International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Impact of Function as a Service on Application Development Efficiency in Cloud Environments

[1]Shantanu Kumar, [2]Harshavardhan Nerella, [3]Shruti Singh

Submitted: 07/02/2024 **Revised**: 16/03/2024 **Accepted**: 24/03/2024

Abstract: Function as a Service (FaaS) has transformed application development in cloud environments by simplifying the deployment process and letting developers concentrate exclusively on coding, barring the need to worry about infrastructure administration. By utilizing FaaS, developers have the option to decompose their applications into smaller, autonomous functions that are executed in response to events, leading to enhanced efficiency and scalability. The server less computing architecture obviates the necessity of providing and managing servers, enabling developers to expedite the delivery of applications in a more cost-efficient manner. The main aim of this study is to analyse the influence of FaaS on the efficiency of application development in cloud environments. The study employs a systematic process for reviewing the literature. This study performed a thorough review of 28 papers that were published from 2018 to 2024.This systematic literature analysis attempts to thoroughly assess the impact of FaaS, a famous model of server less computing, on the productivity of application development in cloud computing settings. The purpose of the review is to analyse and consolidate current research findings on the implementation and usage of FaaS. It aims to assess the impact of FaaS on several aspects such as the development process, performance metrics, operational requirements, scalability, reliability, and cost-effectiveness. This study demonstrates that FaaS greatly improves application development speed in cloud environments by providing a scalable, event-driven architecture, decreasing the burden of managing infrastructure, and enabling quick deployment of code snippets.

Keywords: *FaaS; Application development; Cloud Environment; Efficiency.*

Introduction:

FaaS is a recently developed paradigm that simplifies and streamlines the process of developing cloud-based applications. Within the FaaS paradigm, the primary emphasis of an application developer is to write code and create new features, without the need to be concerned about managing infrastructure. This responsibility is delegated to the cloud provider. FaaS, was initially pioneered by Amazon in 2014 with the introduction of AWS Lambda [1], [2]. Since then, several major cloud providers have followed suit and developed their own server less systems. Notable examples include Google Cloud Function (GCF) [1] from Google, Azure Function from Microsoft, along with IBM Cloud Function from IBM. Additionally, there are numerous open-source initiatives such as Apache Open Whisk, alongside Knative, alongside OpenLambda, alongside Fission, and others [1] [3].

[1]Senior Software Engineer, Amazon, US
Email ID: reach.shantanuk@gmail.com
[2]Senior Cloud Engineer, MassMutual, US
Email ID: nerellaharshavardhan@outlook.com
[3]Program Manager, Washington State University, US
Email ID: sshruti.connect@gmail.com

Early Internet applications relied on dedicated hardware as servers, requiring costly maintenance and resulting in underutilization of resources [4]. Additionally, adding/removing physical resources to scale to demand and troubleshooting an application were difficult. Underutilization of resources and increasing maintenance costs prompted the development of virtualization and container-based technologies [5]. These methods improved resource use and simplified application development, deployment, and management. Numerous tools [6] aid in resource management and application management. While virtualization and container-based approaches improve resource utilization and ease of application development, developers must manage and scale the underlying infrastructure, such as virtual machines or containers, despite reactive or predictive scaling approaches [7]. To simplify infrastructure management alongside application scaling, server less computing became a new paradigm for cloud application development, deployment, and management [8]. The server less computing approach enables developers to concentrate on writing code in a high-level language and creating new features, while the FaaS platform handles server configuration, management, and maintenance.

Scope of the study:

The investigations look to gauge the impact of FaaS on the architecture of cloud-based applications, with a specific focus on the design, deployment, and maintenance stages. This study will analyse actual data on the impact of FaaS on key performance characteristics, including deployment speed, response times, and scalability, in the presence of varying workloads. In addition, this study also involves assessing the financial consequences of using FaaS, including comparing it to established cloud models such as IaaS and PaaS, and analysing its cost-effectiveness for enterprises. Furthermore, the study examines the extent to which the adoption of FaaS differs among various businesses, as well as the distinct obstacles and advantages faced. This systematic literature review (SLR) aims to analyse and synthesize existing knowledge while identifying any existing gaps. The findings of this SLR will offer significant insights to software developers, IT managers, and policymakers regarding the effective utilisation of FaaS to boost the efficiency of cloud-based applications. In addition, it will suggest avenues for future study to tackle unresolved difficulties and improve comprehension of FaaS's capabilities in cloud computing.

