Continuing in the same direction as last year, this year’s Android microconference will be an opportunity to foster collaboration between the Android and Linux kernel communities. Discussions will be centered on the goal of ensuring that both the Android and Linux development moves in a lockstep fashion going forward.

Projected talks:
- io_uring in Android (Akilesh Kailash)
- MGLRU results on Android (Kalesh Singh or Yu Zhao presenting over VC)
- Hermetic builds with Bazel (Matthias Männich)
- Android kernel testing updates (Steve Muckle)
- pKVM (Quentin Perret)
- erofs as a replacement for f2fs and the deprecation of ext4 (David Anderson)
- eBPF-based FUSE (Paul Lawrence)
- stgdiff tools (Giuliano Procida)
- Technical debt (Lee Jones)
- Parallelized suspend/resume (Saravana Kannan)
- CPU DVFS for guest thread migrations (Saravana Kannan)

Accomplishments since the last Android MC:
- fw_devlink: Fixed the correctness of sync_state() callbacks when simple-bus devices are involved
- Implemented a prototype for the cgroup-based accounting of DMA-BUF allocations – current review doc: https://patchwork.kernel.org/project/linux-media/patch/20220328035951.1817417-2-tjmercier@google.com/
- Other dependencies for tracking shared gfx buffers now merged
- Improved community collaboration:
- Collaboration page set up: https://aosp-developers-community.github.io/
- Integrating v4l2_codec2 HAL on v4l2-compliant upstream codecs WIP

Deadline for submitting new talks: August 1st.

MC leads:
Karim Yaghmour karim.yaghmour@opersys.com
Suren Baghdrasaryan surenb@google.com
John Stultz jstultz@google.com
Amit Pundir amit.pundir@linaro.org
Sumit Semwal sumit.semwal@linaro.org

I agree to abide by the anti-harassment policy

Primary authors: PUNDIR, Amit; STULTZ, John; YAGHMOUR, Karim (Opersys inc.); SEMWAL, Sumit (Linaro); BAGHDASARYAN, Suren
Presenter: YAGHMOUR, Karim (Opersys inc.)

Track Classification: LPC Microconference Track (CLOSED)
(Impact of) Recent CPU topology changes

Tuesday, 13 September 2022 17:55 (15 minutes)

Acting on the expectation that both device-tree and ACPI enabled systems must present a consistent view of the CPU topology, Sudeep submitted at [1] (currently v6) a series of patches that firstly fix some discrepancies in the CPU topology parsing from the DT /cpu-map node, as well as improve detection of last level cache sharing. The conference topic aims to discuss the impact of these changes to userspace facing CPU topology information and in the use of more unconventional DT topology descriptions ("phantom domains" - u-arch or frequency subgrouping of CPUs) present in Android systems.

[1] https://lore.kernel.org/lkml/20220704101605.1318280-1-sudeep.holla@arm.com/

I agree to abide by the anti-harassment policy

Yes

Primary authors: EGGEMANN, Dietmar; VOINESCU, Ionela

Presenters: EGGEMANN, Dietmar; VOINESCU, Ionela

Session Classification: Android MC

Track Classification: LPC Microconference: Android MC
Cuttlefish and Kernel Dev

Tuesday, 13 September 2022 16:25 (15 minutes)

Cuttlefish is an Android based VM that can be used for kernel hacking amongst other things. We’ll chat about how to set one up, put a mainline kernel on it, and utilize the devices it supports.

I agree to abide by the anti-harassment policy
Yes

Primary author: MUTHIAH, Ram
Presenter: MUTHIAH, Ram
Session Classification: Android MC

Track Classification: LPC Microconference: Android MC
Dynamic Energy Model to handle leakage power

The Energy Model (EM) framework describes the CPU power model and is used for task scheduling decisions or thermal control. It’s setup during the boot in one of the supported ways and is not modified during the normal run. Although, not every chip has the same power characteristics and cores inside might be sensitive to temperature changes in different way.

To address better the variety of silicon fabrications we want to allow modifications of the EM at runtime. The EM runtime modification would introduce new features:
- allow to provide (after boot) the total power values for each OPP not limited to any formula or DT data
- allow to provide power values proper for a given SoC manufactured - with different binning and read from FW or kernel module
- allow to modify at runtime power values according to current temperature of the SoC, which might increase leakage and shift power-performance curves for Big core more than for other cores

I agree to abide by the anti-harassment policy

Yes

Primary author: LUBA, Lukasz
Presenter: LUBA, Lukasz
Session Classification: Android MC
Track Classification: LPC Microconference: Android MC
The file system in userspace, or fuse filesystem, is a long-standing filesystem in linux that allows a file system to be implemented in user space. Unsurprisingly, this comes with a performance overhead, mostly due to the large number of context switches from the kernel to the user space daemon implementing the file system.

bpf, or berkeley packet filters, is a mechanism to allow user space to put carefully sanitized programs into the kernel, initially as part of a firewall, but now for many uses. fuse-bpf is thus a natural extension of fuse, adding support for backing files and directories that can be controlled using bpf, thus avoiding context switches to the kernel. This allows us to use fuse in many more places in Android as performance is very close to the native file system.

I agree to abide by the anti-harassment policy

Yes

Primary author:  LAWRENCE, Paul (Google Inc)
Presenter:  LAWRENCE, Paul (Google Inc)
Session Classification:  Android MC
Track Classification:  LPC Microconference: Android MC
EROFS as a replacement for EXT4 and Squashfs

Tuesday, 13 September 2022 17:10 (15 minutes)

EROFS is a readonly filesystem that supports compression. It is rapidly becoming popular in the ecosystem. This talk will explore its performance implications and space-saving benefits on the Android platform, as well as ideas for future work.

I agree to abide by the anti-harassment policy

Yes

Primary author: ANDERSON, David (Google)
Presenter: ANDERSON, David (Google)
Session Classification: Android MC
Track Classification: LPC Microconference: Android MC
fw_devlink and parallelization updates

Tuesday, 13 September 2022 15:55 (15 minutes)

fw_devlink parses the firmware (device tree) to figure out device dependencies and uses that to enforce probe ordering and suspend/resume ordering between consumer and supplier devices. It is also used to implement sync_state() callbacks that let a supplier know when all its consumers have probed.

In this presentation, we’ll talk about how some of the issues that were discussed in LPC 2021 [1] have been resolved and any new issues that have come up. In addition, we’ll discuss how we could use fw_devlink to enabled parallelized boot and suspend/resume by default for DT based systems.

[1] https://lpc.events/event/11/contributions/1053/

I agree to abide by the anti-harassment policy

Yes

Primary author:  KANNAN, Saravana (Google Inc)
Presenter:  KANNAN, Saravana (Google Inc)
Session Classification:  Android MC
Track Classification:  LPC Microconference: Android MC
GKI experience

Tuesday, 13 September 2022 15:05 (10 minutes)

Qualcomm will provide an update on commercialization of a GKI-based target. This short talk will discuss the benefits to adopting GKI model (LTS intake frequency, upstream adoption) and some of the challenges we faced. Finally, we will discuss future challenges for GKI products with respect to upstream kernel development.

I agree to abide by the anti-harassment policy

Yes

Primary author: Berman, Elliot
Presenter: Berman, Elliot
Session Classification: Android MC
Track Classification: LPC Microconference: Android MC
Hermetic builds with Bazel

Tuesday, 13 September 2022 15:25 (15 minutes)

Starting with Android 13, Android Kernels can be built with Bazel. While under the hood this still uses Kbuild as the authoritative build system, the guarantees a Bazel build provides are very valuable for building GKI kernels and modules. This talk will explore the Bazel based kernel build and the Driver Developer Kit (DDK) that provides a convenient way to create kernel modules in compliance with GKI.

I agree to abide by the anti-harassment policy

Yes

Primary author:  MÄNNICH, Matthias (Google)

Presenter:  MÄNNICH, Matthias (Google)

Session Classification:  Android MC

Track Classification:  LPC Microconference: Android MC
This presentation will talk about the usage of io_uring in Android OTA, present performance results. Android OTA uses dm-user which is a out of tree user space block device.

We plan to explore io_uring evaluating the RFC patchset: https://lore.kernel.org/io-uring/Yp1jRw6kiUf5jCrW@B-P7TQMD6M-0146.local/T/#m013adcbd97cc4c4d810f51961998ba569ecc2a62

I agree to abide by the anti-harassment policy

Yes

Primary author: KAILASH, Akilesh
Presenter: KAILASH, Akilesh
Session Classification: Android MC
Track Classification: LPC Microconference: Android MC
MGLRU results on Android

Tuesday, 13 September 2022 17:25 (15 minutes)

Multigenerational LRU (MGLRU) is a rework of the kernel’s page reclaim mechanism where pages are categorized into generations representative of their age. It provides a finer granularity aging than the current 2-tiered active and inactive LRU lists, with the aim to make better page reclaim decisions.

MGLRU has shown promising improvements from various platforms/parties. This presentation will underline the results of evaluating the patchset of Android.

[1] https://lore.kernel.org/r/20220309021230.721028-1-yuzhao@google.com/

I agree to abide by the anti-harassment policy

Yes

Primary author: SINGH, Kalesh (Google Inc)
Presenter: SINGH, Kalesh (Google Inc)
Session Classification: Android MC
Track Classification: LPC Microconference: Android MC
STG for ABI monitoring

Tuesday, 13 September 2022 15:40 (15 minutes)

ABI monitoring is an important part of the stability and upgrade-ability promises of the GKI project. In the latest generation of our tooling, we applied concepts from graph theory to the problem space and gained high confidence in the correctness and reliability of the analysis results. What else can we fit into this model and what would be most useful?

I agree to abide by the anti-harassment policy

Yes

Primary author: PROCIDA, Giuliano
Presenter: PROCIDA, Giuliano
Session Classification: Android MC
Track Classification: LPC Microconference: Android MC
For various reasons, the Android Common Kernel (ACK) requires functionality that is not suitable for upstream. This talk will explore the reasons why this delta must exist, how it is maintained & managed and the steps taken to ensure that it remains as small as possible.

I agree to abide by the anti-harassment policy

Yes

Primary author: MÄNNICH, Matthias (Google)
Presenter: MÄNNICH, Matthias (Google)
Session Classification: Android MC
Track Classification: LPC Microconference: Android MC
Virtualization in Android

Tuesday, 13 September 2022 16:10 (15 minutes)

In this presentation we will talk about Protected KVM and the new virtualization APIs introduced with Android 13. You’ll find out more about some of the key Protected KVM design decisions, its upstream status and how we plan to use protected virtualization for enabling a new set of use cases and better infrastructure for device vendors.

I agree to abide by the anti-harassment policy

Yes

**Primary authors:** BRAZDIL, David (Google); CONSTANTINESCU, Serban

**Presenters:** BRAZDIL, David (Google); CONSTANTINESCU, Serban

**Session Classification:** Android MC

**Track Classification:** LPC Microconference: Android MC
Compute Express Link MC

Compute Express Link is a cache coherent fabric that is gaining momentum in the industry. Several hardware vendors have begun to ramp up on CXL 2.0 hardware development and the Linux plumbing needs to keep pace. There is foundational infrastructure in place in the kernel, QEMU, and user tooling that represents the beginning of the CXL enabling journey. However, there is more to do and plenty of opportunities to scale the contributions past the initial set of developers that got the CXL subsystem off the ground.

The Compute Express Link microconference focuses on how to evolve the Linux CXL kernel driver and userspace components for full support of the CXL 2.0 spec (and beyond). The microconference provides a space 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 kernel 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.

Suggested topics:
- Ecosystem & Architectural review
- Regions
- QEMU support
- Security: IDE and SPDM
- Managing vendor specificity
- Type 2 accelerator support (bias flip management)
- RAS (GPF, AER, Poison handling )
- 1.1 to 2.0 compatibility
- Hot add policy, daxctl
- Hot remove
- Industry and academia use cases

Interesting Links
....

MC leads:
- Ben
- Dan
- Adam
- Davidlohr

I agree to abide by the anti-harassment policy

Primary author:  WIDAWSKY, Ben
Co-author:  WILLIAMS, Dan (Intel Open Source Technology Center)
Presenters:  MANZANARES, Adam (Samsung Electronics);  WIDAWSKY, Ben;  WILLIAMS, Dan (Intel Open Source Technology Center)
Track Classification: LPC Microconference Track (CLOSED)
CXL 2.0+ Emulation With QEMU Status, requirements and roadmap.

Wednesday, 14 September 2022 18:00 (25 minutes)

This session will provide a brief status report on emulation of CXL in QEMU: What’s upstream, what’s queued and what’s already in development.

The bulk of the time will focus on discussion of priorities for the next year.

The limited availability of CXL 2.0 hardware, against high priority of support when such hardware is available, meant that the Linux Kernel stack has been developed and tested against QEMU emulation (along with mocking in the kernel).

There are a number of advantages to QEMU:

- Common test platform available to all developers
- Flexible test platform - can emulate many topologies
- More complete emulation than practical with mocking approaches
- Great platform for future specification feature verification.

The base support for CXL Type 3 Devices, root ports and host bridges on x86 has merged. We will provide an up to date status report and in particular highlight some of the other elements that already exist.

However, this emulation is far from feature complete and CXL specification continues to grow. So the main focus of this session will be on discussing a future road map and establishing priority + seeking additional contributors to drive this road map forwards. A straw man proposal will be available to get the discussion going. That road map will need to align with and support OS and other software stack road maps so key to a successful session will be getting input from those active in those related areas.

I agree to abide by the anti-harassment policy

Yes

Primary author:  CAMERON, Jonathan (Huawei Technologies R&D (UK))
Presenter:      CAMERON, Jonathan (Huawei Technologies R&D (UK))
Session Classification: Compute Express Link MC
Track Classification: LPC Microconference: Compute Express Link MC
CXL and confidential computing

Confidential Computing aims to provide isolation to the end user from the infrastructure provider (like a Cloud provider). The infrastructure provider should never be able to access or even handle in plain text the customer data.

CPU vendors have been extending providing solutions to achieve confidential computing on the CPU. Confidential computing needs to extend to accelerators, to that end transport standards like CXL need to address the requirements that come with confidential computing.

This session intends to focus on the software stacks components after giving a short introduction to confidential computing and its requirements (encryption, isolation & integrity).

I agree to abide by the anti-harassment policy

Yes

Primary author:  Mr GLISSE, Jérôme (Google)
Presenter:  Mr GLISSE, Jérôme (Google)
Session Classification:  Compute Express Link MC
Track Classification:  LPC Microconference: Compute Express Link MC
CXL hotplug: spec to reality

Wednesday, 14 September 2022 17:10 (25 minutes)

CXL is an exciting new technology for many reasons. Between promised latency improvements to new device models with CXL.mem and CXL.cache, it has the potential to push peripheral devices into very new territory. However, what is in the specs versus what reality is have been two very different things.

We’d like to generate discussion around hotplug support. The first generation of CXL hitting the market should be CXL 1.1, but 2.0 is where hotplug “officially” arrives. With no useful hotplug support, this can break many vendors’ use of updating FPGA-based devices in the field if switching to CXL.

Given the perceived interest in the CXL Consortium meetings around the reality of hotplug support versus what is in the spec, we felt this could be a useful discussion topic for Linux kernel solutions to fill in the 1.1 -> 2.0 gap.

I agree to abide by the anti-harassment policy

Yes

Primary author:  Mr WASKIEWICZ, PJ (Jump Trading Group)
Presenter:  Mr WASKIEWICZ, PJ (Jump Trading Group)
Session Classification:  Compute Express Link MC
Track Classification:  LPC Microconference: Compute Express Link MC
Contribution ID: 315

CXL MC Welcome and Kickoff

Wednesday, 14 September 2022 15:00 (5 minutes)

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

Session Classification: Compute Express Link MC
CXL new features and interaction with memory management.

Wednesday, 14 September 2022 15:30 (25 minutes)

CXL brings new and increasing challenges wrt to memory management. This session will discuss some new features in this space and via straw man proposals aim to get some initial alignment on how to support these features. More details to be added at a later date.

I agree to abide by the anti-harassment policy

Yes

Primary author:  CAMERON, Jonathan (Huawei Technologies R&D (UK))
Presenter:  CAMERON, Jonathan (Huawei Technologies R&D (UK))
Session Classification:  Compute Express Link MC
Track Classification:  LPC Microconference: Compute Express Link MC
CXL Type-3 device discovery, configuration in firmware and prepare ACPI tables for kernel usage

There has been growing need for increasing memory capacity in enterprise design and CXL has emerged as one of the preferred solution to meet this increasing demand. CXL and other relevant specifications have defined various mechanisms, which allow kernel to utilize the extended memory capacity.

In this presentation, an implementation of firmware support for a AArch64 compliant platform will be discussed. Firmware performs enumeration, Type-3 device discovery and identifying its capabilities. In addition to this, using the DOE mechanisms, CDAT structures are fetched and details are passed on through different phases of booting firmware. Host interconnect is configured by the firmware for device memory access according to the device property. Secure firmware performs all these operations and subsequently in non-secure firmware (EDK2), all Type-3 device details are captured in the ACPI tables (SRAT, HMAT, CEDT etc.) for usage by kernel software. Linux drivers parse and utilize ACPI information, for configuring host and device accordingly and uses the extended memory as separate NUMA node. For this discussion, the implementation of the above on the Neoverse N2 reference design FVP will be used as an example.

I agree to abide by the anti-harassment policy

Yes

Primary author: Mr PATTANAYAK, Sayanta (ARM Ltd.)
Presenter: Mr PATTANAYAK, Sayanta (ARM Ltd.)
Session Classification: Compute Express Link MC
Track Classification: LPC Microconference: Compute Express Link MC
Design and implementation of Autocaching for CXLSSD

Wednesday, 14 September 2022 16:45 (25 minutes)

The emerging CXL interface provides access to storage devices via IO(block) interfaces and character(memory) interfaces. The duality of the interface requires rethinking the current upstream memory and storage subsystem to support these new devices efficiently. Historically, storage devices are considered block devices accessed through a block interface. In this case, the data should be read into host memory, such as the page cache.

To leverage the advantages of both memory and IO interfaces, we propose Autocaching. Autocaching integrates directly accessible storage memory into the virtual file system layer using the device struct page. So that the application or kernel can access page caches or device pages transparently. Autocaching can allocate page cache or device page depending on data or access characteristics. In addition, the access types, indirect access using page cache and direct access using device page, can be dynamically changed with page hotness or memory usage. In this talk, we will detail the kernel changes required for Autocaching as well as share our plans for upstreaming.

I agree to abide by the anti-harassment policy

Yes

Primary authors: PARK, Heekwon (Samsung Electronics); GIM, Jongmin (Samsung); JUNG, Jaemin (Samsung Semiconductor); MANZANARES, Adam (Samsung Electronics); BUESO, Davidlohr (Samsung Semiconductor); GONZALEZ, Javier (Samsung); KI, YANG SEOK (Samsung)

Presenter: PARK, Heekwon (Samsung Electronics)

Session Classification: Compute Express Link MC
Linux plumbing of CXL error reporting

Wednesday, 14 September 2022 17:35 (25 minutes)

With the introduction of CXL Type 3 Memory Devices a system may contain multiple different memory controllers to support and provide volatile memory. To add support of all those, generic and architectural specific implementations across different subsystems (CXL, PCI, ACPI, MCA, EDAC, etc.) are involved. CXL introduces following errors:

• CXL link and protocol errors and
• CXL type 3 memory device errors.

So now a broad variety of error types and sources must be handled additionally compared to what typically exists for mem controllers or pci devices.

The CXL kernel interface also provides an ioctl user interface to retrieve error events. And, there are a couple of new tools to control all that.

All this raises new questions on how to share, rework and plumb existing subsystems to make CXL work. And, should new APIs and tools for collecting errors be introduced what would be more suitable?

Topics of the discussion include:

• Should there be the same look and feel as with dram mem controllers when reporting cxl memory errors?
• Should errors be handled in userspace? How can access be serialized, how to support multiple users? Which tools should be the focus on?
• How can we reuse PCIe AER and RCEC implementations of the pci stack for cxl? Should we join pci and cxl, in particular maintaining a struct pci_dev for each cxl dev?
• What are the challenges in supporting multiple mem controllers in the system?
• Is there a sufficient need to integrate cxl error reporting into the EDAC subsystem?

A very brief overview of CXL error reporting and involved Linux subsystems will be presented to further discuss and find answers for above questions.

I agree to abide by the anti-harassment policy

Yes
Primary authors: Mr RICHTER, Robert (Advanced Micro Devices); Mr GHANNAM, Yazen (Advanced Micro Devices)

Presenters: Mr RICHTER, Robert (Advanced Micro Devices); Mr GHANNAM, Yazen (Advanced Micro Devices)

Session Classification: Compute Express Link MC

Track Classification: LPC Microconference: Compute Express Link MC
Confidential Computing MC

Last year's inaugural Confidential Computing microconference brought together plumbers enabling secure execution features in hypervisors, firmware, Linux Kernel, over low-level user space up to container runtimes.

Good progress was made on a couple of topics, most outstanding here is the development of Linux guest support for Intel TDX and AMD SEV-SNP. The patch-sets for both are under intensive review and come close to being merged upstream.

The discussions in the microconference also helped to move other topics forward, such as support for unaccepted memory or deferred memory pinning for confidential guests.

But enabling Confidential Computing in the Linux ecosystem is an ongoing process, and there are still many problems to solve. The most important ones are:

- Design and implementation of Intel TDX and AMD SEV-SNP host support
- Linux kernel memory management changes for secure execution environments
- Support of upcoming secure execution hardware extensions from ARM and RISC-V
- Pre-launch and runtime attestation workflows
- Interrupt security for AMD SEV-SNP
- Delegatability and live migration of encrypted virtual machines
- Proper testing of confidential computing support code

The Confidential Computing Microconference wants to bring together plumbers working on secure execution features to discuss these and other open problems.

Key Attendees:

- Andi Kleen
- Andy Lutomirski
- Borislav Petkov
- Brijesh Singh
- Dr. David Alan Gilbert
- Dave Hansen
- David Hildenbrand
- David Kaplan
- David Rientjes
- Joerg Roedel
- Jun Nakajima
- Kirill A. Shutemov
- Marc Orr
- Mike Rapoport
- Paolo Bonzini
• Peter Gonda
• Sathya Kuppuswamy
• Sean Christopherson
• Tom Lendacky

I agree to abide by the anti-harassment policy

**Primary author:** ROEDEL, Joerg (SUSE)

**Presenter:** ROEDEL, Joerg (SUSE)

**Track Classification:** LPC Microconference Track (CLOSED)
Hardening Linux guest kernel for CC

Tuesday, 13 September 2022 11:20 (20 minutes)

A lot of effort in past couple of years has been spent in enabling various CC HW technologies (AMD SEV, Intel TDX) to be able to support Linux guests. However in order to be able to provide an adequate level of security for CC Linux guests (regardless of the underlying chosen HW technology), we need to collaborate together to harden the core Linux kernel codebase, as well as drivers that are planned to be used by various Cloud Service Providers (CSPs).

This session will briefly outline the scope of work that have been doing at Intel on this direction for the past 1.5 years, as well as all the future work that still needs to happen. The main goal of the session is to gain feedback and have a discussion on the best possible approach to move forward together as a community.

I agree to abide by the anti-harassment policy

Yes

Primary author: RESHETOVA, Elena (Intel)
Presenter: RESHETOVA, Elena (Intel)
Session Classification: Confidential Computing MC
Track Classification: LPC Microconference: Confidential Computing MC
Identifying and Eliminating Contention from Booting Concurrent SNP VMs

Tuesday, 13 September 2022 12:30 (20 minutes)

We will present an evaluation of concurrent boot time of CVMs running under AMD’s SEV-SNP technology. Specifically we will discuss how booting SNP VMs concurrently can significantly slow down each other due to software bottlenecks in managing the RMP page state.

Then, we will discuss different mitigations that we’ve identified ranging from reducing lock contention to rate limiting Page State Change (PSC) requests from the guest. We hope to generate discussion on how to eliminate the software bottlenecks that we’ve identified to properly isolate concurrent SNP VMs so that they do not degrade each other’s performance.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  LI, Jacky (Google); ORR, Marc (Google)
Co-author:  GUN, Alper (Google)
Presenters:  LI, Jacky (Google); ORR, Marc (Google)
Session Classification:  Confidential Computing MC
Track Classification:  LPC Microconference: Confidential Computing MC
Interrupt Security for AMD SEV-SNP

Tuesday, 13 September 2022 13:10 (20 minutes)

Discussion about SEV-SNP support for Restricted Interrupt Injection and Alternate Interrupt Injection. These features enforce additional interrupt and event injection security protections designed to help protect against malicious injection attacks. Safe isolation between an SNP-protected guest and its host environment requires restrictions on the type of exception and interrupt dispatch that can be performed in the guest. Isolated guests are expected to run with the SNP Restricted Injection feature active, limiting the host to ringing a doorbell with a #HV exception. This essentially means when restricted injection is active, only #HV exceptions can be injected into the SEV-SNP guest. The majority of information communicated by the host is specific to the virtualization architecture (e.g. Virtio or VMBus messages) and will be delivered in a manner that is understood by the specific drivers running within the guest.

Description of the GHCB #HV doorbell communication to inject the exception or interrupt and description of the #HV doorbell page and the #HV Doorbell Page NAE event, as documented in the GHCB specification.

Detailed discussions on current implementation of Restricted Interrupt Injection on KVM, specifically about dispatch handling of system vectors and device interrupt vectors optimally from within the #HV exception handler. Also, changes required in the kernel’s interrupt exit code path to support #HV exception handler and handling #HV exception with respect to kernel’s interrupt enable/disable and idle/wakeup code paths.

I agree to abide by the anti-harassment policy

Yes

Primary author:  KALRA, Ashish
Presenter:  KALRA, Ashish
Session Classification:  Confidential Computing MC
Track Classification:  LPC Microconference: Confidential Computing MC
Securely booting confidential VMs with encrypting disk

Tuesday, 13 September 2022 10:40 (20 minutes)

Confidential Computing technologies offer guest memory encryption, but there's no standard way to securely start a confidential VM with encrypted disk. Such VMs must unlock the disk inside the guest, so the passphrase is not accessible to the host. However, in TDX and SEV-SNP guest attestation and secure secret injection depend on guest kernel features, so grub cannot be used for unlocking. Unlocking the encrypted disk later during boot is possible, but may allow various passphrase-stealing attacks to sneak into the boot stages before unlocking, and therefore requires stricter guest measurement. Unlocking at a later stage also means that upgrading the kernel and initrd inside the guest is more complex.

The talk will present various options for securely starting a confidential VM with encrypted disk for SEV, SEV-ES, SEV-SNP and TDX using embedded grub, measured direct boot, or secure vTPMs. We’d like to gather feedback from plumbers working on the kernel and adjacent projects (QEMU, OVMF, and other guest firmware) towards defining a mechanism for starting confidential VMs with encrypted disk in a way that is secure, works on different TEEs, easy to maintain and upgrade, and open-source.

I agree to abide by the anti-harassment policy

Yes

Primary authors: MURIK, Dov (IBM); FELDMAN-FITZTHUM, Tobin (IBM)

Presenters: MURIK, Dov (IBM); FELDMAN-FITZTHUM, Tobin (IBM)

Session Classification: Confidential Computing MC

Track Classification: LPC Microconference: Confidential Computing MC
Testing Intel TDX functionality with new set of self tests

Tuesday, 13 September 2022 12:50 (20 minutes)

The new TDX architecture makes changes to the hardware and the host and guest software stacks. All of these components are being developed simultaneously and are constantly changing. As the host kernel changes, we need a system to test its functionality which is independent from the guest enlightenment changes and doesn’t rely on a fully functional system which doesn’t exist yet.

We propose a new extension to the selftest framework for running simple code as a TD guest to test various functionality of the TDX hardware and host kernel support.

This framework has been in use by Google for several months and allows us to test memory access interactions between host and guest and allow testing of the Guest-Hypervisor Communication Interface (GHCI). It allowed us to uncover issues in the early development stages of TDX and surface requirements which are not always clear from the SPEC.

The framework was originally proposed in “[RFC PATCH 0/4] TDX KVM selftests” and we intend to send out an updated patch series based on Intel’s latest RFC V6 patch to TDX and include additional tests.

I agree to abide by the anti-harassment policy

Yes

Primary author: SHAHAR, Sagi (Google)
Co-authors: Mr AFRANJI, Ryan (Google); AKTAS, Erdem (Google)
Presenter: SHAHAR, Sagi (Google)
Session Classification: Confidential Computing MC
Track Classification: LPC Microconference: Confidential Computing MC
The elephants in the confidential room: Attestation and verification

Tuesday, 13 September 2022 12:10 (20 minutes)

While a lot of efforts are being put towards platform enabling for confidential computing, there’s one fundamental part of the technology that we ignore more often than not: Attestation.

Without having a way to verify that the data we’re trying to protect with confidential computing platforms is generated by a TCB that we know and validate, the whole confidential computing trust model falls apart.

As an attestation services client, the confidential containers attestation agent is entirely dependent on the local or remote Key Brokering Services implementation that it talks to. While working on this piece of software we realized how fragmented this part of the confidential computing ecosystem is: From the attestation evidence format to the verification policies or the reference values provisioning, each and every combination of a CSP, a silicon vendor and an OEM creates a new flow to support.

In this talk we will present our current proposals for building generic, vendor agnostic frameworks for attestation, verification and reference values providing services. We’ll describe how our modular approach should allow for plugging existing, vendor-specific implementations, formats and flows as services back-ends. But most importantly, we’d like to discuss about a longer term goal: Finding a more uniform, less fragmented path for attestation flows, formats and requirements.

I agree to abide by the anti-harassment policy

Yes

Primary author:  Mr ORTIZ, Samuel (Rivos)
Presenter:  Mr ORTIZ, Samuel (Rivos)
Session Classification:  Confidential Computing MC
Track Classification:  LPC Microconference: Confidential Computing MC
Unmapped Private Memory for Confidential Guests

Tuesday, 13 September 2022 10:20 (20 minutes)

Unmapped Private Memory (UPM) has been proposed as a new way to manage private guest memory for KVM guests. This session is intended to address any outstanding items related to the development/planning of Unmapped Private Memory support (UPM) for confidential guests. Some potential topics are listed below (though the actual agenda will be centered around topics that are still outstanding at that point in time):

- general design of related KVM/memfd interfaces
- pre-populating private memory for in-place encryption as part of guest startup (SEV, SEV-SNP, others?)
- restricting double-allocations due to userspace accessing/faulting in pages from shared backing store while a page has already been allocated from private backing store
- performance-related discussions

I agree to abide by the anti-harassment policy

Yes

Primary author: ROTH, Michael (AMD)
Presenter: ROTH, Michael (AMD)
Session Classification: Confidential Computing MC
Track Classification: LPC Microconference: Confidential Computing MC
Upstream & Guest Distro support for Confidential Compute

Tuesday, 13 September 2022 10:00 (20 minutes)

Confidential compute is developing fast. However, at Google we are facing challenges regarding the maintenance and support of guest distros. In particular, we're finding it difficult to maintain an efficient way of communicating with different distros and hard to test, validate, and merge fixes into guest distros.

**Backporting fixes**

Customers do not always run the guest with the latest kernels. In fact, the majority of our distros are still based on 5.4/5.10 kernels. Thus, bug fixes on the guest side usually require backporting the patches into the stable branches. Unfortunately, our backporting attempts got rejected several times because confidential compute is considered a new feature. Since the patches were considered to be new features instead of bug fixes, they were unable to be merged into stable branches. Ultimately, by working with each of our supported distros individually, we got our patches backported. This approach not only created unnecessary work on both our end and each distros’ end but also significantly delayed the time a patch got backported and merged in distros.

With the upcoming guest patches for Intel TDX and AMD SEV-SNP, we hope we can find a better way of maintaining guest-related confidential compute patch backporting, whether it is for supporting new or existing confidential compute features.

