



Updates on DPDK Analysis Integration in Trace Compass

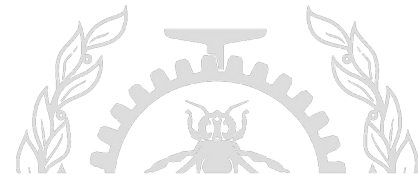
Adel Belkhiri

December 5, 2024

Polytechnique Montréal
DORSAL Laboratory

Agenda

- ① Introduction
- ② Current Work
 - Ethdev and mempool Libraries
- ③ Demo
- ④ Future Directions



Data Plan Development Kit

- Processing network traffic in kernel is inefficient, which explains the rise of high-performance userspace frameworks:
 - Examples include **DPDK**, **Netmap**, **F-Stack**, and **PF_Ring**.
- DPDK: Comprehensive set of data plane libraries and NIC drivers designed to accelerate network packet processing in userspace:
 - Polling Mode Drivers (PMDs)
 - Huge pages
 - Lock-free data structures
 - Batch processing

Context and Motivation

- PhD research track proposed by Martin Pepin (Ciena), focusing on analyzing the performance of DPDK applications.

- Scientific Article:

A. Belkhiri, M. Pepin, M. Bly and M. Dagenais, Performance Analysis of DPDK-based Applications Through Tracing, Journal of Parallel and Distributed Computing, 2023.

- **Key Contributions**

- Instrumentation of DPDK libraries, using LTTng
- Development of comprehensive analyses in Trace Compass:
 - Environment Abstraction Layer (EAL), Vhost, Eventdev, Pipeline, and flow classification libraries (ACL, Hash, LPM.)



Context and Motivation

- Trace library in DPDK (introduced in version 20.05):
 - Implements the Common Trace Format (CTF) natively
 - Provides control and fast-path tracing with minimal performance overhead (~20 cycles per tracepoint)
- Our Goal:
 - Convert existing LTTng instrumentation to DPDK trace format, and integrate implemented DPDK analyses into Trace Compass Incubator
 - DPDK's instrumentation patch (commit #841e87d)
 - Lcore (logical core) analysis merged in TC (commit #717c0a9)



