

A Review Study of Function-as-a-Service for Small Scale Businesses in Pakistan

Muhammad Fahad*

Faculty of Computer Science, Institute of Business Administration, Karachi, Pakistan

E-mail: m.fahad.25346@khi.iba.edu.pk

ORCID iD: <https://orcid.org/0009-0009-6259-2228>

*Corresponding Author

Syed Muhammad Taha Rizvi

Faculty of Computer Science, Institute of Business Administration, Karachi, Pakistan

E-mail: s.rizvi.25358@khi.iba.edu.pk

Waqas Mahmood

Faculty of Computer Science, Institute of Business Administration, Karachi, Pakistan

E-mail: wmehmood@iba.edu.pk

Received: 03 February, 2023; Revised: 11 March, 2023; Accepted: 19 April, 2023; Published: 08 October, 2023

Abstract: In today's digitally driven world, many small to medium Scale Enterprises (SMEs) find it difficult to manage or scale businesses due to impediments in procuring and managing digital and software services. Many SMEs are unable to purchase the off-the-shelf software, as they are not tailor-made for their specific business use-cases and at times may not even fulfill the requirements. Creating custom software applications is an expensive undertaking and is difficult to maintain and scale. Further, the cost of deployment of these applications and ensuring availability either on premises or on cloud becomes a major hurdle. To overcome these hindrances, in this paper we propose the use of function-as-a-service technology in comparison to other options like micro-services and Service Oriented Architecture (SOA) and have highlighted the benefits in terms of ease of development and managing further we have put forth some economic benefits they have. FaaS can prove to be a major game changer service as it follows the model of pay-as-you-use and at idle time it can even scale to zero usage. This paper mainly discusses the option of using FaaS to help SMEs scale their application while cutting down on initial cost as well as economic impact in long run as compared to the other more traditional digital software service delivery models such as SaaS, ERP and custom application development.

Index Terms: FaaS, ERP, Cloud, SMEs, SOA, Cloud Service Provider.

1. Introduction

Function-as-a-service (FaaS) is a cloud computing model that allows users to run and execute small, independent pieces of code, called functions, in response to specific events or triggers. FaaS has the potential to offer a cost-effective and scalable alternative to traditional cloud computing models, making it an attractive option for small and medium-sized enterprises (SMEs) looking to leverage the benefits of the cloud.

However, FaaS is just one of many cloud computing options available, and it is important for SMEs to carefully consider the feasibility of using FaaS in comparison to other options. In this paper, we aim to provide a comparative analysis of FaaS and other cloud computing options, with a specific focus on their feasibility for SMEs in Pakistan.

We begin by discussing cloud computing models, the origins and key characteristics of FaaS, and then compare it to other cloud computing models, such as software-as-a-service (SaaS), infrastructure-as-a-service (IaaS), and platform-as-a-service (PaaS). We examine the various factors that SMEs should consider when choosing a cloud computing model, including cost, scalability, ease of use, and control.

2. Literature Review

2.1 Emergence Of Serverless Computing

In [1] Paul Castro et al. discussed how the beginning of serverless computing took place, they discussed three areas which were the major driving force for the adoption of this new serverless computing concept. The first one became the model of pay-as-you-execute, meaning the moment you piece of code gets executed you are being charged for that and all the complexity of infrastructure gets abstracted from the developers, and they just have to provide function and all the scaling and load balancing is done by the cloud provider.

The authors have also contrasted IaaS, PaaS, SaaS/BaaS and FaaS and have concluded that IaaS provides great flexibility as all low level infrastructure details can be configured by the developer but in turn increases complexity. PaaS is a platform that runs code continuously instead of on demand, this makes it less feasible in terms of cost. SaaS is more of a standardized off the shelf software even though it provides good integrational functionality but still contains standardized components. FaaS becomes an imminent choice where cost is the main constraint, and it also provides great flexibility in terms of scale as it can scale down to zero and is event driven while being granular as it can accept functions instead of service or module. The study related to emergence of serverless computing was further explored by [2] Rajan et al. where authors emphasized that to achieve optimum cost, reduce the efforts in configuring the servers and leverage on demand scaling the serverless architecture must be employed the authors also carried out the experimental study using AWS lambda functions and discussed the benefits of using FaaS.

