LTTng and Related Projects Update

DORSAL Progress Meeting May 2025





Outline

- Last Six Months
- Upcoming Work



In the Last six months





Babeltrace 2.1 – Moving towards CTF 2

Babeltrace 2.1 – Add reading and producing CTF 2 traces

• Released! Q1 2025

LTTng 2.15 – Producing CTF 2 traces

• Release: Q3 2025



Why CTF 2?

Trace metadata expressed in JSON – rather than a custom language

You can add human-readable descriptions – rather than relying on the formal labels



Tackling our memory footprint

Important Problems (2024)

High memory usage of tracing buffers

- Buffers currently allocated per CPU and per container
- Quickly not viable in a typical containerized environment

Wasting time processing incomplete traces

- Know as early as possible if the information you want was not captured, so you can adjust and retry
- Maximize detail without overwhelming the system

Important Problems (2024)

High memory usage of tracing buffers

- Buffers currently allocated per CPU and per container
- Quickly not viable in a typical containerized environment

Wasting time processing incomplete traces

- Know as early as possible if the information you want was not captured, so you can adjust and retry
- Maximize detail without overwhelming the system

One shared buffer!

Feature: Per-channel buffers

Advantages

- Memory footprint does not scale with number of CPUs
- More resistant to variability in trace production
- More intuitive traces

Disadvantage

More contention when writing trace data to buffers



Try LTTng 2.14

Release candidate available since last month

• Official release – In the coming weeks!

What's new?

- Per-channel buffers
- Producing CTF 2 traces (Experimental)
- Log4j2 support



Linux Kernel & Community Work

Linux Kernel & Community Work

Laying the foundation to improve scalability...

- Allow Linux users to express CPU resource usage limits in a cloudnative way within containers
- Allow Linux application developers to benefit from per-CPU data structures
- Improve scalability and performance of Userspace RCU end users
 e.g. BIND 9 name server
- Improve scalability of counters on large many-core systems: provide fast sum approximation



Linux Kernel & Community Work

Laying the foundation to improve instrumentation...

- Minimize the overhead of dormant static instrumentation
- Increase flexibility of static instrumentation
- Dynamic instrumentation



User Interviews

User Interviews

This year we've spoken to 8 people so far...

- 5 tracing end users Developers and troubleshooters
- 3 tracing-related folks Tool dev and tracing researchers

Total of 18 people interviewed about troubleshooting!



Important Problems

Risk of causing performance issues

- Tracing large amounts can impact performance
- Concern especially in live networks

Does the trace contain the data relevant to the issue?

• Know as early as possible if the information needed was not captured, so you can adjust and retry



Upcoming* Work

User Testing

If you...

- Are interested in **reducing tracing performance impact**
- Use LTTng and want to give feedback

Reach out to us!

ebugden@efficios.com





LTTng 2.15 – 2025

- Further improve memory footprint (with sparse files)
- Produce CTF 2 traces

LTTng 2.16 – 2026

- Add Aggregation Maps (with Trace Hit Counters)
- Improve tracing buffer robustness



Tracing Summit 2025



What? – Single-track conference about software and hardware tracing.

When? – Sept. 2 & 3, 2025. In Montreal and remote.

For more information and talk submissions: https://tracingsummit.org/







Questions?

- Links:
 - <u>https://www.efficios.com</u>
 - <u>https://lttng.org</u>
 - <u>https://babeltrace.org</u>
 - <u>https://diamon.org</u>
 - <u>https://barectf.org</u>





Annex



• Common Trace Format 2 Specification

https://diamon.org/ctf

• libside repository

https://github.com/efficios/libside



Presentations

- FOSDEM 2025
 - Container Devroom:
 - A new cgroup cpu.max.concurrency controller interface file
 - <u>https://fosdem.org/2025/schedule/event/fosdem-2025-6283-a-new-cgroup-cpu-max-concurrency-controller-interface-file/</u>
 - Kernel Devroom:
 - Waste-Free Per-CPU Userspace Memory Allocation
 - <u>https://fosdem.org/2025/schedule/event/fosdem-2025-6245-waste-free-per-cpu-userspace-memory-allocation/</u>
 - HPC, Big Data, and Data Science Devroom
 - Exa-Tracer: Tracing HPC Supercomputers with LTTng
 - https://fosdem.org/2025/schedule/event/fosdem-2025-6269-exa-tracer-tracing-hpc-supercomputers-with-lttng/
 - DNS Devroom
 - How to make BIND 9 fast(er)
 - Presenter: Ondřej Surý
 - https://fosdem.org/2025/schedule/event/fosdem-2025-4626-how-to-make-bind-9-fast-er-/
 - (Userspace RCU end user)
- 2 presentations at LSFMM+BPF 2025 conference
 - Allocation of Per-CPU Memory in Userspace
 - <u>https://lore.kernel.org/all/e5bb2190-2688-4491-9d17-574ff925f239@efficios.com/</u>
 - Improve KSM for code patching use-cases
 - https://lore.kernel.org/all/427ca05b-fe39-4365-83c6-a23a0733ee69@efficios.com/



SIDE ABI RFC (libside)

- The SIDE ABI is currently at RFC stage, aiming to create a specification.
 - <u>https://github.com/efficios/libside/blob/master/doc/rfc-side-abi.txt</u>
- Runtime/language agnostic,
- Supports multiple concurrent tracers,
- Instrumentation is not specific to a tracer,
 - No need to rebuild applications if using a different tracer,
- Instrumentation can be either static or dynamic,
- Supports complex/nested types,
- Supports both static and dynamic types,
- Libside is a C/C++ reference implementation for the System V ELF ABI.

