Proceedings of the 8th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2026.33975

Serverless Computing Core Technology and Research Status

Shiyu Liu

School of AI and Advanced Computing College, Xi'an Jiaotong Liverpool University, Suzhou, China
Shiyu.Liu2402@student.xjtlu.edu.cn

Abstract. As an emerging cloud computing paradigm, serverless computing, based on "function as a service", splits applications into fine-grained functions, and automatically manages underlying resources by the platform, significantly improving resource utilization and reducing operation and maintenance costs. It has gradually become the preferred architecture for cloud native application development and deployment with its unique advantages of lightweight and event driven. However, there are still many challenges in performance, security and resource scheduling. This paper systematically summarizes the core technologies of serverless computing, including performance optimization, resource scheduling, security protection, and integration with emerging technologies such as webassembly, micro front end, and artificial intelligence; On this basis, the main challenges faced at present are analyzed, such as insufficient support for large-scale tasks, immature cross domain scheduling, and security model to be improved. The development trends of intelligent scheduling, cross domain computing integration, and trusted execution environment in the future are prospected. This paper aims to provide a systematic reference for the research and application of serverless computing.

Keywords: Serverless computing, Performance optimization, Resource scheduling, Safety protection, Technology convergence

1. Introduction

Serverless computing is not really "no server", but all tasks managed by the server are handed over to the cloud service provider. Developers only need to write and upload function code, and the platform automatically completes execution, scaling and billing. Compared with traditional cloud computing, serverless architecture has higher elasticity, finer billing granularity and lower operation and maintenance burden [1]. With the evolution of cloud computing from data center cloud to ubiquitous cloud, the computing infrastructure is becoming increasingly large-scale, diversified and cross regional. Serverless computing naturally fits the ubiquitous cloud business scenario because of its lightweight, event driven, cross platform and other characteristics [2]. However, the current mainstream serverless programming framework still has obvious deficiencies in cross regional resource scheduling, large-scale service construction, inter function communication, etc., the computing power of single function is limited, the integration of heterogeneous resources is lack of practice, and the communication overhead and cold start delay affect the user experience [2].

DOI: 10.54254/2755-2721/2026.33975

In recent years, scholars at home and abroad have carried out extensive research on serverless computing. Kumari et al. quantitatively analyzed the advantages and disadvantages of serverless architecture in terms of cost-effectiveness and response latency by comparing it with IAAs and PAAS, and pointed out that function cold start, resource isolation and cross tenant performance interference are still unsolved problems [3]. Yang Guang et al. systematically reviewed the performance optimization technology of server insensitive computing system, covering task adaptation, sandbox environment, i/o communication, resource scheduling and other aspects [4]. Zhandongyang et al. proposed cross functional behavior analysis and constraint technology for serverless applications, which effectively improved security [5]. In addition, wangxunan et al. optimized the resource utilization of serverless architecture through passive clustering strategy [6]. On the whole, the existing research has made positive progress in performance, scheduling, security and other aspects, but it still needs to be in-depth in large-scale complex task support, cross domain resource scheduling, security model improvement and other aspects.

This paper aims to systematically review the core technology and research status of serverless computing. The specific research objectives include: combing the latest progress of key technologies such as performance optimization, resource scheduling, and security protection; Analyze the typical solutions of the integration of serverless, webassembly, micro front end, AI platform and other emerging technologies; Summarize the main challenges currently facing; Looking forward to the future development direction of intelligence, efficiency and security. Through the work of this paper, it is expected to provide systematic reference for researchers and practitioners of serverless computing.