Objectives of this study:

To examine how FaaS affects the architecture of cloud-based applications, focusing on the design, deployment, and maintenance phases.

To analyse the impact of FaaS on performance measures including deployment speed, response times, and scalability during varying workloads.

To assess the cost implications of FaaS adoption, comparing it to other cloud models (IaaS, PaaS) and analysing cost-efficiency for enterprises.

To explore FaaS adoption across industries and the various problems and benefits faced.

Research Questions:

How does FaaS influence the overall application development lifecycle in cloud environments?

What key performance metrics are affected by implementing FaaS in cloud-based application development?

How do FaaS platforms compare development efficiency and operational overhead to traditional cloud service models (like IaaS and PaaS)?

What challenges and limitations are developers reporting when integrating FaaS into their application development processes in cloud environments?

How does FaaS affect the scalability and reliability of applications developed in cloud environments?

What cost implications does FaaS have on the budgeting and financial management of cloud-based application projects?

Are there any sector-specific impacts of FaaS on application development efficiency in finance, healthcare, or e-commerce industries?

What future developments in FaaS technology could further impact its efficiency and adoption in cloud application development?

Background Of The Study:

The increasing usage of cloud computing has completely transformed the field of application development, providing unparalleled scalability, flexibility, and cost-efficiency. FaaS has emerged as a transformative force inside this framework, significantly reshaping the approach that developers take in designing and implementing applications [9]. FaaS lets developers concentrate exclusively on creating and running individual pieces of code, known as functions, without worrying about the fundamental infrastructure. This paradigm shift obviates the necessity of provisioning and overseeing servers, enabling developers to exploit event-driven architectures and adjust resource allocation dynamically in response to demand. FaaS has the potential to greatly improve the efficiency of application development in cloud environments [10]. It achieves this by simplifying the development process, dropping the time required to bring a product to market, and maximizing the use of resources. Although the advantages of FaaS are clear, it is necessary to further investigate its effects on other aspects of application development, including security, performance, and vendor lock-in, in order to properly comprehend its influence in various organizational contexts [11].

Methodology:

The Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology will be utilised to conduct a systematic literature review on the influence of FaaS on application development efficiency in cloud environments. The methodology entails the formulation of research inquiries, the development of a thorough search strategy, the selection of pertinent studies based on specific criteria for inclusion alongside exclusion, data extraction from the chosen studies, the evaluation of study quality, the synthesis of data, and the presentation of results in accordance with PRISMA guidelines. This systematic methodology guarantees clarity, precision, and dependability in the process of locating, assessing, and combining existing literature. Ultimately, it offers significant insights into the effect of FaaS on the productivity of application development in cloud settings.

Table 1: Relevant publications were identified from internet repositories in this study.

DIGITAL LIBRARY	URL
Scopus	https://www.scopus.com/sources.uri?zone=TopNavBar&origin=searchbasic
Semantic Scholar	https://www.semanticscholar.org/
IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp
Science Direct	https://www.sciencedirect.com/
Springer	https://link.springer.com/
Web of Science	https://wosjournal.com/
PubMed	https://pubmed.ncbi.nlm.nih.gov/

To conduct a systematic literature review on the influence of FaaS on application development efficiency in cloud environments, this study used a well-designed search strategy to locate relevant literature. This method involved using search strings that consist of appropriate keywords and Boolean operators. Search strings consist of combinations such as *"Function as a Service"* OR "FaaS"* AND "cloud computing"* AND "application development efficiency"*, OR "server less computing"* AND "development productivity"* OR "time-to-market"*.* The search phrases customized to the specific databases and repositories being used, guaranteeing a thorough coverage of relevant material. In addition, synonyms and variations of keywords included to encompass a wide range of perspectives and approaches found in the literature.