**Guest Distro testing and verification**

With upcoming launch for AMD SEV-SNP and Intel TDX, we are starting to support multiple types of confidential compute. For each offering, we will support several different guest distros. It is becoming harder to keep track on what confidential compute technology each guest distro version supports.

In addition to tracking supportability, we are also facing challenges on testing and verification of newly published images. It would be great if we can work on a common validation test suite for confidential compute to make sure new guest distros are indeed qualified before releasing them.

I agree to abide by the anti-harassment policy

Yes

**Primary author:** GAO, Jianxiong

**Presenter:** GAO, Jianxiong

**Session Classification:** Confidential Computing MC

**Track Classification:** LPC Microconference: Confidential Computing MC
Using DICE Attestation for SEV and SNP Hardware Rooted Attestation

Tuesday, 13 September 2022 11:00 (20 minutes)

Device Identifier Composition Engine (DICE) is a measured boot solution for systems without a TPM or similar hardware based capabilities. DICE is a layered approach meaning that each layer or software component of a boot takes inputs from the previous layer, its measurement and certificate, and then generates the same for the next phase of the boot. The output of this layering provides a strong code identity of all components of the boot. Since not all Confidential VM hardware contains TPM-like capabilities for attestation, DICE may be a solution for providing a meaningful attestation story for linux workloads in these environments.

I agree to abide by the anti-harassment policy

Yes

Primary author:  GONDA, Peter (Google)
Presenter:  GONDA, Peter (Google)
Session Classification:  Confidential Computing MC
Track Classification:  LPC Microconference: Confidential Computing MC
Containers and Checkpoint/Restore MC

The Containers and Checkpoint/Restore Microconference 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.

Contributions to the micro-conference are expected to be problem statements, new use-cases, and feature proposals both in kernel- and userspace.

Containers topics:

- User namespace improvements
- System call interception
- LSM improvements and LSM namespace
- CGroup2 transition, emulation and future extensions
- Memory isolation

Checkpoint/restore topics:

- CRIU and hardware security features
- Restartable sequences (rseq()) support
- Shadow stacks support
- GPU support (and other directly accessed hardware)
- Checkpoint/Restore standardization effort (driven by HPC)
- Kubernetes support

Primary authors: REBER, Adrian (Red Hat); BRAUNER, Christian; RAPOPORT, Mike (IBM); GRABER, Stéphane (Canonical Ltd.)

Presenters: REBER, Adrian (Red Hat); BRAUNER, Christian; RAPOPORT, Mike (IBM); GRABER, Stéphane (Canonical Ltd.)

Track Classification: LPC Microconference Track (CLOSED)
Bringing up FUSE mounts C/R support

Wednesday, 14 September 2022 13:00 (25 minutes)

Bringing up FUSE mounts C/R support

Intro

Each filesystem support in CRIU brings their own problems. Block-device based filesystems comparably easy to handle, we just need to save mount options and use it at the restore stage, it is also possible to provide such filesystems as an external mounts. Some virtual filesystems should be handled specially, for instance for tmpfs we care about saving entire fs content, for overlayfs we have to do some special processing to resolve source directories paths. But NFS and FUSE filesystems is totally different story. This talk is aimed to cover and discuss about the ways and problems which connected with FUSE filesystem support. There are some parallels between support for NFS (which is present in CRIU OpenVZ fork), but generally approach is different.

Right now we don’t have ready-to-go solution and support for FUSE C/R, this work was started by Vitaly Ostrosablin and me this year. We have ideas and PoC solutions for some of most important technical problems that comes into mind there but we also have things to discuss with the community.

Plan

Intro

The main problem with FUSE filesystem support is that FUSE tie up different kernel objects (fuse mount, fuse daemon task, fuse control device, fuse file descriptors, fuse file memory mappings). This is very challenging from the CRIU side because we have special order of kernel resources restoration. And this is not a question of our choice.

How CRIU handles files C/R? First of all, CRIU restores all the mounts. Tasks are restored lately. Why?

1. to have ability to restore file memory mappings at the same time as we restore process tree (to get VMAs inherited)
2. To restore memory mappings for files we need to have mount roots descriptors ready to use

Finally, we have a strict order mounts -> tasks and mappings. But FUSE breaks this logic totally. We need to have a FUSE daemon ready at the same time when we creating mount. But we can’t do that, because fuse daemon task may use some external resources like network sockets, pipes, file descriptors opened from another mounts.

What we can do with that? Idea is fairly simple and elegant. Let’s create fake fuse daemon and use it for mount fuse rarely, then, once we are ready we can replace fuse daemon by the original one. Good news here is that kernel allows to do that without any changes.

TBD

Next challenge. Dumping fuse file descriptors info with frozen network TBD
References

• [2] https://github.com/checkpoint-restore/criu/blob/7d7d25f946e10b00c522dc44eb9c60d9eba2e7a0/criu/files-reg.c#L2372

I agree to abide by the anti-harassment policy

Yes

Primary author: MIKHALITSYN, Alexander (Virtuozzo)
Co-author: OSTROSAVLIN, Vitaly
Presenter: MIKHALITSYN, Alexander (Virtuozzo)
Session Classification: Containers and Checkpoint/Restore MC

Track Classification: LPC Microconference: Containers and Checkpoint/Restore MC
rstat is a framework how generic hierarchical stats collection is implemented for cgroups. It is light on the writer (update) side since it works with per-cgroup per-cpu structures only (mostly). It is quick on the reader side since it aggregates only cgroups active since the previous read in a given subtree. It is used for accounting CPU time on the unified hierarchy, blkcg and memcg stats. Readers of the first two are user space queriers, the memcg stats are used additionally by MM code internally and hence memcg builds some optimizations above rstat. Despite that there were reports of readers being negatively affected by occasionally too long stats retrieval. This is suspected to be caused by some shared structures within rstat and their effect may get worse as more subsystems (or even BPF) start building upon rstat.

This talk describes how rstat currently works and then analyzes time complexity of updates and readings depending on number of active use sites. The result could already be a base for discussion and we will further consider some approaches to keep rstat durations under control with more new adopters and also how such methods affect error of collected stats (when tolerance is limited, e.g. for the VM reclaim code).

This presentation and discussion will fit in a slot of 30 minutes (give or take).

I agree to abide by the anti-harassment policy

Yes

Primary author:  KOUTNÝ, Michal
Presenter:  KOUTNÝ, Michal
Session Classification:  Containers and Checkpoint/Restore MC
Track Classification:  LPC Microconference: Containers and Checkpoint/Restore MC
Closing session

Wednesday, 14 September 2022 13:25 (5 minutes)

Presenter: BRAUNER, Christian
Session Classification: Containers and Checkpoint/Restore MC
How can we make procfs safe?

Wednesday, 14 September 2022 11:00 (30 minutes)

Thanks to openat2(2), it is now possible for a container runtime to be absolutely sure that they are accessing the procfs path they intended by using RESOLVE_NO_XDEV!RESOLVE_NO_SYMLINKS (the main limitation before this was the fact that there was no way to safely do the equivalent of RESOLVE_NO_XDEV in userspace on Linux, and implementing the necessary behaviour in userspace was expensive and bug-prone).

However, this method does not help if you need to access magiclinks in procfs (RESOLVE_NO_XDEV blocks all magiclinks and even if we allowed magiclink-jumps within the same vfsmount this wouldn’t help with any of the magiclinks we care about since they all cross the vfsmount boundary). Of particular concern are:

- /proc/self/fd/*
- /proc/self/exe
- When introspecting other processes, /proc/<pid>/ns/*, /proc/<pid>/cwd and /proc/<pid>/root.

The primary attack scenario is that we have seen attacks where not-obviously-malicious Kubernetes configurations have been used to get the container runtime to silently create unsafe containers (we need to access several procfs files when setting up a container and if any of the paths are redirected to "fake" procfs files, we would be silently creating insecure containers) – ideally it should be possible to detect these kinds of attacks and refuse to create containers in such an environment.

In this talk, we will discuss proposed patches to fix some of these endpoints (primarily /proc/self/fd/* through open(fd, "", O_EMPTYPATH)) and open up to a general discussion about how we might be able to solve the rest of them.

I agree to abide by the anti-harassment policy

Yes

Primary author: SARAI, Aleksa (SUSE LLC)
Presenter: SARAI, Aleksa (SUSE LLC)
Session Classification: Containers and Checkpoint/Restore MC
Track Classification: LPC Microconference: Containers and Checkpoint/Restore MC
Opening session

Wednesday, 14 September 2022 10:00 (5 minutes)

**Presenter:** GRABER, Stéphane (Canonical Ltd.)

**Session Classification:** Containers and Checkpoint/Restore MC
Restartable Sequences: Scaling Per-Core Shared Memory Use in Containers

Wednesday, 14 September 2022 12:35 (25 minutes)

Introducing per-memory-space virtual CPU IDs allocation domains helps solving user-space per-core data structure memory scaling issues as long as the data structure is private to a memory space (typically a single process). However, this does not help in use-cases where the data structure sits in shared memory used across processes.

In order to address this part of the problem, a per-container virtual CPU ID domain would be useful. This raises some practical questions about where this belongs: either an existing namespace or a new "vcpu domain" namespace, and whether this type of domain should be nestable or not.

Reference: “Extending restartable sequences with virtual CPU IDs”, https://lwn.net/Articles/885818/

I agree to abide by the anti-harassment policy

Yes

Primary author: DESNOYERS, Mathieu (EfficiOS Inc.)
Presenter: DESNOYERS, Mathieu (EfficiOS Inc.)
Session Classification: Containers and Checkpoint/Restore MC
Track Classification: LPC Microconference: Containers and Checkpoint/Restore MC
Restoring process trees with child-sub-reapers, nested pid-namespaces and inherit-only resources.

Wednesday, 14 September 2022 10:30 (30 minutes)

Re-parenting may put processes having same inherit-only resource into completely different and far away locations in the process tree, so that they don’t have ancestor/descendant relations between each other anymore.

In mainstream CRIU currently we don’t have nested pid-namespaces support and re-parenting to child-sub-reaper support. We just handle the most common case where task was re-parented to container init. To handle this case CRIU simply checks all children of the container init for non-session-leaders which can’t inherit session from init. We “fix” the original tree by moving such children to session leader sub-tree connecting them by helper task. After that we restore tasks based on the “fixed” tree and kill helpers so that we get the right tree which we check to be the same as the dumped one.

In this talk I want to first cover how we handle in Virtuozzo more complex cases with child-sub-reapers, nested pid-namespaces, and cases where re-parenting breaks longer branches in process tree.

And second I want to shed some light on the problem which we can’t handle in CRIU easily because of the lack of information from kernel, this problem was known from the early days of CRIU development and it is still present and vital to support arbitrary process trees.

Also I want to present one possible solution for the problem - “CABA” and hope to see some feedback on it.

Links:
https://src.openvz.org/projects/OVZ/repos/criu/commits/70eee0613acf
https://src.openvz.org/projects/OVZ/repos/criu/commits/aa77967c2f6c
https://lore.kernel.org/lkml/20220615160819.242520-1-ptikhomirov@virtuozzo.com/

I agree to abide by the anti-harassment policy

Yes

Primary author: TIKHOMIROV, Pavel (Virtuozzo)
Presenter: TIKHOMIROV, Pavel (Virtuozzo)
Session Classification: Containers and Checkpoint/Restore MC
Track Classification: LPC Microconference: Containers and Checkpoint/Restore MC
Tracer namespaces

Wednesday, 14 September 2022 10:05 (25 minutes)

There are various use-cases related to tracing which could benefit from introducing a notion of “tracer namespace” rather than playing tricks with ptrace. This idea was introduced in the LPC 2021 Tracing MC.

For instance, it would be interesting to offer the ability to trace system calls, uprobes, and user events using a kernel tracer controlled from within a container. Tracing a hierarchy consisting of a container and its children would also be useful. Runtime and post-processing trace filtering per-container also appears to be a relevant feature, in addition to allow dispatching events into a hierarchy of active tracing buffers (from the leaf going upwards to the root).

It would be preferable if this namespace hierarchy is separate from pid namespaces to allow use-cases similar to “strace” to trace a hierarchy of processes without requiring them to be in a separate pid namespace.

Introduce the idea of “tracer namespaces” and open the discussion on what would be needed to make it a reality.

I agree to abide by the anti-harassment policy

Yes

Primary author: DESNOYERS, Mathieu (EfficiOS Inc.)

Presenter: DESNOYERS, Mathieu (EfficiOS Inc.)

Session Classification: Containers and Checkpoint/Restore MC

Track Classification: LPC Microconference: Containers and Checkpoint/Restore MC
Unprivileged CRIU

Wednesday, 14 September 2022 12:10 (25 minutes)

This talk will discuss on-going changes to CRIU to introduce an “unprivileged” mode, utilizing a minimal set of Linux capabilities that allow for non-root users to checkpoint and restore processes.

It will also touch on a particularly motivating use-case; improving JVM start-up time.

I agree to abide by the anti-harassment policy

Yes

Primary author: MANTON, Younes
Presenter: MANTON, Younes
Session Classification: Containers and Checkpoint/Restore MC
Track Classification: LPC Microconference: Containers and Checkpoint/Restore MC
CPU Isolation MC

CPU Isolation can be considered as an holistic functionality that stems from a close combination of several kernel and userspace components working together to shield workloads with extreme latency or performance requirements from interruptions (also known as Operating System noise). An example of such type of workloads are DPDK (Data Plane Development Kit) use cases (Telco/5G) where even the shortest interruption (e.g., a few microseconds to service an IPI) can cause packet losses, eventually leading to exceeding QoS requirements.

Despite considerable improvements in the last few years towards implementing full CPU Isolation (nohz_full, rcu_noch, isolcpus, etc.), a certain amount of issues remain to be addressed, as it is still relatively simple to highlight sources of OS noise just by running synthetic workloads mimicking polling (always running) type of application similar to the ones mentioned above.

Recently, Improvements and discussions about CPU isolation features have been made and discussed in LKML [1,2,3], and tools such as osnoise tracer and rtlasno noise [3,4] improved the CPU isolation analysis. Nevertheless, this is an ongoing process, and discussions are needed to speed up solutions for existing discussions and improve the existing tools and methods.

With this microconference we thus want to get together to discuss open problems, most notably: how to improve the identification of OS noise sources (expanding runtime tracing and/or code inspection tools), how to track them publicly (repo/DB) and how to tackle the sources of noise that have already been identified (e.g. [1,2,3]).

A non exhaustive list of potential topics is:

- OS noise profiling (format and public DB for the community)
- Tracing to detect OS noise: the rtlasno noise tracer and what it’s missing
- TLB/icache flush deferral
- Extend cpuset v2 cpu partition feature to replace isolcpus kernel cmdline
- rt-trace-bpf tool
- Task isolation
- smp_call_function API improvements

Potential attendees list:
Peter Zijlstra, Thomas Gleixner, Steven Rostedt, Juri Lelli, Daniel Bristot de Oliveira, Sebastian Andrzej Siewior, Valentin Schneider, Clark Williams, Paul E. McKenzie, Paul Turner, Rik van Riel, Phil Auld, Waiman Long, Frederic Weisbecker, Nicolas Saenz Julienne, Marcelo Tosatti, Karl Rister, Andrew Theurer, Luis Claudio R. Goncalves

MC leads:
Juri Lelli
Daniel Bristot de Oliveira

refs.
1 - https://lore.kernel.org/all/20210929151723.162004989@infradead.org/
2 - https://lore.kernel.org/lkml/20220315153132.717153751@fedora.localdomain/
3 - https://lore.kernel.org/lkml/20211103170512.2745765-1-nsaenzju@redhat.com/
Primary authors: LELLI, Juri (Red Hat); BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)
Presenter: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)
Track Classification: LPC Microconference Track (CLOSED)
A long term project for CPU isolation is to allow its features to be enabled and disabled through cpusets. This includes nohz_full, unbound load affinity involving kthreads, workqueues and timers, managed IRQs, RCU nocb mode, etc... These behaviors are currently fixed in stone at boot time and can’t be changed until the next reboot... The purpose is to allow tuning these at runtime, which happens to be very challenging.

Let’s explore the current state of the art!

I agree to abide by the anti-harassment policy

Yes

Primary author: WEISBECKER, Frederic (Suse)
Presenter: WEISBECKER, Frederic (Suse)
Session Classification: CPU Isolation MC
Track Classification: LPC Microconference: CPU Isolation MC
CPU isolation comes with a handful of cpumasks to help determine which CPUs can sanely be interrupted, but those are not always checked when sending an IPI, nor is it always obvious whether a given cross-call could be omitted (or delayed) if targeting an isolated CPU.

1 (with 2 and 3 as required foundations) shows a way to defer cross-call work targeting isolated CPUs to the next kernel entry, but still requires a manual patching of the actual cross-call.

A grep for "on_each_cpu()" and "smp_call()" on a mainline kernel yields about 350 results. This slot will be about discussing ways to detect and classify those (remote data retrieval, system wide synchronization...), if and how to patch them and where to draw the line(s) on system administrator configuration vs what is expected of the kernel.

I agree to abide by the anti-harassment policy

Yes

Primary author: SCHNEIDER, Valentin (Red Hat)

Presenter: SCHNEIDER, Valentin (Red Hat)

Session Classification: CPU Isolation MC

Track Classification: LPC Microconference: CPU Isolation MC
Execution Context Aware IPIs

Interprocessor interrupts (IPIs) are a nuisance to a number of specialized workloads. Like an unwarranted visitor, they impose themselves onto processors that would rather be busy running critical tasks, idling, or waiting for an important event. There are methods for signaling a CPU is off-limits, like CPU isolation, but they not good enough. Things like cache operations slip through the cracks.

There is room for improvement nonetheless. By merging RCU’s extended quiescent state tracking into the context tracking subsystem the kernel has now gained a new API to keep track of its CPUs’ execution context, opening the door to finer-grained IPI targeting.

In this talk, I’ll make a brief review of the state of the art and discuss my efforts and the challenges of using context tracking to avoid TLB shoot-downs on nohz_full or idle x86_64 CPUs.

Among other things, it’ll be interesting to discuss other use-cases for this, as well as the eventual challenges of integrating into other architectures.

I agree to abide by the anti-harassment policy

Yes

Primary author: SAENZ JULIENNE, Nicolas (redhat)
Presenter: SAENZ JULIENNE, Nicolas (redhat)
Session Classification: CPU Isolation MC
Track Classification: LPC Microconference: CPU Isolation MC
Isolation aware smp_call_function/queue_work_on APIs

Changes to smp_call_function/queue_work_on style APIs
to take isolation into consideration, more specifically, would like to possibly return errors
for the callers who can handle them.

I agree to abide by the anti-harassment policy

Yes

Primary author: TOSATTI, Marcelo (Red Hat)
Presenter: TOSATTI, Marcelo (Red Hat)
Session Classification: CPU Isolation MC
Track Classification: LPC Microconference: CPU Isolation MC
Make RCU do less (and disturb CPUs less)!

Tuesday, 13 September 2022 11:15 (35 minutes)

CPUs can be disturbed quite easily by RCU. This can hurt power especially on battery-powered systems, where RCU can be a major consumer of power. Different strategies can be tried to mitigate power which we will show along with power data. Also I have been working on some patches to further reduce RCU activity in frequently-called paths like file close. This presentation is to discuss some test results mostly on the power consumption side on battery-powered Android and ChromeOS systems and the new work mentioned to delay RCU processing.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  FERNANDES, Joel; Mr KADAM, Rushikesh; REZKI, Uladzislau

Presenters:  FERNANDES, Joel; Mr KADAM, Rushikesh; REZKI, Uladzislau

Session Classification: CPU Isolation MC

Track Classification: LPC Microconference: CPU Isolation MC
rtla osnoise: what is missing?

Tuesday, 13 September 2022 12:55 (35 minutes)

The osnoise tracers enable the simulation of common HPC workload while tracing all the external sources of noise in an optimized way. This was discussed two years ago. The rtla osnoise adds an easy-to-use interface for osnoise, enabling the tracer to the masses. rtla was discussed last year. These tools now are available and in use by members of this community in their daily activities.

But that is just the minimum implementation, and there is lots of work to do. For example:

- The addition of other types of workload - not only reading time
- Include information about processor power usage
- Usage of other types of the clock source
- Inclusion of features to identify the source of IPIs

And so on.

In this discussion, the community is invited to share ideas, propose features and prioritize the TODO list.

I agree to abide by the anti-harassment policy

Yes

Primary author: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)
Presenter: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)
Session Classification: CPU Isolation MC
Track Classification: LPC Microconference: CPU Isolation MC
Contribution ID: 263

Type: not specified

**eBPF & Networking**

I agree to abide by the anti-harassment policy
A BPF map for online packet classification

Tuesday, 13 September 2022 11:00 (30 minutes)

There is a growing need in online packet classification for BPF-based networking solutions. In particular, in cilium we have two use cases: the PCAP recorder for the standalone XDP load balancer and the k8s network policies. The PCAP recorder implementation suffers from slow and dangerous updates due to runtime recompilation, and both use cases require specifying port ranges in rules, which is currently not supported.

At the moment there are two competing algorithms for online packet classification: Tuple Merge and Partition Sort. The Tuple Merge algorithm is using hash tables to store rules, and the Partition Sort is using multi-dimensional Interval trees. Thus, both of algorithms are impossible to implement in “pure” BPF due to lack of functionality and also due to verifier complexity limits.

We propose a new BPF map for packet classification and an API which can be used to adapt this map to different practical use cases. The map is not tied to the use of a specific algorithm, so any of brute force, tuple merge, partition sort or a future state-of-art algorithm can be used.

1 https://cilium.io/blog/2021/05/20/cilium-110/#pcap

I agree to abide by the anti-harassment policy

Yes

Primary author: PROTOPOPOV, Anton (Isovalent)
Presenter: PROTOPOPOV, Anton (Isovalent)
Session Classification: eBPF & Networking
Track Classification: eBPF & Networking Track
BPF Signing and IMA integration

Monday, 12 September 2022 15:00 (30 minutes)

Signing BPF programs has been a long ongoing discussion and there has been some more concrete work and discussions since the BPF office hours talk in June.

There was a BoF session at the Linux security summit in Austin between BPF folks (KP and Florent) and IMA developers (Mimi, Stefan and Elaine) to agree on a solution to have IMA use BPF signatures.

The BPF position is to provide maximum flexibility to the user on how the programs are signed. For this, they way the programs are signed (format, kind of hash) and the way the signature is verified should be up-to the user. IMA is one of the users of BPF signatures.

The goal of this session is to discuss a gatekeeper and signing implementation that works with IMA and the options that are available for IMA and agree on a solution to move forward.

The current kernel convention where IMA hard codes a callback into the security_* hooks is at odds with the BPF philosophy of providing flexibility to the user. But, we do see a common ground that can work the best for BPF, IMA and most importantly, the users.

I agree to abide by the anti-harassment policy
Yes

Primary author:  SINGH, KP (Google)
Presenter:  SINGH, KP (Google)
Session Classification:  eBPF & Networking
Track Classification:  eBPF & Networking Track
Bringing packet queueing to XDP

Tuesday, 13 September 2022 15:00 (30 minutes)

Packet forwarding is an important use case for XDP, however, XDP currently offers no mechanism to delay, queue or schedule packets. This limits the practical uses for XDP-based forwarding to those where the capacity of input and output links always match each other (i.e., no rate transitions or many-to-one forwarding). It also prevents an XDP-based router from doing any kind of traffic shaping or reordering to enforce policy.

Our proposal for adding a programmable queueing layer to XDP was posted as an RFC patch set in July[0]. In this talk we will present the overall design for a wider audience, and summarise the current state of the work since the July series. We will also present open issues, in the hope of spurring discussion around the best way of adding this new capability in a way that is as widely applicable as possible.

[0] https://lore.kernel.org/r/20220713111430.134810-1-toke@redhat.com

I agree to abide by the anti-harassment policy

Yes

Primary author: HØILAND-JØRGENSEN, Toke (Red Hat)

Co-authors: ALFREDSSON, Frey (Karlstad University); Dr HURTIG, Per (Karlstad University); Prof. BRUNSTROM, Anna (Karlstad University)

Presenter: HØILAND-JØRGENSEN, Toke (Red Hat)

Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track
Can the Linux networking stack be used with very high speed applications?

*Monday, 12 September 2022 12:30 (30 minutes)*

Ethernet networking speeds continue to increase - 100G is common today for both NICs and switches, 200G has been available for years, 400G is the current cutting edge with 800G on the horizon. As the speed of the physical layer increases how does S/W scale - specifically, the Linux IP/TCP stack? Is it possible to leverage the increasing line-rate speeds for a single flow? Consider a few example data points about what it means to run at 400 Gbps speeds:

1. the TCP sequence number wraps 12.5 times a second - i.e., wrapping every 80 msec, and
2. at an MTU of 1500B, to achieve 400G speeds the system needs to handle 33M pps - i.e., a packet arrives every 30 nsec (for reference, an IPv4 FIB lookup on a modern Xeon processor takes ~25nsec).

We used an FPGA based setup with an off-the-shelf server and CPU to investigate the Linux networking stack to determine how fast it can be pushed and how well it performs at high rates for a single flow. With this setup tailored specifically to push the kernel’s TCP/IP stack, we were able to achieve a rate of more than 670 Gbps (application data rate) and more than 31 Mpps (different tests) for a single flow. This talk discusses how we achieved those rates, the lessons learned along the way and what it suggests are core requirements for deploying very high speed applications that want to use the Linux networking stack. Some of the topics are well established such as the need for GRO, TSO, zerocopy and a reduction of system calls; others are not so prominent. This talk presents a systematic and comprehensive review of the effect of variables involved and serves as a foundation for future work.

I agree to abide by the anti-harassment policy

Yes

**Primary author:**  AHERN, David

**Presenter:**  AHERN, David

**Session Classification:**  eBPF & Networking

**Track Classification:**  eBPF & Networking Track
Since the early days of eBPF, Cilium’s core building block for its datapath is tc BPF. With more adopters of eBPF in the Kubernetes landscape, there is growing risk from a user perspective that Pods orchestrating tc BPF programs might step on each other, leading to hard to debug problems.

We dive into a recently experienced incident, followed by our proposal of a revamped tc ingress/egress BPF datapath for the kernel which incorporates lessons learned from production use, lower overhead as a framework, and supporting BPF links for tc BPF programs in a native, seamless manner (that is, not conflicting with tc’s object model). In particular the latter solve the program ownership and allow for better debugability through a common interface for BPF. We also discuss our integration approach into libbpf and bpftool, dive into the uapi extensions and next steps.

I agree to abide by the anti-harassment policy

Yes

**Primary authors:** BORKMANN, Daniel (Isovalent); ALEKSANDROV, Nikolay (Isovalent)

**Presenter:** BORKMANN, Daniel (Isovalent)

**Session Classification:** eBPF & Networking

**Track Classification:** eBPF & Networking Track
Closing the BPF map permission loophole

Wednesday, 14 September 2022 11:00 (30 minutes)

While working on github.com/cloudflare/tubular we discovered that it’s possible for a program with CAP_BPF to circumvent file permissions of BPF map fds, effectively making it impossible to enforce read-only access. In our case, a process exporting metrics from maps can’t be prevented from also being able to modify those maps. I will outline how permissions, map flags like BPF_F_RDONLY and map freezing interact and explain how current semantics fall short. I’ll also propose a possible solution which changes how the verifier tracks the mutability of map values.

I agree to abide by the anti-harassment policy
Yes

Primary author:  BAUER, Lorenz
Presenter:  BAUER, Lorenz
Session Classification:  eBPF & Networking
Track Classification:  eBPF & Networking Track
Developing eBPF profiler for polyglot cloud-native applications

Monday, 12 September 2022 17:00 (30 minutes)

One of the important jobs of system-wide profilers is to capture stack traces without requiring recompilation or redeployment of profiled applications. This becomes difficult when the profiler has to deal with the binaries compiled in different languages. Heavy lifting for the stack unwinding is done by the kernel if frame pointers are present or if the binary has ORC - in kernel debug information format information available. Although most modern compilers have an option to scrap the frame pointers for performance gain.

In this talk we will talk about how we are experimenting with using eBPF to extend the existing stack unwinding facility in the Linux kernel. We will discuss how we are walking the stacks of interpreted languages, such as Ruby, as well as runtimes with JITs, like the JVM. And how extending the current stack unwinding facility can be useful for such cases.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  THAKKAR, Vaishali; HONDUVILLA COTO, Javier

Presenters:  THAKKAR, Vaishali; HONDUVILLA COTO, Javier

Session Classification:  eBPF & Networking

Track Classification:  eBPF & Networking Track
Ghost is a kernel scheduling class that allows userspace and eBPF programs, called the “agent”, to control the scheduler.

Following up on last year’s LPC talk, I’ll cover:
- How BPF works in Ghost
- An agent that runs completely in BPF: no userspace scheduling required!
- Implementation details of “Biff”: a bpf-hello-world example scheduler.
- Future work, including CFS-in-BPF, as well as a request for new MAP_TYPEs!

I agree to abide by the anti-harassment policy
Yes

Primary author:  RHODEN, Barret (Google)
Presenter:  RHODEN, Barret (Google)
Session Classification:  eBPF & Networking
Track Classification:  eBPF & Networking Track
eBPF Standardization

*Tuesday, 13 September 2022 13:00 (30 minutes)*

At LSF/MM/BPF, the topic was raised about better documenting eBPF and making “standards” like documentation, especially since we are having runtimes other than just Linux now supporting eBPF.

This presentation will summarize the current state of the eBPF Foundation effort on these lines, how it is organized, and invite discussion and feedback on this topic.

I agree to abide by the anti-harassment policy

Yes

**Primary author:** THALER, Dave (Microsoft)

**Presenter:** THALER, Dave (Microsoft)

**Session Classification:** eBPF & Networking

**Track Classification:** eBPF & Networking Track
FW centric devices, NIC customization

For a long time now the industry has been building programmable processors into devices to run firmware code. This is a long standing design approach going back decades at this point. In some devices the firmware is effectively a fixed function and has little in the way of RAS features or configurability. However, a growing trend is to push significant complexity into these devices processors.

Storage has been doing FW centric devices for a long time now, and we can see some evolution there where standards based channels exist that carry device specific data. For instance, looking at nvme-cli we can see a range of generic channels carrying device specific RAS or configuration (smart-log, fw-log, error-log, fw-download). nvme-cli also supports entire device specific extensions to access unique functionality (nvme-intel, nvme-huawei, nvme-micro-

https://man.archlinux.org/man/community/nvme-cli/nvme.1.en

This reflects the reality that standardization can only go so far. The large amount of FW code still needs RAS and configuration unique to each device’s design to expose its full capability.

In the NIC world we have been seeing FW centric devices for a long time, starting with MIPS cores in early Broadcom devices, entire Linux OS’s in early “offload NICs”, to today’s highly complex NIC focusing on complicated virtualization scenarios.

For a long time our goal with devlink has been to see a similar healthy mix of standards based multi-vendor APIs side by side with device specific APIs, similar to how nvme-clie is handling things on the storage side.

In this talk, we will explore options, upstream APIs and mainstream utilities to enjoy FW-centric NIC customizations.

We are focused on:

1) non-volatile device configuration and firmware update - static and preserved across reboots
2) Volatile device global firmware configuration - runtime.
3) Volatile per-function firmware configuration (PF/VF/SF) - runtime.
4) RAS features for FW - capture crash/fault data, read back logs, trigger device diagnostic modes, report device diagnostic data, device attestation

I agree to abide by the anti-harassment policy

Yes
Primary authors:  GUNTHORPE, Jason (Nvidia); MAHAMEED, Saeed (Nvidia); BLOCH, Mark (Nvidia)

Presenters:  MAHAMEED, Saeed (Nvidia); BLOCH, Mark (Nvidia)

Session Classification:  eBPF & Networking

Track Classification:  eBPF & Networking Track
HID (Human Interface Device) is an old protocol which handles input devices. It is supposed to be standard and to allow devices to work without the need for a driver. Unfortunately, it is not standard, merely "standard".

