A Review on Function-as-a-Service in Cloud Computing

[1]Swati C. Pakhale, [2]Prof. Sachin A. Vyavhare
[1]P.G. Student, [2]Assistant Professor
[1]Computer Science and Information Technology
[1]Sanmati Engineering College, Washim,India

Abstract: World relies on cloud today for its routine work. Cloud services offered to its users by cloud service providers. Cloud services are generally implemented on cluster based computers to provide the necessary scale and performance required by such services. The database storage system is not sufficient in the 21 century. As time goes and use and requirement of cloud data servers increasing day by day, it requires more space to store data of cloud users on servers also it is a tough task to maintain and keep it secure. Thus data server storage architecture is the biggest security threat concerning data mining on cloud. Studies point to a new model, known as Function-as-a-Service (FaaS), as an alternative that would offer a more efficient use of resources and lower costs. The conclusions obtained showed that FaaS presented an equivalent performance, a more efficient scalability and the costs influenced by workload type.

Keyword- Cloud, FaaS, data server, scalability.

I. INTRODUCTION

Cloud computing is an evolving trend in the Information Technology (IT) and distributed computing. Clouds provide on-demand dynamic delivery of high quality and low-cost applications, infrastructure, and further IT resources as services through internet with payment per usage pricing approach . Furthermore, cloud computing is a model that enables expanded access to IT resources with minimal management efforts. Customers can customize their cloud usage requirements in terms of storage, servers, applications, operating system, and development environment. Moreover, the customers can utilize the usage of the cloud services according to the defined service level agreement (SLA) that includes the desired Quality of Service (QoS) constrains [1].

The last decade has seen tremendous innovation within the enterprise IT area. The primary focus of this technological innovation has been facultative businesses legerity, improve resiliency and drive price efficiencies. The revolution within the server computing area over the past decade has allowed enterprise IT to develop and deploy ascendable computer code without concern regarding the underlying infrastructure. At constant time, price of technology operations have conjointly gone down considerably whereas rising time-to-market for enterprises. Server computing is currently evolving towards even smaller units of scale – from virtual machines to containers to serverless. The technology evolution has been depicted below. This thought paper provides a vender neutral outline of business counseled principles and key concerns for architecting function-as-a-service (FaaS) systems. With various cloud computing, individuals and enterprise organizations are able to store and process data, manage applications, and deploy applications with the support of virtualization resources [2].

In the last few years the new infrastructure technologies have changed a lot. From metal hardware provisioning, virtualization, to Cloud IaaS, container technologies, the software developer has changed the way it programs an application, with many service type applications moving to micro services architecture. The most recent addition to this new way of program applications could it be called Cloud based event applications which major representation it is the quite recent commercial offer of those services, such as, Amazon Lambda, IBM Blue mix's Open Whisk, Google Cloud Functions, and Microsoft Azure Functions. All those services could be en-globed in what we denominate FaaS Architecture.

In line with cloud computing emergence as the dominant enterprise computing paradigm, our conceptualization of the cloud computing reference architecture and service construction has also evolved. For example, to address the need for cost reduction and rapid provisioning, virtualization has moved beyond hardware to containers. More recently, serverless computing or Function-as-a-Service has been presented as a means to introduce further cost-efficiencies, reduce configuration and management overheads, and rapidly increase an application's ability to speed up, scale up and scale down in the cloud. The potential of this new computation model is reflected in the introduction of FaaS computing platforms by the main hyperscale cloud service providers. This paper provides an overview and multi-level feature analysis of seven enterprise serverless computing platforms. It reviews extant research on these platforms and identifies the emergence of AWS Lambda as a de facto base platform for research on enterprise serverless cloud computing. The paper concludes with a summary of avenues for further research. [4]

II. SERVERLESS CLOUD COMPUTING

Before we get into what FaaS is, it's important to understand another term-serverless computing. Serverless computing is a cloud computing model which aims to abstract server management and low-level infrastructure decisions away from developers. In this model, allocation of resources is managed by the cloud provider instead of the application architect, which can bring some serious benefits. In other words, serverless aims to do exactly what it sounds like-allow applications to be developed without concerns for implementing, tweaking, or scaling a server.

