Linux Plumbers Conference 2025

Contribution ID: **24**

Type: **not specified**

Linux System Monitoring and Observability MC

CFP ends on October 10th (CLOSED)

The Linux System Monitoring and Observability Track brings together developers, maintainers, system engineers, and researchers focused on understanding, monitoring, and maintaining the health of Linux systems at scale. This track addresses the needs of engineers managing millions of Linux servers, where proactive monitoring, rapid problem detection, and automated remediation are essential for operational success.

Track Objectives

This track aims to foster collaboration between kernel developers, system administrators, and monitoring tool creators to advance the state of Linux system monitoring and observability.

Target Audience

- Hyperscaler Engineers: System reliability engineers managing large-scale deployments
- Kernel Developers: Contributors working on tracing, performance counters, and diagnostic interfaces
- Monitoring Tool Developers: Creators of observability platforms and diagnostic utilities
- System Administrators: Operations teams responsible for fleet health and incident response
- Performance Engineers: Specialists focused on system optimization and bottleneck identification

Key attendees:

- Usama Arif
- Guilherme Piccolli
- Ricardo Canuelo
- Gavin Guo
- Systemd folks (Daan)
- Other cloud sysadmins

Key Focus Areas

Kernel Health and Runtime Monitoring
- Real-time kernel health assessment techniques
- Early detection of system degradation

Hardware Integration and Error Detection
- Hardware error correlation and root cause analysis
- Integration between kernel monitoring and hardware telemetry

Problem Correlation
- Virtualization stack monitoring (hypervisor ↔ guest relationships)
- Container runtime observability
- Network stack performance and reliability monitoring
- Storage I/O path analysis and optimization
- BMC information
- Scheduler (sched_ext changes)

Memory Management and Analysis
- Runtime memory profiling techniques
- Out-of-memory (OOM) prediction and prevention
- Memory leak detection and mitigation

Automated Analysis and Remediation
- Automated problem categorization and triage
- Anomaly detection algorithms for system behavior

Visualization and Alerting Infrastructure
- Real-time dashboarding for large-scale deployments
- Historical trend analysis and capacity planning

Tools and frameworks that would fit here
- eBPF/BPF: Advanced tracing and monitoring programs
- ftrace/perf: Low-level kernel tracing infrastructure
- Runtime Sanitizers: KFENCE, KASAN, and similar detection tools
- Hardware Interfaces: EDAC, MCE, ACPI error reporting

System-Level Tools
* bpftrace: Dynamic tracing language and runtime
* systemd: Service monitoring and system state management
* netconsole: Network-based kernel logging

Crash Analysis and Post-Mortem
* kdump/crash/drgn: Kernel crash dump analysis
* Core dump analysis: Userspace failure investigation
* Live debugging: GDB integration and kernel debugging techniques

Performance Analysis
* perf: Hardware performance counter analysis
* Memory profilers: Heap analysis and memory usage optimization
* Runtime memory profilers, such as below, strobelight, Open Telemetry, etc

Example Topics and Presentations. Previous LPC Presentations That Align With This Track

- Livepatch Visibility at Scale
- IOMMU overhead optimizations and observability
- PWRu: BPF-based Datacenter Telemetry with Kernel Bypass
- Performance Analysis Superpowers with Enhanced BPF
- Runtime Verification: Where We've Been, Where We're Going
- Linux Performance Analysis and Observability

Primary author: LEITAO, Breno (Meta)

Co-author: ARIF, Usama

Presenters: LEITAO, Breno (Meta); PICCOLI, Guilherme (Igalia); XING, Jason (Tencent); ARIF, Usama