IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

A Survey on Function Level Scheduler to Minimize the Cloud Provider Resource Cost in Serverless Cloud Computing

Marapaka Rama Raju[1], Ramarao Gose[2], S Vijaya Laxmi[3]
Assistant Professor Department of Computer Science and Engineering[1,2,3]
Christu Jyothi Institute of Technology and Science, Jangaon, India

Abstract: *Industry started providing the services in the form of function executions and charge the clients based on the execution time and not on the machine idle time. This change the paradigm change the way in which people started looking at cloud computing. In this paper we survey on many things which are not known to the user when they are using the platforms like AWS lambda, IBM Open Whisk and Microsoft Azure Funtions, Google cloud functions as the payment model. How the runtime is brought and saved back upon the policy of the service provider in Serverless cloud computing function as a service platforms.*

Keywords: Function as a Service, Serverless Computing, Service Platforms.

I. INTRODUCTION

Serverless computing with a Function-as-a-Service (FaaS) execution model rapidly gaining popularity. FaaS model allows programmers to be focus on the core application development without overhead from server provisioning and runtime management. In the FaaS execution model, containers launched from virtual machines are utilized to run user-defined functions. It is well-known that many cloud service vendors provide serverless computing services with proprietary-library attached to a FaaS model. For example, the Lambda service by AWS, which is the first public FaaS provider, provides a well-integrated service with AWS S3 (object storage), DynamoDB (key-value storage), SNS (notification), and SQS (message queueing).

Due to its popularity, the FaaS model has been employed in the industry and academia to achieve several applications and research breakthroughs, respectively, resulting in the coverage of a wide range of topics such as opportunities and limitation through serverless computing, new applications, function run-time environment optimization , and public service comparison. Issues efficient placement of the incoming workload to minimize the provider capital expenses and dynamic auto scaling of the serverless platform to minimize the providers operating cost. Orchestration among different functions resides in different regions is not possible in serverless computing. So by transferring functions from one region to another region with user acceptance we can achieve user requirements with minimum efforts.

II. CLOUD APPLICATION INVOLVED FROM

Bare metal -> virtualized machines -> container -> towards serverless computing

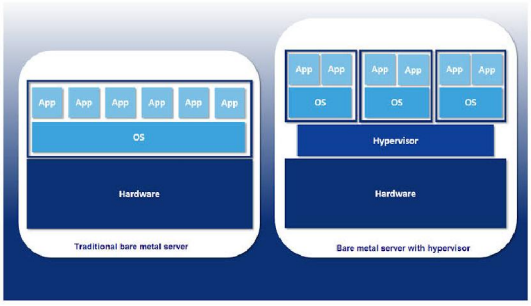

Fig.1:Bare metal

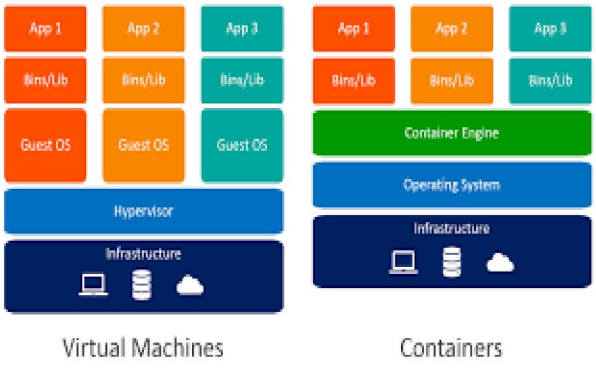

Fig.2:Virtualized machines v/s containers

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

As cloud applications have evolved from bare metal to virtualized machines, containers, towards serverless computing, the efficiency gains have enabled a wide variety of new applications.

Organizations have used containers to run long running services, batch processing at scale, control planes, Internet of Things, and artificial intelligence workloads.

A microservice-based application is one in which the core functionality has been decomposed into many small, nearly stateless units that communicate with each other through messages or events. These atomic units of work are the microservices and they are typically packaged as containers.

Container-based microservice application, attention must be placed on development processes, including cluster management and scheduling tasks, image scanning, network boundary management, service discovery, secrets management, and development lifecycle.

