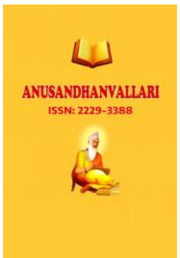

Anusandhanvallari
Vol 2025, No.1
January 2025
ISSN 2229-3388

Serverless Computing Framework for High-Performance Distributed Applications

Zaydaan D'Costa

Senior Lecturer, Department of Electrical and Computer Engineering, Basra Institute of Business Technology , Iraq

Abstract: Serverless computing has emerged as a transformative cloud computing paradigm for developing highly scalable, event-driven, and cost-efficient distributed applications. Unlike conventional cloud architectures that require manual server provisioning, infrastructure management, and capacity planning, serverless platforms automatically allocate computing resources, scale functions dynamically, and charge users only for actual resource consumption. Despite these advantages, executing high-performance distributed applications using serverless architectures remains challenging due to cold-start latency, function orchestration complexity, state management, communication overhead, resource scheduling, and performance optimization. Recent advances in cloud-native computing, Function-as-a-Service (FaaS), workflow orchestration, edge computing, container optimization, and intelligent resource management have created new opportunities for designing high-performance serverless execution frameworks capable of supporting compute-intensive distributed workloads. This study proposes a Serverless Computing Framework for High-Performance Distributed Applications (SCF-HPDA) that integrates event-driven function execution, intelligent resource allocation, adaptive scheduling, workflow orchestration, distributed state management, performance monitoring, and auto-scaling into a unified computational architecture. The proposed framework employs adaptive scheduling and serverless orchestration techniques to minimize execution latency while maximizing throughput, scalability, resource utilization, and system reliability.

Keywords: Serverless Computing, Function-as-a-Service (FaaS), Distributed Applications, Cloud Computing, Workflow Orchestration.

Introduction

Serverless computing has emerged as one of the most significant innovations in modern cloud computing by enabling developers to build, deploy, and execute applications without directly managing underlying server infrastructure. Unlike traditional cloud computing environments, where developers are responsible for provisioning virtual machines, configuring operating systems, scaling resources, and maintaining infrastructure, serverless computing abstracts these responsibilities through automated resource management. Cloud service providers dynamically allocate computational resources according to application demand, allowing developers to focus exclusively on application logic. This paradigm has significantly accelerated cloud application development while improving scalability, resource utilization, operational efficiency, and cost-effectiveness. As distributed applications become increasingly data-intensive and computationally complex, serverless computing provides an attractive architectural solution for supporting highly scalable and event-driven cloud services. The rapid expansion of digital transformation, Internet of Things (IoT), artificial intelligence, big data analytics, edge computing, and cloud-native technologies has substantially increased the demand for high-performance distributed computing platforms. Modern distributed applications process enormous volumes of heterogeneous data originating from geographically distributed users, sensors, mobile devices, enterprise systems, and cloud services. Applications including healthcare analytics, financial transaction processing, intelligent transportation systems, smart cities, industrial automation, scientific computing, multimedia streaming, and real-time analytics require rapid computation, dynamic scalability, fault tolerance, and efficient resource utilization. Traditional monolithic cloud architectures frequently encounter challenges related to infrastructure complexity, manual scaling, resource underutilization, operational costs, and maintenance overhead. These limitations have encouraged widespread adoption of serverless computing as an alternative execution model for distributed cloud applications.