2. Serverless core technology and optimization direction

The performance optimization of serverless computing is a hot research direction in the current academic circles. Cold start is the most prominent performance bottleneck at present. After the function is called for the first time or is idle, when it is reactivated, it needs to go through the process of container creation, runtime initialization, code loading, etc., and the response delay is greatly increased [4]. The academia has optimized the problem from the aspects of sandbox environment, preheating mechanism and operation optimization. When the program is running, it is implemented in rust language, which has made extremely ingenious optimization for memory management, parallel computing and hardware acceleration. The operation efficiency of the implemented function in the resource constrained environment almost reaches the native performance [7]. Yang Guang, et al. Made a systematic and rigorous overview of the performance optimization technology of server insensitive computing system, and made an excellent comb in task adaptation, sandbox environment, i/o communication, resource scheduling and other aspects [4].

Resource scheduling and elastic scaling are also very important technologies for serverless platform, and platform level and user level scheduling strategies are directly related to resource utilization and task execution efficiency [4]. Therefore, the passive clustering strategy proposed by wangxunan et al. Optimized the serverless architecture and improved the framework with a complete performance evaluation system, reducing the average utilization of CPU and memory on the server from 75% and 68% to 65% and 58%, effectively improving the efficiency of resource allocation [6].

In a dynamic, multi tenant serverless environment, the balance between energy efficiency optimization and reliable performance scheduling is particularly difficult. Kumar et al. proposed an application scheduling method based on deep reinforcement learning to solve this problem, which

DOI: 10.54254/2755-2721/2026.33975

can significantly reduce the overall energy consumption of serverless clusters without sacrificing the reliability of function execution [8]. This method uses reinforcement learning agent to adaptively perceive workload fluctuation and the competition state of underlying resources, so as to make a better decision of function instance placement. This provides an important algorithm reference for building a green, high-performance serverless platform in the future.

Since i/o and communication optimization are of great significance to serverless architecture, and inter function communication and i/o operation are common performance bottlenecks, optimizing communication protocols, reducing data transmission overhead and improving i/o concurrency are research hotspots in this field [4]. Especially in large-scale distributed tasks, data transfer and collaborative scheduling between functions directly determine the overall performance of the system [2].

Security is another important issue faced by serverless architecture: because the application is divided into several functions running in different containers, and the internal interfaces of the program are exposed to the network, the attack surface increases, the risk of unauthorized access increases, and the integrity of control flow and data flow is more difficult to guarantee [5]. However, the existing security detection methods can not take into account the integrity of control flow and data flow between functions. Therefore, Zhan Dongyang and other scholars proposed a cross function behavior analysis and constraint technology for serverless applications. First, static analysis was used to extract the complete business access model between functions, and then real-time cross function access security detection was done based on it. The detection rate of abnormal control flow and data flow reached 97.54% and 92.87%, respectively. At the same time, the monitoring false alarm rate was reduced by more than 10%, which effectively improved the security of serverless computing [5].

In addition to the analysis of function behavior, the safe unloading in the mixed cloud scenario is also the current focus. For example, Yu et al. emphasized the integrity and cost constraints of cross trust domain data transmission in the workflow offload framework integrated by private cloud and serverless platform [9]. The deadline aware task offload (Dato) algorithm proposed by Yu et al. Not only ensured that the task was completed on time, but also minimized the economic cost and indirectly reduced the security risks exposed due to long-time operation.

3. Integration of serverless and emerging technologies

Serverless computing has excellent flexibility and scalability, and can be deeply integrated with various emerging technologies. Webassembly is an ideal choice for serverless function runtime due to its lightweight, security and cross platform features [7]. Huangzichen et al. Proposed a very complete and well-designed AI as a service (aiaas) serverless platform based on webassembly. It uses a hierarchical architecture to clearly separate computing resources, middleware and application services, so that it is highly modular and easy to expand. The platform uses the webassembly sandbox technology to provide a safe and isolated execution environment for each AI application, and combines the automatic capacity expansion and load balancing algorithm to schedule resources [7]. The advantages of this scheme are good isolation and fast startup, but the disadvantages are that the webassembly ecology is not yet mature, and the support for some languages and libraries is limited.