Note: organizations have used container to run long running services, batch processing at scale, control planes, internet of things, artificial intelligence workloads

1.1 Concept of Micro Service

To challenge of building large applications that must scale so that they can manage massive load, incrementally upgraded and remain running on platforms that with stand periodic failure

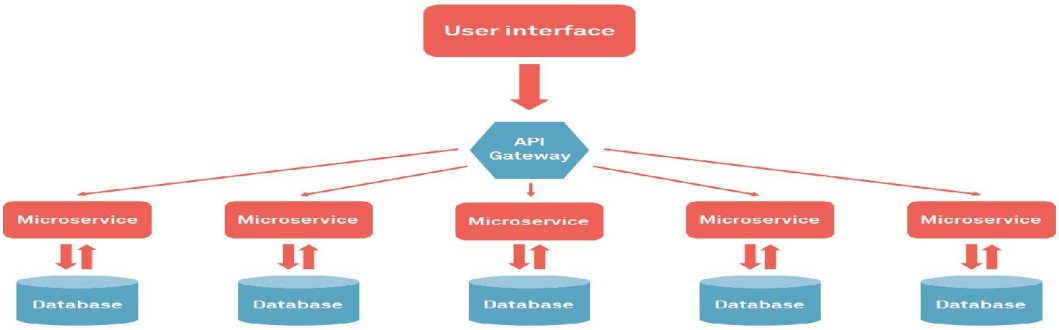

Figure 3: Container based microservice

Container based microservice application should take care of Development processes, including cluster management and scheduling tasks, image scanning, network boundary management, service discovery, secrets managements and development life cycle.

Container orchestration platform framework for integrating and managing containers at scale and multiple containers as one entity for the purpose of availability, scaling and networking

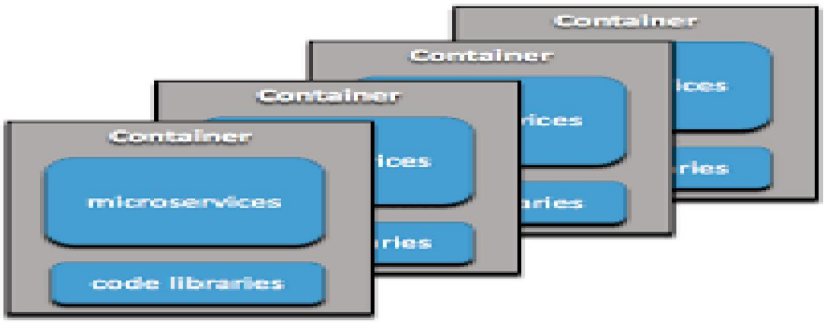

Figure 4: Container

II. CONTAINER ORCHESTRATION PLATFORM CAPABILITIES

- Cluster state management and scheduling
- Providing high availability and fault tolerance
- Ensuring security
- Simplifying networking

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

- Enabling service discovery
- Making continuous deployment possible
- Providing monitoring and governance.

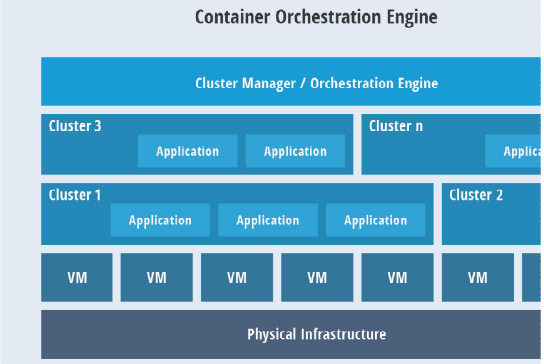

Figure 5: Container Orchestration Engine

Multiple technologies realize the concept of containers mostly used one is **Docker** Others are **LXC, OPENVZ, LINUX-VSERVER, RKT.**

2.1 Load Balancing Aims

- Optimize resource use
- Maximize throughput
- Minimize response time
- Avoid overload of any single resource

