Contribution ID: **258** Type: **not specified**

Shipping sched-ext: Linux distributions roundtable

Wednesday, 18 September 2024 13:00 (20 minutes)

What are the benefits and concerns from the standpoint of distros regarding sched-ext and pluggable schedulers? Are there any roadblocks on the path to making packages available to users?
This session is a **venue for distro maintainers to share experiences, discuss issues and review plans** related to the enablement of sched-ext downstream.

A non-exhaustive list of topics to cover:

- ***QUALITY.*** Once users are empowered to write their own schedulers, they will! Distros could provide **tools so that users can evaluate their own custom schedulers.** This could take, for example, the form of a quality of service test suite. Is there interest, or prior art, in this respect?
- ***INTEGRATION.*** Custom schedulers need to be configured, loaded, and unloaded; **these operations have to be handled by a service manager such as systemd or openRC.** The repository at github.com/sched-ext/scx provides stub configuration files for service managers, but **distributions will likely need to tailor them to suit their policies and guidelines.** What's the field experience in integrating sched-ext with the surrounding environment?
- ***SUPPORT.*** No user can expect its distro vendor to provide support for a custom, out-of-tree scheduler. In practice, though, **the initial troubleshooting steps in any support request will be assessing where the problem lies:** in the distro stock packages, or in a non-standard scheduler, if that's part of the system. **Bugs, especially performance regressions, could be harder to analyze.** Is this a shared sentiment? If so, how can we mitigate the issue?
- ***TRUST.*** In the eventuality that distros begin shipping pluggable bpf schedulers, there will be **need to assure users that these programs come from a trusted source.** Software packages and loadable kernel modules are cryptographically signed; what infrastructure, and **practical experience, is there regarding signed bpf programs?**
- ***TOOLCHAIN.*** Shipping sched-ext and related utilities involves, at the very least, **packaging libbpf and a set of rust libraries.** The former is necessary to write bpf programs, and is a fast moving target. The latter constitutes the framework to write userspace schedulers, and **packaging rust has its own set of challenges as rust doesn't do dynamic linking,** and all dependencies must be fetched and compiled statically. How do distros plan to address the packaging of the sched-ext toolchain?
- ***PROCESS.*** Distros may want to get a sched-ext enabled kernel in the hands of their users, but at least initially, not do so as their primary, officially supported kernel package. **What means are available to deliver unofficial kernel packages?** Eg. Ubuntu's PPA, the Open Build Service from openSUSE, etc.
- ***DOCUMENTATION.*** If it's not documented, it doesn't exist! One of the core strengths of sched-ext is its potential to **democratize scheduler development. To fully capitalize on it, documentation must be top-class.** Some existing blog posts and conference presentations are already emerging as seminal documentation pieces, yet more is needed. Distributions are uniquely placed to create and disseminate such resources among their users. Use this session to **identify existing material, coordinate the creation of missing pieces, and select the most appropriate publication platforms.**

Primary author: GHERDOVICH, Giovanni (SUSE)

Presenter: GHERDOVICH, Giovanni (SUSE)

Session Classification: Sched-Ext: The BPF extensible scheduler class MC

Track Classification: Sched-Ext: The BPF extensible scheduler class MC