Challenges of Serverless Deployment in Edge-MEC-Cloud

Kien Nguyen, Frank Loh, Tobias Hoßfeld
University of Würzburg, Institute of Computer Science, Würzburg, Germany
Email:{firstname.lastname}@uni-wuerzburg.de

I. Introduction and Motivation

Today, virtualization has become the foundation for cloud infrastructure, in which Virtual Machine (VM) directly and indirectly involves in every service provided by cloud vendors like Amazon, Azure, and others in the trillion-dollar cloud market [1]. Recently, to adapt to the fast-changing market, virtualization has evolved from heavy and complicated VM into a more agile and resilient form called containerization. Containers are becoming increasingly popular over VMs in deployment due to their lightweight nature, faster scaling, and high fault tolerance.

In this containerization trend, a typical modern application is often composed of many containers that are highly interdependent, leading to challenges in configuration and resource management [2]. In response to this, serverless computing has been introduced to provide an abstraction layer for detailed technical operations. While not intrinsically changing the virtualization concept, serverless computing manages the lifecycle of a virtual unit in a more fine-grained and automatic way. Serverless functions leverage the scale-to-zero and event-driven concepts, creating several intermediate states in their lifecycle that provide opportunities to optimize the system [3].

While serverless paves the way toward a next-generation deployment platform, Edge Cloud emerges recently as a promising underlying architecture for accommodating applications. Edge Cloud consists of two attributes: the Edge tier provides low latency but has limited resources, while the Cloud constitutes virtually infinite resources but has an unstable network connection to end users. Integrating these tiers offers opportunities to capture the benefits of both, and Serverless dynamicity in operation can benefit from the continuous adaption of Edge Clouds to ensure low latency applications for users while also exploiting the power of the Cloud. However, this integration may face several new challenges due to, among others, extended system complexity:

Computing resources for serverless workloads can be dynamically changed between states, whether completely shutting down, entering sleep mode, or running mode [3]. This flexibility helps operators to save resources that can be allocated to other workloads or even overprovision serverless-based workloads. However, to avoid violating any Service Level Agreement (SLA), operators must ensure that the exact consumption aligns with the performance that serverless can deliver, especially when deployed over an Edge Cloud, where the environment is heterogeneous. This requires careful monitoring and management of serverless workloads.

Energy consumption of serverless units may significantly vary based on the resources provided by the Edge Cloud. The energy impacts of serverless on a heterogeneous system, such as an Edge Cloud, have yet to be fully identified. Moreover, our previous study [3] shows that serverless units consume less or near-zero energy in non-operating states. In contrast, the transition from one state to another if, for example, traffic arrives and functions must switch to operating mode consumes a significant amount of energy. This impact on the Edge Cloud is yet unknown and must be considered in detail.

Quality of Service (QoS) in general must be taken into consideration if analyzing systems and thus, also for serverless in Edge Clouds. Serverless computing operates around triggering events emitted from end-users or other serverless functions, making it highly sensitive to latency. As a result, the performance of different network types, whether wired or wireless, in an Edge Cloud can have a significant impact on the QoS of serverless functions.

Although serverless can synergize with an Edge Cloud to better utilize limited hardware resources, research on this topic is still in the early phase. To address this gap, this paper outlines some of the problems related to consumption and performance first, presents our methodology idea, and discusses the current tasks we are working on to improve energy, resource consumption, and performance of serverless deployment over Edge Cloud. Furthermore, open questions related to our proposed testbed and resource quantification methodology are provided at the end.

II. Backgrounds and Related Works

This section provides an overview of background knowledge and relevant literature necessary to understand our study.

