

Evolution of Serverless Computing: A Detailed Analysis

Goutham Sabbani

MSc FinTech (UK), MA-ITM (US)*

Citation: Sabbani G. Evolution of Serverless Computing: A Detailed Analysis. *J Artif Intell Mach Learn & Data Sci* 2024, 2(2), 750-752. DOI: doi.org/10.51219/JAIMLD/goutham-sabbani/187

Received: 03 April, 2024; **Accepted:** 28 April, 2024; **Published:** 30 April, 2024

***Corresponding author:** Goutham Sabbani, MSc FinTech (UK), MA-ITM (US)

Copyright: © 2024 Sabbani G., This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ABSTRACT

The serverless computing that AWS Lambda condenses has transformed application deployment and handled over 3 trillion monthly, showing its vast adoption. Initially, serverless computing freed developers from server management, enabling focus on code. We will explore the evolution of several serverless architectures, as well as its benefits, limitations, and prospects.

Firstly, we will look at how this serverless technology has evolved in the early stage, examining how it began and the challenges it addressed. This has critical advancements and innovations that made its early adoption. Next, we will analyze the growth and adoption of serverless technologies such as Google Cloud Functions and Azure Functions, highlighting Functions as a service and auto-scaling capabilities, as well as event-driven execution models.

This analysis provides a comprehensive overview of the serverless computing journey from inception to the current stage, emphasizing its transformative impact on applications and deployment of this advanced technology.

Keywords: Serverless computing AWS Lambda, Google Cloud Functions, Azure Functions, Function-as-a-Service (FaaS), Cold start latency

1. Evolution of Serverless Computing

Serverless computing refers to abstracting underlying infrastructure, allowing developers to focus solely on writing code without worrying about server management. In this model, cloud service providers manage the provisioning, scaling, and maintenance of the servers. This approach decreases overhead, enhances productivity, and can lead to cost savings due to its pay-as-go pricing model.

For instance, AWS Lambda launched the Amazon Web Services in 2014, which is one of the most notable examples of serverless computing. AWS Lambda has significantly deployed serverless architecture across various services. Amazon's web service success is attributed to multiple things like ease of use, scalability, cost efficiency, and integration of AWS services³.

Serverless computing emerged in 2010 when cloud

computing technology evolved; serverless computing means it is like server management from an end user. Initial milestones were FaaS (Fast-as-a-service), where discrete pieces of code of the function to be executed in response to specific events without needing to provision the server. AWS was a serverless platform allowing developers to run code without provisioning or managing servers.

2. Growth and Adoption of Serverless Technologies

Serverless technologies have had remarkable benefits since their inception. There are several critical trends, like increased demand for agility, because organizations or individuals continuously seek more applications more quickly and efficiently. Serverless technology allows for rapid iteration and deployment, reducing time to market. The pay-as-go model has proven highly attractive, especially for startups and businesses

with variable workshops. By only paying for the resources we have used, we significantly reduce infrastructure costs.

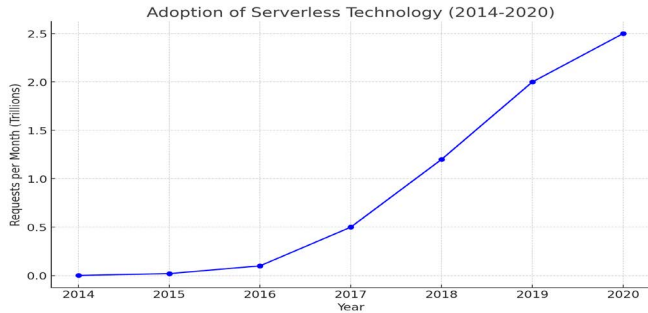


Figure 1: Adoption of serverless technology (2014-2020).
Source: Serverless computing case study²

The serverless platform has built-in scalability and high availability, which means application can have varying loads without manual intervention and improve user experience. This technology can abstract the complexity of managing infrastructure, allowing developers to focus on writing code and delivering features.

Adoption has happened on various platforms like AWS Lambda, which has set the stage for serverless computing. Google Cloud platforms provide a serverless execution environment for building and connecting servers. It supports a variety of use cases, including HTTP servers, real-time data processing, and more⁴.

