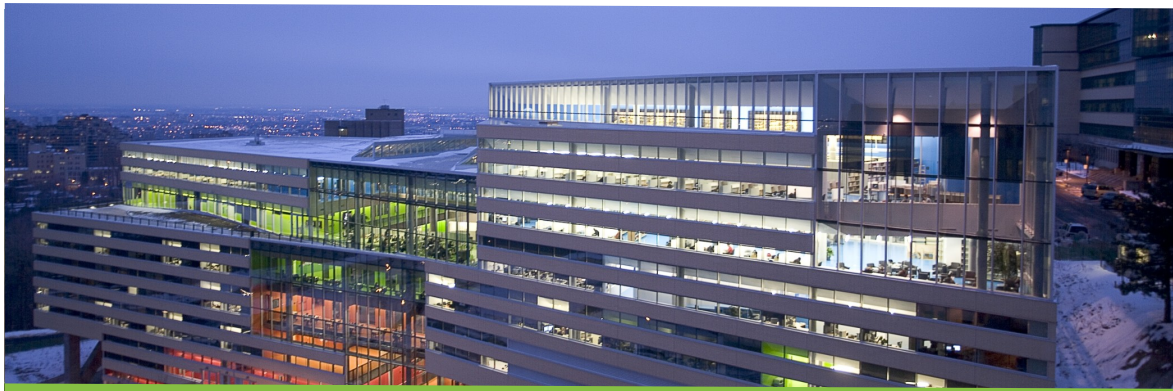


POLYTECHNIQUE MONTRÉAL

TECHNOLOGICAL UNIVERSITY

Improvements to the ROCm plugin in Trace Compass

Arnaud Fiorini

Laboratoire DORSAL

Introduction

- Specific hardware, environment
- Complex toolchain
- Traditional tools are hard to use

Agenda

1. Demonstrations
   - ROCm (ROC-profiler, ROC-tracer, ROCgdb)
   - Trace Compass
2. Scalability of Trace Analysis
   - Trace Compass improvements
   - Distributed trace analysis

ROCm

- AMD open source software
- Compute stack for headless system deployments
- Mainly interfaced with OpenCL, OpenMP and HIP
- HIP is a CUDA-like interface to schedule computations on GPUs

GPU Programming model

- Device code, also called kernel function

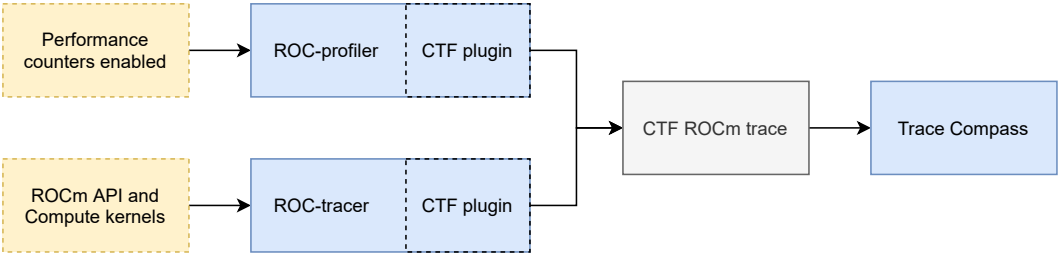
```
1 __global__ void helloworld(char* in, char* out)
2 {
3     int num = hipThreadIdx_x + hipBlockDim_x * hipBlockIdx_x;
4     out[num] = in[num] + 1;
5 }
```

GPU Programming model

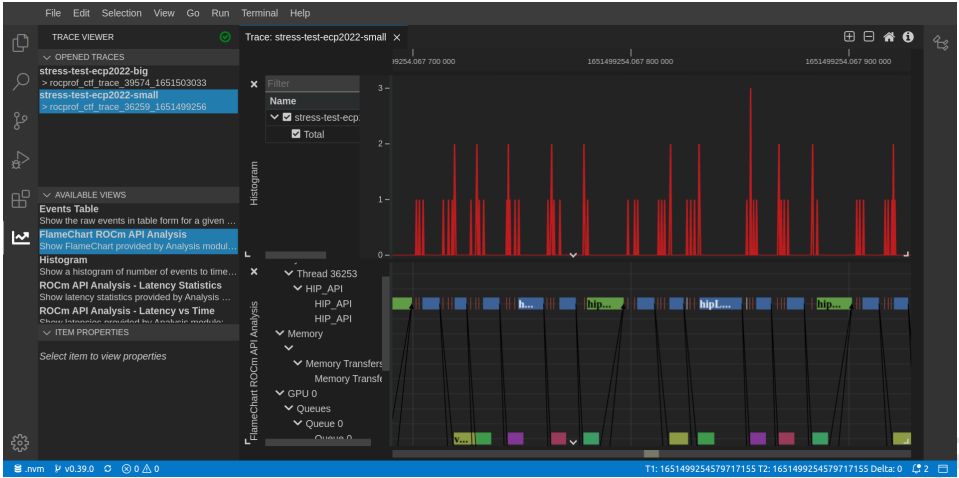
1. Allocating space on the device
2. Copy your memory over to the device
3. Perform a kernel launch with the kernel function
4. Copy back the result to the host
5. Free the space on the device

These steps are done by the user with the HIP API (CUDA-like API)

Tracing pipeline



Demo


File Edit Selection View Go Run Terminal Help
TRACE VIEWER
Trace: stress-test-ecp2022-small
OPENED TRACES
stress-test-ecp2022-big
> rocprof_ctf_trace_39574_1651503033
stress-test-ecp2022-small
> rocprof_ctf_trace_36259_1651499256
AVAILABLE VIEWS
Events Table
Show the raw events in table form for a given …
FlameChart ROCm API Analysis
Show FlameChart provided by Analysis modul…
Histogram
Show a histogram of number of events to time…
ROCm API Analysis - Latency Statistics
Show latency statistics provided by Analysis …
ROCm API Analysis - Latency vs Time
ITEM PROPERTIES
Select item to view properties
1651499254.067 700 000
1651499254.067 800 000
1651499254.067 900 000
Filter
Name
stress-test-ecp…
Total
Histogram
Thread 36253
HIP_API
HIP_API
HIP_API
Memory
Memory Transfers
Memory Transfe
GPU 0
Queues
Queue 0
FlameChart ROCm API Analysis
T1: 1651499254579717155 T2: 1651499254579717155 Delta: 0

Scalability

Current scalability improvements on Trace Compass[1]:

- Preprocessing to reduce the wait time
- Partial State System
- Distribute the analysis on multiple nodes
- Critical path on disk
- Multiple patches that improve performance

[1] Abdellah Rahmani, Quoc-Hao Tran, Geneviève Bastien, Matthew Khouzam

Future Work

- Implement Critical path for GPU and HPC workloads
- Performance metrics analysis
- Overview Analysis for large scale applications

Contributions

- Interfacing with ROCm to handle CPU and GPU traces and performance metrics
- Optimizing Trace Compass scalability to handle large traces
- UX improvements for HPC use cases