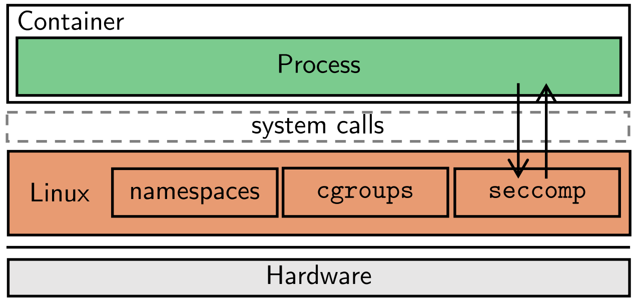
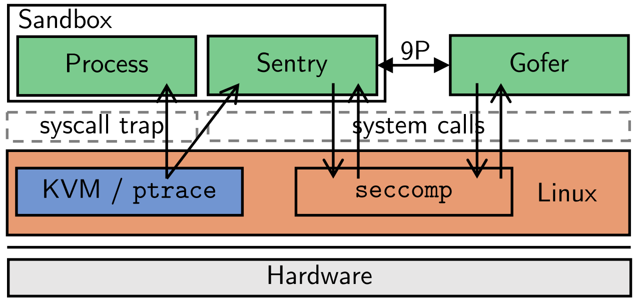
Are Unikernels Ready for Serverless on the Edge?

Felix Moebius, **Tobias Pfandzelter**, David Bermbach | Scalable Software Systems

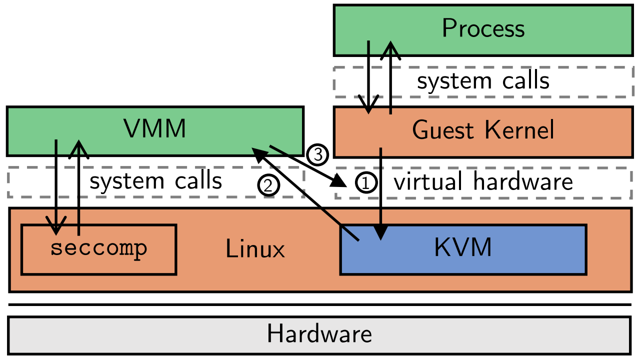
FaaS Runtimes



Containers (e.g., `runc`)



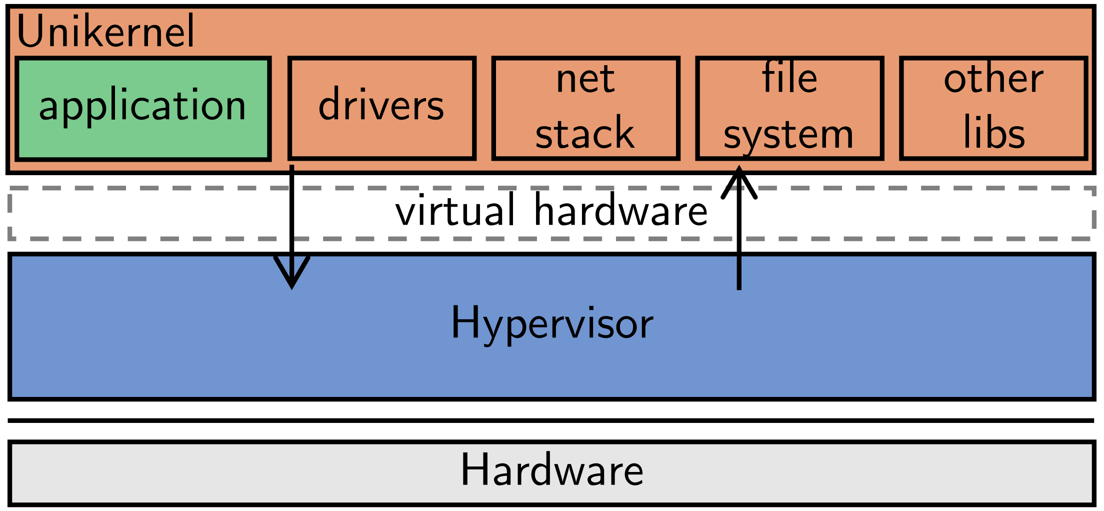
“Secure” Containers (e.g., *gVisor* [1, 2])



microVMs (e.g., *Firecracker* [3])