[3] Savage et al. also highlighted the advantage the serverless computing has specially for various businesses because it lets the developer focus more on functionality building rather than setting up the infrastructure, patching the system and managing the runtimes.

In [4] Erwin Van et al. contrasted the era in which IBM machines could be leased for some time at the cost of thousands of US Dollar but today in the era of cloud computing renting a powerful machine can be very economical and you are only being charged according to the usage. Serverless computing has further fine-grained usage making the architecture more feasible and economical for businesses. The economic benefits of FaaS were further analyzed by Adzic et al. [5] where the authors compared different instances of lambda with EC2 and App Engine and concluded that indeed serverless functions are most economical as they do not utilize a reserved capacity rather charges for actual utilization only.

2.2 Staying Ahead in Serverless Computing Reign

In [6] Mohit Sewak et al. discussed about the comparisons between microservices architecture and serverless architecture and declared FaaS as microservice enabler as it provides more granularity along with elasticity in compute and instead of providing a more robust way of Service Oriented Architecture which is the case with microservices, serverless provides more of an event driven approach with http triggers to which the handler functions get attached.

The authors have also highlighted some tradeoffs of employing FaaS as function counts become increasingly high and there might be some challenges as the size gets really large. Further vendor locking can become an issue as we get stuck with a single cloud provider.

He has also listed some offerings of FaaS like in IoT and Supply Chain business. It can really make the difference by providing a low cost yet a very powerful and robust solution building platform.

In [7] Monteiro et al. proposed a new platform for orchestration of microservice, he highlighted some of the known implications of microservices communication and integration with one another and proposed a Domain Specific Language (DSL) to define orchestrations, in contrast to it FaaS orchestrations are normally managed by the vendor itself so in this regard it provides an edge over microservices based applications.

In [8] Balalaie et al discussed the microservices from the perspective of cloud native applications and the challenges that were faced when migrating Mobile Backend as a Service from monolith to microservices based architecture for better reusability, implementing a distributed system and a cloud native application communicating through REST API or RPC.

2.3 Guide To Implement FaaS In Correct Way and Software Engineering Practices

In [9] Jussi Nupponen et al. highlighted some problems that could arise if the serverless functions were not implemented the correct way. The authors got into more technical details like talking about the async function implementation could get really complex as FaaS provided more of a synchronous way of implementing the functions. But with the use of queues or pub/sub architecture we could find a better solution to that problem. Function that invokes another function could also increase the cost and must be avoided as well.

In [10] Erwin van Eyk et al. have put in front some software engineering practices and have separated out business and operational logic and stated that the gap between PaaS and SaaS have been filled by FaaS. It has excluded operational logic and only focuses on business logic.

In [11] McGrath et al came up with a novel serverless computing platform that was tailored to deliver optimal performance, the authors developed their own FaaS platform through a containerized .NET application running on MS Azure, the goal was perform a comparative analysis of execution performance of functions on FaaS platforms provided by AWS, Azure, GCP and IBM with their performant platform. The results indicated the proposed platform as more efficient and performant with greater throughput.

In [12] Baldini et al conducted a study on edge computing which is an idea to place computing resources closer to user within a network topology. Authors introduce a concept of community leaders which would use control theory to

host serverless functions near the edge to reduce latency. The whole system would be based on self-management and will oversee partitioning, allocation, placement, and scaling.

2.4 Survey On Serverless Computing And Implications:

In [13] Philipp Leitner et al. conducted a research survey in which they interviewed various professionals working in the domain of cloud and gathered some valuable insights.

