Introduction to OmniTools

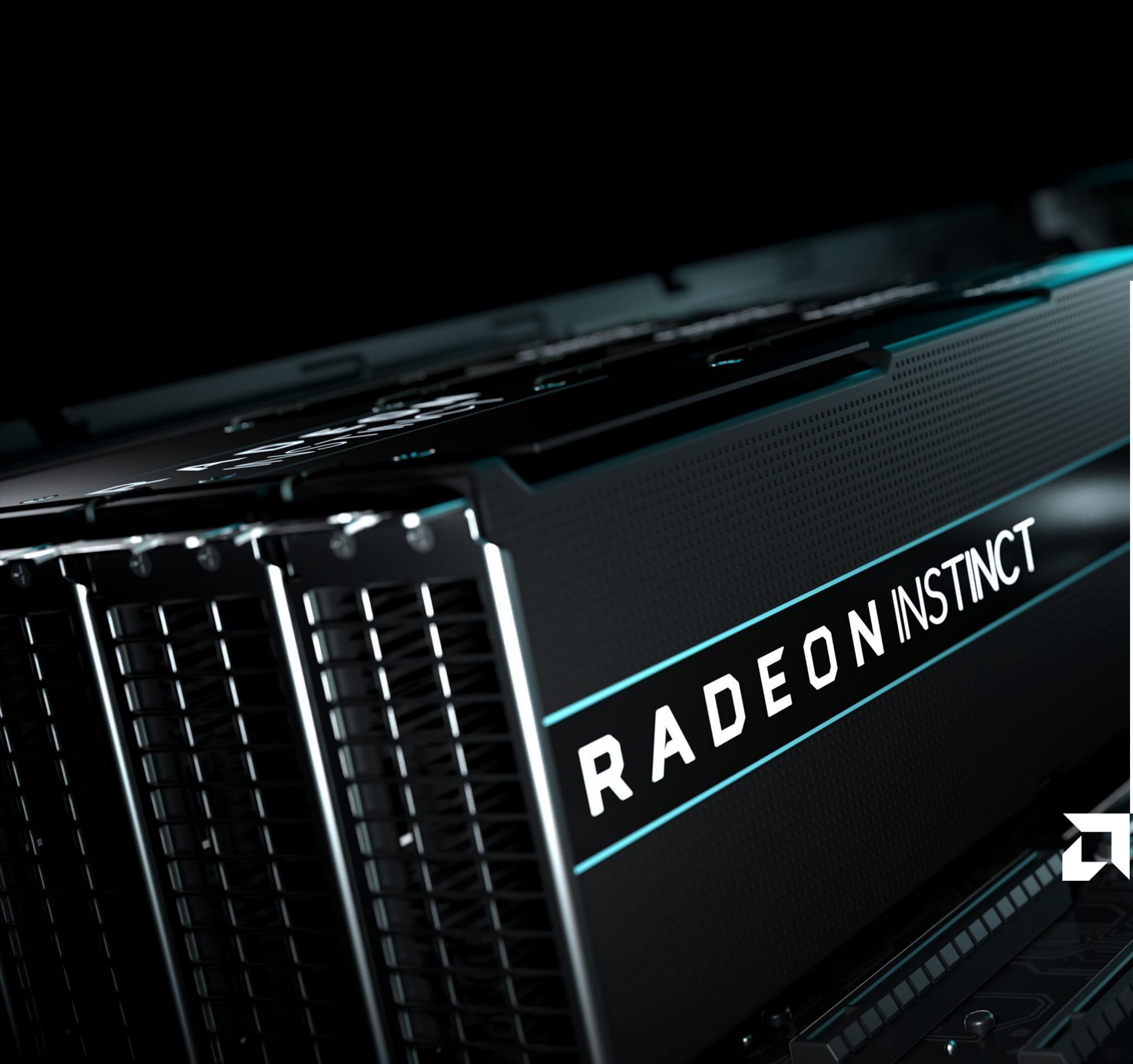
Suyash Tandon and George Markomanolis

LUMI 4-day training/HPE/AMD

Feb 17th, 2023

AMD

together we advance_



Profiling

Background – AMD Profilers

ROC-profiler (rocprof)

Hardware Counters

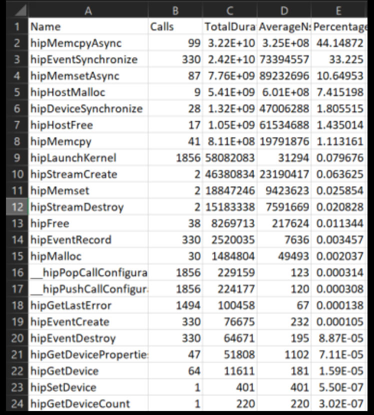
- Raw collection of GPU counters and traces
- Counter collection with user input files
- Counter results printed to a CSV

Traces and timelines

- Trace collection support for
- CPU copy
- HIP API
- HSA API
- GPU Kernels

Visualisation

- Traces visualized with Perfetto



	A	B	C	D	E
1	Name	Calls	TotalDura	AverageN:	Percentage
2	hipMemcpyAsync	99	3.22E+10	3.25E+08	44.14872
3	hipEventSynchronize	330	2.42E+10	73394557	33.225
4	hipMemsetAsync	87	7.76E+09	89232696	10.64953
5	hipHostMalloc	9	5.41E+09	6.01E+08	7.415198
6	hipDeviceSynchronize	28	1.32E+09	47006288	1.805515
7	hipHostFree	17	1.05E+09	61534688	1.435014
8	hipMemcpy	41	8.11E+08	19791876	1.113161
9	hipLaunchKernel	1856	58082083	31294	0.079676
10	hipStreamCreate	2	46380834	23190417	0.063625
11	hipMemset	2	18847246	9423623	0.025854
12	hipStreamDestroy	2	15183338	7591669	0.020828
13	hipFree	38	8269713	217624	0.011344
14	hipEventRecord	330	2520035	7636	0.003457
15	hipMalloc	30	1484804	49493	0.002037
16	__hipPopCallConfigura	1856	229159	123	0.000314
17	__hipPushCallConfigur	1856	224177	120	0.000308
18	hipGetLastError	1494	100458	67	0.000138
19	hipEventCreate	330	76675	232	0.000105
20	hipEventDestroy	330	64671	195	8.87E-05
21	hipGetDevicePropertie:	47	51808	1102	7.11E-05
22	hipGetDevice	64	11611	181	1.59E-05
23	hipSetDevice	1	401	401	5.50E-07
24	hipGetDeviceCount	1	220	220	3.02E-07

Omnitrace

Trace collection

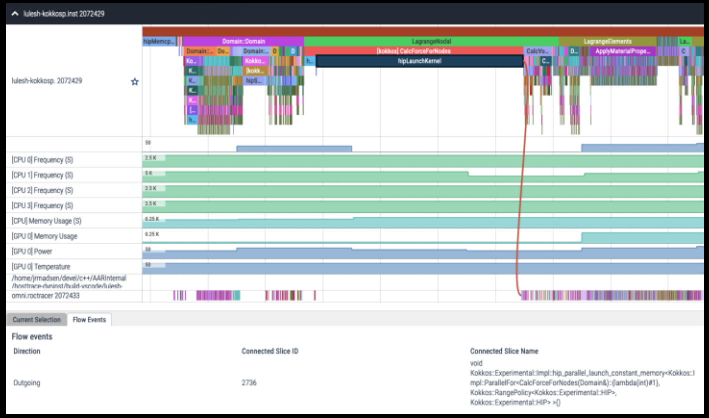
- Comprehensive trace collection:
- CPU
- GPU

Supports

- CPU copy
- HIP API
- HSA API
- GPU Kernels
- OpenMP®
- MPI
- Kokkos
- p-threads
- multi-GPU

Visualisation

- Traces visualized with Perfetto



Flow events

Direction	Connected Slice ID	Connected Slice Name
Outgoing	2736	void Kokkos::Experimental::Impl::hip_parallel_launch_constant_memory<Kokkos::Impl::ParallelFor<CalcForceForNodes(Domain&)::{lambda(int)#1}, Kokkos::RangePolicy<Kokkos::Experimental::HIP>, Kokkos::Experimental::HIP> >()

Omniperf

Performance Analysis

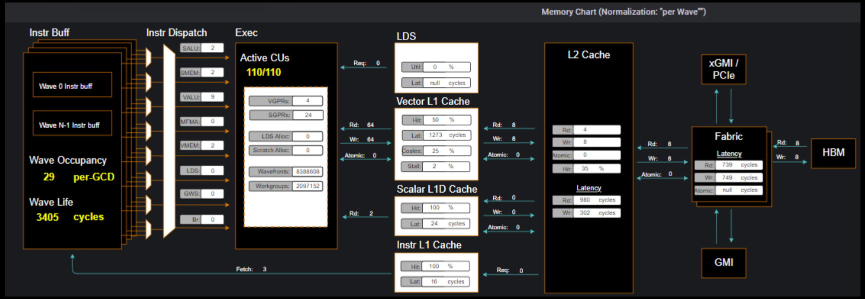
- Automated collection
- Analysis
- Visualisation

Supports

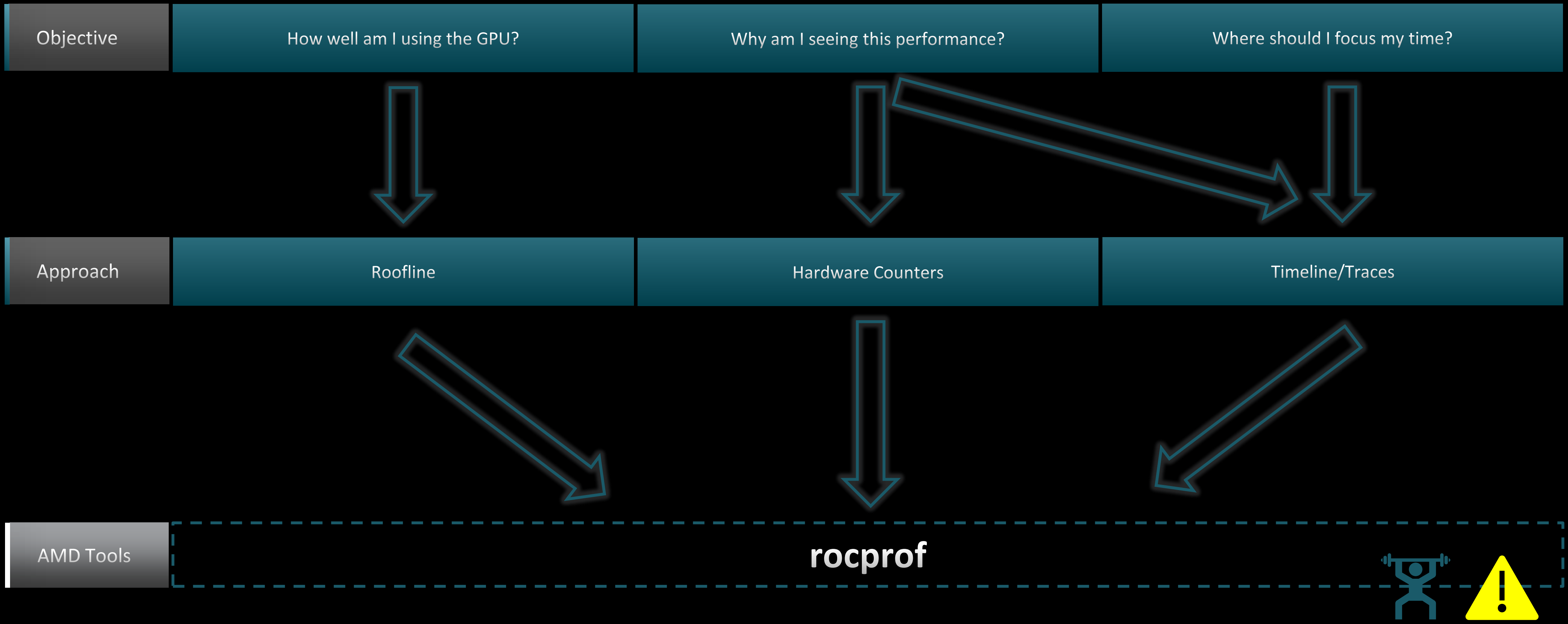
- Speed of Light
- Memory chart
- Rooflines
- Kernel comparison

Visualisation

- With Grafana or standalone GUI



Background – AMD Profilers



Objective

How well am I using the GPU?

Why am I seeing this performance?

Where should I focus my time?

Approach

Roofline

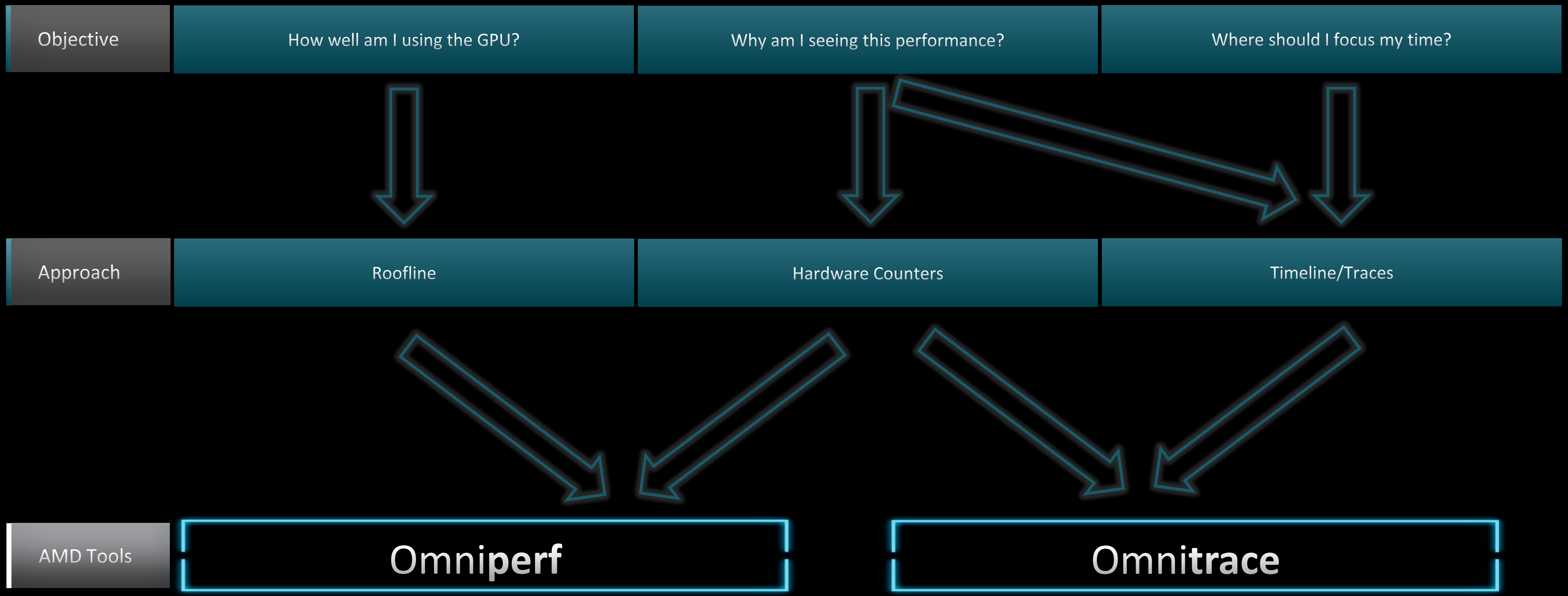
Hardware Counters

Timeline/Traces

AMD Tools

rocprof

Background – AMD Profilers



Objective

How well am I using the GPU?

Why am I seeing this performance?

Where should I focus my time?

Approach

Roofline

Hardware Counters

Timeline/Traces

AMD Tools

Omniperf

Omnitrace

Omnitrace

Omnitrace: Application Profiling, Tracing, and Analysis

AMD Research Tool	Repository: https://github.com/AMDResearch/omnitrace ✕ Not part of ROCm stack					
Language Support	C/C++	Fortran	Python	OpenCL ™		
Data Collection Modes	Dynamic instrumentation	Statistical/process sampling	Critical trace generation			
Data Analysis	High-level summary	Comprehensive trace	Critical trace analysis			
Parallelism Support	MPI	OpenMP®	Pthreads	HIP	HSA	Kokkos
GPU Metrics	HW counters	HSA API	HIP API	HIP trace	HSA trace	Memory & thermal
CPU Metrics	HW counters	Timing metrics	Memory access	Network	I/O	more…

Installation (if required)

To use pre-built binaries, select the version that matches your operating system, ROCm version, etc.

