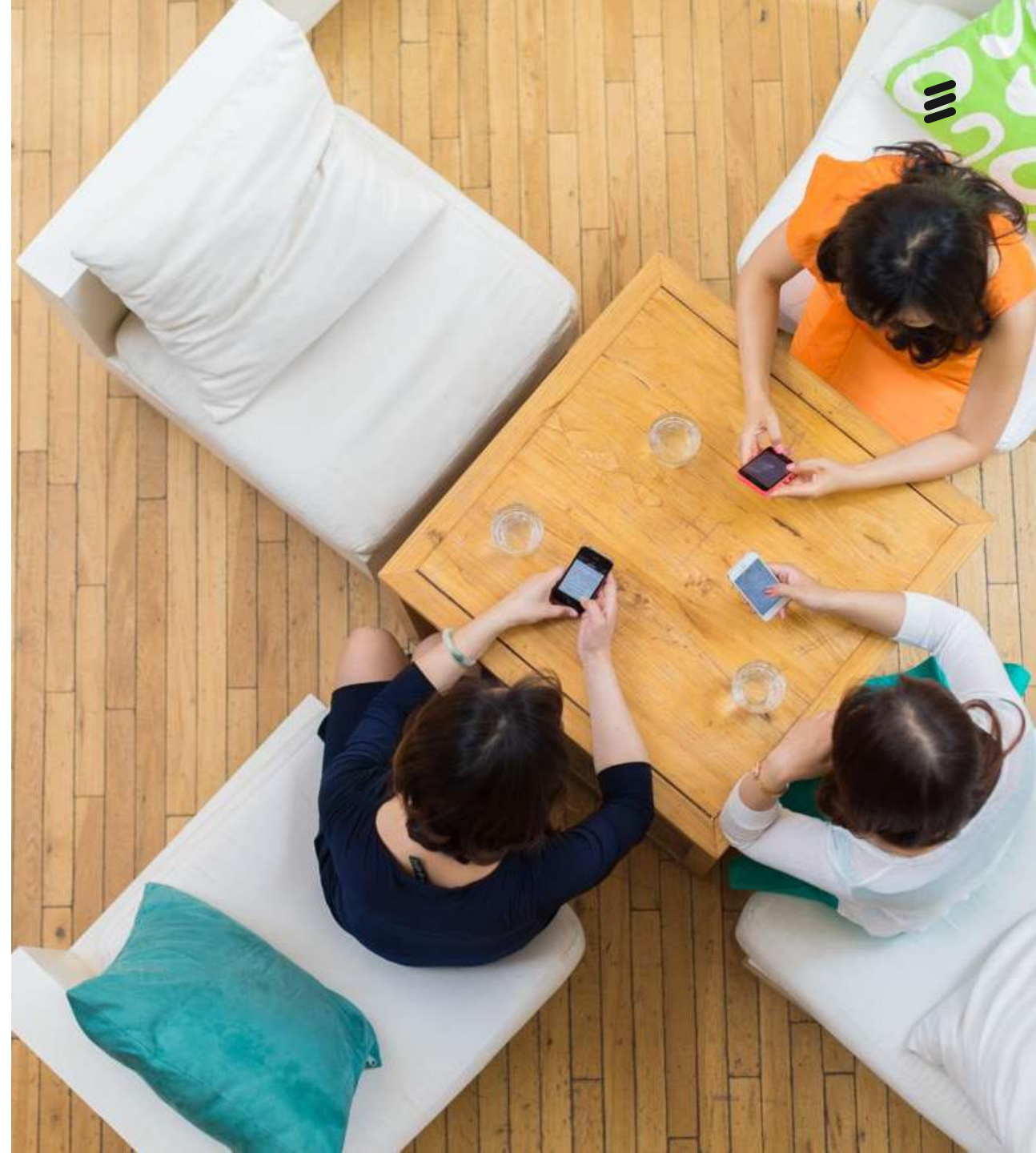


Ericsson Trace Compass update

Matthew Khouzam

Agenda

- Recap of Ericsson's involvement with the project
- The team
- The tracing ecosystem
 - Trace Compass
 - VSCode Extension
 - CDT-Cloud!
- How do we fit into Ericsson?
- Future work
 - Roadmap
 - Where can you help?
- Thank you
- Cool story, time permitting.



No Change!



The goal of this talk is **STILL** to explain where we are coming from to explain our decision making process

Ericsson has been an industrial partner in the research project for over 15 years

We develop and use tools such as LTTng and Trace Compass in the company to solve timing issues and hard to debug problems

Ericsson contributes to and maintains Trace Compass

The Montreal Team has been closely collaborating with the academic and industrial partners. Including US (Texas) and Sweden)

We are working towards having the entire team working at their peak efficiency.

