Serverless Computing a Cloud Technology Revamp

S Ajithkumar[1], Anirudh Raghavan[2], Dr. Azra Nasreen

Department of Computer Science, RV College of Engineering, Bengaluru, India

---***---

Abstract - *Serverless computing architecture is a micro services architecture where developers can focus only on functionality of the system and ignore system operations such as manage servers to optimistically run the software, manage databases, scale storage, memory etc. These frameworks often abstract from the specificities of serverless platforms, or allow the development of a mini serverless model over established clouds. Serverless computing is a modern cloud computing architecture promising to revolutionize the way apps are designed and distributed. Tiny pieces of tool called functions are deployed in this computing form in the cloud without management and software development at lowest possible costs.*

*Key Words***:** Serverless, computing architecture, micro services, database, memory, storage, framework.

1. INTRODUCTION

Serverless computing is a different form of computation that is gaining traction in among cloud application developers at the moment. This is due to the key benefits it offers, including no management required, infinite elasticity and minimal costs. Which makes, serverless computing a better choice when compared to others like PaaS platforms, which allows small software components called functions to be deployed and supplied in the cloud. High-performance computing (HPC) systems have historically been addressed with large-scale issues. High performance computing systems are costly, not elastic and also in the sense that a lot of documentation is required to register for computing resources, write programs that meet those guidelines and use different libraries, determine memory and runtime specifications for such programs. However, this expandability of serverless cloud functions comes at the cost of other step backs. First of all, runtimes without servers are stateless. As such, serverless runtime applications are responsible for ensuring careful monitoring of states in state-of-the-art program realms, i.e. when, for example, performing mathematical computations or solving optimization issues. In this respect, many conventional major cloud vendors have raced to build and deliver serverless solutions that guarantee fast delivery and suitable serverless application provisioning. Most of these projects are currently still under development with some known drawbacks and constraints but will soon be ready for use by cloud app developers. Serverless computing approach in [2] demonstrates that Instead of the conventional approach of these major cloud players, they aim to lock in their serverless clients by providing additional resources that

assist the deployment of serverless applications that are needed to activate them or manage their state, particularly when the deployed functions are stateless. Based on the results of this analysis, in all of the criteria considered, there are no other cloud computing framework that can provide more optimized performance. Many interesting alternatives still seem to exist, however.
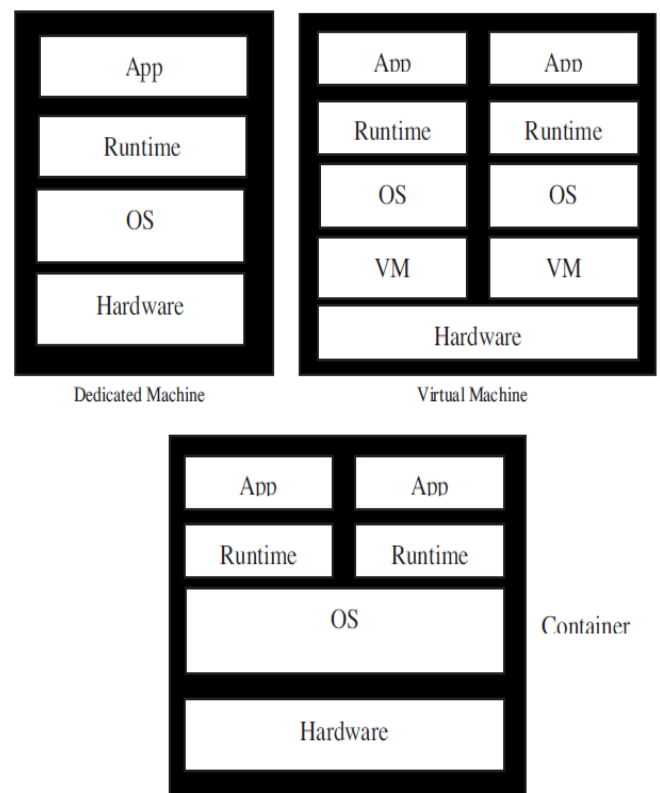

Fig -1: phases of cloud computing development

Fig. 1 Represents phases of cloud computing development. On early phases, virtualization was used as the way for consolidating program and operation, by which optimum resource efficiency and simple management is achieved. There were of the hardware among the VMs during the initial process. In the following step, a bunch of Virtual Machines (VMs) was built on a server and every VM bears an operating system a copy. Subsequently, OS level virtualization later advanced into the concept of containers like docker, these where it included. The containers are the appropriate medium to house the tools required to run a particular program. It achieved higher resource abstraction,

compared to VMs. Resource supplying of containerization is much easier than VMs.