Select OpenSuse operating system for HPE/AMD system:

omnitrace-1.7.4-opensuse-15.4-ROCm-50400-PAPI-OMPT-Python3.sh

There are .rpm and .deb files for installation also

Full documentation: https://amdresearch.github.io/omnitrace/

```
wget https://github.com/AMDResearch/omnitrace/releases/download/v1.7.4/omnitrace-1.7.4-
opensuse-15.4-ROCm-50400-PAPI-OMPT-Python3.sh

mkdir /opt/omnitrace/
module load rocm        // not required if you build it on your laptop
chmod +x omnitrace-1.7.4-opensuse-15.4-ROCm-50400-PAPI-OMPT-Python3.sh
./omnitrace-1.7.4-opensuse-15.4-ROCm-50400-PAPI-OMPT-Python3.sh --prefix=/opt/omnitrace -
-exclude-subdir
export PATH=/opt/omnitrace/bin:$PATH
source omnitrace_installation_path/share/omnitrace/setup-env.sh
```

Omnitrace instrumentation modes

Runtime Instrumentation	Dynamic binary instrumentation		
	Characterize performance	Sample every invocation	Large overheads
Sampling Instrumentation	Periodic sampling of entire application		
	Statistical sampling	Process sampling	

Basic command-line syntax:

```
$ omnitrace [omnitrace-options] -- <CMD> <ARGS>
```

For more information or help use `-h/--help/?` flags:

```
$ omnitrace -h
```

Can also execute on systems using a job scheduler. For example, with SLURM, an interactive session can be used as

```
$ srun [options] omnitrace [omnitrace-options] -- <CMD> <ARGS>
```

For problems, create an issue here: https://github.com/AMDResearch/omnitrace/issues
Documentation: https://amdresearch.github.io/omnitrace/

Omnitrace configuration

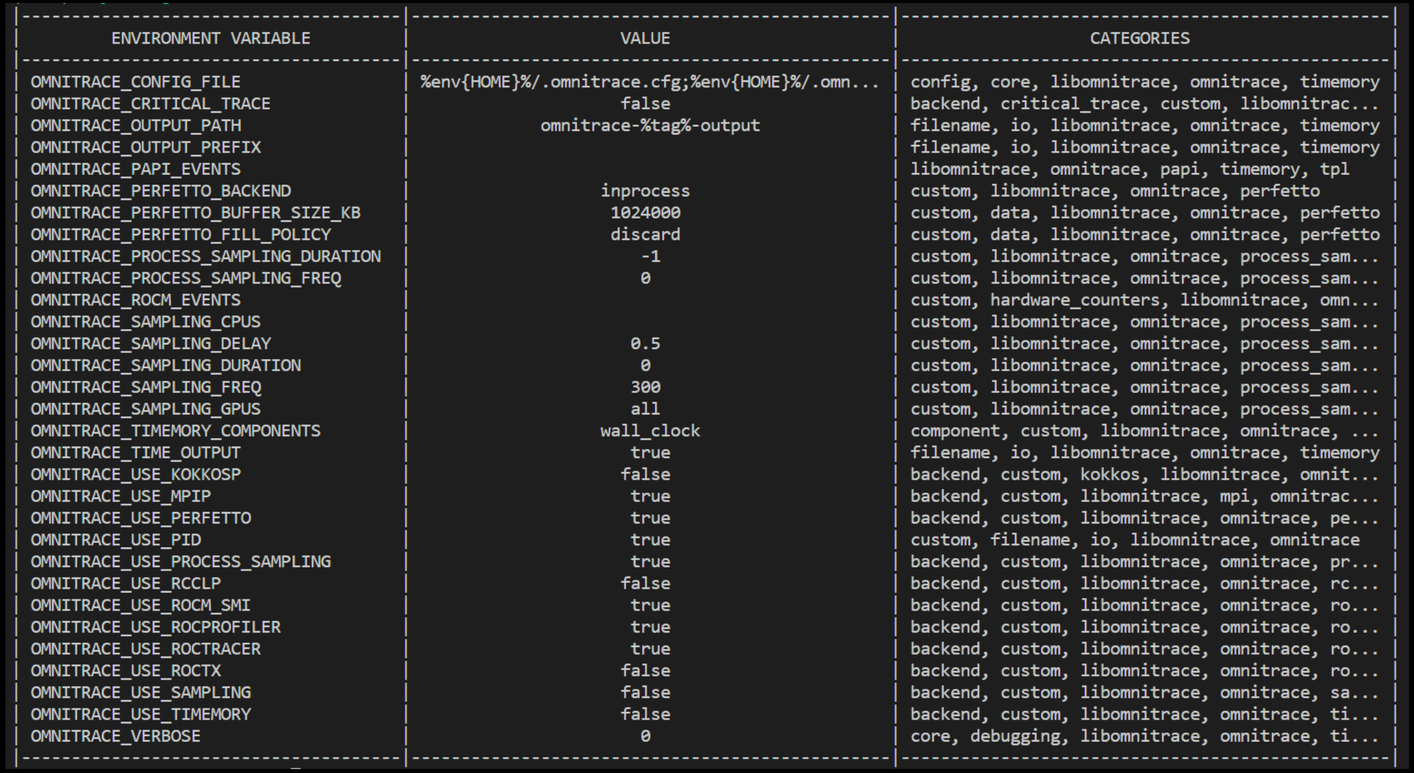
```
$ omnitrace-avail --categories [options]
```

Get more information about run-time settings, data collection capabilities, and available hardware counters. For more information or help use `-h/--help/?` flags:

```
$ omnitrace-avail -h
```

Collect information for omnitrace-related settings using shorthand `-c` for `--categories` :

```
$ omnitrace-avail -c omnitrace
```



```
|--------------------------------------|------------------------------------------------|------------------------------------------------|
|         ENVIRONMENT VARIABLE         |                     VALUE                      |                   CATEGORIES                   |
|--------------------------------------|------------------------------------------------|------------------------------------------------|
| OMNITRACE_CONFIG_FILE                | %env{HOME}%/.omnitrace.cfg;%env{HOME}%/.omn... | config, core, libomnitrace, omnitrace, timemory |
| OMNITRACE_CRITICAL_TRACE             |                     false                      | backend, critical_trace, custom, libomnitrac... |
| OMNITRACE_OUTPUT_PATH                |             omnitrace-%tag%-output             | filename, io, libomnitrace, omnitrace, timemory |
| OMNITRACE_OUTPUT_PREFIX              |                                                | filename, io, libomnitrace, omnitrace, timemory |
| OMNITRACE_PAPI_EVENTS                |                                                | libomnitrace, omnitrace, papi, timemory, tpl    |
| OMNITRACE_PERFETTO_BACKEND           |                   inprocess                    | custom, libomnitrace, omnitrace, perfetto       |
| OMNITRACE_PERFETTO_BUFFER_SIZE_KB    |                    1024000                     | custom, data, libomnitrace, omnitrace, perfetto |
| OMNITRACE_PERFETTO_FILL_POLICY       |                    discard                     | custom, data, libomnitrace, omnitrace, perfetto |
| OMNITRACE_PROCESS_SAMPLING_DURATION  |                       -1                       | custom, libomnitrace, omnitrace, process_sam... |
| OMNITRACE_PROCESS_SAMPLING_FREQ      |                       0                        | custom, libomnitrace, omnitrace, process_sam... |
| OMNITRACE_ROCM_EVENTS                |                                                | custom, hardware_counters, libomnitrace, omn... |
| OMNITRACE_SAMPLING_CPUS              |                                                | custom, libomnitrace, omnitrace, process_sam... |
| OMNITRACE_SAMPLING_DELAY             |                      0.5                       | custom, libomnitrace, omnitrace, process_sam... |
| OMNITRACE_SAMPLING_DURATION          |                       0                        | custom, libomnitrace, omnitrace, process_sam... |
| OMNITRACE_SAMPLING_FREQ              |                      300                       | custom, libomnitrace, omnitrace, process_sam... |
| OMNITRACE_SAMPLING_GPUS              |                      all                       | custom, libomnitrace, omnitrace, process_sam... |
| OMNITRACE_TIMEMORY_COMPONENTS        |                   wall_clock                   | component, custom, libomnitrace, omnitrace, ... |
| OMNITRACE_TIME_OUTPUT                |                      true                      | filename, io, libomnitrace, omnitrace, timemory |
| OMNITRACE_USE_KOKKOSP                |                     false                      | backend, custom, kokkos, libomnitrace, omnit... |
| OMNITRACE_USE_MPIP                   |                      true                      | backend, custom, libomnitrace, mpi, omnitrac... |
| OMNITRACE_USE_PERFETTO               |                      true                      | backend, custom, libomnitrace, omnitrace, pe... |
| OMNITRACE_USE_PID                    |                      true                      | custom, filename, io, libomnitrace, omnitrace   |
| OMNITRACE_USE_PROCESS_SAMPLING       |                      true                      | backend, custom, libomnitrace, omnitrace, pr... |
| OMNITRACE_USE_RCCLP                  |                     false                      | backend, custom, libomnitrace, omnitrace, rc... |
| OMNITRACE_USE_ROCM_SMI               |                      true                      | backend, custom, libomnitrace, omnitrace, ro... |
| OMNITRACE_USE_ROCPROFILER            |                      true                      | backend, custom, libomnitrace, omnitrace, ro... |
| OMNITRACE_USE_ROCTRACER              |                      true                      | backend, custom, libomnitrace, omnitrace, ro... |
| OMNITRACE_USE_ROCTX                  |                     false                      | backend, custom, libomnitrace, omnitrace, ro... |
| OMNITRACE_USE_SAMPLING               |                     false                      | backend, custom, libomnitrace, omnitrace, sa... |
| OMNITRACE_USE_TIMEMORY               |                     false                      | backend, custom, libomnitrace, omnitrace, ti... |
| OMNITRACE_VERBOSE                    |                       0                        | core, debugging, libomnitrace, omnitrace, ti... |
|--------------------------------------|------------------------------------------------|------------------------------------------------|
```

Omnitrace configuration

```
$ omnitrace-avail --categories [options]
```

Get more information about run-time settings, data collection capabilities, and available hardware counters. For more information or help use `-h/--help/?` flags:

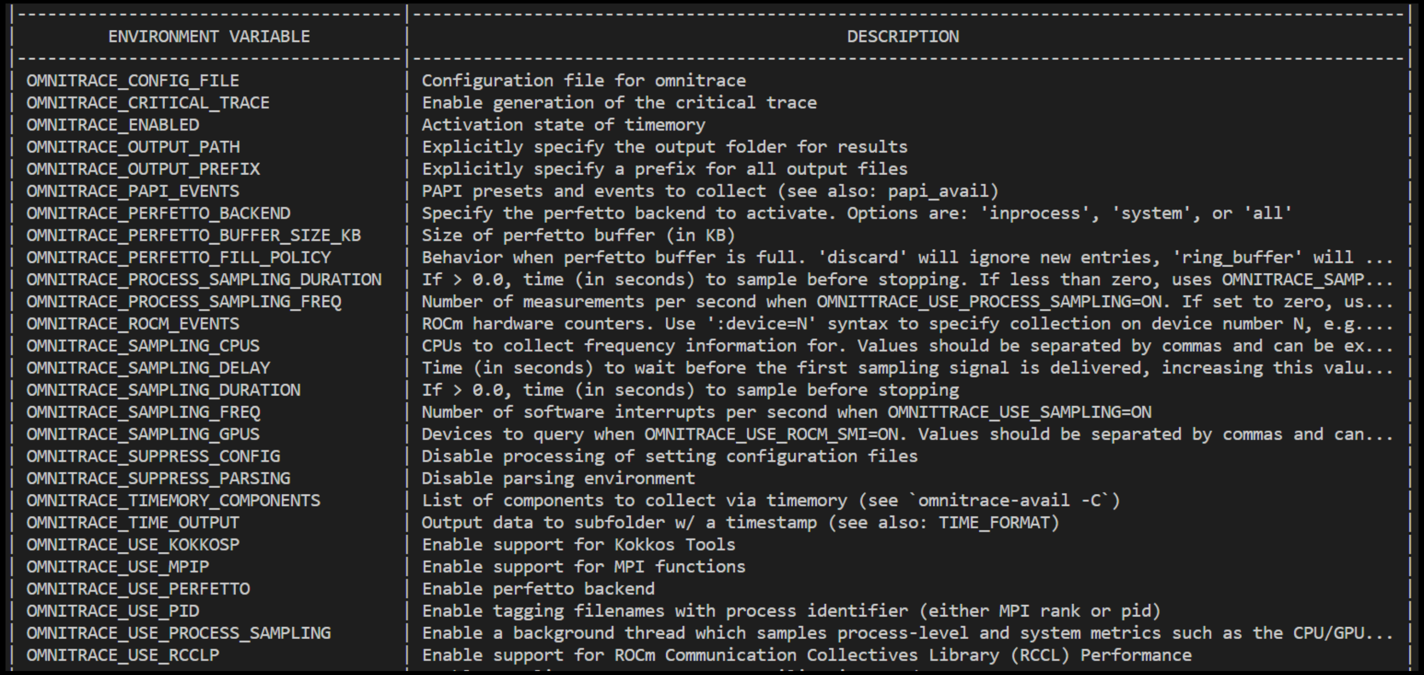
```
$ omnitrace-avail -h
```

Collect information for omnitrace-related settings using shorthand `-c` for `--categories` :

```
$ omnitrace-avail -c omnitrace
```

For brief description, use the options:

```
$ omnitrace-avail -bd
```



```
|---------------------------------------|--------------------------------------------------------------------------------------------------------|
|         ENVIRONMENT VARIABLE          |                                              DESCRIPTION                                               |
|---------------------------------------|--------------------------------------------------------------------------------------------------------|
| OMNITRACE_CONFIG_FILE                 | Configuration file for omnitrace                                                                       |
| OMNITRACE_CRITICAL_TRACE              | Enable generation of the critical trace                                                                |
| OMNITRACE_ENABLED                     | Activation state of timemory                                                                           |
| OMNITRACE_OUTPUT_PATH                 | Explicitly specify the output folder for results                                                       |
| OMNITRACE_OUTPUT_PREFIX               | Explicitly specify a prefix for all output files                                                       |
| OMNITRACE_PAPI_EVENTS                 | PAPI presets and events to collect (see also: papi_avail)                                              |
| OMNITRACE_PERFETTO_BACKEND            | Specify the perfetto backend to activate. Options are: 'inprocess', 'system', or 'all'                 |
| OMNITRACE_PERFETTO_BUFFER_SIZE_KB     | Size of perfetto buffer (in KB)                                                                        |
| OMNITRACE_PERFETTO_FILL_POLICY        | Behavior when perfetto buffer is full. 'discard' will ignore new entries, 'ring_buffer' will ...       |
| OMNITRACE_PROCESS_SAMPLING_DURATION   | If > 0.0, time (in seconds) to sample before stopping. If less than zero, uses OMNITRACE_SAMP...       |
| OMNITRACE_PROCESS_SAMPLING_FREQ       | Number of measurements per second when OMNITTRACE_USE_PROCESS_SAMPLING=ON. If set to zero, us...       |
| OMNITRACE_ROCM_EVENTS                 | ROCm hardware counters. Use ':device=N' syntax to specify collection on device number N, e.g....       |
| OMNITRACE_SAMPLING_CPUS               | CPUs to collect frequency information for. Values should be separated by commas and can be ex...       |
| OMNITRACE_SAMPLING_DELAY              | Time (in seconds) to wait before the first sampling signal is delivered, increasing this valu...       |
| OMNITRACE_SAMPLING_DURATION           | If > 0.0, time (in seconds) to sample before stopping                                                  |
| OMNITRACE_SAMPLING_FREQ               | Number of software interrupts per second when OMNITTRACE_USE_SAMPLING=ON                               |
| OMNITRACE_SAMPLING_GPUS               | Devices to query when OMNITRACE_USE_ROCM_SMI=ON. Values should be separated by commas and can...       |
| OMNITRACE_SUPPRESS_CONFIG             | Disable processing of setting configuration files                                                      |
| OMNITRACE_SUPPRESS_PARSING            | Disable parsing environment                                                                            |
| OMNITRACE_TIMEMORY_COMPONENTS         | List of components to collect via timemory (see `omnitrace-avail -C`)                                  |
| OMNITRACE_TIME_OUTPUT                 | Output data to subfolder w/ a timestamp (see also: TIME_FORMAT)                                        |
| OMNITRACE_USE_KOKKOSP                 | Enable support for Kokkos Tools                                                                        |
| OMNITRACE_USE_MPIP                    | Enable support for MPI functions                                                                       |
| OMNITRACE_USE_PERFETTO                | Enable perfetto backend                                                                                |
| OMNITRACE_USE_PID                     | Enable tagging filenames with process identifier (either MPI rank or pid)                              |
| OMNITRACE_USE_PROCESS_SAMPLING        | Enable a background thread which samples process-level and system metrics such as the CPU/GPU...       |
| OMNITRACE_USE_RCCLP                   | Enable support for ROCm Communication Collectives Library (RCCL) Performance                            |
```