In the field of micro front-end, Li Yipeng proposed a micro front-end solution with components as the minimum aggregation unit, which solved the problems of traditional micro front-end in the aspects of granularity, performance loss and communication complexity. It also borrowed the dynamic configuration injection and automatic capacity expansion capabilities of serverless to

DOI: 10.54254/2755-2721/2026.33975

realize the independent deployment and on-demand scheduling of components, and greatly reduced the difficulty of isolation and communication between different frameworks [10]. This scheme improves the flexibility of front-end development and operation and maintenance efficiency, but the inter component state management and cross component debugging are still complex.

In terms of artificial intelligence experimental teaching platform, lizehui and others designed and implemented an artificial intelligence experimental platform based on serverless architecture, which combined container construction, workload management, event triggering and other mechanisms to support multi-user concurrent experiments and dynamic capacity expansion, and integrated the scikit learn algorithm library and common data sets with docker image, making the migration and reuse of AI basic algorithms extremely convenient [11]. The platform reduces the deployment cost of the teaching environment, but in the face of large-scale deep learning training tasks, the computing resources and storage persistence ability are still insufficient.

4. Main challenges of serverless computing

At present, serverless computing still faces many challenges, which need to be broken through in future research.

Due to the limited computing power of single function, and the immature inter function communication and collaborative scheduling mechanism, serverless architecture is difficult to fully support large and complex tasks (such as large-scale data processing, deep model training, etc.) [2]. Existing functions usually require stateless and short execution time, and lack native support for stateful and long-running tasks.

In ubiquitous cloud and edge computing scenarios, resources are scattered in multiple geographical regions, and network latency and bandwidth differ significantly. At present, the cross regional function scheduling strategy has not been widely used in the industry, lacking effective resource awareness and dynamic routing mechanism [2]. Especially in the private cloud scenario, the lack of resource elasticity is difficult to adapt to sudden workload, and relying solely on the public serverless platform may bring the risk of data security and cost out of control. Yu et al. tried to realize the collaborative scheduling between private cloud and serverless platform through the task unloading framework [9]. The Dato heuristic algorithm proposed by Yu et al. Significantly reduced the economic cost on the premise of meeting the task deadline. However, the modeling of network delay and dependency between functions is still relatively simplified, and the performance under large-scale workflow has not been fully verified.

In serverless applications, the number of functions is large, the life cycle is short, the attack surface increases, and the risk of unauthorized access increases. The existing security detection methods still have room for improvement in the integrity guarantee of control flow and data flow between functions [5]. In addition, resource isolation in multi tenant environment, side channel attack protection during function operation, and secure transmission and storage of sensitive data also need to be solved.

Serverless application development, debugging, monitoring, tracking and other tool chains are not yet mature, affecting development efficiency and system observability [7, 10]. The distributed tracing, log aggregation, performance bottleneck positioning and other capabilities of the function call chain are far less perfect than the traditional microservice architecture, which brings great difficulties to operation and maintenance.

Although Kumar et al. paid attention to the energy efficiency scheduling problem, there are still few energy consumption modeling and green scheduling strategies for serverless computing in the existing literature [8]. With the increasing attention paid to the carbon footprint of data centers, how

DOI: 10.54254/2755-2721/2026.33975

to reduce the overall energy consumption of serverless clusters under the premise of ensuring performance and quality of service has become a new research frontier.

5. Future outlook

Looking forward to the future, serverless computing will continue to evolve in the direction of intelligence, efficiency and security. The following trends deserve attention.

Artificial intelligence technology (such as deep reinforcement learning, online learning, etc.) is used to realize the intelligentization of resource scheduling, adaptively perceive workload fluctuations and the competition state of underlying resources, and dynamically adjust the number and placement strategy of function instances, so as to improve the overall performance and energy efficiency of the system [2, 8]. In the future, it can further combine workload prediction and anomaly detection to achieve active elastic scaling.

