GPU Tracing and Profiling

Arnaud Fiorini

May 6, 2018

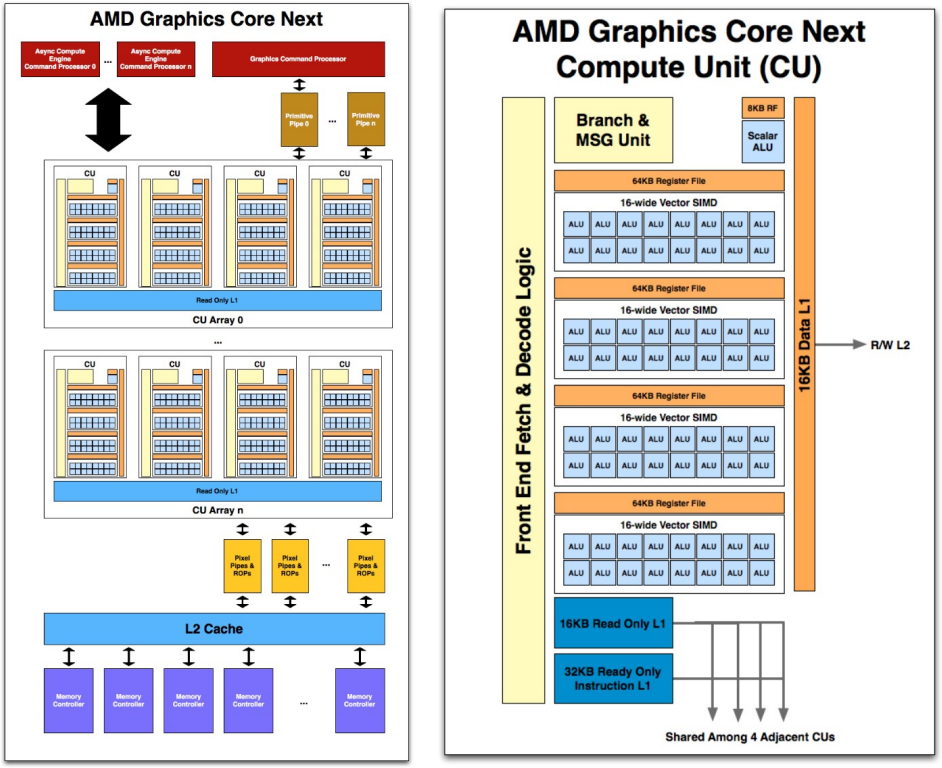
Polytechnique Montréal

Laboratoire **DORSAL**

Context

Context

- AMD GPUs have multiple compute units (equivalent to stream multiprocessor in NVidia)
- Multiple threads can execute at the same time on one compute unit



GCN Architecture (Smith, 2011)

Previous work

Previous work

- LTTng-HSA: Bringing LTTng tracing to HSA-based GPU runtimes by Paul Margheritta