About The Team (May 2024) No Change



Bernd Hufmann – Technical Lead

Patrick Tasse – Employee (Internal Lead)

Marc Dumais - Devops

Elena Giovannetti – Director

Georges Bourret – Line Manager

Steve Crisafulli – Strategic Product Manager

Anh Nguyen – Intern

Kaveh Shahedi – Intern Emiratus

Matthew Khouzam – Product Owner

Austin +Sweden teams – Developing internal use cases in open-source way

The Ericsson Tracing Ecosystem (no change)



- At Ericsson we have very intricate products offered to the public
- At a high level we trace many individual components
- One of the goals of Trace Compass is to provide a unified troubleshooting experience
- **This did not change, there are just more users and more use cases.**

Traces

- LTTng
- Other Linux Tracers
- CTF Hardware
- Chromium style
- Open Telemetry

Logs

- HTTPD
- SSH
- Java (GC)

Trace Compass



- 10.2, 10.3, 11 Soon to be Released
- Customization of Trace Compass analyses
- Using external trace-event-logger library
- Removed incubator callstack
- Minor fixes
- Prep work for Execution Comparison

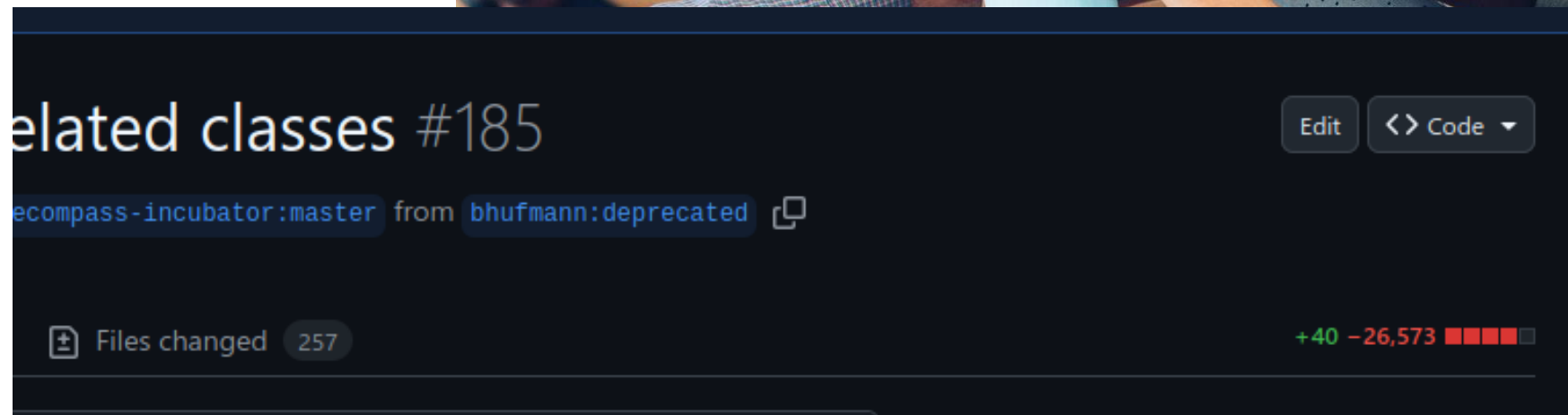


Trace Compass Incubator

- DPDK
- Perfetto on it's way, thanks AMD
- JIFA traces (GC)
- Seems to be stabilizing as a repo for trace types now.

Soon:

- Raw trace events?
- JFR Jifa traces



Eclipse CDT Cloud (repeat)



- Place where contributions are welcome:
 - Full tracing example, like tracevizlab but for an IDE.
 - Expected flow for CDT Cloud Blueprint:
 - Write code
 - Launch with tracing
 - Visualize trace
- PS: for new members, have you done tracevizlab's tutorials? 100% worth it!

<https://github.com/dorsal-lab/Tracevizlab>

TC Cloud Frontend



- Most improvements are UX related as VSCode or Theia is a UI shim over Trace Compass's core logic.
- Features:
 - Filtering , Column headers in gantt.
 - BE/FE URL split to allow running in kubernetes
- Other:
 - **v0.7.3**: Updated for Theia v1.58.5, bumped dependencies (axios, @babel/runtime), and refactored tooltip. JSONBigInt harmonized improved support.
 - **v0.7.0**: Modernized method binding, added sorting/filtering to timegraph, and updated CI to Node 20.
 - **v0.6.0**: Updated to Theia 1.55.0, added ADR for output descriptors, and bumped key dependencies.
 - **v0.5.0**: Improved type safety in SignalManager and added basic issue/pull-request templates.
 - Community is contributing, thanks Blackberry!