Most of them were using AWS as the main cloud platform while others also reported MS Azure, Google Cloud Platform, Heroku and Digital Ocean. Most of the respondents and interviewees were in favor of using FaaS and concluded that having a mindset to grasp the idea behind the FaaS can be pivotal in embracing the technology as it is a bit difficult at first but those having background in functional, reactive and stream programming might find it easier to understand.

In [14] Mohammad Shahradeh et al. have discussed some performance implications of using FaaS like studying latency and throughput and using profiling utility for that. Unavailability of containers for function execution queues that function which causes delay in execution. Initialization time and cold start times also add to major delay after which the function starts running. Some other implications like branch misprediction and increased memory bandwidth usage are also worth mentioning.

In [15] Lee et al. conducted a study to evaluate the performance of serverless functions on actual production environments and analyzed the performance during parallel execution of functions, High performance computing task and CPU intensive tasks and concluded that as of now it is not much suitable for performance-oriented applications.

Similarly [16] Fox et al. discussed the FaaS from the perspective of event driven computing and commented that it can play a pivotal role in applications where incoming traffic is uncertain and can provide a stateless on demand execution, but also highlighted that for HPC applications and longer running task it is not suitable.

In [17] Kuhlenskamp et al. chalked out some of the ifs and buts of serverless functions and monitored the performance of these functions in parallel executions, delays in security patches of application packages and frameworks from vendor's side, less flexibility in controlling elasticity of functions running on containers, volatile cost of executing a function, limited choice of runtime environments. Author not only highlighted these issues but also presented ways to mitigate them as well. Similarly, Armbrust et al. [18] raised concerns over the obstacles in serverless applications like availability, lack of standardization and data auditability, slow data transfer rate, cold start, buggy in large distributed system, performance issues etc.

In [19] Lynn et al. presented a study on enterprise serverless applications and discussed 7 enterprise level serverless platforms and listed some guidelines for selection and reasons to opt for or opt out of any enterprise level cloud platform.

In [20] L. Wang et al. performed experiments to uncover the performance issues of serverless functions specifically related to process isolation, management, and allocation of resources. The authors also highlighted some scheduling related issues and isolation across different tenants, problems related to cold starts and network throughput.

3. Serverless Computing

Serverless computing, in industry terms, is a cloud computing model wherein a cloud provider dynamically allocates and scales computing resources, such as servers and infrastructure, in response to the demands of an application.

In a serverless computing model, a cloud provider charges users based on the execution of their code, rather than on the use of dedicated servers or infrastructure. This is made possible by the availability of rapidly scalable and maintainable server infrastructure.

With serverless computing, users do not need to worry about the underlying infrastructure and can focus on building and deploying their applications. This can be a cost-effective option for SMEs as they can have variable or unpredictable levels of usage, and FaaS allows them to only pay for the resources they consume.

Serverless computing can be implemented using a variety of technologies, including function-as-a-service, event-driven architectures, and managed services. Some examples of cloud providers that offer serverless computing platforms include AWS, Azure, and Google Cloud. Before this we had to manually containerize the services and assign them pods which can be added or removed manually if we were using any orchestrating utility like Kubernetes.

3.1 Function As A Service (FaaS), An Option For Serverless Computing

Serverless computing is being used as a broader term and consists of various approaches for its implementation like function-as-a-service (FaaS) and backend-as-a-service (BaaS).

This paper is specifically concerned with FaaS. A major advantage that FaaS offers is its simplicity in that it allows management of code at a comparatively smaller level. FaaS can be used at a level of a simple function, which is then utilized through a trigger that invokes or calls that function. The trigger can be invoked by calling a REST pattern URL.

The principle behind serverless computing is that it is stateless and elastic and the whole lifetime of the function is created as soon as its invocation is requested, and then goes back to being switched off once the function call is over i.e., the remainder of the time it does not live in any memory. A big challenge for these stateless services is that there must

be persistent storage that also needs to be on demand and must be scalable on run time.