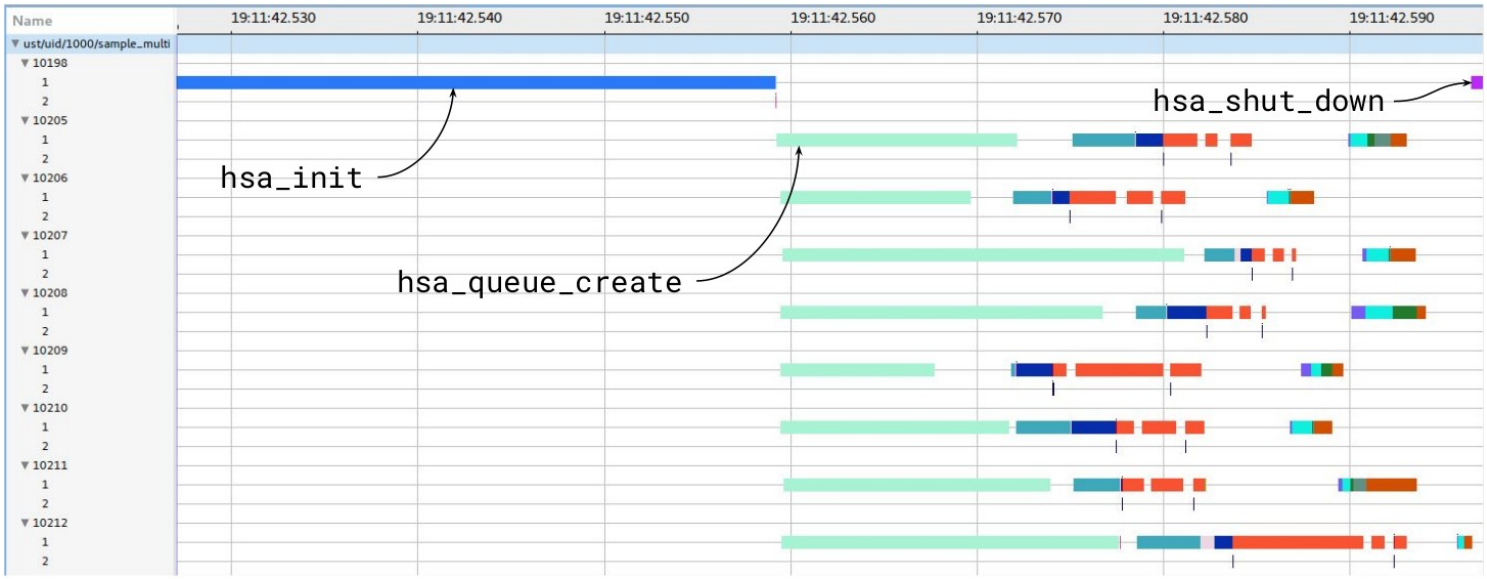


Figure 4.4 The call stack view of a simple application running eight kernels concurrently. The main thread and the eight children threads are shown, with two levels of nested calls in each case. For each segment, the corresponding function call and duration are shown when hovering on it.

Previous work

- LTTng-HSA: Bringing LTTng tracing to HSA-based GPU runtimes by Paul Margheritta

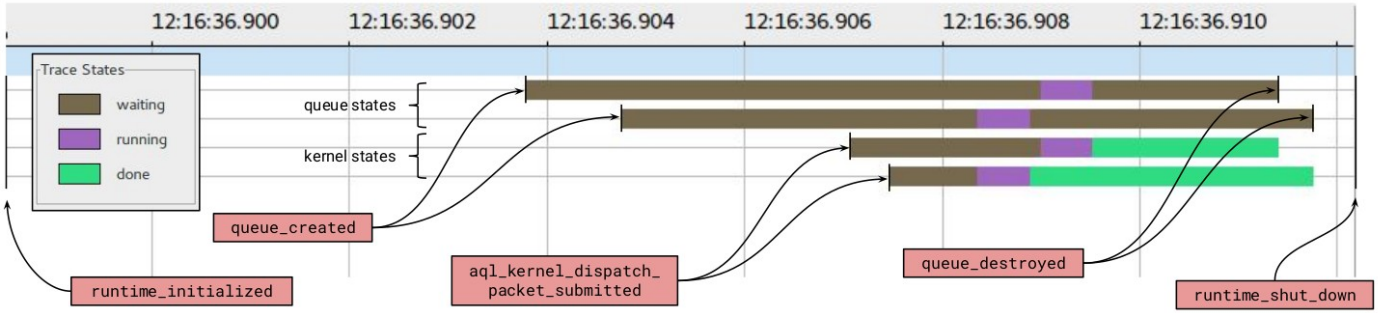
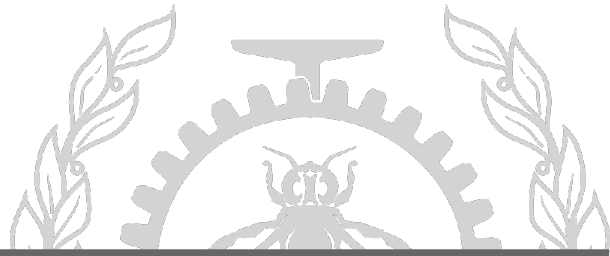
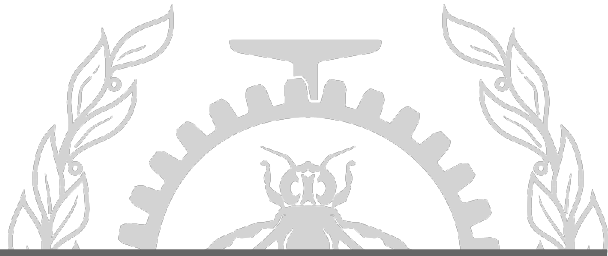


Figure 4.5 The queue profiling view of a simple application running two GPU kernels dispatched from two separate queues. The view shows the state of each queue (first two timelines) and each kernel (last two timelines). For each segment, the corresponding state and duration are shown when hovering on it. All the trace events linked with the view, including those indicated here, are interactively visible when using the view in Trace Compass.