Inclusion/Exclusion Criteria:

The following table presents the potential criteria for inclusion and exclusion in this systematic literature review on the investigation of performance and scalability in cloud object stores:

Table 2: Inclusion/Exclusion Criteria

Criteria	Inclusion	Exclusion
Type of Study	Academic Research papers and Review articles	Editorials, articles and opinions
Topic Relevance	Studies that investigate the impact of FaaS on application development efficiency in cloud environments.	Studies unrelated to FaaS, cloud computing, or application development efficiency.
Publication date	Studies published between 2018 - 2024	Studies beyond and after 2018-2024
Language	English	Non-English studies
Subject Area	Faas; Cloud Computing; application development, efficiency, server less computing, development productivity.	Irrelevant Subjects
Access Availability	Open access studies	Studies behind paywalls or lacking access
Peer – Review	Peer-reviewed studies	Non-peer-reviewed studies

The data was obtained by formulating and evaluating the search query in several reputable databases. Once the references had been loaded onto Endnote, they underwent a filtering process to remove any cases of duplication. As a result, a total of 120 papers were retained, all of which were entirely distinct from one another. Through a thorough examination of abstracts and titles, our study found publications that included the selected keywords. After that, the created references were moved to an Excel spreadsheet for further filtering alongside analysis. This spreadsheet included the names of the writers, the year the work was published, the title, alongside the abstract. The Excel spreadsheet has 81 research publications in total. All the paper abstracts were reviewed using Excel, with a focus on those that were directly connected to the objectives of the study. At first, 52 pieces in total were found; however, some of them had to be eliminated since they were classified as grey literature, books, or parts of books. Moreover, as some of the papers could not be downloaded, they were removed from distribution. The 28 papers that made up the final shortlist. This flow diagram (Figure below) illustrates the process of selecting articles for the SLR during a literature search and is based on the PRISMA Flow Diagram.