Previous Work

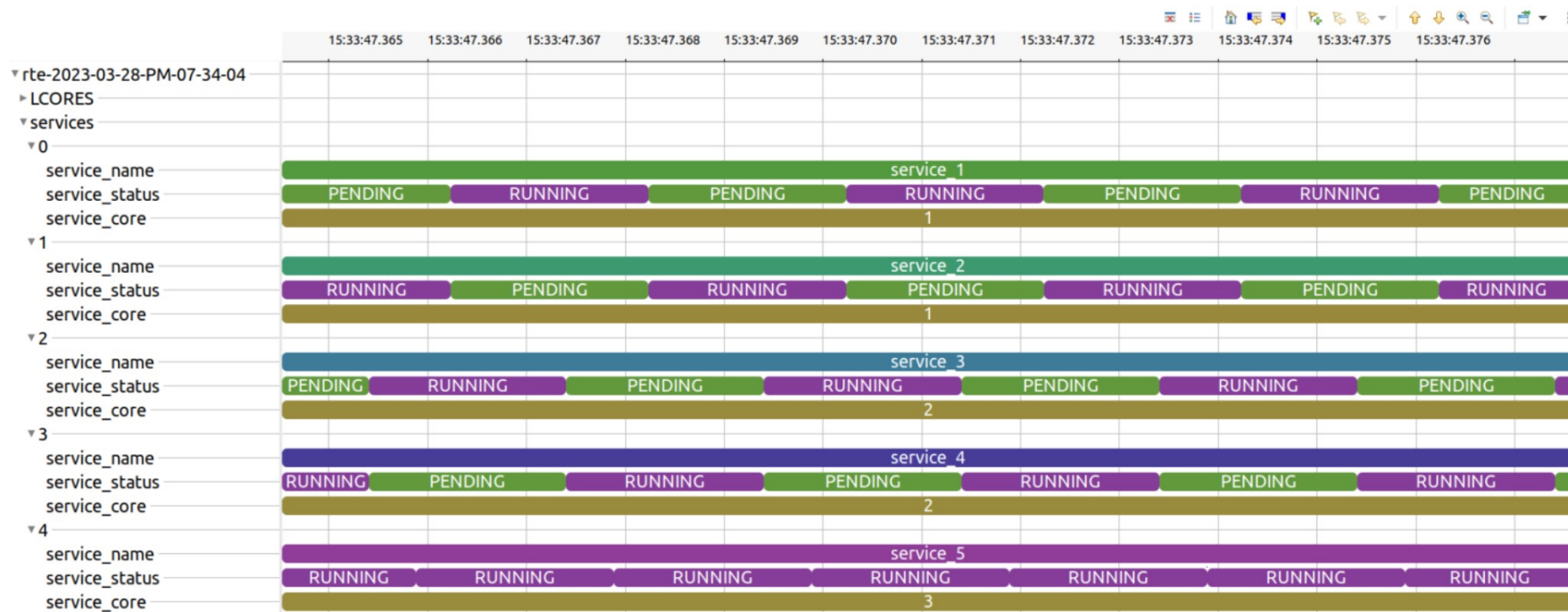


Fig. Service execution on DPDK lcores

Ethdev Library

- Five new analyses for the DPDK Ethernet library:
 - Instrumentation patch merged in DPDK v24.11 (commit #e075ca1)