The HID subsystem has roughly 80 drivers, half of them are fixing only one tiny bit, either in the protocol of the device or in the key mapping for instance.

Historically, fixing such devices requires users to submit a patch to the kernel. The process of updating the kernel has greatly improved over the past few years, but still, we can not safely fix those devices in-place (without rebooting and risking messing up the system).

But here is the future: eBPF. eBPF allows loading kernel-space code from user-space.

Why not apply that to HID devices too? This way we can change the small part of the device that is not working while still relying on the generic processing of the kernel to support it.

In this talk, we will outline this new feature that we are currently upstreaming, its advantages and why this is the future, for HID at least.

I agree to abide by the anti-harassment policy

Yes

Primary author:  TISSOIRES, Benjamin (Red Hat)
Presenter:  TISSOIRES, Benjamin (Red Hat)
Session Classification:  eBPF & Networking
Track Classification:  eBPF & Networking Track
How to share IPv4 addresses by partitioning the port space

Tuesday, 13 September 2022 12:00 (30 minutes)

When establishing connections, a client needs a source IP address. For better or worse, network and service operators often assign traits to client IP addresses such as a reputation score, geolocation or traffic category, e.g. mobile, residential, server. These traits influence the way a service responds.

Transparent Web proxies, or VPN services, obfuscate true client IPs. To ensure a good user experience, a transparent proxy service should carefully select the egress IPs to mirror the traits of the true-client IP.

However, this design is hard to scale in IPv4 due to the scarcity of IP addresses. As the price of IPv4 addresses rise, it becomes important to make efficient use of the available public IPv4 address pool.

The limited pool of IPv4 addresses, coupled with a desire to express traits known to be used by other services, presented Cloudflare with a challenge: The number of server instances in a single Point of Presence exceed the number of IPv4 egress addresses available – a disconnect that is exacerbated by the need to partition available addresses further according to traits.

This has led us to search for ways to share a scarce resource. The result is a system where a single egress IPv4 address, with given traits, is assigned to not one, but multiple hosts. We make it possible by partitioning ephemeral TCP/UDP port space and dividing it among the hosts. Such a setup avoids use of stateful NAT, which is undesirable due to scalability and single-point-of-failure concerns.

From previous work[1] we know that the Linux Sockets API is poorly suited to a task of establishing connections from a given source port range. Opening a TCP connection from a port range is only possible if the user re-implements the free port search - a task that the Linux TCP stack already performs when auto-binding a socket.

On UDP sockets, selecting source port range for a connected socket turns out to be very difficult. Correctly dealing with connected sockets is important because they are a desirable tool for egress traffic, despite their memory overhead. Currently, the Linux API forces the user to choose: Either use a single connected UDP socket owning a local port, which greatly limits the number of concurrent UDP flows; or, alternatively, somehow detect a connected-socket conflict when creating connected UDP sockets, which share the local address.

We previously built a detection mechanism with a combination of querying sock_diag and toggling the port sharing on and off after binding the socket[1]. Depending on perspective, the process might be described by some as arduous, or by others as an ingenious hack that works.

Recent innovations such as these demonstrate that sharing the finite set of ports and addresses among larger sets of distributed processes is a problem not yet completely solved for the Linux Sockets API. At Cloudflare we have come up with a few different ideas to address the shortcomings of the Linux API. Each of them makes the task of sharing an IPv4 address between servers and/or processes easier, but the degree of user-friendliness varies.

In no particular order, the ideas we have evaluated are:

1. Introduce a per-socket configuration option for narrowing down the IP ephemeral port range.
2. Introduce a flag to enforce unicast semantics for connected UDP sockets, when sharing the local address (SO_REUSEADDR). With the flag set, it should not be possible to create two connected UDP sockets with conflicting 4-tuples ({local IP, local port, remote IP, remote port}).

3. Extend the late-bind feature (IP_BIND_ADDRESS_NO_PORT) to UDP sockets, so that dynamically-bound connected UDP sockets can share a local address as long as the remote address is unique.

4. Extend Linux sockets API to let the user atomically bind a socket to a local and a remote address with conflict detection. Akin to what the Darwin connectx() syscall provides.

5. Introduce a post-connect() BPF program to allow user-space processes to prevent creation of connected UDP sockets with conflicting 4-tuples.

During the talk, we will go over the challenges of designing a distributed proxy system that mirrors client IP traits, which led us to look into IP sharing and port space partitioning.

Then, we will shortly explain production tested implementation of TCP/UDP port space partitioning using only existing Linux API features.

Finally, we will describe the proposed API improvement ideas, together with their pros and cons and implementation challenges.

We will accompany the most promising, according to our judgment, ideas with a series of RFC patches posted prior to the talk for the upstream community consideration.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  SITNICKI, Jakub (Cloudflare); MAJKOWSKI, Marek (Cloudflare)

Presenters:      SITNICKI, Jakub (Cloudflare); MAJKOWSKI, Marek (Cloudflare)

Session Classification:  eBPF & Networking

Track Classification:  eBPF & Networking Track
**Linux from power-reset: status, challenges, and opportunities**

In the last few years, the open source host firmware communities made great progress on the coreboot/LinuxBoot stack. Now it is ready for prime time. With this stack, Linux developers gain total control of host firmware (aka. BIOS): host firmware bugs can be tackled as easily as Linux bugs; host firmware can be developed together with kernel/tools to enable new hardware technologies and/or to achieve/improve holistic software solutions.

In the meantime, there are a few pain points that Linux community’s feedback / collaborations are desired for. For example, the compatibility and stability of kexec; potential improvements to PCIe or other subsystems, etc.

In this discussion, the design, ecosystem status and pain points of the coreboot/LinuxBoot stack is presented first, to seed the discussions on how to solve the challenges and how to take advantage of the opportunities with Linux from power-reset.

**I agree to abide by the anti-harassment policy**

Yes

**Primary authors:**  Mr MINNICH, Ron (Google); ZHANG, Jonathan

**Presenters:**  Mr MINNICH, Ron (Google); ZHANG, Jonathan

**Session Classification:**  eBPF & Networking

**Track Classification:**  eBPF & Networking Track
Live coding eBPF with streaming abstractions

Monday, 12 September 2022 18:00 (30 minutes)

eBPF gives us an extraordinary amount of power, allowing to attach custom programs to many subsystems in the kernel. We use it to build lots of helpful observability and security tools, but there is a problem: eBPF is hard and nonintuitive for developers who are not familiar with low-level programming concepts.

In this talk, we will discuss a novel approach to writing programs for the eBPF virtual machine, building on a new set of abstractions borrowed from streaming databases, functional programming, and visual live coding environments. We will see how such abstractions can help us to simplify our development workflows, allowing us to build tools from a set of visual composable blocks. A prototype of a new open source integrated development environment for eBPF, Metalens, will be demonstrated.

I agree to abide by the anti-harassment policy

Yes

Primary author: BAKSALYAR, Nikita

Presenter: BAKSALYAR, Nikita

Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track
Netlink is a TLV based protocol we invented and use in networking for most of our uAPI needs. It supports seamless extensibility, feature discovery and has been hardened over the years to prevent users from falling into uAPI extensibility gotchas.

Nevertheless netlink remains very rarely unused outside of networking. It’s considered arcane and too verbose (requires defining operations, policies, parsers). (The fact it depends on CONFIG_NET doesn’t help either but that’s probably just an excuse most of the time.)

In an attempt to alleviate those issues I have been working on creating a netlink protocol description in YAML. A machine readable netlink message description should make it easy for language bindings to be automatically generated, making netlink feel much more like gRPC, Thrift or just a function call in the user space. Similarly on the kernel side the YAML description can be used to generate the op tables, policies and parsers.

In this talk I’ll cover the basics of netlink (which everyone claims to know but doesn’t), compare it to Thrift/gRPC, and present the YAML work.

I agree to abide by the anti-harassment policy

Yes

Primary author:  KICINSKI, Jakub (Meta)
Presenter:  KICINSKI, Jakub (Meta)
Session Classification:  eBPF & Networking
Track Classification:  eBPF & Networking Track
More CO-RE? Taming the effects of compiler optimizations on BPF tracing

Wednesday, 14 September 2022 12:00 (30 minutes)

BPF Compile Once - Run Everywhere (CO-RE) is a massive help when writing BPF programs that work across kernel versions, especially in the observability space, where we are often at the mercy of internal kernel changes in data structures and the like. However when writing BPF tracing programs, a major pain point is compiler optimizations which often mean the function - though not inlined - is not present in BPF Type Format (BTF), and thus cannot be traced easily via BPF. Worse, simply adding such functions to pahole would often result in them having the wrong arguments, as some of the arguments DWARF describes (and which we would translate into BTF) are optimized out. All of this creates particular problems when trying to maintain BPF tracing programs across kernel versions, because a simple compiler update can make a function have a different name (a suffix is added) and it also effectively disappears from BTF, so we lose the ability to trace via high-performance fentry/fexit programs. Here we examine this problem and propose a potential solution that may resolve it.

I agree to abide by the anti-harassment policy

Yes

Primary author: MAGUIRE, Alan (Oracle)
Presenter: MAGUIRE, Alan (Oracle)
Session Classification: eBPF & Networking
Track Classification: eBPF & Networking Track
MPTCP: Extending kernel functionality with eBPF and Netlink

Tuesday, 13 September 2022 17:30 (30 minutes)

Multipath TCP (MPTCP) was initially supported in v5.6 of the Linux kernel. In subsequent releases, the MPTCP development community has steadily expanded from the initial baseline feature set to now support a broad range of MPTCP features on the wire and through the socket and generic Netlink APIs.

With core MPTCP functionality established, our next goal is to make MPTCP more extensible and customizable at runtime. The two most common tools in the kernel’s networking subsystem for these purposes are generic Netlink and BPF. Each has tradeoffs that make them better suited for different scenarios. Our choices for extending MPTCP show some of those tradeoffs, and also leave our community with some open questions about how to best use these interfaces and frameworks.

This talk will take MPTCP as a use-case to illustrate questions any network subsystems could have when looking at extending kernel functionality and controls from the userspace. Two main examples will be presented: one where BPF seems more appropriate and one where a privileged generic Netlink API can be used.

As one example, we are extending the MPTCP packet scheduler using BPF. When there are multiple active TCP subflows in a MPTCP connection, the MPTCP stack must decide which of those subflows to use to transmit each data packet. This requires low latency and low overhead, and direct access to low-level TCP connection information. Customizable schedulers can optimize for latency, redundancy, cost, carrier policy, or other factors. In the past such customization would have been implemented as a kernel module, with more compatibility challenges for system administrators. We have patches implementing a proof-of-concept BPF packet scheduler, and hope to discuss with the netdev/BPF maintainers and audience how we might best structure the BPF/kernel API – similar to what would be done for a kernel module API – to balance long-term API stability, future evolution of MPTCP scheduler features, and usability for scheduler authors.

The next customization feature is the userspace path manager added in v5.19. MPTCP path managers advertise addresses available for multipath connections, and establish or close additional TCP subflows using the available interfaces. There are a limited number of interactions with a path manager during the life of a MPTCP connection. Operations are not very sensitive to latency, and may need access to a restricted amount of data from userspace. This led us to expand the MPTCP generic Netlink API and update the Multipath TCP Daemon (mptcpd) to support the new commands. Generic Netlink has been a good fit for path manager commands and events, the concepts are familiar and the message format makes it possible to maintain forward and backward compatibility between different kernel versions and userspace binaries. However the overhead of userspace communication does have tradeoffs, especially for busy servers.

MPTCP development for the Linux kernel and mptcpd are public and open. You can find us at mptcp@lists.linux.dev, https://github.com/multipath-tcp/mptcp_net-next/wiki (soon via https://mptcp.dev), and https://github.com/intel/mptcpd

I agree to abide by the anti-harassment policy

Yes
Primary authors: MARTINEAU, Mat (Intel); BAERTS, Matthieu (Tessares)
Presenter: BAERTS, Matthieu (Tessares)
Session Classification: eBPF & Networking
Track Classification: eBPF & Networking Track
Networking resource control with per-cgroup LSM

Google’s container management system runs different workloads on the same host. To effectively manage networking resources, the kernel has to apply different networking policies to different containers.

Historically, most of the networking resource control happened inside proprietary Google networking cgroup. That cgroup is an interesting cross between upstream net_cls and net_prio, has a lot of Google-specific business logic and has no chance of being accepted upstream.

In this talk I’m going to talk about what we’d like to manage on the networking resource side and which BPF mechanisms were added to achieve this (lsm_cgroup).

I agree to abide by the anti-harassment policy

Yes

Primary author:  FOMICHEV, Stanislav (Google)
Co-author:  BANDEWAR, Mahesh
Presenter:  FOMICHEV, Stanislav (Google)
Session Classification:  eBPF & Networking
Track Classification:  eBPF & Networking Track
Case for OPENED for eBPF NF Development

The recent past has been the emergence of eBPF in building high performance networking usecases such as load balancing, K8s CNI, DDoS protection, traffic shaping etc. However, unlike traditional software datapath technologies, eBPF code development exhibits enormous heterogeneity in terms of choice of kernel hook points, data sharing mechanisms as well as kernel loading tools. Today, these decisions are made at code development time; however, to be truly effective such decision must be made holistically using information about other eBPF programs running on the server.

We argue that the developer of an network function (NF) (consisting of multiple eBPF functions) has no idea of the other NFs that will be chained together at run time to create the datapath. Hence, decisions taken at the development stage are bound to be suboptimal. A solution for this problem can be taking eBPF specific decisions (such as hook point) at run-time. Unfortunately the process of altering design choices at run time is non-trivial due to two properties of the eBPF runtime. First, porting code written for one hook point to another requires modification in terms of input data structures and available bpf\_helper functions. Second, deciding the optimal and most efficient combination eBPF specific decisions (e.g., data structures) requires exploring a large number of design choices.

For example, porting and reusing existing functionalities, say GUE encap/decap processing from Meta’s Katran code base, in a new program would require isolating the GUE specific functionalities and their associated control and data dependencies, and modifying them for use in the new program. This process requires complete understanding of the program, is time consuming and typically tends to be error prone.

The porting task is further complicated in eBPF due to its heterogeneity that prevents code written for one hook point from generally being able to run at a different hook point. For example, consider an observability program parsing packet headers and updating counters, that is written in XDP chained with a TC program that also parses headers and performs QoS enforcement. To avoid duplication of parsing, the developer might want to move the observability program to the TC hook point, chain it with the QoS enforcement TC program and share parsed packet headers between both modules. However, without appropriate transformations, XDP code cannot be run at TC.

This difficulty of porting also leads to large eBPF projects working in (strong) siloes and not reusing similar functionalities available in other production grade open source projects. A documented example of such behavior is the decision of the Cloudflare team to develop their own load balancer code, Unimog, instead of reusing Meta’s Katran load balancer. A consequence of this difficulty to port eBPF code is that a typical eBPF solution is built as a monolith consisting of a number of tightly coupled eBPF programs.

Clearly such siloed and monolithic developer community does not augur well for both a) wider eBPF adoption as new developers will either have to rewrite readily available modules or reuse the entire code base, a choice that will likely introduce unnecessary bloat and overheads to their solution. As well as for b), future innovation as developers will waste effort in adding implementations of similar functionalities in same language!) in their siloed codebases, resulting in replication of effort instead of combining forces innovating on the newer paradigm changing design options that eBPF introduces.

While there have been efforts like BTF CORE to streamline deployment of eBPF code across different kernel versions, very little effort exists in making eBPF code reusable amongst codebases. In particular, recent efforts such as Walmart’s L3AF requires rewriting code to use tail calls and is further limited to programs of the same type. We believe that demonstrating the feasibility of
a general approach to transforming eBPF NF code built with certain (development time) eBPF design choices to run time optimal choices based on actual datapath requirements is a key first step towards breaking developer silos and fostering concerted innovation in eBPF based datapath technology.

This motivates us to create tooling that enables 1) automated extraction of specific eBPF code pieces from different projects, 2) hook point specific transformation that facilitates running code written for one hook point against a target hook point and 3) composition of multiple programs to create the necessary pipeline. We envision a world, where NF developers can pick and choose functionalities from different projects and compose them together to build flexible and high performance network datapaths. In this paper, we describe OPENED, a tool that supports extracting specific code functionalities from a given project, transforming them for running at the desired target hook point and composing them together to build flexible packet processing pipelines.

Workflow

Our tool has a three stage workflow corresponding to three major tasks for consuming third party code in one’s project, viz., a) Extraction, b) Transformation and c) Composition. Each of the stages, in turn, consists of a multi-step user-in-the-loop workflows to inform and guide the tool in making appropriate decisions. The input to the system consists of a yaml specification describing the required information for all three stages.

Stage 1: Extraction

For the first stage of extracting code, the specification provides an array of network functions of source code, in the form [URI:kernel_ebpf_code_repository, URI:file_path:line number] of function definitions. For example, the “xdpdecap” function in Katran will be specified as: [github.com/facebookincubator/katran/blob/main/katran/decap/bpf/decap_kern.c:223]. Given this input, our current prototype computes the Minimal Compilable Unit (MCU), i.e. the minimal set of source artefacts e.g. source files, configurations, data sources, build files etc. in third party code base which when taken together will successfully compile, and be able to load and execute in the kernel at the same kernel hook point. The automated extraction of MCU involves identifying both control and data dependencies (in the form of eBPF map updates and look ups) amongst functions. Our prototype extends Codequery tool which provides a sqlite db with querying capabilities on top of CTAGS and Cscope indices of the entire codebase. We extend codequery to determine the function call graph of our extraction target and the functions called by them recursively. We stop exploration of function call graph once the called functions are defined in standard system libraries.

The output of the tool is two JSON arrays corresponding to the list of all functions along with their location (file, start and end line numbers) inside code_repository and the definitions of various maps which are utilized in the code. Our tool also generates two types of warning results for which it needs user-in-the-loop intervention a) specific instances of global maps for which definitions was not found in the source code inside code_repository (maps which are instantiated in user code), b) list of functions for which multiple declarations with same call signature are found. For warnings of first type, the user is expected to ensure that map instantiation (inside user code) is also done for the ported instance. For the second type of warnings the user needs to keep the right function call details and remove the details of duplicates. A simple program then copies all the selected files and map declarations to create a new source file for MCU, which is then compiled and loaded at the original hookpoint to complete the extraction stage.

Stage 2: Transformation

For the hook point transformation, the input yaml specifies the target hook point for the function extracted earlier. Hook point transformation is implemented using source code transformation tools viz. coccinelle and TXL that allow developers to express matching patterns/rules in source code and their corresponding code level transformations. Our choice of using source code transformation tools, as opposed to byte code level transformation is motivated by the need for developers to maintain/debug source code repositories over longer time periods. Source code transformation tools seem to be sufficient for most of the use cases we have encountered so far. For instance, for
XDP to TC transformation, we need to replace XDP decisions such as XDP_PASS and XDP_DROP with corresponding TC actions such as TC_ACT_OK(PIPE) and TC_ACT_DROP. Similarly, we need rules to replace byte offsets such as ethernet header protocol (ethhdr->h_proto) value with corresponding skb struct field (skb->protocol) accesses. Similarly, we require rules that can transform bpf helper functions across hook points. Based on our experiments with large open source code repositories, we find that the combination of coccinelle and TXL is sufficient for our transformation rules. We would also point out, that not all pairs of hook point transformations are feasible, starting from source code, for instance due to the unavailability of corresponding helper functions, e.g. bpf_msg_push_data at XDP layer. In this case, the tool will raise an error that the transformation is not feasible due to missing helper functions or lack of available kernel state (e.g. connection tracking state at XDP). To enable universal hook point transformation, there is a need for a domain specific language (DSL) where the developer expresses packet processing operations at a high level, that are then compiled down to eBPF hook point specific programs. We leave the design of such a DSL to future work. At the end of the transformation stage, we verify that the transformed code is semantically equivalent to the original code by running and verifying program output against function specific unit test cases extracted from the third party codebase.

**Stage 3: Composition**

In the composition stage, the user-in-the-loop input is the order in which the (multiple) eBPF programs for a given interface at different hookpoints. To this end, transformed eBPF programs are chained together using hook point specific mechanisms such as `libxdp` (for multiple XDP progs) or TC multi-prog (for TC), or using generic tail calls for hook points that do not provide specific mechanisms for program chaining.

**Status**

Our current prototype is able to transform XDP programs to TC compatible programs and we have validated results on a variety of opensource codebases viz. xdp tutorial, Mizar, suricata xdp filter and Meta’s Katran. The prototype is written in 425 LoC of C++ code. We currently have seven rules for transforming the TC compatible programs. For the largest program, Katran, our tool took ~500ms. One of the many instances of the user-in-the-loop intervention that we observed while running our tool on Katran was: during the extraction phase, our tool identified many functions defined in multiple files and required the user to determine which to keep (e.g., process_packet).

**I agree to abide by the anti-harassment policy**

Yes

**Primary authors:**  Prof. BENSON, Theophilus (Brown University); Dr KODESWARAN, Palanivel (IBM Research); Dr SEN, Sayandeep (IBM Research)

**Presenters:**  Prof. BENSON, Theophilus (Brown University); Dr KODESWARAN, Palanivel (IBM Research); Dr SEN, Sayandeep (IBM Research)

**Session Classification:**  eBPF & Networking

**Track Classification:**  eBPF & Networking Track
Overview of the BPF networking hooks and user experience in Meta

Monday, 12 September 2022 13:00 (30 minutes)

BPF has grown rapidly. In the networking stack, a BPF program can do much more than a few years ago. It could be overwhelming to figure out which bpf hook should be used, what is available at a particular layer and why. This talk will go through some of the bpf hooks in the networking stack with use cases in Meta. The talk will also get to some common questions/confusions that the users have and how they could be addressed in the future.

I agree to abide by the anti-harassment policy

Yes

Primary author: LAU, Martin (Meta)
Presenter: LAU, Martin (Meta)
Session Classification: eBPF & Networking
Track Classification: eBPF & Networking Track
Percpu hashtab traversal measurement study

Tuesday, 13 September 2022 18:00 (30 minutes)

As platforms grow in cpu count (200+ cpu), using per cpu data structures is becoming more and more expensive. Copying the percpu data from the bpf hashtab map to userspace buffers can take up to 22 us per entry on a platform with 256 cores.

This talk presents a detailed measurement study of the cost of percpu hashtab traversal, covering various methods and systems with core counts.
We will discuss how the current implementation of this data structure makes it hard to amortize cache misses, and solicit proposal for possible enhancements.

I agree to abide by the anti-harassment policy
Yes

Primary author: VAZQUEZ, Brian (Google)
Co-authors: RIZZO, Luigi (Google); FOMICHEV, Stanislav (Google); SHARMA, Sagarika (Columbia University)
Presenter: VAZQUEZ, Brian (Google)
Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track
Performance insights into eBPF step by step

**Monday, 12 September 2022 17:30 (30 minutes)**

Having full visibility throughout the system you build is well established best practice. Usually one knows which metrics to collect, how and what to profile or instrument to understand why the system exhibits this level of performance. All of this becomes more challenging as soon as eBPF layer is included.

In this talk Dmitrii shed some light on those bits of your service that use eBPF, step by step with topics such as:

- How to collect execution metrics of eBPF programs?
- How can we profile these eBPF programs?
- What are the common pitfalls to avoid?

The talk will provide the attendees with an approach to analyze and reason about eBPF programs performance.

**I agree to abide by the anti-harassment policy**

Yes

**Primary author:** DOLGOV, Dmitrii (Red Hat)

**Presenter:** DOLGOV, Dmitrii (Red Hat)

**Session Classification:** eBPF & Networking

**Track Classification:** eBPF & Networking Track
Pressure feedback for LRU map types

Currently, the BPF API for the LRU map type does not give any indication about when insertions into the map result in the eviction of another entry. This session is to discuss use cases when it would be useful to measure LRU eviction in order to provide insight into load and to tweak control plane behaviour. With this insight we can look at a proposal to make this possible through the BPF API.

I agree to abide by the anti-harassment policy

Yes

Primary author: STRINGER, Joe (Isovalent)
Presenter: STRINGER, Joe (Isovalent)
Session Classification: eBPF & Networking
Track Classification: eBPF & Networking Track
So we have the BPF CI, managed by Meta. It picks up patches from Patchwork, turns them into Pull Requests on GitHub, and through the GitHub Actions CI/CD framework runs the selftests with these patches on dedicated runners.

Thanks to this architecture, it is relatively easy to create Pull Requests and run the CI on another Linux repository on GitHub. However, the CI is being worked on and is susceptible to change, and its components have not been designed with reuse in mind, which currently makes it difficult to create robust workflows hooking on the existing infrastructure.

This session is to explore and discuss the possibilities to improve reusability. Use cases would be for developers to validate their patches, or for organisations/projects to detect regressions.

I agree to abide by the anti-harassment policy

Yes

Primary author: MONNET, Quentin (Isovalent)
Presenter: MONNET, Quentin (Isovalent)
Session Classification: eBPF & Networking
Track Classification: eBPF & Networking Track
Revisiting eBPF Seccomp Filters

Monday, 12 September 2022 15:30 (30 minutes)

Seccomp, the widely used system-call security module in Linux, is among the few that still exposes classic BPF (cBPF) as the programming interface, instead of the modern eBPF. Due to the limited programmability of cBPF, today’s Seccomp filters mostly implement static allow-deny lists. The only way to implement advanced policies is to delegate them to user space (e.g., Seccomp Notify); however, such an approach is error prone due to time-of-check time-of-use issues and costly due to the context switch overhead.

Over the past several years, supporting eBPF filters in Seccomp has been brought up (e.g., by Dhillon [1], Hromatka [2] and our team [3] and has raised many offline discussions on the mailing lists [4]. However, the community has not been convinced that eBPF for Seccomp is 1) necessary nor 2) safe, with opinions like “Seccomp shouldn’t need it…” and “rather stick with cBPF until we have an overwhelmingly good reason to use eBPF…” preventing its inclusion.

We have developed a full-fledged eBPF Seccomp filter support and systematically analyzed its security [5]. In the proposed presentation, using the insight from our system, we will (1) summarize and refute concerns on supporting eBPF Seccomp filters, (2) present our design and implementation with a holistic view, and (3) open the discussion for the next steps.

Specifically, to show that eBPF for Seccomp is necessary, we describe several security features we build using eBPF Seccomp filters, the integration with container runtime like crun, and performance benchmark results. To show that it is safe, we further describe the use of root-only eBPF Seccomp in container-based use cases, which strictly obey current kernel security policies and still improve the usefulness of Seccomp. Further, we will go over the key designs for security, including protecting kernel data, maintaining consistent helper function capability, and the potential integration with IMA (the integrity measurement architecture).

Finally, we will discuss future opportunities and concerns with allowing unprivileged eBPF Seccomp and possible avenues to address these concerns.

Reference:
https://groups.google.com/g/libseccomp/c/pX6QkVf0F74/m/ZUJlwI5qAwAJ

I agree to abide by the anti-harassment policy
Yes
**Presenters:** JIA, Jinghao (University of Illinois Urbana-Champaign); Prof. XU, Tianyin (University of Illinois at Urbana-Champaign)

**Session Classification:** eBPF & Networking

**Track Classification:** eBPF & Networking Track
Socket termination for policy enforcement and load-balancing

Tuesday, 13 September 2022 17:00 (30 minutes)

Socket termination for policy enforcement and load-balancing

Cloud-native environments see a lot of churn where containers can come and go. We have compelling use cases like eBPF enabled policy enforcements and socket load-balancing, where we need an effective way to identify and terminate sockets with active as well as idle connections so that they can reconnect when the remote containers go away. Cilium\(^1\) provides eBPF based socket load-balancing for containerized workloads, whereby service virtual ip to service backend address translation happens only once at the socket connect calls for TCP and connected UDP workloads. Client applications are likely to be unaware of the remote containers that they are connected to getting deleted. Particularly, long running connected UDP applications are prone to such network connectivity issues as there are no TCP RST like signals that the client containers can rely on in order to terminate their sockets. This is especially critical for Envoy-like proxies\(^2\) that intercept all container traffic, and fail to resolve DNS requests over long-lived connections established to stale DNS server containers. The other use case for forcefully terminating sockets is around policy enforcement. Administrators may want to enforce policies on-the-fly whereby they want active client applications traffic to be redirected to a subset of containers, or optimize DNS traffic to be sent to node-local DNS cache containers\(^3\) for JVM-like applications that cache DNS entries.

As we researched ways to filter and forcefully terminate sockets with active as well idle connections, we considered various solutions involving the recently introduced BPF iterator, sock_destroy API, and VRFs that we plan to present in this talk. Some of these APIs are network namespace aware, which need some book-keeping in terms of storing containers metadata, and we plan to send kernel patches upstream in order to adapt them for container environments. Moreover, sock_destroy API was originally introduced to solve similar problems on Android, but it’s behind a special config that’s disabled by default. With the VRF approach to terminate sockets, we faced issues with sockets ignoring certain error codes. We hope our experiences, and discussion around the proposed BPF kernel extensions to address these problems help the community.

---

1. [https://github.com/cilium/cilium](https://github.com/cilium/cilium)
2. [https://github.com/envoyproxy/envoy](https://github.com/envoyproxy/envoy)
3. [https://kubernetes.io/docs/tasks/administer-cluster/nodelocaldns/](https://kubernetes.io/docs/tasks/administer-cluster/nodelocaldns/)

---

I agree to abide by the anti-harassment policy

Yes

Primary author: GHAG, Aditi (Isovalent)

Presenter: GHAG, Aditi (Isovalent)

Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track
State of kprobes/trampolines batch attachment

Monday, 12 September 2022 16:00 (30 minutes)

There’s ongoing effort to speed up attaching of multiple probes, which resulted in new bpf ‘kprobe_multi’ link interface. This allows fast attachment of many kprobes (thousands) and is now supported for example in bpftrace.

Similar interface is being developed also for trampolines, but it’s bit more bumpy road than for kprobes for various reasons.

I’ll shortly sum up multi kprobe interface and some of its current users, and mainly focus on state of the trampoline multi attach API changes.

I agree to abide by the anti-harassment policy

Yes

Primary author: OLSA, Jiri (Isovalent)
Presenter: OLSA, Jiri (Isovalent)
Session Classification: eBPF & Networking
Track Classification: eBPF & Networking Track
The journey of BPF from restricted C language towards extended and safe C.

Monday, 12 September 2022 10:00 (30 minutes)

BPF programs can be written in C, Rust, Assembly, and even in Python. The majority of the programs are in C. The subset of C usable to write BPF programs was never strictly defined. It started very strict. Loops were not allowed, global variables were not available, etc. As BPF ecosystem grew the subset of C language became bigger. But then something interesting happened. The C language itself became a limiting factor. Compile Once Run Everywhere technology required new language features and the intrinsics were added. Then the type tagging was added. More compiler and language extensions are being developed. BPF programs are now written in what can be considered a flavor of C language. The C language is becoming a safe C. Other languages rely on garbage collection (like golang) or don’t prevent memory leaks (like C or C++) the extended and safe C is addressing not only this programmer’s concern, but other bugs typical in C code. This talk will explore whether BPF’s safe C one day will become the language of choice for the core kernel and user space programs.

I agree to abide by the anti-harassment policy

Yes

Primary author: STAROVOITOV, Alexei (Meta)
Presenter: STAROVOITOV, Alexei (Meta)
Session Classification: eBPF & Networking
Track Classification: eBPF & Networking Track
The revival of the learning-sync bridgeport flag

Wednesday, 14 September 2022 13:00 (30 minutes)

Originally defined for the switchdev model, learning_sync is now used
to create a Linux bridge that provides a network interface that merges two
very different fabrics.

In this talk you will learn
- About the motivation for our usecase.
- How we converged a vanilla network segment with a high performance fabric,
  that connects only subsets of the segment.
- Why we chose the Linux bridge to do so.
- How we used device-to-bridge and bridge-to-device learning and
  how little we had to extend our device driver to do so.
- What remains to be done.