Create a config file

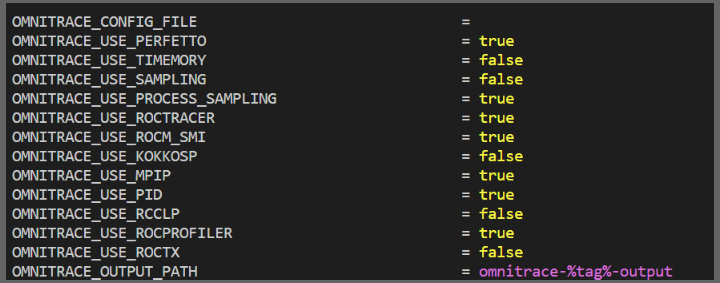
Create a config file in `$HOME`:

```
$ omnitrace-avail -G $HOME/.omnitrace.cfg
```

To add description of all variables and settings, use:

```
$ omnitrace-avail -G $HOME/.omnitrace.cfg --all
```

Modify the config file $HOME/.omnitrace.cfg as desired to enable and change settings:



```
OMNITRACE_CONFIG_FILE                                   =
OMNITRACE_USE_PERFETTO                                  = true
OMNITRACE_USE_TIMEMORY                                  = false
OMNITRACE_USE_SAMPLING                                  = false
OMNITRACE_USE_PROCESS_SAMPLING                          = true
OMNITRACE_USE_ROCTRACER                                 = true
OMNITRACE_USE_ROCM_SMI                                  = true
OMNITRACE_USE_KOKKOSP                                   = false
OMNITRACE_USE_MPIP                                      = true
OMNITRACE_USE_PID                                       = true
OMNITRACE_USE_RCCLP                                     = false
OMNITRACE_USE_ROCPROFILER                               = true
OMNITRACE_USE_ROCTX                                     = false
OMNITRACE_OUTPUT_PATH                                   = omnitrace-%tag%-output
```

Declare which config file to use by setting the environment:

```
$ export OMNITRACE_CONFIG_FILE=path-to/.omnitrace.cfg
```

Executing MatrixTranspose

Get example from: https://github.com/ROCm-Developer-Tools/HIP/tree/develop/samples/2_Cookbook/0_MatrixTranspose/MatrixTranspose.cpp

Requires a ROCm stack, and can be easily compiled with command:

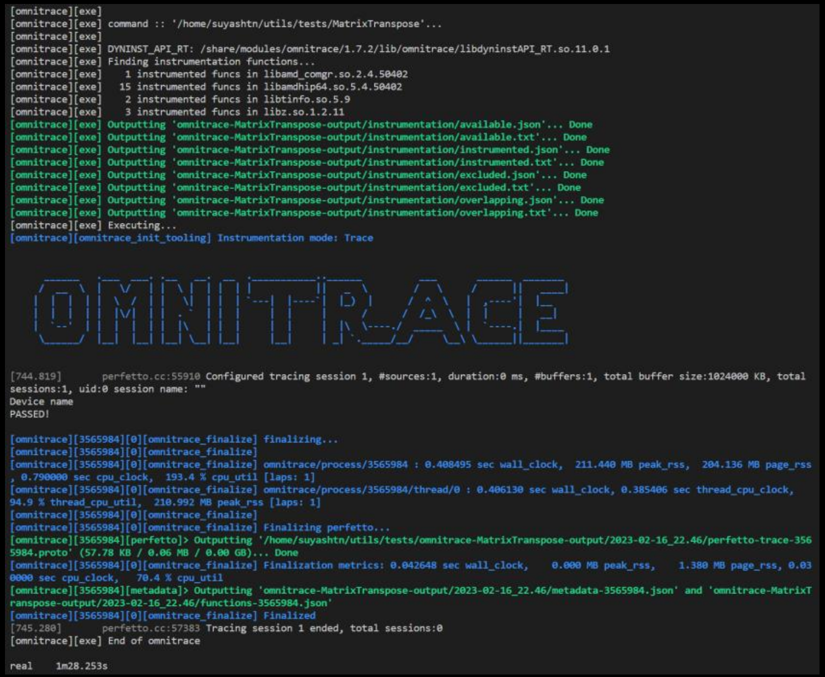
```
$ hipcc --offload-arch=gfx90a -o MatrixTranspose MatrixTranspose.cpp
```

Run the non-instrumented code on a single GPU as:

```
$ time ./MatrixTranspose

real     0m1.245s
```

Dynamic instrumentation

```
$ time omnitrace -- ./MatrixTranspose
real      1m28.253s
```



```
[omnitrace][exe]
[omnitrace][exe] command :: '/home/suyashtn/utils/tests/MatrixTranspose'...
[omnitrace][exe]
[omnitrace][exe] DYNINST_API_RT: /share/modules/omnitrace/1.7.2/lib/omnitrace/libdyninstAPI_RT.so.11.0.1
[omnitrace][exe] Finding instrumentation functions...
[omnitrace][exe]    1 instrumented funcs in libamd_comgr.so.2.4.50402
[omnitrace][exe]   15 instrumented funcs in libamdhip64.so.5.4.50402
[omnitrace][exe]    2 instrumented funcs in libtinfo.so.5.9
[omnitrace][exe]    3 instrumented funcs in libz.so.1.2.11
[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/available.json'... Done
[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/available.txt'... Done
[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/instrumented.json'... Done
[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/instrumented.txt'... Done
[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/excluded.json'... Done
[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/excluded.txt'... Done
[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/overlapping.json'... Done
[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/overlapping.txt'... Done
[omnitrace][exe] Executing...
[omnitrace][omnitrace_init_tooling] Instrumentation mode: Trace


[744.819]       perfetto.cc:55910 Configured tracing session 1, #sources:1, duration:0 ms, #buffers:1, total buffer size:1024000 KB, total
sessions:1, uid:0 session name: ""
Device name
PASSED!

[omnitrace][3565984][0][omnitrace_finalize] finalizing...
[omnitrace][3565984][0][omnitrace_finalize]
[omnitrace][3565984][0][omnitrace_finalize] omnitrace/process/3565984 : 0.408495 sec wall_clock,  211.440 MB peak_rss,  204.136 MB page_rss
, 0.790000 sec cpu_clock,  193.4 % cpu_util [laps: 1]
[omnitrace][3565984][0][omnitrace_finalize] omnitrace/process/3565984/thread/0 : 0.406130 sec wall_clock, 0.385406 sec thread_cpu_clock,
94.9 % thread_cpu_util,  210.992 MB peak_rss [laps: 1]
[omnitrace][3565984][0][omnitrace_finalize]
[omnitrace][3565984][0][omnitrace_finalize] Finalizing perfetto...
[omnitrace][3565984][perfetto]> Outputting '/home/suyashtn/utils/tests/omnitrace-MatrixTranspose-output/2023-02-16_22.46/perfetto-trace-356
5984.proto' (57.78 KB / 0.06 MB / 0.00 GB)... Done
[omnitrace][3565984][0][omnitrace_finalize] Finalization metrics: 0.042648 sec wall_clock,    0.000 MB peak_rss,    1.380 MB page_rss, 0.03
0000 sec cpu_clock,   70.4 % cpu_util
[omnitrace][3565984][metadata]> Outputting 'omnitrace-MatrixTranspose-output/2023-02-16_22.46/metadata-3565984.json' and 'omnitrace-MatrixT
ranspose-output/2023-02-16_22.46/functions-3565984.json'
[omnitrace][3565984][0][omnitrace_finalize] Finalized
[745.280]       perfetto.cc:57383 Tracing session 1 ended, total sessions:0
[omnitrace][exe] End of omnitrace

real    1m28.253s
```

Executing MatrixTranspose

Get example from: https://github.com/ROCm-Developer-Tools/HIP/tree/develop/samples/2_Cookbook/0_MatrixTranspose/MatrixTranspose.cpp

Requires a ROCm stack, and can be easily compiled with command:

```
$ hipcc --offload-arch=gfx90a -o MatrixTranspose MatrixTranspose.cpp
```

Run the non-instrumented code on a single GPU as:

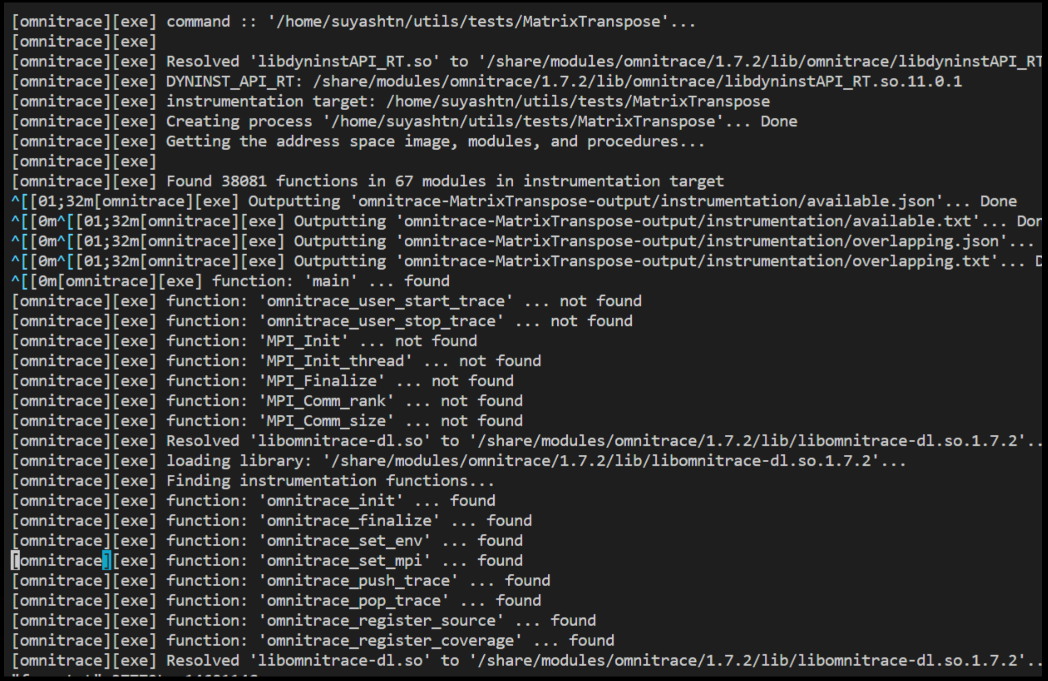
```
$ time ./MatrixTranspose

real    0m1.245s
```

Dynamic instrumentation

```
$ time omnitrace -- ./MatrixTranspose
real     1m28.253s
```

Available functions to instrument:

```
$ omnitrace -v -1 --simulate --print-available functions -- ./MatrixTranspose
```



```
[omnitrace][exe] command :: '/home/suyashtn/utils/tests/MatrixTranspose'...
[omnitrace][exe]
[omnitrace][exe] Resolved 'libdyninstAPI_RT.so' to '/share/modules/omnitrace/1.7.2/lib/omnitrace/libdyninstAPI_RT
[omnitrace][exe] DYNINST_API_RT: /share/modules/omnitrace/1.7.2/lib/omnitrace/libdyninstAPI_RT.so.11.0.1
[omnitrace][exe] instrumentation target: /home/suyashtn/utils/tests/MatrixTranspose
[omnitrace][exe] Creating process '/home/suyashtn/utils/tests/MatrixTranspose'... Done
[omnitrace][exe] Getting the address space image, modules, and procedures...
[omnitrace][exe]
[omnitrace][exe] Found 38081 functions in 67 modules in instrumentation target
^[[01;32m[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/available.json'... Done
^[[0m^[[01;32m[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/available.txt'... Dor
^[[0m^[[01;32m[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/overlapping.json'...
^[[0m^[[01;32m[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/overlapping.txt'... D
^[[0m[omnitrace][exe] function: 'main' ... found
[omnitrace][exe] function: 'omnitrace_user_start_trace' ... not found
[omnitrace][exe] function: 'omnitrace_user_stop_trace' ... not found
[omnitrace][exe] function: 'MPI_Init' ... not found
[omnitrace][exe] function: 'MPI_Init_thread' ... not found
[omnitrace][exe] function: 'MPI_Finalize' ... not found
[omnitrace][exe] function: 'MPI_Comm_rank' ... not found
[omnitrace][exe] function: 'MPI_Comm_size' ... not found
[omnitrace][exe] Resolved 'libomnitrace-dl.so' to '/share/modules/omnitrace/1.7.2/lib/libomnitrace-dl.so.1.7.2'..
[omnitrace][exe] loading library: '/share/modules/omnitrace/1.7.2/lib/libomnitrace-dl.so.1.7.2'...
[omnitrace][exe] Finding instrumentation functions...
[omnitrace][exe] function: 'omnitrace_init' ... found
[omnitrace][exe] function: 'omnitrace_finalize' ... found
[omnitrace][exe] function: 'omnitrace_set_env' ... found
[omnitrace][exe] function: 'omnitrace_set_mpi' ... found
[omnitrace][exe] function: 'omnitrace_push_trace' ... found
[omnitrace][exe] function: 'omnitrace_pop_trace' ... found
[omnitrace][exe] function: 'omnitrace_register_source' ... found
[omnitrace][exe] function: 'omnitrace_register_coverage' ... found
[omnitrace][exe] Resolved 'libomnitrace-dl.so' to '/share/modules/omnitrace/1.7.2/lib/libomnitrace-dl.so.1.7.2'..
```

Executing MatrixTranspose

Get example from: https://github.com/ROCm-Developer-Tools/HIP/tree/develop/samples/2_Cookbook/0_MatrixTranspose/MatrixTranspose.cpp

Requires a ROCm stack, and can be easily compiled with command:

```
$ hipcc --offload-arch=gfx90a -o MatrixTranspose MatrixTranspose.cpp
```

Run the non-instrumented code on a single GPU as:

```
$ time ./MatrixTranspose

real    0m1.245s
```

Dynamic instrumentation

```
$ time omnitrace -- ./MatrixTranspose
real      1m28.253s
```

