensures seamless delivery of multimedia content, even during peak usage periods [10]. Furthermore, serverless computing supports batch processing and event-driven workflows, making it ideal for applications with dynamic workloads.

4.3 IoT and Edge Computing

The integration of serverless computing with IoT and edge devices is gaining traction due to its low-latency processing capabilities. Serverless platforms enable real-time data analysis and decision-making at the edge, reducing the need for centralized processing [11]. Applications in this domain include smart home automation, predictive maintenance, and industrial IoT systems. Workflow modelling and scheduling for serverless edge computing have been explored in recent research, highlighting the potential for optimizing resource allocation and reducing latency [12]. By combining serverless architectures with edge computing, organizations can achieve greater efficiency and scalability in IoT deployments.

4.4 Data Processing and Analytics

Serverless computing is ideal for data-intensive applications, such as real-time analytics and batch processing. Data pipelines built on serverless platforms can ingest, process, and analyze large volumes of data with minimal latency. This is particularly valuable for industries such as finance, healthcare, and retail, where timely insights are critical [13]. Cloud providers offer serverless solutions for data processing, such as AWS Lambda integrated with Amazon Kinesis and Google Cloud Functions with Big Query. These services enable organizations to process streaming data in real time, extract actionable insights, and automate decision-making workflows [14].

4.5 Machine Learning and AI

Serverless computing is increasingly being adopted for machine learning and artificial intelligence (AI) workloads. Training and deploying machine learning models often require substantial computational resources, which can be cost-prohibitive for small organizations. Serverless platforms provide a cost-effective alternative by enabling developers to pay only for the compute resources consumed during model training and inference [15].

Serverless frameworks also facilitate the deployment of machine learning models as microservices, allowing seamless integration with existing applications. Event- driven architectures support real-time predictions and anomaly detection, making serverless computing ideal for AI-powered applications [16].

5. OPTIMIZING ENERGY AND PORTABILITY IN SERVERLESS ARCHITECTURES

5.1 Energy Efficiency

Energy efficiency in serverless computing is a growing research area. With the increasing adoption of serverless platforms, optimizing energy consumption has become essential for reducing operational costs and achieving sustainability goals. Techniques such as dynamic resource scaling, workload distribution, and energy harvesting are being actively explored to enhance energy efficiency [15]. For instance, energy-aware scheduling algorithms can reduce power consumption by optimizing the allocation of computational tasks to energy-efficient hardware. Moreover, incorporating renewable energy sources into serverless infrastructure can further reduce carbon footprints, aligning with global efforts to combat climate change [16].

5.2 Enhanced Abstraction Layers

To address the challenge of vendor lock-in, researchers are focusing on developing standardized abstraction layers and open- source serverless frameworks. These solutions aim to provide portability across multiple platforms, enabling organizations to switch providers without significant reengineering efforts [17]. Open-source tools such as OpenFaaS, Knative, and Kube less are gaining traction in the industry, offering developers the flexibility to deploy serverless functions across diverse environments. These frameworks also promote interoperability, fostering collaboration and innovation within the serverless ecosystem [18].

6. FUTURE TRENDS AND RESEARCH OPPORTUNITIES IN SERVERLESS COMPUTING

6.1 Advancing Resource Allocation and Green Energy Integration

Energy efficiency in serverless computing an increasingly vital research area as the demand for serverless platforms grows. Current practices in dynamic resource allocation have shown promise, but there is considerable scope for innovation. For instance, machine learning algorithms have the capability to predict workload patterns more effectively, allowing for better scaling decisions that minimize energy wastage [4]. Research has also suggested that edge computing integration could significantly reduce energy use By analyzing data nearer to its source, thus reducing the need for extensive data transmission [9]. These strategies align with global sustainability goals, as reducing data center energy consumption directly lowers carbon footprints.

Additionally, serverless computing providers are exploring the potential of renewable energy sources to power their operations. Leveraging renewable energy to power datacenters can drastically reduce environmental impact. Further research into optimizing renewable energy storage and utilization in serverless architectures is expected to advance the field [12].