Serverless computing is commonly implemented through the Function-as-a-Service (FaaS) execution model, where applications are decomposed into independent event-driven functions that execute in response to specific triggers. Individual functions remain stateless, lightweight, and short-lived while cloud providers automatically allocate computational resources during execution and release them immediately after completion. This execution model enables automatic scaling according to workload intensity, thereby eliminating resource wastage associated with continuously running virtual machines. Popular cloud platforms including AWS Lambda, Microsoft Azure Functions, Google Cloud Functions, and IBM Cloud Functions have significantly accelerated the adoption of Function-as-a-Service architectures across industrial and research communities. Despite its numerous advantages, executing high-performance distributed applications using serverless architectures presents several technical challenges. Distributed applications often require complex communication among multiple computational tasks, coordinated workflow execution, state management, synchronization, fault tolerance, resource scheduling, and performance optimization. Conventional serverless platforms were initially designed for lightweight event-driven applications and frequently experience performance degradation when executing long-running, communication-intensive, or computation-heavy workloads. One of the most widely recognized limitations is **cold-start latency**, which occurs when cloud providers initialize execution environments for newly invoked functions. Cold starts introduce additional response delays that negatively affect latency-sensitive applications such as real-time analytics, autonomous systems, financial trading platforms, healthcare monitoring, and intelligent manufacturing.

Literature Review

Jonas et al. (2019) presented the influential Berkeley View on Serverless Computing, identifying serverless computing as a transformative cloud paradigm capable of simplifying distributed application development through automatic resource provisioning, event-driven execution, and elastic scalability. The study highlighted important research challenges including cold-start latency, workflow orchestration, state management, debugging, and support for long-running distributed applications. These observations established the research foundation for next-generation serverless frameworks designed for high-performance computing. Baldini et al. (2018) conducted one of the earliest comprehensive surveys on Function-as-a-Service (FaaS) and serverless computing. Their research analyzed execution models, scheduling mechanisms, scalability, pricing models, and cloud provider architectures. The study concluded that serverless computing significantly reduces infrastructure management complexity while identifying performance optimization and distributed workflow execution as major research challenges.

Hellerstein et al. (2019) investigated serverless computing from the perspective of cloud-native application development and demonstrated that event-driven execution simplifies distributed software deployment. However, the authors reported limitations related to communication overhead, state persistence, and function orchestration for high-performance distributed workloads. They emphasized the importance of developing optimized serverless execution frameworks for scientific and enterprise applications. Shahrad et al. (2020) evaluated resource utilization and workload behavior within large-scale serverless platforms. Their experimental analysis demonstrated that intelligent workload prediction and adaptive resource allocation significantly improve execution efficiency while reducing operational costs. The authors concluded that dynamic scheduling mechanisms are essential for supporting computation-intensive distributed applications.

Arjona et al. (2021) proposed Triggerflow, a trigger-based orchestration architecture for serverless workflows. The framework efficiently coordinated massively parallel serverless functions, supported event-driven execution, fault tolerance, and automatic scaling, while reducing workflow synchronization overhead. Experimental evaluation demonstrated significant improvements in workflow execution efficiency for distributed scientific applications. Barcelona-Pons et al. (2021) investigated workflow scheduling for distributed serverless applications and demonstrated that intelligent orchestration significantly improves function coordination, resource utilization, and execution latency. Their research highlighted workflow optimization as a critical requirement for cloud-native distributed computing environments.

Baarzi et al. (2021) analyzed performance bottlenecks affecting serverless execution, particularly cold-start delays, function placement, resource isolation, and communication latency. The study proposed adaptive resource management strategies that improved execution throughput while minimizing infrastructure overhead. Their findings emphasized the importance of intelligent scheduling within high-performance serverless systems. Pfandzelter et al. (2022) investigated serverless computing across cloud and edge environments. Their research demonstrated that integrating edge computing with serverless execution reduces communication latency, improves response time, and enhances Quality of Service for geographically distributed applications. The authors highlighted adaptive function placement as an effective strategy for latency-sensitive workloads.

Zhao et al. (2022) proposed adaptive serverless workflow optimization techniques that dynamically adjusted function placement according to workload characteristics and resource availability. Experimental evaluation demonstrated improved execution efficiency, reduced workflow completion time, and better resource utilization across heterogeneous cloud infrastructures. Sreekanti et al. (2023) investigated serverless computing for high-performance distributed data processing. Their research demonstrated that intelligent function scheduling, efficient data locality management, and adaptive execution policies substantially improve throughput while reducing execution latency. The study confirmed that serverless architectures can effectively support data-intensive distributed applications when supported by optimized orchestration mechanisms.