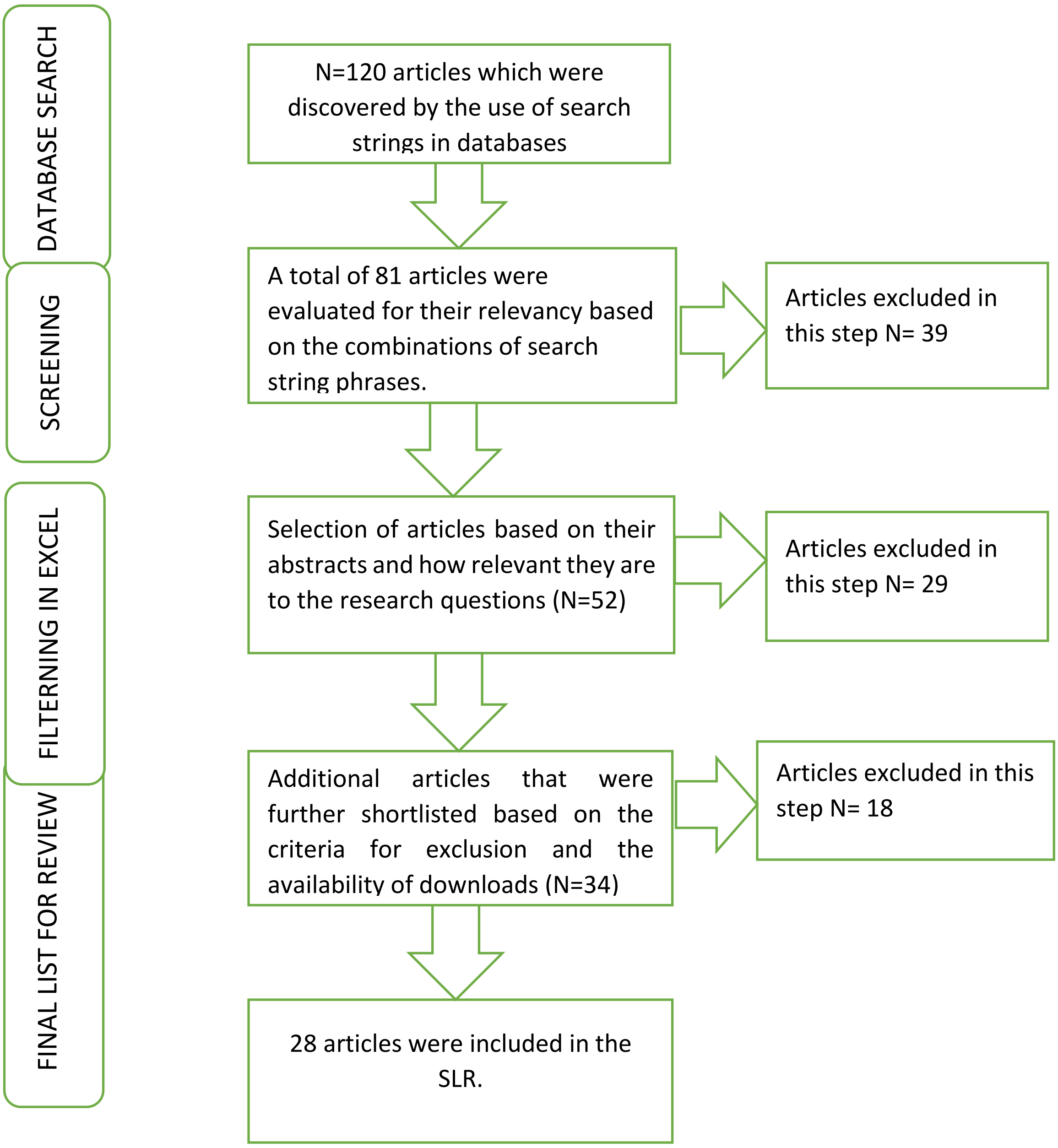

Fig 1: Literature search for SLR publications (based on PRISMA flow diagram).

Results And Discussions:

The introduction of FaaS has significantly increased the efficiency of application development in cloud settings. By removing the complexity of infrastructure administration, FaaS simplifies the development process and frees developers to concentrate only on developing code. This paradigm change improves efficiency in several ways [12]. To start, FaaS speeds up the time to market for new features and updates by enabling quick cycles of iteration and deployment. Second, it provides unmatched scalability, enabling applications to adjust resources automatically in response to demand, improving resource efficiency and cutting expenses. Third, FaaS promotes agility by letting developers react fast to shifting market conditions and requirements. Additionally, maintenance is made simpler and more efficient by shifting operational responsibilities to cloud providers, which frees up important time and resources. All things considered, FaaS has a significant impact on the effectiveness of application development in cloud settings, enabling businesses to innovate more quickly, provide better user experiences, and maintain their competitiveness in the quickly changing digital market [13].

The impact of FaaS on the entire application development lifecycle in cloud environments:

Integrating FaaS has a significant impact on many phases of the application development lifecycle. During the planning phase, FaaS has an impact on architectural decisions as developers create applications with a modular and event-driven approach to take use of the advantages of server less computing [`14]. During the development stage, FaaS expedites coding processes by eliminating the need to worry about infrastructure, enabling developers to concentrate exclusively on developing code for specific services. This modular architecture allows for the implementation of agile development approaches, which in turn enables the quick and efficient execution of iterative processes and improvements. When conducting tests, FaaS allows for more detailed and targeted testing of services, which enhances comprehensive assessment and guarantees dependability [15]. During deployment, FaaS simplifies the process by offering automatic scalability and smooth deployment of functions, resulting in reduced deployment durations and minimized periods of inactivity. In addition, FaaS streamlines maintenance by automating operational activities, such as server provisioning and scalability, allowing developers to focus on improving application functionality. FaaS fundamentally transforms the process of developing applications in cloud environments by promoting agility, scalability, and efficiency throughout all stages of planning, development, testing, deployment, and maintenance [16].