6.2 Advanced Workload Optimization

As workloads become increasingly complex, optimizing their execution in serverless environments presents an exciting avenue for research. Advanced scheduling techniques, such as behavior-tree-based models, can improve resource utilization by dynamically adapting to workload changes in real-time [7]. Such approaches are particularly relevant for applications in edge computing and IoT, where resource constraints and latency requirements are critical [10]. Future studies are also expected to focus on hybrid serverless models that combine on-premises and cloud- based resources to achieve optimal workload distribution. These models could leverage decentralized processing capabilities, enabling organizations to balance cost, performance, and scalability more effectively [15].

6.3 Security Enhancements

Security Continues to be an ongoing challenge in serverless computing. Future research will likely address vulnerabilities specific to serverless platforms, such as insecure APIs and insufficient access controls. Techniques like automated vulnerability detection and real-time threat mitigation using AI-powered solutions are gaining traction [14]. Researchers are also investigating blockchain-based methods to enhance transparency and trust in multi-tenant serverless environments [16].

Additionally, the evolution of zero-trust security models tailored for serverless applications could provide enhanced protection against data breaches and insider threats. These models emphasize continuous authentication and monitoring, reducing the risks associated with distributed and event- driven architectures [17].

6.4 Multi-Cloud and Interoperability

The reliance on single-vendor platforms has long been a limitation in serverless computing. Multi-cloud strategies, which enable applications to run across different providers, offer a promising solution to vendor lock-in [8]. Researchers are exploring abstraction layers and open-source frameworks like Knative to facilitate seamless interoperability between platforms [11]. The adoption of multi-cloud strategies could also enhance disaster recovery capabilities and improve application resilience. By enabling dynamic workload migration between providers, organizations can ensure continuity and reliability even during platform-specific outages [13].

6.5 Emerging Use Cases

Emerging use cases for serverless computing, such as quantum computing and real-time augmented reality (AR) applications, are expected to drive future research. These applications demand high computational power and low latency processing, presenting unique challenges for serverless platforms [10].

For example, serverless frameworks tailored for quantum computing could democratize access to quantum resources by enabling pay-as-you-go models. Similarly, AR applications could benefit from serverless edge computing solutions that reduce latency and enhance user experiences [15].

In conclusion, the future of serverless computing lies in addressing its existing challenges while expanding its capabilities to fulfill the requirements of next-generation applications. By focusing on energy efficiency, security, workload optimization, and multi-cloud interoperability, Researchers and practitioners can fully harness the capabilities of serverless architectures.

CONCLUSION

Serverless computing has emerged as a transformative paradigm in cloud computing, offering unparalleled benefits such as cost efficiency, scalability, and simplified operations. By abstracting infrastructure management, serverless architectures empower developers to focus on innovation and accelerate time-to-market. The pay-as-you-go model ensures optimal resource utilization, making it particularly appealing for organizations with dynamic workloads and budget constraints [2], [4].

However, as this study has highlighted, serverless computing is not without its challenges. Issues such as vendor lock-in, performance variability, and resource management complexities remain significant barriers to widespread adoption [9], [12]. Addressing these challenges will require continued collaboration between cloud providers, developers, and researchers. Advances in standardization, multi-cloud strategies, and enhanced monitoring tools are expected to mitigate these concerns and promote broader adoption of serverless platforms [11], [15].The diverse use cases of serverless computing, ranging from dynamic web hosting to machine learning, illustrate its versatility and potential to revolutionize various industries. Applications in Io T, edge computing, and data analytics showcase how serverless architectures can meet the demands of modern, data-driven ecosystems. Furthermore, emerging fields such as quantum computing and augmented reality are poised to benefit significantly from serverless innovations [10], [16].

Looking ahead, the research prospects and future directions outlined in this paper emphasize the importance of sustainability, security, and interoperability. Energy- efficient practices and renewable energy integration are critical for aligning serverless computing with global sustainability goals. Enhanced security frameworks, such as zero-trust models, will ensure the protection of sensitive data in increasingly distributed environments. Multi-cloud strategies and open-source frameworks will drive interoperability, reducing vendor lock-in and fostering a competitive ecosystem [8], [14].

