Linux Plumbers Conference 2024

Report of Contributions

Contribution ID: **23** Type: **not specified**

Android MC

The Android Micro Conference brings the upstream community and Android systems developers together to discuss issues and changes to the Android platform and their dependencies and interactions with the Linux kernel, allowing for collaboration on solutions for upstream.

Some highlights of progress made since last year's MC:

- For fw_devlink, got post-init-providers accepted into DT schema, as proposed and discussed at LPC. Additionally, as proposed at LPC, fw_devlink=rpm was made the default, so fw_devlink now enforces runtime PM ordering too.
- After discussions last year on board-id property being used to pick a DTB by the bootloader, patches for a shared solution were submitted upstream.
- Initial Pixel6 support has landed upstream, such that it can boot to console with 6.9-rc kernels.
- Having the chance to connect with the right glibc people facilitating a consensus between the bionic folks and the clang driver/lld ELF owners on an approach to mitigate the VMA (Virtual Memory Area) slab memory increase caused by the dynamic linker in devices supporting larger than 4KB page sizes.
- Discussion with the BPF ring buffer maintainer led to the event driven memory notifications from the kernel for low memory killer daemon (lmkd).

Also, listening to feedback from last year, we are planning to have slightly longer slots, so talks are not so rushed, but that also means we will have to be even more selective with topics.

Potential discussion topics for this year include:

- Device Longevity
- Power usage modeling and simulations
- Unified bootloader efforts
- The Power of Perfetto
- Using & tuning with the (soon to be) upstream Dynamic Energy Model
- Android Storage features: ublk, ureadhead, FUSE-BPF
- AVF updates&plans / pVM firmware
- More discussion on 16k pages
- RISCV updates

Primary authors: PUNDIR, Amit; TABBA, Fuad (Google); STULTZ, John (Google); YAGHMOUR, Karim (Opersys inc.); LUBA, Lukasz; SEMWAL, Sumit (Linaro)

Presenters: PUNDIR, Amit; TABBA, Fuad (Google); STULTZ, John (Google); YAGHMOUR, Karim (Opersys inc.); LUBA, Lukasz; SEMWAL, Sumit (Linaro)

Track Classification: LPC Microconference Proposals

Contribution ID: **16** Type: **not specified**

Build System MC

At Plumbers 2023 we held a build systems microconference to provide a place for people interested in build Linux Distributions to discuss the common problems they face. Based on the success of the 2023 microconference, we would like to have another microconference in Vienna. Last year, people discussed, supply chain security, kernel management, user api compatibility, and patch tracking. Each topic generated good discussions and we would like to continue the conversation this year.

The intended audience is anyone building Linux distributions. We would love participation from the Debian, Fedora, Red Hat, Nixos, Buildstream, Buildroot, OpenEmbedded, Yocto Project and other communities with a shared interest in building and maintaining Linux Distributions.

Primary authors: HOLZMAYR, Josef; BALISTER, Philip (OpenEmbedded)

Presenters: HOLZMAYR, Josef; BALISTER, Philip (OpenEmbedded)

Track Classification: LPC Microconference Proposals

Contribution ID: **34** Type: **not specified**

Complex Cameras MC

Camera hardware has undergone a rapid transformation in the last few years. It has evolved from a black box that produces frames, to a group of configurable blocks that stream and process frames, to a set of programmable blocks (like the GPUs) that need to interact with the NPU/TPUs or the GPUs.

Unfortunately, the open-source camera software stack has lagged behind, creating a bottleneck that prevents the full utilisation of the latest hardware innovations.

There have been efforts to tackle these issues openly:

- libcamera provides a high level API for configurable cameras.
- ISP (Kcam) is a kernel subsystem to schedule operations on programmable ISPs.
- SoftISP is an effort to implement complex cameras purely in software.

Until we have a proper solution some distros are using vendor-provided blobs, which is almost impossible to support in a secure way and does not allow fully open developments.

This micro conference is required to finally support Complex Cameras in Linux. More particularly, we need to answer the following questions:

- What kind of Kernel API is required for Complex Cameras?
- What level of hardware documentation do we require from vendors?
- In which kernel subsystems should Complex Cameras reside?
- How can we interact with other subsystems like NPUs/GPUs?
- What does the perfect camera software stack look like?
- How can we support proprietary use cases in an open stack?
- How can we allocate/share memory efficiently between the different subsystems?

The following actors are invited to this micro-conference:

- Linux Kernel Maintainers:
  - V4L2: Hans Verkuil, Mauro Carvalho Chehab, Sakari Ailus
  - DRM: Dave Airlie, Sima Vetter
  - Memory Management: Christoph Hellwig, Sumit Semwal, James Jones
  - Accel: Sima Vetter, Tomeu Vizoso
- Userspace camera stack
  - libcamera: Laurent Pinchart (*)
  - Android camera team: Eddy Talvala
  - ChromeOS camera team: Ricky Liang, Becker Hsieh
  - Pipewire: Wim Taymans, George Kiagiadakis
- Distros:
  - Red Hat: Hans de Goede, Maxime Ripard
  - Ubuntu: Andrea Righi
  - Debian: TBD

    - ChromeOS: Ricardo Ribalda (*), Tomasz Figa, Hidenori Kobayashi
- Vendors:
    - Intel: Jerry W Hu (*)
    - Qualcomm: Suresh Vankadara
    - MediaTek: TBD

Right now the following people marked with (*) has shown their interest in the conference and confirmed their presence.

We have the right contact for the others and we expect that they will join.

Primary authors: RIBALDA, Ricardo (Google); PINCHART, Laurent (Ideas on Board Oy)

Track Classification: LPC Microconference Proposals

Contribution ID: **20** Type: **not specified**

Compute Express Link MC

Compute Express Link is a cache coherent fabric that has been gaining momentum in the industry. Whilst the ecosystem is still catching up with CXL 3.0 and earlier features, CXL 3.1 launched just after the 2023 CXL uconf, bringing yet more challenges for the community (temporal sharing, advanced RAS features). There also has been controversy and confusion in the Linux kernel community about the state and future of CXL, regarding its usage and integration into, for example, the core memory management subsystem. Many concerns have been put to rest through proper clarification and setting of expectations.

The Compute Express Link microconference focuses on how to evolve the Linux CXL kernel driver and userspace components for support of the CXL specifications. The microconference provides a place to open the discussion, incorporate more perspectives, and grow the CXL community with a goal that the CXL Linux plumbing serves the needs of the CXL ecosystem while balancing the needs of the Linux project. Specifically, this microconference welcomes submissions detailing industry and academia use cases in order to develop usage model scenarios. Finally, it will be a good opportunity to have existing upstream CXL developers available in a forum to discuss current CXL support and to communicate areas that need additional involvement.