Key performance criteria for FaaS implementation in cloud-based application development:

Implementing FaaS into cloud-based application development impacts various important performance indicators, each providing valuable information about the efficiency and efficacy of the deployment [17]. They are:

Deployment Speed: FaaS greatly enhances the speed of deployment by automating the process of setting up and configuring the infrastructure. This metric quantifies the duration from the moment code is committed and the subsequent deployment of the application. Using FaaS, the time it takes to deploy can be measured in minutes or even seconds, as opposed to traditional approaches that may require hours or days.

Response Time: FaaS directly affects response time, which is a metric that quantifies the delay between a user's request and the matching response from the application. FaaS designs can optimize response times and provide fast and consistent user experiences by dynamically scaling resources and distributing functions across several instances.

Uptime: FaaS improves uptime, which refers to the proportion of time that an application is accessible and functioning. FaaS solutions utilize autonomous scaling and fault-tolerant design to ensure consistent uptime, reducing the likelihood of service disruptions and downtime events [18].

Scalability: The ability of an application to manage increased workloads without performance loss is called scalability. FaaS architectures allow applications to dynamically scale resources to meet demand spikes. Concurrent requests, throughput, and capacity usage are scalability indicators.

Resource Utilization: FaaS optimises computer resource allocation based on demand. CPU, memory, and storage efficiency is measured by this statistic. By scaling resources as needed, FaaS platforms reduce resource waste and ensure cost-effective operations [19].

Comparison of server less computing (FaaS) and other cloud computing models:

When comparing FaaS platforms with classic cloud service models like Infrastructure as a Service (IaaS) alongside Platform as a Service (PaaS), it is pivotal to assess development efficiency and operational overhead across many aspects [20], as detailed in the table below:

Table 3: Comparison of FaaS and other models [27] [28].

	FaaS	IaaS	PaaS
Deployment Efficiency	FaaS simplifies infrastructure administration so developers can focus on function code. Modular development speeds up agile methods and application time-to-market.	IaaS gives developers better control over virtual machines, storage, and networking. This flexibility allows customization, but developers must manage and maintain the infrastructure, which may impede development.	PaaS simplifies development by providing pre-configured platforms and middleware for app creation, deployment, and management. PaaS abstracts some infrastructure issues but may limit development flexibility.
Operational Overhead	Server management, scaling, and maintenance are automated by FaaS, reducing operational overhead. Developers can focus on application logic and functionality without infrastructure provisioning and management.	Developers must manage virtual computers, storage, networking, and security in IaaS, increasing operational overhead. IaaS provides freedom and control, but it also requires infrastructure monitoring, scaling, and troubleshooting.	PaaS technologies abstract infrastructure management responsibilities more than IaaS, decreasing developer overhead. Developers may still need to configure and optimize applications, depending on the platform.

Limitations and challenges that developers have when incorporating FaaS into their cloud-based application development processes:

Developers who are incorporating FaaS into their application development procedures in cloud environments face several obstacles and restrictions. An important concern is the intricacy of migrating from conventional application architectures to server less paradigms based on FaaS. This transition necessitates developers to reconsider the design of systems, dismantling monolithic architectures into smaller, more detailed services. This can be a challenging undertaking, particularly for older applications [21]. In addition, developers frequently encounter difficulties related to vendor lock-in, as each FaaS provider possesses its own exclusive platform and ecosystem, hence complicating the process of migrating applications across providers or returning them to on-premises environments. In addition, the process of identifying and fixing errors and monitoring distributed server less applications is difficult because the functions are temporary and there is restricted access to the underlying infrastructure. To address these obstacles, developers may have to allocate resources towards acquiring extensive tools for debugging, monitoring, and performance enhancement. Additionally, they should embrace established methodologies for function design and deployment. Cooperative endeavours among developers and improvements in cross-platform compatibility and interoperability standards can reduce worries about being tied to a specific vendor and encourage wider use of FaaS in cloud environments [22].