Build a serverless platform with cross regional and multi-level computing power integration, integrate cloud data centers, edge nodes and terminal devices, and form a serverless computing ecosystem of cloud edge end collaboration. This requires solving key technical issues such as cross domain resource discovery, data aware routing, heterogeneous hardware adaptation, and delay sensitive task scheduling to meet the needs of ubiquitous cloud services [2].

Based on trusted execution environment (TEE), webassembly sandbox and other technologies, the trusted function runtime is constructed to realize strong isolation between functions, data encryption calculation and secure remote certification. At the same time, develop lightweight security monitoring and anomaly detection methods for serverless, such as cross function behavior analysis , zero trust access control, etc., to further enhance data security and privacy protection capabilities [5, 7].

Promote the standardization of serverless API, tool chain and monitoring system, reduce the migration cost between different cloud platforms, and promote the prosperity of open source ecology. Standards including functional programming model, event specification, observability interface, etc. will help to form a unified serverless computing community and accelerate technology maturity and industrial implementation [4, 10].

A special serverless runtime and scheduling strategy are designed for typical scenarios such as artificial intelligence reasoning, real-time data processing, and Internet of things event response. For example, model caching and batch processing mechanisms are introduced for AI inference functions, and cold start and network communication are optimized for edge serverless.

6. Conclusion

This paper reviews the core technology, research status and future trends of serverless computing. As one of the most typical paradigms of cloud native, serverless computing has had a profound impact on the way of application development and deployment. It has achieved effective results in cold start, resource scheduling, security protection, architecture integration and other technologies, and can play a greater role in ubiquitous cloud, edge computing, AI services and other scenarios. However, in order to truly realize the extensive implementation of serverless computing, researchers still need to overcome the challenges of large-scale task support, cross domain scheduling, security isolation, observability and so on. Future research should start from the balance of performance, security and observability, and gradually build an efficient, reliable, intelligent and adaptive serverless computing ecosystem.

DOI: 10.54254/2755-2721/2026.33975

References

[1] Chen, C., Zhou, C. Y., Li, H. M., (2025). Key technologies and development of serverless computing. naval electronic engineering, 45(02): 17-21
[2] Yang, Y. N., Zhang, J. S., Wu, J., (2025). Opportunities and challenges of serverless computing in ubiquitous cloud scenario. computing, 1(05): 71-81
[3] Kumari, A., Behra, R. K., Sahu, K. S., Sahu, B., Gandomi, A. H., (2026). The rise of serverless computing in the cloud: a cutting-edge review of challenges and opportunities. machine learning applications, 24
[4] Yang, G., Liu, J., Qu, M. Z., Wang, S., Ye, D., Zhong, H., (2025). Review on performance optimization technology of server insensitive computing system. Journal of software, 36(01): 47-78
[5] Zhan, D. Y., Huang, Z. L., Tan, K., Yu, Z. F., He, Z., Zhang, H. L., (2025). Cross functional behavior analysis and constraint technology for serverless applications. information network security, 25(09): 1329-1337
[6] Wang, X. N., Li, S. Y., (2025). Research on Optimization and performance evaluation of serverless architecture based on cloud native technology. Journal of Jiamusi University (NATURAL SCIENCE EDITION), 43(01): 133-135
[7] Huang, Z. C., (2024). Design and implementation of deep learning platform based on webassembly and serverless architecture. Guizhou University
[8] Sartengdekumar, Jain, S., Singh, A. K., (2026). Energy saving in server free computing environment deep reinforcement learning based application scheduling. Journal of supercomputing, 82(4)
[9] Yu, S. K., Wu, Q. W., Cai, K., Guo, T. L., Jiang, Z., Sun, T. H., (2026). Efficient offload workflow in private cloud using serverless computing. expert system and applications, 312
[10] Li, Y. P., (2023). Research and application of micro front end technology based on serverless. Harbin Institute of technology
[11] Li, Z. H., Zhang, X. Y., (2024). Design and implementation of artificial intelligence experimental platform based on serverless architecture. computer and digital engineering, 52(02): 590-597