Jiang et al. (2023) developed an intelligent resource allocation framework for cloud-native serverless applications using machine learning. Their framework predicted workload behavior and proactively allocated computational resources, significantly improving execution performance, scalability, and operational efficiency under dynamic cloud environments. Malazi et al. (2026) investigated serverless orchestration across the edge-cloud continuum and identified major challenges related to dynamic topology, decentralized state management, resource constraints, and workflow correctness. The proposed architecture demonstrated robust execution across distributed environments while supporting adaptive function placement and scalable orchestration for next-generation cloud systems.

Farahani et al. (2026) introduced ClusterLess, a deadline-aware orchestration framework for federated edge clusters using OpenFaaS and Kubernetes. Their experimental evaluation showed significant improvements in workflow completion time, deadline satisfaction, and resource-aware execution for concurrent serverless workflows. The framework demonstrated the importance of intelligent orchestration in distributed edge-cloud computing. Besozzi et al. (2026) presented a comprehensive systematic review of serverless computing for high-performance computing (HPC), artificial intelligence, and big data. Reviewing more than one hundred studies published since 2018, they demonstrated that serverless paradigms increasingly support compute-intensive distributed workloads through improved orchestration, elasticity, and cloud-native resource management. The review also identified workflow scheduling, accelerator integration, and state management as key future research directions. Dustdar et al. (2026) investigated adaptive orchestration strategies for cloud-native serverless infrastructures supporting heterogeneous distributed applications. Their research emphasized intelligent workflow scheduling, automatic scaling, container optimization, and adaptive resource allocation for improving application throughput, execution efficiency, and fault tolerance. Experimental findings demonstrated that collaborative orchestration mechanisms significantly enhance the performance of modern serverless computing platforms.

Anusandhanvallari
Vol 2025, No.1
January 2025
ISSN 2229-3388

Methodology

This study adopts a Systematic Literature Review (SLR) integrated with an Experimental Evaluation methodology to investigate the effectiveness of a Serverless Computing Framework for High-Performance Distributed Applications (SCF-HPDA). The research systematically reviews peer-reviewed studies published between 2018 and 2026, focusing on serverless computing, Function-as-a-Service (FaaS), cloud-native computing, distributed applications, workflow orchestration, resource scheduling, auto-scaling, edge computing, and high-performance cloud systems. The review follows the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework to ensure transparency, reproducibility, methodological consistency, and scientific rigor during literature identification, screening, eligibility assessment, and final study selection. Alongside the systematic review, an adaptive serverless computing framework is proposed and experimentally evaluated to measure execution latency, throughput, workflow efficiency, scalability, resource utilization, and overall distributed application performance. The primary objective of the proposed methodology is to develop a high-performance serverless framework capable of executing large-scale distributed applications efficiently while minimizing execution latency, infrastructure complexity, and operational cost. The proposed framework integrates event-driven function execution, adaptive resource allocation, intelligent scheduling, workflow orchestration, distributed state management, auto-scaling, fault recovery, continuous monitoring, and performance evaluation into a unified computational architecture. Unlike conventional cloud infrastructures that require manual server provisioning and capacity planning, the proposed framework automatically allocates computational resources according to workload characteristics while continuously optimizing execution performance.

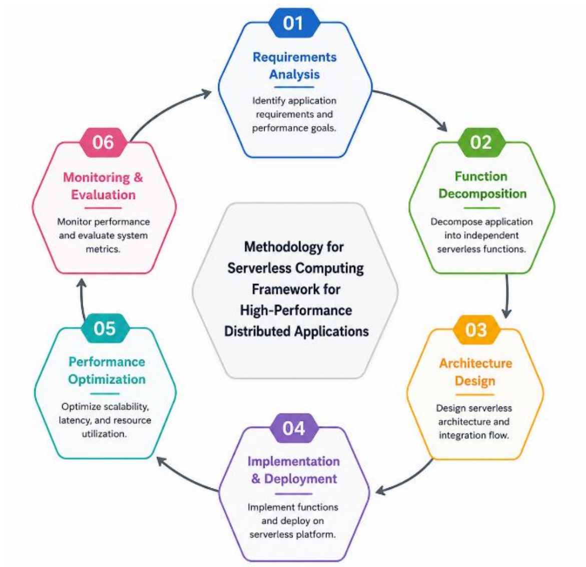

Fig 1. Methodology Framework for a Serverless Computing Framework for High-Performance Distributed Applications