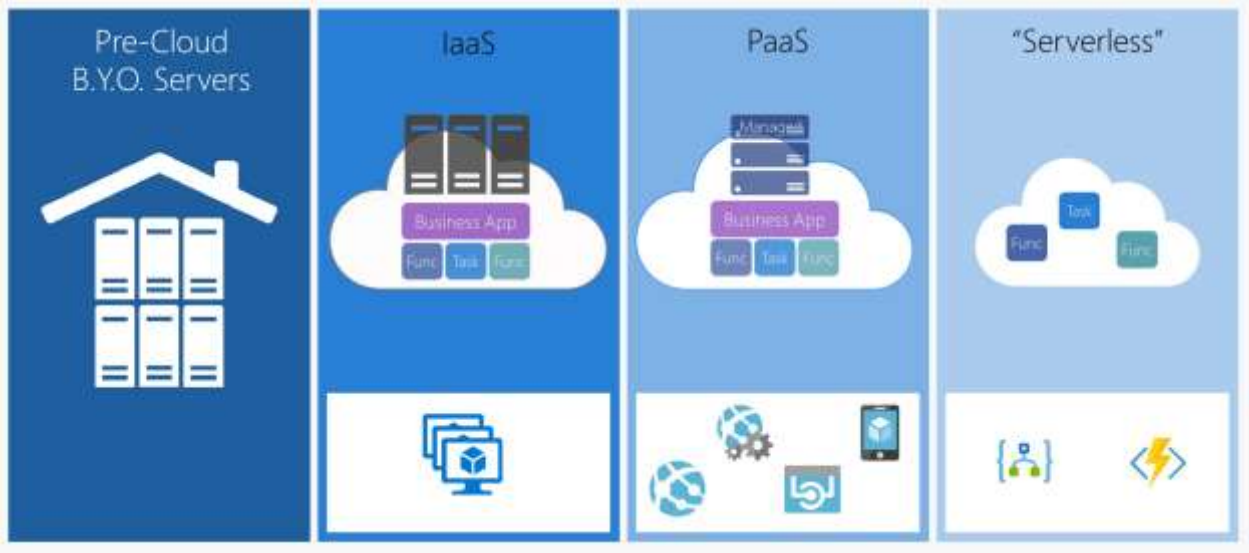

Fig. 1: evolution of pre-cloud to serverless

From the perspective of a cloud provider, serverless computing provides an additional opportunity to control the entire development stack, reduce operational costs by efficient optimization and management of cloud resources, and enabling a serverless ecosystem that encourages the deployment of additional cloud services. Serverless platforms promise new capabilities that make writing scalable microservices easier and cost effective, positioning themselves as the next step in the evolution of cloud computing architectures. Most of the prominent cloud computing providers including Amazon, IBM, Microsoft, and Google have recently released serverless computing capabilities [6].

A taxonomy of serverless design patterns that experimented using Lambda and primitives provided by AWS categorized serverless design patterns into six groups [7]:

1) periodic invocation,
2) event driven,
3) data transformation,
4) data streaming,
5) state machine, and
6) Bundled pattern.

III. So What Exactly is the Function as a Service?

Function as a Service (FaaS) is the concept of serverless computing which works on serverless architectures. Software developers can use FaaS to maximum advantage to deploy an individual "function", action, or piece of business logic. They are expected to start within milliseconds and process individual requests and then the process ends. Following are the principles of FaaS:

- Complete abstraction of servers away from the developer
- Billing based on consumption and executions, not server instance sizes
- Services that are event-driven and instantaneously scalable

The journey of cloud from simple pre-cloud to serverless or Function as a service is gone through many stages, developing many ideas. Cloud provides Infrastructure-as-a-Service for Business App, which provide an infrastructure for use to cloud users. Platform-as-a-Service comes in action providing a platform for developers and general users. There are many more terms comes under cloud like SaaS, DPaaS, and now it jumps to FaaS which is a serverless cloud system. FaaS is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.

