



CTF traces generation in ROCm and support for OTF2 in Trace Compass

Yoann Heitz
2021/06/04

Polytechnique Montreal
DORSAL Laboratory

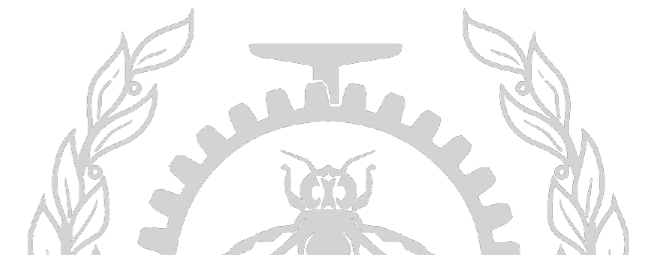
Agenda

CTF traces generation in ROCm :

- Context
- Solution
- Results

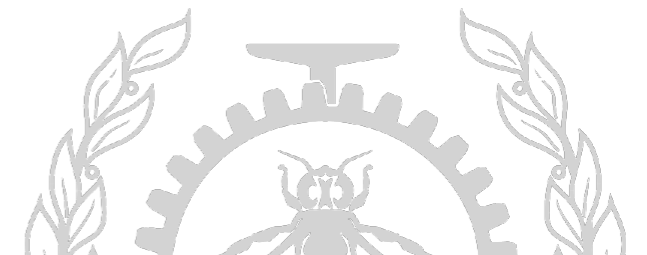
OTF2 support in Trace Compass :