The impact of FaaS on cloud application scalability and reliability:

FaaS significantly influences the ability of applications designed in cloud environments to scale and maintain stability.

Scalability: FaaS allows applications to automatically adjust their capacity in real-time based on changes in workload demands. FaaS enables the invocation of functions based on certain events or even triggers, like HTTP requests, alongside database modifications, or even scheduled events. This event-driven design enables the autonomous allocation and scaling of resources in response to demand. As the volume of incoming requests rises, FaaS systems deploy extra instances of function containers to manage the workload, guaranteeing that the application can expand smoothly to address sudden increases in traffic. The elastic scalability of FaaS platforms allows applications to efficiently handle unpredictable or variable workloads, surpassing the capabilities of traditional infrastructure models [23].

Reliability: FaaS improves the dependability and accessibility of applications by utilizing the foundational architecture offered by cloud providers. FaaS platforms handle the tasks of server provisioning, scaling, and fault tolerance in a way that is not visible to developers, simplifying the management of infrastructure. FaaS providers utilize redundant architectures and high availability zones to guarantee resilience in the face of hardware failures or network outages. In addition, FaaS platforms include integrated functionalities like automated retries, error management, and distributed execution environments, enhancing the overall dependability of applications. Functions are characterized by their statelessness and ephemerality, which allows them to be immediately resumed or relocated to new execution contexts in the event of errors. This minimizes downtime and guarantees uninterrupted availability of services [24].

Cost-effectiveness of using FaaS:

FaaS has many cost implications that have a substantial impact on the planning and financial management of cloud-based application projects:

Cost per Request: FaaS platforms generally employ a pricing strategy that charges users depending on the quantity of function invocations or requests processed, using a pay-per-use approach. Organizations can use this granular pricing model to ensure that prices are precisely proportional to consumption, meaning they only pay for the specific resources that their apps consume. Unlike traditional server-based configurations, which require enterprises to pay for supplied resources regardless of how much they are actually used, FaaS provides cost savings by reducing over-provisioning and maximizing resource consumption.

Total cost of Ownership (TCO): When assessing the cost-effectiveness of utilizing FaaS, it is crucial to take into account the comprehensive cost of ownership for the complete lifespan of an application. Although FaaS may provide cost advantages in terms of initial investment compared to establishing and managing conventional server-based infrastructure, firms must also consider additional aspects such as expenses related to data transport, storage, and the potential consequences of being tied to a specific vendor. In addition, enterprises should include the expenses related to the creation, testing, monitoring, and upkeep of server less applications when determining the TCO [23].

Potential Saving: For applications with uncertain workloads, FaaS can save a lot compared to server-based systems. FaaS platforms optimize costs and efficiency by automatically scaling resources based on demand, eliminating over-provisioning and idle resources. FaaS solutions generally include automatic scaling, load balancing, and pay-per-use pricing, which cut operating expenses and improve cost predictability.

Vendor Lock - in: While FaaS platforms offer cost savings and flexibility, enterprises should evaluate vendor lock-in risks. Migrating apps to other platforms or on-premises settings may cost more and be more complicated, depending on the FaaS provider. Thus, firms should consider how vendor lock-in would affect their budgeting and financial management practices over time [23].

Impacts of FaaS on application development efficiency in finance, healthcare, or e-commerce industries:

The effects of FaaS on the efficiency of application development can range among various industries, including finance, healthcare, and e-commerce, due to distinct regulatory, security, and data handling prerequisites. Within the finance sector, where strict adherence to regulations and protection of data are of utmost importance, Function-as-a-Service (FaaS) can provide notable advantages by offering a safe and adaptable platform for the creation and implementation of financial applications. Nevertheless, compliance with legal standards like as PCI DSS (Payment Card Industry Data Security Standard) or even GDPR (General Data Protection Regulation) may necessitate further deliberations when using FaaS. Within the healthcare industry, where it is crucial to handle sensitive patient information with extreme caution and adhere to standards like HIPAA, FaaS can optimize the creation of applications while guaranteeing the security and privacy of data [25]. However, developers are needed to incorporate suitable security mechanisms and encryption techniques to protect patient information in FaaS systems. In the e-commerce sector, where maintaining competitiveness requires quick innovation and scalability, Function-as-a-Service (FaaS) can facilitate agile development methods and optimize resource allocation, resulting in expedited time-to-market for new features and improvements. Nevertheless, it is imperative to prioritize the safeguarding of consumer payment data and defending against any cyber risks while adopting FaaS. In general, whereas FaaS provides significant advantages for improving application development efficiency in many industries, businesses need to carefully evaluate sector-specific regulatory restrictions and security considerations to effectively leverage the benefits of FaaS while minimizing potential hazards [26].

