Is FaaS Suitable for Edge Computing?

Runyu Jin
Arizona State University

Qirui Yang
Arizona State University

Ming Zhao
Arizona State University

Abstract

The proliferation of edge devices and the rapid growth of edge data challenged the traditional cloud computing. The high costs of cloud services and the network latency caused by data transportation outweigh the high performance cloud can provide [4]. Edge computing emerged to solve the problem. In edge computing, edge devices and edge servers are considered as important resources that can help with computation and data storage [2]; data generated on edge devices is processed locally to avoid network latency. This can greatly reduce application response time which is especially important for time-sensitive edge applications. Also, many edge devices generate data concerning user privacy. The data is more secure to be stored on local devices than on a shared cloud.

A proper abstraction is key to enabling computing across the heterogeneous resources on the edge and across the multiple tiers of resources from the edge to the cloud. In this paper, we argue that Function-as-a-Service (FaaS), an emerging cloud computing model can provide the much needed abstraction for edge computing. In FaaS, the unit of computation is a function. When a request is received, the platform starts an ephemeral execution sandbox for the function. When load increases, it quickly and dynamically increases the number of execution units. As soon as the function finishes, the sandboxes are terminated. FaaS has many benefits compared to the traditional application-based models. Sandboxing functions enhances the security and reliability by isolating misbehaved ones. Functions have faster initialization due to the smaller image sizes and function reuse. In-time reclaiming of functions improves the resource utilization. Finally, the development towards function implementation is simpler.

There are many potential benefits in delivering edge computing as FaaS. First, the fine-grained functions can accelerate the run time of applications by effectively scheduling functions across all tiers from the edge to the cloud. Also, the functions can better exploit the heterogeneous re-sources available at the edge to deliver faster responses. Moreover, functions can be reused for different applications to save function initialization time. Second, it provides additional privacy and reliability since functions are isolated to its own address space and the impact of a function is normally smaller than an application. Privacy and reliability are more of a concern on the edge due to its close relationship with the physical world. Third, FaaS helps reduce the cost of the cloud services by increasing the utilization of edge resources. Last, FaaS platform provides a nice abstraction for developers to improve the productivity.

Despite the many benefits, FaaS-based edge computing also faces several important challenges. First, how to quickly recompose the outputs of decomposed applications. The function chaining overhead may not be significant for functions deployed within an edge device. However, considering the distributed fashion of edge computing across edge devices, edge servers, and cloud, the overhead can be significant [1, 3]. Second, how to balance the performance and isolation that sandbox mechanisms provide, lightweight sandboxes such as containers have small performance overhead but cannot isolate well. Heavy weight sandboxes such as VMs provide strong isolation but bring notable performance overhead. The balance point of the two aspects needs to be figured out. Third, how to leverage the flexibility of distributed scheduling brought by smaller scheduling units and other factors including data locality, network latency and device heterogeneity. Failing to solve any of the above challenges may largely turn the benefits of applying FaaS down.

References

[1] Istemi Ekin Akkus, Ruichuan Chen, Ivica Rimac, Manuel Stein, Klaus Satzke, Andre Beck, Paarijaat Aditya, and Volker Hilt. SAND: Towards high-performance serverless computing. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 923–935, 2018.

[2] Saman Biookaghazadeh, Ming Zhao, and Fengbo Ren. Are FPGAs suitable for edge computing? In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[3] Edward Oakes, Leon Yang, Dennis Zhou, Kevin Houck, Tyler Harter, Andrea Arpaci-Dusseau, and Remzi Arpaci-Dusseau. SOCK: Rapid task provisioning with serverless-optimized containers. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 57–70, 2018.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, Oct 2016.