The earlier editions of the microconference resolved a number of open questions (CXL 1.1 RAS now upstream), and introduced new topics we expect to revisit this year (e.g. dynamic capacity / shared memory and error handling)

Suggested topics:

Ecosystem & Architectural review
Dynamic Capacity Devices - Status and next steps
Inter host shared capacity
Fabric Management - What should Linux enable (blast radius concerns)? Open source solutions?
Error handling and RAS (including OCP RAS API)
Testing and emulation
Security (ie: IDE/SPDM)
Managing vendor specificity
Virtualization of dynamic capacity.
Type 2 accelerator support - CXL 3.0+ approaches.
Coherence management of type2/3 memory (back-invalidation)
Peer2Peer (ie: Unordered IO)
Reliability, availability and serviceability (ie: Advanced Error Reporting, Isolation, Maintenance).
Hotplug (QoS throttling, policies, daxctl)
Hot remove
Documentation
Memory tiering topics that can relate to cxl (out of scope of MM/performance MCs)
Industry and academia use cases

Primary authors: MANZANARES, Adam (Samsung Electronics); WILLIAMS, Dan (Intel Open Source Technology Center); BUESO, Davidlohr (Samsung Semiconductor); CAMERON, Jonathan (Huawei Technologies R&D (UK))

Presenters: MANZANARES, Adam (Samsung Electronics); WILLIAMS, Dan (Intel Open Source

Technology Center); BUESO, Davidlohr (Samsung Semiconductor); CAMERON, Jonathan (Huawei Technologies R&D (UK))

Track Classification: LPC Microconference Proposals

Contribution ID: **10** Type: **not specified**

Confidential Computing MC

Confidential Computing microconferences in the past years brought together developers working secure execution features in hypervisors, firmware, Linux Kernel, over low-level user space up to container runtimes. A broad range of topics were discussed ranging from entablement for hardware features up to generic attestation workflows.

In the past year - guest memfd has been merged, TDX and SNP host support is getting closer to being merged. Next to go in will be support for ARM CCA and RISC V CoVE. In the meantime, there is progress being made on the Trusted I/O front.

But there is still some way to go and problems to be solved before a secure Confidential Computing stack with open source software and Linux as the hypervisor becomes a reality. The most pressing problems right now are:

- Support TEE privilege separation extensions (TDX partitioning and AMD SEV-SNP VM Privilege Levels) both on the guest and host side
- Secure IRQ delivery
- Secure VM Service Module (SVSM) support for multiple TEE architectures
- Trusted I/O software architecture
- Live migration of confidential virtual machines

Other potential problems to discuss are:

- Remote attestation architectures
- Deployment of Confidential VMs
- Linux as a CVM operating system across hypervisors
- Unification of various confidential computing API

The Confidential Computing Microconference wants to bring developers working on confidential computing together again to discuss these and other open problems.

Key attendees:

- Ashish Kalra ashish.kalra@amd.com
- Atish Patra atishp04@gmail.com
- Borislav Petkov bp@alien8.de
- Carlos Bilbao carlos.bilbao@amd.com
- Chao Peng chao.p.peng@linux.intel.com
- Dan Williams dan.j.williams@intel.com
- Daniel P. Berrangé berrange@redhat.com
- Dr. David Alan Gilbert dgilbert@redhat.com
- David Hansen dhansen@linux.intel.com
- David Kaplan David.Kaplan@amd.com
- David Rientjes rientjes@google.com

- Dhaval Giani dhaval.giani@amd.com
- Dionna Amalie Glaze dionnaglaze@google.com
- Elena Reshetova elena.reshetova@intel.com
- James Bottomley jejb@linux.ibm.com
- Jeremy Powell jeremy.powell@amd.com
- Joerg Roedel jroedel@suse.de
- Kirill A. Shutemov kirill.shutemov@linux.intel.com
- Michael Roth michael.roth@amd.com
- Mike Rapoport rppt@kernel.org
- Paolo Bonzini pbonzini@redhat.com
- Peter Gonda pgonda@google.com
- Sean Christopherson seanjc@google.com
- Tom Lendacky thomas.lendacky@amd.com

Primary authors: GIANI, Dhaval; ROEDEL, Joerg (SUSE)

Presenter: ROEDEL, Joerg (SUSE)

Track Classification: LPC Microconference Proposals

Contribution ID: **9** Type: **not specified**

Containers and checkpoint/restore MC

The Containers and Checkpoint/Restore micro-conference focuses on both userspace and kernel related work. The micro-conference targets the wider container ecosystem ideally with participants from all major container runtimes as well as init system developers.

The microconference will be discussing recent advancements in container technologies with some of the usual candidates being:

- VFS API improvements (new system calls, idmap, …)
- CGroupV2 feature parity with CGroupV1 and migration path
- Dealing with the eBPF-ification of the world
- Mediating and intercepting complex system calls
- Making user namespaces more accessible
- Verifying the integrity of containers

On the checkpoint/restore front, some of the potential topics include:

- Making CRIU work with modern Linux distributions
- Handling GPUs
- Restoring FUSE daemons
- Dealing with restartable sequences

And quite likely a variety of other container and checkpoint/restore topics as things evolve between now and the event.

Past editions of this micro-conference have been the source of many developments in the Linux kernel, including:

- PIDfds
- VFS idmap (and adding it to a slew of filesystems)
- FUSE in user namespaces
- Unprivileged overlayfs
- Time namespace
- A variety of CRIU features and checkpoint/restore kernel interfaces with the latest among them being
- Unpriviledged checkpoint/restore
- Support of rseq(2) checkpointing
- IMA/TPM attestation work

Primary authors: Mr BRAUNER, Christian; RAPOPORT, Mike (IBM); GRABER, Stéphane (Zabbly)

Presenters: Mr BRAUNER, Christian; RAPOPORT, Mike (IBM); GRABER, Stéphane (Zabbly)

Track Classification: LPC Microconference Proposals

Contribution ID: **11** Type: **not specified**

Graphics & DRM MC

The Graphics & DRM Microconference welcomes the community to discuss topics around the Linux graphics stack and the DRM subsystem, with the goal of solving long standing and complex problems together.

The MC CfP is open to all proposals related to graphics, including the following potential topics:

- Rust and DRM
- Color management and HDR
- Automated tests of GPUs and the stack
- cgroups support
- Device reset management
- DRM and IA accelerators

MC Leads:
- André Almeida
- Daniel Stone

Primary authors: ALMEIDA, André (Igalia); STONE, Daniel (Collabora)

Presenter: ALMEIDA, André (Igalia)

Track Classification: LPC Microconference Proposals

Contribution ID: **32** Type: **not specified**

Internet of Things & Embedded MC

The IoT and Embedded Micro-conference is a forum for developers to discuss all things IoT and Embedded. Topics include tools, telemetry, device drivers, protocols and standards in not only the Linux kernel but also Real-Time Operating Systems such as Zephyr.

Current Problems that require attention (stakeholders):

- IEEE 802.15.4 SubGHz improvement areas in Zephyr and Linux (Florian Grandel, Stefan Schmidt, BeagleBoard.org)
- WpanUSB driver upstreaming in the Linux kernel, companion firmware implementations (BeagleBoard.org)
- IEEE 802.15.4 Linux subsystem UWB phy support and link-layer security (Miquel Raynal, Alexander Aring, Stefan Schmidt)
- Sync device tree description of hardware between U-Boot, Linux and Zephyr (Nishanth Menon)
- Zephyr LTSv2 to LTSv3 transitions (Chris Friedt)
- CAN subsystem (Marc Kleine-Budde, Oleksij Rempel)

Since last year, there has been a number of significant updates on the topics discussed at IoT MC:

- Linux-wpan gained support for associations between devices, including PAN coordinator and disassociation handling in kernel and userspace
- For device tree sync OF_UPSTREAM has been enabled in U-Boot, this also sets the path for Zephyr sync
- TI dts code re-licensing triggered by last year IoT MC discussion
- From the Arduino Core on Zephyr project an API license discussion between LF and Arduino has been started to move forward.

One topic we'd like to cover in detail is technology or standards to help improve boot time. If there is work in this area, on Linux or Zephyr, we'd like to hear about it. Examples of boot time reduction, or of fast un-hibernate from low-power state would be welcome. Also, we're interested in discussing ideas for standards for passing pre-initialized hardware to Linux at kernel boot time.

We hope you will join us either in-person or remote for what is shaping up to be another great event full of collaboration, discussion, and interesting perspectives.

