# Challenges of Serverless Computing Paradaigm

*Note: Sub-titles are not captured for https://ieeexplore.ieee.org and should not be used

1st Mingyu Jo
*School of Computer Science and Engineering*
*Chung-Ang University*
Seoul, Republic of Korea
mgjo@cslab.cau.ac.kr

2nd Donghyeon Kim
*School of Computer Science and Engineering*
*Chung-Ang University*
Seoul, Republic of Korea
dhkim@clsab.cau.ac.kr

3rd Ingyu Baek
*School of Computer Science and Engineering*
*Chung-Ang University*
Seoul, Republic of Korea
igbaek@clsab.cau.ac.kr

4th Sangoh Park*
*School of Computer Science and Engineering*
*Chung-Ang University*
Seoul, Republic of Korea
sopark@cau.ac.kr

***Abstract*—Serverless computing is an innovative cloud computing paradigm that alleviates infrastructure management burdens for developers while offering cost-efficiency and scalability. However, cold start latency, which occurs during the initial execution of functions, and the difficulty in implementing complex business logic through workflows remain significant challenges. This paper reviews various approaches and recent research trends aimed at addressing these issues. Regarding the cold start problem, we analyze solutions such as function pre-warming, container reuse, and optimized runtime environments. We also examine orchestration tools and frameworks for workflow management. Furthermore, we propose future research directions, including intelligent optimization using machine learning, novel programming models, and integration with edge computing.**

***Index Terms*—Serverless computing, Cloud computing, FaaS, Cold start problem, Workflow management**

## I. Introduction

The majority of services provided by traditional cloud computing operate on a serverful basis. This serverful approach necessitates continuous resource availability, even during periods of inactivity, leading to issues in terms of cost-effectiveness and, moreover, increasing energy consumption, which raises environmental concerns. While various studies, such as dynamic load balancing, are being conducted to address these issues, they cannot be considered fundamental solutions as they do not eliminate or significantly reduce the costs and energy required for standby resources.

Serverless computing emerged as a new paradigm to address the aforementioned issues in traditional cloud computing. Introduced by AWS in 2007 under 'Lambda,' the serverless paradigm resolves cloud computing challenges by utilizing virtualized computing units such as containers or Virtual Machines (VMs) in response to client requests rather than constantly maintaining available resources. The serverless paradigm virtualizes specific tasks for processing client requests as lambda instances and executes these instances to handle incoming requests.

The characteristics of serverless computing offer benefits to both cloud users and providers. Cloud users can focus solely on the logic for processing requests, namely 'Functions', without the need for complex resource optimization or cost calculations. Concurrently, cloud providers can concentrate on rapid request processing and resource distribution technologies. Due to these features, serverless computing is also referred to as Function-as-a-Service (FaaS), offering advantages such as reduced operational costs, improved development productivity, and faster time-to-market. According to a Gartner report [1], it is anticipated that over 50% of all enterprises will adopt serverless computing by 2025 [REP-02]. Furthermore, a report [2] by MarketsandMarkets projects that the serverless computing market will grow from $21.9 billion to $44.7 billion by 2029, with an expected compound annual growth rate of 15% [REP-01].

Despite these advantages, serverless computing still faces several challenges. In particular, the cold start problem and difficulties in managing complex workflows are significant factors hindering serverless architecture's widespread adoption.

This paper aims to provide an in-depth analysis of the critical challenges in the serverless computing paradigm, explicitly focusing on the cold start problem and workflow management issues. This paper presents current solutions and future research directions for these challenges. Through this analysis, we seek to comprehensively understand the current state of serverless computing and explore future developmental pathways.

The structure of this paper is as follows: Chapter 2 provides an overview and characteristics of serverless computing. Chapter 3 presents a detailed analysis of the main challenges in serverless computing, focusing on the cold start problem and workflow management issues. Finally, Chapter 4 concludes the

979-8-3315-0694-0/25/$31.00 ©2025 IEEE

study and suggests directions for future research.

