



Contribution ID: 296

Type: **not specified**

## Developing Continuous eBPF Profiler to look Beneath the Kernel to Beyond the Clouds

*Tuesday, 14 November 2023 17:00 (30 minutes)*

At LPC 2022, we talked about experimenting with eBPF to extend the existing stack unwinding facility in the Linux kernel for interpreted languages, such as Ruby and Python, as well as runtimes emitting JITed code, like NodeJS.

While we have successfully implemented these features in parca-agent across both Arm64 and x86 architectures, there is scope for improvement in scalability and uniformity across compilers and kernel versions.

Despite having pre-existing tools to leverage underlying debug information in stripped binaries, integration with fully open-sourced dynamic profilers is sparse in the cloud-native ecosystem.

In this talk, we would like to discuss the unexpected edge cases we ran into while walking the stack for different runtimes (such as V8), different compilers (Rust, C++), different platforms (Arm64) and their interactions with DWARF and ELF formats.

We want to touch upon the challenges we faced in implementing low-level observability in the cloud and shed light on what goes into extending eBPF to fetch stacktraces for interpreted runtimes (like JIT code) or to understand GitHub CI better and various use cases in between. We will elaborate on the challenges while leveraging eBPF to bridge the gap between kernelspace and userspace while dealing with systems at scale.

One of our goals with this talk is to introduce to the community how we can use parca-dev to understand control flow and optimise systems in the Linux Kernel. We want feedback on how we can make it easier to debug large codebases such as the Linux Kernel and make dynamic continuous profiling more developer-friendly.

**Primary author:** PRIYADARSINI, Sumera (Polar Signals)

**Presenter:** PRIYADARSINI, Sumera (Polar Signals)

**Session Classification:** eBPF & Networking

**Track Classification:** eBPF & Networking Track