FaaS could be a comparatively new thought that was initial created out there in 2014 by hook.io and is currently enforced in services like AWS Lambda, Google Cloud Functions, IBM OpenWhisk and Microsoft Azure Functions. It provides a method to attain the serverless dream permitting developers to execute code in response to events while not building out or maintaining a fancy infrastructure. What this suggests is that you just will simply transfer standard chunks of practicality into the cloud that area unit dead severally. Imagine the possibilities! Instead of scaling a monolithic REST server to handle potential load, you'll be able to currently split the server into a bunch of functions which might be scaled mechanically and severally. If you're at home with microservices, this image may cause you to feel one thing [5].

Fig. 2: technology evolution of FaaS

IV. A BRIEF ON FAAS SERVICES

Function as a Service or serverless cloud has many pros and cons over other well known cloud platform and services. FaaS required very few developer logistics, the server management is handled by someone else. It provides higher developer velocity which makes it more time focused on writing code, app specific logics. By using FaaS services, rather than scaling your entire application you can scale your functions automatically and independently with usage, this makes it more scalable. There is no requirement to pay for idle resources. It provides built in availability and fault tolerance mechanism.

As the many things in favor of users of FaaS, some areas are not so attractive. When someone else is managing your infrastructure so it can be tough to understand the entire system and it decreased the transparency. FaaS systems are potentially tough to debug, many tools allow remote debugging and some services provide a mirrored local development environment but there is still a need for improved tools. It is very tough to keep track of ton of functions deployed and it need for a better tooling like developmental, diagnostics and visualization.

From a cost perspective, the benefits of a serverless architecture are most apparent for bursty, compute intensive workloads. Bursty workloads fare well because the developer offloads the elasticity of the function to the platform, and just as important, the function can scale to zero, so there is no cost to the consumer when the system is idle. Compute intensive workloads are appropriate since in most platforms today, the price of a function invocation is proportional to the running time of the function. Hence, I/O bound functions are paying for compute resources that they are not fully taking advantage of. In this case, a multi-tenant server application that multiplexes requests may be cheaper to operate. From a programming model perspective, the stateless nature of serverless functions lends themselves to application structure similar to those found in functional reactive programming [6].

V. CONCLUSION

In this paper, we discussed about the Function as a Service, the challenges and showed that by using FaaS we can simplify the structure of cloud. Due to the restriction on the running time in FaaS, an extension for the running time limit by the cloud vendors could allow the option of totally running the complete workflow using the serverless architecture. Serverless has the potential to revolutionize the manner that we tend to write and deploy code. We're now not thinking in terms of apps and API's or long running processes, instead we're thinking in terms of functions that can respond to requests and process data based on events. We're thinking of recent readying models and new systems to manage little chunks of code that run short periods of your time.

REFERENCES

[1] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds," Futur. Gener. Comput. Syst., vol. 29, no. 1, pp. 158– 169, 2013.

[2] Bardsiri and S. Hashemi, "A Review of Workflow Scheduling in Cloud Computing Environment," Int. J. Comput. Sci. Manag. Res., vol. 1, no. 3, pp. 348–351, 2012.

[3] Omar Alqaryouti and Nur Siyam, "Serverless Computing and Scheduling Tasks on Cloud: A Review", ASRJETS journal, ISSN:2313-4402, Page: 235-247.

[4] Theo Lynn, Arnaud Lejeune, V.Emeakaroha,"A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms", 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 28 December 2017.

[5] Luke Angel , "Docker+Serverless = Functions as a Service (FaaS)" http://lukeangel.co/cross-platform/docker-servless-faas-functions-as-a-service/; April,28,2017.

[6] Vatche Ishakian, Aleksander Slominski, Vinod Muthusamy; "Serverless Programming (Function as a Service)",A conference paper published on Researchgate,in June 2017.

[7] Sanghyun Hong, Abhinav Srivastava, William Shambrook ,Tudor Dumitras," Go Serverless: Securing Cloud via Serverless Design Patterns", Frame.io, 2018.