Primary authors: LÜBBE, Jan (Pengutronix); Mr SCHMIDT, Stefan; BIRD, Tim (Sony)

Presenters: LÜBBE, Jan (Pengutronix); Mr SCHMIDT, Stefan; BIRD, Tim (Sony)

Track Classification: LPC Microconference Proposals

Contribution ID: **36** Type: **not specified**

Kernel <-> Userspace/Init/System Management boundaries and APIs MC

The focus of this microconference will be on topics related to the APIs and interfaces sitting at the boundary between the kernel and init systems/system management layers, with a special attention directed towards current pain points and omissions.

For example, issues around the current way initrd are loaded and set up between the bootloader and the kernel as we move towards immutable systems, or the interfaces provided by the kernel around the mount or cgroup or pidfd APIs as consumed by systemd or other service managers, or the uevent interactions between the kernel and udev.

We expect submissions to be either open discussions or presentations that discuss new proposals/ideas, ongoing work, or problems we are/should be solving in this space. Submissions are recommended to be 15 - 45 minutes long. Please specify the format, the desired length of your submission, and how much, if any, additional time to allocate for discussion in your abstracts.

Primary authors: POETTERING, Lennart; Mr BOCCASSI, Luca (Microsoft)

Track Classification: LPC Microconference Proposals

Contribution ID: **25** Type: **not specified**

Kernel Memory Management MC

Memory management has become exciting again. Some controversial subjects which might merit discussion:

- Should we add memory policy zones?
- How far should we go to support CXL?
- How do we handle page allocation in a memdesc world?
- Should we switch the slab allocator from partial slabs to sheaves?
- Can we get rid of non-compound multi-page allocations?
- What other improvements might we see from mTHP?
- How might we make allocations guaranteed to not fail?
- Can we share the pagecache between reflinked files?
- Is there a better way to share page tables between processes than hugetlb? -

Primary authors: WILCOX, Matthew (Oracle); BABKA, Vlastimil (SUSE Labs)

Track Classification: LPC Microconference Proposals

Contribution ID: **6** Type: **not specified**

Kernel Testing & Dependability MC

The Linux Plumbers 2024 Kernel Testing & Dependability track focuses on advancing the current state of testing of the Linux Kernel and its related infrastructure. The main purpose is to improve software quality and dependability for applications that require predictability and trust. We aim to create connections between folks working on similar projects, and help individual projects make progress.

This track is intended to promote collaboration between all the communities and people interested in kernel testing and dependability. This will help move the conversation forward from where we left off at the LPC 2023 Kernel Testing & Dependability MC.

We ask that any topic discussions focus on issues/problems they are facing and possible alternatives to resolving them. The Microconference is open to all topics related to testing on Linux, not necessarily in the kernel space.

Potential testing and dependability topics:

KernelCI: Improving user experience and new web dashboard (https://github.com/kernelci/kernelci-project/discussions/28)

Growing KCIDB, integrating more sources (https://kernelci.org/docs/kcidb/)

Better sanitizers: KFENCE, improving KCSAN. (https://lwn.net/Articles/835367/)

Using Clang for better testing coverage: Now that the kernel fully supports building with clang, how can all that work be leveraged into using clang's features?

How to spread KUnit throughout the kernel?

Building and testing in-kernel Rust code.

Identify missing features that will provide assurance in safety critical systems.

Which test coverage infrastructures are most effective to provide evidence for kernel quality assurance? How should it be measured?

Explore ways to improve testing framework and tests in the kernel with a specific goal to increase traceability and code coverage.

Regression Testing for safety: Prioritize configurations and tests critical and important for quality and dependability

Transitioning to test-driven kernel release cycles for mainline and stable: How to start relying on passing tests before releasing a new version?

Explore how do SBOMs figure into dependability?

Things accomplished from last year:

Storing and Outputting Test Information: KUnit Attributes and KTAPv2 has been upstreamed.

KUnit APIs for managing devices has been upstreamed.

MC Leads:

Sasha Levin

Guillaume Tucker

Shuah Khan

Unconfirmed to-be attendees:

Sasha Levin

Kevin Hilman

Guillaume Tucker

Alice Ferrazzi

Veronika Kabatova

Nikolai Kondrashov

Antonio Terceiro
Mark Brown
Don Zickus
Enric Balletbo
Tim Orling
Gustavo Padovan
Bjorn Andersson
Milosz Wasilewski
Shuah Khan
Martin Peres
Arnd Bergmann
Remi Duraffort
Peter Zijlstra
Daniel Stone
Jan Lübbe
Dmitry Vyukov
Brendan Higgins
Greg KH
Anders Roxell
Guenter Roeck
Jesse Barnes
Kees Cook

Primary authors: TUCKER, Guillaume; LEVIN, Sasha; LEVIN, Sasha; KHAN, Shuah (The Linux Foundation); KHAN, Shuah

Presenters: TUCKER, Guillaume; LEVIN, Sasha; LEVIN, Sasha; KHAN, Shuah (The Linux Foundation); KHAN, Shuah

Track Classification: LPC Microconference Proposals

Contribution ID: **33** Type: **not specified**

KVM Microconference

KVM (Kernel-based Virtual Machine) enables the use of hardware features to improve the efficiency, performance, and security of virtual machines (VMs) created and managed by userspace. KVM was originally developed to accelerate VMs running a traditional kernel and operating system, in a world where the host kernel and userspace are part of the VM's trusted computing base (TCB).

KVM has long since expanded to cover a wide (and growing) array of use cases, e.g. sandboxing untrusted workloads, deprivileging third party code, reducing the TCB of security sensitive workloads, etc. The expectations placed on KVM have also matured accordingly, e.g. functionality that once was "good enough" no longer meets the needs and demands of KVM users.

The KVM Microconference will focus on how to evolve KVM and adjacent subsystems in order to satisfy new and upcoming requirements. Of particular interest is extending and enhancing guest_memfd, a guest-first memory API that was heavily discussed at the 2023 KVM Microconference, and merged in v6.8.

The KVM MC is expected to have strong representation from maintainers (KVM and non-KVM), hardware vendors (Intel, AMD, ARM, RISC-V, etc), cloud (AWS, Google, Oracle, etc), client (Android, ChromeOS), and open source stalwarts such as Red Hat and SUSE.

Potential Topics:
- Removing guest memory from the host kernel's direct map[1]
- Mapping guest_memfd into host userspace[2]
- Hugepage support for guest_memfd[3]
- Eliminating "struct page" for guest_memfd
- Passthrough/mediated PMU virtualization[4]
- Pagetable-based Virtual Machine (PVM)[5]
- Optimizing/hardening KVM usage of GUP[6][7]
- Live migration support for guest_memfd
- Defining KVM requirements for hardware vendors
- Utilizing "fault" injection to increase test coverage of edge cases

[1] https://lore.kernel.org/all/cc1bb8e9bc3e1ab637700a4d3defeec95b55060a.camel@amazon.com
[2] https://lore.kernel.org/all/20240222161047.402609-1-tabba@google.com
[3] https://lore.kernel.org/all/CABgObfa=DH7FySBviF63OS9sVog_wt-AqYgtUAGKqnY5Bizivw@mail.gmail.com
[4] https://lore.kernel.org/all/20240126085444.324918-1-xiong.y.zhang@linux.intel.com
[5] https://lore.kernel.org/all/20240226143630.33643-1-jiangshanlai@gmail.com
[6] https://lore.kernel.org/all/CABgObfZCay5-zaZd9mCYGMeS106L055CxsdOWWvRTUk2TPYycg@mail.gmail.com
[7] https://lore.kernel.org/all/20240320005024.3216282-1-seanjc@google.com

Primary authors: BONZINI, Paolo (Red Hat); CHRISTOPHERSON, Sean (Google)

Presenters: BONZINI, Paolo (Red Hat); CHRISTOPHERSON, Sean (Google)

Track Classification: LPC Microconference Proposals

Contribution ID: **35** Type: **not specified**

Power Management and Thermal Control MC

The Power Management and Thermal Control microconference is about all things related to saving energy and managing heat. Among other things, we care about thermal control infrastructure, CPU and device power-management mechanisms, energy models, and power capping. In particular, we are interested in improving and extending thermal control support in the Linux kernel and utilizing energy-saving features of modern hardware.

