2026 Volume: 02 Issue: 01 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 01-09

SERVERLESS COMPUTING: A NEW ERA OF APPLICATION DEPLOYMENT

Pulkit Gupta, Er. Ram Babu Buri, Dr. Vishal Shrivastava, Dr. Akhil Pandey

Computer Science & Engineering, Arya College of Engineering & I.T. Jaipur, India.

Article Received: 15 December 2025
Article Revised: 03 January 2026
Published on: 23 January 2026

Corresponding Author : *Pulkit Gupta
Computer Science & Engineering, Arya College of Engineering & I.T. Jaipur, India.

DOI : https://doi-doi.org/101555/ijrpa.9747

ABSTRACT

Serverless computing – a cloud-native model where infrastructure is abstracted away – is redefining the way modern applications are built and deployed. In this paradigm, developers write stateless functions that are triggered by events, while cloud providers handle provisioning, scaling, and fault tolerance. Unlike traditional server-based or containerized approaches, serverless emphasizes event-driven execution and a pay-as-you-go billing model, making it highly cost-efficient for unpredictable workloads. Leading platforms such as AWS Lambda, Azure Functions, and Google Cloud Functions support diverse triggers and integrations, enabling real-time processing, web backends, IoT pipelines, and automation tasks. Performance challenges, however, persist: cold start latency introduces delays in function execution, vendor lock-in limits portability, and monitoring distributed stateless services remains complex. Recent advancements show promise in mitigating these issues, including cold start optimizations (e.g. AWS SnapStart), serverless containers (Knative, AWS Fargate), and hybrid models that blend serverless with virtual machines. Industry reports forecast rapid adoption, with the global market projected to grow from USD 24.51 billion in 2024 to over USD 52 billion by 2030, reflecting a CAGR above 14%. Research is also exploring integration of serverless with AI/ML workloads and multi-cloud strategies to overcome vendor dependency. This paper reviews the evolution of serverless computing, outlines its architecture, highlights benefits and limitations, and examines empirical studies on performance and cost-effectiveness. Finally, open challenges such as debugging complexity, execution limits, and enterprise-scale adoption are discussed, along with future trends shaping this new era of cloud-native deployment.

INTRODUCTION

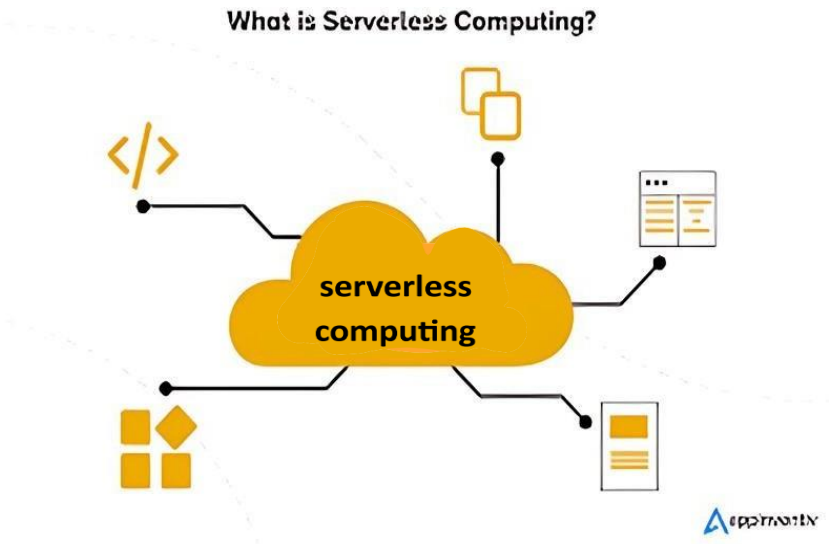
Since the rise of cloud computing in the early 2000s, enterprises and developers have relied on virtual machines and later containers to deploy scalable applications. Early models such as Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) allowed teams to shift from physical hardware to managed platforms, but still required oversight of operating systems, scaling, and runtime environments. The emergence of Function-as-aService (FaaS) in 2014, pioneered by AWS Lambda, marked the beginning of serverless computing – a paradigm where infrastructure management is completely abstracted away. In this model, applications are composed of stateless functions executed on-demand, while the underlying servers, resource allocation, and scaling are transparently handled by cloud providers.

The first wave of adoption was driven by startups and small businesses seeking agility without large infrastructure costs. Over time, major organizations began integrating serverless into production systems for tasks such as realtime event processing, web backends, and IoT data pipelines. By decoupling business logic from infrastructure, serverless architectures enabled shorter development cycles and reduced operational overhead. Similar to how virtualization revolutionized enterprise computing in the 2000s, serverless is now viewed as the next step in the evolution of cloud-native computing.