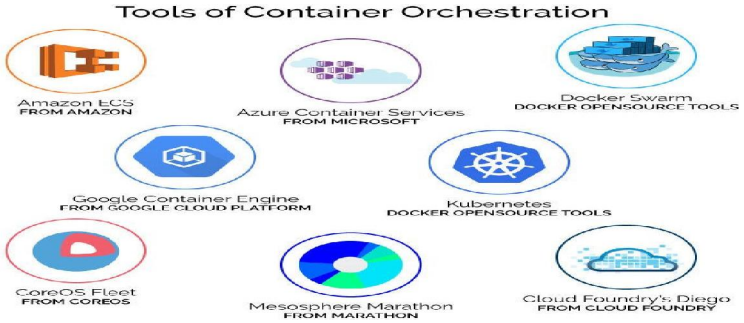

Figure 6: Container Orchestration Tools

Note: continuous delivery and deployment can be pictured as pipeline. Key stages of a deployment pipeline—commit locally, build, staging, production, feedback.

III. MONITORING

- Infrastructure on which container moving on VM
- Container activity

Application model Workload

- Long running jobs
- Batch jobs
- Cron jobs (time based jobs)

Job composition

- Single task
- Multiple independent task
- Multiple collocated tasks
- Graph of tasks

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

Serverless computing platforms provide function(s)-as-a-Service (FaaS) to end users while promising reduced hosting costs, high availability, fault tolerance, and dynamic elasticity for hosting individual functions known as microservices. Serverless Computing environments, unlike Infrastructure-as-a-Service (IaaS) cloud platforms, abstract infrastructure management including creation of virtual machines (VMs), operating system containers, and request load balancing from users.

IV. FUNCTION AS A SERVICE

FaaS model

Focus on application development rather than server provisioning and runtime management

FaaS Execution Model

Containers launched from VM are utilized to run user defined functions

Serverless computing → FaaS+ library
Ex : lamda by AWS
FaaS—S3, DynamoDB, SNS, SQS

4.1 Opportunities and Limitations

- New application
- Function runtime environment optimization
- Public service comparision
- Functionbench = microbench(measure performance of resources) + application workload(data + resource utilization).

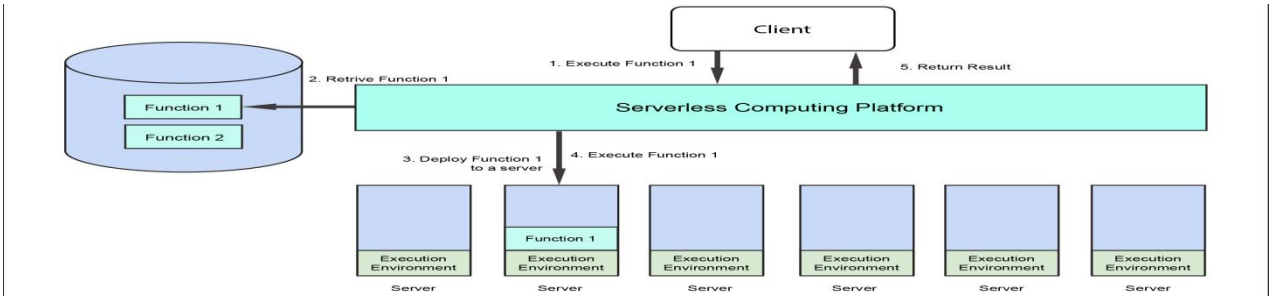

Figure 6: FaaS architecture

4.2 Evaluating Parameters

Containers and serverless serve as an excellent option over traditional servers to host their applications. It's easy, flexible and less time consuming. In both kinds of architecture, it is less complex and more flexible to build application and they are gaining a lot of popularity. But what among the two is the best way to develop and manage your application?

A. Control over the Infrastructure

Containers let the users control their own infrastructure, as opposed to serverless, which lack the control of the infrastructure. A controlled infrastructure must be implemented keeping in mind that it has to provide with the best optimization. The platform should be efficient and scalable to the developers and relevant skills are needed to be deployed. Audio-scaling will also have to be set up by the user and a complete control of the resources is available for scaling, as long as the provider allows the user to. In case of a server the user doesn't have to manage any infrastructure. There are no operating system updates to install. The provider handles all the updates for the user. It is therefore easy to manage one's own infrastructure.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

B. Tooling Support

Containers have a set of diverse and more mature set of tools, whereas serverless do not have a tools as rich as serverless does. Tools in serverless aren't as good as containers but they are improving overtime. It has tools that are not very deep and are suitable to beginners. Containers have an excellent support for tooling.