Trace Event Logger



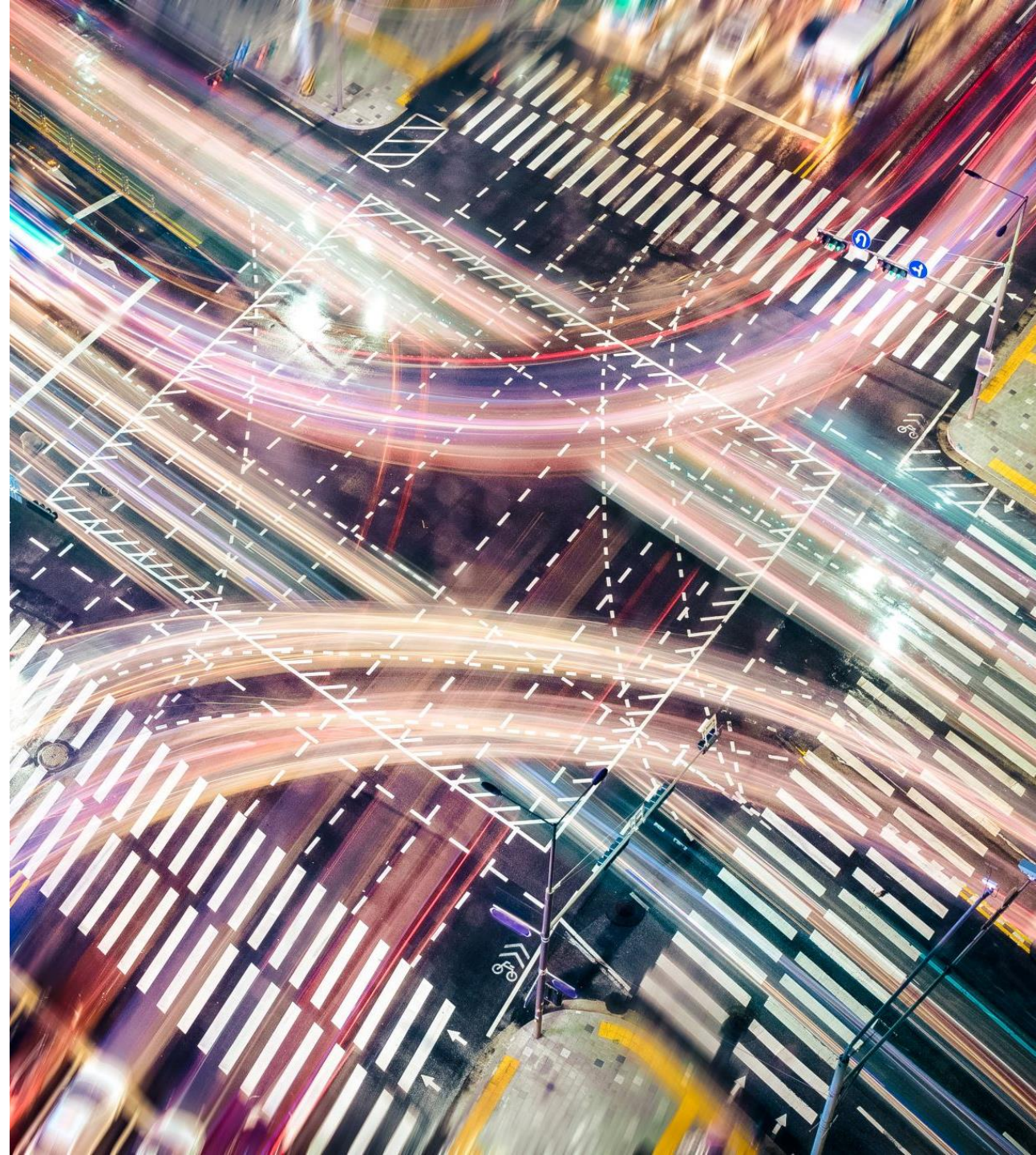
- High Speed simple java logger.
- Generates Trace Even Traces
- Uses JUL
 - ~75 ns/event instrumentation only
 - ~1 us/event write to disk
- Using plain JUL
 - ~45 us/event write to disk
- If you're instrumenting a java application, use it instead of hand crafting event.

Where do we fit into Ericsson?