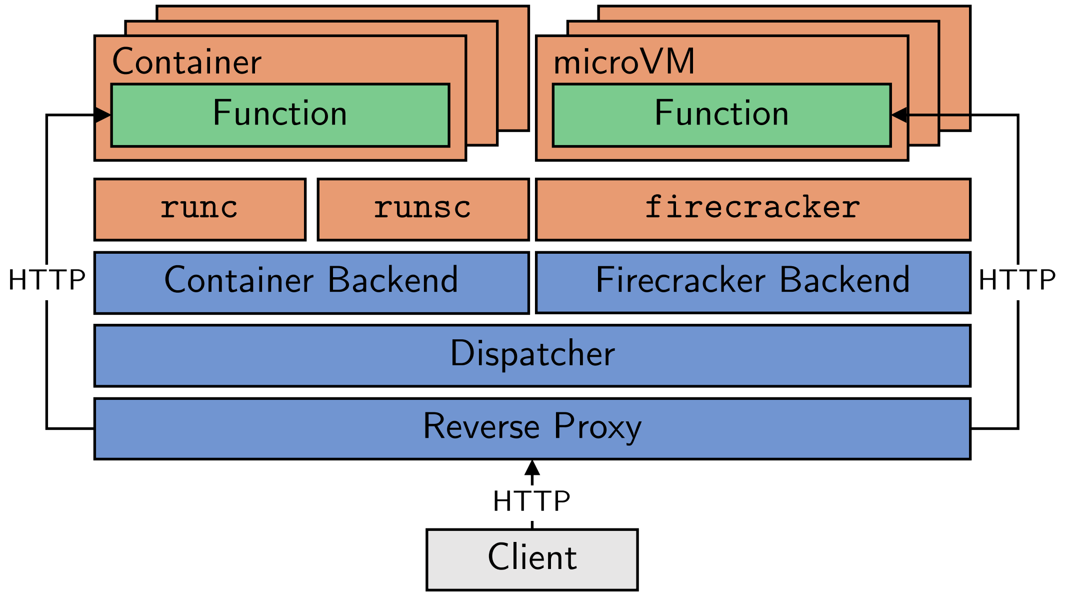
Unikernels

- Application is executed directly on top of a hypervisor, embedded into binary including OS primitives, virtual hardware drivers
- Lean, fast boot times, VM isolation, small attack surface
- POSIX-compatible unikernels (Nanos [4], OSv [5], Unikraft [6]) vs. language-based unikernels (Clive [7], runtime.js [8], Hermit [9])



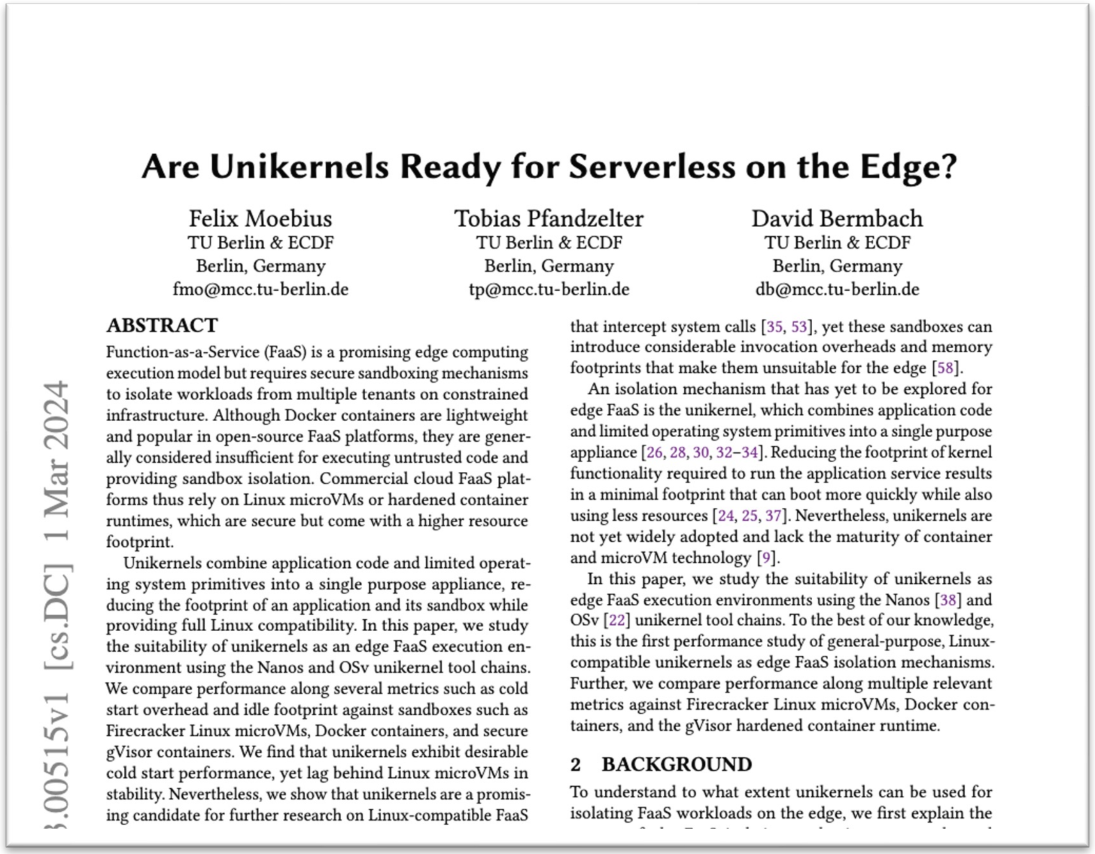
Evaluation

- Lightweight FaaS sandbox experiment harness on a single node
- Go and Node.js microbenchmarks
- Evaluating for: cold start times, memory footprint, idle resource usage, IO performance…
- Comparing OSv, Nanos, gVisor, Linux on Firecracker, Docker
- Source: https://github.com/OpenFogStack/unikernel-edge-faas/



Results

- Unikernels are promising! Great cold start performance, OK memory footprint
- But: not mature enough! Performance bugs in Nanos (network IO, memory page deduplication) and OSv (idle CPU operations)
- `runc` is hard to beat in terms of performance (but considered insecure)
- gVisor has terrible performance
- Full results in our paper (under review)



Are Unikernels Ready for Serverless on the Edge?

