Linux Plumbers Conference 2025

Contribution ID: **367** Type: **not specified**

Introduction to eBPF Load-Acquire and Store-Release Instructions

The growing demand for sophisticated, high-performance eBPF programs on weakly-ordered architectures like arm64 necessitates finer-grained control over memory ordering, instead of relying on the default of full memory barriers. To that end, the eBPF ISA has been expanded by two new BPF_ATOMIC instructions that provide load-acquire and store-release semantics.

This talk introduces these new instructions, which are now supported in both mainline LLVM and the upstream Linux kernel. We will begin by detailing their encoding design, and demonstrating their usage via C intrinsics through concrete examples. We will then present a deep dive into the end-to-end implementation, covering the necessary changes to the LLVM BPF backend and the Linux kernel—mainly including the verifier and Just-In-Time (JIT) compilers for arm64, x86-64, and riscv64. Finally, we will explore future directions, including a proposal to implement explicit eBPF memory barriers using nocsr kfuncs.

The goal is for every attendee to leave with a thorough understanding of these new additions to the eBPF ISA, ready to leverage them to build more featureful and performant eBPF applications.

Primary author: YE, Peilin (Google LLC)

Presenter: YE, Peilin (Google LLC)

Session Classification: eBPF Track

Track Classification: eBPF Track