In conclusion, serverless computing represents a promising and evolving frontier in cloud computing. By addressing its current limitations and exploring its full potential, organizations and researchers can unlock new opportunities for innovation and efficiency. The future of serverless computing resides in its capability to adapt and scale to meet the demands of next-generation applications, establishing it as a cornerstone of modern technology landscapes.

REFERENCES

[1] A. Eivy, "Concerns Over the Economics of 'Serverless' Cloud Platforms," IEEE Cloud Computing, vol. 4, no. 2, pp. 6–12, Mar. 2017.

[2] S. Hendrickson, O. Stancevic, and S. Fink, "Serverless Computing: Design, Implementation, and Performance," *Proc. 2016 IEEE Int. Conf. Cloud Eng. (IC2E)*, pp. 116-121, 2016.

[3] E. Jonas et al., "A Simplified Approach to Cloud Programming: A Berkeley Perspective on Serverless Computing," arXiv preprint arXiv:1902.03383, 2019.

[4] S. Eismann, J. Scheuner, and P. Leitner, "A Comprehensive Study of Serverless Function Cold Start Duration," *Proc. 2021 IEEE/ACM Int. Conf. Utility and Cloud Compute. (UCC)*, pp. 1-10, 2021.

[5] G. Adzic and R. Chatley, "The Impact of Serverless Computing on Economics and System Architecture," in P on proceedings of the 2017 IEEE International Conference Software Architecture Workshops (ICSAW), Gothenburg, Sweden, 2017, pp. 239–242.

[6] T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha, "A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms," *Proc. 2017 IEEE Int. Conf. Cloud Compute. Technol. and Sci. (CloudCom)*, pp. 162-169, 2017.

[7] G. McGrath and P. Brenner, "Serverless Computing: Design, Implementation, and Performance," *Proc. 2017 IEEE 37th Int. Conf. Distributed Comput. Systems Workshops (ICDCSW)*, pp. 405-410, 2017.

[8] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... & Suter, P. (2017). Serverless Computing: Current Trends and Open Problems. In Research Advances in Cloud Computing (pp. 1-20). Springer.

[9] A. Roberts and A. Brutus, "Serverless Architectures: Considerations for Startups and SMEs," *Proc. 2018 IEEE Int. Conf. Comput. and Inf. Technol. (CIT)*, pp. 1-8, 2018.

[10] G. Adzic and R. Chatley, "Serverless Computing: Economic and Architectural Impact," *Proc. 2017 IEEE Int. Conf. Cloud Eng. (IC2E)*, pp. 1-8, 2017.

[11] J. Spillner, "Transitory Programming in the Serverless Age: A Paradigm Shift in Distributed Computing," *Proc. 2017 IEEE 37th Int. Conf. Distributed Comput. Systems Workshops (ICDCSW)*, pp. 169-174, 2017.

[12] Wang, L., Li, M., Zhang, Y., Ristenpart, T., & Swift, M. (2018). Peeking Behind the Curtains of Serverless Platforms. In Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18) (pp. 133-146). USENIX Association.

[13] Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C., Khandelwal, A., Pu, Q., ... & Stoica, I. (2019). Cloud Programming Simplified: A Berkeley Vssiew on Serverless Computing. arXiv preprint arXiv:1902.03383.

[14] Hellerstein, J. M., Faleiro, J., Gonzalez, J., Schleier-Smith, J., Sreekanti, V., & Wu, C. (2018). Serverless Computing: One Step Forward, Two Steps Back. In Proceedings of the 9th Biennial Conference on Innovative Data Systems Research (CIDR 2019).

[15] Eivy, A. (2017). Be Wary of the Economics of "Serverless" Cloud Computing. IEEE Cloud Computing, 4(2), 6-12.

[16] Roberts, M. (2016). Serverless Architectures. martinfowler.com.

[17] S. Hendrickson, O. Stancevic, and S. Fink, "Serverless Computing: Design, Implementation, and Performance," *Proc. 2016 IEEE Int. Conf. Cloud Eng. (IC2E)*, pp. 116-121, 2016.

[18] G. McGrath and P. Brenner, "Serverless Computing: Design, Implementation, and Performance," *Proc. 2017 IEEE 37th Int. Conf. Distributed Comput. Syst. Workshops (ICDCSW)*, pp. 405-410, 2017.