- Background
- OTF2 to CTF converter
- MPI views in Trace Compass

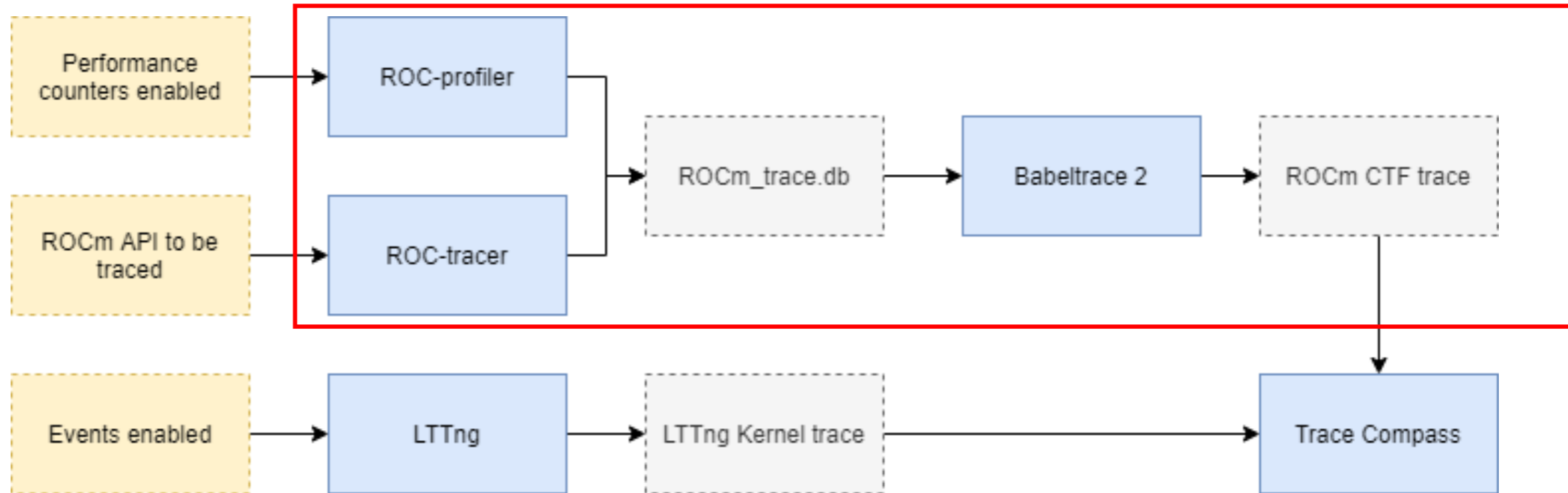


CTF traces generation in ROCm : Context

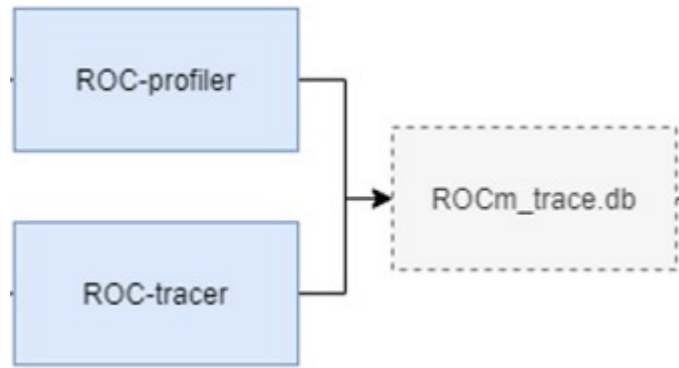
- ROCm : a platform for GPU computing
- ROC-profiler and ROC-tracer allow to trace several APIs (HIP, HSA, KFD) and performance metrics
- Arnaud developed analyzes in Trace Compass



CTF traces generation in ROCm : Context



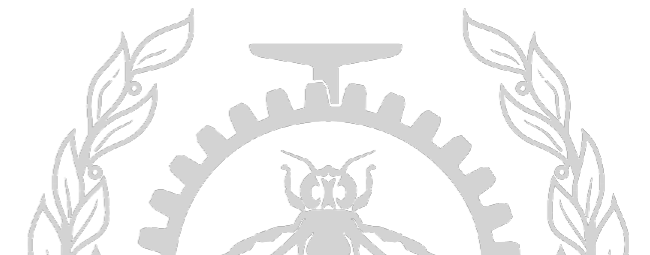
CTF traces generation in ROCm : Context



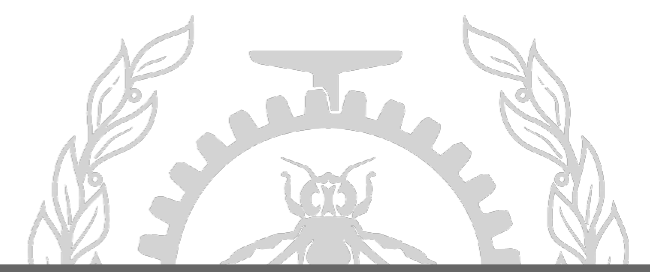
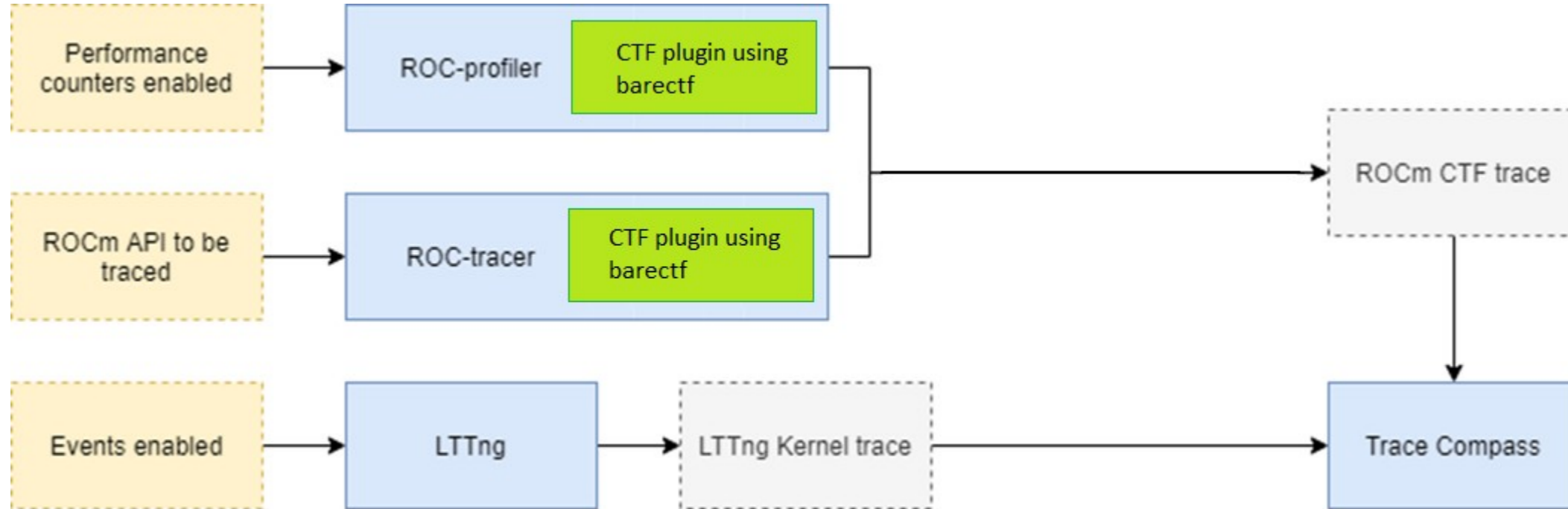
- Overhead due to tracing
- Intermediate text format
- Overhead due to conversion to .db