Over the years, with the developments in virtualization technology, Web services companies begun supplying their users with specific memory and processing capacity settings for dedicated virtual machines (VMs). Since these dedicated VMs remove the paperwork pressure, have personalized Development environments and doesn't require task submissions, they have rapidly gained wider acceptance. This paper will be explaining key advantages in using serverless.

2. SERVERLESS FRAMEWORKS OVERVIEW

Serverless computing provides new possibilities for architects and developers of cloud-based applications. It offers primarily a simplistic programming model for the creation of centralized cloud-based applications, with abstract architecture.

The developers are no longer concerned with controlling the load balancers, the provisioning and the distribution of resources.as said in [6] A clear description of them should be given before beginning to analyze the serverless architectures, which will also help articulate the scope of the study. In this regard, we find a serverless architecture as a middleware program that enables the abstraction of the specificities of a certain serverless network or cloud infrastructure and thus facilitates the deployment and the delivery of single- or cross-domain serverless applications, i.e. applications that could be distributed across multiple platforms or that could leverage two separate domain infrastructures simultaneously. The dream supports solutions that draw the most out of the cloud environment and make the right decisions on what has been selected and implemented in the cloud infrastructure.
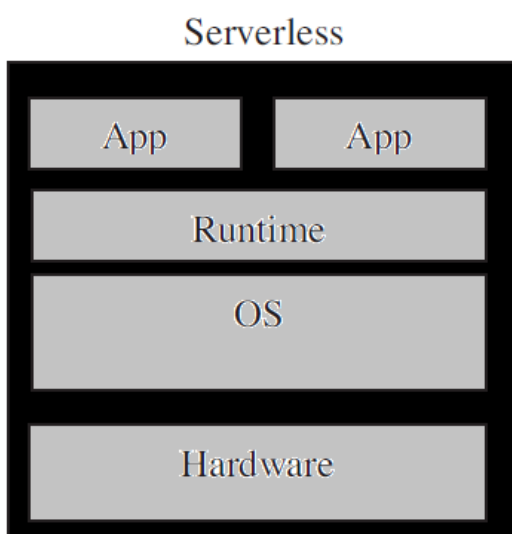

Fig -2: serverless cloud computing environment

2. MULTI CLOUD BACKEND INFRASTRUCTURE

Serverless also provide developers to deploy their cloud functions and microservices in multiple clouds environments this will remove the single point of failure issue and can also be used to distribute incoming requests to reduce network traffic.

These are the few of serverless tools provided by leading web services, amazon web services, API Gateway, EventBridge, Fargate, Lambda, Simple Notification Service, SimpleQueueService, Serverless Application Model and Serverless Application Repository. IBM Open Whisk, AzureFunctions, Google Cloud Functions and AWS Lambda. It can be combined with other serverless architectures such as Firebase and Redux Fn, as well. It adopts the modeling language Cloud Formation to configure serverless system delivery and provisioning. Finally, it allows tracking of product safety when promoting WSGI applications via the Flask[1], Django[2] and Pyramid[3] frameworks.
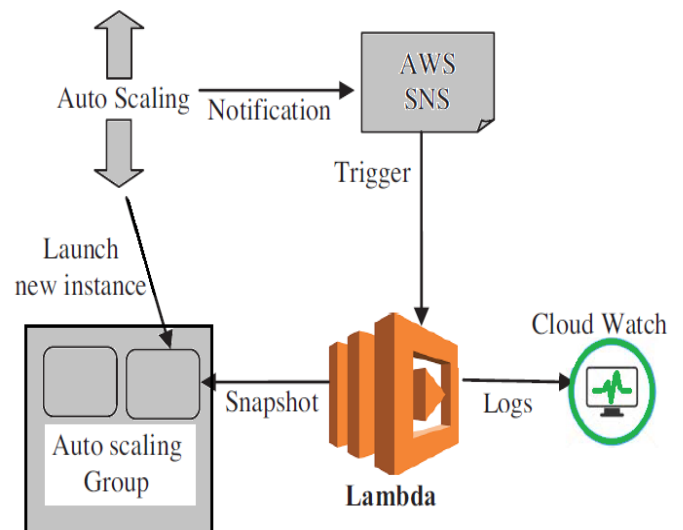

Fig -3: AWS serverless backend work flow

The serverless framework of GoogleCloud[1] allows you to develop an app any style, without thinking about the technology behind it. Install functions or programs either as source code or as containers. Develop serverless full stack apps with servers, databases, machine Learning.

Fig-4 shows different serverless tools provided by Google cloud, these micro services can be used to develop and deploy a serverless backend application.

