AuctionWhisk: Seeing Through the Fog

Example Student
Technische Universität Berlin
example.student@campus.tu-berlin.de

***Abstract*—Why does everything break**

I. Introduction

Function-as-a-service (FaaS), often also referred to as serverless computing, is a relatively new paradigm, where code is only run on demand in isolated environments. This makes it possible to bill users in a fine-grained manner, depending on metrics like number of executions or execution time. [1] In practice FaaS deployments are often optimized to either run in large datacenters of big public cloud providers [2], [3] (from here on out also referred to as Cloud) or in designated edge-distributed networks[1]. However, as researchers and industries attempt to improve performance and reduce latency further, the attention has also shifted towards alternatives.

One of those potential alternatives is Fog Computing, which we will here define as a fusion of Cloud and Edge Computing where computation resources are deployed on the network paths in between as to split up load over a more diverse infrastructure. That is, each layer between the Cloud and the Edge offers a unique trade-off between resource availability and latency. [4]

Particularly, Fog Computing offers three advantages that originate from the preceding definition. Firstly, it promises lower latency by not exclusively running customer code in few, potentially geographically distant datacenters or at the edge where resources are scarce but additionally on infrastructure at the edge and in between. [5] Secondly, it allows for enhanced privacy since user data can be processed relatively local and decentralized. [5] Lastly, by reducing the mean geographical distance between client and server, it minimizes global network bandwidth consumption. [6]

Using the distributed and heterogeneous resources of aforementioned infrastructure efficiently, creates a new set of challenges such as determining appropriate scheduling algorithms or optimal allocation of available resources. AuctionWhisk proposed a new way of solving those issues by using an auction inspired approach. In this model, developers can bid on resources in different environments with the main trade-off consisting in a generally higher price for infrastructure that offers better performance.

II. Background and Related Work

Although, there has been much research on FaaS and the paradigm has arrived in mainstream technology, with many providers creating their own offering. Be it running in large datacenters (e.g. AWS Lambda, Google Cloud Run Functions, Microsoft Azure Functions) [7] or distributed at the edge (Cloudflare Workers, Fastly Compute) [8]. At the same time, there are only few commercial offerings for FaaS on fog infrastructure [9] and research on the topic is also more scarce.

Deployments in large datacenters and at the edge are architecturally considerably different and offer their respective set of challenges. Nevertheless, one can make assumptions about resource availability in the corresponding environments. To illustrate, a deployment in the Cloud is expected to have virtually infinite, mostly reliable resources [2] but a deployment at the edge is expected to have very little, often unreliable resources [10]. However, fog computing resources are often heterogeneous, with servers closer to the Edge typically possessing lower computational capacity. Consequently, we cannot make the same assumptions about resources here, as we could above. [5]

Furthermore, FaaS is traditionally run at a single layer and although requests may be routed through multiple load balancers, proxies, etc., they are only processed at one level of the call graph. Fog computing, on the other hand, commonly uses resources as proxies and servers [2] simultaneously. This is often implemented by checking resource availability at request arrival and forwarding it if resources are insufficient for execution. [6]

In this article, we propose two contributions:

1) This is my first great idea
2) This is my second great idea

III. Distributed Systems

A. Challenges in Distributed Systems

A Distributed System does inherently not have a centralized, reliable source of truth, which creates a set of challenges that researchers have long tried to mitigate. One well-known example is the CAP theorem: [3] It states that a distributed system can not be consistent, available and partition-tolerant simultaneously. Issues of this type become more pronounced in a fog environment. There, computers are not just communicating at large scale – as in a datacenter – but additionally face increasingly unreliable infrastructure such as network or power outages.

[1]For example, Cloudflare's Worker platform is deployed on their globally distributed CDN

[2]Server is a broad term, but here it intends to describe a computing instance that serves requests

[3]Related to this is also the PACELC theorem which differentiates between a partitioned and non-partitioned system

IV. Distributed Systems

A. Challenges in Distributed Systems

V. Conclusion

References

[1] M. Shahrad, J. Balkind, and D. Wentzlaff, "Architectural implications of function-as-a-service computing," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-52. New York, NY, USA: Association for Computing Machinery, 2019, p. 1063–1075. [Online]. Available: https://doi.org/10.1145/3352460.3358296

[2] M. Brooker, M. Danilov, C. Greenwood, and P. Piwonka, "On-demand container loading in AWS lambda," in *2023 USENIX Annual Technical Conference (USENIX ATC 23)*. Boston, MA: USENIX Association, Jul. 2023, pp. 315–328. [Online]. Available: https://www.usenix.org/conference/atc23/presentation/brooker

[3] K. Djemame, M. Parker, and D. Datsev, "Open-source serverless architectures: an evaluation of apache openwhisk," in *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, 2020, pp. 329–335.

[4] O. Ascigil, A. G. Tasiopoulos, T. K. Phan, V. Sourlas, I. Psaras, and G. Pavlou, "Resource provisioning and allocation in function-as-a-service edge-clouds," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2410–2424, 2022.

[5] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, p. 27–32, Oct. 2014. [Online]. Available: https://doi.org/10.1145/2677046.2677052

[6] D. Bermbach, J. Bader, J. Hasenburg, T. Pfandzelter, and L. Thamsen, "Auctionwhisk: Using an auction-inspired approach for function placement in serverless fog platforms," *Software: Practice and Experience*, vol. 52, no. 5, pp. 1143–1169, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.3058

[7] T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha, "A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms," in *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2017, pp. 162–169.

[8] N. Ekwe-Ekwe and L. Amos, "The state of faas: An analysis of public functions-as-a-service providers," in *2024 IEEE 17th International Conference on Cloud Computing (CLOUD)*, 2024, pp. 430–438.

[9] B. Cheng, J. Fuerst, G. Solmaz, and T. Sanada, "Fog function: Serverless fog computing for data intensive iot services," in *2019 IEEE International Conference on Services Computing (SCC)*, 2019, pp. 28–35.

[10] M. Ciavotta, D. Motterlini, M. Savi, and A. Tundo, "Dfaas: Decentralized function-as-a-service for federated edge computing," in *2021 IEEE 10th International Conference on Cloud Networking (CloudNet)*, 2021, pp. 1–4.