## II. Overview of serverless

Serverless Computing is an advanced form of cloud computing that enables developers to build and run applications without directly managing server infrastructure. The term 'serverless' does not imply the absence of servers but rather the transfer of server management and operational responsibilities to cloud providers. Key features of this model include event-driven execution, automatic scaling, precise usage-based billing, and stateless characteristics.

In several important aspects, serverless computing differs from traditional Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) models. Primarily, serverless eliminates the need for developers to perform infrastructure management tasks such as server provisioning, patching, and scaling. Additionally, serverless allocates and utilizes resources only when necessary, charging only for actual execution time, thus avoiding costs associated with idle time.

Serverless computing finds applications in various domains. Primary use cases include web application and API development, real-time data processing, Internet of Things (IoT) backend implementation, chatbot, AI service operations, and scheduled tasks such as periodic backups or notifications. These diverse applications demonstrate the flexibility and scalability of serverless computing.

Currently, the market offers several serverless computing platforms. Notable examples include AWS Lambda, Microsoft Azure Functions, Google Cloud Functions, IBM Cloud Functions, and Cloudflare Workers. Each platform possesses unique features and trade-offs, allowing developers to select the most suitable platform based on project requirements.

While serverless computing offers numerous benefits, it also faces several challenges. In particular, the cold start problem and difficulties in managing complex workflows are major factors hindering serverless architecture's widespread adoption. These challenges will be crucial in determining the future development direction of serverless computing and will be examined in more detail in the following chapter.

## III. Challenges of Serverless Computing

Despite the numerous benefits of serverless computing, Serverless computing still faces several significant challenges. This chapter discusses two key issues: the cold start problem and workflow management challenges.

### *A. Cold Start Problems*

The cold start problem refers to the latency when a serverless function is executed for the first time or after prolonged inactivity. This delay is due to the time required to initialize the function's execution environment and load necessary resources. The delay caused by cold starts can vary from milliseconds to several seconds, which can be a critical issue for applications requiring real-time responses. Several solutions have been proposed to address this problem. For example, methods include periodically invoking functions to maintain a 'warm' state, reusing containers, and utilizing optimized runtime environments. Additionally, some cloud providers offer 'pre-warming' features to mitigate the cold start issue.

FaaSBatch [3] employs two primary techniques. It optimizes container reuse by batch processing similar function calls and reduces cold start overhead by scaling smaller functions into larger units. This approach performs batching and scaling operations while considering dependencies between function calls. Experimental results show that FaaSBatch can reduce execution time by up to 90% and resource usage by up to 70% compared to conventional serverless platforms.

FaaSlight [4] utilizes an optimal function grouping strategy by analyzing dependencies between Functions to optimize cold start latency in Function-as-a-Service (FaaS) environments. FaaSlight is implemented at the application level without platform modifications, making it universally applicable to various FaaS platforms. Experimental results demonstrate that this approach can significantly reduce cold start latency compared to existing methods.

In [5], a replayable execution technique is proposed to optimize page sharing in isolated environments. It aims to enhance memory efficiency in isolated environments such as virtual machines or containers. By making program execution deterministic and replayable, it enables sharing of identical page content across multiple instances. This is achieved by modifying the runtime environment to control memory allocation, garbage collection, and thread scheduling behaviors. Consequently, this approach significantly reduces memory duplication and improves overall system memory efficiency. It can also be used for debugging, error reproduction, and security analysis, enhancing the overall performance and utility of managed runtime environments.

[6] proposes the 'Agile Cold Starts' approach. The core of this method is to partially prepare the function's execution environment in advance and dynamically load necessary dependencies. Specifically, it analyzes the function's code and dependencies to generate a lightweight base image containing only the minimal essential components. During execution, this base image is rapidly loaded, and additional dependencies are dynamically loaded as needed. Furthermore, it reduces cold start frequency by predictively preparing environments for frequently used functions based on function call pattern analysis. This method significantly reduces cold start time while optimizing resource usage, thereby improving serverless applications' performance and scalability.

[7] proposes a 'function fusion' technique to address the cold start problem in serverless computing. Function fusion combines multiple small functions into a single larger function, reducing the number of function calls and decreasing the frequency of cold starts. The researchers developed an algorithm to determine optimal fusion combinations by analyzing inter-function dependencies and execution patterns. They also present a method to balance performance and cost by considering the execution time and memory usage of fused functions. Experimental results show that this approach reduces overall

execution time and improves resource utilization, proving particularly effective in microservice architectures.