The general goal is to facilitate cross-framework and cross-platform discussions that can help improve energy-awareness and thermal control in Linux.

Since the previous iteration of this microconference, several topics covered by it have been addressed, including:

- Writable trip points support:

https://lore.kernel.org/linux-pm/6017196.lOV4Wx5bFT@kreacher/

- Limiting thermal netlink messaging to the cases when there are subscribers:

https://lore.kernel.org/linux-pm/20240223155942.60813-1-stanislaw.gruszka@linux.intel.com/

- Support for runtime-modifiable Energy Models:

https://lore.kernel.org/linux-pm/20240117095714.1524808-1-lukasz.luba@arm.com/

- Thermal control diagnostics and debug support:

https://lore.kernel.org/linux-pm/20240109094112.2871346-1-daniel.lezcano@linaro.org/
https://lore.kernel.org/linux-pm/20240109094112.2871346-2-daniel.lezcano@linaro.org/

and there is work in progress related to some of them:

- Temperature sensor aggregation support:

https://lore.kernel.org/linux-pm/20240119110842.772606-1-abailon@baylibre.com/

- Virtualized CPU performance scaling:

https://lore.kernel.org/linux-pm/20240127004321.1902477-1-davidai@google.com/

The topics that we would like to cover this year include, but are not limited to:

- Support for user-defined trip points.
- Remaining rough edges in thermal control.
- Latency-oriented CPU idle time management improvements.
- Energy-aware scheduling limitations and possible improvements.
- Support for performance QoS in non-frequency domains.
- cpufreq without frequency.
- Selecting target power state for system sleep.

and the key people we would like to participate in the session are Rafael Wysocki, Daniel Lezcano, Łukasz Łuba, Srinivas Pandruvada, Ulf Hansson, and Viresh Kumar.

Primary author: WYSOCKI, Rafael (Intel Open Source Technology Center)

Track Classification: LPC Microconference Proposals

Contribution ID: **12** Type: **not specified**

Real-time MC

The real-time community around Linux has been responsible for important changes in the kernel over the last few decades. Preemptive mode, high-resolution timers, threaded IRQs, sleeping locks, tracing, deadline scheduling, and formal tracing analysis are integral parts of the kernel rooted in real-time efforts, mostly from the PREEMPT_RT patch set. The real-time and low latency properties of Linux have enabled a series of modern use cases, like low latency network communication with NFV and the use of Linux in safety-critical systems.

This MC is the space for the community to discuss the advances of Linux in real-time and low latency features. For example (but not limited to):

- Bits left for the PREEMPT_RT merge
- Advances in the fully preemptive mode
- CPU isolation (mainly about how to make it dynamic)
- Tools for PREEMPT_RT and low latency analysis
- Tools for detecting non-optimal usages of the PREEMPT_RT
- Improvement on locks non-protected for priority inversion
- General improvements for locking
- General improvements for scheduling
- Other RT operating systems that run in parallel with Linux and the integration with Linux
- Real-time virtualization

Examples of topics that the community discussed over the last years that made progress in the RT MC:

- timerlat/osnoise tracers and RTLA
- DL server for starvation avoidance
- Proxy execution (still under discussion)
- Tracing improvements - for example, to trace IPIs

Join us to discuss the future of real-time and low-latency Linux.

Primary authors: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.); WEISBECKER, Frederic (Suse); LELLI, Juri (Red Hat); SIEWIOR, Sebastian; ROSTEDT, Steven

Presenters: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.); WEISBECKER, Frederic (Suse); LELLI, Juri (Red Hat); SIEWIOR, Sebastian

Track Classification: LPC Microconference Proposals

Contribution ID: **18** Type: **not specified**

RISC-V MC

We are excited to propose the next edition of the RISC-V micro conference to be held during the Plumbers Conference in 2024. This event has consistently served as a pivotal gathering for developers, enthusiasts, and stakeholders in the RISC-V ecosystem, especially those focused on its integration and evolution within the Linux environment. Broadly speaking anything related to both Linux and RISC-V is on topic, but discussion tend to involve the following categories:

- How to support new RISC-V ISA features in Linux mainly vendor-specific extensions.
- Discussions related to RISC-V based SOCs, which frequently include interactions with other Linux subsystems as well as core arch/riscv code.
- Coordination with distributions and toolchains on userspace-visible behavior.

Possible Topics

The actual list of topics tends to be hard to pin down this early, but here's a few topics that have been floating around the mailing lists and may be easier to resolve real-time:

- Unified discovery - What to do with this ? RVI spec which has little use in kernel land
- Control-flow integrity on RISC-V kernel.
- Hardware breakpoints / watchpoints
- OPTEE preemption model (interrupt delivery)
- riscv64 text patching w/o stop_machine()
- RISCV kernel control flow integrity
- non-PCI MSI capable devices in ACPI
- Napot
- BULTIN_DTB

Key Stakeholders

Apologies if I've missed anyone, but I've tried to list a handful of the people who frequently show up and help drive discussions at the RISC-V microconferences we've held at past Plumbers:

Regular RISC-V contributors/maintainers (I probably forgot few more)

- Palmer Atish Anup Conor Sunil Bjorn Alex Clement Andrew
- Soc stakeholders (Arnd, Conor, Heiko, Emil: There are many new SOC families showing up with RISC-V ports, and much of the new)
- We usually have attendance from a handful of the arm/arm64/ppc/mips/loongarch contributors/maintainers, as we share a lot of code and thus find many cross-arch issues. There's probably going to be even more now that we've got many shared SOC families.
- Carlos/Nick: Due to the nature of RISC-V we end up with many complicated toolchain interactions, so it's always good to have some time to discuss toolchain topics.

Accomplishments post 2023 Microconference

- All the talks at the 2023 Plumbers microconference have made at least some progress, with many of them resulting in big chunks of merged code. Specifically:

- Futile attempt to deprecate nommu after agreement in RISC-V MC :) [1]
- In fact, More support for nommu landed as a result of that discussion :[2]
- Perf feature improvement patches under discussion in lore [3]
- Good progress on supervisor software events [4] and more use cases (CFI, double trap)
- Kernel mode vector support merged[5]

[1] https://lore.kernel.org/lkml/a49546e8-6749-4458-98da-67fd37b7df18@rivosinc.com/
[2] https://lore.kernel.org/lkml/20240325164021.3229-1-jszhang@kernel.org/
[3] https://lore.kernel.org/lkml/20240217005738.3744121-1-atishp@rivosinc.com/
[4] https://lore.kernel.org/lkml/20240112111720.2975069-1-cleger@rivosinc.com/
[5] https://lore.kernel.org/all/20240115055929.4736-3-andy.chiu@sifive.com/t/#m1d48afa31c6040e4433cbf3bae2de998ae2ca

Primary authors: PATRA, ATISH (Rivos); DABBELT, Palmer (Google)

Presenters: PATRA, ATISH (Rivos); DABBELT, Palmer (Google)

Track Classification: LPC Microconference Proposals

Contribution ID: **15** Type: **not specified**

Rust MC

Rust is a systems programming language that is making great strides in becoming the next big one in the domain. Rust for Linux is the project adding support for the Rust language to the Linux kernel.