A. Containerization and Serverless

As modern deployment demands better performance and cost reduction, applications have been evolving from virtual machines to containerization, which has a smaller footprint and higher resilience [4]. Containerization also gave birth to a new application design named microservices, in which a large application is separated into multiple connected functions residing in multiple containers. This application design can deliver better fault tolerance and lifecycle management. However, microservices, with a high number of containers,

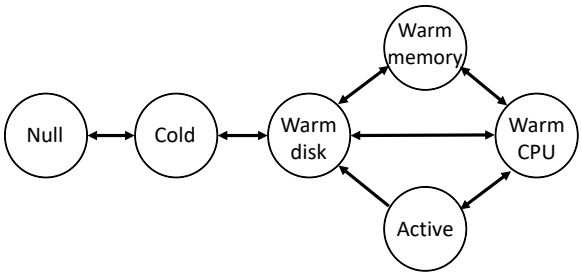

Fig. 1: Serverless function's lifecycle

pose complexity in management. To this end, serverless was introduced to entirely alleviate management and orchestration tasks, such as deployment, scaling, and load-balancing, from the developer's responsibilities. Unlike traditional containers, the lifecycle of a serverless function is more flexible, with different states that have varying consumption and performance characteristics, as demonstrated in our previous work [3] (see Figure 1). Specifically, the useful state is *Active*, the intermediate states are *Warm*, and the preliminary states are *Null* and *Cold*. Our results also showed that the state transition considerably consumes resources and significantly affects performance. Therefore, it is crucial to have a comprehensive understanding of serverless before adopting it in any environment.

B. Mobile Edge Computing, Cloud and Serverless

Edge computing is a distributed computing paradigm that aims to bring computational resources and storage capacity closer to the end-user by means of IoT devices, user equipment (UE), or even small-scale data centers. This approach improves privacy, reduces latency, and alleviates pressure on the backbone network. In the context of mobile networks, Mobile Edge Computing (MEC) is a specific edge architecture that deploys micro data centers in the proximity of base stations to minimize latency for UE workloads. However, the increasing demand for Big Data and complex Machine Learning models still require the support of cloud computing. An Edge-MEC-Cloud interplay model presents a significant opportunity to balance the advantages of the three tiers. For simplicity, we refer to this model as Edge Cloud in this paper.

Serverless in the Edge Cloud is gaining attraction recently due to its event-driven nature that fits perfectly with Edge use cases, such as Internet of Things (IoT). Wang [5] built a testbed and evaluates the performance of serverless-based smart home and agriculture IoT use cases. Their results prove that serverless consumes fewer resources than traditional deployment. Kjorveziroski [6] and Javed [7] benchmark different workloads over different Kubernetes-based serverless platforms at Edge. The results show that lightweight platforms offer better resource utilization but have limitations in device coverage and Cloud integration. To the best of our knowledge, no work in literature has comprehensively considered the consumption (CPU, RAM, etc.) and performance of serverless workloads over a real Edge Cloud environment, where computing devices and networks are heterogeneous.

III. Problem Formulation and Open Questions

There are several open research questions about serverless adoption in an Edge Cloud environment. The goal of this work is to identify these research questions and discuss a roadmap on how to solve them. In detail, we identified to following four questions:

RQ1: A well-known problem that has long existed in virtualization is the *placement* of the virtual units. While a serverless function is essentially a container, its flexible operating mode (lifecycle's state) requires thorough performance and consumption profiles when being placed over such a heterogeneous environment as an Edge Cloud.

RQ2: A Serverless function can be overprovisioned to maximize resource utilization thanks to its event-driven characteristic. On the other hand, overprovisioning may jeopardize the entire system if the system capacity does not meet the requirement during peak traffic. Due to this uncertainty in allocating resources for serverless, operators must align the amount of resources required by serverless and what the system can actually offer.

RQ3: Devices nowadays are heterogeneous, especially at the Edge. In this context, different types of computing hardware, like CPU and RAM, the chosen architecture, or data transmission and reception frequency and amount may affect differently to not both the performance and the energy and resource consumption of serverless functions. A general quantification model of the performance and cost can open the gate for operators to adopt serverless over different environments.