Future developments in FaaS technology:

The potential for future advancements in FaaS technology to enhance its efficiency and increase its usage in cloud application development is substantial. An area of research that shows promise is the optimization of cold start times. This pertains to the latency that arises when a function is invoked for the first time or following a period of inactivity. Scientists are investigating methods like pre-warming function instances, optimizing container start-up, and utilizing new programming languages and runtime environments to reduce cold start latency. This will enhance application responsiveness and improve the user experience. In addition, the progress in server less orchestration and composition tools are improving the capacity to construct and oversee intricate, multi-function applications with greater efficiency. These technologies empower developers to establish workflows, dependencies, and event-driven structures in a more intuitive and efficient way, promoting increased productivity and scalability. In addition, the incorporation of machine learning and artificial intelligence functionalities into FaaS platforms is creating opportunities for intelligent automation, auto-scaling, and predictive resource allocation. This enhances efficiency and resource utilization in the development of cloud applications. With the ongoing progress of these and other breakthroughs, FaaS technology is positioned to have a growing and crucial impact on fostering innovation and flexibility in the creation of cloud-based applications.

Conclusion:

In summary, FaaS has transformed the way applications are developed in cloud environments by simplifying procedures, improving scalability, and facilitating quick deployment cycles. The event-driven design, automatic scaling, and decreased operational overhead greatly enhance efficiency, enabling faster time-to-market and empowering developers to prioritize innovation.

References:

[1] A. Raza, I. Matta, N. Akhtar, V. Kalavri, and V. Isahagian, “SoK: Function-as-A-service: From an application developer’s perspective,” Journal of Systems Research, vol. 1, no. 1, 2021.

[2] N. Ponomarev, A Proof of Concept for an AWS Lambda malicious code generator tool. 2022

[3] A. Palade, A. Kazmi, and S. Clarke, “An evaluation of open source serverless computing frameworks support at the edge,” in 2019 IEEE World Congress on Services (SERVICES), 2019.

[4] P. Seda, P. Masek, J. Sedova, M. Seda, J. Krejci, and J. Hosek, “Efficient Architecture Design for Software as a Service in Cloud Environments,” in 2018 10th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT), 2018.

[5] M. Shahrad, J. Balkind, and D. Wentzlaff, “Architectural implications of function-as-a-service computing,” in Proceedings of the 52nd