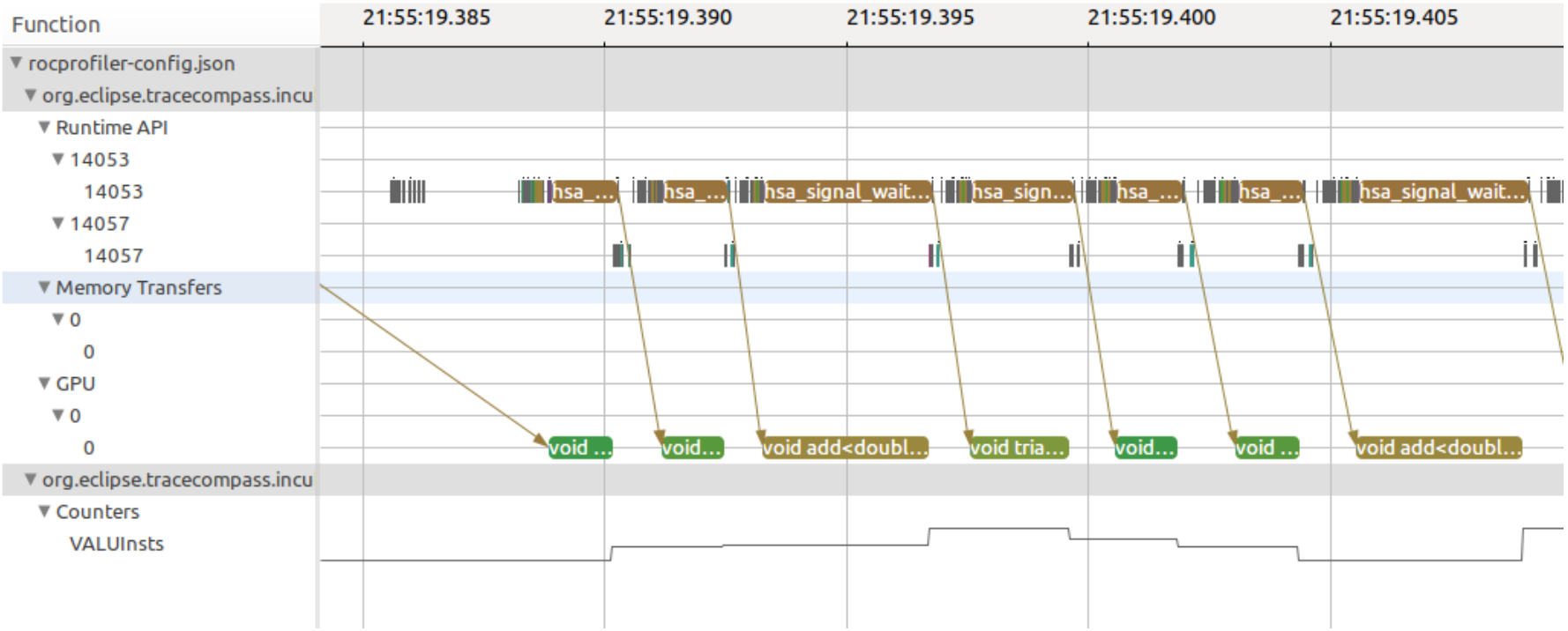
Previous work

- The views shown previously are hard to interpret
- It was necessary to go further in the analysis to help the user understand the behavior of his program