Rust has a key property that makes it very interesting as the second language in the kernel: it guarantees no undefined behavior takes place (as long as unsafe code is sound). This includes no use-after-free mistakes, no double frees, no data races, etc. It also provides other important benefits, such as improved error handling, stricter typing, sum types, pattern matching, privacy, closures, generics, etc.

This microconference intends to cover talks and discussions on both Rust for Linux as well as other non-kernel Rust topics.

Possible Rust for Linux topics:

- Rust in the kernel (e.g. status update, next steps…).
- Use cases for Rust around the kernel (e.g. subsystems, drivers, other modules…).
- Discussions on how to abstract existing subsystems safely, on API design, on coding guidelines…
- Integration with kernel systems and other infrastructure (e.g. build system, documentation, testing and CIs, maintenance, unstable features, architecture support, stable/LTS releases, Rust versioning, third-party crates…).
- Updates on its subprojects (e.g. klint, pinned-init…).

Possible Rust topics:

- Language and standard library (e.g. upcoming features, stabilization of the remaining features the kernel needs, memory model…).
- Compilers and codegen (e.g. rustc improvements, LLVM and Rust, rustc_codegen_gcc, gccrs…).
- Other tooling and new ideas (Coccinelle for Rust, bindgen, Compiler Explorer, Cargo, Clippy, Miri…).
- Educational material.
- Any other Rust topic within the Linux ecosystem.

Last year was the second edition of the Rust MC and the focus was on presenting and discussing the ongoing efforts by different parties that are using and upstreaming new Rust abstractions and drivers (Using Rust in the binder driver, Block Layer Rust API, Rust in V4L2: a status report and Converting a DRM driver to Rust) as well as those that are improving the ergonomics and tooling around it (Klint: Compile-time Detection of Atomic Context Violations for Kernel Rust Code, pin-init: Solving Address Stability in Rust and Coccinelle for Rust).

Since the MC last year, there has been continued progress from users (e.g. the Android Binder Driver getting closer to upstreaming all its dependencies) as well as new project announcements (e.g. Nova), the first Rust reference driver merged together with its abstractions (the Rust Asix PHY driver), Rust support for new architectures mainlined (LoongArch and arm64)…

Primary authors: OJEDA, Miguel; ALMEIDA FILHO, Wedson

Presenters: OJEDA, Miguel; ALMEIDA FILHO, Wedson

Track Classification: LPC Microconference Proposals

Contribution ID: **22** Type: **not specified**

Safe Systems with Linux

As Linux is increasingly deployed in systems with varying criticality constraints, distro providers are being expected to ensure that security fixes in their offerings do not introduce regressions for customer products that have safety considerations. The key question arises: How can they establish consistent linkage between code, tests, and the requirements that the code satisfies?

This MC addresses critical challenges in requirements tracking, documentation, testing, and artifact sharing within the Linux kernel ecosystem. Functionality has historically been added to the kernel with requirements explained in the email justifications for adding, but not formalized as "requirements"in the kernel documentation. While tests are contributed for the code, the underlying requirement that the tests satisfies is likewise not documented in a consistent manner.

Potential topics to be discussed:

- where should requirements that the kernel code and testing satisfies be tracked? In kernel documentation, in the code, etc.
- incorporating requirement linkage to the kernel code and tests that minimizes the impact to kernel maintainers and contributors.
- examples and strategies for enhancing documentation quality and level of detail within the Linux kernel so that effective safety analysis can be performed for products. Some starting points have been started [1], but what else is needed.
- connecting artifacts in a shareable format: how to effectively link and share testing, documentation, bug reports, and CVE information across multiple projects, infrastructures, and contribution processes.
- traceability and change identification in requirements to keep in sync with the evolving kernel code functionality and security fixes.
- increasing code test coverage of the Linux kernel to satisfy the higher safety assurance considerations. There's been some recent studies conducted by Boeing and the University of Illinois on various coverage types, that should be considered.
- requirements introduced by the Cyber Resilience Act in the EU [2] on product manufacturers might have on the Linux Kernel development process and documentation.
- improving systematic error responses when using Linux as well as runtime verification monitoring.

Last year, we had several talks on the need for safe systems [3][4] in various domains with Linux as a component (with varying safety criticality levels). This miniconference is targetted at getting those interested together, and working up a framework for collecting relevant evidence and sharing it.

MC Leads:
Kate Stewart, Philipp Ahmann

Potential Participants (not confirmed yet):

Syed Mohammed Khasim
Jonathan Corbet
Shuah Khan
Greg Kroah-Hartman
Chuck Wobler
Daniel Bristot de Oliveira

Thomas Gleixner
Gabrielle Paoloni
Olivier Charrier
Jiri Kosina
Joachim Werner
Paul Albertela
Bertrand Boisseau

[1] https://docs.kernel.org/admin-guide/workload-tracing.html
[2] https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act
[3] https://lpc.events/event/17/contributions/1499/
[4] https://lpc.events/event/17/contributions/1518/

Primary authors: STEWART, Kate (Linux Foundation); AHMANN, Philipp (Robert Bosch GmbH)

Presenters: STEWART, Kate (Linux Foundation); AHMANN, Philipp (Robert Bosch GmbH)

Track Classification: LPC Microconference Proposals

Contribution ID: **13** Type: **not specified**

Sched MC

The scheduler is at the core of Linux performance. With different topologies and workloads, giving the user the best experience possible is challenging, from low latency to high throughput and from small power-constrained devices to HPC.
The following accomplishments have been made as a result of last year's micro-conference:

- Progress on proxy execution https://lore.kernel.org/lkml/20240224001153.2584030-1-jstultz@google.com/
- Progress on system pressure https://lore.kernel.org/lkml/170073688055.398.12687414937207369825.tip-bot2@tip-bot2/ https://lore.kernel.org/lkml/20240220145947.1107937-1-vincent.guittot@linaro.org/
- Progress in the DL server
- The EEVDF scheduler and improvements in latency nice
- Progress on adding tracepoints for IPI

Ideas of topics to be discussed include (but are not limited to):

- Improve responsiveness for CFS tasks
- The improvements on the EEVDF scheduler proposal
- Impact of new topology on CFS, including hybrid or heterogeneous system
- Taking into account task profile with IPCC or uclamp
- Locking improvements –e.g., proxy execution
- Improvements on SCHED_DEADLINE
- Tooling for debugging scheduling

It is fine if you have a new topic not on the list. People are encouraged to submit any topic related to real-time and scheduling.
The goal is to discuss open problems, preferably with patch set submissions already in discussion on LKML. The presentations are concise, and the central portion of the time should be given to the debate –thus, the importance of having an open and relevant problem with people in the community engaged in the solution.

Primary authors: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.); LELLI, Juri (Red Hat); ROSTEDT, Steven; GUITTOT, Vincent (Linaro)

Presenters: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.); LELLI, Juri (Red Hat); ROSTEDT, Steven; GUITTOT, Vincent (Linaro)

Track Classification: LPC Microconference Proposals

Contribution ID: **3** Type: **not specified**

Sched-Ext: The BPF extensible scheduler class

Overview

sched_ext is a Linux kernel feature which enables implementing host-wide, safe kernel thread schedulers in BPF, and dynamically loading them at runtime. sched_ext enables safe and rapid iterations of scheduler implementations, thus radically widening the scope of scheduling strategies that can be experimented with and deployed, even in massive and complex production environments.

sched_ext was first sent to the upstream list as an RFC patch set back in November 2022. Since then, the project has evolved a great deal, both technically, as well as in the significant growth of the community of sched_ext users and contributors.