I agree to abide by the anti-harassment policy

Yes

Primary author:  WINTER, Alexandra (IBM)
Presenter:  WINTER, Alexandra (IBM)
Session Classification:  eBPF & Networking

Track Classification:  eBPF & Networking Track
Tuning Linux TCP for data-center networks

Monday, 12 September 2022 12:00 (30 minutes)

For better or worse, TCP remains the main transport of many hyperscale data-center networks. Optimizing TCP has been a hot topic in both academic research and industry R&D. However individual research paper often focuses on solving a specific problem (e.g. congestion control for data-center incast) and the industry solutions are often not public or generically applicable. Since Linux TCP default configurations are more or less tuned for wide-area Internet settings, it’s not easy to tune Linux TCP for low-latency data-center environments. For example, simply changing the congestion control to the well-known “dctcp” congestion control may not fully deliver all the benefits Linux TCP can provide.

In this talk, we’d like to share our knowledge and best practices after a decade-long experience of tuning TCP for data-center networks and applications, covering congestion control, protocol and IO enhancements. We will discuss the trade-offs among latency, utilization, CPU, memory, and complexity. In addition we’ll present the inexpensive instrumentation to trace application frame-aware latency beyond general flow level statistics. It’s worth emphasizing that the goal is not to promote the authors’ own works but to help promote interesting discussions with other data center networking developers and guide new comers. After the meeting we hope to synthesize our recommendations into Documentation/networking/tcp_datacenter.txt

I agree to abide by the anti-harassment policy

Yes

Primary authors: CHENG, Yuchung (Google); CARDWELL, Neal (Google)

Presenter: CHENG, Yuchung (Google)

Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track
The idea for XDP-hints, which is XDP gaining access HW offload hints, dates back to Nov 2017. We believe the main reason XDP-hints work have stalled are that upstream we couldn’t get consensus on the layout of the XDP metadata. BTF was not ready at that time.

We believe the flexibility of BTF can resolve the layout issues, especially since BTF have evolved to include support for kernel modules.

This talk is for hashing out upstream XDP-hints discussions and listening to users/consumers of this facility.

There are multiple users of this facility that all needs to be satisfied:

1. **BPF-progs** first obvious consumer (either XDP or TC hooks)
2. **XDP to SKB** conversion (in veth and cpumap) for traditional HW offloads
3. **AF_XDP** can consume BTF info in userspace to decode metadata area
4. **Chained** BPF-progs can communicate state via metadata

---

**I agree to abide by the anti-harassment policy**

Yes

**Primary author:**  BROUER, Jesper Dangaard (Red Hat)

**Presenter:**  BROUER, Jesper Dangaard (Red Hat)

**Session Classification:**  eBPF & Networking

**Track Classification:**  eBPF & Networking Track
Hotel room sharing

- Clayton Hotel Ballsbridge('), Sep 11th till the 16th (Sun till Fri) – I have a two-bed room just for myself; if anyone has trouble finding a place to crash for LPC/OSS EU and interested in sharing the room just contact me: Thorsten Leemhuis, linux@leemhuis.info (¹) that’s not the Clayton where LPC will be!

I agree to abide by the anti-harassment policy
IoTs a 4-Letter Word MC

The IoT microconference is back for its fourth year and our Open Source HW / SW / FW communities are productizing Linux and Zephyr in ways that we have never seen before.

A lot has happened in the last year to discuss and bring forward:
- The Zephyr Project released LTSv2
- PyFive went from concept to an Open Source Silicon SoC via Google’s eFabless shuttle. How will Open Source Silicon affect IoT?
- Linux-wpan gains mac802154 support for scanning, beaoning and coordinator handling
- The Sound Open Firmware (SoF) project combines Linux ASOC drivers with Zephyr audio DSP firmware
- The Oniro Project breaks ground with a Matter + OpenThread gateway blueprint
- A Matter open source project is being actively developed.
- Linux-and-Zephyr-based products built with Yocto.

Each of the above items were large efforts made by Linux centric communities actively pushing the bounds of what is possible in IoT.

Whether you are an apprentice or master, we welcome you to bring your plungers and join us for a deep dive into the pipework of Linux IoT!

I agree to abide by the anti-harassment policy

Primary authors: FRIEDT, Christopher (Friedt Professional Engineering Services); SCHMIDT, Stefan

Presenters: FRIEDT, Christopher (Friedt Professional Engineering Services); SCHMIDT, Stefan

Track Classification: LPC Microconference Track (CLOSED)
Abusing zephyr and meta-zephyr

Wednesday, 14 September 2022 10:55 (25 minutes)

This talk will talk about the work done to switch from cmake to west in meta-zephyr and how I leveraged this work to do bad things with zephyr and meta-zephyr to generate Yocto Project machine definitions for meta-zephyr. We’ll discuss why these patches are not zephyr upstreamable and why autogenerated machine definitions are not included in meta-zephyr.

I agree to abide by the anti-harassment policy

Yes

Primary author: NÍ FHLANNAGÁIN, Eilís (Oniro Project)
Presenter: NÍ FHLANNAGÁIN, Eilís (Oniro Project)
Session Classification: IoTs a 4-Letter Word MC
Track Classification: LPC Microconference: IoTs a 4-Letter Word MC
All types of wireless in Linux are terrible and why the vendors should feel bad

Wednesday, 14 September 2022 12:35 (25 minutes)

The wireless experience in Linux is terrible, whether it be 802.11, bluetooth or one of the other random standards we support? Why is it so bad? One word... Vendors! Vendors do the bare minimum, regress for “stable” users, either rarely or never update or even ship appropriate firmware and expect us to just accept it! This is the perspective of a linux-firmware maintainer for a distribution that has a key role working on Edge and IoT. How can we help vendors (or require them) to improve the wireless firmware user experience in Linux?

I agree to abide by the anti-harassment policy

Yes

Primary author:  ROBINSON, Peter (Red Hat)
Presenter:  ROBINSON, Peter (Red Hat)
Session Classification:  IoTs a 4-Letter Word MC
Track Classification:  LPC Microconference: IoTs a 4-Letter Word MC
libgpiod V2: New Major Release with a Ton of New Features

Wednesday, 14 September 2022 11:20 (30 minutes)

The linux GPIO subsystem exposes a character device to the user-space that provides a certain level of control over GPIO lines. A companion C library (along with command-line tools and language bindings) is provided for easier access to the kernel interface. The character device interface has been rebuilt last year with a number of new ioctl()s and data structures that improve the user experience based on feedback and feature requests that we received since the first release in 2016.

Now libgpiod has been entirely rewritten to leverage the new kernel features and fix issues present in the previous API. The new interface breaks compatibility and requires a different programmatic approach but we believe it is a big improvement over v1. The goal of this talk is to present the new version of the library, reworked command-line tools and high-level language bindings. We will go over the software concepts used in the new architecture and describe new features that provide both a more fine-grained control over GPIOs as well as expose more detailed information about interrupts.

I agree to abide by the anti-harassment policy

Yes

Primary author: GOLASZEWSKI, Bartosz (BayLibre)
Presenter: GOLASZEWSKI, Bartosz (BayLibre)
Session Classification: IoTs a 4-Letter Word MC
Track Classification: LPC Microconference: IoTs a 4-Letter Word MC
Libre Silicon in IoT

Wednesday, 14 September 2022 13:00 (30 minutes)

Few have achieved what many would have thought impossible; bringing together a distributed community of engineers, then designing, prototyping, and fabricating a custom RISC-V SoC. The project was largely a success - in the first revision no less!

Designated PyFive, the intent was a libre silicon MCU capable of easily running Micropython and CircuitPython. It was designed and tested from the ground-up using open-source design & synthesis tools as well as an open-source PDK (Physical Design Kit). PyFive was one of 40 designs selected for MPW-1 in 2020, the first run of the Google-sponsored eFabless and Skywater foundry collaboration. There is now a GroupFund campaign which is truly the first of its kind.

Since then, the community has created a follow-up project called ICE-V Wireless which targets IoT. This board pairs an ESP32-C3 and an iCE40 FPGA (notably with OSS CAD suite support from YosysHq). The ESP32 BLE5 / WiFi module is fully capable of standing up on its own two legs without the FPGA. However, having fabric capable of hosting a soft-core CPU with peripherals next to the radio opens a world of possibilities for the average SoC designer on a budget.

This talk will go into detail on the successes and challenges encountered along the way, interfacing & tooling between Linux and a custom ASIC, and bringing up a custom Wireless device with Linux and Zephyr.

With platforms like PyFive and ICE-V, what future doors might be opened with libre silicon in the Linux IoT space?

I agree to abide by the anti-harassment policy

Yes

Primary author:  WELLING, Michael
Presenter:  WELLING, Michael
Session Classification:  IoTs a 4-Letter Word MC
Track Classification:  LPC Microconference: IoTs a 4-Letter Word MC
Linux IEEE 802.15.4 MLME improvements

Wednesday, 14 September 2022 12:05 (30 minutes)

As of today, Linux has relatively poor support for 802.15.4 MLME operations such as scanning and beaconning. These operations are at the base of beacon enabled PAN (Personal Area Networks) where devices can dynamically discover each other, associate to a PAN and maintain it as the devices move relatively to each other.

While some embedded RTOS like Zephyr already have a quite featureful support for these commands, Linux is a bit lagging behind. This talk will be an occasion to present the work done and still on-going to fill these gaps in the Linux kernel 802.15.4 stack.

I agree to abide by the anti-harassment policy
Yes

Primary author: RAYNAL, Miquèl
Presenter: RAYNAL, Miquèl
Session Classification: IoTs a 4-Letter Word MC
Track Classification: LPC Microconference: IoTs a 4-Letter Word MC
Open source FPGA NVMe accelerator platform for BPF driven ML processing with Linux/Zephyr

Wednesday, 14 September 2022 10:25 (30 minutes)

The talk will describe an open source NVMe development platform developed by Western Digital and Antmicro for server-based AI applications. The system combines an FPGA SoC with programmable logic and an AMP CPU, running Zephyr on the Corex-R cores handling NVMe transactions and Linux on Cortex-A in an openAMP setup.

The system utilizes Zephyr RTOS to perform time critical tasks including handling the base set of the NVMe commands, while all the custom commands are passed to the Linux system for further processing. The Linux system runs an uBPF virtual machine allowing users to upload and execute custom software processing the data stored on the NVMe drive.

The platform (custom hardware from Western Digital and open source software and FPGA firmware from Antmicro) was designed to enable users running ML pipelines designed in Tensorflow. To make it possible, the uBPF virtual machine has been extended to include functionalities to delegate certain processing to external, native, libraries interfacing the BPF code with hardware ML accelerators.

The platform includes an example showcasing a TensorFlow AI pipeline executed via the uBPF framework accelerating the AI model inference on an accelerator implemented in FPGA using TVM/VTA.

The platform intends to be a development platform for edge, near data processing acceleration research.

I agree to abide by the anti-harassment policy

Yes

Primary author:  GUGALA, Karol (Antmicro)
Presenter:  GUGALA, Karol (Antmicro)
Session Classification:  IOTs a 4-Letter Word MC
Track Classification:  LPC Microconference: IOTs a 4-Letter Word MC
Putting firmware on the device: a Zephyr+Yocto+Mender hike

Wednesday, 14 September 2022 10:00 (25 minutes)

One of the biggest real-life challenges for embedded developers is putting the various bits and pieces of technology together to form an actual product. Usually, each component offers good documentation and resources to get started, but documentation examples that encompass bigger, interconnected parts of a pipeline are often hard to come by.

In this presentation, we will start by building a firmware binary to be run on a coprocessor in an NXP iMX7-based AMP system. The resulting artifact will be included in a subsequent build process which generates a full Linux distribution. To facilitate this, a Yocto Project feature called “multiconfig” will be harnessed to orchestrate the successive tasks and integrate the results in a single artifact. This will constitute the actual, complete application binary that the device hardware will run.

Still a real product needs more than this...

It’s not enough to have the binary somewhere on the development machine - chances are that it also needs to be deployed as an update to devices in the field multiple times during the lifecycle of the product. At that point, Mender provides an OTA solution which can directly integrate into the Yocto Project based build process, helping to streamline the last step of distributing the generated binary image to a fleet of devices.

I agree to abide by the anti-harassment policy
Yes

Primary author:  Mr HOLZMAYR, Josef
Presenter:  Mr HOLZMAYR, Josef
Session Classification:  IoTs a 4-Letter Word MC
Track Classification:  LPC Microconference: IoTs a 4-Letter Word MC
Kernel Memory Management MC

Current problems of interest to kernel developers who focus on memory management:
- Multi-generational LRU vs traditional LRU
- Do we need three different slab allocators?
- How far do we take the folio conversion?
- Can we handle page pinning and page mapcount more effectively?
- How can we effectively cache reflinked files?
- Can we support 1GB pages other than through hugetlbfs?
- How should we handle memory failures better?

More problems will undoubtedly present themselves before the start of the conference.

I agree to abide by the anti-harassment policy

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

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

Track Classification:  LPC Microconference Track (CLOSED)
Copy On Write, Get User Pages, and Mysterious Counters

Monday, 12 September 2022 10:00 (30 minutes)

As we learned throughout the last decade (!), Copy On Write (COW) paired with Get User Pages (GUP) can be harder then it seems. Fortunately, it looks like that we might have both mechanisms working completely reliable in combination soon – at least for most types of anonymous memory.

In this talk, I’ll explain recent changes to our GUP and COW logic for anonymous memory, how they work, where we stand, what the tradeoffs are, what we’re missing, and where to go from here.

Also, I will talk about which mysterious counters are we using nowadays in our COW logic(s), what their semantics are, and what options we might have for simplifying one of them (hint: mapcount), and what the tradeoffs might be.

But also, what about the shared zeropage, private mappings of files, KSM ... ?

I agree to abide by the anti-harassment policy

Yes

Primary author: Mr HILDENBRAND, David (Red Hat)
Presenter: Mr HILDENBRAND, David (Red Hat)
Session Classification: Kernel Memory Management MC
Track Classification: LPC Microconference: Kernel Memory Management MC
Low-overhead memory allocation tracking

Tracking memory allocations for leak detection is an old problem with many existing solutions such as kmemleak and page_owner. However these solutions have relatively high performance overhead which limits their use. This talk will present memory allocation tracking implementation based on code tagging framework. It is designed to minimize performance overhead, while capturing enough information to discover kernel memory leaks.

I agree to abide by the anti-harassment policy
Yes

Primary authors: OVERSTREET, Kent; BAGHDASARYAN, SUREN
Presenters: OVERSTREET, Kent; BAGHDASARYAN, SUREN
Session Classification: Kernel Memory Management MC
Track Classification: LPC Microconference: Kernel Memory Management MC
CXL enables exploration of a more diverse range of memory technology beyond the DDR supported by the CPU. Those memory technologies come with different performance characteristics from a latency & bandwidth point of view. This means the memory topology of platforms becomes even more complex.

There is a large burden on how to leverage tiered memory, from letting the end user control placement to trying to automatically place memory on behalf of the end user. This presentation intends to be a review of the choices and technology that are in development (memory monitoring, NUMA, ...) and try to identify roadblocks.

I agree to abide by the anti-harassment policy

Yes

Primary author: Mr GLISSE, Jerome (Google)
Presenter: Mr GLISSE, Jerome (Google)
Session Classification: Kernel Memory Management MC
Track Classification: LPC Microconference: Kernel Memory Management MC
Multi-Gen LRU: Current Status & Next Steps

Monday, 12 September 2022 11:45 (30 minutes)

- Latest performance benchmark results on ARM64 servers and POWER9
- How to make MGLRU the default for everyone
- How to use MGLRU page table scanning to reclaim unused page-table pages
- A BPF program built on top of MGLRU to create per-process (access) heat maps

I agree to abide by the anti-harassment policy

Yes

Primary authors: BARNES, Jesse (Google); LEMARCHAND, Rom (Google); LUO, Hao (Google); ZHAO, Yu (Google); XIE, Yuanchu (Google)

Presenters: BARNES, Jesse (Google); LEMARCHAND, Rom (Google)

Session Classification: Kernel Memory Management MC

Track Classification: LPC Microconference: Compute Express Link MC
Preserving guest memory across kexec

Monday, 12 September 2022 12:15 (25 minutes)

Live update is a mechanism to support deploying updates to a running hypervisor in a way that has limited impact to virtual machines. This is done by pausing the virtual machines, stashing KVM state, kexecing into a new kernel, and restarting the VMM process. The challenge is guest memory: how can it be preserved and restored across kexec?

This talk describes a solution to this problem: moving guest memory out of the kernel managed domain, and providing control of memory mappings to userspace. Userspace is then able to restore the memory mappings of the processes and virtual machines via a FUSE-like interface for page table management.

We describe some requirements, options, why the FUSE-style options was chosen, an an overview of the work-in-progress implementation. Opinions are collected around other use cases this functionality could support. Next steps around finalising the design and working to get this included upstream are discussed.

This is a follow-on the the initial RFC presented at LSF-MM a few months ago: https://lwn.net/SubscriberLink/895453/71c46dbe09426f59/

I agree to abide by the anti-harassment policy

Yes

Primary author: GOWANS, James (Amazon EC2)
Co-author: WOODHOUSE, David
Presenter: GOWANS, James (Amazon EC2)
Session Classification: Kernel Memory Management MC

Track Classification: LPC Microconference: Kernel Memory Management MC
Scalability solutions for the mmap_lock - Maple Tree and per-VMA locks

Monday, 12 September 2022 10:30 (30 minutes)

The main issue of mmap_lock is its process-wide scale, which prevents handling page faults in one virtual memory area (VMA) of a process when another VMA of the same process is being modified.

The maple tree has been simplifying the way VMAs are stored to avoid multiple updates to the 3 data structures used to keep track of the VMAs.

Latest respin of Speculative Page Faults patchset (https://lwn.net/ml/linux-mm/20220128131006.67712-1-michel@lespinasse.org/) was deemed too complex to be accepted and the discussion concluded with a suggestion that “a reader/writer semaphore could be put into the VMA as a sort of range lock”. Per-VMA lock patchset implements this approach.

This talk will cover maple tree and per-VMA lock patchsets as well as the future of Speculative Page Faults patchset and new mmap_lock contention findings.

I agree to abide by the anti-harassment policy

Yes

Primary author: BAGHDASARYAN, SUREN

Presenters: HOWLETT, Liam (Oracle); BAGHDASARYAN, SUREN; LESPINASSE, Michel (Facebook)

Session Classification: Kernel Memory Management MC

Track Classification: LPC Microconference: Kernel Memory Management MC
The slab allocators of past, present, and future

Monday, 12 September 2022 13:05 (25 minutes)

- A summary of how we got to have SLAB, SLOB and SLUB.
- The strengths and weaknesses of each - performance, debugging, memory overhead.
- The issues with having three implementations.
- Code complexity and bitrot
- Other features having to implement for each variant or limit choice (kmemcg, PREEMPT_RT...)
- Imperfect common code, recent attempts to unify it more
- API improvement issues - we would like kfree() to work on kmem_cache_alloc() objects, but SLOB would have to adapt and increase memory overhead.
- Can we drop SLOB and/or SLAB? What changes would SLUB need in order to replace their use cases?

I agree to abide by the anti-harassment policy

Yes

Primary author: BABKA, Vlastimil (SUSE Labs)
Presenter: BABKA, Vlastimil (SUSE Labs)
Session Classification: Kernel Memory Management MC
Track Classification: LPC Microconference: Kernel Memory Management MC
Contribution ID: 262
Type: not specified

Kernel Summit

I agree to abide by the anti-harassment policy
Current Status and Future Plans of DAMON

Wednesday, 14 September 2022 12:45 (45 minutes)

DAMON is Linux kernel’s data access monitoring framework that provides best-effort accuracy under user-specified overhead range. It has been about one year after it has been merged in the mainline. Meanwhile, we received a number of new voices for DAMON from users and made efforts for answering to those. Nevertheless, many things to do for that are remaining.

This talk will share what such voices we received, what patches are developed or under development for those, what requests are still under plan only, and what the plans are. With that, hopefully we will have discussions that will be helpful for improving and prioritizing the plans and specific tasks, and finding new requirements.

Specific sub-topics will include, but not limited to:

- Making DAMON ABI more stable and flexible
- Extending DAMON for more usages
- Improving DAMON accuracy
- DAMON-based kernel/user space optimization policies
- Making user space DAMON policies more efficient
- Making kernel space DAMON policies just work (auto-tuning)

I agree to abide by the anti-harassment policy

Yes

Primary author: PARK, SeongJae
Presenter: PARK, SeongJae
Session Classification: Kernel Summit
Track Classification: Kernel Summit Track
Modernizing the kdump dump tools

Wednesday, 14 September 2022 10:45 (45 minutes)

kdump is a mechanism to create dump files after kernel panics for later analysis. It is particularly important for distributions as kdump often is the only way to debug problems reported by customers. Internally kdump the two user space tools `makedumpfile`, for dump creation, and `crash`, for dump analysis.

For both `makedumpfile` and `crash` to work they need to parse and interpret kernel internal, unstable data structures. This is problematic as both tools claim to be downward compatible. In the decades of their existence this lead to more and more history accumulating up to the point that it often takes hours of git archaeology to find out why the code is the way it is. This is not only time consuming but also leads to many bugs that could be prevented.

This talk shows how moving `makedumpfile` and `crash` to the tools/ directory in the kernel tree can help to simplify the code and thus reduce the maintenance needed for both tools. It also shows what consequences this move has for downstream partners and how these consequences can be minimized.

I agree to abide by the anti-harassment policy

Yes

Primary author: RUDO, Philipp
Presenter: RUDO, Philipp
Session Classification: Kernel Summit
Track Classification: Kernel Summit Track
Regression tracking & fixing: current state, problems, and next steps

Wednesday, 14 September 2022 10:00 (45 minutes)

This session will provide a very quick and brief overview about Thorsten’s recent regression tracking efforts, which are performed with the help of the regression tracking bot “regzbot”. Thorsten afterwards wants to outline and discuss a few oddities and problems in Linux development he noticed during his work that plague users – for example how bugzilla.kernel.org is badly handled and how some regressions are resolved only slowly, as the fixes take a long time to get mainlined.

In addition to that he will also outline some of the problems that make regression tracking hard for him in the hope a discussion will find ways to improve things. The session is also meant as a general forum to provide feedback to Thorsten about his work and discuss the further direction.

I agree to abide by the anti-harassment policy

Yes

Primary author: LEEMHUIS, Thorsten
Presenter: LEEMHUIS, Thorsten
Session Classification: Kernel Summit
Track Classification: Kernel Summit Track
The effort to add Rust support to the kernel is ongoing. There has been progress in different areas during the last year, and there are several topics that could benefit from discussion:

- Dividing the kernel crate into pieces, dependency management between internal crates, writing crates in the rest of the kernel tree, etc.
- Whether to allow dependencies on external crates and vending of useful third-party crates.
- Toolchain requirements in the future and status of Rust unstable features.
- The future of GCC builds: upcoming compilers, their status and ETAs, adding the kernel as a testing case for them...
- Steps needed for further integration in the different kernel CIs, running tests, etc.
- Documentation setup on kernel.org and integration between Sphinx/kernel-doc and rustdoc (this can be part of the documentation tech topic submitted earlier by Jon).
- Discussion with prospective maintainers that want to use Rust for their subsystem.

I agree to abide by the anti-harassment policy

Yes

Primary authors: OJEDA, Miguel; ALMEIDA FILHO, Wedson

Presenters: OJEDA, Miguel; ALMEIDA FILHO, Wedson

Session Classification: Kernel Summit

Track Classification: Kernel Summit Track
The Maple Tree

Wednesday, 14 September 2022 17:45 (45 minutes)

The maple tree is a kernel data structure designed to handle ranges. Originally developed to track VMAs but found new users before inclusion in mainline, the tree has many uses outside of the MM subsystem. I would like to talk about the current use cases that have arose and find out about any other uses that could be integrated into future plans.

I agree to abide by the anti-harassment policy

Yes

Primary author: HOWLETT, Liam (Oracle)
Presenter: HOWLETT, Liam (Oracle)
Session Classification: Kernel Summit
Track Classification: Kernel Summit Track
What kernel documentation could be

Wednesday, 14 September 2022 15:00 (45 minutes)

The development community has put a lot of work into the kernel’s documentation directory in recent years, with visible results. But the kernel’s documentation still falls far short of the standard set by many other projects, and there is a great deal of “tribal knowledge” in our community that is not set down. In this talk, the kernel documentation maintainer will look at the successes and failures of the work so far, but will focus on what our documentation should be and what we can do to get it there.

I agree to abide by the anti-harassment policy

Yes

Primary author: CORBET, Jonathan (Linux Plumbers Conference)
Presenter: CORBET, Jonathan (Linux Plumbers Conference)
Session Classification: Kernel Summit
Track Classification: Kernel Summit Track
Why is devm_kzalloc() harmful and what can we do about it

Wednesday, 14 September 2022 12:00 (45 minutes)

devm_kzalloc() has been introduced more than 15 years ago and has steadily grown in usage through the kernel sources (more than 6000 calls and counting). While it has helped lowering the number of memory leaks, it is not the magic tool that many seem to think it is.

The devres family of functions tie the lifetime of the resources they allocate to the lifetime of a struct device bind to a driver. This is the right thing to do for many resources, for instance MMIO or interrupts need to be released when the device is unbound from its driver at the latest, and the corresponding devm_* helpers ensure this. However, drivers that expose resources to userspace have, in many cases, to ensure that those resources can be safely accessed after the device is unbound from its driver. A particular example is character device nodes, which userspace can keep open and close after the device has been unbound from the driver. If the memory region that stores the struct cdev instance is allocated by devm_kzalloc(), it will be freed before the file release handler gets to run.

Most kernel developers are not aware of this issue that affects an ever growing number of drivers. The problem has been discussed in the past [1, 2] - interestingly in the context of Kernel Summit proposals, but never scheduled there - but never addressed.

This talk proposal aims at raising awareness of the problem, present a possible solution that has been proposed as an RFC [3], and discuss what we can do to solve the issue. Solutions at the technical, community and process levels will be discussed, as addressing the devm_kzalloc() hamr also requires a plan to teach the kernel community and catch new offending code when it gets submitted.

[1] https://lore.kernel.org/all/2111196.TG1k3f53YQ@avalon/
[2] https://lore.kernel.org/all/YOagA4bgdGYos5aa@kroah.com/
[3] https://lore.kernel.org/linux-media/20171116003349.19235-1-laurent.pinchart+renesas@ideasonboard.com/

I agree to abide by the anti-harassment policy

Yes

Primary author: PINCHART, Laurent (Ideas on Board Oy)
Presenter: PINCHART, Laurent (Ideas on Board Oy)
Session Classification: Kernel Summit
**Track Classification:**  Kernel Summit Track
The current trend in memory sizes lead me to believe that we’ll need 128-bit pointers by 2035. Hardware people are starting to think about it [1a] [1b]. We have a cultural problem in Linux where we believe that all pointers (user or kernel) can be stuffed into an unsigned long and newer C solutions (uintptr_t) are derided as “userspace namespace mess”.

The only sane way to set up a C environment for a CPU with 128-bit pointers is sizeof(void *) == 16, sizeof(short) == 2, sizeof(int) == 4, sizeof(long) == 8, sizeof(long long) == 16.

That means that casting a pointer to a long will drop the upper 64 bits, and we’ll have to use long long for the uintptr_t on 128-bit. Fixing Linux to be 128-bit clean is going to be a big job, and I’m not proposing to do it myself. But we can at least start by not questioning when people use uintptr_t inside the kernel to represent an address.

Getting the userspace API fixed is going to be the most important thing (eg io_uring was just added and is definitely not 128-bit clean). Fortunately, no 128-bit machines exist today, so we have a bit of time to get the UAPI right. But if not today, then we should start soon.

There are two purposes for this session:

- Agree that we do need to start thinking about 128-bit architectures (even if they’re not going to show up in our offices tomorrow)
- Come to terms with needing to use uintptr_t instead of unsigned long

[1b] https://github.com/riscv/riscv-opcodes/blob/master/unratified/rv128_i

I agree to abide by the anti-harassment policy

Yes

**Primary author:** WILCOX, Matthew (Oracle)

**Presenter:** WILCOX, Matthew (Oracle)

**Session Classification:** Kernel Summit

**Track Classification:** Kernel Summit Track
Kernel Testing & Dependability MC

The Linux Plumbers 2022 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 a merge of the Linux Plumbers 2021 Testing and Fuzzing and the Kernel Dependability and Assurance MC tracks into a single session. These two tracks have a lot of overlap in topics and attendees. A dependable kernel is the goal of testing it and combining the two tracks will promote collaboration between all the interested communities and people.

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:

- KCIDB is continuing to gather results from many test systems: KernelCI, Red Hat’s CKI, syzbot, ARM, Gentoo, Linaro’s TuxSuite etc. The current focus is on generating common email reports based on this data and dealing with known issues.
- KFENCE was successfully merged. (https://lwn.net/Articles/835367/)
- Clang: CFI, weeding out issues upstream, etc.
- KUnit started acting as the standard for some drivers. (https://www.youtube.com/watch?v=78gioY7VYxc)
The Runtime Verification (RV) interface patch series (https://lore.kernel.org/linux-doc/cover.1621414942.git.bristot@redhat.com) from Daniel Bristot de Oliveira is in review.

ELISA Tools WG (https://lists.elisa.tech/g/tool-investigation) as incorporated syzkaller in their testing workflow (https://elisa-builder-00.iol.unh.edu/syzkaller/reportid=34e8ccdc0c7fb916735e544ea99234de75).

MC Leads:

- Sasha Levin
- Guillaume Tucker
- Shuah Khan

Proposed Participants:

- 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
- Kate Stewart
• Sudip Mukherjee
• Daniel Bristot de Oliveira
• Gabriele Paoloni
• Paul Albertella
• Elana Copperman
• Lukas Bulwahn

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

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

**Track Classification:** LPC Microconference Track (CLOSED)
Checking your work: Linux kernel testing and CI

There are a number of tools available for writing tests in the kernel. One of them is kselftest, which is a system for writing end-to-end tests. Another is KUnit, which runs unit tests directly in the kernel.

These testing tools are very useful, but they lack the ability for maintainers to configure how the tests should be run. For example, patches submitted to the RCU tree and branch should run a quick subset of the full gamut of rcutorture tests, whereas it is prudent to run heavyweight and comprehensive rcutorture tests once a day on the linux-rcu tree, as well as various mainline trees, etc. Similarly, cgroup tests can be run on every patch sent to the cgroup tree, but certain testcases have known flakiness that could be signaled by the developer.

Maintainers and contributors would benefit from being able to configure their test suites to illustrate the intent of individual testcases, and the suite at large, to signal both to contributors and to CI systems, how the tests should be run and interpreted. This MC discussion would ask the question of whether we should implement this, and if so, what it would look like.

Key problems:
- Updating kernel test subsystem structure (kselftest, kunit) to allow maintainers to express configurations for test suites. The idea here is to avoid developers having to include patches to each separate CI system to include and configure their test, and instead have all of that located in configuration files that are included in the test suite, with CI systems consuming and using this information as necessary.
- Discuss whether we should bring xfstests into the kernel source tree, and whether we could make it a kselftest.
- Discuss whether we can and should include coverage information in KernelCI.

Key people:
- Guillaume Tucker and other KernelCI maintainers
- Ideally Shuah Khan as kselftest maintainer, and Brendan Higgins as KUnit maintainer
- Anyone interested in testing / CI signal (Thorsten Leemhuis perhaps, given his KR talk about how to submit an actionable kernel bug report?)

I agree to abide by the anti-harassment policy

Yes

Primary author:  VERNET, David (Meta)
Presenter:  VERNET, David (Meta)
Session Classification:  Kernel Testing & Dependability MC
Track Classification:  LPC Microconference: Kernel Testing & Dependability MC
Fuzzing (randomized testing) become an important part of the kernel quality assurance. syzkaller/syzbot report a hundred of bugs each month. However, the fuzzer coverage of the kernel code is far from being complete and some subsystems are easier to fuzz/reach, while others are harder/impossible to fuzz/reach.