 - Trace Compass Incubator - PR #83
 - Thank you Matthew and Arnaud for the comments and discussions!

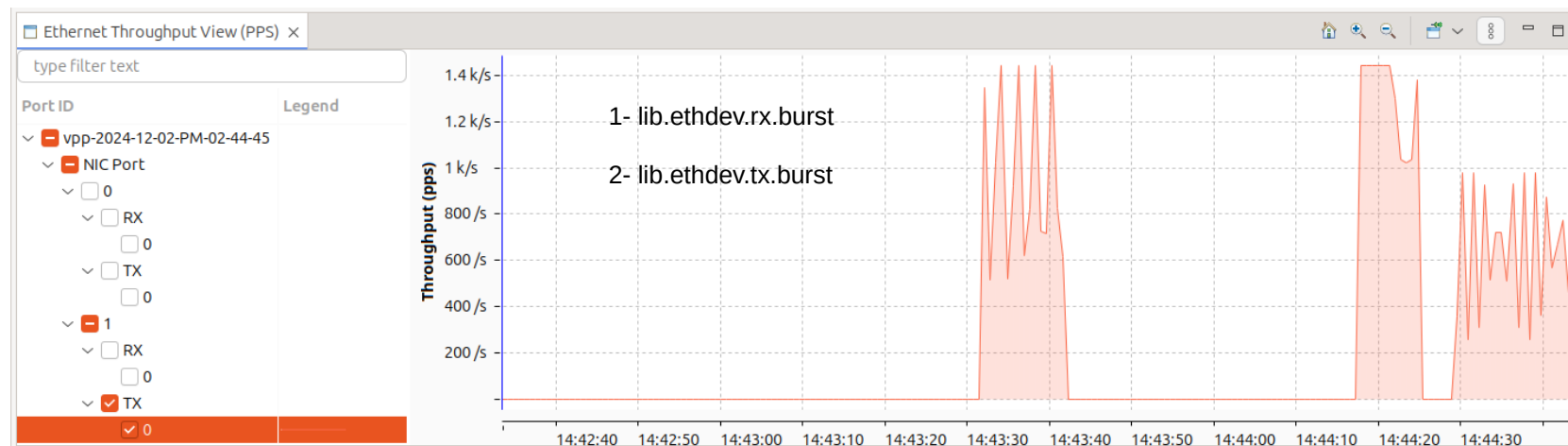
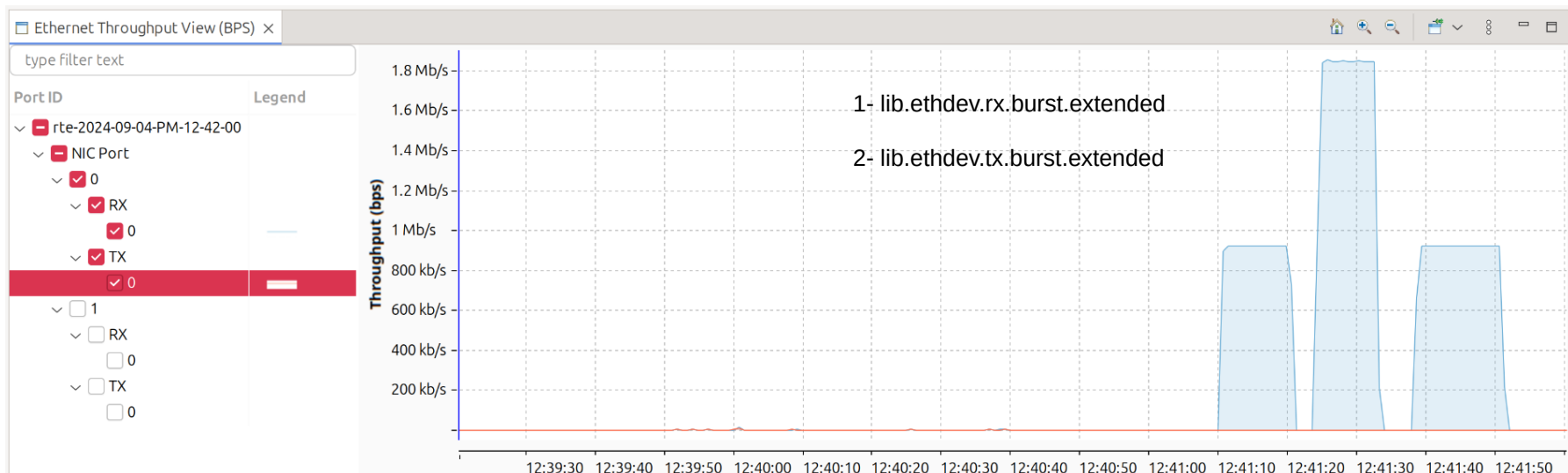


