

Serverless Computing and Faas Model - The Next Stage in Cloud Computing

Gourav Bansal
Kurukshetra University

Abstract:- Serverless Computing is an implementation approach in which a cloud-based service provider manages the distribution of computing resources on the associated server on a real-time basis. It represents a progression of cloud programming paradigms, platforms, and abstractions, as well as proof of cloud technology maturity and widespread acceptance. And the understanding of the compute reference model, as well as the construction of facilities, has taken on new dimensions. Cloud computing has allowed businesses to focus more on their core products and services and less on their IT infrastructure. Furthermore, the cloud is no longer considered a viable option to hosting infrastructure. We additionally go over the pressing topics that serverless computing is encountering, as well as how future research could make it easier to deploy and use. This research article provides a thorough analysis of serverless computing architecture as well as more extensive testing of the serverless computing operating concept. In addition, this paper covers the considerable advantages, efficacy, cheap prices, and time, among other things. In addition, the study paper assesses some of the unique possibilities and potentials that serverless computing has to offer.

INTRODUCTION

Serverless computing is the next-generation cloud, which combines cloud computing with full application capabilities. You log into the (virtual) server in traditional cloud; in serverless cloud, you simply run the whole functional application programming interface (API). Businesses may concentrate completely on building value-added functions and features with this serverless configuration.

Rather than using a stream-based system, Serverless uses an event-based system. Each component of the program is independent under an event-based design. Events are triggered by each other. There are links to each service in stream-based. If something goes wrong, it only affects that one event, not the entire log.

Serverless delegates all management responsibilities for backend cloud infrastructure and operational chores to the cloud provider, including provisioning, scheduling, scaling, patching, and more. This frees up time for developers to work on and improve their front-end application code and business logic. Customers never pay for unused capacity with serverless. They just pay for the resources they need to operate their apps, and only when they're running.

SERVERLESS ARCHITECTURE EVOLUTION

The many aspects of Serverless computing can be portrayed as the stages of cloud computing from Infrastructure-as-a-Service (IaaS) to Platform-as-a-Service (PaaS) to Function-as-a-Service (FaaS) to Function-as-a-Service (FaaS) to Function-as-a-Service (FaaS) to Function-as-a-Service (FaaS). Because Infrastructure-as-a-Service (IaaS) separates the hidden infrastructure to provide virtual machines for immediate use and Platform-as-a-Service (PaaS) separates the middleware layer and the operating system involved, Infrastructure-as-a-Service (IaaS) separates the middleware layer and the operating system involved. While playing an important role in the process, Function-as-a-Service (FaaS) takes it a step further by abstracting the complete programming runtime to enable choices for quickly deploying and executing code without having to worry about the infrastructure in place.

The pay-as-you-go model, in which a client pays only for resources leased from the cloud provider and has the freedom to access as many resources as needed with no up-front fee, was a crucial reason in the cloud's rapid adoption by enterprise IT (elasticity).

CLOUD 2.0: SERVERLESS ARCHITECTURE

Organizations used to have to operate their own servers using physical resources only a few years ago. As a result, the cost of production, maintenance, and labor increases. However, new services and solutions were introduced as soon as the Cloud computing sector began to emerge. A cloud computing execution model known as serverless computing has recently gained popularity as a solution to the server dilemma. A new paradigm is represented by serverless architecture. Companies can use it to bypass the normal cost and effort barriers that often stymie IT transitions. It also allows businesses to convert their IT spending from a capex to an opex model for the duration of application development, which can save up to 60% on overall costs[8].

Managing some form of server infrastructure is frequently required when hosting a software application on the internet. In most cases, this entails managing a virtual or physical server, as well as the operating system and other web server hosting operations essential for your application to function. While using a virtual server from a cloud provider like Amazon or Microsoft eliminates the need for physical hardware, it still necessitates some level of operating system and web server software control [1]. Users prepurchase units of capacity in an Infrastructure-as-a-Service (IaaS) cloud computing paradigm, which means you pay a public cloud provider for always-on server components to operate your apps. It is the responsibility of the user to scale up server capacity during periods of high demand and scale down when that capacity is no longer required. Even when an app isn't being used, the cloud infrastructure required to support it is operational.

Serverless architecture, on the other hand, allows apps to be launched only when they are needed. The public cloud provider dynamically provides resources for app code that is triggered by an event. When the code has completed its execution, the user is no longer charged. Serverless frees developers from mundane and laborious activities associated with app scalability and server provisioning, in addition to the cost and efficiency benefits.

SERVERLESS COMPUTING FAAS MODEL

FaaS is an event-driven computing execution architecture in which developers create logic that is deployed in containers that are fully managed by a platform and then performed on demand. FaaS, in contrast to BaaS, gives developers more autonomy by allowing them to design unique apps rather than relying on a library of prewritten services.

FaaS is a serverless computing approach in which developers write business logic that is then run in Linux containers that are fully controlled by a platform. The paradigm is evolving to incorporate on-premise and hybrid installations, while it is typically a cloud computing platform leveraging cloud computing services. Developers can focus on creating code and providing business value by delegating infrastructure concerns like managing or provisioning servers and resource allocation to a platform (such as Red Hat OpenShift).

THE NEXT PHASE OF CLOUD COMPUTING

Perhaps the simplest approach to grasp the significance of serverless computing is to concentrate on the first of the main characteristics (as previously mentioned): providing an abstraction that hides servers and so simplifies the programming and operating model. Cloud computing promised a simpler operating paradigm from the start, but reduced programming comes from hiding servers. Efforts to offer abstractions that simplify cloud programming will steer the future direction of serverless computing, and in our opinion, cloud computing.

