Contribution ID: **4** Type: **not specified**

Crafting a Linux kernel scheduler that runs in user-space using Rust

Friday 20 September 2024 17:45 (45 minutes)

In the realm of operating systems, the heart of performance lies in the CPU scheduler: a critical component responsible for managing the execution of tasks on a system.

Traditionally, delving into CPU scheduling policies was largely confined to a small group of experienced kernel developers. Yet, there is an increasing aspiration to democratize this domain, facilitating experimentation and accessibility to a wider audience of researchers, developers, and learners.

scx_rustland is a fully functional Linux scheduler written in Rust, that runs entirely in user-space. It uses sched-ext and eBPF to channel scheduling events and communication between kernel and user-space.

One notable advantage of a user-space implementation is the availability of a large pool of debugging and profiling tools, libraries, and services. Moreover, with proper Rust's abstractions, developers can readily experiment with scheduling policies, without needing to navigate the complexities of deep kernel internals. This approach can help to lower the barrier of CPU scheduling experimentation and make this field more accessible to a wider audience of emerging kernel developers.

This scheduler is still in its proof of concept stage, however, with a well-defined API, it has the potential to evolve into an easily accessible user-space framework that allows to implement and test kernel scheduling policies.

This talk will cover the results obtained so far, highlighting the challenges faced, unsolved issues, and the trade-offs encountered along the way. The goal is to gather a feedback to pinpoint the necessary features and capabilities for defining the API for the generic scheduling subsystem.

Primary author: RIGHI, Andrea (NVIDIA)

Presenter: RIGHI, Andrea (NVIDIA)

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track