Fig: Packet RX and TX rate, per port queue

Ethdev Library

- Packet RX and TX throughput (in bps), per port queue ...
 - ➔ Preloading a custom library (librte_profile.so) is, however, required to obtain the size of packet batches.



`LD_PRELOAD=./librte_profile.so /usr/bin/vpp -c /etc/startup.conf`

Ethdev Library

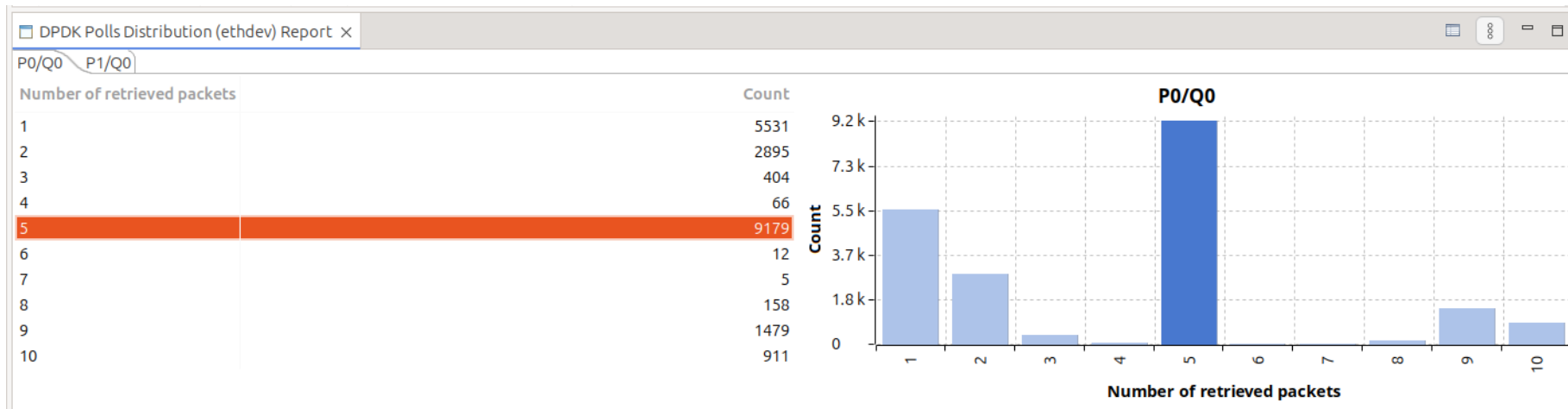


Fig: Distribution of packets retrieved in a single RX batch

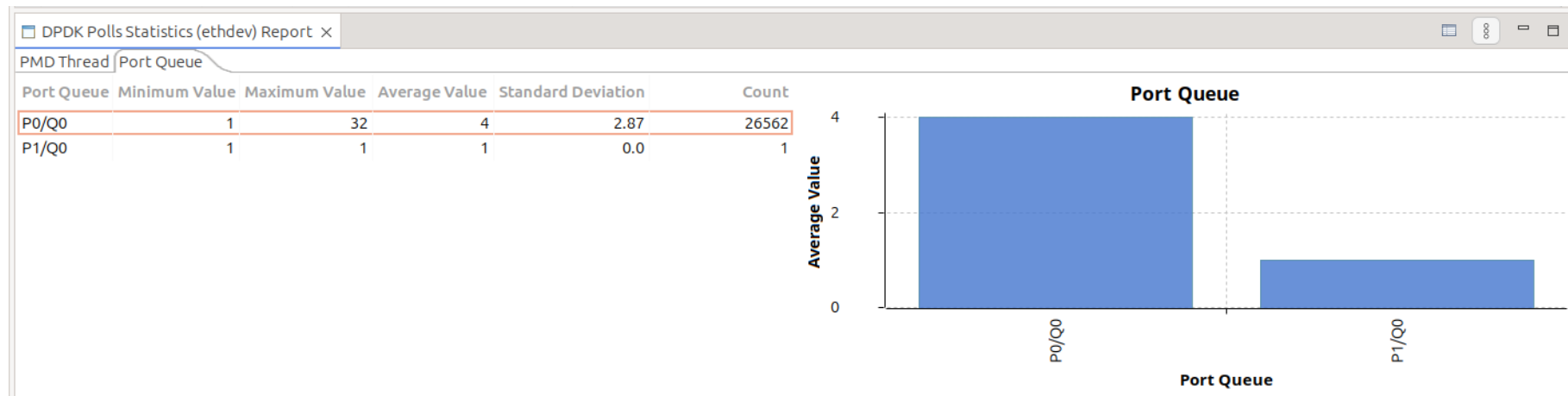
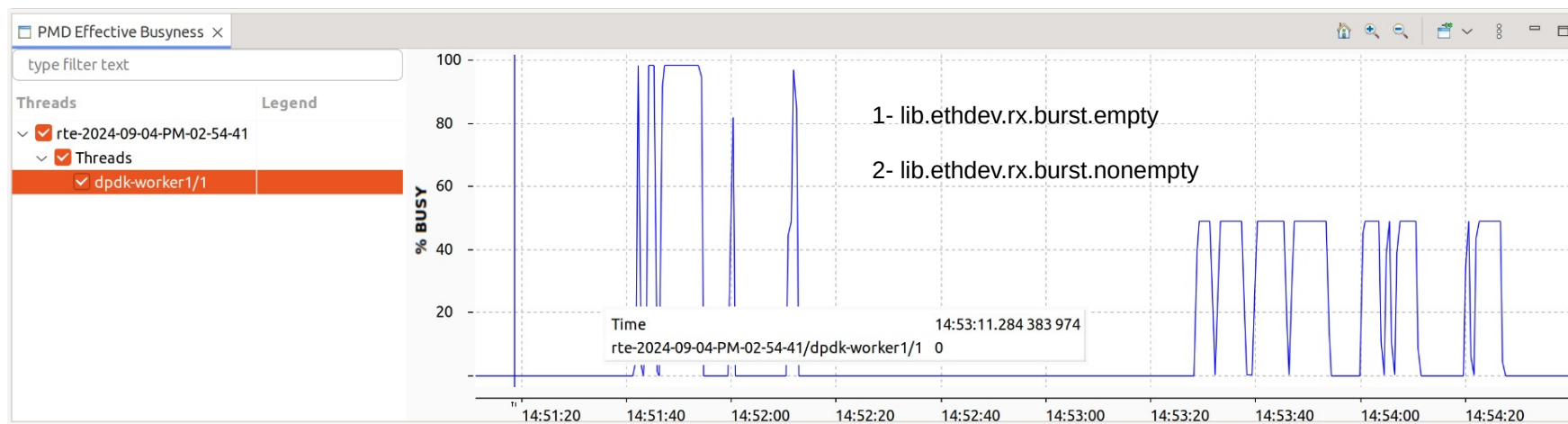