In this talk Dmitry will talk about patterns and anti-patterns of UAPI/subsystem design with respect to fuzz-ability:

- what makes it impossible to fuzz a subsystem
- what leads to unreproducible crashes
- why a subsystem may be excluded from fuzzing
- what makes a perfect interface/subsystem for fuzzing

I agree to abide by the anti-harassment policy

Yes

**Primary author**: VYUKOV, Dmitry (Google)

**Presenter**: VYUKOV, Dmitry (Google)

**Session Classification**: Kernel Testing & Dependability MC

**Track Classification**: LPC Microconference: Kernel Testing & Dependability MC
How to introduce KUnit to physical device drivers?

Monday, 12 September 2022 17:40 (20 minutes)

Unit testing is a great way to ensure code reliability, leading to organic improvements, as it’s often possible to integrate it with developers’ workflows. It is also of great help when refactoring, which should be a primordial task in large code bases. When it comes to the Linux kernel, the KUnit framework looks very promising, as it works natively from inside the kernel, and provides an infrastructure for running tests easily.

We are seeing a growing interest in unit testing on the DRM subsystem, with amazing initiatives to add KUnit tests to the DRM API. Moreover, three GSoC projects under the X.Org Foundation umbrella target unit tests for AMDGPU display drivers, as it is currently the largest one in the kernel. It is, thus, of great importance to discuss problems and possible solutions regarding the implementation of KUnit tests, especially for hardware drivers.

Bearing this in mind, and as part of our GSoC projects, we introduce unit testing to the AMDGPU driver departing from the Display Mode Library (DML), which is a library focused on mathematical calculations for DCN (Display Core Next); we also explore the addition of new tests to DCE (Display Controller Engine). Since AMD’s CI already relies on IGT GPU Tools (a test suite for DRM drivers) we also propose an integration between it and KUnit which allows for DRM KUnit tests to be run through IGT as well.

In this talk, we present the tests’ development process and the current state of KUnit in GPU drivers. We discuss the obstacles we faced during the project, such as generating coverage reports, mocking a physical device, and especially in regards to the implementation of tests for the AMDGPU driver stack, with the additional difficulties associated with making them IGT compatible. Finally, we want to discuss with the community lessons learned using KUnit in GPU drivers and how to reuse these strategies for other GPU drivers and also drivers in other subsystems.

https://summerofcode.withgoogle.com/programs/2022/organizations/xorg-foundation

I agree to abide by the anti-harassment policy

Yes

Primary authors:  BASSO, Isabella; LEMES, Magali; CANAL, Maira; DA APARECIDA, Tales

Presenters:  BASSO, Isabella; LEMES, Magali; CANAL, Maira; DA APARECIDA, Tales

Session Classification:  Kernel Testing & Dependability MC

Track Classification:  LPC Microconference: Kernel Testing & Dependability MC
Integrating testing with maintainer flows

Monday, 12 September 2022 15:00 (30 minutes)

Currently we often have a fairly big disconnect between generic testing and quality efforts and the work of developers and maintainers. There is a lot of testing that is done by people working on the kernel that is not covered by what the general testing efforts do, and conversely it is often hard for developers and maintainers to access broader community resources when extra testing is needed. This impacts both development but also downstream users like stable kernels.

How can we join these efforts together?

Areas with successful reuse include:

- ksselftest
- igt
- v4l-compliance

Other testing efforts more confined to their domains:

- Intel’s audio testing setup
- Filesystem coverage (xfstests and so on)

Ideas/topics for discussion:

- Improved tooling
- Testsuite/system configuration
- Proposals from Cyril for describing hardware for tests
- Test interface standards
- ...

I agree to abide by the anti-harassment policy

Yes

Primary author:  BROWN, Mark
Presenters:  BROWN, Mark;  KABATOVA, Veronika
Session Classification:  Kernel Testing & Dependability MC
Track Classification:  LPC Microconference: Kernel Testing & Dependability MC
KUnit: Function Redirection and More

Monday, 12 September 2022 17:20 (20 minutes)

Despite everyone’s efforts, there’s still more kernel to test. One problem area that keeps popping up is the need to replace functions with ‘fake’ or ‘mock’ equivalents in order to test hardware or less-self-contained subsystems. We will discuss two methods of replacing functions: one based on ftrace, and another based on “static stubbing” using a function prologue.

We will also provide a brief “KUnit year in review” retrospective, and a prospective look on what we are doing/what we hope to achieve in the coming year.

I agree to abide by the anti-harassment policy

Yes

Primary authors: HIGGINS, Brendan (Google LLC); GOW, David (Google)

Presenters: HIGGINS, Brendan (Google LLC); GOW, David (Google)

Session Classification: Kernel Testing & Dependability MC

Track Classification: LPC Microconference: Kernel Testing & Dependability MC
Making syzbot reports more developer-friendly

Since 2017, syzbot (powered by syzkaller - a coverage-guided kernel fuzzer) has already reported thousands of bugs to the Linux kernel mailing lists and thousands have already been fixed.

However, as our statistics show, a lot of reported issues get fixed only after a long delay or don’t get fixed at all. That means we could still do much better in addressing the needs of the community than we currently do.

This talk will summarize and present the changes that have been made to syzbot over the last year. Also, we want to share and discuss with the audience our further plans related to making our reports and our tool more developer-friendly.

I agree to abide by the anti-harassment policy

Yes

Primary author: NOGIKH, Aleksandr (Google)
Co-author: VYUKOV, Dmitry (Google)
Presenter: NOGIKH, Aleksandr (Google)
Session Classification: Kernel Testing & Dependability MC
Track Classification: LPC Microconference: Kernel Testing & Dependability MC
Simple KernelCI Labs with Labgrid

Monday, 12 September 2022 18:00 (20 minutes)

While most current KernelCI labs use Lava to deploy and test the Kernels on real hardware, other approaches are supported by KernelCI’s design.

To allow using boards connected to an existing labgrid installation, Jan build a small adapter from KernelCI’s API to labgrid’s hardware control Python API.

As labgrid has a way to support board-specific deployment steps, this should also make it easier to run tests on boards which are not easily supported in Lava (such as without Ethernet) or requiring special button settings.

The main goal of the discussion is to collect feedback from the MC participants, on how to make this adapter most useful for the KernelCI community.

I agree to abide by the anti-harassment policy

Yes

Primary author:  LÜBBE, Jan (Pengutronix)
Presenter:  LÜBBE, Jan (Pengutronix)
Session Classification:  Kernel Testing & Dependability MC
Track Classification:  LPC Microconference: Kernel Testing & Dependability MC
The emerging of the virtual QA team for Linux kernel

Monday, 12 September 2022 17:00 (20 minutes)

Linux kernel community has formed a virtual QA team and testing process gradually, from developing unit testing, to testing service (various CI that covers build, fuzzing and runtime), to result consolidation (KCIDB) to bug scrub (regzbot), which largely formalize the testing effort in community wide. 0-Day CI is glad to be part of this progress.

In this topic, we will talk about the status of such trend, and how each part work together. And a few words regarding 0-Day CI's current effort in this trend. Then we want to exchange the ideas and have the discussion around any enhancements or missing parts of this virtual team, like

- Common testing methodology, such as bisection optimization, selective testing to reduce overall tests based on what is really changed.
- Test matrix to map feature and tests so to know whether a feature is covered and enough
- The connection with OSV, user and feature developer to convert their tests to be in the test pool like kselftests
- Reduce breakage between different architectures to avoid test bias
- Roadmap to adopt new test tools such as kismet
- Extend the shift left testing to detect issue earlier so likely to reduce the overall Reported-by tags on mainline

We look forward to having more collaboration with other players in the community to jointly move this trend forward.

I agree to abide by the anti-harassment policy

Yes

Primary author: LI, philip
Presenter: LI, philip
Session Classification: Kernel Testing & Dependability MC
Track Classification: LPC Microconference: Kernel Testing & Dependability MC
Historically, the code in arch/ was developed for one architecture and then copied and adjusted by others. This created a lot of duplicated or almost duplicated code with subtle differences which prevents easy refactoring and consolidation.

The linux/arch microconference aims to bring architecture maintainers in one room to discuss how the code in arch/ can be improved, consolidated and generalized, at least where it makes sense.

The discussion at the previous linux/arch microconfernece in 2020 lead to updates in RISC-V kprobes implementation\(^1\), removal of DISCOTIGMEM memory model\(^2\) and enablement of generic entry on s390\(^3\).

\(^1\) [https://git.kernel.org/torvalds/c/c22b0bceb1dd0](https://git.kernel.org/torvalds/c/c22b0bceb1dd0)
\(^2\) [https://git.kernel.org/torvalds/c/bb1c50d39f67f](https://git.kernel.org/torvalds/c/bb1c50d39f67f)
\(^3\) [https://git.kernel.org/torvalds/c/56e62a737028](https://git.kernel.org/torvalds/c/56e62a737028)

The possible topics could be:

- reducing code duplication and generalizing the common code in arch/
- making headers in include/asm consistent
- on-boarding more architectures to use common entry code
- devicetree (unless they have their own microconf)
- identifying old machine support that may be either still in active use vs only in hobbyist/retro-computing vs completely obsolete and broken

MC Leads:
Mike Rapoport <rppt@kernel.org>, Arnd Bergmann <arnd@arndb.de>

I agree to abide by the anti-harassment policy

**Primary authors:** BERGMANN, Arnd (Linaro); RAPOPORT, Mike (IBM)

**Presenters:** BERGMANN, Arnd (Linaro); RAPOPORT, Mike (IBM)

**Track Classification:** LPC Microconference Track (CLOSED)
Consolidating representations of the physical memory

Tuesday, 13 September 2022 11:00 (30 minutes)

We have several coarse representations of the physical memory consisting of [start, end, flags] structures per memory region. There is memblock that some architectures keep after boot, there is iomem_resource tree and “System RAM” nodes in that tree, there are memory blocks exposed in sysfs and then there are per-architecture structures, sometimes even several per architecture.

The multiplication of such structures and lack of consistency between some of them does not help the maintainability and can be a reason for subtle bugs here and there.

The layout of the physical memory is defined by hardware and firmware and there is not much room for its interpretation; single abstraction of the physical memory should suffice and a single [start, end, flags] type should be enough. There is no fundamental reason why we cannot converge per-architecture representations of the physical memory, like e820, drmem_lmb, memblock or numa_meminfo into a generic abstraction.

I suggest to use memblock as the basis for such abstraction. It is already supported on all architectures and it is used as the generic representation of the physical memory at boot time. Closing the gaps between per architecture structures and memblock is anyway required for more robust initialization of the memory management. Addition of simple locking of memblock data for memory hotplug, making the memblock “allocator” part discardable and a mechanism to synchronize “System RAM” resources with memblock would complete the picture.

I agree to abide by the anti-harassment policy

Yes

Primary author: RAPOPORT, Mike (IBM)
Presenter: RAPOPORT, Mike (IBM)
Session Classification: linux/arch MC
Track Classification: LPC Microconference: linux/arch MC
Extending EFI support in Linux to new architectures

Tuesday, 13 September 2022 12:30 (30 minutes)

An overview will be presented of recent work in the Linux/EFI subsystem and associated projects (u-boot, Tianocore, systemd), with a focus on generic support for the various new architectures that have adopted EFI as a supported boot flow. This includes UEFI secure boot and/or measured boot on non-Tianocore based EFI implementations, generic decompressor support in Linux and early handling of RNG seeds provided by the bootloader.

Note that topics related to confidential computing (TDX, SEV) will not be covered here: there are numerous other venues at LPC and the KVM Forum that already cover this in more detail.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  BIESHEUVEL, Ard (Google); APALODIMAS, Ilias
Presenters:  BIESHEUVEL, Ard (Google); APALODIMAS, Ilias
Session Classification:  linux/arch MC
Track Classification:  LPC Microconference: linux/arch MC
High memory management API changes

Tuesday, 13 September 2022 10:00 (30 minutes)

There was a time when the Linux kernel was 32bit but hardware systems had much more than 1GB of memory. A solution was developed to allow the use of high memory (HIGHMEM). High memory was excluded from the kernel direct map and was temporarily mapped into and out of the kernel as needed. These mappings were made via kmap_*() calls.

With the prevalence of 64bit kernels the usefulness of this interface is waning. But the idea of memory not being in the direct map (or having permissions beyond the direct map mapping) has brought about the need to rethink the HIGHMEM interface.

This talk will discuss the changes to the kmap_*() API and the motivations driving them. This includes the status of a project to rework the HIGHMEM interface as of the LPC conference.

Finally how does HIGHMEM affect the modern architectures in use? Is it finally time to remove CONFIG_HIGHMEM? Or is there still a need for 32 bit systems to support large amounts of memory in production environments?

I agree to abide by the anti-harassment policy

Yes

Primary author: WEINY, Ira
Presenter: WEINY, Ira
Session Classification: linux/arch MC
Track Classification: LPC Microconference: linux/arch MC
LoongArch: What we will do next

Tuesday, 13 September 2022 12:00 (30 minutes)

- cleaner way forward for compatibility with the "old-world" (the earlier in-house MIPS-clone firmware and kernel ABI of LoongArch), if possible;
- cleaner approach to support both standard UEFI and the Loongson-custom boot protocols, if possible;
- way forward for supporting zboot in EFI stub boot flow.

I agree to abide by the anti-harassment policy

Yes

Primary authors: CHEN, Huacai; LV, Jianmin (Loongson); WANG, Xuerui
Presenters: CHEN, Huacai; LV, Jianmin (Loongson); WANG, Xuerui
Session Classification: linux/arch MC
Track Classification: LPC Microconference: linux/arch MC
Make LL/SC arch has a strict forward guarantee

For the architecture that uses load-link/store-conditional to implement atomic semantics, ll/sc can effectively reduce the complexity and cost of embedded processors and is very attractive for products with up to two cores in a single cluster. However, compared with the AMO architecture, it may not have enough forward guarantee, causing the risk of livelock. Therefore, CPUs based on the ll/sc architecture such as csky, openrisc, riscv, and loongarch haven’t met the requirements of using qspinlock in NUMA scenarios. In this presentation, we will introduce how to make ll/sc have strict forward guarantees, solve the mixed-size atomic & dcas problem incidentally, and discuss the hardware solution’s advantages and disadvantages. I hope this presentation will help ll/sc architecture solve NUMA series issues in Linux.

I agree to abide by the anti-harassment policy

Yes

Primary author: Mr GUO, Ren
Presenter: Mr GUO, Ren
Session Classification: linux/arch MC
Track Classification: LPC Microconference: linux/arch MC
Mitigating speculative execution attacks with ASI - follow up

Tuesday, 13 September 2022 10:30 (30 minutes)

In this talk we will argue the case for adopting ASI in upstream Linux.

Speculative execution attacks, such as L1TF, MDS, LVI, (and many others) pose significant security risks to hypervisors and VMs, from neighboring malicious VMs. The sheer number of proposed patches/fixes is quite high, each with its own non-trivial impact on performance. A complete mitigation for these attacks requires very frequent flushing of several buffers (L1D cache, load/store buffers, branch predictors, etc. etc.) and halting of sibling cores. The performance cost of deploying these mitigations is unacceptable in realistic scenarios.

Two years ago, we presented Address Space Isolation (ASI) - a high-performance security-enhancing mechanism to defeat these speculative attacks. We published a working proof of concept in https://lkml.org/lkml/2022/2/23/14. ASI, in essence, is an alternative way to manage virtual memory for hypervisors, providing very strong security guarantees at a minimal performance cost.

In the talk, we will discuss what new vulnerabilities have been discovered since our previous presentation, what are the existing approaches, and their estimated costs. We will then present our performance estimation of ASI, and argue that ASI can mitigate most of the speculative attacks as is, or by a small modification to ASI, at an acceptable cost.

I agree to abide by the anti-harassment policy

Yes

Primary authors: SHAHID, Junaid (Google); WEISSE, Ofir (Google)

Presenters: SHAHID, Junaid (Google); WEISSE, Ofir (Google)

Session Classification: linux/arch MC

Track Classification: LPC Microconference: linux/arch MC
LPC Refereed Track

I agree to abide by the anti-harassment policy
Exercising the Linux scheduler with Yogini

Monday, 12 September 2022 12:00 (45 minutes)

Introducing “yogini”, a flexible Linux tool for stretching the Linux scheduler and measuring the result.

Yogini includes an extensible catalogue of simple workloads, including execution, cache and memory bound, as well as advanced (Intel) ISAs. The workloads are assigned to threads, which can be run at prescribed rates at prescribed times.

At the same time, yogini can run a periodic system monitor, which tracks frequency, power, sched stats, temperature and other hardware and software metrics. Since yogini tracks both power and performance, it can combine them to report energy efficiency.

Measurement results are buffered in memory and dumped to a .TSV file upon completion – to be read as text, imported to your favorite spreadsheet, or plotted via script.

As the workloads are well controlled, yogini lends itself well to be used for creating Linux regression tests – particularly those relating to scheduler-related performance and efficiency.

Yogini is new. The goal of this session it let the community know it is available, and hopefully useful, and to solicit ideas for making it even more useful for improving Linux.

I agree to abide by the anti-harassment policy

Yes

Primary author: BROWN, Len (Intel Open Source Technology Center)
Presenter: BROWN, Len (Intel Open Source Technology Center)
Session Classification: LPC Refereed Track
Track Classification: LPC Refereed Track
How I started chasing speculative type confusion bugs in the kernel and ended up with ’real’ ones

Tuesday, 13 September 2022 10:00 (45 minutes)

This talk will illustrate my journey in kernel development as a PhD student in Computer Systems Security. I’ve started with Kasper, a tool I have co-designed and implemented, that finds speculative vulnerabilities in the Linux kernel. With the help of compilers Kasper emulates speculative execution to apply sanitizers on the speculative path. Building a generic vulnerability scanner allows finding gadgets that were previously undiscovered by pattern matching with static analysis. Spectre is not limited to a bounds check bypass! Kasper tries to find speculative gadgets and present them in a web UI for developers to analyse. I will also discuss ongoing efforts to improve the precision of the analysis and reason over practical exploitability.

After we found a speculative type confusion within the list iterator macros, I posted a patch set with a suggested mitigation strategy. By looking at different uses of the list iterator variable after the loop, I entered territory of actual type confusions. I will also discuss ongoing efforts in building an automatic tool for the Linux kernel to detect invalid downcasts with container_of since they otherwise stay completely undetected. We would also gladly like to open a discussion with the audience on the interest and welcome feedback from the community.

Primary author:  KOSCHEL, Jakob (VUSec Amsterdam)
Presenter:  KOSCHEL, Jakob (VUSec Amsterdam)
Session Classification:  LPC Refereed Track
Track Classification:  LPC Refereed Track
Inside the Linux Kernel Random Number Generator

Over the last year, the kernel’s random number generator has seen significant changes and modernization. Long a contentious topic, filled with all sorts of opinions on how to do things right, the RNG is now converging on a particular threat model, and makes use of cryptographic constructions to meet that threat model. This talk will be an in depth look at the various algorithms and techniques used inside of random.c, its history and evolution over time, and ongoing challenges. It will touch on issues such as entropy collection, entropy estimation, boot time blocking, hardware cycle counters, interrupt handlers, hash functions, stream ciphers, cryptographic sponges, LFSRs, RDRAND and related instructions, bootloader-provided randomness, embedded hardware, virtual machine snapshots, seed files, academic concerns versus practical concerns, performance, and more. We’ll also take a look at the interfaces the kernel exposes and how these interact with various userspace concerns. The intent is to provide an update on everything you’ve always wondered about how the RNG works, how it should work, and the significant challenges we still face. While this talk will address cryptographic issues in depth, no cryptography background is needed. Rather, we’ll be approaching this from a kernel design perspective and soliciting kernel-based solutions to remaining problems.

I agree to abide by the anti-harassment policy

Yes

Primary author: DONENFELD, Jason
Presenter: DONENFELD, Jason
Session Classification: LPC Refereed Track
Track Classification: LPC Refereed Track
Kernel Live Patching at Scale

Tuesday, 13 September 2022 15:00 (45 minutes)

Kernel live patching (KLP) makes it possible to apply quick fixes to a live Linux kernel, without having to shut down the workload to reboot a server. The kpatch tool chain and the livepatch infrastructure generally work well. However, using them on a closely monitored fleet with several million servers uncovers many corner cases. During the deployment of KLP at Meta, we ran into issues, including performance regressions, conflicts with tracing & monitoring tools, and KLP transitions sporadically failing depending on the behavior of the kernel at the time the patch is applied. In this presentation, we will share our experiences working with KLP at scale, describe the top issues we are facing, and discuss some ideas for future improvements.

First, we would like to briefly introduce how we build, deploy, and monitor KLPs at scale. We will then present some recent work to improve KLP infrastructure, including: eliminating performance hit when applying KLPs; making sure KLP works well with various tracing mechanisms; and fixing various corner cases with kpatch-build tool chain and livepatch infrastructure. Finally, we would like to discuss the remaining issues with KLP at scale, and how to address them. Specifically, we will present different reasons for KLP transition errors, and a few ideas/WIPs to address these errors.

I agree to abide by the anti-harassment policy

Yes

Primary authors: LIU, Song (Meta); VAN RIEL, Rik (Meta); VERNET, David (Meta)

Presenters: LIU, Song (Meta); VAN RIEL, Rik (Meta); VERNET, David (Meta)

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track
Launching new processes with 'io_uring_spawn' for fast builds

Monday, 12 September 2022 10:45 (45 minutes)

io_uring allows running a batch of operations fast, on behalf of the current process. As the name suggests, this works exceptionally well for I/O workloads. However, one of the most prominent workloads in software development involves executing other processes: make and other build systems launch many other processes over the course of a build. How can we launch those processes faster?

What if we could launch other processes, and give them initial work to do using io_uring, ending with an exec? What if we could handle the pre-exec steps for a new process entirely in the kernel, with no userspace required, eliminating the need for fork or even vfork, and eliminating page-table CoW overhead?

In this talk, I’ll introduce io_uring_spawn, a mechanism for launching empty new processes with an associated io_uring. I’ll show how the kernel can launch a blank process, with no initial copy-on-write page tables, and initialize all of its resources from an io_uring. I’ll walk through both the successful path and the error-handling path, and show how to get information about the launched process. Finally, I’ll show how existing userspace can take advantage of io_uring_spawn to speed up posix_spawn, and provide performance numbers for common workloads, including kernel compilation.

I agree to abide by the anti-harassment policy

Yes

Primary author:  TRIPLETT, Josh
Presenter:  TRIPLETT, Josh
Session Classification:  LPC Refereed Track
Track Classification:  LPC Refereed Track
Live in a world with multiple memory types

Tuesday, 13 September 2022 12:45 (45 minutes)

Initially, all memory are DRAM, then we have graphics memory, PMEM, CXL, ... Linux kernel has recently gained the basic support to manage systems with multiple memory types and memory tiers, and the ability to optimize performance by demoting/promoting memory between the tiers. And we are working on enhancing Linux’s capabilities further.

In this talk, we will discuss the current development and future direction to manage and optimize these systems, including,

- Explicit memory tiers and user space interface
- Support complex memory topology with help of firmware and device drivers
- Use NUMA memory policy and cpusets to help manage memory types
- Possible improvement of demoting with MGLRU
- Further optimize page promoting with hot page selection and alternatives
- Control the trashing among memory types
- Possible user space based demoting/promoting

We also want to discuss about the possible solution choices and interfaces in kernel and user space.

I agree to abide by the anti-harassment policy

Yes

Primary author:  Mr HUANG, Ying
Presenter:  Mr HUANG, Ying
Session Classification:  LPC Refereed Track
Track Classification:  LPC Refereed Track
Make RCU do less (save power)!

Monday, 12 September 2022 17:00 (45 minutes)

On battery-powered systems, RCU can be a major consumer of power. Different strategies can be tried to mitigate power which we will show along with power data. Also I have been working on some patches to further reduce RCU activity in frequently-called paths like file close. This presentation is to discuss some test results mostly on the power consumption side on battery-powered Android and ChromeOS systems and the new work mentioned to delay RCU processing.

I agree to abide by the anti-harassment policy

Yes

Primary authors: FERNANDES, Joel; KADAM, Rushikesh; REZKI, Uladzislau

Presenters: FERNANDES, Joel; KADAM, Rushikesh; REZKI, Uladzislau

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track
Meta’s CXL Journey and Learnings in Linux

Tuesday, 13 September 2022 17:00 (45 minutes)

Compute Express Link (CXL) is a new open interconnect technology built on top of PCIe. Among other features, it enables memory expansion, unified system address space and cache coherency. It has the potential to enable SDM (Software Defined Memory) and emerging usage models of accelerators.

Meta has been working on CXL with current focus on memory expansion. This presentation will discuss Meta’s experiences, learnings, pain points and expectations for Linux kernel/OS to support CXL’s value proposition and at-scale data center deployment. It touches upon aspects such as transparent memory expansion, device management at scale, RAS, etc. Meta looks forward to further collaboration with the Linux community to improve CXL technology and to enable the CXL ecosystem.

Primary author: ZHANG, Jonathan
Presenter: ZHANG, Jonathan
Session Classification: LPC Refereed Track
Track Classification: LPC Refereed Track
Modularization for Lockdep

Monday, 12 September 2022 15:45 (45 minutes)

Lockdep is a powerful tool for developers to uncover lock issues. However there are things that still need to improve:

- The error messages are sometimes confusing and difficult to understand, and require experts to decode them. This not only makes read deadlock scenarios challenging to understand, but also makes internal bugs hard to debug.

- Once one lock issue is reported, all the lockdep functionalities are turned off. Although this is reasonable, because once a lock issue is detected the whole system is subject to lock bugs and it’s pointless to continue running the system until the bugs are fixed. However this is frustrating for developers when they hit some lock issues that happen in other subsystems, they cannot test their code for lock issues until the existing ones are fixed.

- Detection takes time to run and creates extra synchronization points than production environments. It’s not surprising that lockdep uses an internal lock to protect the data structures for lock issue detections. However, this lock creates synchronization points and may make some issues difficult to detect (because the issues may only happen for a particular even sequence, and the extra synchronization points may prevent such a sequence from happening).

This session will show some modularization effort for lockdep. The modularization use a frontend-backend design: the frontend tracks the current held locks for every task/contexts and reports lock dependencies to the backend, and the backend maintains the lock dependency graph and detect lock issues based on what the frontend reports.

Along with the design, a draft implementation will be shown in the session too, providing something concrete to discuss about the design and the future work.

I agree to abide by the anti-harassment policy

Yes

Primary author: FENG, Boqun
Presenter: FENG, Boqun
Session Classification: LPC Refereed Track
Track Classification: LPC Refereed Track
nouveau in the times of nvidia firmware and open source kernel module

Tuesday, 13 September 2022 17:45 (45 minutes)

This talk will look at the recent NVIDIA firmware release and open source kernel module contents, define what exists, what can happen.

It will then address the nouveau project and what this means to it, and what sort of plans are in place to use what NVIDIA has provided to move the project forward.

It will also discuss possible future projects in the area.

I agree to abide by the anti-harassment policy

Yes

Primary author: ARLIE, David
Presenter: ARLIE, David
Session Classification: LPC Refereed Track
Track Classification: LPC Refereed Track
Once upon an API

Monday, 12 September 2022 17:45 (45 minutes)

Even a seemingly simple API can turn out to have complex and surprising behaviors, as I illustrate by telling the story of an API feature that was added to Linux in 1997 and looking how it interacts (and has evolved) with other parts of the Linux API. These kinds of complexities and surprises of course create pain for user-space programmers, and so I also muse about some of the reasons that these API design problems occur and a few things we can do to reduce the likelihood of such problems reoccurring in the future.

I agree to abide by the anti-harassment policy

Yes

Primary author:  KERRISK, Michael (man7.org Training and Consulting)

Presenter:  KERRISK, Michael (man7.org Training and Consulting)

Session Classification:  LPC Refereed Track

Track Classification:  LPC Refereed Track
OS Scheduling with Nest: Keeping Tasks Close Together on Warm Cores

Monday, 12 September 2022 12:45 (45 minutes)

To best support highly parallel applications, Linux’s CFS scheduler tends to spread tasks across the machine on task creation and wakeup. It has been observed, however, that in a server environment, such a strategy leads to tasks being unnecessarily placed on long-idle cores that are running at lower frequencies, reducing performance, and to tasks being unnecessarily distributed across sockets, consuming more energy. In this talk, we propose to exploit the principle of core reuse, by constructing a nest of cores to be used in priority for task scheduling, thus obtaining higher frequencies and using fewer sockets. We implement the Nest scheduler in the Linux kernel. While performance and energy usage are comparable to CFS for highly parallel applications, for a range of applications using fewer tasks than cores, Nest improves performance 10%-2x and can reduce energy usage.

I agree to abide by the anti-harassment policy

Yes

Primary author: LAWALL, Julia (Inria)
Co-authors: CHHAYA-SHAILESH, Himadri (Inria); LOZI, Jean-Pierre (Oracle Labs); LEPERS, Baptiste (University of Sydney); ZWAENEPOL, Willy (University of Sydney); MULLER, Gilles (Inria)
Presenter: LAWALL, Julia (Inria)
Session Classification: LPC Refereed Track
Track Classification: LPC Refereed Track
The PREEMPT_RT patch set has only a handful patches left until it can be enabled on the X86 Architecture at the time of writing. The work has not finished once the patches are fully merged. A new issue is how to not break parts of PREEMPT_RT in future development by making assumption which are not compatible or lead to large latencies. Another problem is how to address limitations on PREEMPT_RT like the big softirq/ bottom halves lock which can lead to high latencies.

I agree to abide by the anti-harassment policy

Yes

Primary author: SIEWIOR, Sebastian
Presenter: SIEWIOR, Sebastian
Session Classification: LPC Refereed Track
Track Classification: LPC Refereed Track
Profiling data structures

Tuesday, 13 September 2022 10:45 (45 minutes)

The Linux perf tools shows where, in terms of code, a myriad of events take place (cache misses, CPU cycles, etc), resolving instruction pointer addresses to functions in the kernel, BPF or user space.

There are tools such as ‘perf mem’ and ‘perf c2c’ that help translating data addresses where events take place to variables, and those will be described, both where the data comes from, such as AMD IBS, Intel PEBS and similar facilities in ARM that are recently being enabled in perf as well as how these perf tools use that data to provide ‘perf report’ output.

The open space is to do data structure profiling and annotating, that is to print a data structure and show how data accesses cause cache activity and in what order, mapping back not just for a variable, but to its type, to help in reorganizing data structures in an optimal fashion to avoid false sharing and maximize cache utilization.

The talk will try to show recent efforts into bringing together the Linux perf tools and pahole and the problems that remain in mapping back from things like cache misses to variables and types.

I agree to abide by the anti-harassment policy

Yes

Primary author:  CARVALHO DE MELO, Arnaldo (Red Hat Inc.)
Presenter:  CARVALHO DE MELO, Arnaldo (Red Hat Inc.)
Session Classification:  LPC Refereed Track
Track Classification:  LPC Refereed Track
RV: Where are we?

Monday, 12 September 2022 15:00 (45 minutes)

Over the last years, I've been exploring the possibility of verifying the Linux kernel behavior using Runtime Verification.

Runtime Verification (RV) is a lightweight (yet rigorous) method that complements classical exhaustive verification techniques (such as model checking and theorem proving) with a more practical approach for complex systems.

Instead of relying on a fine-grained model of a system (e.g., a re-implementation of instruction-level), RV works by analyzing the trace of the system’s actual execution, comparing it against a formal specification of the system behavior.

The research has become reality with the proposal of the RV interface[1]. At this stage, the proposal includes:

- An interface for controlling the verification;
- A tool and set of headers that enable the automatic code generation of the RV monitor (Monitor Synthesis);
- Sample monitors to evaluate the interface;
- A sample monitor developed in the context of the Elisa Project demonstrating how to use RV in the context of safety-critical systems.