- We make a tool that's used by tools to support developers and users.
- Most of our users don't know what trace compass is.
- 3 main use cases:
 - Antenna
 - Software layer (Many KPI and aggregates)
 - Hardware layer (Gantt charts)
 - Backplane: Trace Compass with LTTng
 - Support: Trace Compass with custom logs



Future Work



VSCode extension

- Customization
- Support for reports (progress!)
 - Generic XY
 - Flame Graph
 - Pie Charts
 - Roofline Models
- Hardware specific views, will help AMD?
- Special plots, better TMLL integration:
 - Bubble
 - Sankey
- Collaborative investigations when tech is there



Road Map



Be a good Ericsson
Citizen – Ongoing

Software Compliance –
We want to use FOSS
in Ericsson!

Expand community –
Poly committers?

Expand internal
community

Open-Source Activity
and Leadership
(General) - Ongoing

Trace Compass
Releases (Eclipse) -
Ongoing

Open-Source Features
to support internal
implementation of
trace viewer (Theia) -
Ongoing

Continue to support
internal hardware team

Continue to support
internal Linux teams

Trace Compass



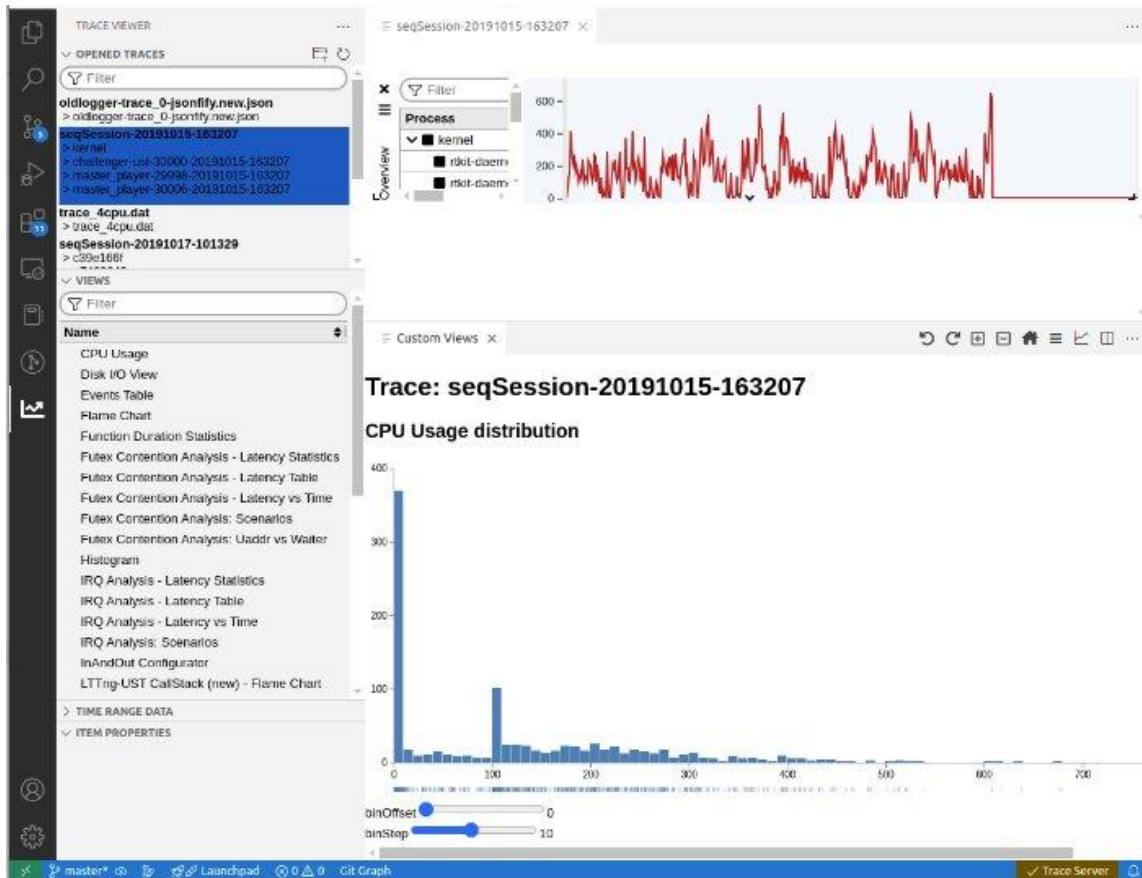
- Internal Support – Ongoing
- Additional tracer support:
 - CTF2
 - DPDK
 - SQLite DB
 - Perf (intel PT)
 - Perfetto (Thanks AMD!)
 - ~~Protobuf?~~
- Support Poly's efforts re: Partial State system
- Support Poly's efforts re: Diff Flamegraph/stats/critical path
- **Performance improvement – On demand**

Cool Story



Trace Compass.ai

Custom Charting



- Testing using VEGA
- Able to get a density chart out of XY
- Able to plot arbitrary columns
- INVESTIGATION, not delivered

LLM help with Traces

Single events,
Series == hallucinations

Ollama Analysis Result Using llama3.2

Event Explanation

The given event is from a trace and represents a `sched_switch` event.

