# High-Performance Serverless for HPC and Clouds

Marcin Copik
Thesis advisor: Torsten Hoefler
Scalable Parallel Computing Lab, ETH Zürich

## 1. Motivation

Function-as-a-Service (FaaS) computing brought a fundamental shift in resource management. It allowed for new and better solutions to the problem of low resource utilization, an issue that has been known for decades in data centers. The problem persists as the frequently changing resource availability cannot be addressed entirely with the techniques employed so far, such as persistent cloud allocations and batch jobs. The elastic fine-grained tasking and largely unconstrained scheduling of FaaS create new opportunities. Still, modern serverless platforms struggle to achieve the high performance needed for the most demanding and latency-critical workloads. Furthermore, many applications cannot be “FaaSified” without non-negligible loss in performance, and the short and stateless functions must be easy to program, debug, and optimize. By solving the fundamental performance challenges of FaaS, we can build a fast and efficient programming model that brings innovative cloud techniques into HPC data centers, allowing users to benefit from pay-as-you-go billing and helping operators to decrease running costs and their environmental impact. My PhD research attempts to bridge the gap between high-performance programming and modern FaaS computing frameworks. I have been working on tailored solutions for different levels of the FaaS computing stack: from computing and network devices to high-level optimizations and efficient system designs.

## 2. Prior work

While serverless functions are one of the most promising directions for the future of cloud services, the quickly moving technology and black-box commercial platforms hinder performance analysis and reproducibility of research results. We address this challenge by designing SeBS, a benchmark for FaaS computing that systematically covers a wide spectrum of cloud resources and applications. We complement the benchmark suite with experiments analyzing and modeling overheads of serverless platforms to understand performance challenges better. The work has been presented at the ACM/IFIP Middleware 2021 [1]. We released the open-source benchmark suite, which has already been used in multiple peer-reviewed publications.
The second issue we addressed was the need for low-latency invocations that can be integrated into high-performance applications. We designed rFaaS, the first serverless platform accelerated with Remote Direct Memory Access (RDMA). Thanks to the single-digit microsecond latency, we can demonstrate function invocations up to 28 times faster over other low-latency serverless platforms. The work will be presented at the IEEE IPDPS 2023 [2].

## 3. ONGOING WORK

**Serverless Services** While serverless is gaining attention for building stateful and complex applications, the wide variety of cloud services and function triggers makes the design process particularly difficult. Their performance and reliability expectations are unclear, and related research work is often limited to FaaS with VM-deployed and non-serverless services. Instead, we evaluate the cloud offerings by designing a ZooKeeper service with high availability and consistency but a purely serverless deployment

and billing model. We build a cloud-native blueprint and design methodology for FaaS applications with price and performance models. Our results indicate areas for improvement, particularly in the latency and throughput of serverless systems in the cloud. The goal of this work is to help in selecting cloud storage, queue, and triggering services to build elastic and scalable serverless systems with predictable reliability and performance [3].

**Function Message Interface** Even though there is an increasing interest in serverless from the high-performance and machine-learning communities, communication abilities are severely limited in FaaS, and the platforms do not offer native support for parallel invocations. For serverless to become a general-purpose platform for programming parallel applications, it must offer fast and reliable collective operations. Unfortunately, state-of-the-art solutions rely on problem-specific optimizations, employ high-latency cloud storage, or use resources with persistent and non-serverless allocations. We developed a new approach that standardizes the communication interface in serverless and introduces a new communication channel based on NAT hole punching. Combined with cloud storage, both channels offer reliable messing with flexible resource management. With the example of distributed machine learning, we show that the set of application-agnostic and reliable collective operations significantly improves the performance of FaaS and makes it a more viable programming model for high-performance and distributed applications [4].

**Software Disaggregation** HPC systems suffer from low resource utilization that cannot be resolved with static and rigid batch jobs only. FaaS functions usually have a short lifetime, and they could increase the overall utilization of supercomputers by filling nodes that would otherwise be idle. We attempt to co-locate traditional batch jobs with dynamically allocated serverless functions, aiming to utilize idle CPU, memory, and GPU resources. With dynamic function placements on fine-grained resource allocations, we can offer a solution equivalent to systems with hardware disaggregation. Furthermore, we derive a set of programming principles for high-performance serverless and describe challenges that must be overcome for FaaS to become a usable runtime for HPC applications. With the co-location of HPC applications and serverless functions, we demonstrate improved throughput with moderate performance overheads [3].

[1] Marcin Copik, Grzegorz Kwasniewski, Maciej Besta, Michal Podstawski, and Torsten Hoefelr. 2021. SeBS: A Serverless Benchmark Suite for Function-as-a-Service Computing. In Proceedings of the 22nd International Middleware Conference (Middleware '21).
[2] Marcin Copik, Konstantin Taranov, Alexandru Calotoiu, Torsten Hoefler, and Marcin Chrapek. 2023. rFaaS: Enabling High Performance Serverless with RDMA and Leases (IPDPS 2023 (to appear)).
[3] Marcin Copik, Alexandru Calotoiu, Konstantin Taranov, and Torsten Hoefler. 2022. FaasKeeper: a Blueprint for Serverless Services. Technical Report. https://arxiv.org/abs/2203.14859
[4] Marcin Copik, Roman Böhringer, Alexandru Calotoiu, and Torsten Hoefler. 2022. FMI: Fast and Cheap Message Passing for Serverless Functions. Technical Report. https://spcl.inf.ethz.ch/Publications/.pdf/2022_copik_serverless_collectives_report.pdf
[5] Marcin Copik, Marcin Chrapek, Alexandru Calotoiu, and Torsten Hoefler. 2022. Software Resource Disaggregation for HPC with Serverless Computing. Technical Report. https://spcl.inf.ethz.ch/Publications/.pdf/2022_copik_serverless_hpc_report.pdf