RQ4: The influence of Edge Cloud networks on serverless' QoS is not fully understood yet. The current deployment of Edge Clouds or MEC uses heterogeneous network types, such as Ethernet, Wifi, 4G, or 5G. As traffic from the end may reach serverless functions that are placed at nearby or remote devices, network QoS may induce tremendous influence on the functions' performance.

By investigating these questions, we model the serverless performance and consumption in the current deployment of an Edge Cloud. The solution can help operators to optimize their systems to this recent change in Edge Cloud technology.

IV. Proposed Methodology

In order to have a more detailed investigation of the series of research questions, we opt to propose a testbed and conduct measurements for different use case of serverless in industry.

A. Developed Testbed

Figure 2 illustrates the proposed testbed, which comprises devices belonging to three different roles: computing, network emulation, and monitoring. The computing devices in this testbed are Edge device, MEC, and VM Cloud, on which workloads will be placed. These devices follow the common concept that computing capability increases with distance from the edge. A separate device, known as a network emulator (NETem), is used to simulate realistic network conditions by utilizing datasets of real network traces from 4G, 5G, and WiFi networks from literature [8]. Additionally, a central control

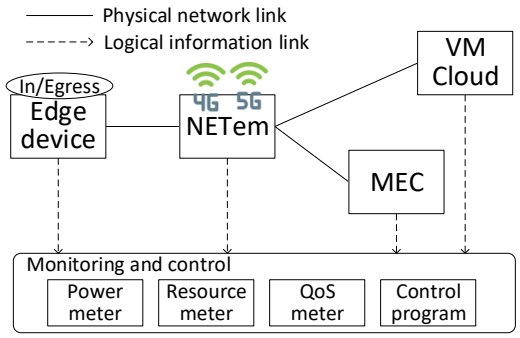

Fig. 2: Proposed testbed

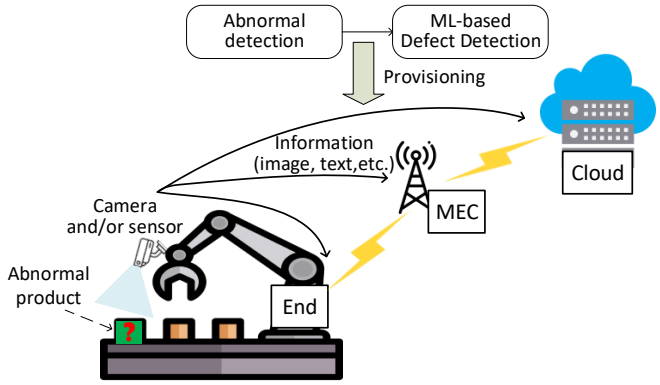

Fig. 3: Serverless-based SFC use case in product defect detection in the industry.

device is utilized to manage the testing scenario and collect measurement results.

We implement Knative [9], an open-source serverless platform for our testbed. When serverless workloads are deployed, they are placed on one of the computing nodes as described earlier. Their consumption is monitored in real-time by a hardware-based power meter and a software-based resources meter (CPU, GPU, and RAM). As serverless functions are triggered by events (i.e., service requests), we assume that these requests emit from UE (Edge device) and will be processed either locally or at one of the remote devices (MEC and VM Cloud). After being processed, the results are sent back to the UE. Thus, the UE plays both the ingress and egress role in the traffic path. The response time of the service request is measured by the QoS meter. The control program, written in Python, handles the entire process of placing the serverless functions, controlling their lifecycle, and capturing measurement results.