Event Details

- **Timestamp :** 18:09:06.277 342 899 - This is the timestamp when the event occurred, with nanosecond precision.
- **Channel :** k_3 - The channel through which the event occurred.
- **CPU :** 0 - The CPU on which the event occurred (in this case, CPU 0).
- **Event type :** `sched_switch` - This indicates that a context switch occurred, where the CPU switched from one process to another.
- **Contexts :**
 - `prev_comm="swapper/0"` - The name of the previous process executing on the CPU (swapper/0).
 - `prev_tgid=0` - The ID of the previous process.
 - `prev_prio=20` - The priority of the previous process.
 - `prev_state=0` - The state of the previous process (0 indicates running).
 - `next_comm="kiting-consumer"` - The name of the next process that will be executing on the CPU (kiting-consumer).
 - `next_tgid=2043` - The ID of the next process.
 - `next_prio=20` - The priority of the next process.
 - `context.packet_seq_num=0` - The packet sequence number for the context switch.
 - `context.cpu_id=0` - The CPU ID associated with the context switch.

Packet Context

- **Packet Context :**
 - `packet_size=35777236` - The size of the packet containing the event data (in bytes).
 - `events_discarded=0` - The number of events that were discarded during processing.
 - `cpu_id=0` - The CPU ID associated with the packet.

Timestamp

- **Timestamp :** ns : 1539814746277342899 - A timestamp in nanoseconds, representing the time when the event occurred.

Customization made easier!

Should fix a lot, critical path, in-and-out, xml
loading, filtering...

TRACE VIEWER

OPENED TRACES

kernel

VIEW

NAME

CPU Usage

Disk I/O View

Events Table

Futex Contention Analysis - Latency Statistics

Futex Contention Analysis - Latency Table

Futex Contention Analysis - Latency vs Time

Futex Contention Analysis: Scenarios

Futex Contention Analysis: Uaddr vs Waiter

Histogram

IRQ Analysis - Latency Statistics

IRQ Analysis - Latency Table

IRQ Analysis - Latency vs Time

IRQ Analysis: Scenarios

InAndOut Configurator

My Latency In And Out Analysis

My Latency In And Out Analysis - Flame Chart

My Latency In And Out Analysis - Latency Statistics

My Latency In And Out Analysis - Latency Table

My Latency In And Out Analysis - Latency vs Time

Memory Usage

Resources Status

TIME RANGE DATA

View Range Start:
1332170682440133097

View Range End:
1332170682443757196

Selection Range Start:

Selection Range End:

kernel

Filter

kernel

total

Overview

My Latency In And Out Analysis

Label	Min	Max	Average	Std Dev	Count	Total	Min Time Range
kernel	88 ns	122.082 μs	9.392 μs	11.06 μs	5236	330.988 ms	[133217068299374
total	88 ns	122.082 μs	9.393 μs	11.06 μs	5236	330.988 ms	[133217068299374
Latency	838	122.082 μs	9.393 μs	11.06 μs	5236	330.988 ms	[133217068299374

Select a schema

In And Out Analysis org.eclipse.tracecompass.incubator.internal.inandout.core.config
Configure In And Out analysis using file description

1 /**
2 * A toolbar is located in the top-right
3 * • Submit the current config
4 * • Save this config for future use
5 * • Load an existing config file
6 *
7 * You can also submit by simply closing the file
8 */
9 {
10 "id": "org.eclipse.tracecompass.incubator.internal.inandout.core
11 "name": "My Latency In And Out Analysis",
12 "description": "My Latency In And Out Description",
13 "options": {
14 "specifiers" : [
15 {
16 "label": "Latency",
17 "inRegex": "(\\S*)_entry",
18 "outRegex": "(\\S*)_exit",
19 "contextInRegex": "(\\S*)_entry",
20 "contextOutRegex": "(\\S*)_exit",
21 "classifier": "CPU"
22 }
23]
24 }
25 }

Where can you help?

Drop a review, a bug or a feature request

Fix a bug, especially if it helps you and is not in our interest

Share use cases

We have limited bandwidth, we would appreciate code contributions

Thank You



[This Photo](#) by Unknown author is licensed under [CC BY-SA](#).

Code Contributors

Issue Reporters

Designers

Community Maintainers

Enthusiasts

Steak Holders