- Overhead due to conversion to CTF
- Plugin written in Python



CTF traces generation in ROCm : Solution



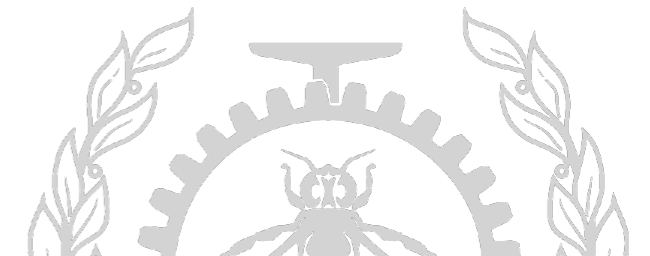
CTF traces generation in ROCm : Solution

Issues :

- events representing intervals
- unordered events
- ROC-profiler and ROC-tracer are multithreaded tools

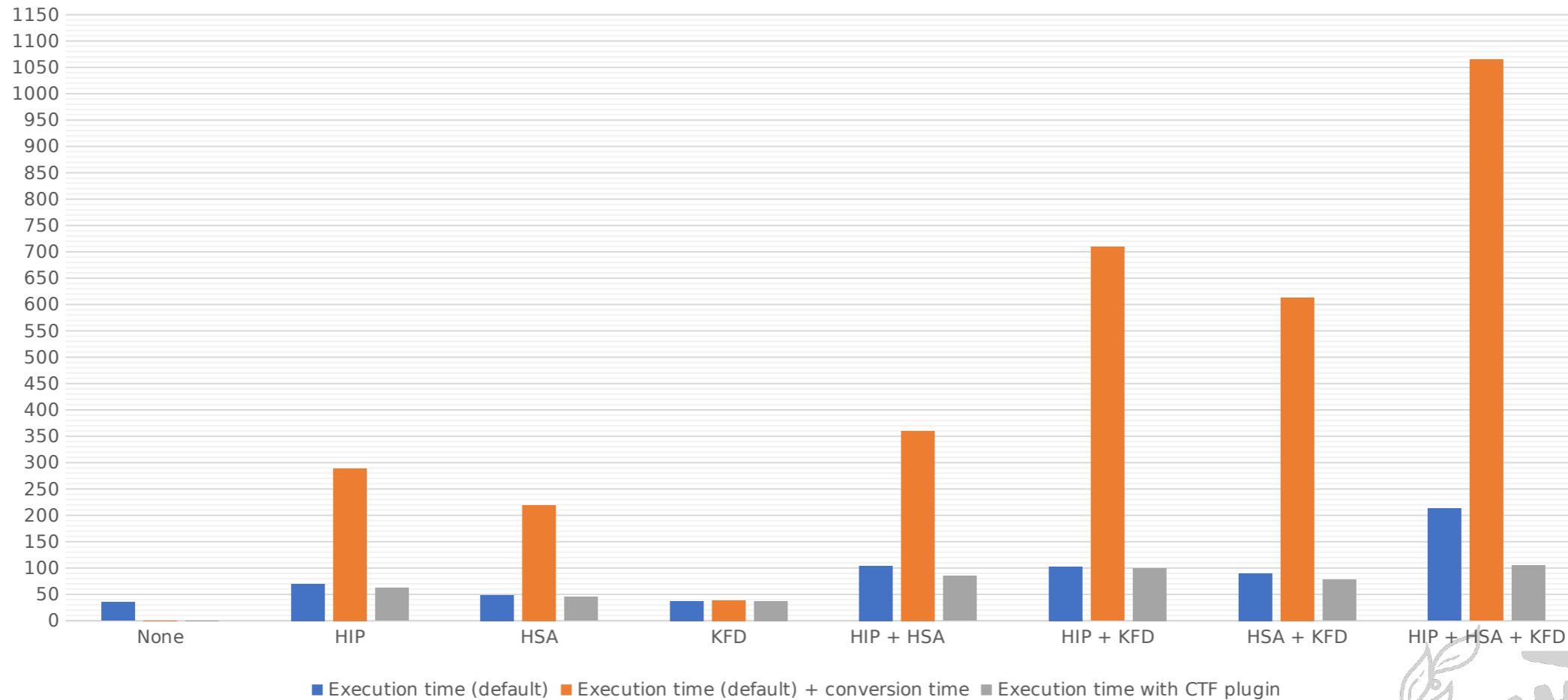
Solutions :