Felix Moebius
TU Berlin & ECDF
Berlin, Germany
fmo@mcc.tu-berlin.de

Tobias Pfandzelter
TU Berlin & ECDF
Berlin, Germany
tp@mcc.tu-berlin.de

David Bermbach
TU Berlin & ECDF
Berlin, Germany
db@mcc.tu-berlin.de

ABSTRACT

Function-as-a-Service (FaaS) is a promising edge computing execution model but requires secure sandboxing mechanisms to isolate workloads from multiple tenants on constrained infrastructure. Although Docker containers are lightweight and popular in open-source FaaS platforms, they are generally considered insufficient for executing untrusted code and providing sandbox isolation. Commercial cloud FaaS platforms thus rely on Linux microVMs or hardened container runtimes, which are secure but come with a higher resource footprint.

Unikernels combine application code and limited operating system primitives into a single purpose appliance, reducing the footprint of an application and its sandbox while providing full Linux compatibility. In this paper, we study the suitability of unikernels as an edge FaaS execution environment using the Nanos and OSv unikernel tool chains. We compare performance along several metrics such as cold start overhead and idle footprint against sandboxes such as Firecracker Linux microVMs, Docker containers, and secure gVisor containers. We find that unikernels exhibit desirable cold start performance, yet lag behind Linux microVMs in stability. Nevertheless, we show that unikernels are a promising candidate for further research on Linux-compatible FaaS

that intercept system calls [35, 53], yet these sandboxes can introduce considerable invocation overheads and memory footprints that make them unsuitable for the edge [58].

An isolation mechanism that has yet to be explored for edge FaaS is the unikernel, which combines application code and limited operating system primitives into a single purpose appliance [26, 28, 30, 32–34]. Reducing the footprint of kernel functionality required to run the application service results in a minimal footprint that can boot more quickly while also using less resources [24, 25, 37]. Nevertheless, unikernels are not yet widely adopted and lack the maturity of container and microVM technology [9].

In this paper, we study the suitability of unikernels as edge FaaS execution environments using the Nanos [38] and OSv [22] unikernel tool chains. To the best of our knowledge, this is the first performance study of general-purpose, Linux-compatible unikernels as edge FaaS isolation mechanisms. Further, we compare performance along multiple relevant metrics against Firecracker Linux microVMs, Docker containers, and the gVisor hardened container runtime.

2 BACKGROUND

To understand to what extent unikernels can be used for isolating FaaS workloads on the edge, we first explain the

https://arxiv.org/abs/2403.00515

References

1. The gVisor Authors, “What is gvisor?” https://gvisor.dev/docs/, 2024, last accessed: February 12, 2024.
2. E. Manor, “Bringing the best of serverless to you,” https://cloudplatform.googleblog.com/2018/07/bringing-the-best-of-serverless-to-you.html, Jul. 2018, last accessed: February 12, 2024.
3. A. Agache, M. Brooker, A. Iordache, A. Liguori, R. Neugebauer, P. Piwonka, and D.-M. Popa, “Firecracker: Lightweight virtualization for serverless applications,” in *Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’20)*, Feb. 2020, pp. 419–434.
4. NanoVMs Inc., “Nanos.org,” https://nanos.org/, 2023, last accessed: February 12, 2024.
5. A. Kivity, D. Laor, G. Costa, P. Enberg, N. Har’El, D. Marti, and V. Zolotarov, “Osv – optimizing the operating system for virtual machines,” in *Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC ’14)*, Jun. 2014, pp. 61–72.
6. S. Kuenzer, S. Santhanam, Y. Volchkov, F. Schmidt, F. Huici, J. Nider, M. Rapoport, and C. Lupu, “Unleashing the power of unikernels with unikraft,” in *Proceedings of the 12th ACM International Conference on Systems and Storage (SYSTOR ’19)*, May 2019, p. 195.
7. F. Ballesteros, “Clive,” https://lsub.org/clive/, 2014, last accessed: February 12, 2024.
8. S. Iefremov, A. Abreu, A. Frey, and D. Björklund, “runtimejs - lightweight javascript library operating system for the cloud,” http://runtimejs.org/, Dec. 2019, last accessed: February 12, 2024.
9. S. Lankes, J. Klimt, J. Breitbart, and S. Pickartz, “Rustyhermit: A scalable, rust-based virtual execution environment,” in *Proceedings of the ISC High Performance 2020 (ISC ’20)*, Jun. 2020, pp. 331–342.