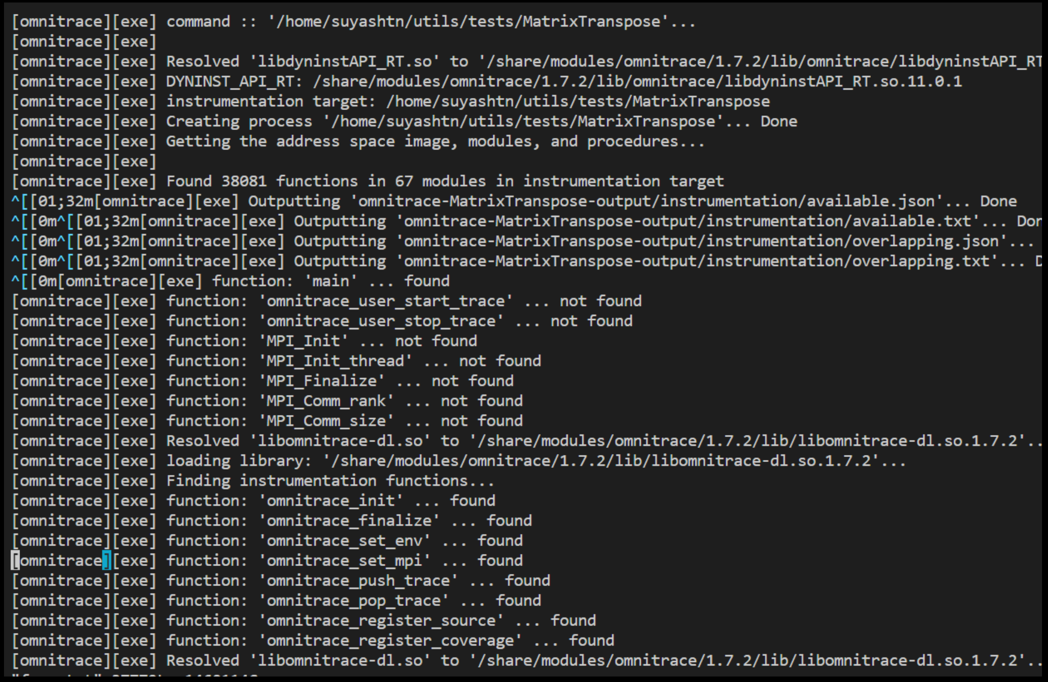
Available functions* to instrument:

```
$ omnitrace -v -1 --simulate --print-available
functions -- ./MatrixTranspose
```

Custom include/exclude functions* with –I or –E, resp. For e.g:

```
$ omnitrace -v -1 --simulate --print-available
functions -I 'function_name1' 'function_name2' --
./MatrixTranspose
```

*The simulate flag does not run the executable, but only demonstrates the available functions



```
[omnitrace][exe] command :: '/home/suyashtn/utils/tests/MatrixTranspose'...
[omnitrace][exe]
[omnitrace][exe] Resolved 'libdyninstAPI_RT.so' to '/share/modules/omnitrace/1.7.2/lib/omnitrace/libdyninstAPI_RT
[omnitrace][exe] DYNINST_API_RT: /share/modules/omnitrace/1.7.2/lib/omnitrace/libdyninstAPI_RT.so.11.0.1
[omnitrace][exe] instrumentation target: /home/suyashtn/utils/tests/MatrixTranspose
[omnitrace][exe] Creating process '/home/suyashtn/utils/tests/MatrixTranspose'... Done
[omnitrace][exe] Getting the address space image, modules, and procedures...
[omnitrace][exe]
[omnitrace][exe] Found 38081 functions in 67 modules in instrumentation target
^[[01;32m[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/available.json'... Done
^[[0m^[[01;32m[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/available.txt'... Dor
^[[0m^[[01;32m[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/overlapping.json'...
^[[0m^[[01;32m[omnitrace][exe] Outputting 'omnitrace-MatrixTranspose-output/instrumentation/overlapping.txt'... D
^[[0m[omnitrace][exe] function: 'main' ... found
[omnitrace][exe] function: 'omnitrace_user_start_trace' ... not found
[omnitrace][exe] function: 'omnitrace_user_stop_trace' ... not found
[omnitrace][exe] function: 'MPI_Init' ... not found
[omnitrace][exe] function: 'MPI_Init_thread' ... not found
[omnitrace][exe] function: 'MPI_Finalize' ... not found
[omnitrace][exe] function: 'MPI_Comm_rank' ... not found
[omnitrace][exe] function: 'MPI_Comm_size' ... not found
[omnitrace][exe] Resolved 'libomnitrace-dl.so' to '/share/modules/omnitrace/1.7.2/lib/libomnitrace-dl.so.1.7.2'..
[omnitrace][exe] loading library: '/share/modules/omnitrace/1.7.2/lib/libomnitrace-dl.so.1.7.2'...
[omnitrace][exe] Finding instrumentation functions...
[omnitrace][exe] function: 'omnitrace_init' ... found
[omnitrace][exe] function: 'omnitrace_finalize' ... found
[omnitrace][exe] function: 'omnitrace_set_env' ... found
[omnitrace][exe] function: 'omnitrace_set_mpi' ... found
[omnitrace][exe] function: 'omnitrace_push_trace' ... found
[omnitrace][exe] function: 'omnitrace_pop_trace' ... found
[omnitrace][exe] function: 'omnitrace_register_source' ... found
[omnitrace][exe] function: 'omnitrace_register_coverage' ... found
[omnitrace][exe] Resolved 'libomnitrace-dl.so' to '/share/modules/omnitrace/1.7.2/lib/libomnitrace-dl.so.1.7.2'..
```

Decreasing the profiling overhead

Binary re-write

```
$ omnitrace [omnitrace-options] -o <new-name-of-exec> -- <CMD> <ARGS>
```

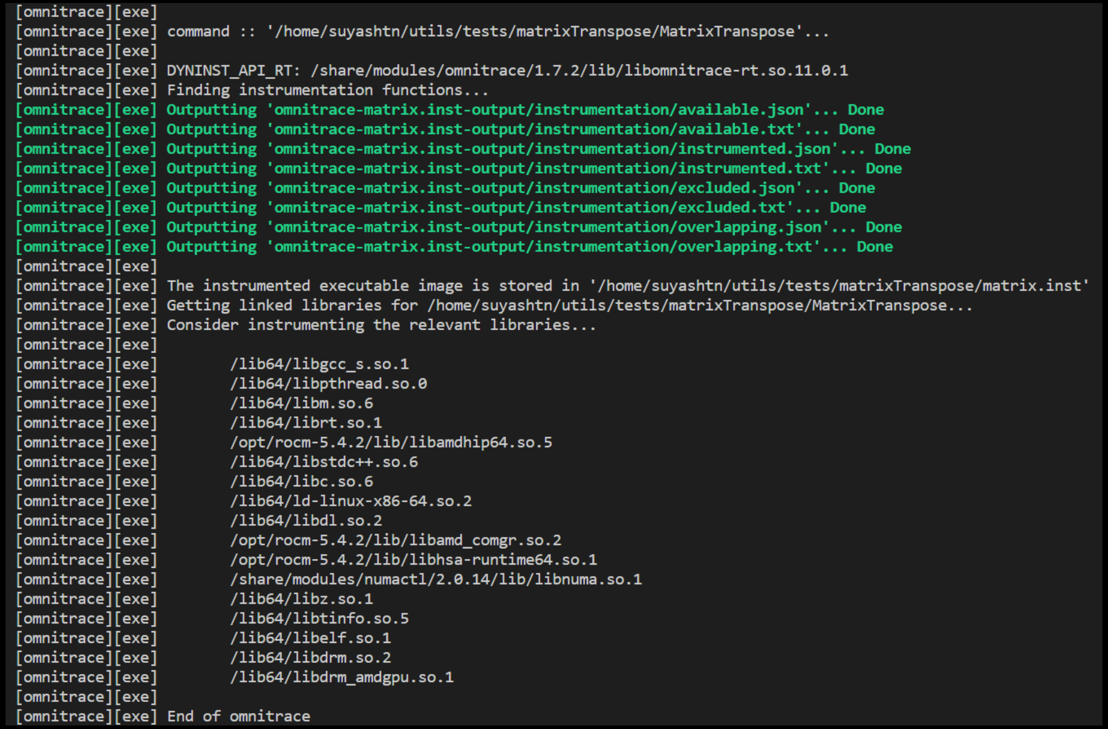
Generating a new executable/library with instrumentation built-in. For example:

```
$ omnitrace -o matrix.inst -- ./MatrixTranspose
```

subroutine instrumentation

Default instrumentation is main function and functions of 1024 instructions and more (for CPU)

To instrument routines with for example 50 instructions, add the option "–i 50" to instrument function of 50 instructions and above (move overhead)



```
[omnitrace][exe]
[omnitrace][exe] command :: '/home/suyashtn/utils/tests/matrixTranspose/MatrixTranspose'...
[omnitrace][exe]
[omnitrace][exe] DYNINST_API_RT: /share/modules/omnitrace/1.7.2/lib/libomnitrace-rt.so.11.0.1
[omnitrace][exe] Finding instrumentation functions...
[omnitrace][exe] Outputting 'omnitrace-matrix.inst-output/instrumentation/available.json'... Done
[omnitrace][exe] Outputting 'omnitrace-matrix.inst-output/instrumentation/available.txt'... Done
[omnitrace][exe] Outputting 'omnitrace-matrix.inst-output/instrumentation/instrumented.json'... Done
[omnitrace][exe] Outputting 'omnitrace-matrix.inst-output/instrumentation/instrumented.txt'... Done
[omnitrace][exe] Outputting 'omnitrace-matrix.inst-output/instrumentation/excluded.json'... Done
[omnitrace][exe] Outputting 'omnitrace-matrix.inst-output/instrumentation/excluded.txt'... Done
[omnitrace][exe] Outputting 'omnitrace-matrix.inst-output/instrumentation/overlapping.json'... Done
[omnitrace][exe] Outputting 'omnitrace-matrix.inst-output/instrumentation/overlapping.txt'... Done
[omnitrace][exe]
[omnitrace][exe] The instrumented executable image is stored in '/home/suyashtn/utils/tests/matrixTranspose/matrix.inst'
[omnitrace][exe] Getting linked libraries for /home/suyashtn/utils/tests/matrixTranspose/MatrixTranspose...
[omnitrace][exe] Consider instrumenting the relevant libraries...
[omnitrace][exe]
[omnitrace][exe]        /lib64/libgcc_s.so.1
[omnitrace][exe]        /lib64/libpthread.so.0
[omnitrace][exe]        /lib64/libm.so.6
[omnitrace][exe]        /lib64/librt.so.1
[omnitrace][exe]        /opt/rocm-5.4.2/lib/libamdhip64.so.5
[omnitrace][exe]        /lib64/libstdc++.so.6
[omnitrace][exe]        /lib64/libc.so.6
[omnitrace][exe]        /lib64/ld-linux-x86-64.so.2
[omnitrace][exe]        /lib64/libdl.so.2
[omnitrace][exe]        /opt/rocm-5.4.2/lib/libamd_comgr.so.2
[omnitrace][exe]        /opt/rocm-5.4.2/lib/libhsa-runtime64.so.1
[omnitrace][exe]        /share/modules/numactl/2.0.14/lib/libnuma.so.1
[omnitrace][exe]        /lib64/libz.so.1
[omnitrace][exe]        /lib64/libtinfo.so.5
[omnitrace][exe]        /lib64/libelf.so.1
[omnitrace][exe]        /lib64/libdrm.so.2
[omnitrace][exe]        /lib64/libdrm_amdgpu.so.1
[omnitrace][exe]
[omnitrace][exe] End of omnitrace
```

Decreasing the profiling overhead

Binary re-write

```
$ omnitrace [omnitrace-options] -o <new-name-of-exec> -- <CMD> <ARGS>
```

Generating a new executable/library with instrumentation built-in. For example:

```
$ omnitrace -o matrix.inst -- ./MatrixTranspose
```

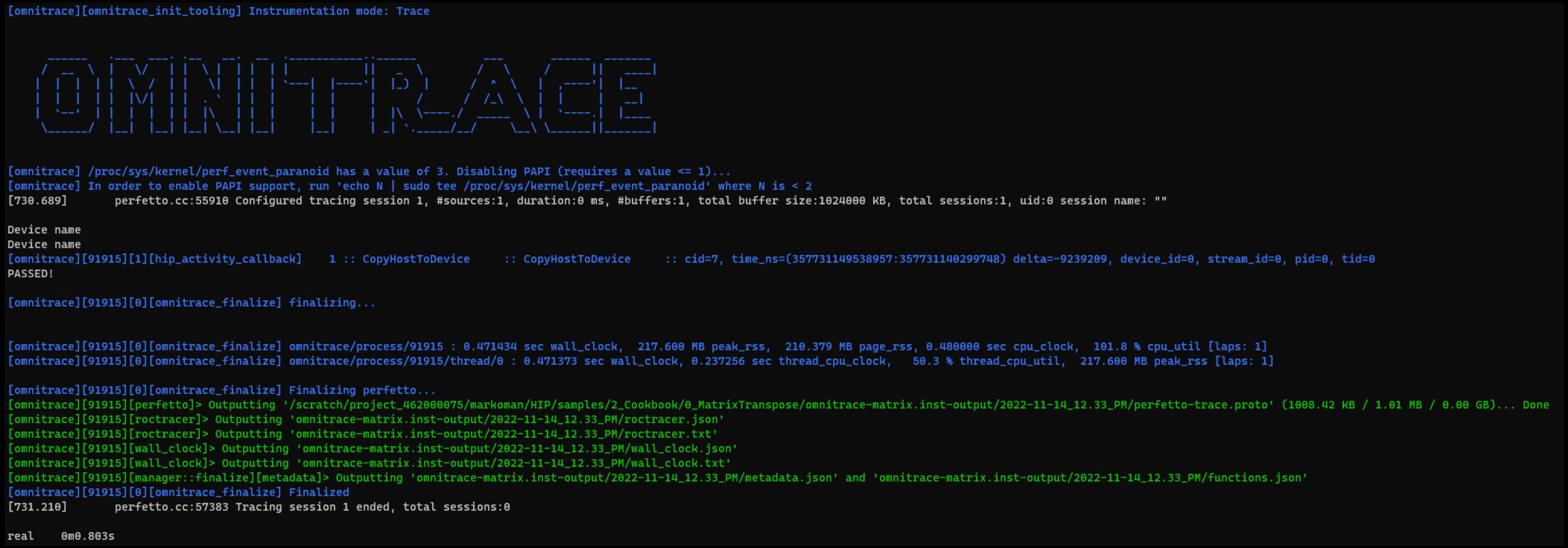
Run the instrumented binary on a single GPU as:

```
$ time ./matrix.inst

real     0m0.727s
```

subroutine instrumentation

Default instrumentation is main function and functions of 1024 instructions and more (for CPU)

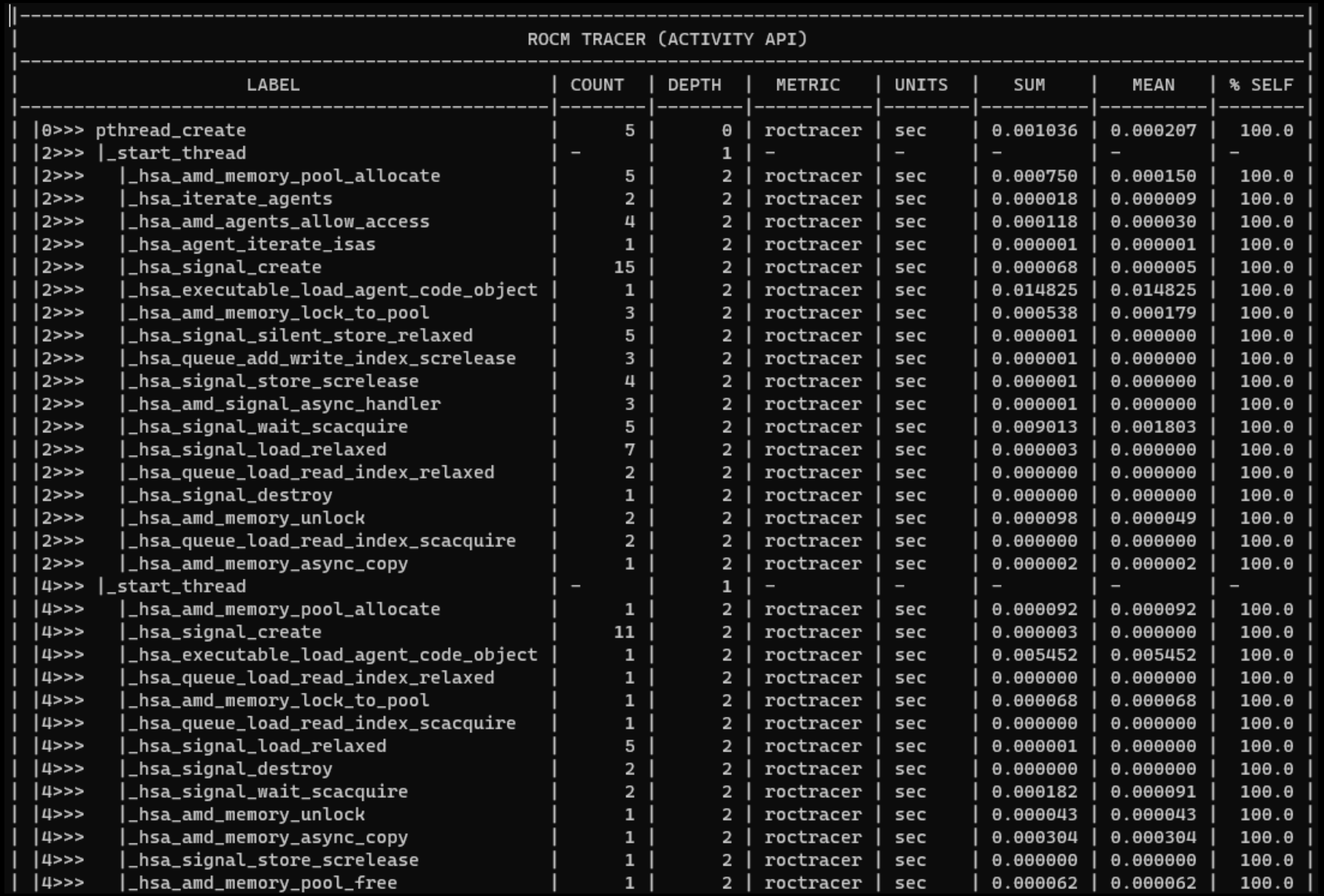
To instrument routines with for example 50 instructions, add the option "–i 50" to instrument function of 50 instructions and above (move overhead)