[8] proposes a new approach applying reinforcement learning techniques to reduce the frequency of cold starts in serverless functions. They propose a reinforcement learning model to optimize decisions on maintaining or terminating function instances in serverless environments. The model dynamically determines the retention time of each function instance by considering various factors such as function call patterns, resource usage, and costs. The model derives an optimal policy during the learning process by balancing cold start occurrences and resource waste. Experimental results show that this approach significantly reduces cold start frequency while improving overall cost-effectiveness compared to static timeout policies. It also demonstrated high adaptability to various workload patterns.

However, these solutions have limitations. A perfect solution for the cold start problem is yet to be found, and balancing performance improvement with cost remains a significant challenge. Research and development are actively ongoing to address these challenges. For example, intelligent function prewarming techniques using machine learning and improvements in distributed tracing and monitoring tools are emerging as crucial research directions. Additionally, there are attempts to mitigate the cold start problem through integration with edge computing [9].

### *B. Managing Workflows of Serverless*

Next, the workflow management problem refers to the difficulties in implementing and managing complex business logic in serverless environments. Serverless functions are inherently designed as small, independent units, often necessitating the combination of multiple functions to construct complex workflows. In this process, managing function dependencies, controlling data flow, handling errors, and managing state emerge as key challenges. To address these issues, workflow orchestration tools such as AWS Step Functions and Azure Durable Functions have been developed and are in use. These tools enable visual design and management of complex workflows and facilitate inter-function communication and state management.

[10] proposes an 'intent-driven' approach for efficiently orchestrating serverless applications in a computing continuum environment. This technique suggests methods for optimal placement and execution of serverless functions utilizing various computing resources, including cloud, edge, and IoT devices. The system interprets high-level intents defined by developers (e.g., performance requirements, cost constraints, data locality) and dynamically selects execution environments based on these. Additionally, it continuously optimizes application performance using real-time monitoring and machine-learning techniques. This approach improves resource utilization, reduces application response time, and lowers operational costs.

[11] proposes 'Gaffer', a cloud computing-based serverless orchestration framework. Gaffer is designed to manage and execute complex and unprecedented workflows efficiently. This framework dynamically combines serverless functions and optimizes execution order to implement complex business logic. Key features of Gaffer include automatic workflow partitioning and parallel processing, real-time performance monitoring, and error recovery mechanisms. It also utilizes machine learning techniques to learn workflow execution patterns and optimize resource allocation through future-state prediction. Experimental results show that Gaffer reduces execution time and improves cost-efficiency compared to serverless platforms.

A new serverless framework for workflow orchestration, 'Jolteon' [12], has been proposed. The core functionality of this framework is its ability to partition and optimize workflows dynamically, converting them into serverless functions. Jolteon analyzes data dependencies between functions to maximize parallel execution opportunities and generates execution plans that minimize execution time and cost. Furthermore, it continuously optimizes workflow performance through real-time monitoring and automatic scaling features. Experimental results indicate that Jolteon reduces execution time and improves resource utilization compared to existing workflow management systems.

## IV. Conclusion

This study has conducted an in-depth analysis of the significant challenges in the serverless computing paradigm, mainly focusing on the cold start problem and workflow management issues. Serverless computing, an evolved form of cloud computing, is an innovative technology that relieves developers of the burden of infrastructure management while providing cost-efficiency and scalability. However, despite these benefits, significant challenges remain to be addressed.

The cold start problem, which causes delays in the initial execution of serverless functions, is particularly problematic for applications requiring real-time responses. While solutions such as function pre-warming, container reuse, and optimized runtime environments have been proposed, a perfect solution is yet to be found. The workflow management problem refers to the difficulties in implementing and managing complex business logic in serverless environments. Although various workflow orchestration tools have been developed and are in use, maintaining consistency and debugging in large-scale distributed systems remain challenging.

