Serverless Computing: Architecture and Applications

A Project Work Synopsis

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE WITH SPECIALIZATION IN DEVOPS

Submitted by:

22BDO10052 - Sushant Jha

22BDO10048 - Sneha Mehrotra

22BDO10049 - Agamjyot Singh Maan

22BDO10007 - Atinshay Awasthi

Under the Supervision of:

Dr. Mandeep Singh

Associate Professor

Department of Computer Science and Program(AIT)

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

Abstract

The research paper explores serverless computing, where server management is delegated to cloud providers, liberating developers from the task. The focus is on architectural paradigms like Function as a Service (FaaS) and Backend as a Service (BaaS), conducting a comparative analysis of use cases and benefits.

Advantages such as reduced server management complexity, cost-effectiveness, and auto-scaling are discussed. Challenges, including security responsibilities and drawbacks, are addressed. Major cloud providers like AWS Lambda, Google Cloud Functions (GCF), and Azure Functions are analyzed. mk9

The paper details FaaS implementation, emphasizing event-driven nature and application response to triggers. Security aspects, with code security entrusted to developers, are covered. Real-world use cases of serverless computing are outlined.

The problem statement acknowledges the promises of serverless computing but highlights challenges such as security, cold start times, vendor lock-in, and state management difficulties. The research aims to comprehensively explore these issues, proposing solutions and best practices for effective deployment and management.

Table of Contents

1. INTRODUCTION

1.1 Problem Definition

Traditional server-based architectures entail cumbersome and costly server provisioning and management, lacking efficiency, scalability, and cost-effectiveness. In contrast, serverless computing streamlines this process by abstracting infrastructure management. Developers focus solely on writing code without dealing with servers. Serverless platforms dynamically allocate resources in response to events, optimizing scalability, and offer a pay-per-use pricing model, reducing costs. By simplifying infrastructure management and accelerating development, serverless computing aims to deliver efficient, scalable, and cost-effective code execution, empowering organizations to innovate more rapidly and compete effectively in today's digital landscape[5].

1.2 Problem Overview

Our project will delve into key aspects of serverless computing

- **Event-Driven Model:** Examining how serverless functions react to events like HTTP requests, database updates, or file uploads.
- **Scalability:** Harnessing auto-scaling features of serverless platforms to adapt resources dynamically based on workload.
- **Cost Optimization:** Assessing cost advantages vis-a-vis traditional server setups, considering pay-per-use pricing and resource efficiency.
- **Use Cases:** Investigating real-world scenarios where serverless architecture thrives, such as microservices, real-time data processing, and IoT applications. Through these analyses, we aim to showcase the agility, scalability, and cost-effectiveness of serverless computing in diverse applications.

1.3 Software Specification

To engage in serverless development, we require specific software elements of Serverless Frameworks: Software tools such as Serverless Framework, AWS SAM, or Azure Functions Core Tools.

- **Programming Languages:** Widely utilized languages encompass JavaScript/Node.js, Python, Go, and Java.
- **Cloud Provider Account:** Entry to a cloud provider like AWS, Azure, or Google Cloud.
- **Event Sources:** These comprise databases, APIs, storage services, and similar entities that activate our serverless functions.

2. LITERATURE SURVEY

Serverless computing is a paradigm that abstracts infrastructure management, allowing developers to focus solely on writing code without dealing with servers. It offers dynamic resource allocation and a pay-per-use pricing model, optimizing scalability and reducing costs [1] [2]. Traditional server-based architectures lack efficiency, scalability, and cost-effectiveness due to cumbersome server provisioning and management. In contrast, serverless computing simplifies infrastructure management and accelerates development, aiming to deliver efficient, scalable, and cost-effective code execution [2]. By integrating Terraform and serverless computing, organizations can construct a scalable and efficient cloud infrastructure. Terraform, an infrastructure-as-code tool, combined with serverless computing, enables the creation of a dynamic and robust infrastructure while addressing the difficulties that arise in constructing a serverless computing infrastructure [3]. This integration can help organizations create an efficient, adaptable, and scalable infrastructure while decreasing expenses and enhancing developer productivity [4]. Overall, serverless computing empowers organizations to innovate more rapidly and compete effectively in today's digital landscape .