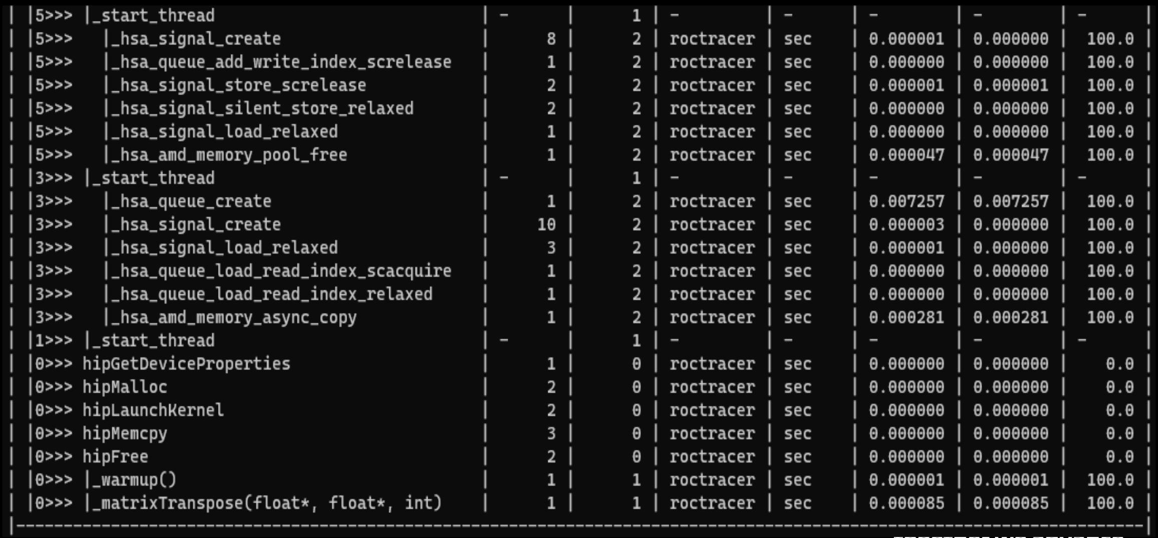
```
[omnitrace][omnitrace_init_tooling] Instrumentation mode: Trace

[omnitrace] /proc/sys/kernel/perf_event_paranoid has a value of 3. Disabling PAPI (requires a value <= 1)...
[omnitrace] In order to enable PAPI support, run 'echo N | sudo tee /proc/sys/kernel/perf_event_paranoid' where N is < 2
[730.689]       perfetto.cc:55910 Configured tracing session 1, #sources:1, duration:0 ms, #buffers:1, total buffer size:1024000 KB, total sessions:1, uid:0 session name: ""

Device name
Device name
[omnitrace][91915][1][hip_activity_callback]    1 :: CopyHostToDevice     :: CopyHostToDevice     :: cid=7, time_ns=(357731149538957:357731140299748) delta=-9239209, device_id=0, stream_id=0, pid=0, tid=0
PASSED!

[omnitrace][91915][0][omnitrace_finalize] finalizing...


[omnitrace][91915][0][omnitrace_finalize] omnitrace/process/91915 : 0.471434 sec wall_clock,  217.600 MB peak_rss,  210.379 MB page_rss, 0.480000 sec cpu_clock,  101.8 % cpu_util [laps: 1]
[omnitrace][91915][0][omnitrace_finalize] omnitrace/process/91915/thread/0 : 0.471373 sec wall_clock, 0.237256 sec thread_cpu_clock,   50.3 % thread_cpu_util,  217.600 MB peak_rss [laps: 1]

[omnitrace][91915][0][omnitrace_finalize] Finalizing perfetto...
[omnitrace][91915][perfetto]> Outputting '/scratch/project_462000075/markoman/HIP/samples/2_Cookbook/0_MatrixTranspose/omnitrace-matrix.inst-output/2022-11-14_12.33_PM/perfetto-trace.proto' (1008.42 KB / 1.01 MB / 0.00 GB)... Done
[omnitrace][91915][roctracer]> Outputting 'omnitrace-matrix.inst-output/2022-11-14_12.33_PM/roctracer.json'
[omnitrace][91915][roctracer]> Outputting 'omnitrace-matrix.inst-output/2022-11-14_12.33_PM/roctracer.txt'
[omnitrace][91915][wall_clock]> Outputting 'omnitrace-matrix.inst-output/2022-11-14_12.33_PM/wall_clock.json'
[omnitrace][91915][wall_clock]> Outputting 'omnitrace-matrix.inst-output/2022-11-14_12.33_PM/wall_clock.txt'
[omnitrace][91915][manager::finalize][metadata]> Outputting 'omnitrace-matrix.inst-output/2022-11-14_12.33_PM/metadata.json' and 'omnitrace-matrix.inst-output/2022-11-14_12.33_PM/functions.json'
[omnitrace][91915][0][omnitrace_finalize] Finalized
[731.210]       perfetto.cc:57383 Tracing session 1 ended, total sessions:0

real    0m0.803s
```

Check the list of the GPU calls instrumented

```
$ cat omnitrace-matrix.inst-output/2022-11-14_12.33_PM/roctracer.txt
```



```
|-----------------------------------------------------------------------------------------------------------------------|
|                                               ROCM TRACER (ACTIVITY API)                                              |
|-----------------------------------------------------------------------------------------------------------------------|
|                      LABEL                       | COUNT  | DEPTH  |  METRIC   | UNITS  |    SUM    |   MEAN    | % SELF |
|--------------------------------------------------|--------|--------|-----------|--------|-----------|-----------|--------|
| |0>>> pthread_create                             |      5 |      0 | roctracer | sec    | 0.001036  | 0.000207  |  100.0 |
| |2>>> |_start_thread                             | -      |      1 | -         | -      | -         | -         | -      |
| |2>>>   |_hsa_amd_memory_pool_allocate           |      5 |      2 | roctracer | sec    | 0.000750  | 0.000150  |  100.0 |
| |2>>>   |_hsa_iterate_agents                     |      2 |      2 | roctracer | sec    | 0.000018  | 0.000009  |  100.0 |
| |2>>>   |_hsa_amd_agents_allow_access            |      4 |      2 | roctracer | sec    | 0.000118  | 0.000030  |  100.0 |
| |2>>>   |_hsa_agent_iterate_isas                 |      1 |      2 | roctracer | sec    | 0.000001  | 0.000001  |  100.0 |
| |2>>>   |_hsa_signal_create                      |     15 |      2 | roctracer | sec    | 0.000068  | 0.000005  |  100.0 |
| |2>>>   |_hsa_executable_load_agent_code_object  |      1 |      2 | roctracer | sec    | 0.014825  | 0.014825  |  100.0 |
| |2>>>   |_hsa_amd_memory_lock_to_pool            |      3 |      2 | roctracer | sec    | 0.000538  | 0.000179  |  100.0 |
| |2>>>   |_hsa_signal_silent_store_relaxed        |      5 |      2 | roctracer | sec    | 0.000001  | 0.000000  |  100.0 |
| |2>>>   |_hsa_queue_add_write_index_screlease    |      3 |      2 | roctracer | sec    | 0.000001  | 0.000000  |  100.0 |
| |2>>>   |_hsa_signal_store_screlease             |      4 |      2 | roctracer | sec    | 0.000001  | 0.000000  |  100.0 |
| |2>>>   |_hsa_amd_signal_async_handler           |      3 |      2 | roctracer | sec    | 0.000001  | 0.000000  |  100.0 |
| |2>>>   |_hsa_signal_wait_scacquire              |      5 |      2 | roctracer | sec    | 0.009013  | 0.001803  |  100.0 |
| |2>>>   |_hsa_signal_load_relaxed                |      7 |      2 | roctracer | sec    | 0.000003  | 0.000000  |  100.0 |
| |2>>>   |_hsa_queue_load_read_index_relaxed      |      2 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |2>>>   |_hsa_signal_destroy                     |      1 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |2>>>   |_hsa_amd_memory_unlock                  |      2 |      2 | roctracer | sec    | 0.000098  | 0.000049  |  100.0 |
| |2>>>   |_hsa_queue_load_read_index_scacquire    |      2 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |2>>>   |_hsa_amd_memory_async_copy              |      1 |      2 | roctracer | sec    | 0.000002  | 0.000002  |  100.0 |
| |4>>> |_start_thread                             | -      |      1 | -         | -      | -         | -         | -      |
| |4>>>   |_hsa_amd_memory_pool_allocate           |      1 |      2 | roctracer | sec    | 0.000092  | 0.000092  |  100.0 |
| |4>>>   |_hsa_signal_create                      |     11 |      2 | roctracer | sec    | 0.000003  | 0.000000  |  100.0 |
| |4>>>   |_hsa_executable_load_agent_code_object  |      1 |      2 | roctracer | sec    | 0.005452  | 0.005452  |  100.0 |
| |4>>>   |_hsa_queue_load_read_index_relaxed      |      1 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |4>>>   |_hsa_amd_memory_lock_to_pool            |      1 |      2 | roctracer | sec    | 0.000068  | 0.000068  |  100.0 |
| |4>>>   |_hsa_queue_load_read_index_scacquire    |      1 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |4>>>   |_hsa_signal_load_relaxed                |      5 |      2 | roctracer | sec    | 0.000001  | 0.000000  |  100.0 |
| |4>>>   |_hsa_signal_destroy                     |      2 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |4>>>   |_hsa_signal_wait_scacquire              |      2 |      2 | roctracer | sec    | 0.000182  | 0.000091  |  100.0 |
| |4>>>   |_hsa_amd_memory_unlock                  |      1 |      2 | roctracer | sec    | 0.000043  | 0.000043  |  100.0 |
| |4>>>   |_hsa_amd_memory_async_copy              |      1 |      2 | roctracer | sec    | 0.000304  | 0.000304  |  100.0 |
| |4>>>   |_hsa_signal_store_screlease             |      1 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |4>>>   |_hsa_amd_memory_pool_free               |      1 |      2 | roctracer | sec    | 0.000062  | 0.000062  |  100.0 |
```



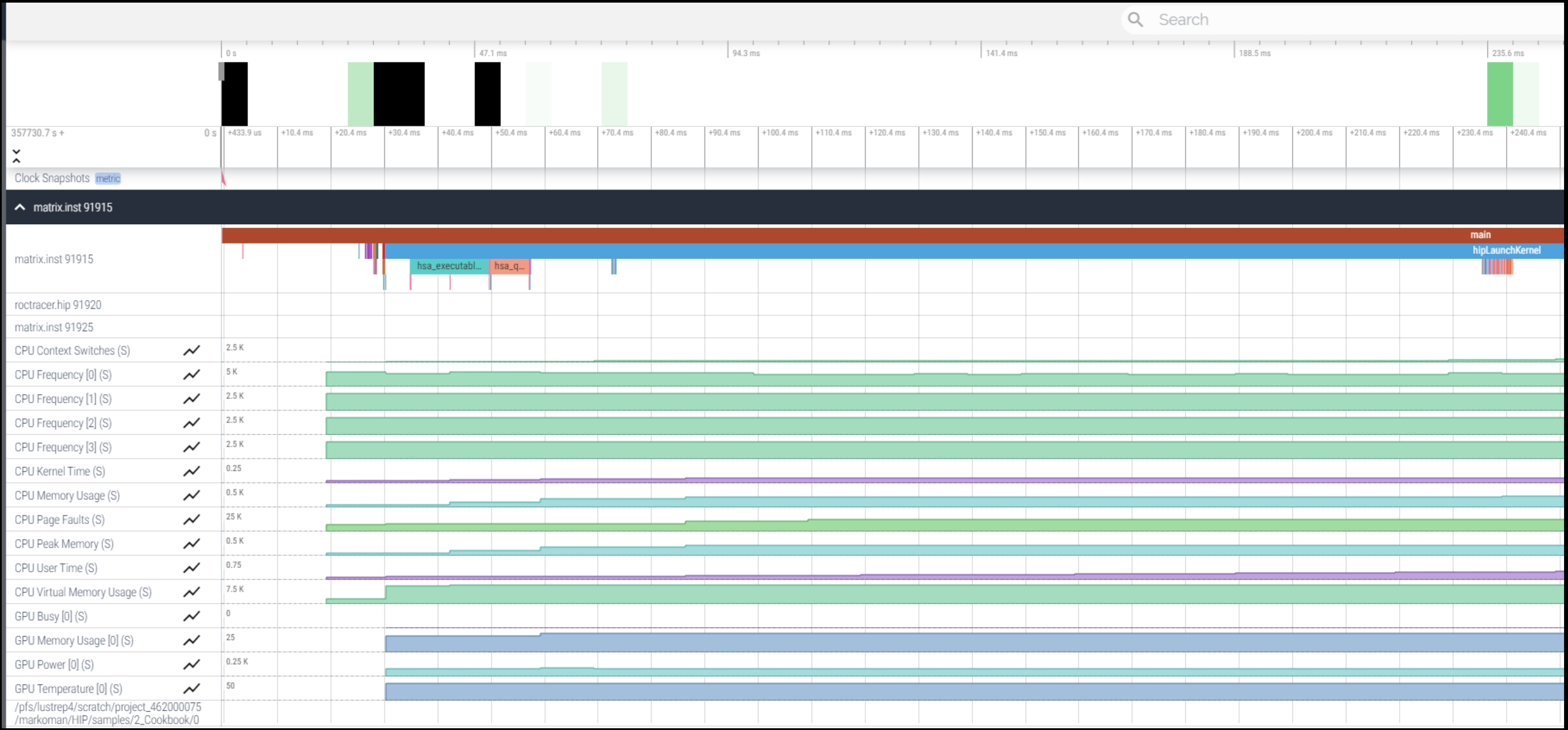
```
| |5>>> |_start_thread                             | -      |      1 | -         | -      | -         | -         | -      |
| |5>>>   |_hsa_signal_create                      |      8 |      2 | roctracer | sec    | 0.000001  | 0.000000  |  100.0 |
| |5>>>   |_hsa_queue_add_write_index_screlease    |      1 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |5>>>   |_hsa_signal_store_screlease             |      2 |      2 | roctracer | sec    | 0.000001  | 0.000001  |  100.0 |
| |5>>>   |_hsa_signal_silent_store_relaxed        |      2 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |5>>>   |_hsa_signal_load_relaxed                |      1 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |5>>>   |_hsa_amd_memory_pool_free               |      1 |      2 | roctracer | sec    | 0.000047  | 0.000047  |  100.0 |
| |3>>> |_start_thread                             | -      |      1 | -         | -      | -         | -         | -      |
| |3>>>   |_hsa_queue_create                       |      1 |      2 | roctracer | sec    | 0.007257  | 0.007257  |  100.0 |
| |3>>>   |_hsa_signal_create                      |     10 |      2 | roctracer | sec    | 0.000003  | 0.000000  |  100.0 |
| |3>>>   |_hsa_signal_load_relaxed                |      3 |      2 | roctracer | sec    | 0.000001  | 0.000000  |  100.0 |
| |3>>>   |_hsa_queue_load_read_index_scacquire    |      1 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |3>>>   |_hsa_queue_load_read_index_relaxed      |      1 |      2 | roctracer | sec    | 0.000000  | 0.000000  |  100.0 |
| |3>>>   |_hsa_amd_memory_async_copy              |      1 |      2 | roctracer | sec    | 0.000281  | 0.000281  |  100.0 |
| |1>>> |_start_thread                             | -      |      1 | -         | -      | -         | -         | -      |
| |0>>> hipGetDeviceProperties                     |      1 |      0 | roctracer | sec    | 0.000000  | 0.000000  |    0.0 |
| |0>>> hipMalloc                                  |      2 |      0 | roctracer | sec    | 0.000000  | 0.000000  |    0.0 |
| |0>>> hipLaunchKernel                            |      2 |      0 | roctracer | sec    | 0.000000  | 0.000000  |    0.0 |
| |0>>> hipMemcpy                                  |      3 |      0 | roctracer | sec    | 0.000000  | 0.000000  |    0.0 |
| |0>>> hipFree                                    |      2 |      0 | roctracer | sec    | 0.000000  | 0.000000  |    0.0 |
| |0>>> |_warmup()                                 |      1 |      1 | roctracer | sec    | 0.000001  | 0.000001  |  100.0 |
| |0>>> |_matrixTranspose(float*, float*, int)     |      1 |      1 | roctracer | sec    | 0.000085  | 0.000085  |  100.0 |
|-----------------------------------------------------------------------------------------------------------------------|
```