Fig: Packet RX statistics per thread and per port queue

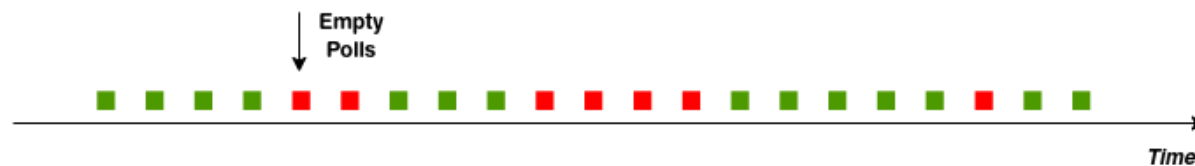
Ethdev Library

- PMD threads effective busyness



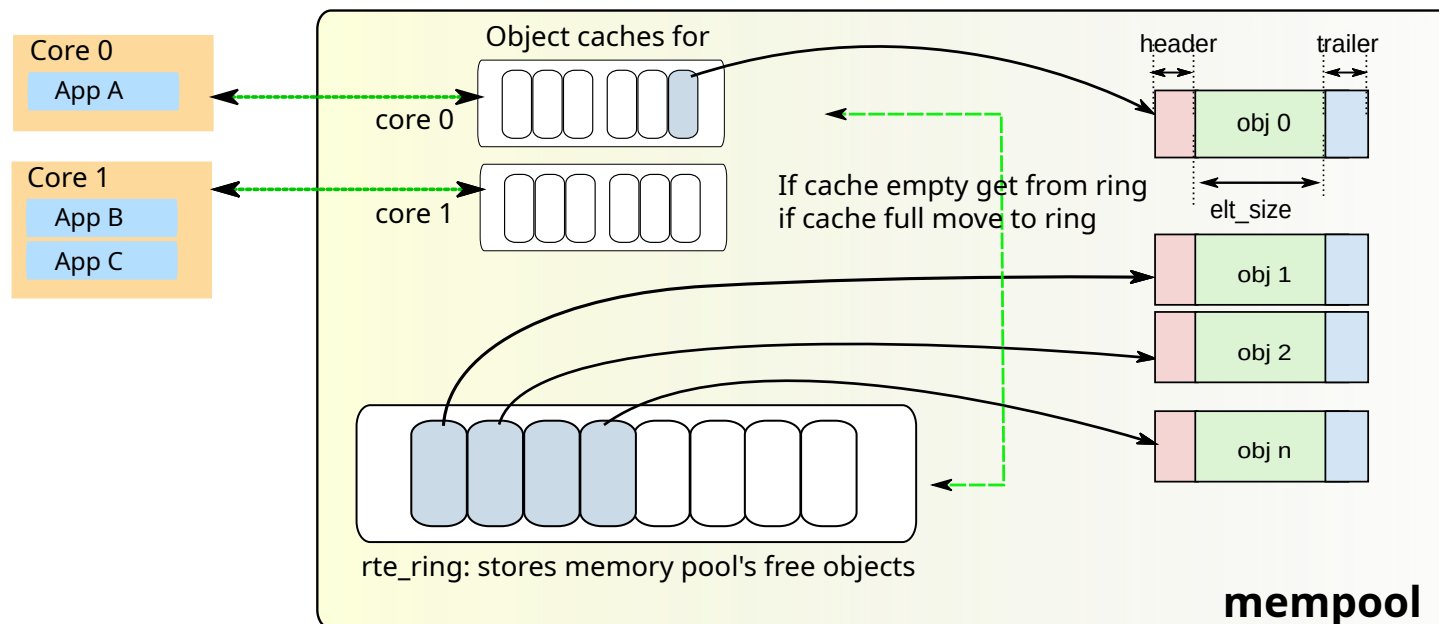
Formula:

$\%Busyness = \text{Durations of successful polls} * 100 / \text{Total Duration}$



Mempool Library

- Mempool is a memory management structure used to manage and recycle a fixed number of memory objects of the same size.



*Source: https://doc.dpdk.org/guides/prog_guide/mempool_lib.html

Mempool Library

- Three analyses to assess the efficiency of mempool utilization in DPDK-based applications:
 - Mempool usage
 - Object allocation and freeing rates
 - Mempool cache misses