2.1 Existing System

Legacy systems: Old technologies, techniques, hardware, and software systems that are still in use can be accessed and migrated to serverless cloud computing.

Lift and shift approach: This technique transfers the entire infrastructure of a legacy system to the cloud, including the monolithic applications, to be deployed as serverless cloud functions.

Lambda: A tool proposed in a research paper that automatically refactors, tests, and deploys monolithic applications (Java) into microservices (AWS Lambda Node.js) .

Software platforms: These platforms are implemented to deal with resources from multiple cloud providers and ensure the proper running of client applications in a heterogeneous cloud infrastructure.

2.2 Proposed System

A model for highly parallel file processing applications based on serverless architecture is proposed, which provides users with different ways to process their files, such as using the API gateway to submit files for processing .

An electrical overload warning system is implemented in the smart grid, based on serverless architecture. It utilizes Amazon web services, including S3, lambda functions, simple notification service (SNS), and CloudWatch. S3 is used as a storage service, lambda functions execute the code, CloudWatch monitors resources, and SNS handles notifications.

3. PROBLEM FORMULATION

Lack of Comprehensive Understanding:

Issue: Even though more and more people are using serverless computing, we haven't fully looked into how it works, what makes it tick, and how we can use it in real-life situations.

Rationale: We really need to take a close look at serverless computing to fill in the gaps in our knowledge about how it's built and how we can practically use it. This is super important for companies that want to make the most out of serverless solutions.

Security Challenges in Serverless Computing:

Issue: When we use serverless computing, there are big challenges in keeping things secure, like making sure data is private and controlling who can access it.

Rationale: Because serverless is kind of special, it brings its own security issues that we need to be very careful about. Tackling these problems is a must to make sure our data stays safe and confidential in serverless applications.

Practical Implications and Best Practices:

Issue: Right now, there aren't clear rules or best ways to make and use serverless apps.

Rationale: Finding out the practical things that work and don't work when making serverless apps is super important. This knowledge can really help developers make better decisions and be successful when building and using serverless applications.

Limited Exploration of Industry-Specific Applications:

Issue: Not enough research has been done on how different industries can use serverless computing in their own special ways.

Rationale: Every industry might have special needs and problems when they use serverless solutions. Looking into these specific cases is super important to give useful advice and insights tailored to each industry.

Addressing the Cold Start Problem:

Issue: There's a common issue in serverless computing called "cold starts," and it makes apps slow to respond.

Rationale: Solving the problem of "cold starts" is really important to make sure our apps respond quickly and work well. We need to figure out smart strategies to deal with this issue and make sure our systems are reliable.

Economic Considerations and Cost Optimization:

Issue: We don't know enough about how money works when using serverless computing and the best ways to save costs.

Rationale: Figuring out the money side of serverless, like pay-as-you-go models and ways to save money, is a big deal for companies adopting serverless tech. Understanding these money matters helps keep things financially stable and helps in making smart decisions[6].

4. RESEARCH OBJECTIVES

The paper aims to research into various facets of serverless computing, including its benefits, challenges, and real-world applications. It seeks to explore the event-driven model, scalability, and cost optimization inherent in serverless architectures, shedding light on their potential for efficient, scalable, and cost-effective code execution. Furthermore, the paper aims to define the distinctions between traditional server-based architectures and serverless computing, emphasizing the latter's capacity to simplify development processes and minimize operational burdens. By investigating these aspects, the paper endeavors to provide comprehensive insights into the advantages and considerations of adopting serverless computing, ultimately contributing to a better understanding of its role and significance in contemporary application development paradigms.