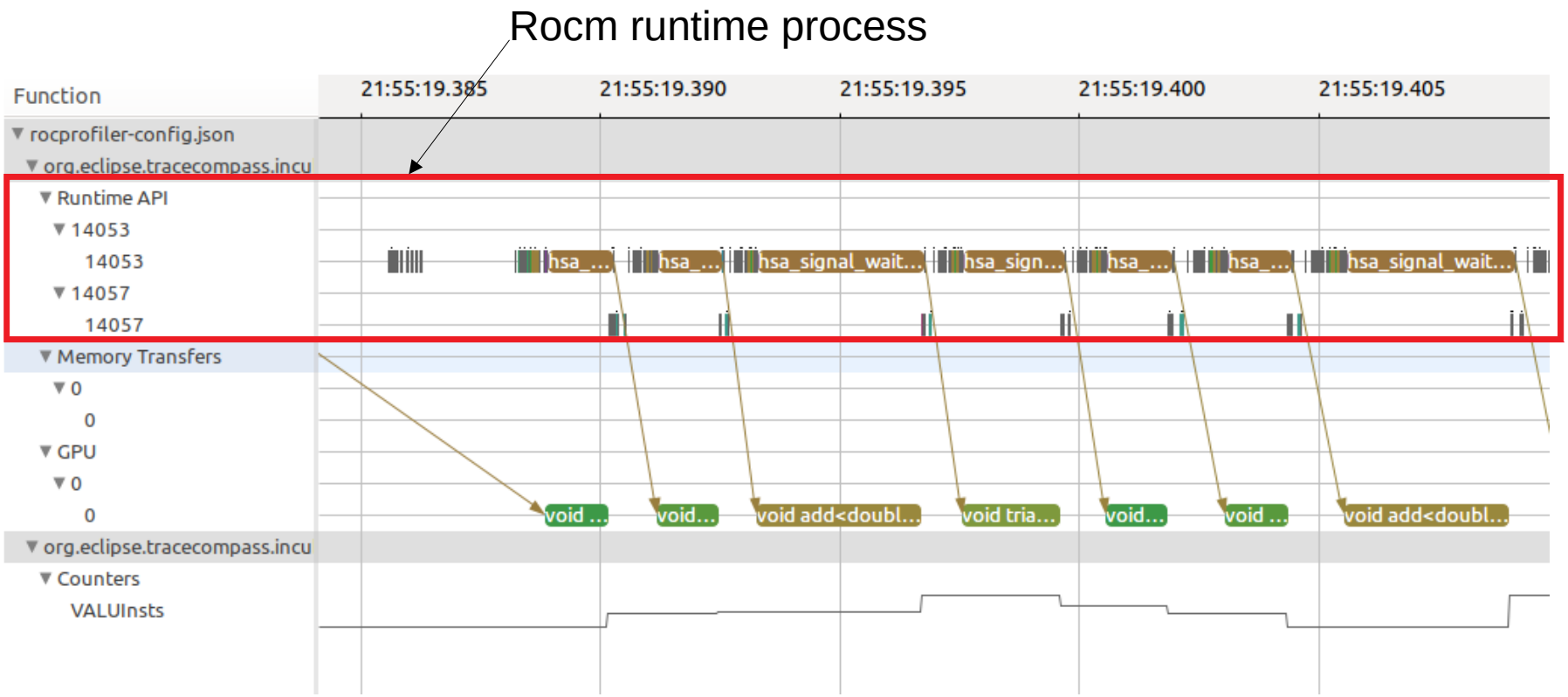
Work in progress

Work in progress

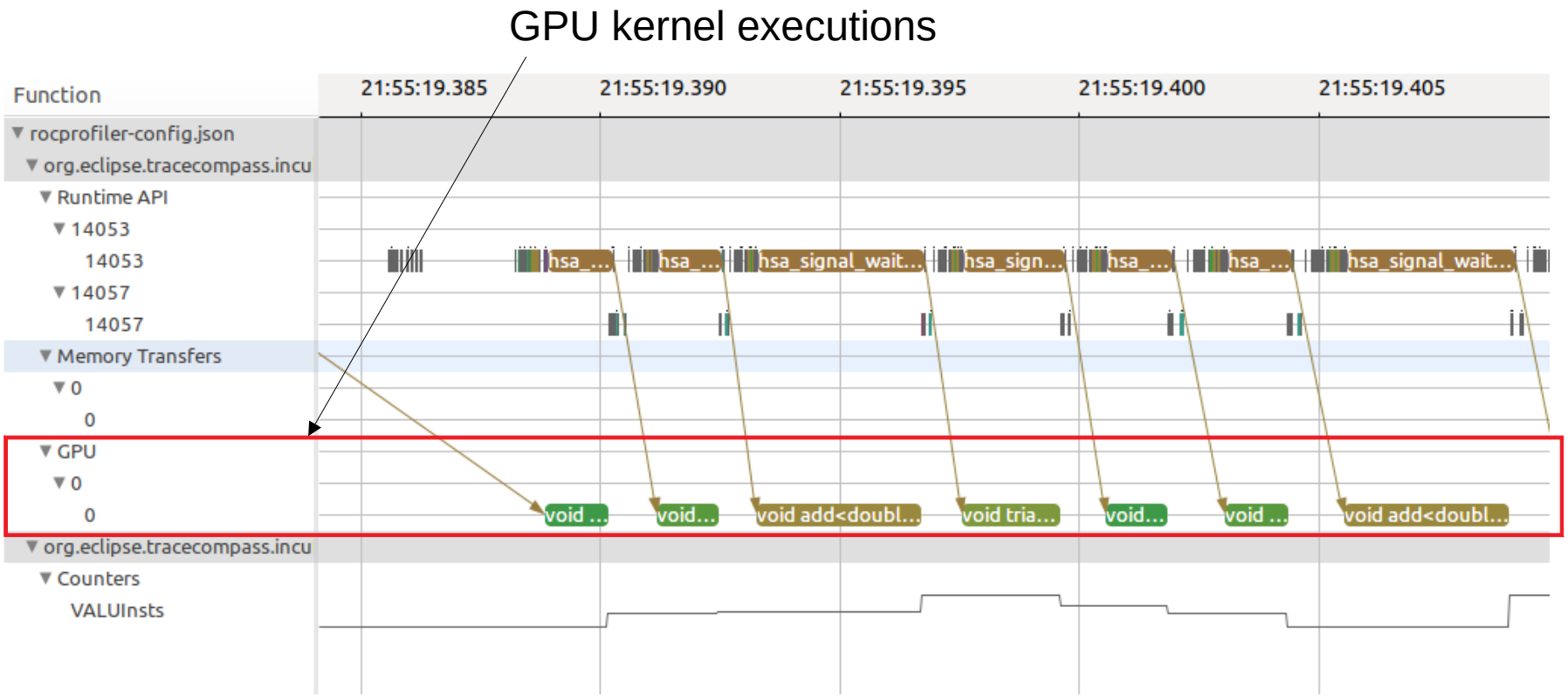


- It will be possible to have this plugin in Theia and VS Code

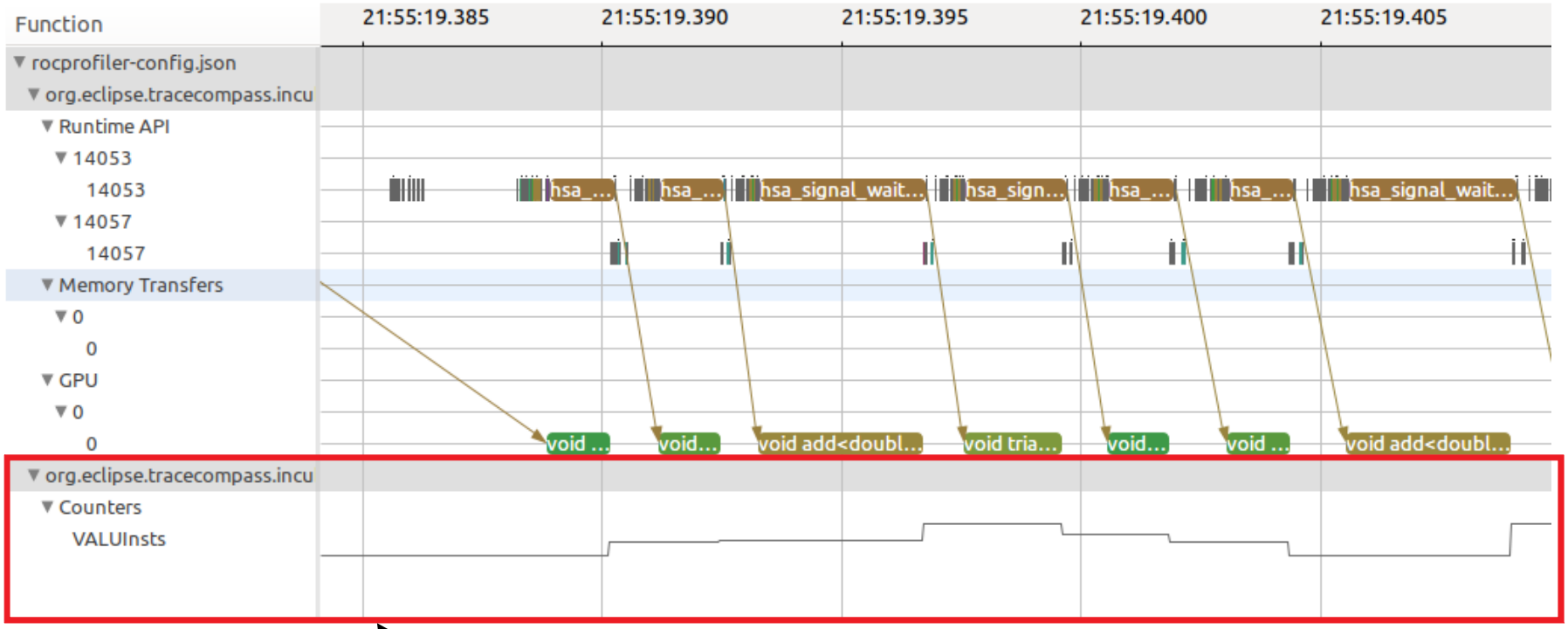