- 2 events for each interval
- reordering at runtime
- use of exclusive CTF streams and tracing structures

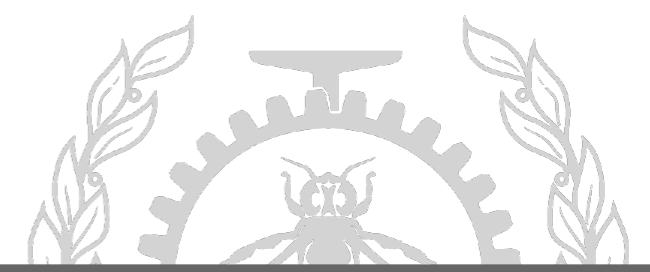
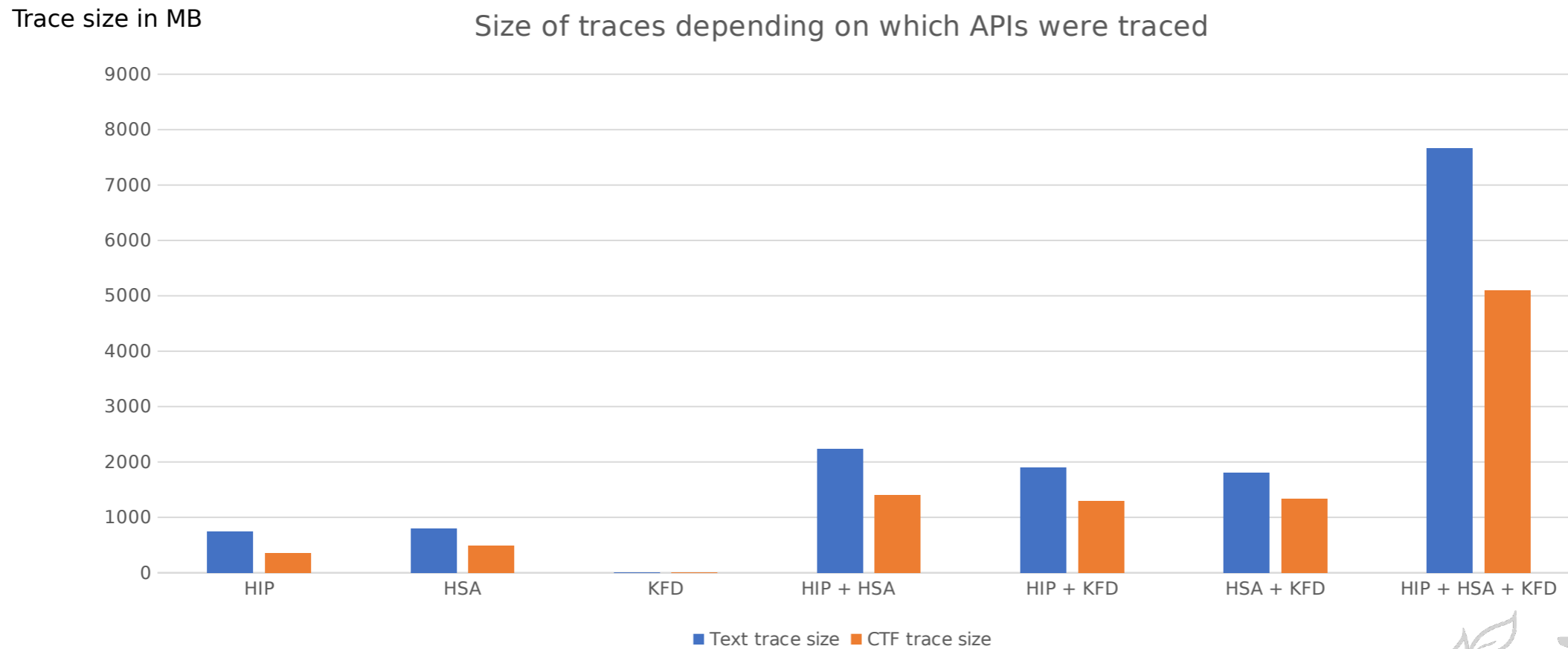