Industry statistics illustrate this rapid growth. In 2024, the global serverless computing market was valued at USD 24.51 billion, with forecasts projecting an expansion to USD 52.13 billion by 2030 at a 14.1% CAGR. North America currently dominates with ~40% of the market, while Asia-Pacific is the fastest-growing region, driven by digital transformation initiatives in India and China. Use cases continue to diversify, with serverless increasingly powering machine learning inference pipelines, chatbots, scheduled automation, and edge computing applications. Despite its promise, serverless computing faces limitations. Execution environments are short-lived and subject to cold start latency, often introducing delays of hundreds of milliseconds to several seconds.

Functions typically run in stateless containers with strict time and resource limits, making them unsuitable for long-running tasks. Furthermore, vendor lock-in remains a concern, as organizations that tightly couple workloads with a specific provider’s ecosystem face challenges in migrating to alternative platforms. These concerns mirror earlier debates during the rise of cloud computing, highlighting the balance between convenience and flexibility.

Recent research and development aim to address these issues. Techniques such as provisioned concurrency and AWS SnapStart reduce cold start penalties, while open-source frameworks like Knative and OpenFaaS promote portability across providers. Hybrid approaches that lend virtual machines, containers, and serverless functions are emerging to combine elasticity with predictability. In parallel, enterprises are experimenting with integrating serverless into I/ML workflows, real-time analytics, and multi-cloud architectures to overcome vendor dependency.



PROCEDURES AND METHODS

1. Serverless Architecture and Execution Model

Serverless computing operates on a fundamentally different architecture compared to traditional cloud models. Applications are decomposed into small, stateless functions that are executed in isolated containers. These containers are ephemeral in nature—spun up when a function is invoked and destroyed immediately after execution completes. This ensures optimal resource utilization and eliminates the cost of idle servers. Functions are typically triggered by events, such as HTTP requests via an API Gateway, changes in a database record, messages arriving on a queue, or file uploads to a storage service.

For example, in AWS Lambda, a function might be automatically triggered when a user uploads an image to Amazon S3. The Lambda function can then process the image (e.g., compress, resize, or classify it) and store the output in another storage bucket, all without provisioning or managing a single server. The event-driven model underpins this architecture, allowing highly reactive applications that scale seamlessly with workload. Execution is billed in fine-grained units (e.g., gigabyte-seconds for AWS Lambda), ensuring that users pay only

for actual compute usage rather than for provisioned capacity.

2. Key Features of Serverless Computing

Serverless platforms exhibit several defining features that distinguish them from conventional cloud deployment models:

- No Server Management: Developers are entirely abstracted from the complexity of provisioning, configuring, and maintaining servers. The cloud provider is responsible for infrastructure lifecycle, security patches, and fault tolerance.
- Automatic Scaling: Functions scale seamlessly with demand. When a workload spikes—such as during peak traffic of an e-commerce sale—the serverless platform can instantiate hundreds or thousands of function instances in parallel, then scale down automatically once demand decreases.
- Pay-per-Use Billing: Unlike virtual machines or containers, which incur costs even when idle, serverless functions are billed only for execution time and memory consumption. This is particularly advantageous for intermittent workloads.
- High Availability and Reliability: Serverless providers distribute execution across multiple data centers, ensuring that functions are inherently fault-tolerant without user intervention. These characteristics make serverless computing especially appealing for applications with unpredictable traffic patterns or event-driven workloads.

3. Popular Platforms

Several commercial and open-source platforms dominate the serverless computing landscape, each offering unique strengths and integrations:

- AWS Lambda: The pioneering FaaS platform introduced in 2014, tightly integrated with AWS ecosystem services such as API Gateway, S3, DynamoDB, and CloudWatch. AWS Lambda is currently the market leader and widely adopted across industries.
- Microsoft Azure Functions: Designed for enterprise environments, Azure Functions integrates with Microsoft's ecosystem, including Azure Event Hub, Cosmos DB, and Office 365 services. It supports multiple programming languages and enterprise authentication mechanisms.
- Google Cloud Functions: Known for integration with data-intensive services like BigQuery,
- Firebase, and Cloud Pub/Sub. It is favored for analytics, mobile app backends, and real-time event processing.

- IBM Cloud Functions: Based on the open-source Apache OpenWhisk framework, IBM's offering emphasizes flexibility and enterprise-scale solutions.
- Open-Source Solutions: Tools like OpenFaaS, Kubeless, and Knative allow organizations to deploy serverless workloads on Kubernetes clusters, reducing dependency on commercial vendors and promoting multi-cloud strategies.

The choice of platform often depends on organizational requirements, cost considerations, and ecosystem compatibility.