In this discussion, we can talk about the steps missing for an RV merge and what are the next steps for the interface. Also, to discuss the needs of safety-critical and testing communities, to better understand what kind of models and what kind of new features they need.

[1] https://lore.kernel.org/all/cover.1651766361.git.bristot@kernel.org/

I agree to abide by the anti-harassment policy

Yes

Primary author: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)
Presenter: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)
Session Classification: LPC Refereed Track
Track Classification: LPC Refereed Track
TCP memory isolation on multi-tenant servers

On Linux, `tcp_mem` sysctl is used to limit the amount of memory consumed by active TCP connections. However that limit is shared between all the jobs running on the system. Potentially a low priority job can hog all the available TCP memory and starve the high priority jobs collocated with it. Indeed we have seen production incidences of low priority jobs negatively impacting the network performance of collocated high priority jobs.

Through cgroups, Linux does provide TCP memory accounting and isolation for the jobs running on the system but that comes with its own set of challenges which can be categorized into two buckets:

1. New and unexpected semantics of memory pressure and OOM for cgroup based TCP memory accounting.
2. Logistical challenges related to resource and quota management for large infrastructures running millions of jobs.

This is an ongoing work and new challenges keep popping up as we expand cgroup based TCP memory in our infrastructure. In this presentation we want to share our experience in tackling these challenges and would love to hear how others in the community have approached the problem of TCP memory isolation on their infrastructure.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  WARLOE, Christian (Google); BUTT, Shakeel (Google); WANG, Wei (Google)

Presenters:  WARLOE, Christian (Google); BUTT, Shakeel (Google); WANG, Wei (Google)

Session Classification:  LPC Refereed Track

Track Classification:  LPC Refereed Track
Open Printing MC

OpenPrinting has been improving the way we print in Linux. Over the years we have changed many conventional ways of printing and scanning. Over the last few years we have been emphasizing on the fact that driverless print and scan has made life easier however this does not make us stop improving. Every day we are trying to design new ways of printing to make your printing and scanning experience better than that of today.

To make this a success, Linux Plumbers conference is helping us immensely. This will be our fourth year of participation at the Plumbers.

The wonderful discussion that we had last year has helped us achieve the following:

Restarted and organized development activity on CUPS:

Since Apple has abandoned the CUPS development and Michael Sweet and Till Kamppeter have moved the upstream home of CUPS to OpenPrinting, the accumulated bugs got fixed and development restarted. Following our discussion from the OpenPrinting MC in 2021 we have started a 2.4.x series, will start 2.5.x near the end of this year and 3.x near the end of 2023, where the latter is a new generation not using the obsolete PPD (PostScript Printer Description, we have PDF workflow for a decade now) files, splitting daemons in user daemon and printer-sharing server, ... We also assigned release managers for each series, Zdenek Dohnal (Red Hat) for 2.4.x, Till Kamppeter for 2.5.x, and Michael Sweet for 3.x.

Coordinated and initiated work on Printer Setup Tool in the GNOME Control Center:

We also discussed on the 2021 MC how a printer setup tool for the New Architecture of pure IPP and now PPD-file-and-filter-based printer drivers any more should look like and this made the base for designing a new GNOME Control Center module which I am currently discussing with the Desktop Team and Design Team at Canonical for an implementation which is intended to submit to GNOME upstream. Coding is also carried out partially in Google Summer of Code projects.

All free software printer drivers available as Printer Applications now:

For the New Architecture for printing and scanning using pure IPP and not classic CUPS and SANE drivers any more we need a way to keep the non-driverless printers working and the solution are the Printer Applications, software emulations of driverless IPP printers which on their back end pass the jobs on to the physical printer and internally they convert incoming PDF or Raster into the printer’s native format. We have discussed concept and design on all our three MCs from 2019 to 2021 and with this Michael Sweet created Printer Application framework PAPPL and Till Kamppeter made use of it to retro-fit all free software classic printer drivers which are available for Debian Linux into 4 retro-fitting Printer Application Snaps in the Snap Store. This way it is assured that printers which currently work do not lose support and so we can smoothly switch over to the New Architecture.

Planned further development for scanning support in PAPPL:

To support multi-function devices containing a printer and a scanner but also for general scanner support which works with sandboxed, distribution-independent packaging we are extending the concept of Printer Applications to also provide scan services via IPP Scan or eSCL. For this we want to add scanning support to PAPPL.

Based on Bhavna Kosta’s GSoC work to start the scanning support in PAPPL we have discussed the further development and divided up the work to propose it as GSoC projects so that we can get the project completed this year.

Proposed Topics:
CUPS 2.5 and 3.0 Development (90 Mins)
Speaker: Michael Sweet
Topic Details: In this session we will first finalize the features and the roadmap for CUPS 2.5, especially finalize how we will support OAuth here, but also which further containerization/sandboxing/distribution-independent packaging systems we will explicitly support besides Snap, Docker for example. Then, most importantly, we will discuss the details of the CUPS 3.0 architecture, user daemon, printer-sharing daemon, how to integrate cups-filters, ...
Link: https://github.com/OpenPrinting/cups

3D Printing (45 Mins)
Speakers: Till Kamppeter, Michael Sweet, TBD
Topic Details: 3D printing gets more and more popular, not only in the industry but also in the DIYer’s garages. Especially for consumers but also for professional users it would be great to have an easy workflow, so that 3D printing “just works” like conventional 2D printing. Somehow one could think about clicking “Print” in the 3D design/CAD/CAM software and the object gets printed. As we already have CUPS filters for 2D printing data formats get converted to get the 3D printer’s native language in the end. Also we need some standard format for the applications to send, like as we have PDF for 2D printing. We will discuss here how such a 3D printing workflow could look like.

Testing and CI for OpenPrinting projects (45 Mins)
Speakers: Till Kamppeter, TBD
Topic Details: cups-filters (and also other projects on OpenPrinting) get larger and more and more complex with the time. It is always harder to overview the code and to predict the exact effects of a change, adding a feature or fixing a bug one can easily cause a regression. One tests the code but has one really tested all types of input, all settings, ... As human beings easily forget we need some automated testing, useful things being done when running “make check”, and tests being triggered on each GIT commit. Here we will discuss strategies of automatic testing. We will also take CUPS’ testing as an example and see whether we can proceed similarly on cups-filters.

Documentation for OpenPrinting projects (30 Mins)
Speakers: Till Kamppeter, TBD
Topic Details: Good documentation is something neglected a lot in the free software world. One is driving the coding of certain projects quickly forward to get something which actually works and one can try it out. One wants to get one’s new library finally released. But how should people know how to use it? Documentation! CUPS is well documented, but cups-filters (and pappl-retrofit) lack API documentation. Also the user documentation on the sites of distributions like Debian or SUSE are often much better than our upstream documentation. Here we will discuss how to solve this. API documentation generators for libraries? Upstreamizing documentation from distros to OpenPrinting? ...

Sandboxing/Containerizing alternatives to Snap for Printer Applications or CUPS (30 Mins)
Speaker Name: Till Kamppeter
Topic Details: There are Snaps of CUPS and 5 Printer Applications, but Snap has also disadvantages, most prominently the one-and-only Snap Store. Are there alternatives to create distribution-independent packages, especially of Printer Applications? Docker? Flatpak? AppImage? ...?

Participants who have confirmed they plan to attend:
Till Kamppeter (till.kamppeter@gmail.com)
Aveek Basu (basu.aveek@gmail.com)
Jeff Licquia jeff@licquia.org
Kate Stewart kstewart@linuxfoundation.org
Michael Sweet msweet@msweet.org
I agree to abide by the anti-harassment policy

**Primary authors:**  BASU, Aveek; SWEET, Michael (Lakeside Robotics Corporation); KAMPETER, Till (OpenPrinting / Canonical)

**Presenters:**  BASU, Aveek; SWEET, Michael (Lakeside Robotics Corporation); KAMPETER, Till (OpenPrinting / Canonical)

**Track Classification:**  LPC Microconference Track (CLOSED)
3D Printing

3D printing gets more and more popular, not only in the industry but also in the DIYer’s garages. Especially for consumers but also for professional users it would be great to have an easy workflow, so that 3D printing “just works” like conventional 2D printing. Somehow one could think about clicking “Print” in the 3D design/CAD/CAM software and the object gets printed. As we already have CUPS filters for 2D printing data formats get converted to get the 3D printer’s native language in the end. Also we need some standard format for the applications to send, like as we have PDF for 2D printing. We will discuss here how such a 3D printing workflow could look like.

I agree to abide by the anti-harassment policy

Yes

Primary authors: SWEET, Michael (Lakeside Robotics Corporation); KAMPPETER, Till (Open-Printing / Canonical)

Presenters: SWEET, Michael (Lakeside Robotics Corporation); KAMPPETER, Till (OpenPrinting / Canonical)

Session Classification: Open Printing MC

Track Classification: LPC Microconference: Open Printing MC
CUPS 2.5 and 3.0 Development

Wednesday, 14 September 2022 15:00 (1 hour)

In this session we will first finalize the features and roadmap for CUPS 2.5 which has a focus on OAuth and new container/sandbox/distribution-independent-packaging support. Then we will discuss the features and roadmap for CUPS 3.0 which implements a new simplified architecture for driverless printing.

https://github.com/OpenPrinting/cups

I agree to abide by the anti-harassment policy

Yes

Primary author: SWEET, Michael (Lakeside Robotics Corporation)
Presenter: SWEET, Michael (Lakeside Robotics Corporation)
Session Classification: Open Printing MC
Track Classification: LPC Microconference: Open Printing MC
Good documentation is something neglected a lot in the free software world. One is driving the coding of certain projects quickly forward to get something which actually works and one can try it out. One wants to get one’s new library finally released. But how should people know how to use it? Documentation! CUPS is well documented, but cups-filters (and pappl-retrofit) lack API documentation. Also the user documentation on the sites of distributions like Debian or SUSE are often much better than our upstream documentation. Here we will discuss how to solve this. API documentation generators for libraries? Upstreamizing documentation from distros to OpenPrinting? ...?

http://www.openprinting.org/
https://github.com/OpenPrinting/openprinting.github.io

I agree to abide by the anti-harassment policy

Yes

Primary authors: KAMPPETER, Till (OpenPrinting / Canonical); BASU, Aveek

Presenters: KAMPPETER, Till (OpenPrinting / Canonical); BASU, Aveek

Session Classification: Open Printing MC

Track Classification: LPC Microconference: Open Printing MC
Restricting access to IPP printers with OAuth2 framework

Wednesday, 14 September 2022 16:45 (45 minutes)

Native printing in Linux leverages Internet Printing Protocol (IPP), the standard supported by the vast majority of printers available on the market. While it is quite sufficient for personal use, it has some drawbacks for enterprise customers, such as a lack of standard, OAuth2-based user’s authorization mechanisms necessary for print management systems. We tried to address this issue by developing a standard solution that can be implemented in various IPP-based systems.

The problem can be defined as a general protocol between an IPP client and a printing system, consisting of IPP printers and an authorization server. To get access to the printer’s resources, the IPP client redirects the user to the authentication webpage provided by the authorization server. When the user authenticates successfully, the server issues an access token that must be used by the IPP client during communication with the printer. The printer uses the access token to verify the user’s access rights.

We would like to discuss security-related issues of this problem and propose a general protocol working for printing systems with different architectures. Other possible solutions will also be discussed.

I agree to abide by the anti-harassment policy

Yes

Primary author: PAWLICZEK, Piotr (Google)
Presenter: PAWLICZEK, Piotr (Google)
Session Classification: Open Printing MC
Track Classification: LPC Microconference: Open Printing MC
There are Snaps of CUPS and 5 Printer Applications, but Snap has also disadvantages, most prominently the one-and-only Snap Store and also that some desktop apps start up slowly. Are there alternatives to create distribution-independent packages, especially of Printer Applications? Docker? Flatpak? AppImage? ...

https://github.com/OpenPrinting/

I agree to abide by the anti-harassment policy

Yes
cups-filters (and also other projects on OpenPrinting) get larger and more and more complex with the time. It is always harder to overview the code and to predict the exact effects of a change, adding a feature or fixing a bug one can easily cause a regression. One tests the code but has one really tested all types of input, all settings, ... As human beings easily forget we need some automated testing, useful things being done when running "make check", and tests being triggered on each GIT commit. Here we will discuss strategies of automatic testing. We will also take CUPS testing as an example and see whether we can proceed similarly on cups-filters.


I agree to abide by the anti-harassment policy

Yes

Primary author: KAMPPETER, Till (OpenPrinting / Canonical)

Presenters: SWEET, Michael (Lakeside Robotics Corporation); KAMPPETER, Till (OpenPrinting / Canonical)

Session Classification: Open Printing MC

Track Classification: LPC Microconference: Open Printing MC
Power Management and Thermal Control MC

The Power Management and Thermal Control microconference focuses on frameworks related to power management and thermal control, CPU and device power-management mechanisms, and thermal-control methods. In particular, we are interested in extending the energy-efficient scheduling concept beyond the energy-aware scheduling (EAS), improving the thermal control framework in the kernel to cover more use cases and making system-wide suspend (and power management in general) more robust.

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

Suggested topics:

- Energy-efficient scheduling beyond EAS (Len Brown).
- Per-CPU idle injection from user space for thermal control (Srinivas Pandruvada).
- A generic energy model description (Daniel Lezcano).
- Extending the DTPM framework by adding more supported devices to it (Daniel Lezcano).
- Thermal control core code improvements (Daniel Lezcano, Rafael Wysocki).
- Combining DTPM with the thermal control framework (Daniel Lezcano).
- Generic DVFS support for SCMI-based platforms (Ulf Hansson).
- Improving the genpd governor for CPUs (Ulf Hansson).
- More integration between PM-runtime and system-wide PM (Rafael Wysocki).

More topics will be added based on CiP for this microconference.

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

Presenter: WYSOCKI, Rafael (Intel Open Source Technology Center)

Track Classification: LPC Microconference Track (CLOSED)
A generic energy model description

Wednesday, 14 September 2022 16:50 (25 minutes)

The energy model is dispatched through implicit values in the device tree and the power values are deduced from the formula \( P = C \times F \times V^2 \) by the energy model in the kernel. Unfortunately, the description is a bit fuzzy if the device is using the Adaptative Voltage Scaling or not performance based, as a battery or a backlight. On the other side, complex energy models exist on out of tree kernels like Android, meaning there is a need for such a description.

A generic energy model description will help to have a clear of view of the power contributors for thermal, power consumers for accounting and performance.

I agree to abide by the anti-harassment policy

Yes

Primary author: LEZCANO, Daniel (Linaro)
Presenter: LEZCANO, Daniel (Linaro)
Session Classification: Power Management and Thermal Control MC
Track Classification: LPC Microconference: Power Management and Thermal Control MC
Combining DTPM with the thermal control framework

Wednesday, 14 September 2022 15:50 (25 minutes)

The DTPM framework and the thermal control framework are using the same algorithm and mechanism when the power numbers are involved. That results in duplicated code. The DTPM framework interacts with the user space but nothing prevent to provide an in-kernel API where the power based cooling devices can directly act on. That will result in a simpler code and very explicit power value usage. In addition, if the SCMI is supported by DTPM, no changes will be needed in the thermal cooling devices. The result will be one generic power based cooling device supporting any device (devfreq, cpufreq, ...) with an energy model (DT or SCMI based).

I agree to abide by the anti-harassment policy

Yes

Primary author:  LEZCANO, Daniel (Linaro)
Presenter:  LEZCANO, Daniel (Linaro)
Session Classification:  Power Management and Thermal Control MC
Track Classification:  LPC Microconference: Power Management and Thermal Control MC
CPUfreq/sched and VM guest workload problems

Wednesday, 14 September 2022 17:15 (25 minutes)

Running a workload on VM results in very disparate CPUfreq/sched behavior compared to running the same workload on the host. This difference in CPUfreq/sched behavior can cause significant power/performance regression (on top of virtualization overhead) for a workload when it is run on a VM instead of the host.

This talk will highlight some of the CPUfreq and scheduler load tracking issues/questions both at the guest thread level and at the host vCPU thread level and explore potential solutions.

I agree to abide by the anti-harassment policy

Yes

Primary author: KANNAN, Saravana
Presenter: KANNAN, Saravana
Session Classification: Power Management and Thermal Control MC
Track Classification: LPC Microconference: Power Management and Thermal Control MC
Energy model accuracy

Wednesday, 14 September 2022 16:25 (25 minutes)

Energy-aware scheduling (EAS) introduced a simply, yet at that time, effective energy model to help guide task scheduling decisions and DVFS policies. As CPU core micro-architecture has evolved the error bars on the energy model to grow potentially leading to sub-optimal task placement. Are we getting to the point where we need to enhance the energy model, or look at new ways to bias task placement decisions?

I agree to abide by the anti-harassment policy

Yes

Primary author:  RASMUSSEN, Morten (Arm)
Presenter:    RASMUSSEN, Morten (Arm)
Session Classification:  Power Management and Thermal Control MC
Track Classification:  LPC Microconference: Power Management and Thermal Control MC
Fine grain frequency control with kernel governors

We introduced AMD P-State kernel CPUFreq driver early of this year that is using ACPI CPPC based fine grain frequency control instead of legacy ACPI P-States, and it is merged into kernel 5.17. The AMD P-State will be used on most of the Zen2/Zen3 and future AMD processors.

There are two types of hardware implementations: “full MSR solution” and “shared memory solution”. “full MSR solution” provides the architected MSR set of CPPC level registers to manage the performance hints which is the fast way to control frequency updates. However, “shared memory solution” is that CPUs only support a mailbox model for the CPPC registers in the system memory, we have to map the system memory which shared with CPPC and use kernel RCU locks to manage synchronization with the way in kernel ACPI CPPC libraries.

The initial driver is developed on “full MSR solution” processors and can get better performance per watt scaling in some CPU benchmarks. However, we face the performance drops which compared with legacy ACPI CPUFreq driver in “shared memory solution” processors. The traditional kernel governors such as ondemand, schedutil, and etc. might not be fully suitable for the fine grain frequency control, because there were 166~255 performance states in AMD P-State that compared with only 3 ACPI P-States. CPU CFS scheduler governor might face more frequently performance change in AMD P-State.

In following days, we will support more features include Energy-Performance Preference which is the balance between performance and energy and Preferred Core which is to have a best performance single core/thread in one package. We want to discuss how to refine the CPU scheduler or kernel governors to allow the platform to specify an order of preference that processes should be scheduled on the different cores.

In this session, we would like to have a discussion to how to improve the kernel governors which have better performance per watt scaling on fine grain frequency control, how to leverage the new Energy-Performance Preference and Preferred Core features to improve the Linux kernel performance and power efficiency.

For details of AMD P-State, please see.

References:
1. https://lore.kernel.org/lkml/20211224010508.110159-1-ray.huang@amd.com/
3. https://lore.kernel.org/linux-pm/a0e932477e9b826c0781dda1d0d295e57f904cc.camel@suse.cz/

Primary author: HUANG, Ray (AMD)

Presenter: HUANG, Ray (AMD)

Session Classification: Power Management and Thermal Control MC
Frequency-invariance gaps in current kernel

Wednesday, 14 September 2022 15:00 (25 minutes)

The kernel’s load tracking scales the observed load by the frequency the CPU is running at, this scaled value is used to determine how loaded a CPU truly is and how its frequency should change. Currently, on X86, four-core turbo level is used as the maximum ratio for every CPU. However, Intel client Hybrid platforms have Pcores and Ecores, and Intel server platforms with Intel-Speed-Select-Technology enabled have high-priority cores and low-priority cores.

The Pcore/High-Priority-Core can run at higher maximum frequency, while the remaining cores can only run at lower maximum frequency.

In these cases, unified maximum ratio for every CPU doesn’t reflect the truth and brings unfairness to the load balance.

Also, the current code doesn’t handle special cases where the frequencies for one or more CPUs are clamped via sysfs.

We would like to demonstrate the impacts brought by those issues for further discussion.

I agree to abide by the anti-harassment policy

Yes

Primary authors: ZHANG, Rui; CHEN, Yu

Presenter: ZHANG, Rui

Session Classification: Power Management and Thermal Control MC

Track Classification: LPC Microconference: Power Management and Thermal Control MC
Isolation for broken hardware during system suspend

Wednesday, 14 September 2022 18:20 (10 minutes)

When a device is broken and return failure during suspend, the whole system is blocked from entering system low-power states.
Thus user loses the top one power saving feature on their systems due to non-fatal device failures for their usage.
In this case, making the system suspend work with tolerance of device failures is a gain. This may be achieved by a) disabling the device on behalf of BIOS, b) do device unbind upon suspend, c) skip the devices’ suspend callback or d) ignore the suspend callback failures, etc. It also helps when debugging device related suspend issues reported.

I agree to abide by the anti-harassment policy

Yes

Primary author: ZHANG, Rui
Presenter: ZHANG, Rui
Session Classification: Power Management and Thermal Control MC
Track Classification: LPC Microconference: Power Management and Thermal Control MC
Per core/cpu idle injection is very effective in controlling thermal conditions, without using CPU offline which has its own drawbacks. Since CPU temperature ramp up and ramp down is very fast, idle injection provides a fast enter and exit path.

Linux has support for per core idle injection for a while ([https://www.kernel.org/doc/html/latest/driver-api/thermal/cpu-idle-cooling.html](https://www.kernel.org/doc/html/latest/driver-api/thermal/cpu-idle-cooling.html)). But this solution has some limitations as it blocks soft IRQs and have negative effect on pinned timers. I am working on a solution for unblocking soft IRQ issue but there is no good solution for pinned timers yet.

The purpose of this discussion is to find possible solutions for the above issues.

I agree to abide by the anti-harassment policy

Yes

**Primary author:** PANDRUVADA, Srinivas

**Presenter:** PANDRUVADA, Srinivas

**Session Classification:** Power Management and Thermal Control MC

**Track Classification:** LPC Microconference: Power Management and Thermal Control MC
Unified structure for thermal zone device registration

Wednesday, 14 September 2022 15:25 (25 minutes)

To register a thermal zone device, the number of parameters required has been increased from 4 when it is first introduced to 8, and people are still willing to add more. This is hard to maintain because every time a new parameter is needed, either a new wrapper is added, or all the current thermal zone drivers need to be updated. Plus, there is already a structure, aka “struct thermal_zone_params”, available, and it has already been used for registration time configuration.

Here, I propose to use one structure for registration phase configuration, or combine with the existing struct thermal_zone_params for better maintenance.

I agree to abide by the anti-harassment policy

Yes

Primary author: ZHANG, Rui
Presenter: ZHANG, Rui
Session Classification: Power Management and Thermal Control MC
Track Classification: LPC Microconference: Power Management and Thermal Control MC
The real-time and scheduling micro-conference joins these two intrinsically connected communities to discuss the next steps together.

Over the past decade, many parts of PREEMPT_RT have been included in the official Linux codebase. Examples include real-time mutexes, high-resolution timers, lockdep, ftrace, RCU_PREEMPT, threaded interrupt handlers and more. The number of patches that need integration has been significantly reduced, and the rest is mature enough to make their way into mainline Linux.

The scheduler is the core of Linux performance. With different topologies and workloads, it is not an easy task to give the user the best experience possible, 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 microconference:

- The Real-time Linux Analysis tool was merged in 5.17[1]
- Progress on tools to facilitate maintenance of the stable RT releases.
- Progress on the full mainline merge, but some challenges were raised and more is to be done.
- Core scheduling has been merged. [2]
- Progress in the latency nice scheduling feature [3]

This year’s topics to be discussed include:

- How to scale PREEMPT_RT for very-large systems
- Improve overall system partitioning for real-time HPC workloads
- New tools for PREEMPT_RT analysis.
- How do we teach the rest of the kernel developers how not to break PREEMPT_RT?
- The usage of PREEMPT_RT on safety-critical systems: what do we need to do?
- The merge’s status, and how can we resolve the last issues that block the merge.
- Latency nice scheduling feature
- Better support for new processors
- What’s next?

[3] https://lore.kernel.org/lkml/20220311161406.23497-1-vincent.guittot@linaro.org/

I agree to abide by the anti-harassment policy
Primary authors: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.); LELLI, Juri (Red Hat); ROSTEDT, Steven; GUITTOT, Vincent (Linaro); STEWART, Kate (Linux Foundation)

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

Track Classification: LPC Microconference Track (CLOSED)
Bringing Energy-Aware Scheduling to x86

Tuesday, 13 September 2022 15:30 (25 minutes)

Energy-Aware Scheduling (EAS) is not a straight fit for x86 hybrid processors. Thus, x86 hybrid processors do not make use of EAS yet. A large range of turbo frequencies, inter-CPU dependencies, simultaneous multithreading, and instruction-specific differences in throughput makes it difficult to feed the scheduler with a simple, timely, accurate model of CPU capacity. Dependencies between CPUs and other on-chip components makes it difficult to create an energy model. The widespread use of hardware-controlled frequency-scaling on systems based on Intel processors needs to be reconciled with a model in which the kernel controls the operating point of the CPU.

The goal of this talk is to discuss the level of support from hardware, the challenges of EAS on x86, and proposed solutions to provide simple capacity and energy models that are sufficiently accurate for the scheduler to use.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  BROWN, Len (Intel Open Source Technology Center);  NERI, Ricardo (Intel Corporation)

Presenters:  BROWN, Len (Intel Open Source Technology Center);  NERI, Ricardo (Intel Corporation)

Session Classification:  Real-time and Scheduling MC

Track Classification:  LPC Microconference: Real-time and Scheduling MC
Latency hints for CFS task

Tuesday, 13 September 2022 15:55 (30 minutes)

RT schedulers are traditionally used for everything concerned with the latency but it’s sometimes not possible to use RT for all parts of the system because of the variance of the runtime or the trust of some parts as an example. At the opposite side, some apps don’t care at all about latency and preemption of the running task but prefer to let the current task move forward.

The latency nice priority aims to let userspace to set such latency requirements for CFS tasks but one difficulty is to find a suitable interface that stays meaningful for user but is not tied to one particular implementation. [1] has resurrected the latency_nice interface with an implementation that is system agnostic.

This talk will present the current status and how to move forward on the interface.

https://lore.kernel.org/lkml/20220512163534.2572-7-vincent.guittot@linaro.org/T/

I agree to abide by the anti-harassment policy

Yes

Primary author: GUITTOT, Vincent (Linaro)
Presenter: GUITTOT, Vincent (Linaro)
Session Classification: Real-time and Scheduling MC
Track Classification: LPC Microconference: Real-time and Scheduling MC
Limit the idle CPU search depth and use CPU filter during task wake up

Tuesday, 13 September 2022 17:10 (25 minutes)

When a task is woken up in the last level cache (LLC) domain, the scheduler tries to find an idle CPU for the task. But when the LLC domain is fully busy, the search for an idle CPU may be in vain, adding long latency to the task wakeup and yet does not lead to an idle CPU. The latency gets worse when the number of CPUs in the LLC increases, which will be the case for future platforms. During LPC2021 there was a discussion on how to find the idle CPU effectively. This talk is an extended discussion on that. We will illustrate how we encountered the issue and how to debug this issue on a high core count system. We’ll introduce the proposal that leverages the util_avg of the LLC to decide how much effort is spent to scan for idle CPUs. We’ll also present a proposal to filter out the busy CPUs so as to speed up the scan. We’ll share the current test data using the mechanism. We hope to get feedback on advice/feedback on tuning the scan policy and making it viable for upstreaming.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  YU, Chen; WU, Yun (Bytedance)

Presenters:  YU, Chen; WU, Yun (Bytedance)

Session Classification:  Real-time and Scheduling MC

Track Classification:  LPC Microconference: Real-time and Scheduling MC
Linux Kernel Scheduling and split-LLC architectures:
Overview, Challenges and Opportunities

Tuesday, 13 September 2022 16:45 (25 minutes)

Linux Task Scheduler has seen several enhancements to make task scheduling better and smarter for split last level cache (split-LLC) environments. With wider adoption of the chiplet-like technology in current and future processors, these continued efforts become key to squeeze the most out of the silicon.

Work has already gone in to accurately model the domain topology for split-LLC architectures: Optimizing task wakeups to target cache-hot LLCs, reducing cross-LLC communication. NUMA imbalance metrics have been reworked to enable better task distribution across NUMA nodes with multiple LLCs. These enhancements have enabled several workloads to benefit from architectural advantages of split-LLCs. That being said, there is still lot of performance left on the table.

In this talk we provide an overview of recent scheduler changes that have benefitted workloads in a split-LLC environment. We will describe challenges, opportunities and some ambitious ideas to make the Linux Scheduler more performant on split-LLC architectures.

I agree to abide by the anti-harassment policy

Yes

Primary authors: SHENOY, Gautham R (AMD Inc.); NAYAK, Prateek (AMD Inc.)

Presenters: SHENOY, Gautham R (AMD Inc.); NAYAK, Prateek (AMD Inc.)

Session Classification: Real-time and Scheduling MC
Linux needs a Scheduler QOS API – and it isn’t nice(2)

Tuesday, 13 September 2022 17:35 (25 minutes)

Optimal task placement decisions and hardware operating points impact application performance and energy efficiency.

The Linux scheduler and the hardware export low level knobs that allow an expert to influence these settings. But that expert needs to know details about the hardware, about the Linux scheduler, and about every (other) task that is running on the system.

This is not a reasonable demand for multi-platform applications. Here we look at what, say Chromium, must do to run on Linux, Windows, and MacOS; and how we can make it easier for apps to run more optimally on Linux.

I agree to abide by the anti-harassment policy

Yes

Primary author: BROWN, Len (Intel Open Source Technology Center)
Presenter: BROWN, Len (Intel Open Source Technology Center)
Session Classification: Real-time and Scheduling MC
Track Classification: LPC Microconference: Real-time and Scheduling MC
PREEMPT_RT Q&A with tglx

Tuesday, 13 September 2022 18:00 (30 minutes)

In this topic, Thomas Gleixner will answer all the questions about the present of future of the PREEMPT_RT, mainly about the status of the merge and how things will work after the merge.

I agree to abide by the anti-harassment policy

Yes

Primary author: GLEIXNER, Thomas
Presenter: GLEIXNER, Thomas
Session Classification: Real-time and Scheduling MC

Track Classification: LPC Microconference: Real-time and Scheduling MC
rtla: what is next?

Tuesday, 13 September 2022 15:00 (30 minutes)

Presented last year, RTLA made its way to the kernel set of tools. RTLA includes an interface for timerlat and osnoise tracers in the current state. However, the idea is to expand RTLA to include a vast set of real-time Linux analysis tools, combining tracing and methods to stimulate the system.

In this discussion, we can talk about ways to extend tracers and rtla, including:

- Supporting histogram for ftrace tracers (timerlat events)
- Adding new interface for tracers, like hwlat
- Supporting trace-cmd file support to allow ftrace & osnoise/timerlat/hwlat in parallel
- Adding smi counters for timerlat
- Adding other tools
- Adding pseudo-random task simulator

But the main idea is to hear what people have to say about how to make the tool even better!

I agree to abide by the anti-harassment policy

Yes

Primary author: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)
Presenter: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)
Session Classification: Real-time and Scheduling MC
Track Classification: LPC Microconference: Real-time and Scheduling MC
RISC-V MC

The RISC-V software ecosystem continues to grow tremendously with many RISC-V ISA extensions being ratified last year. There are many features supporting the ratified extensions that are under development i.e. svpbmt, sstc, sscofpmf, cbo. We would like to continue the RISC-V MC platform to discuss these issues with a wider community to arrive at a solution as we have done in the past.

In the past, these discussions have been very fruitful resulting in several major milestones. A few of those are described below.

1. KVM upstreaming
2. An agreement on D1 support
3. ACPI/UEFI improvement
4. We have a concrete interpretation of what is an errata, according to the ISA specs.

