High-Level Interface for Asynchronous I/O using C++20 Coroutines, io_uring, and eBPF

HENDRIK SIECK, Technische Universität Hamburg, Germany

In today's operating systems, switches between user- and kernel mode account for non-negligible run-time overheads while doing system calls. Combined with expensive mitigations against Meltdown and Spectre, system calls can be slowed down by a factor of five. For reducing overheads, multi-system-call interfaces allow to batch successive system calls and submit them with a single mode switch to the kernel.

Previous work shows that io_uring can be used as an asynchronous multi-system-call interface in Linux for common userland applications. To reduce mode switches further, io_uring can be combined with eBPF to execute system-call dependent logic inside the kernel. The implementation of both approaches requires to port existing programs to new programming models, which is a time-consuming effort.

This talk presents work of a research project on a high-level programming interface that eases the usage of multi-system-calls. It uses C++20 Coroutines and an alternative lowering path for coroutine resumptions as eBPF programs to push code sections between coroutine suspension points into kernels pace.