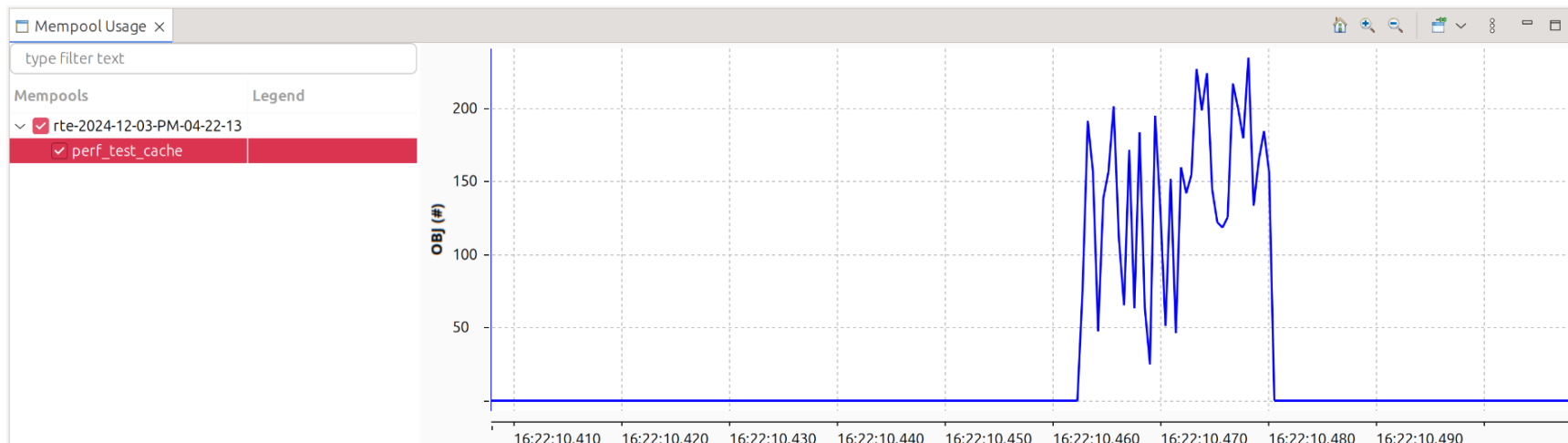


Fig: View showing the variation in the number of used mempool objects over time

Mempool Library

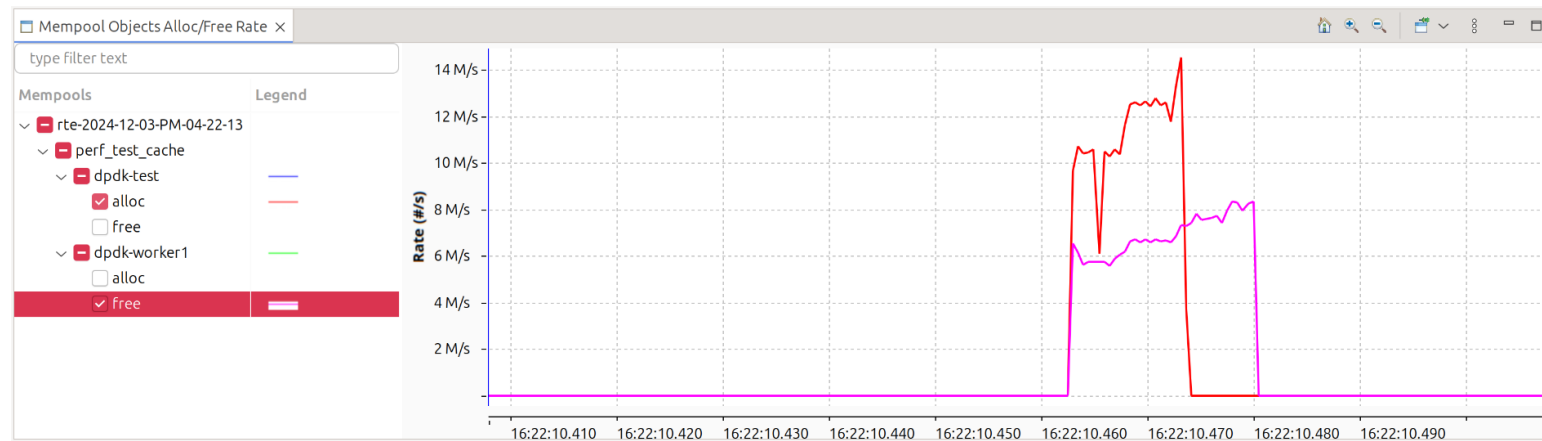


Fig: View showing the rate at which threads allocate and release objects

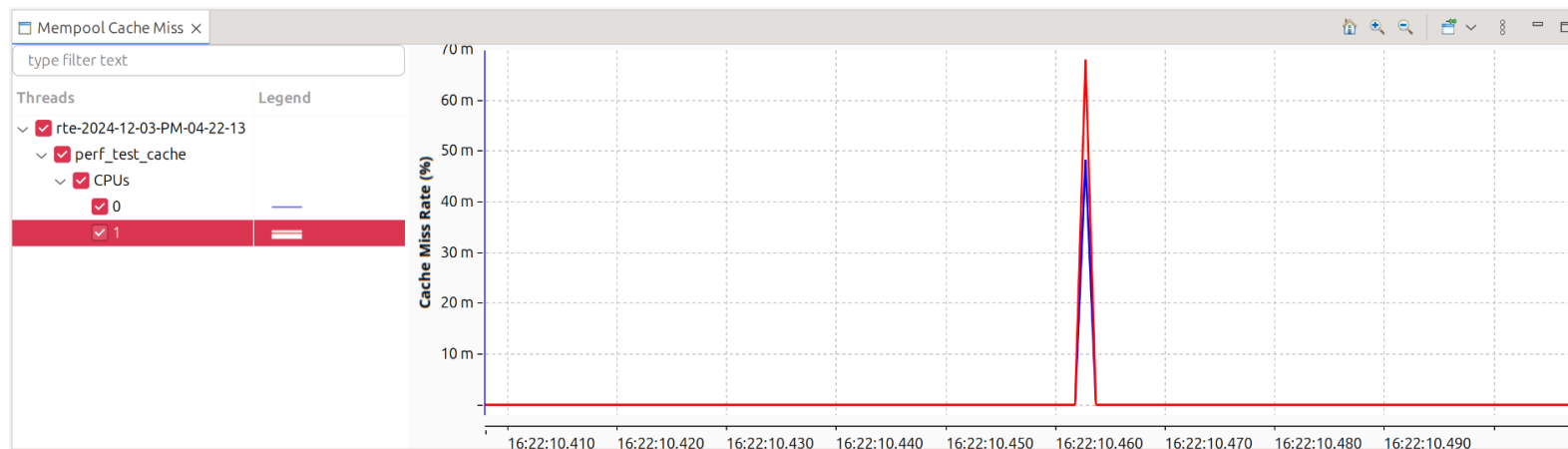
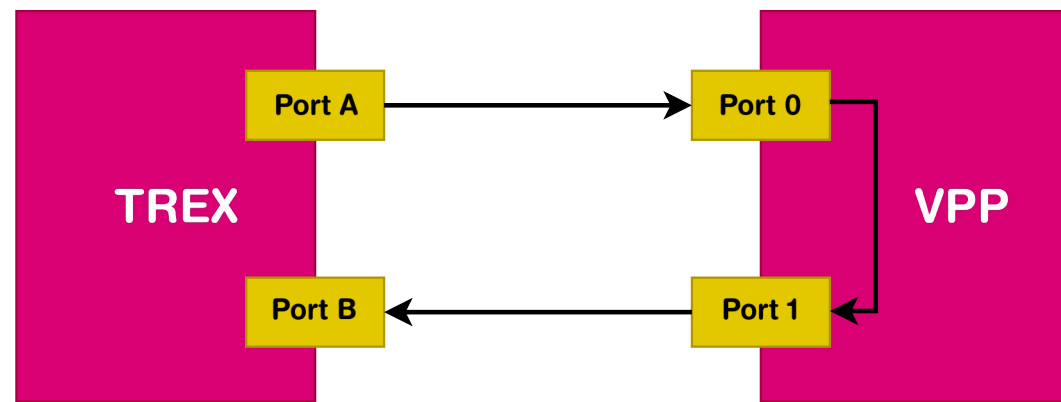


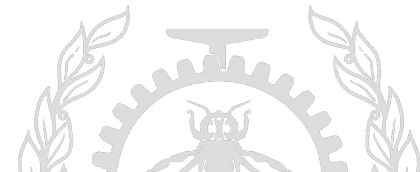
Fig: Measuring the cache misses within the mempool



Video

Future Work

- Continue integrating remaining DPDK analyses into the Trace Compass Incubator and submitting instrumentation patches to the DPDK community
- Develop comprehensive analyses combining kernel and user-space tracing using **LTTng** and the **DPDK native tracer**





Questions?

adel.belkhiri@polymtl.ca