This is where elastic databases come into picture like AWS Aurora, Microsoft CosmosDB, Mozilla Licensed FaunaDB and Open-Source Databases like DGraph.

4. Application Development, Deployment and Maintenance For Small and Medium Scale Businesses Using FaaS

With the exponential increase in internet access around the globe businesses find it almost impossible to grow without any digital presence. Websites for targeting potential customers, managing sales and supply chain often use web applications, while dashboards are used to view reports critical for decision making. In managing all these mentioned areas complex software solutions are used and maintained.

While large businesses often have enough capital to invest in developing, deploying, and maintaining all these applications, they even have options to purchase off the shelf enterprise solutions that can be tailored based on their needs. In contrast to SMEs, which have limited initial capital and struggle to increase their reach or manage their operations using software applications.

The cost of application development is considered prohibitively high as it involves either engaging a vendor or hiring an in-house team. This is followed by initiating a project in a waterfall or agile model, beginning with requirement gathering, establishing proper business workflows, and then initiating development. For many SMEs, as the business develops, requirement gathering cannot be carried out in a traditional manner, and this results in a gradual change in application architecture as the project progresses. The scale of the application is also a major concern as the business owners really have no idea when the customer base might increase, resulting in application server failure and running out of computing resources. To manage this, the cost of purchasing powerful servers must be considered as well. The powerful infrastructure may also be needed for a small amount of time.

Purchasing off-the-shelf software might not be feasible for them as well because the cost will be high and most of the components of that software might not be needed plus its maintenance and deployment will also be very expensive.

To overcome these challenges and hurdles using a cloud platform is the best possible option as SMEs will only be paying for the computing, storage and network resources that are being used. In addition to that most of the services on cloud are managed by the providers hence SMEs will not need any budget for the operational cost of maintaining and upgrading servers, network devices and virtual machines.

Therefore, using a cloud platform and using IaaS or Infrastructure as a service, it gives great flexibility and cost SMEs somewhat less. However, within the ambit of cloud services, FaaS still, as a service model appears less costly than other cloud services.

Function-as-a- service will solve the first problem that we stated earlier, that is the requirements will gradually become clearer and SMEs can begin development with a very lean approach by building first some very core components and using pure functions to implement them and adding a trigger or timer for the execution of that action or function. In this way the application will be lean and simple in the beginning and will be scalable as the business starts expanding. Compared to other solutions, this will still be manageable.

Having solved the first challenge, we come towards the second problem of avoiding spending huge sums of initial valuable capital in purchasing computing resources either on premises or on cloud and bearing all the cost upfront even when they might not be needed at all. With the new concepts of elastic computing the resource usage can be made on demand and CPU usage might even be reduced to 0% when there is no execution and CPU is idle resulting in 0 cost in monetary units when there is no usage.

This will help immensely in saving the huge upfront cost and the best part is that overall cloud platforms are very reliable and reasonably secure outside of the box.

5. Comparing Monolithic, Microservices and FaaS for creating Service Oriented Architecture for Businesses

5.1 Monolithic Architecture

It is one of the legacy architectures that works on the principles of n-tier architecture and scaling up instead of scaling out, the application is mostly tightly coupled and is spread out into different modules that communicate with each other through service calls, sharing dynamic linked library references or using socket real time communication approaches.

Monolithic architecture makes application development fast, easy and till date many legacy systems are successfully using it. But some of the major drawbacks of this architecture is that the application is tightly coupled which can cause tons of problems on its own apart from that scaling the application is also a major hurdle as components cannot be separately deployed on different machines hence really powerful servers are needed to deploy and maintain them.

5.2 Microservice Architecture

As the name suggests it breaks down the application into numerous micro level services that are independent and are loosely coupled. It is one of the very successful architectural patterns in modern times. Even though building a microservice architecture is difficult, the biggest challenge being inter-service communication, for which we need queues and pub/sub architecture. Microservices help us in scaling our application out without needing a very powerful server; rather we can deploy different services on different reasonably powerful servers which are just responsible for that one service. Adding to the complexity, the deployment of microservices is not that simple. We normally first containerize different services and then use different orchestration services to maintain and scale them.