Annual IEEE/ACM International Symposium on Microarchitecture, 2019.
[6] P. Leitner, E. Wittern, J. Spillner, and W. Hummer, "A mixed-method empirical study of Function-as-a-Service software development in industrial practice," *J. Syst. Softw.*, vol. 149, pp. 340–359, 2019.
[7] M. Zhang, Y. Zhu, C. Zhang, and J. Liu, "Video processing with server less computing: A measurement study," in *Proceedings of the 29th ACM workshop on network and operating systems support for digital audio and video*, 2019, pp. 61–66.
[8] H. Shafiei, A. Khonsari, and P. Mousavi, "Serverless computing: A survey of opportunities, challenges, and applications," ACM Comput. Surv., vol. 54, no. 11s, pp. 1–32, 2022.
[9] J. Scheuner and P. Leitner, "Function-as-a-Service performance evaluation: A multivocal literature review," *J. Syst. Softw.*, vol. 170, no. 110708, p. 110708, 2020.
[10] H. Qiu et al., "Is function-as-a-service a good fit for latency-critical services?," in Proceedings of the Seventh International Workshop on Serverless Computing (WoSC7) 2021, 2021.
[11] A. Raza, I.Matta, N. Akhtar, V. Kalavari, & V. Isahagian. "Function-as-a-service: From an application developer's perspective". *JSys*, *1*(1), 2021, 1-20.
[12] A.-D. Lipitakis, G. Kousiouris, M. Nikolaidou, C. Bardaki, and D. Anagnostopoulos, "Empirical investigation of factors influencing function as a service performance in different cloud/edge system setups," Simul. Model. Pract. Theory, vol. 128, no. 102808, p. 102808, 2023.
[13] H. Tian, S. Li, A. Wang, W. Wang, T. Wu, and H. Yang, "Owl: Performance-aware scheduling for resource-efficient function-as-a-service cloud," in Proceedings of the 13th Symposium on Cloud Computing, 2022, pp. 78–93.
[14] M. Szalay, P. Matray, and L. Toka, "Real-time faas: Towards a latency bounded server less cloud," IEEE Transactions on Cloud Computing, 2022.
[15] O. Habala, M. Bobák, M. Šeleng, V. Tran, & L. Hluchý. "Architecture of a Function-as-a-Service Application". *Computing and Informatics*, *42*(4), 2021, 878-895.
[16] R. Hartauer, J. Manner, and G. Wirtz, "Cloud function lifecycle considerations for portability in function as a service," in *Proceedings of the 12th International Conference on Cloud Computing and Services Science*, 2022.
[17] F. M. Monetti, M. Bertoni, and A. Maffei, "A Systematic Literature Review: Key Performance Indicatorson Feeding-as-a-Service," in Swedish Production Symposium, 2024.
[18] I. A. G. Rosa, Reducing execution time in faaS cloud platforms. 2022.
[19] B. Pandya, D. Shah, A. Pourabdollah, and A. Lotfi, "Developing and comparing cloud-based fuzzy systems for monitoring health related signals in assistive environments," in The15th International Conference on PErvasive Technologies Related to Assistive Environments, 2022.
[20] N. Mahmoudi, & H. Khazaei, "Performance modeling of metric-based server less computing platforms". *IEEE transactions on cloud computing*, 2022.
[21] H. U. Khan, F. Ali, and S. Nazir, "Systematic analysis of software development in cloud computing perceptions," *J. Softw. (Malden)*, 2022.
[22] A. Miran, "The distributed systems landscape in cloud computing is transforming significantly because of several developing trends. This review focuses on critical trends transforming distributed systems' structure and operation in cloud environments," Journal of Information Technology and Informatics, vol. 3, no. 1, 2024.
[23] C. Lin, N. Mahmoudi, C. Fan, and H. Khazaei, "Fine-grained performance and cost modeling and optimization for FaaS applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 1, pp. 180–194, 2023.
[24] I. Zeebaree, "The Distributed Machine Learning in Cloud Computing and Web Technology: A Review of Scalability and Efficiency," Journal of Information Technology and Informatics, vol. 3, no. 1, 2024.
[25] K. Thanasegaran and M. F. Zolkipli, "Comparative Study on Cloud Computing Implementation and Security Challenges," Borneo International Journal, vol. 2636, no. 3, pp. 14–26, 2022.
[26] A. Egon, K. Potter, L. Doris, and R. Yamaganti, Evaluating the Development and Significance of Cloud Computing: Transforming the Digital Society. 2022.

[27] L. F. Albuquerque, F. S. Ferraz, R. F. A. P. Oliveira, and S. Galdino, "Function-as-a-service x platform-as-a-service: Towards a comparative study on FaaS and PaaS," in ICSEA, 2017, pp. 206–212.

[28] S. Malla and K. Christensen, "HPC in the cloud: Performance comparison of function as a service (FaaS) vs infrastructure as a service (IaaS)," Internet Technol. Lett., vol. 3, no. 1, p. e137, 2020.