Work in progress



Work in progress



Work in progress

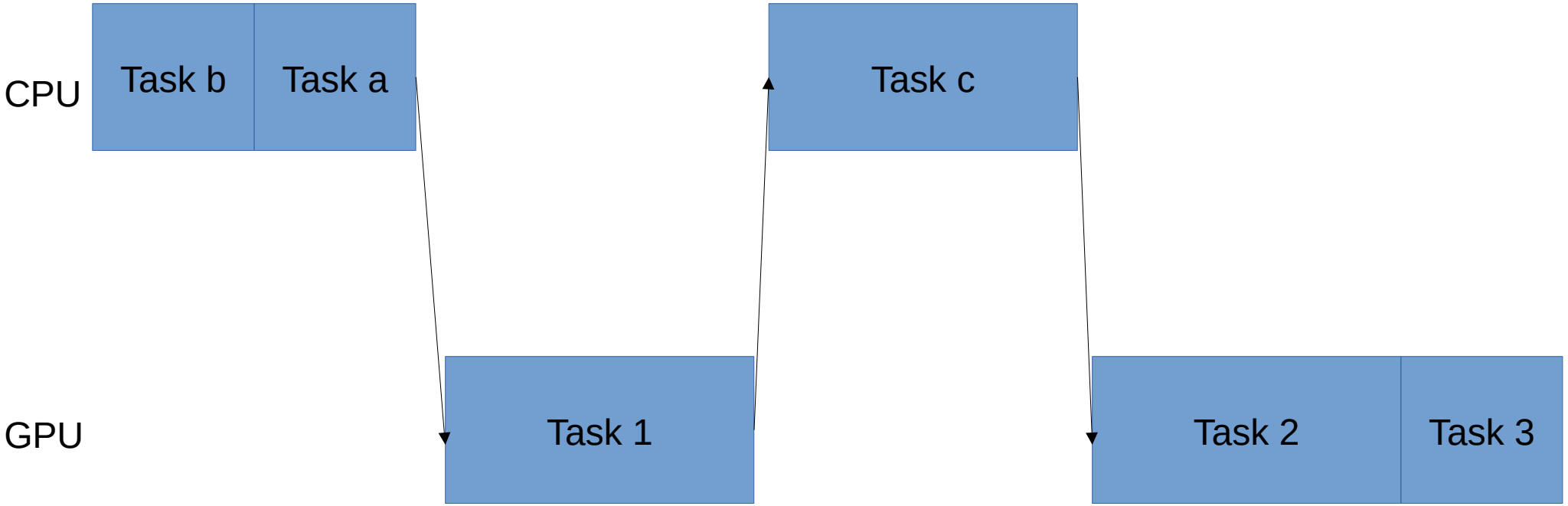


Performance counters
with respect to time

Future work

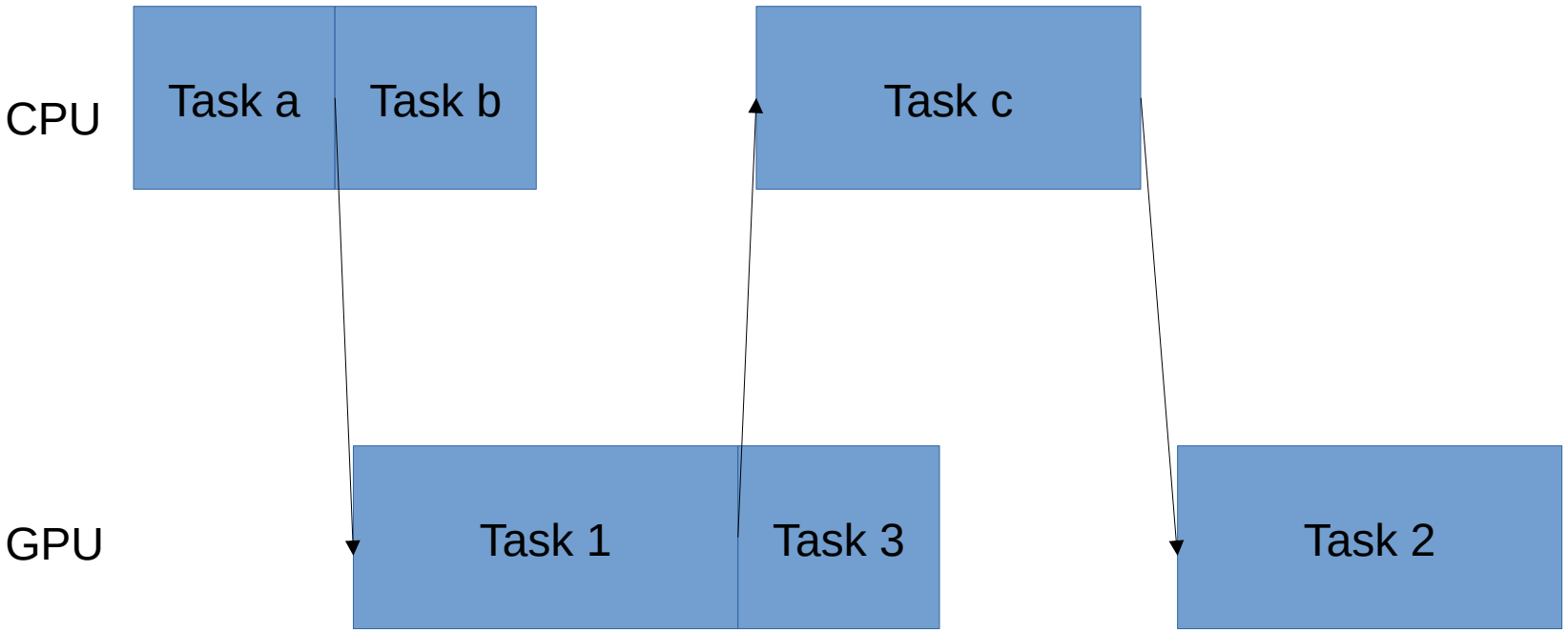
Future work