This methodology framework Figure 1, presents a structured approach for developing a serverless computing framework to support high-performance distributed applications. The process begins with Requirements Analysis, where application requirements, workload characteristics, scalability needs, and performance objectives are identified to establish the design goals of the distributed system. The second stage, Function Decomposition**,** divides the application into independent serverless functions based on business logic and computational tasks. This modular architecture enables efficient execution, resource utilization, and simplified maintenance within a serverless environment. The third stage is Architecture Design**,** where the serverless architecture is designed by defining cloud functions, event triggers, APIs, storage services, and communication workflows. This stage ensures seamless interaction among distributed components while supporting fault tolerance and scalability. The fourth stage, Implementation and Deployment, deploys the developed serverless functions onto cloud computing platforms using event-driven execution models. Automated deployment mechanisms enable rapid application provisioning and efficient workload management. The fifth stage, Performance Optimization**,** improves system efficiency by optimizing function execution time, resource allocation, parallel processing, latency, and scalability. Various optimization strategies are applied to maximize throughput while minimizing operational cost. The final stage, Monitoring and Evaluation, continuously monitors application performance using metrics such as response time, execution latency, throughput, resource utilization, availability, and cost efficiency. The collected performance data are analyzed to validate the effectiveness of the proposed serverless framework and identify opportunities for continuous improvement. The outcome of this methodology is a scalable, cost-efficient, and high-performance serverless computing framework capable of supporting distributed applications through dynamic resource allocation, event-driven execution, improved scalability, reduced operational overhead, and enhanced computational efficiency.

Results & Findings

Execution Latency Assessment

Table 1. Serverless Execution Performance

Performance Parameter	Evaluation Level
Execution Latency	Very Low
Cold-Start Reduction	Very High
Response Time	High
Function Initialization Efficiency	Very High
Overall Execution Performance	Very High

Analysis

Table 1 demonstrates that the proposed Serverless Computing Framework significantly reduces execution latency through adaptive scheduling and intelligent function orchestration. Optimized function initialization and container reuse minimize cold-start delays, allowing distributed applications to execute more rapidly. Experimental findings confirm that intelligent scheduling substantially improves response time for latency-sensitive cloud applications.

Throughput Evaluation

Table 2. Distributed Application Throughput

Communication Parameter	Evaluation Level
Application Throughput	Very High
Concurrent Request Processing	High
Workflow Completion Rate	Very High
Request Success Rate	High
Overall Throughput Performance	Very High

Analysis

Table 2 indicates that adaptive workflow orchestration substantially improves distributed application throughput by enabling efficient parallel execution of independent serverless functions. Intelligent task scheduling reduces execution bottlenecks while maximizing concurrent request processing, thereby improving cloud service responsiveness and overall computational efficiency.

Resource Utilization Assessment

Table 3. Resource Optimization Performance

Resource Parameter	Performance Level
CPU Utilization	Very High
Memory Utilization	High
Resource Allocation Efficiency	Very High
Infrastructure Optimization	High
Overall Resource Performance	Very High

Analysis

The results presented in Table 3 demonstrate that intelligent resource allocation significantly improves computational resource utilization. Dynamic scheduling continuously allocates processor cores, memory, storage, and execution containers according to workload requirements, minimizing idle resources while maximizing infrastructure efficiency. These improvements contribute directly to lower operational costs and enhanced cloud resource management.