B. Example Use Case

To illustrate the trade-off of serverless, we implement it in an industrial IoT use case involving a robot arm that detects and discards defective products in a production line, as shown in Figure 3. Functionally, the robot uses mounted sensors and cameras to gather product's information, which is then processed by smart functions to determine if the product is impaired or normal. Computer vision techniques such as machine learning has been widely proposed for this use case [10]. However, they are often computing-intensive, which may result in high latency, causing wrong movement of the robot arm, and also drain significant energy and resources. In this matter, the adoption of an Edge Cloud and serverless may effectively decrease latency and consumption. To investigate this hypothesis, we introduce a serverless-based service chain that includes two different types of workloads:

Lightweight abnormal detection function to pre-filter abnormal products, which may or may not be defective ones. This function has a small footprint and low latency. It can be placed right at the embedded computer at the robot arm. If an abnormal object is detected with high uncertainty, its image will be sent to the next function for more precise detection.

Precise defect detection based on computer vision that will handle the job once an abnormal product has been detected by the previous function. This function often consumes huge resources and energy, and thus, may benefit from high-resource MEC and a Cloud server.

These two workloads constitute two opposite characteristics, one is fast and lightweight but less precise, and the other is slow and heavy but highly precise. By measuring and profiling them, we get an overview of resource requirements and the performance dependent on the workload.

V. Conclusion and Discussion

A testbed shown in Figure 2 is set up at the University of Würzburg, Germany. However, several challenges in testbed implementation are detected during deployment.

The testbed hosts serverless functions inside a VM to imitate realistic scenarios, however, *measurement of exact power consumed by serverless nested within the VM is a challenging task*. In addition, while serverless can be turned on/off, the underlying VM is required to operate around the clock. Thus, besides a challenging measurement process, also a *precise power consumption monitoring of serverless* must be deployed, evaluated, and validated.

Furthermore, to provide a near-realistic view of the performance and consumption of the aforementioned industrial use cases, an emulation of our serverless proposal operating in a production line in a period of time (hour, day, or month) could be conducted. In this scenario, *datasets relating to the assembly line – such as the product rate and the defect rate – should be available* as input for the emulation. Ultimately, we expect results that show how much energy and resources the serverless system saves over the course of a day or month, as well as the overall performance compared to a normal deployment without serverless.

Nevertheless, the idea of resource and energy consumption and performance quantification for serverless is interesting but also challenging. For example, different RAM and CPU architectures may yield different results in serverless performance. To achieve a general consumption model for serverless regardless of the system's resource type, *a quantification technique must be applied.*

REFERENCES

[1] MarketsandMarkets, "Cloud computing market size, share, growth drivers, opportunities and statistics," 2022, accessed May 10, 2023. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/cloud-computing-market-234.html

[2] G. Liu, B. Huang, Z. Liang, M. Qin, H. Zhou, and Z. Li, "Microservices: architecture, container, and challenges," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2020.

[3] N. Kien, F. Loh, N. Tung, D. Duong, N. H. Thanh, and T. Hossfeld, "Serverless Computing Lifecycle Model for Edge Cloud Deployments," accepted for publication in 2nd workshop on green and sustainable networking (GREENNET), 2023.

[4] A. Randal, "The ideal versus the real: Revisiting the history of virtual machines and containers," *ACM Computing Surveys (CSUR)*, vol. 53, 2020.

[5] I. Wang, E. Liri, and K. K. Ramakrishnan, "Supporting IoT Applications with Serverless Edge Clouds," in *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*, 2020.

[6] V. Kjorveziroski and S. Filiposka, "Kubernetes distributions for the edge: serverless performance evaluation," *The Journal of Supercomputing*, vol. 78, 2022.

[7] H. Javed, A. N. Toosi, and M. S. Aslanpour, "Serverless platforms on the edge: a performance analysis," in *New Frontiers in Cloud Computing and Internet of Things*. Springer, 2022.

[8] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond throughput: A 4G LTE dataset with channel and context metrics," in *Proceedings of the 9th ACM multimedia systems conference*, 2018.

[9] Knative. Serverless Containers in Kubernetes Environments. [Online]. Available: https://knative.dev/docs/

[10] Z. Kang, C. Catal, and B. Tekinerdogan, "Machine learning applications in production lines: A systematic literature review," *Computers & Industrial Engineering*, vol. 149, 2020.