Visualizing trace

Use Perfetto

Copy the perfetto-trace.proto to your laptop

Go to https://ui.perfetto.dev/ click open trace and select the perfetto-trace.proto



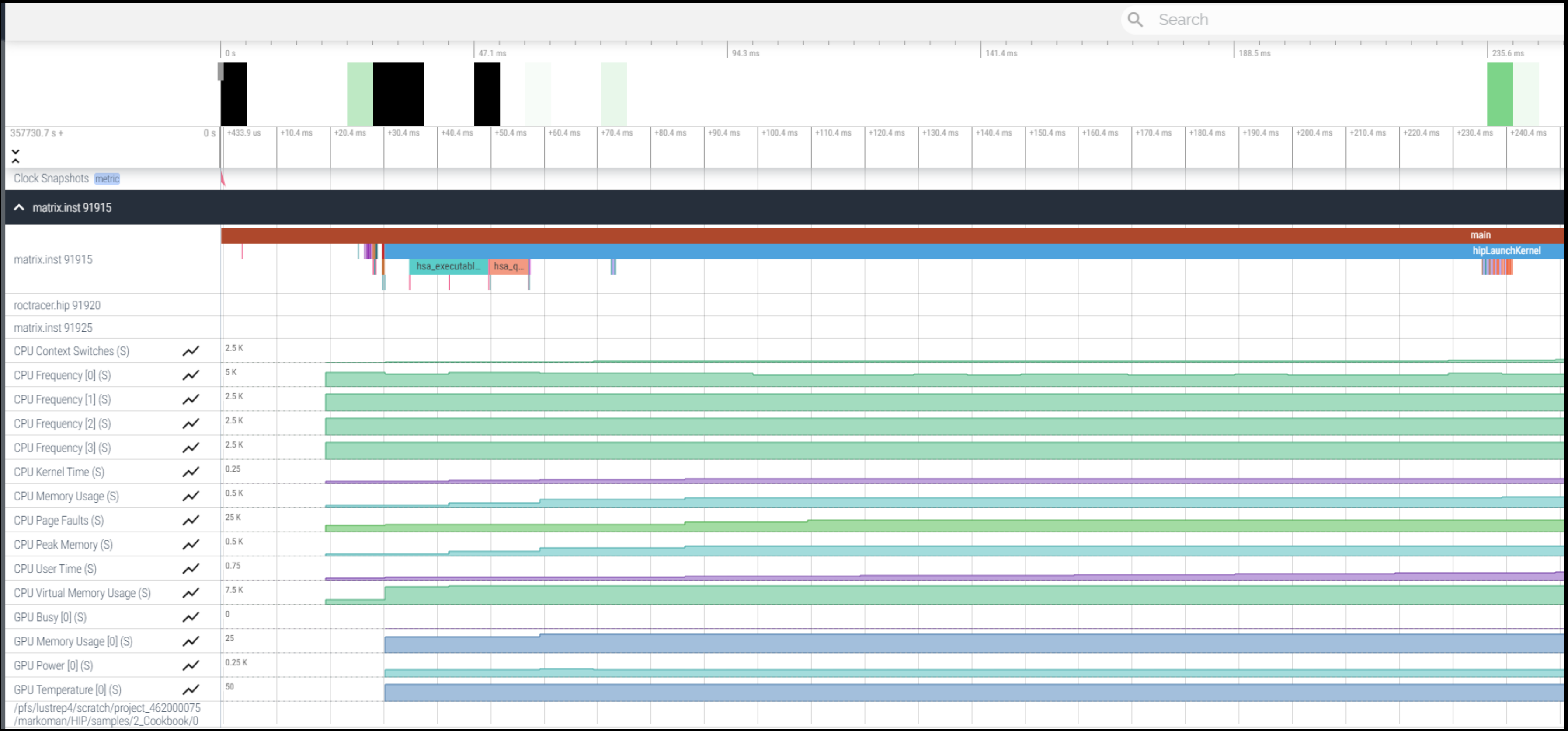
Visualizing trace

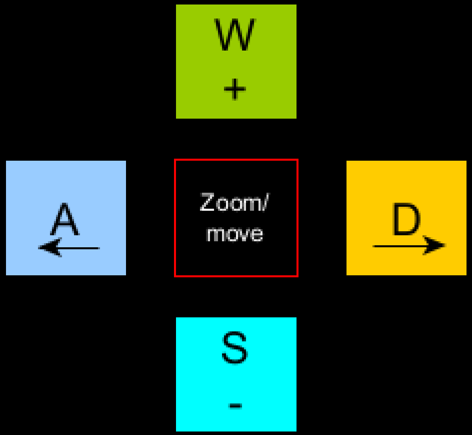
Use Perfetto

Copy the perfetto-trace.proto to your laptop

Go to https://ui.perfetto.dev/ click open trace and select the perfetto-trace.proto

Zoom and investigate the regions of interest





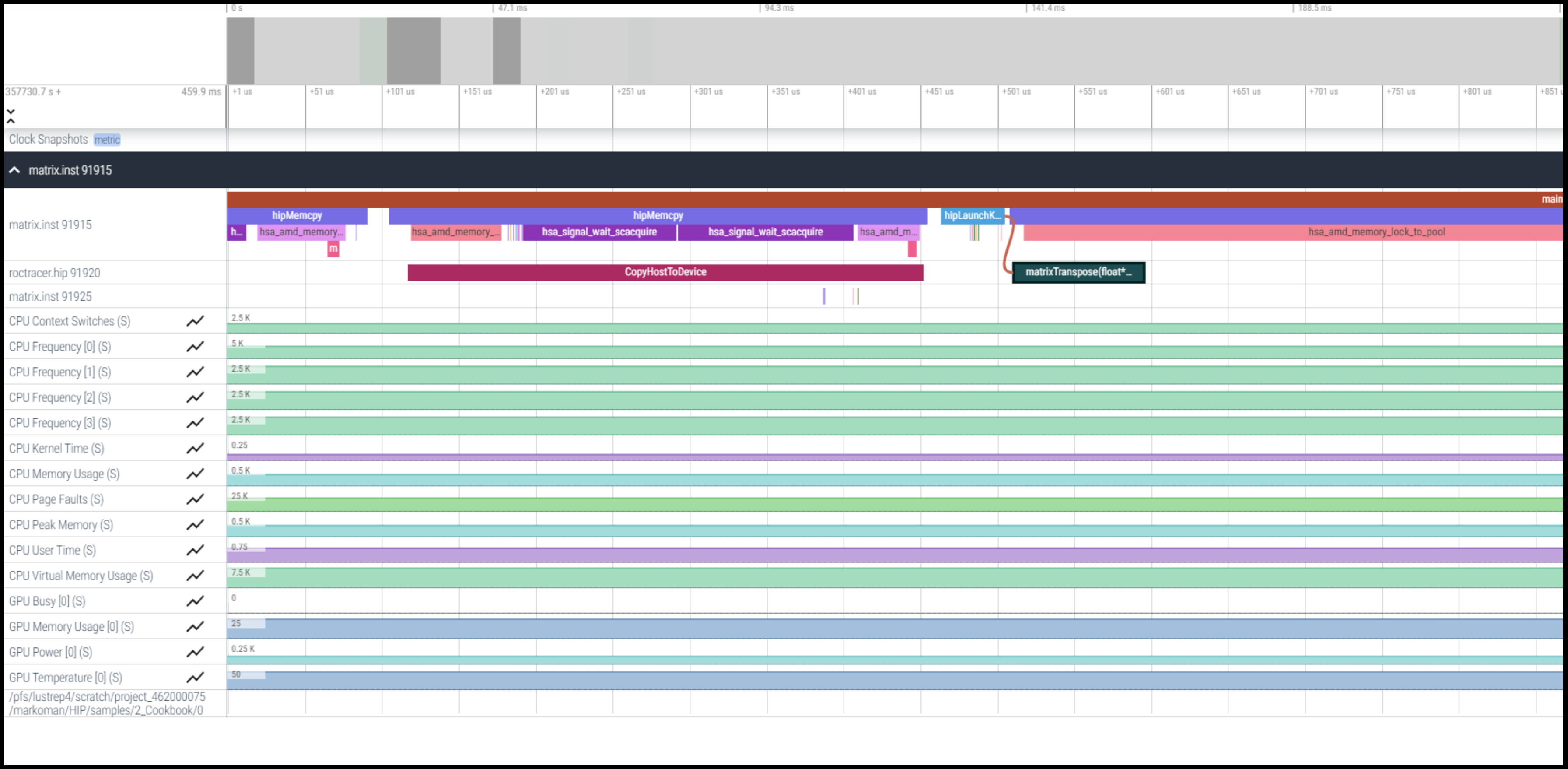
Visualizing trace

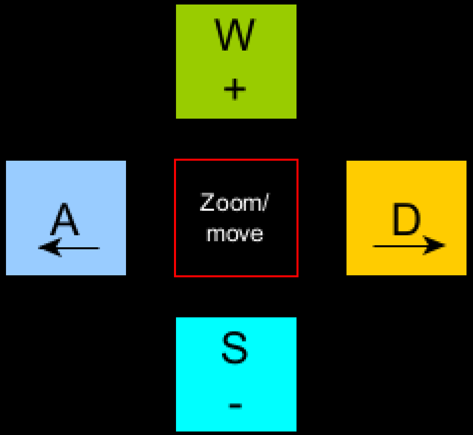
Use Perfetto

Copy the perfetto-trace.proto to your laptop

Go to https://ui.perfetto.dev/ click open trace and select the perfetto-trace.proto

Zoom and investigate the regions of interest





W +

A

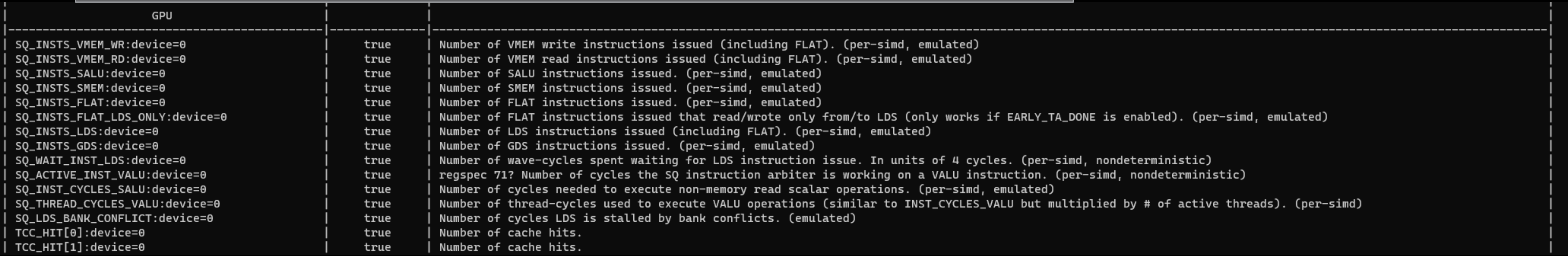
Zoom/ move

D

S -

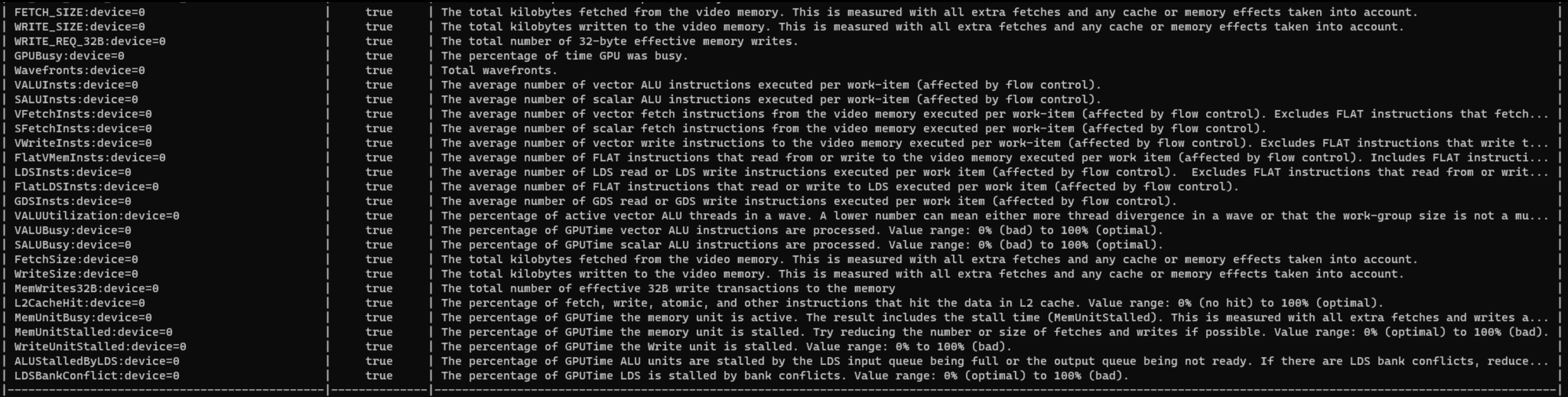
Hardware counters

```
$ omnitrace-avail --all
```



```
|                     GPU                       |               |                                                                                                                                        |
|-----------------------------------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------|
| SQ_INSTS_VMEM_WR:device=0                     |     true      | Number of VMEM write instructions issued (including FLAT). (per-simd, emulated)                                                        |
| SQ_INSTS_VMEM_RD:device=0                     |     true      | Number of VMEM read instructions issued (including FLAT). (per-simd, emulated)                                                         |
| SQ_INSTS_SALU:device=0                        |     true      | Number of SALU instructions issued. (per-simd, emulated)                                                                               |
| SQ_INSTS_SMEM:device=0                        |     true      | Number of SMEM instructions issued. (per-simd, emulated)                                                                               |
| SQ_INSTS_FLAT:device=0                        |     true      | Number of FLAT instructions issued. (per-simd, emulated)                                                                               |
| SQ_INSTS_FLAT_LDS_ONLY:device=0               |     true      | Number of FLAT instructions issued that read/wrote only from/to LDS (only works if EARLY_TA_DONE is enabled). (per-simd, emulated)     |
| SQ_INSTS_LDS:device=0                         |     true      | Number of LDS instructions issued (including FLAT). (per-simd, emulated)                                                               |
| SQ_INSTS_GDS:device=0                         |     true      | Number of GDS instructions issued. (per-simd, emulated)                                                                                |
| SQ_WAIT_INST_LDS:device=0                     |     true      | Number of wave-cycles spent waiting for LDS instruction issue. In units of 4 cycles. (per-simd, nondeterministic)                      |
| SQ_ACTIVE_INST_VALU:device=0                  |     true      | regspec 71? Number of cycles the SQ instruction arbiter is working on a VALU instruction. (per-simd, nondeterministic)                 |
| SQ_INST_CYCLES_SALU:device=0                  |     true      | Number of cycles needed to execute non-memory read scalar operations. (per-simd, emulated)                                             |
| SQ_THREAD_CYCLES_VALU:device=0                |     true      | Number of thread-cycles used to execute VALU operations (similar to INST_CYCLES_VALU but multiplied by # of active threads). (per-simd) |
| SQ_LDS_BANK_CONFLICT:device=0                 |     true      | Number of cycles LDS is stalled by bank conflicts. (emulated)                                                                          |
| TCC_HIT[0]:device=0                           |     true      | Number of cache hits.                                                                                                                  |
| TCC_HIT[1]:device=0                           |     true      | Number of cache hits.                                                                                                                  |
```