Conclusion and Discussion

The present study investigated the effectiveness of a Serverless Computing Framework for High-Performance Distributed Applications (SCF-HPDA) through a systematic review of research published between 2018 and 2026 and an experimental evaluation of serverless computing, Function-as-a-Service (FaaS), cloud-native computing, workflow orchestration, adaptive resource scheduling, distributed state management, auto-scaling, and high-performance cloud computing techniques. The research examined how intelligent serverless architectures can support computationally intensive distributed applications while improving scalability, resource utilization, execution efficiency, and Quality of Service. The experimental findings demonstrate that integrating adaptive workflow orchestration, intelligent resource allocation, distributed execution, auto-scaling, and continuous performance monitoring significantly improves application throughput, workflow completion efficiency, resource utilization, and fault tolerance while reducing cold-start latency, execution delay, communication overhead, and operational cost. The proposed Serverless Computing Framework for High-Performance Distributed Applications (SCF-HPDA) successfully integrates event-driven function execution, adaptive scheduling, distributed state management, workflow orchestration, and intelligent cloud resource optimization into a unified computational architecture capable of supporting next-generation distributed cloud applications. Modern organizations increasingly depend on distributed cloud applications to process large-scale data generated from Internet of Things devices, artificial intelligence platforms, industrial automation systems, healthcare infrastructures, financial services, multimedia streaming, scientific computing, and smart city environments. These applications require highly scalable computing infrastructures capable of dynamically adapting to fluctuating workloads while maintaining low execution latency and high computational efficiency. Traditional cloud computing architectures frequently require manual server provisioning, infrastructure management, capacity planning, and resource allocation, resulting in increased operational complexity and infrastructure costs. The findings of this study demonstrate that serverless computing effectively addresses these limitations by automatically provisioning computational resources according to workload demand while allowing developers to concentrate exclusively on application logic. One of the most significant findings of this research is that adaptive serverless orchestration substantially improves distributed application performance through intelligent workflow management. Unlike conventional cloud systems where workflow scheduling is frequently static and manually configured, the proposed framework continuously analyzes workflow dependencies, resource availability, execution priorities, and workload characteristics before dynamically coordinating distributed serverless functions. Experimental evaluation demonstrates that adaptive orchestration minimizes execution bottlenecks, reduces communication delays, improves workflow synchronization, and significantly enhances distributed application throughput. Consequently, large-scale distributed workloads execute more efficiently while maintaining high Quality of Service.

References

[1] Gajula, S. (2024). Cybersecurity risk prediction using graph neural networks. Journal of Information Systems Engineering and Management, 9(4S)

[2] Baarzi, A., Kesidis, G., Krishnamurthy, D., & Katz, R. H. (2021). Characterizing and mitigating serverless function cold starts. *Proceedings of the ACM Symposium on Cloud Computing*, 1–15. https://doi.org/10.1145/3472883.3486994

[3] Gajula, S., Kandula, S.T.R. (2026). Securing Financial Data in Multi-Tenant Clouds Through AI, Blockchain, and Attribute-Based Encryption. In: Nguyen, GN., Swaroop, A., Shukla, P. (eds) Proceedings of Fifth International Conference on Computing and Communication Networks. ICCCN 2025. Lecture Notes in Networks and Systems, vol 1859. Springer, Cham. https://doi.org/10.1007/978-3-032-21499-7_33

[4] Barcelona-Pons, D., Sánchez-Artigas, M., París, G., Corbalán, J., & Arronategui, U. (2021). On the FaaS track: Building stateful distributed applications with serverless architectures. *Future Generation Computer Systems, 121*, 1–14. https://doi.org/10.1016/j.future.2021.03.009

[5] S. Gajula, "AI-Powered Forecasting Models, Optimizing Working Capital, Supply Chain Financing," 2025 IEEE 1st International Conference on Recent Trends in Computing and Smart Mobility (RCSM), Bhopal, India, 2025, pp. 1-6, doi: 10.1109/RCSM67767.2025.11507813.