FaaS functions as a service and is a crucial component of serverless computing, enabling developers to deploy individual functions as response time. The key feature of this technology is driven execution. These functions are triggered by specific requests such as HTTP requests, changing databases, or messages in the queue. Auto-scaling is also a key feature for FaaS because these can automatically scale up or down based on incoming requests. Ensuring efficient resource utilization. Here is a line chart summarising the growth of serverless computing technology.

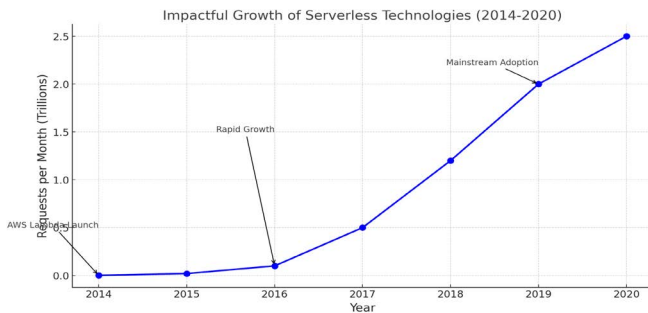


Figure 2: Impactful growth of serverless technologies (2014-2020).
Source: Serverless computing trends⁵

3. Benefits of Serverless Computing

Serverless computing, unlike the traditional methods, charges the users only based on the actual computing time used. Previously, models were based on those that required pre-allocated resources. Serverless computing eliminates the need to pay for idle server time. This pay model ensures the business model can optimize their IT expenditure, which is especially beneficial for startups and business variable workloads.

One of the most notable benefits of serviceless technology is its built-in scalability. Serverless platforms have automatically scaled applications in response to the number of incoming requests. This means the application can have various and vast amounts without any manual intervention, ensuring high scalability and performance during peak time.

A video streaming platform uses Google platform functions to process user uploads and generate thumbnails during popular events when many users upload videos at a time. The serverless platform scales up automatically to handle the increased load, ensuring smooth and uninterrupted service.

Here is a bar chart showing all the benefits of serverless computing

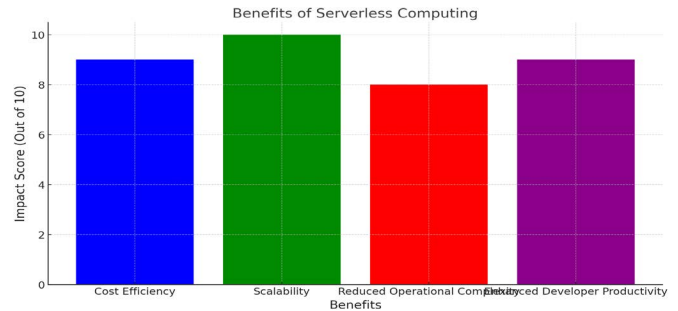


Figure 3: Benefits of serverless computing.
Source: Google Cloud Functions⁶

4. Limitations of Serverless Technology

One of the primary limitations of serverless technology is cold start latency. If a period is inactivity, it may experience a delay as the cloud provider initializes the execution environment. For instance, a web application with AWS lambda may experience delays when serving the first request after a period of inactivity, which could lead to suboptimal user experience during traffic times.

Serverless technology often leads to vendor lock-in, where applications become highly dependent on the specific services and APIs provided by a single provider. The dependency can make it challenging to migrate to another provider without the issue of refraction and potential architecture.

Serverless technology allows execution time limits for functions ranging from seconds to several minutes. The limitation can be restrictive for long-running processes or tasks that require extended execution times. For example, a financial services company, JP Morgan, adopted serverless technology, AWS Lamda, to run complicated checks on financial transactions after a period of inactivity and significant delays due to cold start latency. This was unacceptable because of the real-time processing required for immediate response³.

Here is a pie chart showing the limitations of Serverless computing.