CTF traces generation in ROCm : Results

Time in seconds Execution and conversion (to .db and JSON) times depending on which APIs were traced



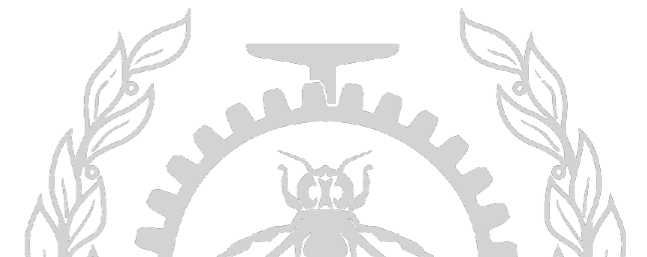
CTF traces generation in ROCm : Results



OTF2 support in Trace Compass : Background

OTF2 :

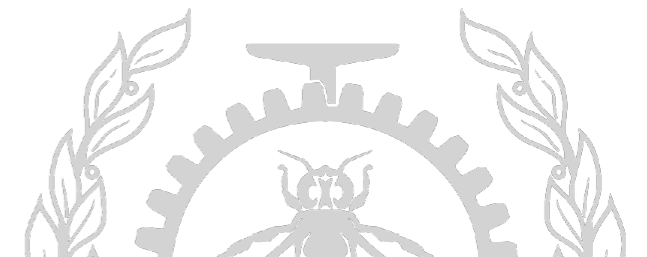
- Binary trace format
- C API
- MPI, OpenMP, Pthreads events



OTF2 support in Trace Compass : Background

Supporting a new trace type in Trace Compass :

- 1) Parse the trace
- 2) Convert events into the Trace Compass internal event format
- 3) Read the events and provide analysis



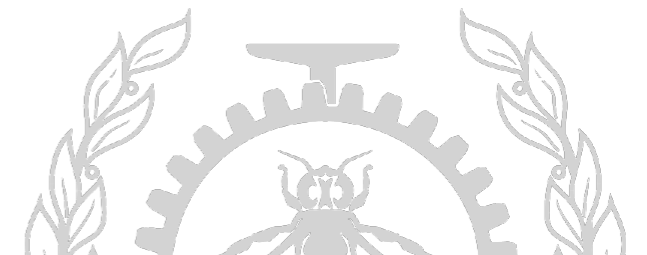
OTF2 support in Trace Compass : OTF2 to CTF converter

Parse and read the trace in Trace Compass:

- Convert it in CTF format
- Use the Trace Compass CTF parser

Tools for writing CTF traces :

- babeltrace 2 : write source plugin
- barectf : directly write the events in CTF streams



OTF2 support in Trace Compass : OTF2 to CTF converter

Pros and cons of the tools for the converter:

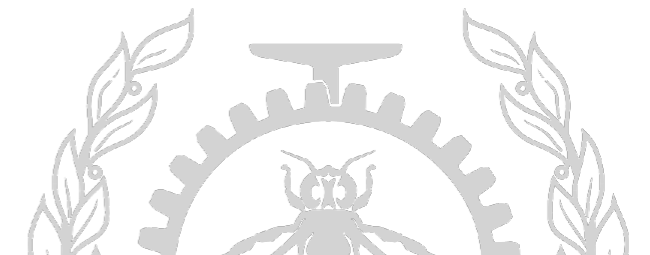
Babeltrace 2

Pros :

- Modularity
- Popular : wider use and maintenance for the converter

Cons :

- Monothread
- Intermediate conversion



OTF2 support in Trace Compass : OTF2 to CTF converter

Pros and cons of the tools for the converter:

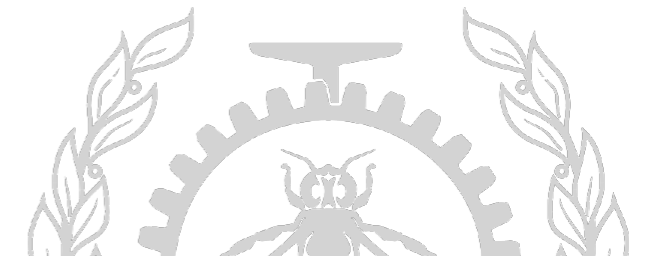
barectf

Pros :

- Possibility for multithreading
- Write directly to CTF

Cons :

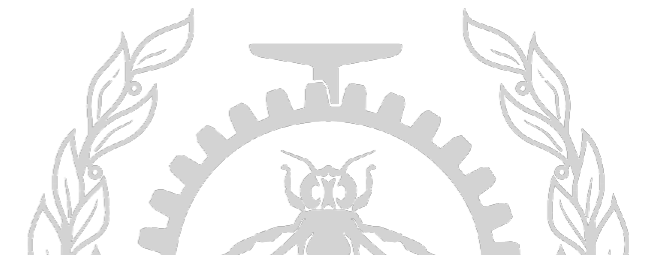
- No modularity



OTF2 support in Trace Compass : OTF2 to CTF converter

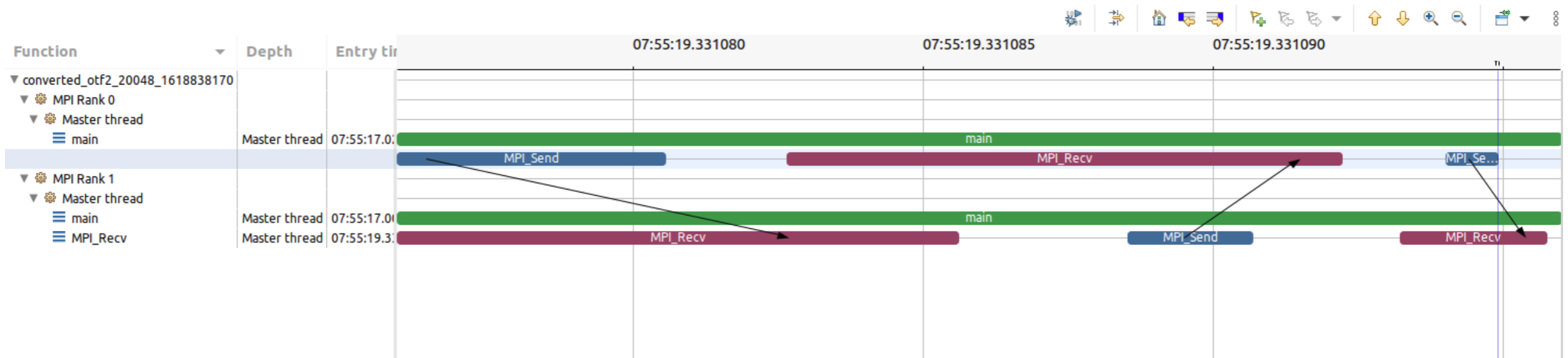
Babeltrace 2 / barectf conversion speed for a 1,1 GB trace :

Tool	Conversion time
Babeltrace 2 C plugin	40s
barectf : monothread	10s
barectf : multithread (5 threads)	2.5s