C. Deployment

Serverless has an advantage over this feature. The user only needs to deploy the code to his provider for the code to work and no additional Dockerfiles or Kubernetes configurations are required**.** Your time-to-market will be amazing.

D. Size Constraint

Container-based application can be very large and complex. According to the system that the developer wants to design, it can be as large and complex as desired. Serverless computing comes with plenty of restrictions in terms of sizes. The resulting application in this case could have an amalgamation fragmented microservices, with a high degree of uncertainty about availability and latency time for each fragment. It is difficult for monitoring tools with serverless functions, since there is no access to the function's container or container-management system.

E. Cost

The cost of containers is much higher than serverless computing. Containers need a long-running hosting location and are hence more expensive. The user has to pay for the server usage even if he is not using it at that time. In case of serverless, the user only has to pay for the time when the server is executing the action, as it runs only when it is given a trigger. The user pays for the services execute their function, so he only pays when the server is active.

F. Speed

In case of containers, it has to be ensured that all containers communicate with each other before deploying into production, every time the codebase is changed. The operating system also have to be updated all the time. All of this can slow down the development process. On the other hand, serverless computing helps in reducing the development time and thereby getting the products to the market faster.

Cloud server capacity conservation based on

- Server capacity and energy
- Cloud provider hosting infrastructure to go COLD.
- Deprovisioning containers when service demand is low freeing infrastructure to be harnessed

4.3 Hosting Implication

Infrastructure elasticity

- Load balancing
- Provisioning variation
- Infrastructure retention
- Memory reservation size

4 states of serverless infrastructure include

1. Provider COLD
2. VM COLD
3. CONTAINER COLD
4. 4 WARM

Cloud providers are responsible for creating destroying and load balancing requests across container pools
Containers can be aggregated and re-provisioned more rapidly than bulky VMs
Payment model on execution time of a function in a container it depends on SLA of service providers to survey on AWS LAMBDA, IBM OPEN WHISK, AND MICROSOFT AZURE FUNCTIONS, GOOLE CLOUD FUNCTION.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

REFERENCES

[1]. Emerging Trends, Techniques and Open Issues of Containerization: A Review
[2]. Junzo WATADA1, IEEE Senior Member; Arunava ROY2; Ruturaj KADIKAR3; Hoang PHAM4, IEEE Fellow; and Bing XU
[3]. Key Characteristics of a Container Orchestration Platform to Enable a Modern Application Asif Khan, Amazon Web ServicesIEEE CLOUD COMPUTING PUBLISHED BY THE IEEE COMPUTER SOCIET Y
[4]. 2325-6095/17/$33.00 © 2017IEEE
[5]. Serverless Computing: An Investigation of Factors Influencing Microservice Performance. Wes Lloyd, Shruti Ramesh, Swetha Chinthalapati, Lan Ly, Shrideep Pallickara
[6]. FunctionBench : A Suite of Workloads for Serverless Cloud Function Service Jeongchul Kim College of Computer Science Kookmin University, South Korea kjc5443@kookmin.ac.kr Kyungyong Lee College of Computer Science Kookmin University, South Korea leeky@kookmin.ac.kr
[7]. Estimating Cloud Application Performance Based on Micro-Benchmark Profiling Joel Scheuner Software Engineering Division Chalmers | University of Gothenburg Gothenburg, Sweden scheuner@chalmers.se Philipp Leitner Software Engineering Division Chalmers | University of Gothenburg Gothenburg, Sweden philipp.leitner@chalmers.se
[8]. Serverless Computing: An Investigation of Factors Influencing Microservice Performance Wes Lloyd, Shruti Ramesh, Swetha Chinthalapati, Lan Ly, Shrideep Pallickara
[9]. Cold Start Influencing Factors in Function as a Service Johannes Manner, Martin Endreß, Tobias Heckel and Guido Wirtz Distributed Systems Group University of Bamberg Bamberg, Germany
[10]. A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) PlatformsTheo Lynn, Pierangelo Rosati, Arnaud Lejeune