Here are a few of the expected topics and current problems in RISC-V Linux land that we would like to cover this year.

Various specification updates and plans for supporting them, with candidates including SBI, EFI, memory models (WMO, IO, etc), IOMMU, TEE
How we're going to handle user-visible errata, with the most notable current example being the many present in the D1.
How to move forward with support for the V extension, including probing from userspace (VLENMAX, performance, etc). A similar set of issues will likely arise for the B and K extensions.
How we're going to handle runtime probing of various performance knobs in the kernel, like strings.h and locks.
What, if any, rules we’re going to have for portable/distro kernels? What about non-portable kernels?
How are we going to deal with the ABI fallout from the pre-formal-model GCC interpretation of WMO (and by the time Plumbers comes around, maybe TSO as well)?
Is using WRS for pthread_mutex() sane? Either way, how do we handle mtime in userspace?
Ongoing development for Nested hypervisor

Possible participant list:

Palmer Dabbelt
Atish Patra
Dylan Reid
Ravi Sahita
Kumar Shankaran
Guro Ren
Drew Fustini
Anup Patel
Nick Kossifidis
Paolo Bonzini
Arnd Bergman
Al Stone
Philipp Tomsich
Alexandre Ghiti
Christoph Müllner
Heiko Stübner
Come join us and participate in the discussion on how we can improve the support for RISC-V in the Linux kernel.

MC Runners: Atish Patra atishp@rivosinc.com, Palmer Dabbelt palmer@rivosinc.com

1 https://www.phoronix.com/scan.php?page=news_item&px=Linux-5.16-KVM-RISC-V
2 https://lore.kernel.org/lkml/20220331100139.GG23422@lst.de/T/
3 https://github.com/riscv-non-isa/riscv-acpi
4 https://github.com/riscv-non-isa/riscv-uefi

I agree to abide by the anti-harassment policy

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

Presenters: PATRA, ATISH (Western Digital); DABBELT, Palmer (Google)

Track Classification: LPC Microconference Track (CLOSED)
Confidential Computing for RISC-V-based Platforms

Tuesday, 13 September 2022 17:00 (30 minutes)

Confidential computing aims to protect data in use on computing platforms. Via confidential computing mechanisms, we aim to remove host software (OS/VMM, service VMs and firmware), other tenants (VMs), host software developers, operators and administrators of multi-tenant systems from the Trusted Computing Base (TCB) of tenant workloads. For RISC-V-based platforms, we propose an Application Platform-Trusted Execution Environment (AP-TEE) reference architecture and the ABI between host software and the TCB components on the platform (a TEE Security Manager aka TSM). The interfaces describe the use of the RISC-V Hypervisor extension to enforce confidentiality for virtualized workloads as well as the hardware changes that should be considered to enforce mitigations for a threat model. The proposal discusses the ABI proposed for TSM-Host/VMM interactions (TH-ABI) and TSM-Guest interactions (TG-ABI), and directions of hardware/ISA extensions. In addition to the proposed normative specifications, the proposal will document implementation-specific guidelines and relevant standard protocols for attestation to assist implementers of the AP-TEE confidential computing capability on RISC-V platforms.

I agree to abide by the anti-harassment policy

Yes

Primary author: SAHITA, RAVI (Rivos)
Presenter: SAHITA, RAVI (Rivos)
Session Classification: RISC-V MC

Track Classification: LPC Microconference: Real-time and Scheduling MC
RISC-V ACPI and UEFI Updates

Tuesday, 13 September 2022 15:40 (30 minutes)

To support server class Operating Systems, ACPI needs to be supported on RISC-V. We had discussed what it takes to enable basic ACPI support for RISC-V in last year’s LPC. In this session, we discuss the progress we made on

1) ACPI specification ECRs
2) Linux/Qemu patches required to support basic ACPI
3) RISCV_EFI_BOOT_PROTOCOL support required to enable ACPI
4) New RIMT proposal for RISC-V IOMMU

I agree to abide by the anti-harassment policy

Yes

Primary author:  V L, Sunil
Presenter:  V L, Sunil
Session Classification:  RISC-V MC
Track Classification:  LPC Microconference: RISC-V MC
RISC-V ftrace: working with preemption

Tuesday, 13 September 2022 18:00 (30 minutes)

We found several issues linked inherently with ISA of RISC-V itself when using ftrace after turning on kernel preemption. In RISC-V, we must use 2 instructions to perform a jump to a target which is further than 4KB, and we cannot promise any 2 instructions being executed on the same process context if preemption is enabled. However, this is how we patch code in ftrace in current implementation. Thus, we proposed a change that could possibly solve it, making kernel preemption work with ftrace. The patch has been published on the mailing list. We would like to share and discuss our thoughts on LPC. And the talk will cover following content:

- Current Implementation of RISC-V ftrace
- How does stop_machine() work
- Reviews of ftrace Implementations on other Architectures
- Mixing with Kernel Preemption
- Limitation of RISC-V ftrace due to RISC-V ISA
- Possible Solutions to Enable ftrace with a Preemptible Kernel
- Proposed Solution
- Experiment and Results

I agree to abide by the anti-harassment policy

Yes

Primary author: CHIU, Tao
Presenter: CHIU, Tao
Session Classification: RISC-V MC
Track Classification: LPC Microconference: RISC-V MC
It’s always been challenging for operating systems to tell the (userspace) programs about the underlying hardware capabilities on RISC-V platforms.

For most computer architectures, a bit vector may suffice since the 64 bit platforms are most likely the enhancement of their 32 bit predecessors.

Yet sadly that’s more than complicated for RISC-V, which has a much more diverse ecosystem.

In this talk we would like to discuss a proof of concept on Linux in which we utilize the vDSO data section as the intermediate for showing hardware capabilities - either directly accessing it via a pointer passed from HWCAP2 or a vdso function call (with an architecture-specific syscall as the fallback).

We will tell the story about the good, the bad and the ugly sides of this approach and we sincerely hope to hear the comments from the community.

I agree to abide by the anti-harassment policy

Yes

Primary author:  TSAI, "Ruinland" ChuanTzu (SiFive)
Presenter:  TSAI, "Ruinland" ChuanTzu (SiFive)
Session Classification:  RISC-V MC
Track Classification:  LPC Microconference: RISC-V MC
The kernel comes with its own implementation of common routines of the C libraries (memcpy, memcmp, strcmp, etc.). Since the kernel already has a rich infrastructure to handle architecture and platform-specific features, such as code patching or static calls, there is an opportunity to speed up these routines for certain platforms. The goal of this discussion is to identify the routines that can benefit from such an optimized implementation, what ISA extensions would be in focus (e.g. only stateless - so no vector instructions?) and what a reasonable grouping of target platforms could look like (e.g. a generic C implementation, one for each profile plus one for additional fast unaligned memory accesses).

I agree to abide by the anti-harassment policy

Yes

Primary author: Mr STUEBNER, Heiko (Vrull)
Presenter: Mr STUEBNER, Heiko (Vrull)
Session Classification: RISC-V MC
Track Classification: LPC Microconference: RISC-V MC
The goal of kconfig.socs originally was to have SOC_FOO symbols so that a user "can just push a button and have everything they need to boot", which was implemented via selects. This sort of behaviour for a kconfig symbol is at odds to other architectures and not maintainable in the long term as the number of SoCs grows and/or the select dependencies change.

As things stand, different SOC_FOO symbols have different behaviour:
- some directly select the drivers if a prereq is set
- others use SOC_FOO symbol as a prerequisite to expose drivers during configuration
- some enable prerequisites to ensure drivers will be exposed & rely on a depends on SOC_FOO + default SOC_FOO combination in the driver’s kconfig entry to enable the driver itself.

It would be great to have a discussion and settle on a single, consistent approach for SOC_FOO symbols (or if someone has a better idea for a replacement…) before it becomes unwieldy.

Secondly, depending on what is decided on, what should the scope of the symbol be?
Should it enable a bare minimum for boot, and then expose other options as possibilities?
Or should it turn on all bells/whistles for that SoC?

**Primary authors:**  DOOLEY, Conor; DABBELT, Palmer (Google)

**Presenter:**  DOOLEY, Conor

**Session Classification:**  RISC-V MC

**Track Classification:**  LPC Microconference: RISC-V MC
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 aims to bring it into the kernel since it has a key property that makes it very interesting to consider 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.

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:

- Bringing Rust into the kernel (e.g. status update, next steps...).
- Use cases for Rust around the kernel (e.g. drivers, subsystems...).
- Integration with kernel systems and infrastructure (e.g. wrapping existing subsystems safely, build system, documentation, testing, maintenance...).

Possible Rust topics:

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

I agree to abide by the anti-harassment policy

Primary authors:  OJEDA, Miguel;  ALMEIDA FILHO, Wedson

Presenters:  OJEDA, Miguel;  ALMEIDA FILHO, Wedson

Track Classification:  LPC Microconference Track (CLOSED)
Buffer & Farewell from the Rust MC

Session Classification: Rust MC
Linux Rust NVMe Driver Status Update

Monday, 12 September 2022 17:00 (30 minutes)

Rust for Linux aims to bring in Rust as a second programming for the Linux Kernel. The Rust for Linux project is making good progress towards being included in upstream Linux sources.

In this talk we discuss status of the Rust NVMe driver. The Rust NVMe driver is interesting as a reference implementation of a high performance driver because NVMe already has a mature and widely deployed driver in the Linux kernel that can be used as a baseline for benchmark purposes.

We discuss Rust language abstractions required to enable the Rust NVMe driver, benchmark results comparing the Rust implementation to the C implementation, and future work.

I agree to abide by the anti-harassment policy

Yes

Primary author: HINDBORG, Andreas (Western Digital)

Presenter: HINDBORG, Andreas (Western Digital)

Session Classification: Rust MC

Track Classification: LPC Microconference: Rust MC
Rust is a systems programming language with desirable properties in the context of the Linux kernel, such as no undefined behavior in its safe subset (when unsafe code is sound), including memory safety and the absence of data races.

Rust for Linux is a project that aims to bring Rust support to the Linux kernel as a first-class language. This means providing support for writing kernel modules in Rust, such as drivers or filesystems, with as little unsafe code as possible (potentially none). That is, it prevents misusing kernel APIs with respect to memory-safety.

This session will give an status update on the project:

- What features are currently supported.
- Infrastructure improvements.
- Rust unstable (nightly) features status.
- Rust ecosystem news: language, toolchains, etc.
- Planned features and future.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  OJEDA, Miguel;  ALMEIDA FILHO, Wedson

Presenters:  OJEDA, Miguel;  ALMEIDA FILHO, Wedson

Session Classification:  Rust MC

Track Classification:  LPC Microconference: Rust MC
Toolchain support for the Rust language is a question central to adopting Rust in the Linux kernel. So far, the LLVM-based rustc compiler has been the only option for Rust language compilers. GCC Rust is a work-in-progress project to add a fully-featured front-end for Rust to the GNU toolchain. As a part of GCC, this compiler benefits from the common GCC flags, optimizations, and back-end targets.

As work on the project continues, supporting Linux kernel development and the adoption of Rust in the kernel has become an essential guiding target. In this discussion, we would like to introduce the project’s current state and consult with Rust-for-Linux developers about their needs from the toolchain; for example, how to prioritize work in Rust GCC or how we handle language versioning. Some particular topics for discussion:

- Procedural macros
- libcore, liballoc
- Language versioning
- Debug integration
- Unstable language features
- Bindings and FFI

I agree to abide by the anti-harassment policy

Yes

Primary authors:  HERRON, Philip (Embecosm);  FAUST, David (Oracle)

Presenters:  HERRON, Philip (Embecosm);  FAUST, David (Oracle)

Session Classification:  Rust MC

Track Classification:  LPC Microconference: Rust MC
We are very excited (and impatient) to have Rust supported in the Kernel. In fact we are so impatient we decided to develop a means of getting Rust in the Kernel today, using eBPF!

Aya is an eBPF library built with a focus on operability and developer experience. It allows for both user-land and kernel-land programs to be written in Rust - and even allows for sharing of code between the two! It has minimal dependencies and supports BPF Compile-Once:Run-Anywhere (CO:RE). When linked with musl, it creates a truly portable, self-contained binary that can be deployed on many Linux distributions and kernel versions.

In this talk we would like to deep dive into the present state of Aya, with focus on:

- How it works
- Currently supported features
- How Rust for Linux and Aya can benefit from each other
- Our future plans, which include changes in Rust ecosystem

I agree to abide by the anti-harassment policy

Yes

**Primary authors:** Mr TUCKER, Dave (Red Hat); ROSTECKI, Michal (Deepfence Inc)

**Presenters:** Mr TUCKER, Dave (Red Hat); ROSTECKI, Michal (Deepfence Inc)

**Session Classification:** Rust MC

**Track Classification:** LPC Microconference: Rust MC
The Rust programming language is becoming more and more popular: it’s even considered as another language allowed in the Linux kernel. That brought up the question of architecture support as the official Rust compiler is based on LLVM.

This project, rustc_codegen_gcc, is meant to plug the GCC backend to the Rust compiler frontend as a relatively low-effort: it’s a shared library reusing the same API provided by the Rust compiler as the cranelift backend.

As such, it could be used by some Linux projects as a way to provide their Rust softwares to more architectures.

This talk will present this project, its progress and will feature a discussion about what needs to be done to start using it for projects like Rust for Linux.

I agree to abide by the anti-harassment policy

Yes

Primary author:  BOUCHER, Antoni
Presenter:  BOUCHER, Antoni
Session Classification:  Rust MC

Track Classification:  LPC Microconference: Rust MC
The Integration of Rust with Kernel Testing Service

Monday, 12 September 2022 17:30 (30 minutes)

Rust for Linux aims to bring Rust into the kernel as the second programming language. With the great advancing of this target, a corresponding testing service for Rust is becoming a potential requirement.

0-Day CI team has been working closely with the maintainers of Rust for Linux to integrate Rust into kernel test robot. We’d like to share our experience of enabling Rust test. Here are some of the progress we have made:

- Kernel test robot is a bisection-driven CI, we not only scan build errors, but also run bisections to look for the first bad commits which introduced the errors. To maintain the capability of bisection, we setup automatic upgrade and adaptive selection for Rust toolchain, thus to match the required toolchain version of different commits during the process of bisection.

- We provide both random config and a specific config with all Rust samples enabled to have different level of code coverage for Rust in kernel.

Most of the work we have done is about building kernel with Rust enabled, and we are considering runtime test in the next step. We are also interested in various topics which may help to enhance Rust test. Some further work we are looking forward to happen are:

- Boot/fuzzing testing for Rust code such as leveraging syzkaller.
- Functional testing for core Rust code and modules, which could be part of common framework like kunit/kselftests to be easily used in CI service.
- Collect and aggregate Rust code coverage data in kernel to better design and execute tests.
- Wrap a tool to setup Rust environment based on min-tool-version.sh for consistent compiling and issue reproducing.
- Testing for the potential impact of different compiling options of Rust, such as optimization level and build assert config.

We hope that our work can give inspiration to other CIs wishing to integrate it, and help to facilitate the development of Rust for Linux.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  Mr LIU, Yujie (Intel); Mr LI, Philip (Intel)

Presenter:  Mr LIU, Yujie (Intel)

Session Classification:  Rust MC

Track Classification:  LPC Microconference: Rust MC
Welcome to the Rust MC

Session Classification: Rust MC
Service Management and systemd MC

The focus of this microconference will be on topics related to the current state of host-level service management and ideas for the future.

As systemd is one of the most widely adopted service managers in Linux distributions today, we expect that most of the contributions will be based around the systemd ecosystem. We’re also welcome to submissions that are not specific to systemd so we can discover and share new ideas on how to improve service management in general.

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.

I agree to abide by the anti-harassment policy

Primary authors:  ZHANG, Anita (Meta); BOCCASSI, Luca (Microsoft)

Presenters:  ZHANG, Anita (Meta); BOCCASSI, Luca (Microsoft)

Track Classification:  LPC Microconference Track (CLOSED)
#snapsafe: restoring uniqueness in Virtual Machine clones

**Monday, 12 September 2022 15:25 (40 minutes)**

**short version**

When a virtual machine gets cloned, it still contains old data that believes are unique - random number generation seeds, UUIDs, etc. Linux recently included support for VMGenID to reseed its in-kernel PRNG, but all other RNGs and UUIDs are still identical after a clone.

In this session, we will discuss approaches to solve this and reveal experiments on which we worked on, such as creating a user space readable system generation counter and going through a systemd inhibitor list for pre-snapshot/post-snapshot phases.

**long(er) version**

Linux recently added support for the Virtual Machine Generation ID (VMGenID) feature, an emulated device that informs the guest kernel about VM restore events by exposing a 128-bits UUID which changes every time a VM is restored from a snapshot. The kernel uses the UUID to reseed its PRNG, thus de-duplicating the PRNG state across VMs.

Although, VMGenID definitely works towards the correct direction, it does not provide a mechanism for notifying user-space applications of VM restore events. In this presentation, we introduce Virtual Machine Generation Counter, an extension to vmgenid which provides a low-latency and race-free mechanism for communicating restore events to user-space. Moreover, we will speak about why VM Generation Counter is not enough for ensuring across-the-stack snapshot safety. We will present an effort which builds on top of Systemd inhibitor locks to make snapshot-restore cycle a first-class citizen in the life-cycle of a system, achieving end-to-end snapshot safety.

I agree to abide by the anti-harassment policy

Yes

**Primary author:** CHALIOS, Babis (Amazon Web Services)

**Co-author:** GRAF, Alexander

**Presenter:** CHALIOS, Babis (Amazon Web Services)

**Session Classification:** Service Management and systemd MC

**Track Classification:** LPC Microconference: Service Management and systemd MC
New design for initrds

Monday, 12 September 2022 17:00 (40 minutes)

Distributions ship signed kernels, but initrds are generally built locally. Each machine gets a “unique” initrd, which means they cannot be signed by the distro, the QA process is hard, and development of features for the initrd duplicates work done elsewhere.

Systemd has gained "system extensions" (sysexts, runtime additions to the root file system), and "credentials" (secure storage of secrets bound to a TPM). Together, those features can be used to provide signed initrds built by the distro, like the kernel. Sysexts and credentials provide a mechanism for local extensibility: kernel-commandline configuration, secrets for authentication during emergency logins, additional functionality to be included in the initrd, e.g. an sshd server, other tweaks and customizations.

Mkosi-initrd is a project to build such initrds directly from distribution rpms (with support for dm-verity, signatures, sysexts). We think that such an approach will be more maintainable than the current approaches using dracut/mkinitcpio/mkinitramfs. (It also assumes we use systemd to the full extent in the initrd.)

During the talk I want to discuss how the new design works at the technical level, but also how distros can use it to provide more secure and more manageable initrds, and the security assumptions and implications.

I agree to abide by the anti-harassment policy

Yes

Primary author: JĘDRZEJEWSKI-SZMEK, Zbigniew (Red Hat)
Presenter: JĘDRZEJEWSKI-SZMEK, Zbigniew (Red Hat)
Session Classification: Service Management and systemd MC
Track Classification: LPC Microconference: Service Management and systemd MC
Slimming down the journal

Monday, 12 September 2022 16:10 (40 minutes)

In this talk, I'll discuss the new proposed compact mode for systemd-journald. Via a number of different optimizations, we can substantially reduce the disk space used by systemd-journald. I'll discuss each of the optimizations that were implemented, as well potential improvements that might further reduce disk usage but haven’t been implemented yet.

Accompanying PR: https://github.com/systemd/systemd/pull/21183

I agree to abide by the anti-harassment policy

Yes

Primary author:  DE MEYER, Daan
Presenter:  DE MEYER, Daan
Session Classification:  Service Management and systemd MC
Track Classification:  LPC Microconference: Service Management and systemd MC
systemd cgroup delegation and control processes

systemd manages the cgroup hierarchy from the root. This is considered an exclusive operation and it is sufficient when system units don’t encompass any internal cgroup structure. To facilitate arbitrary needs of units, it is possible to delegate the subtree to the unit (a necessity for such units executing as unprivileged users). However, the unified cgroup hierarchy comes with so called internal node constraint that prevents hosting processes in internal nodes of the cgroup tree (when controllers are enabled).

This creates a potential conflict between processes of the delegated unit and processes that systemd needs to run on behalf of the unit (e.g. ExecReload=). Currently, it is avoided by putting systemd control processes into an auxiliary child cgroup directly under delegated subtree root. This approach is broken when the subtree delegation is used to enable threaded cgroups since those require explicit setup and the auxiliary cgroup would miss that. Generally, this is a problem of placing the control and payload processes within the cgroup hierarchy.

I’m putting forward a few patches that allow per-unit configuration of target cgroup of control and payload processes for units that have delegated subtrees. This is a generic approach that keeps a backwards compatible default, avoids creation of unnecessary wrap cgroups and additionally allows new customization of control process execution.

It is a simple idea to present, this brings the topic up for discussion and comparison with similar situations that are affected by the internal node constraint too (e.g. joining a container) and the goal is to come up with a consent or at least the direction how to structure cgroup trees for delegated units that work well both for controller and threaded delegation.

This presentation and discussion will fit in a slot of 20 minutes.

I agree to abide by the anti-harassment policy

Yes

Primary author: KOUTNÝ, Michal
Presenter: KOUTNÝ, Michal
Session Classification: Service Management and systemd MC
Track Classification: LPC Microconference: Service Management and systemd MC
Towards Secure Unified Kernel Images for Generic Linux Distributions and Everyone Else

Monday, 12 September 2022 17:45 (40 minutes)

In this talk we’ll have a look at:

- systemd-stub (the UEFI stub for the Linux kernel shipped with systemd)
- unified kernels (i.e. kernel images glued together from systemd-stub, the kernel itself, an initrd, and more)
- systemd-sysext (an extension mechanism for initrd images and OS images)
- systemd service credentials (a secure way to pass authenticated and encrypted bits of information to services, possibly stored on untrusted media)
- systemd’s Verity support (i.e. setup logic for file system images authenticated by the kernel on IO, via dm-verity)
- systemd’s TPM2 support (i.e. ability to lock credentials or disks to TPM2 devices and software state)
- systemd’s LUKS support (i.e. ability to encrypt disks, possibly locked to TPM2)

And all that with the goal of providing a conceptual framework how to implement simple unified kernel images, that are immutable, yet extensible and parameterizable, are fully authenticated and measured, and that allow binding the root fs encryption or verity to them, in a reasonably manageable way.

The intention is to show a path for generic distributions to make use of UEFI SecureBoot and actually provide useful features for a trusted boot, putting them closer to competing OSes such as Windows, MacOS and ChromeOS, without losing too much of the generic character of the classic Linux distributions.

I agree to abide by the anti-harassment policy

Yes

Primary author:  POETTERING, Lennart
Presenter:  POETTERING, Lennart
Session Classification:  Service Management and systemd MC
Track Classification:  LPC Microconference: Service Management and systemd MC
System Boot and Security MC

In the fourth year in a row, we are going to bring together people interested in the firmware, bootloaders, system boot, security, etc., and discuss all these topics during the System Boot and Security microconference. This year we would like to focus on better communication and closer cooperation between Free Software and Open Source projects which are building blocks of Free OSes. Past months and even years showed us that the lack of one or both very often delays introduction of very interesting and important features. A good example is the TrenchBoot project. Lack of or sporadic communication and cooperation between the TrenchBoot project and Linux kernel developers made it difficult to understand each other’s requirements and needs. This has resulted in the TrenchBoot project being stuck in limbo. This situation is improving due to the interactions from last year’s System Boot and Security MC, but there are still areas for improvement. As such we think the System Boot and Security MC is a good place to discuss technical and organizational problems which may happen due to not ideal communication and cooperation. Though we do not limit this MC to this kind of problems and want to encourage all stakeholders to bring and discuss issues that they are encountering in broadly understood system boot and security.

Below is the list of topics that would be nice to cover. This is not exhaustive and can be extended if needed.

Expected topics:
- TPMs, HSMs, secure elements
  - https://trustedcomputinggroup.org/work-groups/pc-client/
- Roots of Trust: SRTM and DRTM
  - https://trustedcomputinggroup.org/resource/d-rtm-architecture-specification/
- Intel TXT, SGX, TDX
- AMD SKINIT, SEV,
- ARM DRTM
- Growing Attestation ecosystem,
- IMA,
  - https://www.redhat.com/en/blog/how-use-linux-kernels-integrity-measurement-architecture
- TrenchBoot, tboot,
  - https://github.com/TrenchBoot
  - https://sourceforge.net/projects/tboot/
- UEFI, coreboot, U-Boot, LinuxBoot, hostboot,
- Measured Boot, Verified Boot, UEFI Secure Boot, UEFI Secure Boot Advanced Targeting (SBAT),
- shim,
  - https://github.com/rhboot/shim
- boot loaders: GRUB2, SeaBIOS, network boot, PXE, iPXE,
- u-root,
- OpenBMC, u-bmc,
  - https://github.com/openbmc/openbmc
  - https://github.com/u-root/u-bmc
- legal, organizational and other similar issues relevant for people interested in system boot and security.

If you are interested in participating in this microconference and have topics...
to propose, please use the CFP process. Please note that submissions should be targeted at new developments, innovations and solving new problems on the boundary of firmware, bootloader and operating system. Be sure to explain well why and what is going to be discussed.

I agree to abide by the anti-harassment policy

**Primary authors:** KIPER, Daniel; ŻYGOWSKI, Michał (3mdeb Embedded Systems Consulting); GARRETT, Matthew (Google); KRÓL, Piotr (3mdeb Embedded Systems Consulting)

**Presenters:** KIPER, Daniel; ŻYGOWSKI, Michał (3mdeb Embedded Systems Consulting)

**Track Classification:** LPC Microconference Track (CLOSED)
Contribution ID: 271

Session Classification: System Boot and Security MC

Type: not specified
Contribution ID: 268
Type: not specified

Session Classification: System Boot and Security MC
Contribution ID: 270

Type: not specified

Session Classification: System Boot and Security MC
Kernel TEE subsystem evolution

Monday, 12 September 2022 11:00 (30 minutes)

A Trusted Execution Environment (TEE) is an isolated execution environment running alongside an operating system. It provides the capability to isolate security-critical or trusted code and corresponding resources like memory, devices, etc. This isolation is backed by hardware security features such as Arm TrustZone, AMD Secure Processor, etc.

This session will focus on the evolution of the TEE subsystem within the kernel, shared memory management between the Linux OS and the TEE, and the concept of the TEE bus. Later, we'll look at its current applications, which include firmware TPM, HWRNG, Trusted Keys, and a PKCS#11 token. Along with this, we will brainstorm on its future use-cases as a DRTM for remote attestation, among others.

I agree to abide by the anti-harassment policy

Yes

Primary author: GARG, Sumit
Presenter: GARG, Sumit
Session Classification: System Boot and Security MC
Track Classification: LPC Microconference: System Boot and Security MC
Lamenting on Secure Boot, module signatures and shims

Secure boot intend to be a way for OEMs to ensure only trusted software is capable of booting on the machines. For Linux distributions this is mostly being done by utilizing a pre-bootloader shim which is signed by a UEFI Certificate Authority, and unique for each individual Linux distribution.

One of the promises of Secure Boot was the intention of having users be able to regain control of the platform, and selectively enroll keys they would like to trust. Recently this has become problematic as module signatures are locked behind the shim, removing some certificates from the chain can prevent machines from booting and we are seeing a push away from OEMs trusting the CA at all.

This talk will give an introduction of Secure Boot is used by Linux distros and how we got here.

I agree to abide by the anti-harassment policy

Yes

Primary author:  LINDERUD, Morten (Independent)
Presenter:       LINDERUD, Morten (Independent)
Session Classification:  System Boot and Security MC
Track Classification: LPC Microconference: System Boot and Security MC
My Talk

Session Classification: System Boot and Security MC
Remote Attestation of IoT devices using a discrete TPM 2.0

Monday, 12 September 2022 11:45 (30 minutes)

There are billions of networked IoT devices and most of them are vulnerable to remote attacks. We are developing a remote attestation solution for IoT devices based on Arm called EnactTrust. The project started with PoC for a car manufacturer in 2021.

Today, we have an open-source agent at GitHub[1] that performs attestation. The EnactTrust agent leverages a discrete TPM 2.0 module and has some unique IoT features like attestation of the TPM’s GPIO for safety-critical embedded systems.

Currently, we are working on integrating our open-source agent with Arm’s open-source Trusted Firmware implementation. We are targeting both TF-A and TF-M.

Our goal is to demonstrate bootloader attestation using EnactTrust. Bootloader candidates are TrenchBoot, Tboot, and U-Boot. Especially interesting is the case of U-Boot since it does not have the same level of security capabilities as TrenchBoot and Tboot.

EnactTrust consists of an agent application (running on the device) and a connection to a private or public cloud[2]. We believe that the security of ARM-based IoT devices can be greatly improved using attestation.

[1] https://github.com/EnactTrust/enact
[2] https://a3s.enacttrust.com

I agree to abide by the anti-harassment policy

Yes

Primary authors:  Mr TOMOV, Dimitar (TPM.dev);  Mr KALCHEV, Svetlozar (EnactTrust)

Presenters:  Mr TOMOV, Dimitar (TPM.dev);  Mr KALCHEV, Svetlozar (EnactTrust)

Session Classification:  System Boot and Security MC

Track Classification:  LPC Microconference: System Boot and Security MC
Secure Boot auto enrollment

Monday, 12 September 2022 10:30 (30 minutes)

Based on a current systemd PR (https://github.com/systemd/systemd/pull/20255) that I submitted, I would like to talk about auto enrollment of Secure Boot.

I would be especially glad to have feedback on any unanticipated issues. Also it is a systemd PR, I think it fits the system boot and security micro conference as it deals with Secure Boot.

One major issue already identified is proprietary signed option ROMs and the rather low deployment of UEFI audit mode.

I agree to abide by the anti-harassment policy

Yes

Primary author: DAGONNEAU, vincent
Presenter: DAGONNEAU, vincent
Session Classification: System Boot and Security MC
Track Classification: LPC Microconference: System Boot and Security MC
Secure bootloader for Confidential Computing

Monday, 12 September 2022 10:00 (30 minutes)

Confidential computing (CC) provides a solution for data protection with hardware-based Trusted Execution Environment (TEE) such as Intel TDX, AMD SEV, or ARM RME. Today, Open Virtual machine Firmware (OVMF) and shim+grub provided necessary initialization for confidential virtual machine (VM) guest. More important, they acted as the chain of trust for measurement to support TEE attestation. In this talk, we would like to introduce the CC measurement infrastructure in the OVMF together with shim and grub, and how the VM guest uses the measurement information to support TEE runtime attestation. Finally we would like to discuss the attestation-based disk encryption solution in CC and compare the options in pre-boot phase (OVMF), OS loader phase (grub) or kernel early boot phase (initrd) and related cloud use case.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  LU, Ken (Intel);  YAO, Jiewen;  XU, min
Presenters:  LU, Ken (Intel);  YAO, Jiewen
Session Classification:  System Boot and Security MC
Track Classification:  LPC Microconference: System Boot and Security MC
Contribution ID: 273
Type: not specified

Session Classification: System Boot and Security MC
Contribution ID: 269

Session Classification: System Boot and Security MC

This

Type: not specified
Contribution ID: 272

Type: not specified

Session Classification: System Boot and Security MC
Contribution ID: 267
Type: not specified

Session Classification: System Boot and Security MC
TrenchBoot Update

Monday, 12 September 2022 12:30 (30 minutes)

Presented here will be an update on TrenchBoot development, with a focus on the Linux Secure Launch upstream activities and the building of the new late launch capability, Secure ReLaunch. The coverage of the upstream activities will focus on the redesign of the Secure Launch start up sequence to accommodate efi-stub’s requirement to control Linux setup on EFI platforms. This will include a discussion of the new Dynamic Launch Handler (dl-handler) and the corresponding Secure Launch Resource Table (SLRT). The talk will then progress into presenting the new Secure ReLaunch capability and its use cases. The conclusion will be a short roadmap discussion of what will be coming next for the launch integrity ecosystem.