When compared to the influence cloud computing has had on operators, it's remarkable how little it's influenced how programmers work. Much of the software used in cloud computing is the same as that used in traditional data centers[4]. When you compare the most in-demand programming talents today to those needed ten years ago, you'll see that the basic skill set hasn't altered much, even as individual technologies come and go. Configuring and maintaining servers, storage, and networks is essentially obsolete, replaced by an emphasis on managing virtualized infrastructure via cloud provider APIs, as well as the DevOps movement, which stresses the technical and organizational components of change management.

Two application-specific serverless offerings are presented that adapt to this situation and demonstrate how the category allows for diverse methods. To conduct queries against data in object storage, one might utilize the AWS Athena big data query engine, a tool written in SQL (Structured Query Language). SQL is especially well suited to analytics because it allows you to express this computation in a single statement. Alternatively, one may utilize a framework like the one provided by Google Cloud Dataflow. To do so, write a basic MapReduce-style program in Java or Python, for example, with two functions: one that computes a weighted average for a chunk of data, and another that aggregates weighted averages for distinct chunks into one for their union. The framework handles data flow into and out of these services, as well as autoscaling, reliability, and other distributed system issues. This abstraction, unlike the SQL-based tool, can run any code, making it suited for a larger range of analytics tasks.

AWS SERVERLESS PLATFORM

AWS serverless computing services make it possible to develop and deploy applications on the AWS cloud without needing to manage servers. Vendors can use the AWS serverless platform to deploy cloud solutions without having to worry about server provisioning, deployment, maintenance, or monitoring. Customized setup, ease of maintenance, strong security, scalability, high availability, and monitoring are just a few of the characteristics. Some of the services provided are:

Amazon DynamoDB is a NoSQL database service, and Amazon Aurora Serverless (for Amazon Aurora) allows for automatic database start-up, shut-down, and capacity scaling up and down. Amazon RDS Proxy is a database proxy service offered by AWS.

Amazon SNS (pub or sub messaging service), Amazon SQS (message queuing service), AWS AppSync (to securely access, manipulate, and combine data), and Amazon EventBridge are all made possible by it to transfer application data from different sources to your AWS environment.

To cope with data analytics, AWS offers services like Amazon Kinesis (for streaming data on AWS) and Amazon Athena (an interactive query service to analyze data). Apart from the forementioned services, it also offers a comprehensive set of tools for the building of serverless applications[9]. Continuous integration and testing tools, deployment, monitoring, and diagnostics tools, frameworks, and IDE plugins are among the tools available.

CONCLUSION

By assigning to the platform, serverless computing alleviates many of the operational difficulties of monitoring and growing large-scale applications for developers. However, due to the stateless nature of their duties, the developer must currently find out how to map their application's service level agreements to those of the serverless platform as well as other reliant services. Despite the fact that numerous obstacles remain, the industry, scientific community, and open-source organizations have made rapid progress in providing the essential tools and programming models. It can be difficult to migrate legacy applications. However, if you're building a modern new app, you should think about serverless architecture. Serverless computing has a numerous advantage, notably rapid development, serverless deployment, lower operational costs, and easier management. However, this does not imply that there will be no operations. Although you don't need a system administrator to

handle the infrastructure, you should keep an eye on specific tasks to make sure your application is up and operating. But don't worry; FaaS offers monitoring data services to assist you in monitoring your apps. Finally, keep in mind that it may not be the best strategy for every problem, so be astute enough to choose intelligently. Prepare to take use of serverless technologies and focus on the development process rather than the provisioning of servers.

REFERENCES

- [1] Chrissy Kidd, Shanika Wickramasinghe (August, 2021) .Serverless vs Function-as-a-Service (FaaS): What's The Difference? <https://www.bmc.com/blogs/serverless-faas/>
- [2] SQLShack- An introduction to Serverless Applications- June 25, 2020 by Aveek Das<https://www.sqlshack.com/an-introduction-to-serverless-applications/>
- [3] Großmann M, Ioannidis C, Le DT (2019) Applicability of serverless computing in fog computing environments for iot scenarios In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC '19 Companion), 29–34.. Association for Computing Machinery, New York..
- [4] Armbrust, M., et al. A view of cloud computing. *Commun. ACM* 53, 4 (Apr. 2010).
- [5] Castro, P., et al. The rise of serverless computing. *Commun. ACM* 66, 12 (Dec. 2019).
- [6] Yussupov V, Breitenbücher U, Leymann F, Wurster M (2019) A systematic mapping study on engineering function-as-a-service platforms and tools In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing (UCC'19), 229–240.. Association for Computing Machinery, New York.
- [7] Introduction to Serverless Architecture in Cloud-based Applications- IhorFeoktistov CTO at Relevant <https://relevant.software/blog/serverless-architecture/>
- [8] Barga, R.S. Serverless computing: Redefining the cloud [Internet]. In Proceedings of the 1st Intern. Workshop on Serverless Computing (Atlanta, GA, USA, June 5, 2017)
- [9] Community Blog- Serverless vs. Traditional Architecture: What Are the Differences? Alibaba Clouder March 14, 2019https://www.alibabacloud.com/blog/serverless-vs--traditional-architecture-what-are-the-differences_594578
- [10] SQLShack- An introduction to Serverless Applications- June 25, 2020 by Aveek Das<https://www.sqlshack.com/an-introduction-to-serverless-applications/>