Nevertheless, the advantages of using microservices largely outnumber the pitfalls but it is good for large scale applications and not needed for small applications with limited users and scope.

5.3 Function-as-a-service

As talked about earlier, FaaS granularizes the application further but this time without making it more complex. Instead, we can use FaaS as a way of creating a microservice as primarily it is just a combination of functions. The functions in FaaS are event driven and communicate with each other via triggers and connected with API endpoints.

Instead of services communicating with each other now functions are communicating with each other which makes it straightforward to just make API calls. However, there is still a caveat that its functions must be pure and must be loose coupled with one and other. Reactive or functional programming paradigm is an example of that. The problem of deployment and maintenance has also been taken away by the cloud providers as the scale execution and reliability are all ensured by the cloud providers themselves. All these benefits make FaaS a really good choice for small businesses starting with a limited budget.

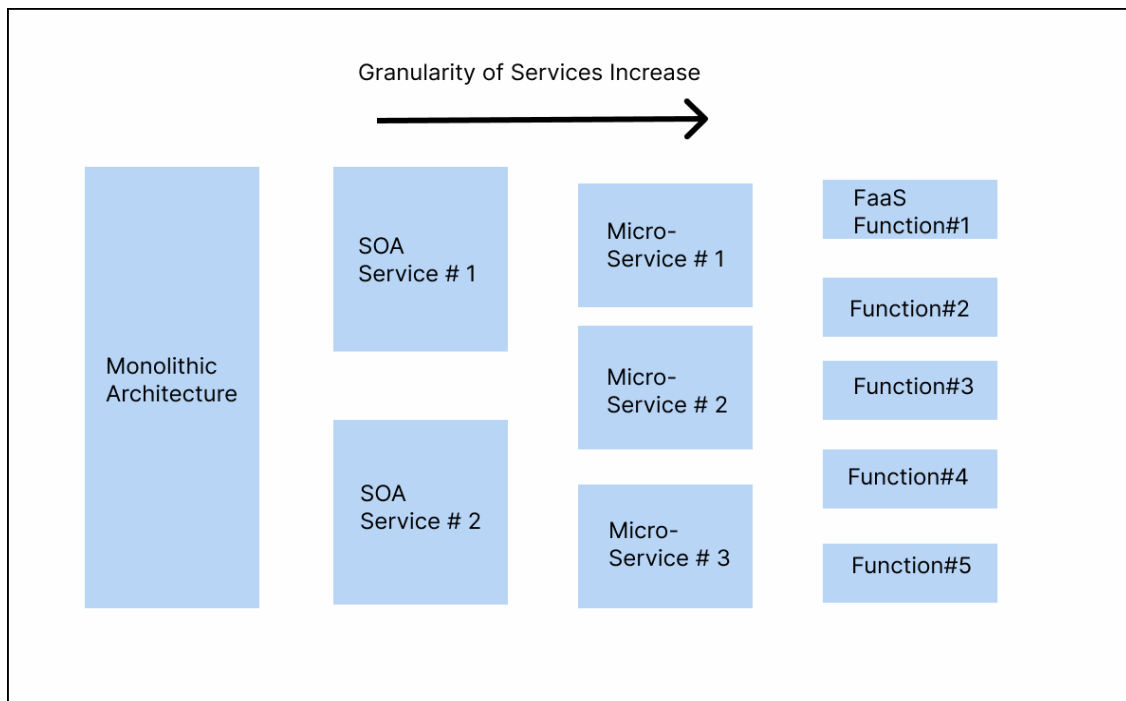


Fig. 1. Shows the comparisons among various architectural patterns for software systems from **Monolithic** being the least flexible as all functionalities are tightly coupled with one and other, **SOA** large services separated out but still difficult to scale out, **micro-services** loosely coupled functionality but difficult to manage and maintain, **FaaS** highest level of granularity but easier to maintain.