OTF2 support in Trace Compass : MPI views in Trace Compass

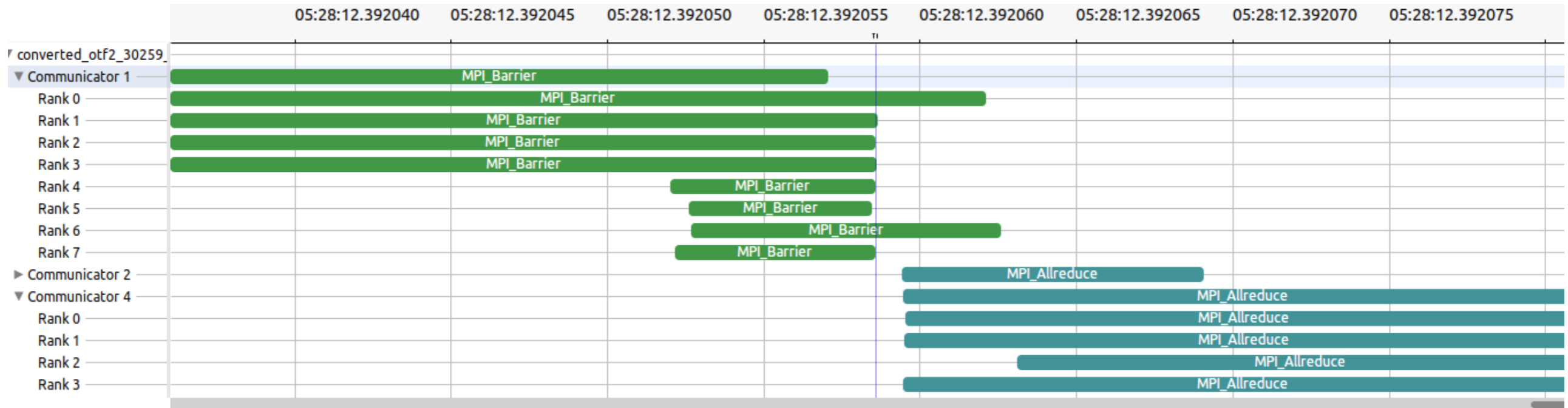
Prototypes : views for MPI programs instrumented with Score-p