...



```
| FETCH_SIZE:device=0                           |     true      | The total kilobytes fetched from the video memory. This is measured with all extra fetches and any cache or memory effects taken into account.  |
| WRITE_SIZE:device=0                           |     true      | The total kilobytes written to the video memory. This is measured with all extra fetches and any cache or memory effects taken into account.   |
| WRITE_REQ_32B:device=0                        |     true      | The total number of 32-byte effective memory writes.                                                                                            |
| GPUBusy:device=0                              |     true      | The percentage of time GPU was busy.                                                                                                            |
| Wavefronts:device=0                           |     true      | Total wavefronts.                                                                                                                               |
| VALUInsts:device=0                            |     true      | The average number of vector ALU instructions executed per work-item (affected by flow control).                                               |
| SALUInsts:device=0                            |     true      | The average number of scalar ALU instructions executed per work-item (affected by flow control).                                               |
| VFetchInsts:device=0                          |     true      | The average number of vector fetch instructions from the video memory executed per work-item (affected by flow control). Excludes FLAT instructions that fetch... |
| SFetchInsts:device=0                          |     true      | The average number of scalar fetch instructions from the video memory executed per work-item (affected by flow control).                       |
| VWriteInsts:device=0                          |     true      | The average number of vector write instructions to the video memory executed per work-item (affected by flow control). Excludes FLAT instructions that write t... |
| FlatVMemInsts:device=0                        |     true      | The average number of FLAT instructions that read from or write to the video memory executed per work item (affected by flow control). Includes FLAT instructi... |
| LDSInsts:device=0                             |     true      | The average number of LDS read or LDS write instructions executed per work item (affected by flow control).  Excludes FLAT instructions that read from or writ... |
| FlatLDSInsts:device=0                         |     true      | The average number of FLAT instructions that read or write to LDS executed per work item (affected by flow control).                           |
| GDSInsts:device=0                             |     true      | The average number of GDS read or GDS write instructions executed per work item (affected by flow control).                                    |
| VALUUtilization:device=0                      |     true      | The percentage of active vector ALU threads in a wave. A lower number can mean either more thread divergence in a wave or that the work-group size is not a mu... |
| VALUBusy:device=0                             |     true      | The percentage of GPUTime vector ALU instructions are processed. Value range: 0% (bad) to 100% (optimal).                                      |
| SALUBusy:device=0                             |     true      | The percentage of GPUTime scalar ALU instructions are processed. Value range: 0% (bad) to 100% (optimal).                                      |
| FetchSize:device=0                            |     true      | The total kilobytes fetched from the video memory. This is measured with all extra fetches and any cache or memory effects taken into account.  |
| WriteSize:device=0                            |     true      | The total kilobytes written to the video memory. This is measured with all extra fetches and any cache or memory effects taken into account.   |
| MemWrites32B:device=0                         |     true      | The total number of effective 32B write transactions to the memory                                                                              |
| L2CacheHit:device=0                           |     true      | The percentage of fetch, write, atomic, and other instructions that hit the data in L2 cache. Value range: 0% (no hit) to 100% (optimal).      |
| MemUnitBusy:device=0                          |     true      | The percentage of GPUTime the memory unit is active. The result includes the stall time (MemUnitStalled). This is measured with all extra fetches and writes a... |
| MemUnitStalled:device=0                       |     true      | The percentage of GPUTime the memory unit is stalled. Try reducing the number or size of fetches and writes if possible. Value range: 0% (optimal) to 100% (bad). |
| WriteUnitStalled:device=0                     |     true      | The percentage of GPUTime the Write unit is stalled. Value range: 0% to 100% (bad).                                                             |
| ALUStalledByLDS:device=0                      |     true      | The percentage of GPUTime ALU units are stalled by the LDS input queue being full or the output queue being not ready. If there are LDS bank conflicts, reduce... |
| LDSBankConflict:device=0                      |     true      | The percentage of GPUTime LDS is stalled by bank conflicts. Value range: 0% (optimal) to 100% (bad).                                           |
|-----------------------------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
```

Commonly Used Counters

VALUUtilization	The percentage of ALUs active in a wave. Low VALUUtilization is likely due to high divergence or a poorly sized grid
VALUBusy	The percentage of GPUTime vector ALU instructions are processed. Can be thought of as something like compute utilization
FetchSize	The total kilobytes fetched from global memory
WriteSize	The total kilobytes written to global memory
L2CacheHit	The percentage of fetch, write, atomic, and other instructions that hit the data in L2 cache
MemUnitBusy	The percentage of GPUTime the memory unit is active. The result includes the stall time
MemUnitStalled	The percentage of GPUTime the memory unit is stalled
WriteUnitStalled	The percentage of GPUTime the write unit is stalled

Full list at: https://github.com/ROCm-Developer-Tools/rocprofiler/blob/amd-master/test/tool/metrics.xml

Modify config file

Create a config file in `$HOME`:

```
$ omnitrace-avail -G $HOME/.omnitrace.cfg
```

Modify the config file $HOME/.omnitrace.cfg to add desired metrics and for concerned GPU#ID:

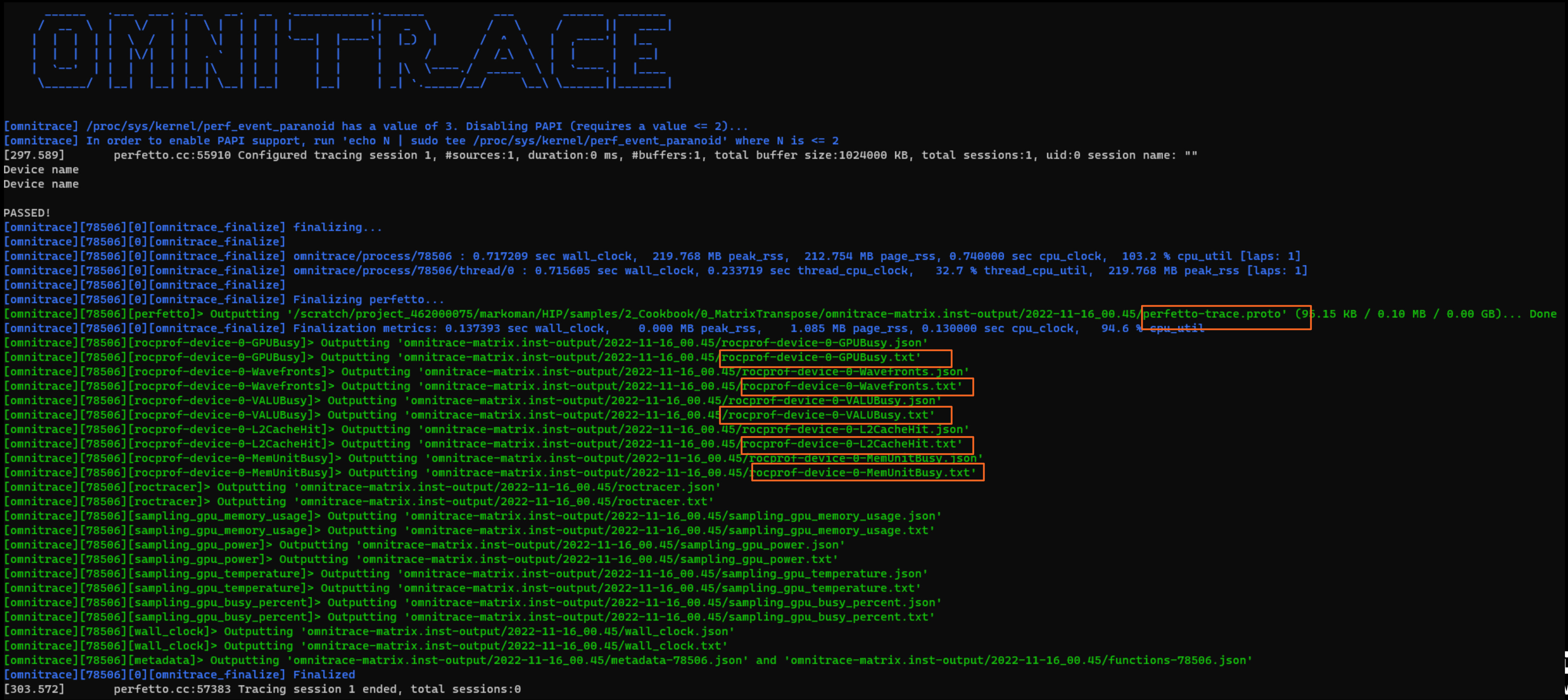
```
…
OMNITRACE_ROCM_EVENTS = GPUBusy:device=0,
Wavefronts:device=0, VALUBusy:device=0,
L2CacheHit:device=0, MemUnitBusy:device=0
…
```

To profile desired metrics for all participating GPUs:

```
…
OMNITRACE_ROCM_EVENTS = GPUBusy, Wavefronts,
VALUBusy, L2CacheHit, MemUnitBusy
…
```

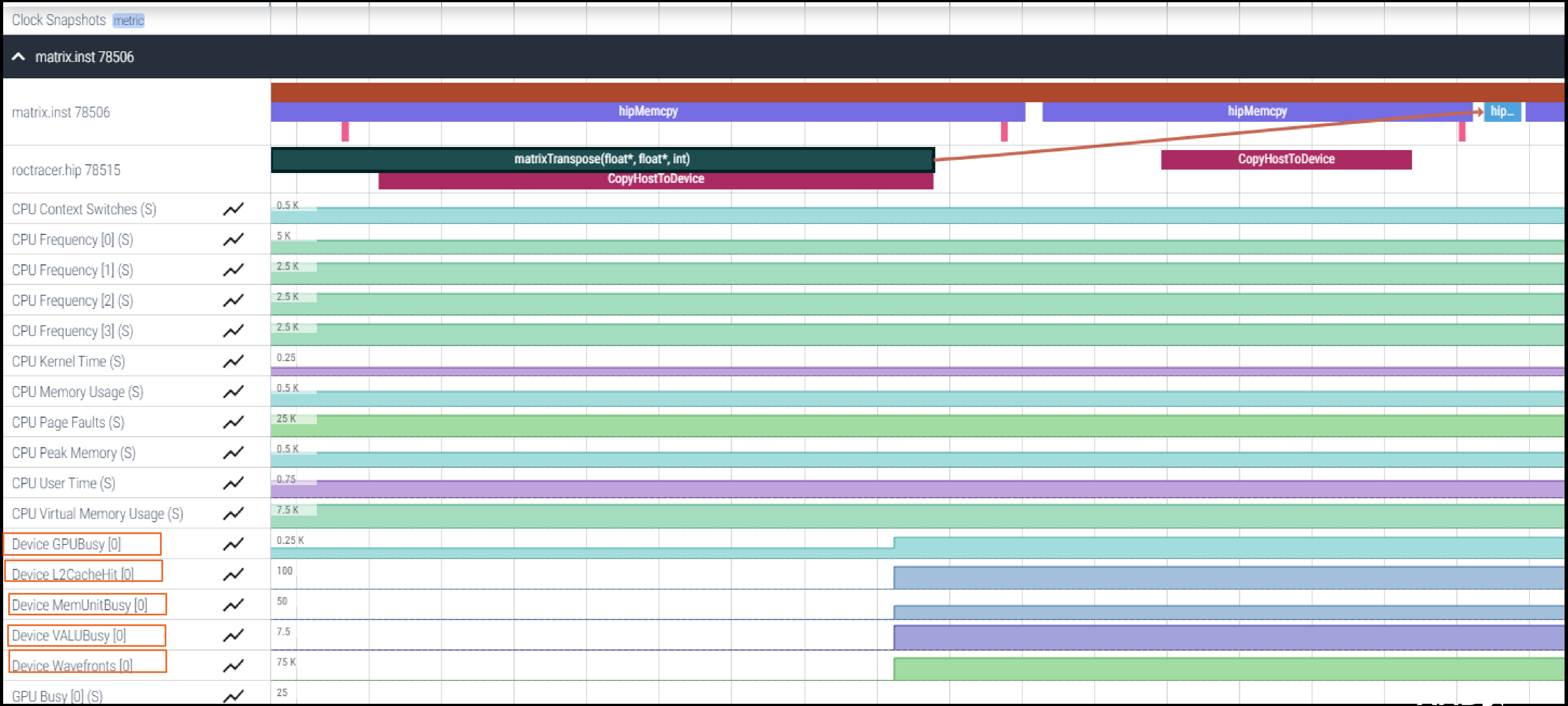
Execution with hardware counters

```
$ ./matrix.inst
```



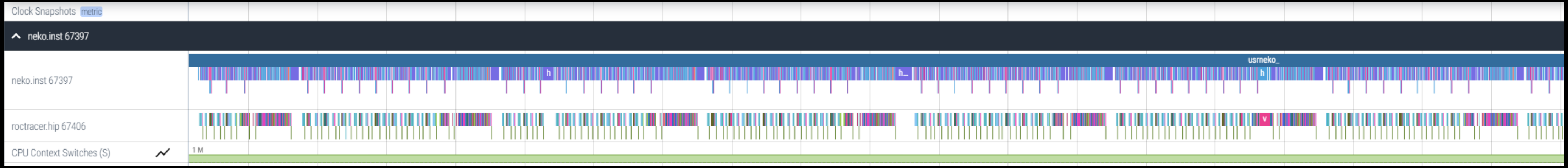
```
[omnitrace] /proc/sys/kernel/perf_event_paranoid has a value of 3. Disabling PAPI (requires a value <= 2)...
[omnitrace] In order to enable PAPI support, run 'echo N | sudo tee /proc/sys/kernel/perf_event_paranoid' where N is <= 2
[297.589]       perfetto.cc:55910 Configured tracing session 1, #sources:1, duration:0 ms, #buffers:1, total buffer size:1024000 KB, total sessions:1, uid:0 session name: ""
Device name
Device name

PASSED!
[omnitrace][78506][0][omnitrace_finalize] finalizing...
[omnitrace][78506][0][omnitrace_finalize]
[omnitrace][78506][0][omnitrace_finalize] omnitrace/process/78506 : 0.717209 sec wall_clock,  219.768 MB peak_rss,  212.754 MB page_rss, 0.740000 sec cpu_clock,  103.2 % cpu_util [laps: 1]
[omnitrace][78506][0][omnitrace_finalize] omnitrace/process/78506/thread/0 : 0.715605 sec wall_clock, 0.233719 sec thread_cpu_clock,   32.7 % thread_cpu_util,  219.768 MB peak_rss [laps: 1]
[omnitrace][78506][0][omnitrace_finalize]
[omnitrace][78506][0][omnitrace_finalize] Finalizing perfetto...
[omnitrace][78506][perfetto]> Outputting '/scratch/project_462000075/markoman/HIP/samples/2_Cookbook/0_MatrixTranspose/omnitrace-matrix.inst-output/2022-11-16_00.45/perfetto-trace.proto' (95.15 KB / 0.10 MB / 0.00 GB)... Done
[omnitrace][78506][0][omnitrace_finalize] Finalization metrics: 0.137393 sec wall_clock,    0.000 MB peak_rss,    1.085 MB page_rss, 0.130000 sec cpu_clock,   94.6 % cpu_util
[omnitrace][78506][rocprof-device-0-GPUBusy]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-GPUBusy.json'
[omnitrace][78506][rocprof-device-0-GPUBusy]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-GPUBusy.txt'
[omnitrace][78506][rocprof-device-0-Wavefronts]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-Wavefronts.json'
[omnitrace][78506][rocprof-device-0-Wavefronts]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-Wavefronts.txt'
[omnitrace][78506][rocprof-device-0-VALUBusy]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-VALUBusy.json'
[omnitrace][78506][rocprof-device-0-VALUBusy]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-VALUBusy.txt'
[omnitrace][78506][rocprof-device-0-L2CacheHit]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-L2CacheHit.json'
[omnitrace][78506][rocprof-device-0-L2CacheHit]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-L2CacheHit.txt'
[omnitrace][78506][rocprof-device-0-MemUnitBusy]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-MemUnitBusy.json'
[omnitrace][78506][rocprof-device-0-MemUnitBusy]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/rocprof-device-0-MemUnitBusy.txt'
[omnitrace][78506][roctracer]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/roctracer.json'
[omnitrace][78506][roctracer]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/roctracer.txt'
[omnitrace][78506][sampling_gpu_memory_usage]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/sampling_gpu_memory_usage.json'
[omnitrace][78506][sampling_gpu_memory_usage]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/sampling_gpu_memory_usage.txt'
[omnitrace][78506][sampling_gpu_power]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/sampling_gpu_power.json'
[omnitrace][78506][sampling_gpu_power]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/sampling_gpu_power.txt'
[omnitrace][78506][sampling_gpu_temperature]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/sampling_gpu_temperature.json'
[omnitrace][78506][sampling_gpu_temperature]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/sampling_gpu_temperature.txt'
[omnitrace][78506][sampling_gpu_busy_percent]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/sampling_gpu_busy_percent.json'
[omnitrace][78506][sampling_gpu_busy_percent]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/sampling_gpu_busy_percent.txt'
[omnitrace][78506][wall_clock]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/wall_clock.json'
[omnitrace][78506][wall_clock]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/wall_clock.txt'
[omnitrace][78506][metadata]> Outputting 'omnitrace-matrix.inst-output/2022-11-16_00.45/metadata-78506.json' and 'omnitrace-matrix.inst-output/2022-11-16_00.45/functions-78506.json'
[omnitrace][78506][0][omnitrace_finalize] Finalized
[303.572]       perfetto.cc:57383 Tracing session 1 ended, total sessions:0
```