6. Comparing Different Cloud Providers Providing FaaS and Pricing Model:

6.1 Microsoft Azure Pricing Model:

The pricing model of Microsoft Azure is simple and easy to understand. Let's take an example of an Azure Function that runs 4 million times and each time for 2 sec taking up memory of around 256MB per second.

There are two types of cost that are consumption cost and number of executions cost.

First, we will discuss consumption cost for the given example. $4 \text{ million} * 2 \text{ sec}$ will be 8 million seconds and the amount of memory consumed will be $256/1024 * 8 \text{ million}$ will give 2 million GB-sec, now current consumption price is 0.000016 GB-sec that will be multiplied by 2 million which will yield a total of USD \$32.00 consumption cost plus

USD \$0.20 per million execution cost that will become $0.20 * 4$ resulting in USD \$0.80 making it a grand total of USD \$32.80 per month (excluding all free monthly grants).

6.2 Amazon Web Services Pricing Model:

Amazon offers two different types of CPU architecture which are x86 and ARM. We will be covering only x86 in this paper for uniformity. Amazon pricing model can be explained with a similar scenario as MS Azure. A Lambda Function that runs 4 million times and each time for 2 sec or (2000ms) and memory consumption of about 256MB per second.

First, we will cover consumption costs and then request costs. $4 \text{ million} * 2 \text{ sec}$ will give 8 million sec and $256/1024 \text{ GB}$ consumed for that 8 million second making a total of 2 million GB-sec. Again, AWS uses similar pricing to Azure that is 0.000016 GB-sec making the consumption cost of USD \$32.00 and USD \$0.20 for every million requests resulting in a total of USD \$0.80. Hence the grand total becomes USD \$32.80 per month.

The extra flexibilities that AWS offer are the CPU architecture choices like x86 and ARM and more flexible memory consumption model which is capped to 10GBs.

6.3 Google Cloud Functions Pricing Model

The pricing model of Google Cloud Platform (GCP) is a bit different from AWS and Azure; all the functions are containerized in GCP and there is an image registry or artefact registry cost associated with every function as well. GCP provides even more flexibility in terms of selecting CPU for execution of these functions, plus max memory that can be allocated can be up to 32GB.

GCP provides two tiers that are tier 1 and tier 2 based on geography and services available within that region in addition to that tier 2 prices are higher compared to tier 1. Now taking the same example of a Google Cloud Function that consumes 256MB per second and runs 4 million times and each time runs for exactly 2 sec and utilizes 0.167 GHz CPU so the cost would be $4 \text{ million} * 2 \text{ sec}$ will give us 8 million invocations, now for memory consumption we can say that $256/1024$ that is $0.25 * 8 \text{ million}$ which gives us 2 million Gb-sec plus $167/1000 \text{ GHz}$ will give $0.167 \text{ GHz} * 8 \text{ million}$ will give 1,336,000 GHz-second, now finally the cost would be $0.4 * 4 \text{ million}$ that is USD \$1.60 invocation cost plus $2 \text{ million} * 0.0000025$ will give USD \$5 plus $1336000 \text{ GHz-sec} * 0.0000100$ will result in USD \$13.36 which makes a grand total of USD \$19.96 a month (without monthly free grants and container registry cost).

Similarly, IBM, Alibaba and Oracle provide their own pricing models.

6.4 Using IaaS or Inhouse for Application Deployment

In contrast to above mentioned cloud based event driven on demand executing functions we can also use cloud infrastructure service and can use cloud based VMs but the cost would be significantly higher just for the example an EC2 instance provided by AWS with specifications of 8GB RAM and EC2 t3(large) with 2 instances of general purpose VCPU with windows installed can cost USD \$0.11 hourly for one month availability it can be around USD \$80 and will be barely enough to run the application and in peak times more instances will be required which will significantly increase the cost further licensing cost of other software are also not included in it.