Callstack analysis : Flame Chart view

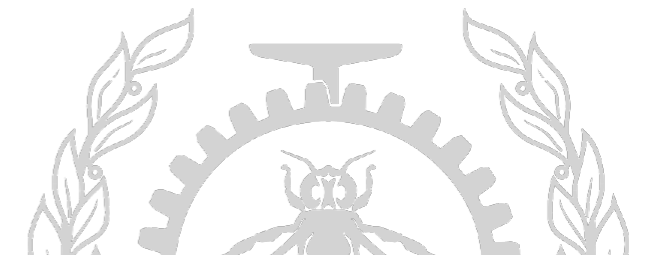


OTF2 support in Trace Compass : MPI views in Trace Compass

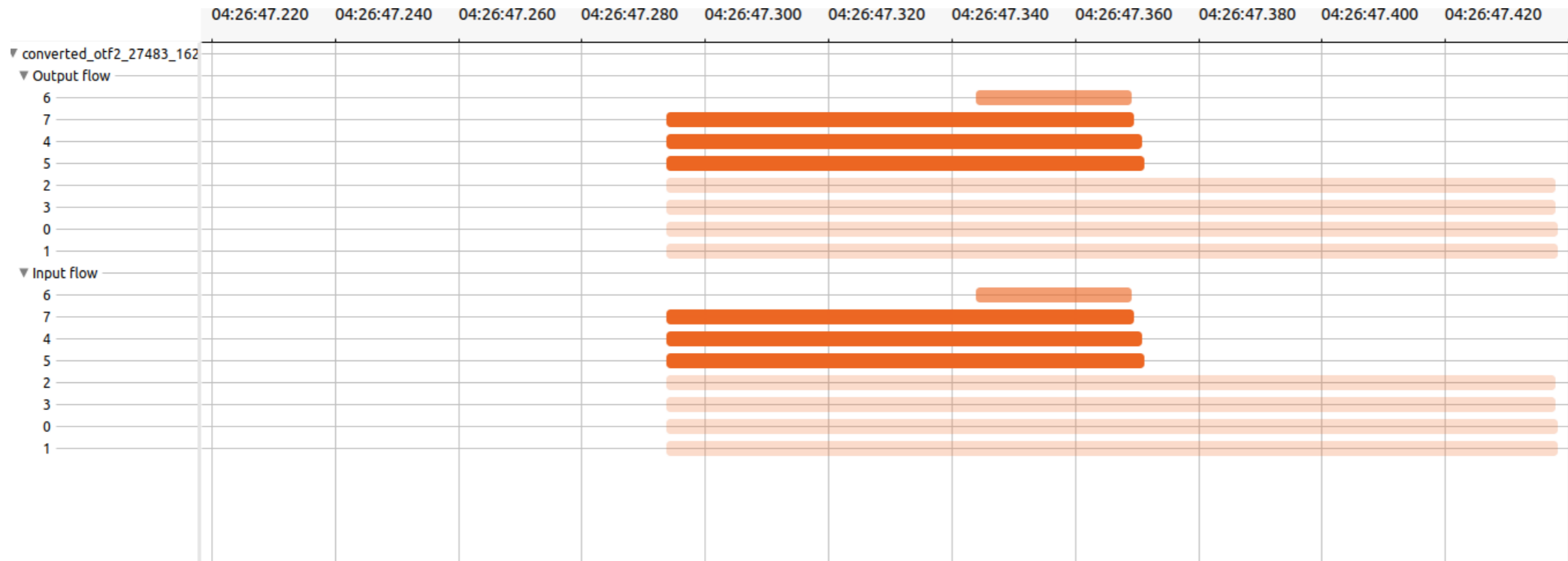


Prototype : MPI routines for each rank into a communicator

Work in progress : Manage overlapping collective communications in the same communicator



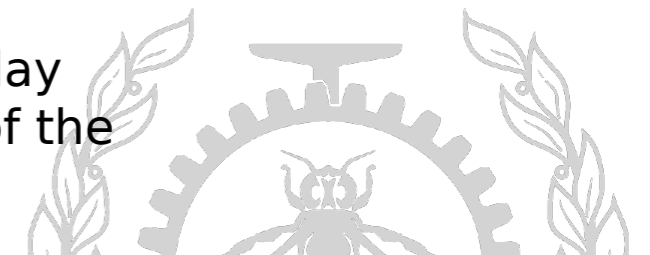
OTF2 support in Trace Compass : MPI views in Trace Compass



The opacity scales opposed to the flow

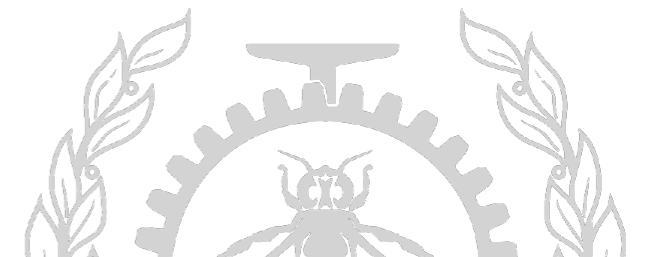
Prototype : Input/output flow of data for each OTF2 location

Work in progress : group locations by communicators and display more information about the flow and the state of the locations



Thank you for your
attention!

Any questions?



Annex :

Babeltrace 2 vs barectf benchmark :

- 1,1 GB OTF2 trace containing « EnterRegion » events equally distributed in 10 local events files (1e8 events)
- Babeltrace in production mode : time is doubled in average in developer mode
- « OTF2 » prototype plugin with source component based on « dust » example plugin : <https://babeltrace.org/docs/v2.0/libbabeltrace2/examples.html>
- Tests were executed 5 times on 2 different machines :
 - Machine 1 : 96 CPUs, max 2800MHz, 512G RAM, SSD
 - Machine 2 : 8 CPUs, max 3900MHz, 32GB RAM , HDD

Tool	Mean conversion time : Machine 1	Mean conversion time : Machine 2
Babeltrace 2 C plugin	40s	30s
barectf : monothread	10s	7,8s
barectf : multithread (5 threads)	2.5s	2s