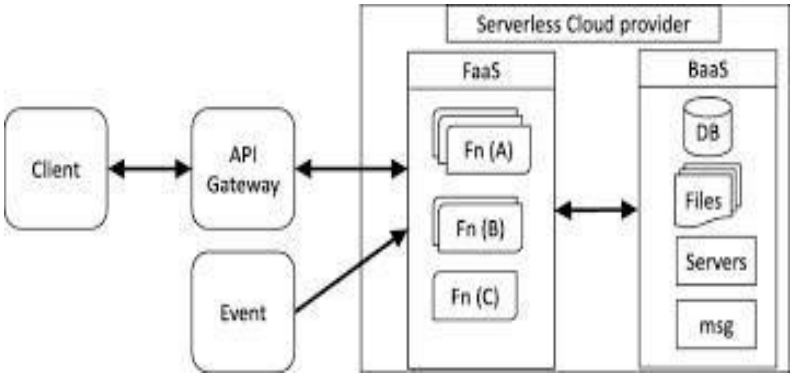
4. Advantages

Serverless computing delivers several significant advantages that contribute to its growing adoption:

Rapid Deployment and Faster Time to Market: Developers can focus exclusively on writing business logic without worrying about server provisioning or scaling, thereby accelerating software delivery cycles.

- Cost Savings for Variable Workloads: Since users are charged only for actual execution time, workloads with irregular or unpredictable usage patterns can yield cost reductions of up to 70% compared to traditional always-on servers.
- Increased Developer Productivity: By removing infrastructure management tasks, teams can dedicate more effort toward innovation, feature development, and improving customer experience.
- Scalability and Fault Tolerance: Scaling is automatic and built into the platform. Applications can handle sudden surges in demand without manual intervention, while redundancy ensures resilience against infrastructure failures.

These advantages have made serverless computing particularly attractive to startups, small businesses, and enterprises building microservices-based architectures.



6. Challenges

Despite its many benefits, serverless computing is not without limitations and operational concerns:

- Cold Start Latency: When a function is invoked after a period of inactivity, the platform must initialize a new container and runtime environment. This “cold start” can introduce latency ranging from 200 milliseconds for lightweight Node.js functions to over 6 seconds for Javabased applications. While acceptable for batch processing, it can negatively impact user- facing applications requiring low latency.
- Vendor Lock-in: Serverless platforms often provide proprietary integrations, making it challenging for organizations to migrate workloads between providers. For example, functions written for AWS Lambda may require significant modification to run on Google Cloud Functions.
- Limited Execution Time: Functions are constrained by maximum runtime limits (e.g., AWS Lambda allows up to 15 minutes per invocation). Long-running or compute-heavy tasks must be broken into smaller steps or ofloaded to other services.
- Debugging and Monitoring Difficulties: Because functions are stateless, ephemeral, and distributed across multiple environments, traditional debugging and monitoring tools are less effective. Specialized observability solutions are required to trace execution across eventdriven pipelines.

These challenges highlight the trade-offs inherent in adopting serverless architectures. Research and development efforts are actively addressing these issues through innovations like provisioned concurrency, serverless containers, and multi-cloud orchestration frameworks.

RESULTS

1. Market Trends

The global adoption of serverless computing has accelerated significantly in recent years, reflecting its growing importance in cloud-native application design. According to market research, the global serverless computing market was valued at USD 24.51 billion in 2024 and is projected to expand to USD 52.13 billion by 2030, representing a compound annual growth rate (CAGR) of approximately 14.1%. This growth trajectory underscores the increasing reliance of enterprises on Function-as-a- Service (FaaS) and related event-driven computing models for application deployment.

2. Performance Studies

Performance has been a central focus in evaluating the practicality of serverless platforms, particularly regarding cold start latency and cost optimization. Cold starts occur when a function is invoked after a period of inactivity, requiring the platform to initialize a container and runtime environment. In certain workloads, this can introduce delays detrimental to user experience. For instance, AWS reported that Java-based Lambda functions could exhibit cold start latencies as high as 6.1 seconds under certain conditions. With the introduction of AWS SnapStart, which preinitializes execution environments and caches them for reuse, the latency has been significantly reduced to 1.4 seconds, demonstrating substantial improvements in responsiveness for highperformance applications.

3. Applications and Use Cases

The versatility of serverless computing is reflected in its diverse applications across industries. One of the most common use cases is the development of web and mobile backends. Here, serverless functions are invoked through API gateways to handle user authentication, data retrieval, and session management. This model reduces the operational burden on development teams, enabling rapid iteration and scaling during unpredictable traffic surges such as e-commerce flash sales or product launches.