5. METHODOLOGY

In our research paper, we delve into the world of serverless computing. This is a model where the task of managing servers is handled by cloud service providers, freeing developers from this responsibility.Our focus is on understanding the architectural paradigms of serverless computing, particularly Function as a Service (FaaS) and Backend as a Service (BaaS). We conduct a comparative analysis of these paradigms, considering their respective use cases and benefits.

We discuss the advantages of serverless computing, such as reduced server management complexity, cost-effectiveness due to its pay-as-you-go model, and auto-scaling. However, we also address the challenges and drawbacks associated with it. We include an analysis of the offerings of major cloud providers like AWS Lambda, Google Cloud Functions (GCF), and Azure Functions.

In the implementation section, we explore how developers write their application code as a set of discrete functions in FaaS, each performing a specific task when triggered by an

event. We explain the event-driven nature of serverless architectures and how applications respond to certain events or triggers[3].

While the security of the physical servers is shifted onto the cloud provider, the responsibility for securing the code still lies with the developers. We discuss this aspect of security in serverless computing. We also provide an overview of real-world use cases for a serverless computing architecture.

The problem statement for our research paper on "Serverless Computing: Architecture and Applications" is: "Despite the promise of serverless computing to significantly reduce the cost and complexity of deploying and managing applications, there are still many challenges and uncertainties associated with its architecture and applications. These include security concerns, cold start times, vendor lock-in, and the difficulty of managing state in a stateless environment. Our research aims to explore these issues in depth, providing a comprehensive understanding of serverless computing's architecture and applications, and proposing potential solutions and best practices for overcoming these challenges."

This methodology provides a structured approach to exploring serverless computing in depth. It allows us to understand the benefits and challenges of serverless computing and propose solutions to overcome these challenges. It also provides valuable insights into the

7.CONCLUSION

In conclusion, our team's research paper explores serverless computing. This is a way of running programs where the people who write the programs don't have to worry about the servers that run them. That's all taken care of by companies that provide cloud services. We're looking at two main types of serverless computing: Function as a Service (FaaS) and Backend as a Service (BaaS), and comparing them.

We talk about the good things about serverless computing, like how it's easier because you don't have to manage servers, it can save money because you only pay for what you use, and it can handle lots of work easily. But we also talk about the problems, like security risks, delays in starting up, being stuck with one provider, and difficulties in managing data. We look at what big cloud providers like AWS Lambda, Google Cloud Functions (GCF), and Azure Functions offer.

We'll also look at how it's used in the real world, and what developers need to think about when they use it. Our goal is to understand serverless computing better and suggest ways to deal with its challenges.

In the end, we hope that our research will help people understand more about serverless computing, its benefits, and its challenges. We believe that this knowledge can help developers make better decisions when they're deciding how to run their programs.

8.REFERENCES

[1] Software Architecture Design of a Serverless System-Muhammad Hamza-14 Jun 2023 https://link.springer.com/chapter/10.1007/978-3-031-29315-3_2

[2] A comparison between traditional and Serverless technologies in a microservices setting - Jose Emilio Labra Gayo (University of Oviedo) -May2023 https://www.researchgate.net/publication/370981938_A_comparison_between_traditional_and_Serverless_technologies_in_a_microservices_setting

[3] Evaluating Serverless Computing - Charitra Maharjan - 05 Jun 2023 https://www.techtarget.com/searchcloudcomputing/tip/Evaluate-serverless-computing-best-practices

[4] Hassan et al. Journal of Cloud Computing: Advances, Systems and Applications (2021) 10:39 https://doi.org/10.1186/s13677-021-00253-7

[5]geeks for geeks serverless computing overview and details https://www.geeksforgeeks.org/serverless-computing

[6] Serverless Computing: Introduction and Research Challenges by Karim Djemame https://link.springer.com/chapter/10.1007/978-3-031-29315-3_2