- Critical Path Analysis between CPU and GPU



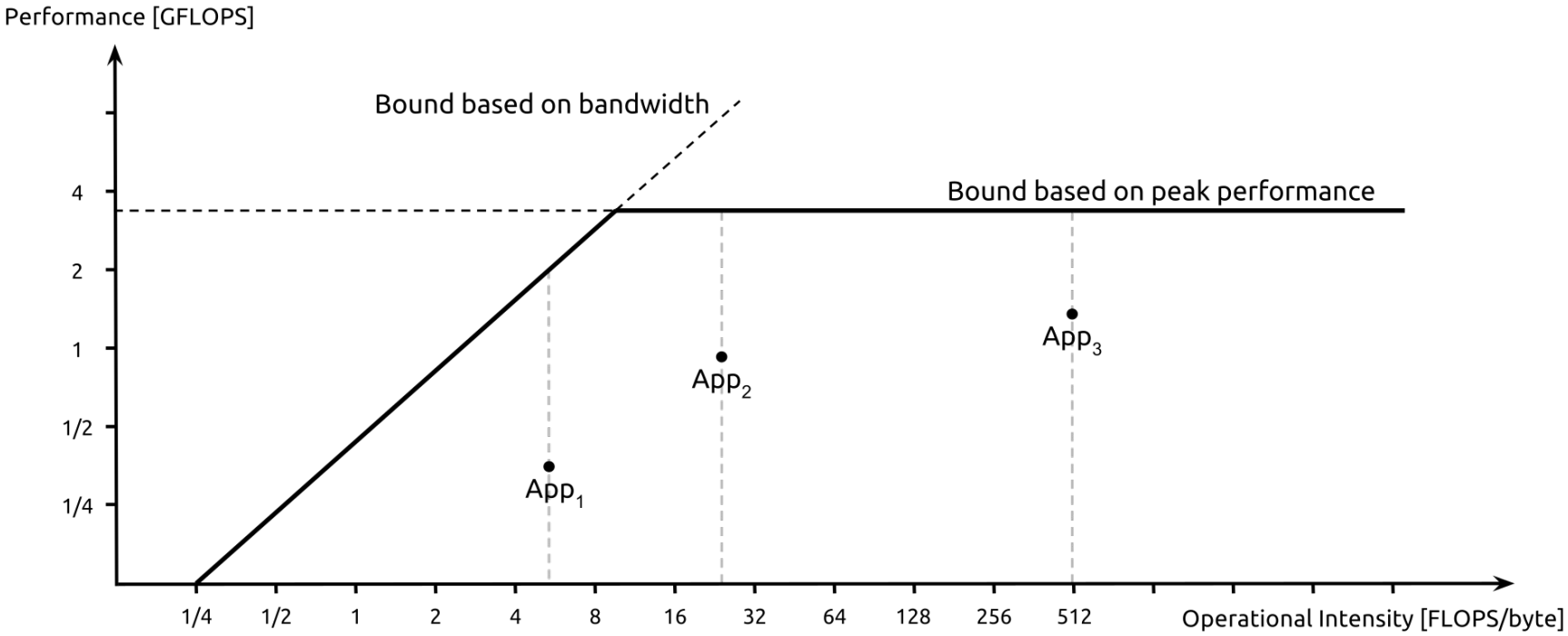
Future work

- Critical Path Analysis between CPU and GPU



Future work

- Roofline Model



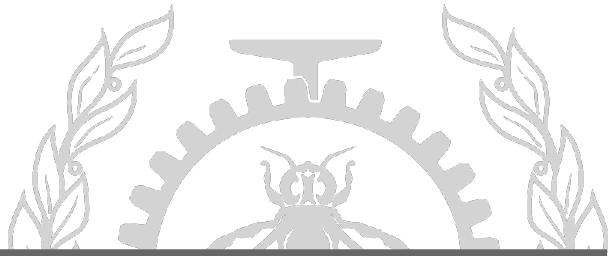