Discussions

Note that there is as-yet no particular order to the following discussions, with the exception that we'd ideally like for David to present first to open the session.

David Vernet (void@manifault.com): The current status and future potential of sched_ext

- Opening the MC by giving the room an update on the latest status of sched_ext, and getting discussions started with respect to future directions we could take, as well as how the community is growing.
- Depending on the outcomes of discussions at LSFMM, may discuss some current open questions as well; especially related to componentization. These may also be discussed by others, such as Andrea as described below.

Andrea Righi (andrea.righi@canonical.com): Crafting a Linux kernel scheduler that runs in user-space using Rust

- Necessary to coordinate how to properly componentize the scx repo to accommodate user space rust schedulers, rust hybrid schedulers, and C hybrid schedulers.
- The scheduler is interactive, so it will be an opportunity to discuss techniques that are appropriate for other schedulers such as scx_lavd being experimented with to optimize the SteamDeck.

Kevin Becker (kevin.becker@canonical.com): Enabling sched-ext in the Ubuntu real-time kernel

- sched_ext is thus far not thoroughly used or tested in PREEMPT_RT kernels. We need to get distro maintainers in the same room as sched_ext developers to discuss implications and plan for support.

Giovanni Gherdovich (ggherdovich@suse.cz): Enabling sched-ext in SuSE

- SuSE is interested in exploring the use of sched_ext, and this would give SuSE distro maintainers an opportunity to discuss plans, use cases, and road blocks. This is especially useful given that other distro maintainers will be in the room.

Piotr Górski (lucjan.lucjanov@gmail.com) and Peter Jung (admin@ptr1337.dev): Deploying and managing sched_ext schedulers in CachyOS

- CachyOS is a somewhat new, but powerful distribution that focuses on performance and scheduling. They were the first distribution to adopt sched_ext, and want to share their plans for building a management layer and coordinating with other distros to avoid duplicated efforts.

Andrea Righi (andrea.righi@canonical.com): Distro-centric meeting session: solving generic toolchain and other dependency issues for deploying sched_ext

- Leaving aside time to discuss general distribution problems with deploying sched_ext, such as toolchain dependencies, backwards compatibility challenges, etc. There may be some overlap with other distro-centric discussions, but this discussion can be a way to ground them and develop action items that will apply to all distros.

Changwoo Min (changwoo@igalia.com): Using sched_ext to improve frame rates on the SteamDeck

- Igalia has been working to leverage sched_ext support on the SteamDeck, and has been seeing encouraging results. This discussion will allow us to discuss techniques that do and don't work for interactive workload testing, optimizing for gaming workloads, and how to validate interactive scheduler changes.

Himadri Chhaya-Shailesh (himadrispandya@gmail.com): Adaptive workload parallelization on oversubscribed hosts using sched_ext

- A potential use case of sched_ext is leveraging it for optimizing virtualization; both with paravirt, or with no guest awareness. Himadri has been experimenting with this, and can discuss pain points she's encountered, and roadmap items that will be needed to further enable this effort.

Logistics notes

- This is the first time we've proposed a sched_ext microconference, so we have no results and accomplishments to discuss from prior meetings.
- We've discussed the possibility of combining with either the sched or sched-RT microconferences with the authors of those MCs, and we all agree that there are already too many topics to combine the MCs. That said, if at all possible, it would be great if we could schedule the scheduler-related MCs such that they don't overlap, so that we can attend sessions in the other MCs when possible.
- We would ideally like to allow around 30 minutes per presentation, if at all possible, so around 4 hours + break times. If this is too long, let us know and we can accommodate.

Primary author: VERNET, David (Meta)

Co-authors: RIGHI, Andrea (Canonical); MIN, Changwoo (Igalia); Mr GHERDOVICH, Giovanni (SuSE); Mrs CHHAYA-SHAILESH, Himadri (Inria-Paris); Mr BECKER, Kevin (Canonical); Mr JUNG, Peter (CachyOS); Mr GÓRSKI, Piotr (CachyOS)

Presenters: RIGHI, Andrea (Canonical); MIN, Changwoo (Igalia); VERNET, David (Meta); Mr GHERDOVICH, Giovanni (SuSE); Mrs CHHAYA-SHAILESH, Himadri (Inria-Paris); Mr BECKER, Kevin (Canonical); Mr JUNG, Peter (CachyOS); Mr GÓRSKI, Piotr (CachyOS)

Track Classification: LPC Microconference Proposals

Contribution ID: **30** Type: **not specified**

System Boot and Security MC

The System Boot and Security Microconference has been a critical platform for enthusiasts and professionals working on firmware, bootloaders, system boot, and security. This year, the conference focuses on the challenges that arise when upstreaming boot process improvements to Linux kernel. Cryptography, which is an ever evolving field, poses unique demands on secure elements and TPMs as newer algorithms are introduced and older ones are deprecated. Additionally, new hardware architectures with DRTM capabilities, such as ARM's D-RTM specification, and the increased use of fTPMs in innovative applications, add to the complexity of the task. This is the fifth time in the last six years that the conference is being held.

Trusted Platform Modules (TPMs) for encrypting disks have become widespread across various distributions. This highlights the vital role that TPMs play in ensuring platform security. As the field of confidential computing continues to grow, virtual machine firmware must evolve to meet end-users demands, and Linux would have to leverage exposed capabilities to provide relevant security properties. Mechanisms like UEFI Secure Boot that were once limited to OEMs now empower end-users. The System Boot and Security Microconference aims to address these challenges collaboratively and transparently. We welcome talks on the following technologies that can help achieve this goal.

- TPMs, HSMs, secure elements
- Roots of Trust: SRTM and DRTM
- Intel TXT, SGX, TDX
- AMD SKINIT, SEV
- ARM DRTM
- Growing Attestation ecosystem,
- IMA
- TrenchBoot, tboot
- TianoCore EDK II (UEFI), SeaBIOS, coreboot, U-Boot, LinuxBoot, hostboot
- Measured Boot, Verified Boot, UEFI Secure Boot, UEFI Secure Boot Advanced Targeting (SBAT)
- shim
- boot loaders: GRUB2, systemd-boot/sd-boot, network boot, PXE, iPXE,
- UKI
- u-root
- OpenBMC, u-bmc
- legal, organizational, and other similar issues relevant to people interested in system boot and security.

If you want to participate in this microconference and have ideas to share, please use the Call for Proposals (CFP) process. Your submissions should focus on new advancements, innovations, and solutions related to firmware, bootloader, and operating system development. It's essential to explain clearly what will be discussed, why and what outcomes you expect from the discussion.

P.S. We can only make it on September 18 because of conflict with other event.

Primary authors: KIPER, Daniel; KRÓL, Piotr (3mdeb Embedded Systems Consulting)

Co-author: GARRETT, Matthew (Google)

Track Classification: LPC Microconference Proposals

Contribution ID: **27** Type: **not specified**

Tracing / Perf events Microconference

The Linux kernel has grown in complexity over the years. Complete understanding of how it works via code inspection has become virtually impossible. Today, tracing is used to follow the kernel as it performs its complex tasks. Tracing is used today for much more than simply debugging. Its framework has become the way for other parts of the Linux kernel to enhance and even make possible new features. Live kernel patching is based on the infrastructure of function tracing, as well as BPF. It is now even possible to model the behavior and correctness of the system via runtime verification which attaches to trace points. There is still much more that is happening in this space, and this microconference will be the forum to explore current and new ideas.