Serverless is also highly effective in managing IoT data pipelines, where millions of devices continuously generate small bursts of data. Functions can process, filter, and store incoming streams in real-time, making serverless ideal for smart city infrastructure, industrial monitoring, and connected healthcare applications. The ability to automatically scale with fluctuating device activity ensures cost efficiency and reliability in handling massive IoT workloads.

In addition, serverless computing is being increasingly adopted in AI/ML workloads, particularly for on- demand inference. Functions can be invoked to load trained machine learning models and provide predictions only when required, thereby reducing the cost of keeping inference endpoints continuously active. Google Cloud Functions, for example, is frequently paired with BigQuery and TensorFlow for real-time analytics and predictive modeling tasks.

Another significant application lies in automation and scheduled tasks. Organizations leverage serverless functions to perform ETL (Extract, Transform, Load) operations, system backups, log aggregation, and batch processing on a recurring basis. This event-driven

automation simplifies DevOps workflows and integrates seamlessly with CI/CD pipelines.

Collectively, these applications highlight how serverless computing extends beyond simple eventdriven triggers to form the foundation of modern distributed systems. Its combination of costeffectiveness, elasticity, and ease of integration positions it as a core enabler of cloud-native architectures.

CONCLUSION

Serverless computing has emerged as one of the most transformative paradigms in cloud-native application deployment, redefining how developers conceptualize, design, and deliver software. By abstracting away infrastructure management, serverless shifts the focus entirely toward business logic and innovation, thereby accelerating time-to-market and reducing operational complexity. This decoupling of application logic from server provisioning marks a continuation of the evolution that began with virtualization and containers, yet goes further by eliminating infrastructure management responsibilities altogether.

The empirical evidence from both industry reports and academic studies confirms the significance of this shift. Market analyses reveal a robust growth trajectory, with the global serverless computing market projected to more than double between 2024 and 2030, expanding from USD 24.51 billion to USD 52.13 billion at a CAGR exceeding 14%. This growth is driven not only by startups and small enterprises seeking agility but also by large organizations adopting serverless for mission-critical workloads, from e-commerce transaction systems to real-time IoT pipelines and machine learning inference engines. Function-as-a-Service remains the dominant segment, underscoring the industry's preference for lightweight, event-driven execution models that minimize idle costs.

Nevertheless, the widespread adoption of serverless is tempered by notable limitations. Chief among these are cold start latency, execution time restrictions, and vendor lock-in, all of which complicate its application to latency-sensitive and enterprise-scale systems. Debugging and observability remain non- trivial, given the distributed and ephemeral nature of serverless workloads. These challenges echo earlier transitions in computing paradigms, where technological gains were accompanied by operational complexities that demanded innovation.

REFERENCES

1. Amazon Web Services, AWS Lambda Documentation, Amazon Web Services, 2025.

[Online]. Available: https://aws.amazon.com/lambda

2. Microsoft, Azure Functions Overview, Microsoft Corporation, 2025. [Online]. Available: https://learn.microsoft.com/azure/azure-functions
3. Google Cloud, Serverless Computing Solutions, Google LLC, 2025. [Online]. Available: https://cloud.google.com/functions
4. IBM, IBM Cloud Functions Documentation, IBM Corporation, 2024. [Online]. Available: https://www.ibm.com/cloud/functions
5. Grand View Research, Serverless Computing Market Report, 2024–2030, Grand View Research, 2025.
6. Precedence Research, Global Serverless Computing Market Forecast 2025–2034, Precedence Research, 2025.
7. MarketsandMarkets, Serverless Architecture Market Report, MarketsandMarkets, 2025. Sedai.io, What Are Cold Starts in Lambda?, Sedai, 2024. [Online]. Available:
8. https://www.sedai.io/blog/what-is-cold-starts-in-lambda
9. AWS Compute Blog, Optimizing Cold Start Performance with AWS SnapStart, Amazon Web Services, 2024. [Online]. Available: https://aws.amazon.com/blogs/compute
10. CNCF, Knative Project Overview, Cloud Native Computing Foundation, 2025. [Online]. Availabl https://knative.dev
11. *OpenFaaS, Serverless Functions Made Simple, OpenFaaS Ltd., 2024. [Online]. Available: https://www.openfaas.com/*
12. *Kubeless, Kubernetes Native Serverless Framework, Kubeless Project, 2023. [Online]. Available: https://kubeless.io/*
13. *G. Adzic and R. Chatley, "Serverless Computing: Economic and Architectural Impact," in Proceedings of the IEEE Software Engineering Conference, 2018, pp. 1–10.*
14. *Baldini et al., "Serverless Computing: Current Trends and Open Problems," in Research Advances in Cloud Computing, Springer, 2017, pp. 1–20.*