I agree to abide by the anti-harassment policy

Yes

Primary authors: SMITH, Daniel (Apertus Solutions, LLC); PHILIPSON, Ross (Oracle)

Presenter: SMITH, Daniel (Apertus Solutions, LLC)

Session Classification: System Boot and Security MC

Track Classification: LPC Microconference: System Boot and Security MC
Contribution ID: 264
Type: not specified

Toolchains

I agree to abide by the anti-harassment policy
GCC’s -fanalyzer and the Linux kernel

Wednesday, 14 September 2022 15:00 (45 minutes)

I’m the author of GCC’s -fanalyzer option for static analysis.

I’ve been working on extending this option to better detect various kinds of bugs in the Linux kernel (infoleaks, use of attacker controlled values, etc).

I’ve also created antipatterns.ko, a contrived kernel module containing examples of the bugs that I think we ought to be able to detect statically.

In this session I will:

• present the current status of -fanalyzer on the Linux kernel, and
• ask a bunch of questions about how this GCC option and the kernel should best interact.

I have various ideas on ways that we can extend C via attributes, named address spaces, etc for marking up the expected behavior of kernel code in a way that I hope is relatively painless. I’m an experienced GCC developer, but a relative newcomer to the kernel, so I’m keen on having a face-to-face discussion with kernel developers and other toolchain maintainers on how such an analyzer should work, and if there are other warnings it would be useful to implement.

I agree to abide by the anti-harassment policy

Yes

Primary author: MALCOLM, David (Red Hat)
Presenter: MALCOLM, David (Red Hat)
Session Classification: Toolchains
Track Classification: Toolchains Track
The new CTF(Compact C Type Format) supported in libabigail is able to extract a corpus representation for the debug information in Kernel binary and its modules, i.e, entire Kernel release (kernel + modules). Using CTF reader improvements the time to extract and build the corpus compared with DWARF reader, for example, extracting ABI information from the Linux kernel takes up to ~4.5times less time, this was done using a Kernel compiled by GCC, nowadays LLVM doesn’t support binaries generation with CTF debug info, would be nice to have this.

But what about of the modules inserted (loaded) at runtime in the Kernel image?. To make the comparison it uses kABI scripts this is useful among other things to load modules with compatible kABI, this mechanism allows modules to be used with a different kernel version that of the kernel for which it was built. So what of using a single notion of ABI (libabigail) also for the modules loader?

Since we add support for CTF in libabigail, is needed the patch for building the Kernel with CTF enabled in the Kernel upstream configuration. Also some GCC attributes that affect the ABI and are used by kernel hackers like noreturn, interrupt, etc. are not represented in DWARF/CTF debug format and therefore they are not present in the corpus.

A stricter conformance to DWARF standards would be nice, full DWARF 5 support, getting things like ARM64 ABI extensions (e.g., for HWASAN) into things like elfutils at the same time as the compile-link toolchain, more consistency between Clang and GCC debug info for the same sources, the same for Clang and Clang with full LTO. And an extending ABI monitoring coverage beyond just architecture, symbols and types / dealing with header constants, macros and more.

The interest in discussing ways to standardize ABI and type information in a way that it can be embedded into binaries in a less ambiguous way. In other words, what can we do to not rely entirely on intermediate formats like CTF or DWARF to make sense of an ABI? Maybe CTF is already a good starting point, yet some additions are needed (e.g. other language features like for C++)?

I agree to abide by the anti-harassment policy

Yes

Primary authors: Mr SEKETELI, Dodji; Mr PROCIDA, Giuliano; Mr MARTINEZ, Guillermo E.; Mr MÄNNICH, Matthias
Presenters: Mr SEKETELI, Dodji; Mr PROCIDA, Giuliano; Mr MARTINEZ, Guillermo E.; Mr MÄNNICH, Matthias

Session Classification: Toolchains

Track Classification: Toolchains Track
Control-Flow Integrity (CFI) is a technique used to ensure that indirect branches are not diverted from a pre-defined set of valid targets, ensuring, for example, that a function pointer overwritten by an exploited memory corruption bug is used to arbitrarily redirect the control-flow of the program. The simpler way to achieve CFI is through instrumenting the binary code being executed with proper checks that verify the sanity of the indirect branches whenever they happen. To help with this goal, some CPU vendors enhanced their hardware with extensions that make these checks simpler and faster. Currently there are 4 different instrumentation setups being more broadly discussed for upstream: kCFI, which is a full software instrumentation that employs a fine-grained policy to validate indirect branch targets; ARM’s BTI, which is an ARM hardware extension that achieves CFI in a coarse-grained, more relaxed form; Intel’s IBT, which is an X86 hardware extension that similarly to BTI also achieves coarse-grained CFI, but with the benefit of also enforcing this over speculative paths; and FineIBT, which is a software/hardware hybrid technique which combines Intel’s IBT with software instrumentation to make it fine-grained without losing its good performance while still adding resiliency against speculative attacks.

In this session, kernel developers and researchers (Sami Tolvanen, Mark Rutland, Peter Zijlstra, Joao Moreira) will provide an overview on the different implementations, their upstream enablement and discuss the contrast in approaches such as granularity or implications of design differences such as callee/caller-side checks.

I agree to abide by the anti-harassment policy

Yes

**Primary authors:** MOREIRA, Joao (Intel Corporation); RUTLAND, Mark (Arm Ltd); ZIJLSTRA, Peter (Intel OTC); TOLVANEN, Sami (Google)

**Presenters:** MOREIRA, Joao (Intel Corporation); RUTLAND, Mark (Arm Ltd); ZIJLSTRA, Peter (Intel OTC); TOLVANEN, Sami (Google)

**Session Classification:** Toolchains

**Track Classification:** Toolchains Track
Programmable debuggers and the Linux kernel
(drgn, GDB+poke)

Wednesday, 14 September 2022 12:00 (45 minutes)

This activity is about programmable debuggers and their usage in the Linux kernel. By “programmable debugger” we understand debuggers that are able to understand the data structures handled by the target program, and to operate on them guided by user-provided scripts or programs.

First we will be doing a very brief presentation of two of these debuggers: drgn and GDB+poke, highlighting what these tools provide on top of the more traditional debuggers.

Then we will discuss how these tools (and the new style of debugging they introduce) can be successfully be used to debug the Linux kernel. The main goal of the discussion is to collect useful feedback from the kernel hackers, with the goal of making the tools as useful as possible for real needs; for example, we would be very interested in figuring out what are the kernel areas/structures/abstractions for which support in the tools would be most useful.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  ZANNONI, Elena; MARCHESI, Jose E. (GNU Project, Oracle Inc.); BRENNAN, Stephen (Oracle)

Presenters:  ZANNONI, Elena; MARCHESI, Jose E. (GNU Project, Oracle Inc.); BRENNAN, Stephen (Oracle)

Session Classification:  Toolchains

Track Classification:  Toolchains Track
Status Report: Broken Dependency Orderings in the Linux Kernel

Wednesday, 14 September 2022 10:45 (45 minutes)

Potentially broken dependency orderings in the Linux kernel have been a recurring theme on the Linux kernel mailing list and even Linux Plumbers Conference. The Linux kernel community fears that with ever-more sophisticated compiler optimizations, it would become possible for modern compilers to undermine the Linux kernel memory consistency model when optimizing code for weakly-ordered architectures, e.g. ARM or POWER.

Specifically, the community was worried about address and control dependencies being broken, with the latter having already seen several unfruitful [PATCH RFC]’s on LKML.

This “fear” of optimizing compilers eventually lead to READ_ONCE() accesses being promoted to memory-order-acquire semantics for arm64 kernel builds with link-time optimization (LTO) enabled, leaving valuable performance improvements on the table as this imposes ordering constraints on non-dependent instructions.

However, the severity of this problem had not been investigated as of yet, with previous discussions lacking the evidence of concrete instances of broken dependency orderings.

We are pleased (or not so pleased) to report that, based on our work, we have indeed found broken dependency orderings in the Linux kernel. We would now like to open the discussion about, but not limited to, the severity of the broken dependencies we found thus far, whether they warrant dedicating more attention to this problem, and potential (lightweight or heavyweight) fixes.

I agree to abide by the anti-harassment policy

Yes

Primary authors: MAINAS, Charalampos (Technical University of Munich); ELVER, Marco (Google); FINK, Martin (Technical University of Munich); HEIDEKRÜGER, Paul (Technical University of Munich); Prof. BHATOTIA, Pramod (Technical University of Munich); Dr CHAKRABORTY, Soham (Delft University of Technology)

Presenters: ELVER, Marco (Google); HEIDEKRÜGER, Paul (Technical University of Munich)

Session Classification: Toolchains

Track Classification: Toolchains Track
Toolchain support for objtool in the Linux kernel

Wednesday, 14 September 2022 12:45 (45 minutes)

The Linux kernel relies on objtool for performing a host of validations, metadata generation, and other fixups and annotations. One of Objtool’s feature is stack metadata validation and generation which forms the backbone of kernel’s reliable stack unwinding needs.

In this session, we will discuss what components of the objtool, in general, can get some help from the toolchain. We will also discuss what assistance can be provided for the usecase of reliable stack tracing. For reliable stack traces, correct and complete metadata is only one of the pillars. We will discuss the additional components that are required. At LPC 2021, we talked about the proposal to define and generate CTF Frame unwind information in the GNU Toolchain. There was also a discussion on objtool on arm64. In this session, we plan to converge these discussions with a perspective of what toolchain support can be provided to support objtool in the Linux kernel.

I agree to abide by the anti-harassment policy

Yes

Primary authors: BHAGAT, Indu; POIMBOEUF, Josh (Red Hat)
Presenters: BHAGAT, Indu; POIMBOEUF, Josh (Red Hat)
Session Classification: Toolchains

Track Classification: Toolchains Track
Welcome to the toolchain track from the organizers.

Presenters:  MARCHESI, Jose E. (GNU Project, Oracle Inc.); DESAULNIERS, Nick (Google)
Session Classification:  Toolchains
Where are we on security features?

*Wednesday, 14 September 2022 10:00 (45 minutes)*

There has been tons of work across both GCC and Clang to provide the Linux kernel with a variety of security features. Let’s review and discuss where we are with parity between toolchains, approaches to solving open problems, and exploring new features.

Parity reached since last year:

- zero call-used registers
- structure layout randomization

Needs work:

- stack protector guard location
- Link Time Optimization
- forward edge CFI
- backward edge CFI
- array bounds checking
- -fstrict-flex-arrays
- __builtin_dynamic_object_size
- C language extension for bounded flexible arrays
- builtin for answering “does this object end with a flexible array?”
- -fsanitize=bounds
- integer overflow protection
- Spectre v1 mitigation

I agree to abide by the anti-harassment policy

Yes

**Primary authors:** COOK, Kees (Google); ZHAO, Qing

**Presenters:** COOK, Kees (Google); ZHAO, Qing

**Session Classification:** Toolchains

**Track Classification:** Toolchains Track
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, virtualization, 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 micro-conference focuses on the kernel code that enables these new system features that often require coordination between the VFIO, IOMMU and PCI sub-systems.

Following up the successful LPC 2017, 2019, 2020 and 2021 VFIO/IOMMU/PCI micro-conference, the Linux Plumbers Conference 2022 VFIO/IOMMU/PCI track will therefore focus on promoting discussions on the current kernel patches aimed at VFIO/IOMMU/PCI sub-systems with specific sessions targeting discussion for the kernel patches that enable technology (e.g., device/sub-device assignment, PCI core, IOMMU virtualization, VFIO updates, DMA ownership models, Trusted Computing, etc.) requiring the three sub-systems coordination. The micro-conference will also cover VFIO/IOMMU/PCI sub-system specific tracks to debate the status of patches for the respective sub-systems.

See the following video recordings from LPC 2019, 2021 and 2022 VFIO/IOMMU/PCI micro-conference:
And the archived LPC 2017 VFIO/IOMMU/PCI micro-conference web page at Linux Plumbers Conference 2017, where the audio recordings from the micro-conference 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 years successful Linux Plumbers micro-conference:

- Support for the /dev/iommufd device has been discussed and then later refined moving it closer to the final design before implementation work would start
- The groundwork for refactoring the Shared Virtual Address (SVA) and I/O Page Fault (IOPF) support in IOMMU has been laid out readying it for future inclusion in the mainline kernel
- The SVA for in-kernel users support has taken a completely different direction following a discussion that was held concerning the proposed implementation (for instance, the KVA approach has since been retired). A new effort is to converge on the DMA API offering support through a set of possible extensions to it, however, the work is still ongoing and the final solution is yet to be decided upon
- A framework for managing group DMA ownership and removal of the BUG_ON in VFIO is now ready to be merged into the mainline kernel
- A series of enhancements for extending IOMMU domain to support more I/O Page Table types have been included in the mainline kernel
- The work to bring support for the Compute Express Link (CXL) in the Linux kernel is ongoing and it has been widely reviewed and debated, especially concerning Data Object Exchange (DOE), Component Measurement and Authentication (CMA) and Security Protocol and Data Model (SPDM) support is under heavy development. A lot of work has been put into resolving and addressing issues around the implementation, however, it will take many kernel releases before the CXL support is refined to the point where it would be considered stable to use, nonetheless good progress has been achieved thus far

Tentative topics that are under consideration for this year include (but 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 Recover (EDR)
  - Power management and devices supporting Active-state Power Management (ASPM)
  - Peer-to-Peer DMA (P2PDM)
  - 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 virtualization
  – 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 device-driver core, etc.)

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

Come and join us in the discussion in helping Linux keep up with the new features being 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
• Lorenzo Pieralisi
• Lu Baolu
• Marc Zyngier
• Pali Rohár
• Peter Zijlstra
• Thomas Gleixner

Contacts:
• Alex Williamson (alex.williamson@redhat.com)
• Bjorn Helgaas (bhelgaas@google.com)
• Jörg Roedel (jroedel@suse.de)
• Lorenzo Peralisi (lorenzo.peralisi@arm.com)
• Krzysztof Wilczyński (kw@linux.com)

Primary authors: WILLIAMSON, Alex (Red Hat); HELGAAS, Bjorn (Google); ROEDEL, Joerg (SUSE); WILCZYŃSKI, Krzysztof; PIERALISI, Lorenzo (Arm)

Presenters: WILLIAMSON, Alex (Red Hat); HELGAAS, Bjorn (Google); ROEDEL, Joerg (SUSE); WILCZYŃSKI, Krzysztof; PIERALISI, Lorenzo (Arm)

Track Classification: LPC Microconference Track (CLOSED)
Cooperative DMA in a memory-oversubscribed environment

Monday, 12 September 2022 10:30 (30 minutes)

Running virtual machines with memory subscription and DMA device passthrough is a challenge:
1. If devices/IOMMUs don’t support faults or ATS, the hypervisor can’t know which pages to map to ensure that DMA will not fault.
2. VFIO pins all memory when the memory range is mapped for DMA; this makes overcommit a challenge!

We describe a solution to both of these problems:
- support VFIO DMA (re)mapping: when a page is reclaimed via madvise or swap, remove it from IOMMU page table mappings; when a page is faulted in add it to IOMMU mappings. Similar to how KVM page tables are kept in sync with userspace page tables
- provide an light-weight enlightenment to virtual machine kernels which can cooperate with the hypervisor to ensure that pages mapped for DMA are resident

The overview of this solution is presented, and some open questions are posed for consideration by the audience:
- how to connect IOMMU page tables to userspace page tables? Callbacks?
- how to expose the DMA cooperative device to the guest virtual machine (or process)

Finally discussion about next steps and a path to upstreaming is discussed.

I agree to abide by the anti-harassment policy

Yes

Primary author:  GOWANS, James (Amazon EC2)
Presenter:  GOWANS, James (Amazon EC2)
Session Classification:  VFIO/IOMMU/PCI MC
Track Classification:  LPC Microconference: VFIO/IOMMU/PCI MC
Exposing PCIe topology to Guest OS for peer-to-peer

Monday, 12 September 2022 12:30 (30 minutes)

Doing peer-to-peer (aka p2p) is becoming more common these days. Whether it is done between GPUs and RDMA NICs, or between AI accelerators and NVME devices, doing p2p can decrease the CPU load, increase the b/w and improve the latency of data movement.

When implementing p2p code, whether it is using the p2pdma infrastructure or dma-buf framework, the kernel code eventually needs to calculate the PCI distance between the peers in order to validate whether they can perform p2p.

This is done today by calling pci_p2pdma_distance_many() which validates that either all the peer devices are behind the same PCI root port or the host bridges connected to each of the devices are listed in the 'pci_p2pdma_whitelist'.

The problem is that this function is not able to calculate the distance when working inside a Virtual Machine because the PCIe topology is not exposed to the Guest OS. Also the host bridges are not exposed, so even if they are whitelisted, the Guest OS wouldn’t know that.

I would like to brainstorm on how to solve this problem because today the only way to do p2p inside a VM is to run a modified kernel which bypass this function.

I agree to abide by the anti-harassment policy

Yes

Primary author: GABBAY, Oded
Presenter: GABBAY, Oded
Session Classification: VFIO/IOMMU/PCI MC
Track Classification: LPC Microconference: VFIO/IOMMU/PCI MC
Instant Detection of Virtual Devices

Monday, 12 September 2022 11:45 (30 minutes)

During boot-time of Guest there are many (in thousands) PCI config reads and significantly increases Guest boot-time.

Currently, when these reads are performed by a Guest, they all cause a VM-exit, and therefore each one of them induces a considerable overhead.

This overhead can be further improved, by mapping MMIO region of virtual machine to memory area that holds the values that the “emulated hardware” is supposed to return. The memory region is mapped as “read-only” in the NPT/EPT, so reads from these regions would be treated as regular memory reads. Writes would still be trapped and emulated by the hypervisor.

This helps to reduce virtual machine PCI scan and initialization time by ~65%. In our case it reduced to ~18 mSec from ~55 mSec.

I agree to abide by the anti-harassment policy

Yes

Primary author: KAHER, Ajay
Presenter: KAHER, Ajay
Session Classification: VFIO/IOMMU/PCI MC
Track Classification: LPC Microconference: VFIO/IOMMU/PCI MC
Integrated PCIe monitoring and tracing facilities

Monday, 12 September 2022 13:00 (30 minutes)

PMU for monitoring PCIe link events
- purpose and introduction
- Usage and event supported

PTT for tracing and tuning PCIe link
- purpose and introduction
- Usage of tracing and capabilities
- Usage of tuning and event supported

Potential Scenarios
- performance improvement with PMU + tune
- tracing for validating and monitoring, more convenient compared to PCIe analyzer
- etc.

More on monitoring and tracing?
- need feedbacks for the design, usage and future plan. What we want and what’ll be more helpful?
- possible for more platforms and devices, like PMU/PTT for a switch?
- besides tracing TLP, possible for DLLP etc?
- more trace/monitor facilities and events for PCIe stuff?
- possible for standardization, either specification or software framework
- etc.

I agree to abide by the anti-harassment policy
Yes

Primary author:  YANG, Yicong
Presenter:  YANG, Yicong
Session Classification:  VFIO/IOMMU/PCI MC
Track Classification:  LPC Microconference: VFIO/IOMMU/PCI MC
IOMMUFD Discussion

Monday, 12 September 2022 10:00 (30 minutes)

Session to focus on open items related to iommufd and its path to upstream.
A short overview to ground the discussion in the current state of affairs followed by a discussion
on any open points related to its design an implementation and to conclude what should be in the
first merged series.

Should iommufd progress to be merged before the conference then this session would focus on the
large list of advanced features that are expected to ride on top of it and the discussion can focus
on the uAPI design of those.

IOMMUFD is a new user API for IOMMU that replaces vfio-type1 and allows userspace to create
iommu_domains and place pages into them. IOMMUFD is designed to overcome several limita-
tions of type 1 and support advanced virtualization focused IOMMU features.

I agree to abide by the anti-harassment policy

Yes

Primary authors:  LU, Baolu;  Mr GUNTHORPE, Jason (NVIDIA Networking);  AUGER, Eric (Red
Hat);  TIAN, Kevin (Intel);  LIU, Yi

Presenter:  Mr GUNTHORPE, Jason (NVIDIA Networking)

Session Classification:  VFIO/IOMMU/PCI MC

Track Classification:  LPC Microconference: VFIO/IOMMU/PCI MC
PCIe Endpoint Subsystem Open Items Discussion

Monday, 12 September 2022 11:15 (30 minutes)

PCIe Endpoint Framework is a relatively new framework added to Linux Kernel. There are two upstreamed generic function drivers added; one for PCIe Endpoint Test (simple test function to test the communication between root-complex and endpoint) and the other for providing NTB functionality to the host (two endpoints within SoC facilitate two hosts to communicate).

A new endpoint function was added that kind-of hacks around to use NTB framework for PCIe RC to EP communication. While this provided network interface between RC and EP, is this the right way to do? Should we build on PCIe VirtIO standard or allow ad-hoc drivers?

Other Open Items to discuss:
1) PCIe Endpoint Notifier
2) Moving to genalloc for outbound window memory allocation
3) Interrupts handling in EP mode
4) Device Tree Integration

I agree to abide by the anti-harassment policy

Yes

Primary authors: I, KISHON VIJAY ABRAHAM; Mr SADHASIVAM, Manivannan (Mani)

Presenters: I, KISHON VIJAY ABRAHAM; Mr SADHASIVAM, Manivannan (Mani)

Session Classification: VFIO/IOMMU/PCI MC

Track Classification: LPC Microconference: VFIO/IOMMU/PCI MC
Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC

The Zoned Storage interface has been introduced to make more efficient use of the storage medium, improving both device raw capacity and performance. Initially implemented for Shingled magnetic recording (SMR) HDDs, and recently also for flash-based SSDs through Zoned Namespace (ZNS) SSDs. Zoned storage devices expose their storage through zone semantics. Each zone has a set of read/write rules associated. One example is requiring zones to be written sequentially, which requires the host to be aware of such requirements.

The Linux kernel has had a general zoned subsystem for SMR HDDs since kernel 4.10 and gained support for ZNS SSDs in kernel 5.9. Furthermore, to help faster adoption of Zoned Storage devices, a few parts of the storage stack has been extended with zone support. Examples are filesystems such btrfs and f2fs, and the device mapper targets dm-linear and dm-zoned. This all eases the barrier to adoption.

The Zoned Storage micro-conference is to communicate the benefits of Zoned Storage to a broader audience, present the current and future challenges within the Linux storage stack, and collaborate with the wider community.

I agree to abide by the anti-harassment policy

Primary authors: MANZANARES, Adam (Samsung); CHAMBERLAIN, Luis (Samsung); BJØRLING, Matias (Western Digital)

Presenters: MANZANARES, Adam (Samsung); CHAMBERLAIN, Luis (Samsung); REINECKE, Hannes (SUSE Labs); BJØRLING, Matias (Western Digital)

Track Classification: LPC Microconference Track (CLOSED)
A zone-aware cache system for distributed databases

Wednesday, 14 September 2022 10:05 (20 minutes)

Modern analytical distributed database platform requires massive data from remote filesystems (e.g. HDFS). A cache layer is necessary to eliminate the network bottleneck by caching data items with smaller granularity (e.g. 64KB ~ 128KB).

There are three major challenges to implementing such a cache system:
1. Predictable latency (latency spike is not acceptable)
2. Good enough user-side throughput (low end-to-end write amplification)
3. High-density storage per server with reasonable cost (SSD Cache is required)

Our previous solution is to use a traditional storage engine TerarkDB (which is a fork of RocksDB with much better throughput, and with a lot of optimization) on the EXT4 filesystem. But the result still can’t meet our expectations:
1. A lot of latency spikes (we don’t want to magically tuning it again and again under different workloads)
2. Too much write amplification so we cannot get enough write throughput.
2. Cannot provide predictable space cost (space amplification) and cannot make use of QLC SSD (random write), thus high-density storage cost is not acceptable

To solve these problems, we re-designed our Cache system by following the ZNS principles:
1. In-memory metadata and record-level indexing (thanks to large item size), so we have no read amplification.
2. Append only IO model with limited active write point, so we can make use of ZNS devices
3. User-controlled GC, so we would be able to use almost all space of the disk (a few reserved zones for data migration is enough), this is not possible on lsm-tree and conventional drives
4. Emergency data sacrifice (a cache system can usually tolerance some data loss), so we can make sure the device space is always fully utilized

Under benchmarks, we’ve got: 1) Much lower storage cost (QLC SSD & fully utilize disk space); 2) Stable latency (user-controlled GC & record-level indexing); 3) 5X+ better write throughput (append-only IO model);

Further works: we still haven’t tested it under ZNS QLC SSD yet but expect to have a stable performance.

I agree to abide by the anti-harassment policy

Yes

Primary author: GUO, Kuankuan
Presenter: GUO, Kuankuan
Session Classification: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Track Classification: LPC Microconference: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Btrfs RAID on zoned devices

Currently there is no possibility to use btrfs' builtin RAID feature with zoned block-devices, for a variety of reasons.

This talk gives a status update on my work on this subject's matter and possibly a roadmap for further development and research activities.

I agree to abide by the anti-harassment policy

Yes

Primary author: THUMSHIRN, Johannes (Western Digital Corporate)
Presenter: THUMSHIRN, Johannes (Western Digital Corporate)
Session Classification: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Track Classification: LPC Microconference: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Experiences implementing zonefs support in ZenFS

Wednesday, 14 September 2022 13:00 (25 minutes)

The talk will cover the main challenges in porting an zoned block device aware application using raw block device access (ZenFS using libzbd) to zonefs. In addition to this, a performance comparison between ZenFS using zbdlib and zonefs will be presented.

I agree to abide by the anti-harassment policy

Yes

Primary author: HANSEN, Jorgen (WDC)
Presenter: HANSEN, Jorgen (WDC)
Session Classification: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Track Classification: LPC Microconference: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Improving data placement for Zoned Linux File systems

*Wednesday, 14 September 2022 10:45 (20 minutes)*

In this talk I’ll present what I’ve learned from building ZenFS, a user-space RocksDB file system for zoned block devices, and what features could be transferable to kernel file systems.

I’ll go over the goals and high-level design of zenfs, focusing on the extent allocator, present what performance gains we’ve measured and what the trade-offs are when constructing a file system for zoned block devices.

Finishing up, I’d like to open a discussion on how to enable similar levels of performance in posix-compliant, general purpose file systems with zone support. BTRFS would be a good first target but bcachefs could also benefit from this.

Unless we do data separation (separating files into different zones) we will not reap the full benefits of zoned storage.

1. [https://github.com/westerndigitalcorporation/zenfs/](https://github.com/westerndigitalcorporation/zenfs/)
2. [https://www.usenix.org/conference/atc21/presentation/bjorling](https://www.usenix.org/conference/atc21/presentation/bjorling)

I agree to abide by the anti-harassment policy

Yes

**Primary author:**  HOLMBERG, Hans

**Presenter:**  HOLMBERG, Hans

**Session Classification:**  Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC

**Track Classification:**  LPC Microconference: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Object caching on Zoned Storage

Object caching is a great use case for SSDs but it comes with a big device write amplification penalty—often as much as 50%-- of the capacity is reserved for over-provisioning to reduce the impact of this on conventional SSDs.

There is a great opportunity to address this problem using zoned storage, as the garbage collection can be co-designed with the cache eviction policy.

Objects stored in flash caches have a limited life time and a common approach to eviction is to simply throw out the oldest objects in the cache to free space. Conventional drives have no notion of how old objects are and are not allowed to just throw out data out of erase units on the drive to reclaim space. If the garbage collection of the drive data is done cooperatively with a ZNS cache FTL on the host however, objects can be chosen to be evicted in stead of relocated when space is reclaimed.

To get there, we will need an ZNS cache FTL and an interface between the FTL and the cache implementation for indicating which LBAS that should be re-located or invalidated to minimize write amplification.

How could this be implemented? What options do we have? A ZNS Cache userspace library? A cache block device?

The user/integration point of this I have in mind would be CacheLib[^2], what other potential users do we have?

This is a great opportunity to work together on a common solution for several use cases and vendors, pushing the eco-system forward!

[^2]: https://github.com/facebook/CacheLib

I agree to abide by the anti-harassment policy

Yes

Primary author: HOLMBERG, Hans
Presenter: HOLMBERG, Hans
Session Classification: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Track Classification: LPC Microconference: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
SSDFS: ZNS SSD ready file system with zero GC overhead

Wednesday, 14 September 2022 10:25 (20 minutes)

The architecture of SSDFS is the LFS file system that can: (1) exclude the GC overhead, (2) prolong NAND flash device's lifetime, (3) achieve a good performance balance even if the NAND flash device's lifetime is a priority. The fundamental concepts of SSDFS: (1) logical segment, (2) migration scheme, (3) background migration stimulation, (4) diff-on-write. Every logical block is described by \{segment_id, block_index_inside_segment, length\}. This concept completely excludes block mapping metadata structure updates that results in decreasing the write amplification factor. Migration scheme implies that after erase block exhaustion every update of logical block results in storing new state in the destination erase block and invalidation of logical block in the exhausted one. Regular I/O operations are capable to completely invalidate the exhausted erase block for the case of “hot” data (no necessity in GC operations). SSDFS is using the migration stimulation technique as complementary to migration scheme. It means that if some LEB is under migration then a flush thread is checking the opportunity to add some additional content into the log under commit. SSDFS is using the inline techniques to combine metadata/data pieces into one I/O request of decreasing write amplification factor. SSDFS architecture is ZNS SSD friendly and it can run efficiently even with limited number of active/open zones (14 active zones, for example). Preliminary benchmarking and estimations of conventional SSDs has showed the ability of SSDFS to decrease write amplification 2x - 10x times and prolong SSD lifetime 2x - 10x for real life use-cases comparing with other file systems (ext4, xfs, btrfs, f2fs, nilfs2).

I agree to abide by the anti-harassment policy

Yes

Primary author:  DUBEYKO, Viacheslav (ByteDance)

Presenter:  DUBEYKO, Viacheslav (ByteDance)

Session Classification:  Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC

Track Classification:  LPC Microconference: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Supporting non-power of 2 zoned devices

Wednesday, 14 September 2022 11:45 (25 minutes)

The zone storage implementation in Linux, introduced in v4.10, first targeted SMR drives with a power of 2 (po2) zone size alignment requirement. The newer NAND-based zoned storage devices do not naturally align to po2, so the po2 requirement introduces a gap in each zone between its actual capacity and size. This talk explores the various efforts that have been going on to allow non-power of 2 (npo2) zoned devices so that LBA gaps in each zone can be avoided. The main goal of this talk is to raise community awareness and get feedback from current/future adopters of zoned storage devices.

[https://lore.kernel.org/linux-block/20220615101920.329421-1-p.raghav@samsung.com/](https://lore.kernel.org/linux-block/20220615101920.329421-1-p.raghav@samsung.com/)

I agree to abide by the anti-harassment policy

Yes

**Primary authors:** Mr GONZALEZ, Javier (Samsung); CHAMBERLAIN, Luis (Samsung); RAGHAV, Pankaj

**Presenter:** RAGHAV, Pankaj

**Session Classification:** Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC

**Track Classification:** LPC Microconference: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Zoned MC Conference Kick Off

Wednesday, 14 September 2022 10:00 (5 minutes)

Welcome and kick off presented by Adam and Matias!

**Presenters:**  MANZANARES, Adam (Samsung Electronics); BJØRLING, Matias (Western Digital)

**Session Classification:**  Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
This presentation will discuss planned new features and improvements for the zonefs file system: asynchronous zone append IOs, relaxing of O_DIRECT write constraint and memory consumption reduction. Feedback from the audience will also be welcome to discuss other ideas and performance enhancements.

I agree to abide by the anti-harassment policy

Yes

Primary author: LE MOAL, Damien (Western Digital)
Presenter: LE MOAL, Damien (Western Digital)
Session Classification: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC
Track Classification: LPC Microconference: Zoned Storage Devices (SMR HDDs & ZNS SSDs) MC