Fig -4: Google Cloud serverless backend modules [8]

3. PERFORMANCE ENHANCEMENT CONCEIVABLE

Throughout the device design the vast number of interconnected components results in many Lambdas wide call stacks. There is potential for cold starts with each connection in the chain, additional latencies as stated in section [11] and device errors. It means that the Lambda call stack will be held small, ideally three or less Lambdas in a chain while fulfilling a time-critical request, while there are situations where instantiating an asynchronous loop will suit the needs of the consumer best. For e.g., in some situations where the execution is the product of a front-end request, such as from a website, several, concurrent calls to the back-end may be made in instances where all the data is not needed at the same time. Therefore, coping with major increases in demand may be a difficulty but when employed in conjunction with the other strategies discussed in this section it has the ability to minimize issues.

4. MIMD IN CLOUD FUNCTIONS

Developers can run multiple instruction multiple data code on cloud functions and micro services to perform tasks like video encoding , game streaming machine learning model training and more computations that can utilize parallel processing .GPU enabled Serverless computing approach in [5] demonstrates that Customers actively designing and installing GPU technology in a cloud setting are renting a GPU instance from a supplier of cloud computing resources, like Amazon Web Services and Azure cloud. Additionally, you need to load configurations and libraries on the VM after you buy an instance to support the GPU which is not an easy operation. The Cloud Functions Architecture (CFA) is becoming increasingly common in the cloud storage world, leading to the advent of Dockers utilizing containers for virtualization. Approach by integrating serverless computing platform with NVIDIA-Docker to use GPU on serverless computing system Using the proposed architecture, we've seen developers can run highly intense cloud functions and

use remote GPU to run deep learning technology. GPU-based micro-services displayed greater results than CPU-based GPUs with higher efficiency, and training neural network models with remote GPU demonstrated no difference while learning time was longer than local GPU.

5. CONCLUDING REMARKS

Serverless cloud framework envisage an abstraction architecture enabling consumers to have multi-domain hybrid apps through growing web systems and infrastructures. First, serverless architectures need to be combined with multi-cloud application management systems to enable for hybrid applications to be managed. Additionally, serverless architectures continue to be developed to become fully viable program maintenance facilities without server. Serverless computation is at the developers' point of conceptualization and business development. It is expected that the emergence of this modern Cloud infrastructure model would certainly contribute to a faster, easier, and more effective distribution of capital. Serverless computing provides efficient, event-driven interfaces with various cloud providers, easy templates for programming and implementation, and fine-grained cost control and scaling. We expect to see expanded participation from academics in serverless computing and greater accessibility from market leaders to the broader value of serverless technology.

REFERENCES

[1] Kyriakos Kritikos and Paweł Skrzypek, "A Review of Serverless Frameworks," 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion.

[2] Arda Aytekin and Mikael Johansson, "Exploiting Serverless Runtimes for Large-Scale Optimization" 2019 IEEE 12th International Conference on Cloud Computing (CLOUD).

[3] Jaewook Kim, Tae Joon Jun, Daeyoun Kang, Dohyeun Kim, Daeyoung Kim, "GPU Enabled Serverless Computing Framework",26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing Jan 2019.

[4] Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski, and P. Suter," *Serverless Computing: Current Trends and Open Problems"*. Singapore: Springer Singapore, 2017, pp. 1–20.

[5] A. R. Brodtkorb, T. R. Hagen, and M. L. Sætra, "Graphics processing unit (GPU) programming strategies and trends in GPU computing," Journal of Parallel and Distributed Computing, vol. 73, no. 1, pp. 4–13, 2013.

[6] Z. Al-Ali, "Making Serverless Computing More Serverless," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 456-459, 2018.

[7] Microsoft, "Azure Functions Documentation", Accessed on 4th May 2020. https://docs.microsoft.com/en-us/azure/azure-functions/

[8]   Google, "Google Cloud Functions Documentation". Accessed on 5th may 2020 https://cloud.google.com/functions/

[9]   Amazon Web Services, "AWS Lambda Developer Guide", Accessed on 3rd May 2020 http://docs.aws.amazon.com/lambda/latest/dg/lambda-dg.pdf

[10] E. van Eyk, L. Toader, S. Talluri, L. Versluis, A. Uță and A. Iosup, "Serverless is More: From PaaS to Present Cloud Computing," in IEEE Internet Computing, vol. 22, no. 5, pp. 8-17, 2018.

[11] Daniel Bardsley, Larry Ryan and John Howard, "Serverless Performance and Optimization Strategies", 2018 IEEE International Conference on Smart Cloud.