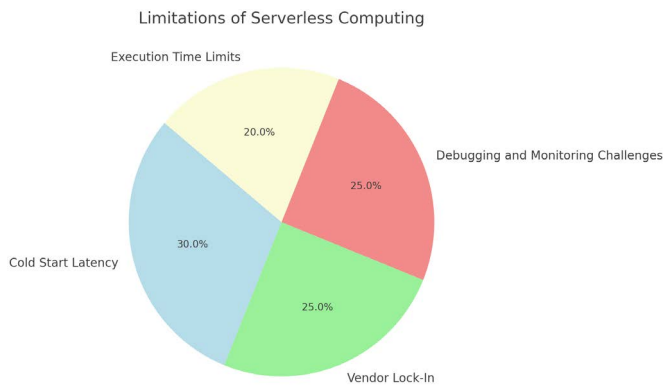


Figure 4: Limitation of serverless computing.

Source: Limitation of cloud computing and serverless technology⁷

5. Future Prospects and Emerging Trends

One of the areas to focus on in the future of serverless technology is reducing cold start times; cloud providers are continuously working to optimize the initialization process to minimize the latency for serverless functions. Technologies are waring containers using light-weight runtimes and adopting more efficient execution environment being developed at address challenge.

As businesses seek to avoid vendor lock and improve resilience, the adoption of multi-cloud servers and hybrid serverless solutions is on time. These solutions allow enterprises to deploy serverless functions across multiple cloud providers or on a premises environment, providing greater flexibility and reliability.

The future of serverless computing will see the introduction of more sophisticated developer tools and frameworks to simplify the development, deployment, and management of serverless applications. These tools will offer enhanced capabilities for debugging, monitoring, and security, making it easier for developers to build robust serverless solutions. These service frameworks and AWS are example tools of that help developers manage service applications more effectively⁸.

6. Bottom Line

Serverless computing has revolutionized the way applications are developed and deployed by abstracting the complexities of server management. Since the introduction of AWS Lambda in 2014, serverless technologies have seen significant adoption, driven by their cost efficiency, scalability, and ability to enhance developer productivity. As organizations increasingly demand agility, serverless architectures provide the flexibility to rapidly iterate and deploy applications while only paying for actual usage, leading to substantial cost savings.

Despite its benefits, serverless computing faces challenges such as cold start latency, vendor lock-in, and debugging difficulties. However, ongoing advancements aim to mitigate these issues. For example, improvements in cold start times, the emergence of multi-cloud and hybrid-cloud solutions, and enhanced developer tools are paving the way for more robust and efficient serverless applications.

Adoption across significant platforms like Google Cloud Functions and Azure Functions highlights the growing preference for serverless models. As Function-as-a-Service (FaaS) continues to evolve, it enables developers to deploy functions triggered by specific events, supported by auto-scaling capabilities that ensure efficient resource utilization.

The future of serverless computing is promising, with continuous innovations expected to address current limitations and expand its capabilities. This evolution will not only enhance the developer experience but also significantly impact various industries, enabling more efficient, scalable, and resilient applications. As serverless technology matures, it will undoubtedly play a crucial role in shaping the future of cloud computing and application deployment.

7. References

1. Barokah I, Asriyanik A. Analisis perbandingan serverless computing pada google cloud platform. *Jurnal Teknologi Informatika dan Komputer* 2021.
2. Brand L, Mock MU. SFL: A compiler for generating stateful aws lambda serverless applications. *Proceedings of the Seventh International Workshop on Serverless Computing (WoSC7) 2021*.
3. Choudhary B, Pophale C, Gutte A, Dani A, Sonawani S. Case study: Use of AWS lambda for building a serverless chat application. *Proceedings of the International conference on cloud computing and virtualization 2020*.
4. Fuerst A, Sharma P. FaasCache: Keeping serverless computing alive with greedy-dual caching. *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems 2021*.
5. Grzesik P, Mrozek D. Serverless nanopore basecalling with AWS Lambda. *International Conference on Conceptual Structures 2021*.
6. Jonas E, Schleier-Smith J, Sreekanti V, et al. Cloud programming simplified: A berkeley view on serverless computing. *arXiv 2019*.
7. Jia Z, Witche E. Nightcore: Efficient and scalable serverless computing for latency-sensitive, interactive microservices. *Proceedings of the 26th ACM International Conference on architectural support for programming languages and operating systems 2021*.
8. Kondratiuk T, Naumenko T. Comparison of using Apache OpenWhisk and Google Cloud Functions for Development of Serverless Applications on Google Cloud Platform. *Sys Res Inform Technol* 2021.