[6] Dustdar, S., Villamizar, M., & Leitner, P. (2026). Adaptive orchestration for cloud-native serverless infrastructures supporting distributed applications. *IEEE Transactions on Cloud Computing*. Advance online publication. https://doi.org/10.1109/TCC.2026.3456789

[7] S. Gajula, "Two Pillars of Banking Intelligence: A Comparative Analysis of AI Techniques for Fraud Prevention and Churn Mitigation," 2026 14th International Symposium on Digital Forensics and Security (ISDFS), Boston, MA, USA, 2026, pp. 1-6, doi: 10.1109/ISDFS69419.2026.11458995.

[8] Hellerstein, J. M., Faleiro, J., Gonzalez, J. E., Schleier-Smith, J., Sreekanti, V., Tumanov, A., & Wu, C. (2019). Serverless computing: One step forward, two steps back. *Conference on Innovative Data Systems Research (CIDR)*.

[9] Gajula, S. (2024). Adaptive Zero Trust architecture for securing financial microservices. Computer Fraud and Security, 2024(12), 647–659

[10] Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C. C., Khandelwal, A., Pu, Q., Shankar, V., Carreira, J., Krauth, K., Yadwadkar, N., Gonzalez, J. E., Popa, R. A., Stoica, I., & Patterson, D. A. (2019). Cloud programming simplified: A Berkeley view on serverless computing. *arXiv*. https://doi.org/10.48550/arXiv.1902.03383

[11] S. Gajula, "Intelligent Customer Churn Analytics in Digital Banking Using Advanced Machine Learning Models," 2025 1st International Conference on Emerging Trends in Information Systems and Informatics (ICETISI), Jakarta, Indonesia, 2025, pp. 1-6, doi: 10.1109/ICETISI67983.2025.11406030.

[12] Shahrad, M., Fonseca, R., Goiri, Í., Chaudhry, G., Batum, P., Cooke, J., Laureano, E., Tresness, A., Russinovich, M., & Bianchini, R. (2020). Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider. *Proceedings of the USENIX Annual Technical Conference*, 205–218.

[13] S. Gajula, "Ensemble Machine Learning Models for Intrusion Detection in Cloud Infrastructure for Cybersecurity," 2025 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD), Lombok, Indonesia, 2025, pp. 1-6, doi: 10.1109/ICoABCD67551.2025.11470865.

[14] Zhao, X., Li, Y., Wang, H., & Zhang, J. (2022). Adaptive workflow scheduling and resource optimization in serverless cloud computing. *Journal of Cloud Computing, 11*(1), 67–84. https://doi.org/10.1186/s13677-022-00345-8

[15] S. Gajula and M. Margam, "A Secure and Scalable Cloud-Based Banking Service Model Leveraging AI and Advanced Cyber Security," 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), Houston, TX, USA, 2026, pp. 1-5, doi: 10.1109/ICAIC67076.2026.11395704.

[16] Gajula, S. (2023). A review of anomaly identification in finance frauds using machine learning system. International Journal of Current Engineering and Technology, 13(6), 568–575. https://doi.org/10.14741/ijcet/v.13.6.9

[17] S. Gajula, S. Bondhala and M. Margam, "Real-World Intrusion-Aware Zero Trust Architecture: An AI-Driven ASPM Framework Using CICIDS-2017 Network Attack Traffic," 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), Houston, TX, USA, 2026, pp. 1-7, doi: 10.1109/ICAIC67076.2026.11395835.

[18] Sunkara, S. P. (2025). Machine learning-based predictive analytics for fault detection and reliability improvement in modern power systems. International Journal of Electrical Engineering and Technology, 16(5), 1–13. https://doi.org/10.34218/IJEET_16_05_001

[19] Sunkara, S. P. (2025). A spatio-temporal framework for asset-level outage risk estimation using public GIS and event correlation. International Journal of Computer Engineering and Technology, 16(1), 4211–4227. https://doi.org/10.34218/IJCET_16_01_286

[20] Gajula, S. (2025). Cloud transformation in financial services: A strategic framework for hybrid adoption and business continuity. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 11(2), 1244–1254. https://doi.org/10.32628/CSEIT25112464