Finally, establishing the in-house data center will be the costliest option as a lot of equipment, server, backups, and energy costs will be associated with it. Further maintenance, upgradation and management costs of in-house infrastructure would be significantly high which will make it infeasible for the medium and small-scale businesses.

7. Results and Discussion

The research conducted signifies the adoption of FaaS for SMEs as we have taken different metrics into account like ease of development, ease of maintenance and management, in addition to it while addressing the major concerns for SMEs we have considered the reduction of total upfront cost both in terms of development and deployment and benefits of faster time to market. The good thing about FaaS is development of application with more granularity and control meaning that one can create the features and functionalities that are deemed necessary without thinking too much about complete end to end application as compared to other architectural patterns like micro-services or SOA that requires complete design and architecture to be put in place from the very beginning and all the development must be carried out by keeping all these things in mind which are usually not the area of focus and concern for SMEs. Since FaaS is offered as a managed service the upgradation and patching of all the software utilities are carried out by the Cloud Service Provider (CSP) themselves. It is also the job CSP to be in compliance with the SLAs as they are accountable for it.

Finally highlighting and pricing models of 3 major cloud service providers can prove to be vital for any SME as they must first understand how the pricing works because it will help them in future as they start scaling out their application which will incur more cost. The pricing models typically affect 3 areas which are number of invocations of a function, the processing/CPU time required for that function and the amount of memory consumed during the execution. So, to make best use of the FaaS one must tailor the function in a way that cost is minimized. Further the developer should also try to figure out the Cloud Service Provider whose pricing model fits best for the organizational needs.

The comparison made in this research clearly presents that the use of FaaS is most suitable model for SMEs in the light of above-mentioned metrics as it cuts down upfront cost and helps in faster time to market plus help the developers focus more on implementing the functionality rather than worrying about infrastructure and application.

8. Future Work and Conclusion

Based on the analysis and the comparison drawn among all the viable options for SMEs, as laid out in this paper, it will not be difficult to say that for SMEs who want to scale their businesses and have an online reach, FaaS can be among the best alternatives to the traditional software development and deployment options. In future this work can be extended to help the SMEs identify how they should optimize the implementation of their functions in a way that can make this model viable in the long run especially when the application starts expanding out. FaaS can provide a lot of flexibility in terms of costing, scaling, and performance of the application. Cloud computing is here to stay, and it is safe to say that FaaS can be the next big thing for SMEs as it provides simpler, reliable, and much more scalable products at a very reasonable cost. It also enables the SMEs to make more room in their budget for investing in other parts of the business-like operations and management where more resources and investments are required.