https://en.wikipedia.org/wiki/Roofline_model#/media/File:Example_of_a_Roofline_model.svg

Future work

- Roofline Model

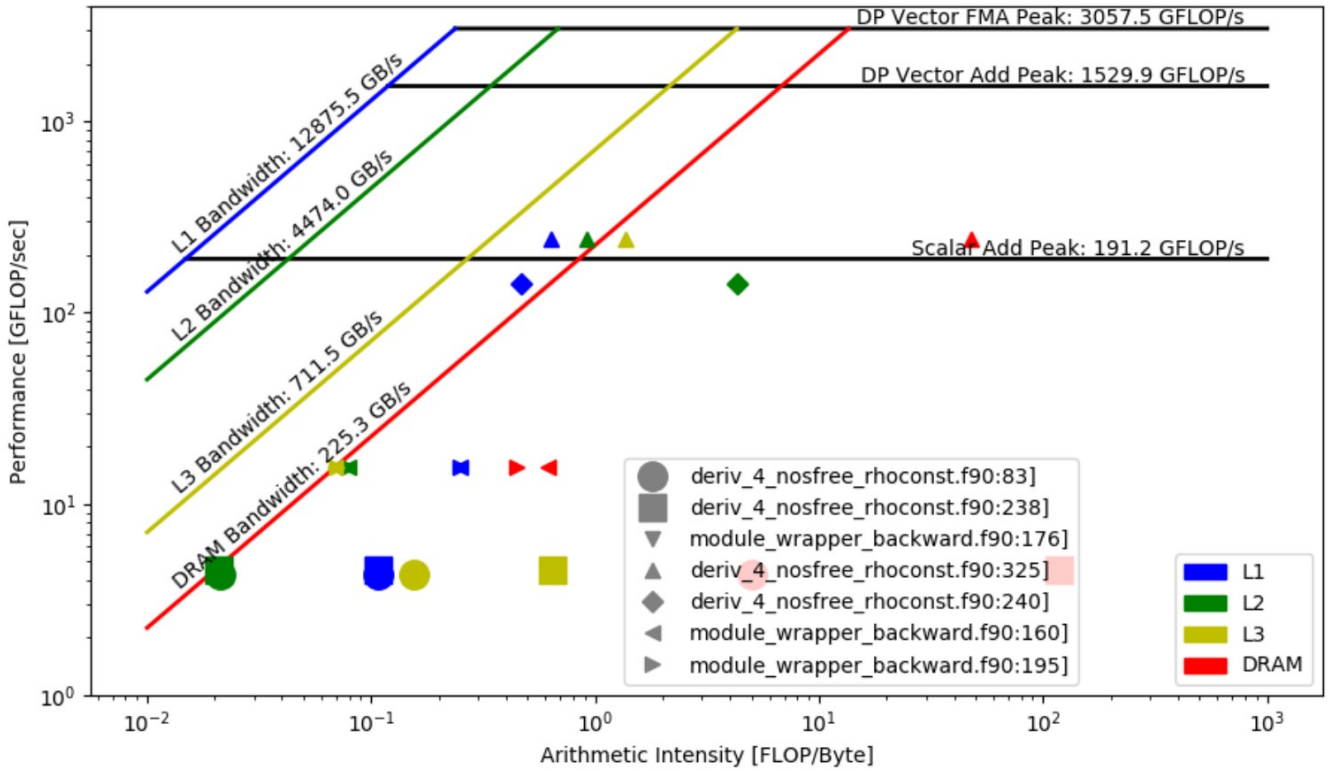
Three measurements are needed for this model:

- Number of floating point operations executed
- Bytes transferred from memory
- Execution time



Future work

- Roofline Model “Cache-aware”



T. Koskela et al., “A Novel Multi-Level Integrated Roofline Model Approach for Performance Characterization”

Future work

- Top-Down Analysis
  - Shows at which step of the processor pipeline, the program is bound

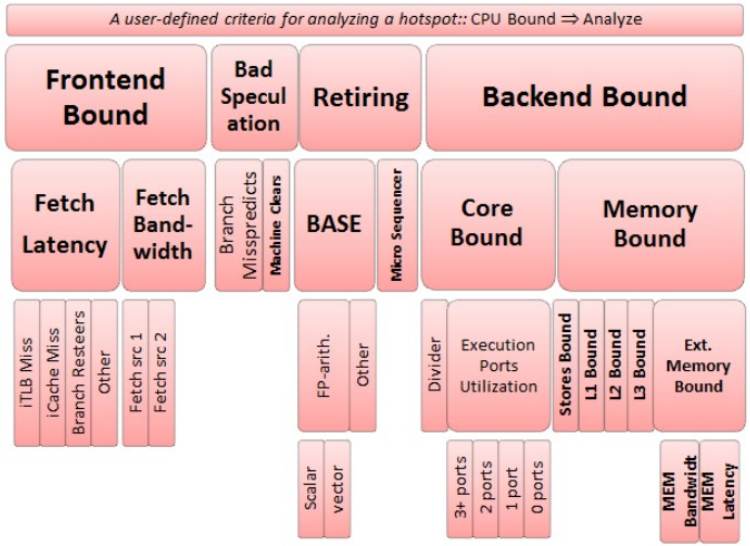
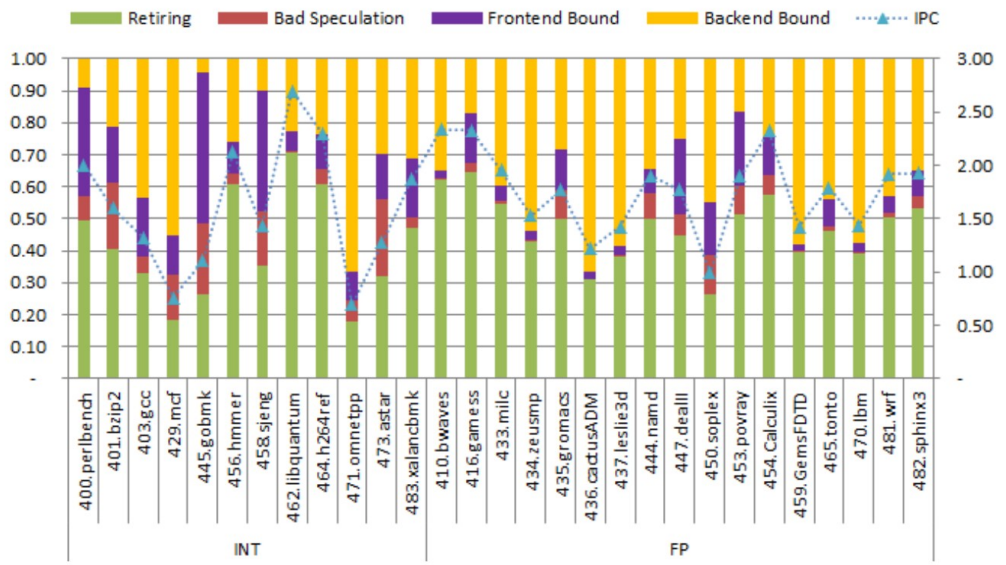


Figure 2: The Top-Down Analysis Hierarchy



(a) Top Level

Ahmad Yasin, “A Top-Down Method for Performance Analysis and Counters Architecture”

Thank you for listening !

Questions?

arnaud.fiorini@polymtl.ca