This year, focus will also be on perf events:

Perf events are a mechanism for presenting performance counters and software events that occur running Linux to users. There are kernel and userland components to perf events, with the kernel presenting or extending APIs and the perf tool presenting this to users

Results and accomplishments from the last time (2023):

- Masami's work on accessing function entry data from function *return* probes (kprobe and fprobe) was merged for v6.9.
- eventfs is now dynamically created and fully working following *robust* discussions with Linus.
- Work on sframes was paused due to other priorities but is still a topic of interest.
- Discussions on integrating User events with libside are ongoing.
- User events added multi-format events.

Topics for this year:

- Feedback about the tracing/perf subsystems overall (e.g. how can people help the maintainers).
- Reboot persistent in-memory tracing buffers, this would make ftrace a very powerful debugging and performance analysis tool for kexec and could also be used for post crash debugging.
- Dynamic change of ftrace events to improve symbolic printing.
- Userspace instrumentation (libside), including discussion of its impacts on the User events ABI.
- Collect state dump events from kernel drivers (e.g. dump wifi interfaces configuration at a given point in time through trace buffers).
- Current work implementing performance monitoring in the kernel,
- User land profiling and analysis tools using the perf event API,
- Improving the kernel perf event and PMU APIs,
- Interaction between perf events and subsystems like cgroups, kvm, drm, bpf, etc.,
- Improving the perf tool and its interfaces in particular w.r.t. to scalability of the tool,
- Implementation of new perf features and tools using eBPF, like the ones in tools/perf/util/bpf_skel/.
- Further use of type information to augment the perf tools,

- Novel uses of perf events for debugging and correctness,
- New challenges in performance monitoring for the Linux kernel,
- Regression testing/CI integration for the perf kernel infrastructure and tools,
- Improving documentation,
- Security aspects of using tracing/perf tools,

Key attendees:

- Steven Rostedt
- Masami Hiramatsu
- Mathieu Desnoyers
- Alexei Starovoitov
- Peter Zijlstra
- Mark Rutland
- Beau Belgrave
- Daniel Bristot de Oliveira
- Florent Revest
- Jiri Olsa
- Tom Zanussi
- Alexander Graf
- Johannes Berg
- Arnaldo Carvalho de Melo
- Ian Rogers
- Namhyung Kim
- Stephane Eranian

Primary authors: CARVALHO DE MELO, Arnaldo (Red Hat Inc.); ROGERS, Ian (Google); DESNOYERS, Mathieu (EfficiOS Inc.); JEANSON, Michael (EfficiOS); KIM, Namhyung (Google); ROSTEDT, Steven

Track Classification: LPC Microconference Proposals

Contribution ID: **29** Type: **not specified**

VFIO/IOMMU/PCI MC

The PCI interconnect specification, the devices that implement it, and the system IOMMUs that provide memory and access control to them are nowadays a de-facto standard for connecting high-speed components, incorporating more and more features such as:

- Address Translation Service (ATS)/Page Request Interface (PRI)
- Single-root I/O Virtualization (SR-IOV)/Process Address Space ID (PASID)
- Shared Virtual Addressing (SVA)
- Remote Direct Memory Access (RDMA)
- Peer-to-Peer DMA (P2PDMA)
- Cache Coherent Interconnect for Accelerators (CCIX)
- Compute Express Link (CXL)/Data Object Exchange (DOE)
- Component Measurement and Authentication (CMA)
- Integrity and Data Encryption (IDE)
- Security Protocol and Data Model (SPDM)
- Gen-Z

These features are aimed at high-performance systems, server and desktop computing, embedded and SoC platforms, virtualisation, and ubiquitous IoT devices.

The kernel code that enables these new system features focuses on coordination between the PCI devices, the IOMMUs they are connected to, and the VFIO layer used to manage them (for userspace access and device passthrough) with related kernel interfaces and userspace APIs to be designed in-sync and in a clean way for all three sub-systems.

The VFIO/IOMMU/PCI MC focuses on the kernel code that enables these new system features, often requiring coordination between the VFIO, IOMMU and PCI sub-systems.

Following the success of LPC 2017, 2019, 2020, 2021, 2022, and 2023 VFIO/IOMMU/PCI MC, the Linux Plumbers Conference 2024 VFIO/IOMMU/PCI track will focus on promoting discussions on the PCI core and current kernel patches aimed at VFIO/IOMMU/PCI subsystems. Specific sessions will target discussions requiring coordination between the three subsystems.

See the following video recordings from 2023: LPC 2023 - VFIO/IOMMU/PCI MC.

Older recordings can be accessed through our official YouTube channel at @linux-pci and the archived LPC 2017 VFIO/IOMMU/PCI MC web page at Linux Plumbers Conference 2017, where the audio recordings from the MC track and links to presentation materials are available.

The tentative schedule will provide an update on the current state of VFIO/IOMMU/PCI kernel sub-systems, followed by a discussion of current issues in the proposed topics.

The following was a result of last year's successful Linux Plumbers MC:

- The first version of work on improving the IRQ throughput using coalesced interrupt delivery with MSI has been sent for review to be included in the mainline kernel
- The work surrounding support for /dev/iommufd continues with the baseline VFIO support replacing the "Type 1", has been merged into the mainline kernel, and discussions around introducing accelerated viommu to KVM are in progress. Both Intel and AMD are working on supporting iommufd in their drivers

- Changes focused on IOMMU observability and overhead are currently in review to be included in the mainline kernel
- The initial support for generating DT nodes for discovered PCI devices has been merged into the mainline kernel. Several patches followed with various fixes since then
- Following a discussion on cleaning up the PCI Endpoint sub-system, a series has been proposed to move to the genalloc framework, replacing a custom allocator code within the endpoint sub-system

Tentative topics that are under consideration for this year include (but are not limited to):

- PCI
  - Cache Coherent Interconnect for Accelerators (CCIX)/Compute Express Link (CXL) expansion memory and accelerators management
  - Data Object Exchange (DOE)
  - Integrity and Data Encryption (IDE)
  - Component Measurement and Authentication (CMA)
  - Security Protocol and Data Model (SPDM)
  - I/O Address Space ID Allocator (IOASID)
  - INTX/MSI IRQ domain consolidation
  - Gen-Z interconnect fabric
  - ARM64 architecture and hardware
  - PCI native host controllers/endpoints drivers' current challenges and improvements (e.g., state of PCI quirks, etc.)
  - PCI error handling and management, e.g., Advanced Error Reporting (AER), Downstream Port Containment (DPC), ACPI Platform Error Interface (APEI) and Error Disconnect Recovery (EDR)
  - Power management and devices supporting Active-state Power Management (ASPM)
  - Peer-to-Peer DMA (P2PDMA)
  - Resources claiming/assignment consolidation
  - Probing of native PCIe controllers and general reset implementation
  - Prefetchable vs non-prefetchable BAR address mappings
  - Untrusted/external devices management
  - DMA ownership models
  - Thunderbolt, DMA, RDMA and USB4 security
- VFIO
  - Write-combine on non-x86 architectures
  - I/O Page Fault (IOPF) for passthrough devices
  - Shared Virtual Addressing (SVA) interface
  - Single-root I/O Virtualization(SRIOV)/Process Address Space ID (PASID) integration
  - PASID in SRIOV virtual functions
  - Device assignment/sub-assignment
- IOMMU
  - /dev/iommufd development
  - IOMMU virtualisation
  - IOMMU drivers SVA interface
  - DMA-API layer interactions and the move towards generic dma-ops for IOMMU drivers
  - Possible IOMMU core changes (e.g., better integration with the device-driver core, etc.)