References

- [1] Shahradd, M., Balkind, J., & Wentzlaff, D. (2019). Architectural Implications of Function-as-a-Service Computing. Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture - MICRO '52. doi:10.1145/3352460.3358296
- [2] Rajan, D. R. A. P. (2018). Serverless Architecture - A Revolution in Cloud Computing. 2018 Tenth International Conference on Advanced Computing (ICoAC). doi:10.1109/icoac44903.2018.89390
- [3] Savage, N. (2018). Going serverless. Communications of the ACM, 61(2), 15–16. doi:10.1145/3171583
- [4] Eyk, Erwin Van et al. "Serverless is More: From PaaS to Present Cloud Computing." IEEE Internet Computing 22 (2018): 8-17.
- [5] Adzic, G., Chatley, R., 2017. Serverless computing: Economic and architectural impact. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering. ACM, New York, NY, USA, pp. 884–889. doi: 10.1145/3106237.
- [6] Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The rise of serverless computing. Communications of the ACM, 62(12), 44–54. doi:10.1145/3368454
- [7] Monteiro, D., Gadelha, R., Maia, P. H. M., Rocha, L. S., & Mendonça, N. C. (2018). Beethoven: An Event-Driven Lightweight Platform for Microservice Orchestration. Lecture Notes in Computer Science, 191–199. doi:10.1007/978-3-030-00761-4_13
- [8] Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture. IEEE Software, 33(3), 42–52. doi:10.1109/ms.2016.64
- [9] Sewak, M., & Singh, S. (2018). Winning in the Era of Serverless Computing and Function as a Service. 2018 3rd International Conference for Convergence in Technology (I2CT). doi:10.1109/i2ct.2018.8529465
- [10] Van Eyk, E., Iosup, A., Seif, S., & Thömmes, M. (2017). The SPEC cloud group's research vision on FaaS and serverless architectures. Proceedings of the 2nd International Workshop on Serverless Computing - WoSC '17. doi:10.1145/3154847.3154848
- [11] McGrath, M. Garrett and Paul R. Brenner. "Serverless Computing: Design, Implementation, and Performance." 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW) (2017): 405-410.
- [12] Baldini, I., Castro, P., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., ... Suter, P. (2016). Cloud-native, event-based programming for mobile applications. Proceedings of the International Workshop on Mobile Software Engineering and Systems - MOBILESoft '16. doi:10.1145/2897073.2897713
- [13] Nupponen, J., & Taibi, D. (2020). Serverless: What it Is, What to Do and What Not to Do. 2020 IEEE International Conference on Software Architecture Companion (ICSA-C). doi:10.1109/icsa-c50368.2020.00016
- [14] Leitner, P., Wittern, E., Spillner, J., & Hummer, W. (2019). A mixed-method empirical study of Function-as-a-Service software development in industrial practice. Journal of Systems and Software, 149, 340–359. doi:10.1016/j.jss.2018.12.013
- [15] Lee, H., Satyam, K., & Fox, G. (2018). Evaluation of Production Serverless Computing Environments. 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). doi:10.1109/cloud.2018.00062
- [16] Fox, Geoffrey & Isahagian, Vatche & Muthusamy, Vinod & Slominski, Aleksander. (2017). Status of Serverless Computing and Function-as-a-Service(FaaS) in Industry and Research. 10.13140/RG.2.2.15007.87206.
- [17] Kuhlenskamp, J., Werner, S., & Tai, S. (2020). The Ifs and Buts of Less is More: A Serverless Computing Reality Check. 2020 IEEE International Conference on Cloud Engineering (IC2E). doi:10.1109/ic2e48712.2020.00023
- [18] Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., ... Rabkin, A. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50. doi:10.1145/1721654.1721672
- [19] Lynn, T., Rosati, P., Lejeune, A., & Emeakaroha, V. (2017). A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms. 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). doi:10.1109/cloudcom.2017.15
- [20] L. Wang, M. Li, Y. Zhang, T. Ristenpart, and M. Swift, "Peeking behind the curtains of serverless platforms," in 2018 USENIX Annual Technical Conference (ATC 18)

Authors' Profiles



Muhammad Fahad is currently pursuing MS in Computer Science from Institute of Business Administration Karachi (IBA) and is also currently working as a Full Stack Software Engineer, he has great interest in Cloud Computing and writing his thesis in the very area of computer science.



Syed Muhammad Taha Rizvi is currently pursuing MS in Computer Science from Institute of Business Administration Karachi (IBA) and is working in capacity of Business Analyst and Enterprise Services, he has great interest in understanding the business flows and their automation.



Waqas Mahmood has been serving as a faculty member at Institute of Business Administration Karachi IBA since 1999, he has extensive experience at various National and International Organizations. He has also worked on key position at Government Organizations as well.

How to cite this paper: Muhammad Fahad, Syed Muhammad Taha Rizvi, Waqas Mahmood, "A Review Study of Function-as-a-Service for Small Scale Businesses in Pakistan", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.15, No.5, pp. 23-30, 2023. DOI:10.5815/ijieeb.2023.05.03