Various research and technological developments are underway to address these challenges. Key research directions include intelligent function pre-warming using machine learning, developing serverless-specific programming models, and improvements in distributed tracing and monitoring tools. Additionally, attempts are being made to mitigate the cold start problem through integration with edge computing.

Serverless computing is expected to remain significant in the cloud computing market. If current challenges are effectively resolved, serverless computing could be utilized in an even wider range of fields and has the potential to change how enterprises operate their IT infrastructure fundamentally.

Future research will require more innovative approaches to solve these challenges. In particular, automated optimization techniques using artificial intelligence and machine learning, the development of new programming paradigms, and methods for applying serverless computing in hybrid cloud environments could be significant research topics. Research on enhancing security and ensuring regulatory compliance in serverless computing will also be necessary.

In conclusion, despite its current challenges, serverless computing is poised to become a key technology driving the future of cloud computing. As these challenges are addressed, serverless computing will mature and ultimately become a core driver accelerating the digital transformation of businesses.

## Acknowledgment

This paper was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea Government(MSIT)(RS-2024-00345869) and Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0020632, HRD Program for Industrial Innovation)

## References

[1] M. Cooney. (2022, Oct) Gartner: By 2025, half of enterprise it spending will be for cloud. Network World. [Accessed: 15-May-2023]. [Online]. Available: https://www.networkworld.com/article/970698/gartner-by-2025-half-of-enterprise-it-spending-will-be-for-cloud.html

[2] "Serverless architecture market size, share, growth analysis, by deployment model (public cloud, hybrid cloud), by end-user (bfsi, it and telecommunications), by component (service types, monitoring, others), by organization size (large enterprises, smes), by region - industry forecast 2024-2031," Report, 2024. [Online]. Available: https://www.skyquestt.com/report/serverless-architecture-market

[3] Z. Wu, Y. Deng, Y. Zhou, J. Li, and S. Pang, "Faasbatch: Enhancing the efficiency of serverless computing by batching and expanding functions," in *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2023, pp. 372–382.

[4] X. Liu, J. Wen, Z. Chen, D. Li, J. Chen, Y. Liu, H. Wang, and X. Jin, "Faaslight: General application-level cold-start latency optimization for function-as-a-service in serverless computing," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 5, pp. 1–29, 2023.

[5] K.-T. A. Wang, R. Ho, and P. Wu, "Replayable execution optimized for page sharing for a managed runtime environment," in *Proceedings of the Fourteenth EuroSys Conference 2019*, 2019, pp. 1–16.

[6] A. Mohan, H. Sane, K. Doshi, S. Edupuganti, N. Nayak, and V. Sukhomlinov, "Agile cold starts for scalable serverless," in *11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19)*, 2019.

[7] S. Lee, D. Yoon, S. Yeo, and S. Oh, "Mitigating cold start problem in serverless computing with function fusion," *Sensors*, vol. 21, no. 24, p. 8416, 2021.

[8] S. Agarwal, M. A. Rodriguez, and R. Buyya, "A reinforcement learning approach to reduce serverless function cold start frequency," in *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2021, pp. 797–803.

[9] M. S. Aslanpour, A. N. Toosi, C. Cicconetti, B. Javadi, P. Sbarski, D. Taibi, M. Assuncao, S. S. Gill, R. Gaire, and S. Dustdar, "Serverless edge computing: vision and challenges," in *Proceedings of the 2021 Australasian computer science week multiconference*, 2021, pp. 1–10.

[10] N. Filinis, I. Tzanettis, D. Spatharakis, E. Fotopoulou, I. Dimolitsas, A. Zafeiropoulos, C. Vassilakis, and S. Papavassiliou, "Intent-driven orchestration of serverless applications in the computing continuum," *Future Generation Computer Systems*, vol. 154, pp. 72–86, 2024.

[11] S. Roy, S. Kolanu, and S. Krishnaveni, "Gaffer: Cloud computing based serverless orchestration framework for unprecedented workflow," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2021, pp. 1054–1060.

[12] Z. Zhang, C. Jin, and X. Jin, "Jolteon: Unleashing the promise of serverless for serverless workflows," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024, pp. 167–183.