If you are interested in participating in this MC and have topics to propose, please use the Call for Proposals (CfP) process. More topics might be added based on CfP for this MC.

Otherwise, join us in discussing how to help Linux keep up with the new features added to the PCI interconnect specification. We hope to see you there!

Key Attendees:

- Alex Williamson
- Arnd Bergmann
- Ashok Raj
- Benjamin Herrenschmidt
- Bjorn Helgaas
- Dan Williams
- Eric Auger
- Jacob Pan
- Jason Gunthorpe
- Jean-Philippe Brucker
- Jonathan Cameron
- Jörg Rödel
- Kevin Tian
- Krzysztof Wilczyński
- Lorenzo Pieralisi
- Lu Baolu
- Marc Zyngier
- Peter Zijlstra
- Thomas Gleixner

Contacts:

- Alex Williamson (alex.williamson@redhat.com)
- Bjorn Helgaas (bhelgaas@google.com)
- Jörg Roedel (jroedel@suse.de)
- Lorenzo Pieralisi (lorenzo.pieralisi@linaro.org)
- Krzysztof Wilczyński (kw@linux.com)

Primary authors: WILLIAMSON, Alex; HELGAAS, Bjorn (Google); ROEDEL, Joerg (SUSE); Mr WILCZYŃSKI, Krzysztof (Individual); PIERALISI, Lorenzo

Track Classification: LPC Microconference Proposals

Contribution ID: **28** Type: **not specified**

x86 Microconference

X86-focused material has historically been spread out at Plumbers. This will be an x86-focused microconference. Broadly speaking, anything that might affect arch/x86 is on topic, except where there may be a more focused discussion occurring, like around Confidential Computing or KVM.

This microconference would look at how to address new x86 processor features and also look back at how older issues might be made less painful. For new processor features like APX, what is coming? Are the vendors coordinating and are they compatible? For older issues like hardware security vulnerabilities, is the current approach working? If not, how should they be dealt with differently? Can new hardware features or vendor policies help?

As always, the microconference will be a great place for coordination among distributions, toolchains and users up and down the software stack. All the way from guest userspace to VMMs.

Potential Problem Areas to Address:

- CPU Vulnerabilities
- Default options for mitigations
- Are they being mitigated right?
- Are hardware interfaces for Data Independent Execution being plumbed into applications?
- FRED - new kernel entry/exist hardware
- What doors does FRED open?
- What things will be FRED-only?
- CET - Control flow Enforcement
- Security Hardware feature, includes Shadow Stacks and Indirect Branch Tracking
- Kernel Shadow Stacks
- User IBT/FineIBT?
- APX - new Intel ISA, more general purpose registers (GPRs) ⋯(and more)
- What would a kernel with more GPRs look like?
- What plumbing implications does the MPX XSAVE offset reuse have?
- x86-S - Some future x86 CPUs may have a Smaller feature set and not be backward compatible
- SIPI64 is nice-ish, but other aspects are going to be especially nasty for virt
- Memory Protection Keys
- Userspace: Should we expand the ABI to cover more use cases?
- Can it be used to improve userspace security?
- Kernel: Page Table protections, mitigate malicious writes
- Memory Tagging / LAM / UBI
- CoCo Pain Points - what should the vendors be doing to ease them?
- XSAVE - Stay the course, or give up?
- How to ease the pain on gdb of AMD and Intel format divergence?

- x86 feature detection
- X86_FEATURE* - Is the code patching variants worth it? Should we pare down the choices? Do they really need to be per-cpu or should they be global?
- Should we impose more order in early boot about when it is OK to start checking feature flags or other parts of ‘boot_cpu_data’? Is this a good idea? Should ‘cpuinfo_x86’ be slimmed down further? - DaveH Boot
- Can the decompressor be entirely separated from the rest of the kernel proper?
- What old code imposes a maintenance burden and might be removed?

Primary authors: PETKOV, Borislav; HANSEN, David

Track Classification: LPC Microconference Proposals

Contribution ID: **31** Type: **not specified**

Zoned Storage Devices MC

Zoned Storage Devices MC - SMR HDDs, ZNS SSDs, Zoned mobile flash (UFS)

We making good progress with zoned storage support in Linux, improving and adding support throughout the stack from low level drivers to file systems, user space tooling and cloud infrastructure.

Since the last LPC2022 MC on the topic 1, lots of stuff has happened:

- Zoned Mobile flash is now supported in UFS3 and F2FS 4
- The deadline scheduler is no longer required for zoned storage devices 5
- Ceph Crimson supports Zoned Namespace SDDS and Host managed SMR drives 1
- ZoneFS continues to be improved
- Data placement is back in fashion, now because of zoned storage 4.
- Btrfs zoned support is improving 5
- UBLK added support for zoned storage
- XFS is growing support for zoned rt sub volumes (off list for now, but should be presentable at LPC)
- Loads of research has been done 9

Data placement, btrfs and zonefs were discussed at LPC2022.

I propose that we would spend half of the time allotted to summing up where we are today with quick overviews and then spend the second half with BOFs, kicking of discussions. If we can get a room for a day it would be awesome. It would be fun to finish up with post-mc beverages somewhere.

BoF ideas:

Data placement
Garbage collection
Write throttling
Testing
Benchmarking

People that would be great to have in the room - usual suspects in this area along with people who have done research on the subject, in semi-random order:

Johannes Thumshirn(BTRFS)
Naohiro Aota(BTRFS)
Joseph Josef Bacik (BTRFS)
Bart Van Assche (Block layer, F2FS)
Daeho Jeong (F2FS)
Jaegeuk Kim (F2FS)
Boris Burkov (BTRFS)
Damien Le Moal (ZoneFS, block layer..)
Niklas Cassel (block layer)
Kuankuan Guo (User space file systems)
Pankaj Raghav (support non-power of 2 zoned devices)
Kanchan Joshi (block layer)

Keith Busch (NVMe)
Viacheslav Dubeyko(ssdfs)
Shai Bergman (swap research 7)
Abutalib Aghayev (research on ceph, ext4 8)
Luis Chamberlain (testing)
Javier Gonzales (research)
Andreas Hindborg (ublk)
Ming Lei (ublk)
Hans Holmberg(ZNS enablement, research, ZenFS, XFS)
Matias Bjorling (ZNS, research, ..)
Dennis Maisenbacher (cloud infrastructure, gc research)
Jorgen Hansen(research)
Hannes Reinecke
Christoph Hellwig

References:

1 https://lpc.events/event/16/sessions/149/#20220914
2 https://www.sniadeveloper.org/sites/default/files/SDC/2023/presentations/SNIA-SDC23-Bj%C3%B8rling-Towards-Large-scale-Deployments-with-Zoned-Namespace-SSDs.pdf
3 https://lore.kernel.org/all/20230822191822.337080-1-bvanassche@acm.org/T/
4 https://lwn.net/Articles/939342/
5 https://lore.kernel.org/all/20240328004409.594888-22-dlemoal@kernel.org/T/
6 https://lwn.net/Articles/960486/
7 https://shai.pub/assets/pdf/atc22-bergman.pdf
8 https://scholar.google.com/citations?user=Kfb6IZ4AAAAJ&hl=en
9 https://scholar.google.com/scholar?hl=sv&as_sdt=0%2C5&q=%22Zoned+storage%22&oq=

Primary author: HOLMBERG, Hans

Track Classification: LPC Microconference Proposals