Visualization with hardware counters

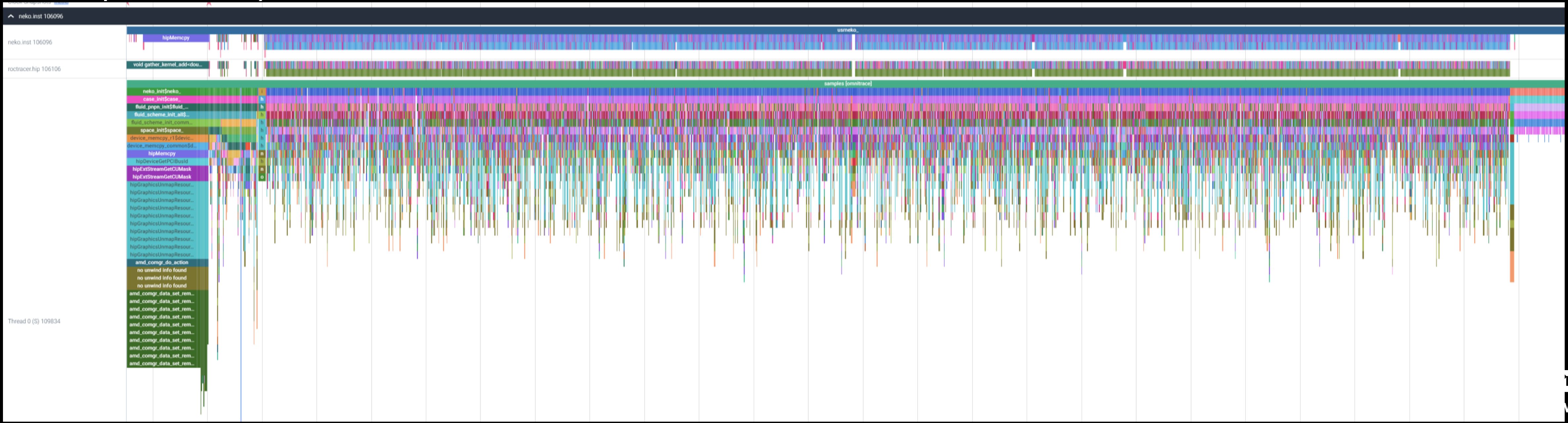

Clock Snapshots metric
matrix.inst 78506
matrix.inst 78506
hipMemcpy
hipMemcpy
hip...
roctracer.hip 78515
matrixTranspose(float*, float*, int)
CopyHostToDevice
CopyHostToDevice
CPU Context Switches (S) 0.5 K
CPU Frequency [0] (S) 5 K
CPU Frequency [1] (S) 2.5 K
CPU Frequency [2] (S) 2.5 K
CPU Frequency [3] (S) 2.5 K
CPU Kernel Time (S) 0.25
CPU Memory Usage (S) 0.5 K
CPU Page Faults (S) 25 K
CPU Peak Memory (S) 0.5 K
CPU User Time (S) 0.75
CPU Virtual Memory Usage (S) 7.5 K
Device GPUBusy [0] 0.25 K
Device L2CacheHit [0] 100
Device MemUnitBusy [0] 50
Device VALUBusy [0] 7.5
Device Wavefronts [0] 75 K
GPU Busy [0] (S) 25

Sampling call-stack (I)

- Another application with OMNITRACE_USE_SAMPLING = false

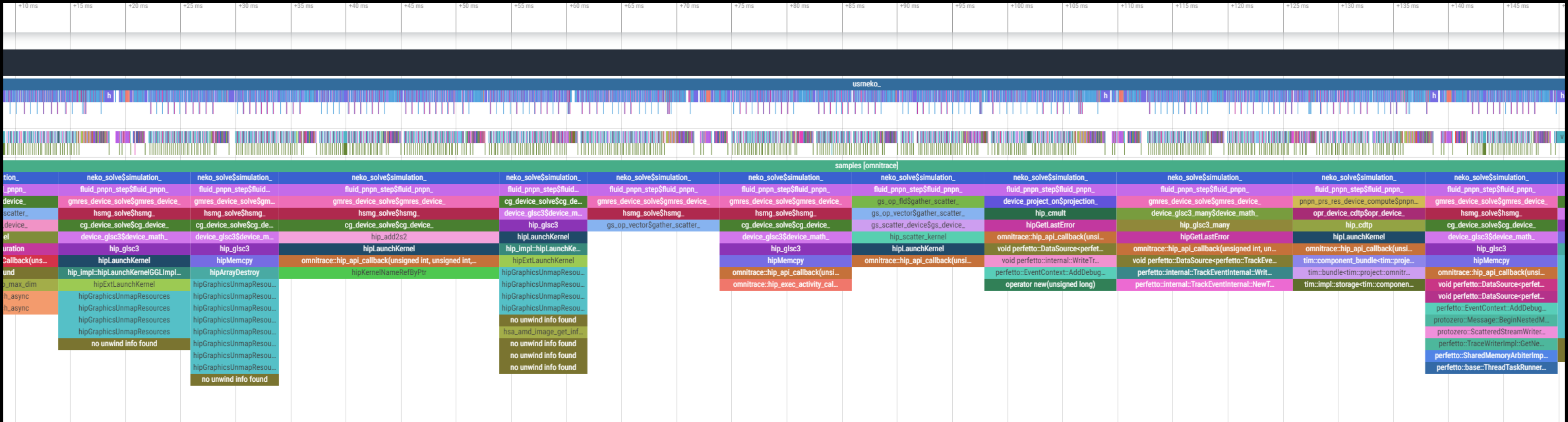


- With OMNITRACE_USE_SAMPLING = true and OMNITRACE_SAMPLING_FREQ = 100 (100 samples per second)


neko.inst 106096
neko.inst 106096
hipMemcpy
usrneko_
roctracer.hip 106106
void gather_kernel_add<dou...
samples [omnitrace]
Thread 0 (S) 109834
neko_init$neko_
case_init$case_
fluid_pnpn_init$fluid_...
fluid_scheme_init_all$...
fluid_scheme_init_comm...
space_init$space_
device_memcpy_r1$devic...
device_memcpy_common$d...
hipMemcpy
hipDeviceGetPCIBusId
hipExtStreamGetCUMask
hipExtStreamGetCUMask
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
hipGraphicsUnmapResour...
amd_comgr_do_action
no unwind info found
no unwind info found
no unwind info found
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...
amd_comgr_data_set_rem...

Sampling call-stack (II)

- Zoom in call-stack sampling


+10 ms
+15 ms
+20 ms
+25 ms
+30 ms
+35 ms
+40 ms
+45 ms
+50 ms
+55 ms
+60 ms
+65 ms
+70 ms
+75 ms
+80 ms
+85 ms
+90 ms
+95 ms
+100 ms
+105 ms
+110 ms
+115 ms
+120 ms
+125 ms
+130 ms
+135 ms
+140 ms
+145 ms
usmeko_
samples [omnitrace]
tion_
pnpn
device_
scatter_
device_
el
uration
Callback(uns…
und
_max_dim
h_async
h_async
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
gmres_device_solve$gmres_device_
hsmg_solve$hsmg_
cg_device_solve$cg_device_
device_glsc3$device_math_
hip_glsc3
hipLaunchKernel
hip_impl::hipLaunchKernelGGLImpl…
hipExtLaunchKernel
hipGraphicsUnmapResources
hipGraphicsUnmapResources
hipGraphicsUnmapResources
hipGraphicsUnmapResources
no unwind info found
neko_solve$simulation_
fluid_pnpn_step$fluid…
gmres_device_solve$gm…
hsmg_solve$hsmg_
cg_device_solve$cg_de…
device_glsc3$device_m…
hip_glsc3
hipMemcpy
hipArrayDestroy
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
no unwind info found
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
gmres_device_solve$gmres_device_
hsmg_solve$hsmg_
cg_device_solve$cg_device_
hip_add2s2
hipLaunchKernel
omnitrace::hip_api_callback(unsigned int, unsigned int,…
hipKernelNameRefByPtr
neko_solve$simulation_
fluid_pnpn_step$fluid…
cg_device_solve$cg_de…
device_glsc3$device_m…
hip_glsc3
hipLaunchKernel
hip_impl::hipLaunchKe…
hipExtLaunchKernel
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
hipGraphicsUnmapResou…
no unwind info found
hsa_amd_image_get_inf…
no unwind info found
no unwind info found
no unwind info found
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
gmres_device_solve$gmres_device_
hsmg_solve$hsmg_
gs_op_vector$gather_scatter_
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
gmres_device_solve$gmres_device_
hsmg_solve$hsmg_
cg_device_solve$cg_device_
device_glsc3$device_math_
hip_glsc3
hipMemcpy
omnitrace::hip_api_callback(unsi…
omnitrace::hip_exec_activity_cal…
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
gs_op_fld$gather_scatter_
gs_op_vector$gather_scatter_
gs_scatter_device$gs_device_
hip_scatter_kernel
hipLaunchKernel
omnitrace::hip_api_callback(unsi…
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
device_project_on$projection_
hip_cmult
hipGetLastError
omnitrace::hip_api_callback(unsi…
void perfetto::DataSource<perfet…
void perfetto::internal::WriteTr…
perfetto::EventContext::AddDebug…
operator new(unsigned long)
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
gmres_device_solve$gmres_device_
device_glsc3_many$device_math_
hip_glsc3_many
hipGetLastError
omnitrace::hip_api_callback(unsigned int, un…
void perfetto::DataSource<perfetto::TrackEve…
perfetto::internal::TrackEventInternal::Writ…
perfetto::internal::TrackEventInternal::NewT…
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
pnpn_prs_res_device_compute$pnpn…
opr_device_cdtp$opr_device_
hip_cdtp
hipLaunchKernel
omnitrace::hip_api_callback(unsi…
tim::component_bundle<tim::proje…
tim::bundle<tim::project::omnitr…
tim::impl::storage<tim::componen…
neko_solve$simulation_
fluid_pnpn_step$fluid_pnpn_
gmres_device_solve$gmres_device_
hsmg_solve$hsmg_
cg_device_solve$cg_device_
device_glsc3$device_math_
hip_glsc3
hipMemcpy
omnitrace::hip_api_callback(unsi…
void perfetto::DataSource<perfet…
void perfetto::DataSource<perfet…
perfetto::EventContext::AddDebug…
protozero::Message::BeginNestedM…
protozero::ScatteredStreamWriter…
perfetto::TraceWriterImpl::GetNe…
perfetto::SharedMemoryArbiterImp…
perfetto::base::ThreadTaskRunner…

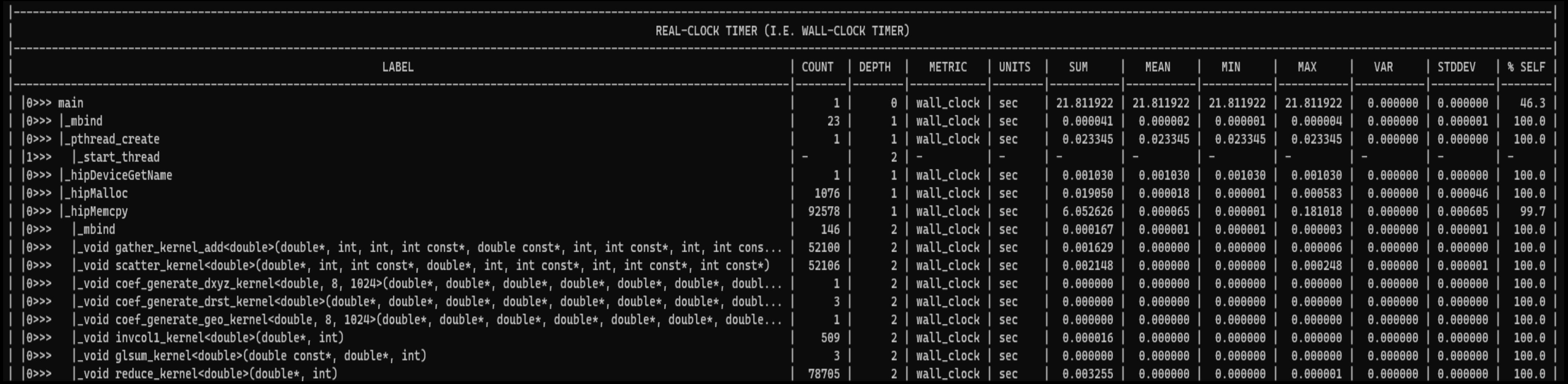
AMD
together we advance_

How to see kernels timing?

```
$ cat omnitrace-binary-output/timestamp/wall_clock.txt
```

If you do not see a wall_clock.txt dumped by omnitrace, try modify the config file $HOME/.omnitrace.cfg and enable OMNITRACE_USE_TIMEMORY:

```
…
OMNITRACE_USE_PERFETTO                                    = true
OMNITRACE_USE_TIMEMORY                                    = true
OMNITRACE_USE_SAMPLING                                    = false
…
```



```
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                        REAL-CLOCK TIMER (I.E. WALL-CLOCK TIMER)                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                          LABEL                                                           | COUNT  | DEPTH  |   METRIC   | UNITS |    SUM    |   MEAN    |    MIN    |    MAX    |    VAR    |  STDDEV   | % SELF |
|--------------------------------------------------------------------------------------------------------------------------|--------|--------|------------|-------|-----------|-----------|-----------|-----------|-----------|-----------|--------|
| |0>>> main                                                                                                               |      1 |      0 | wall_clock | sec   | 21.811922 | 21.811922 | 21.811922 | 21.811922 |  0.000000 |  0.000000 |   46.3 |
| |0>>> |_mbind                                                                                                            |     23 |      1 | wall_clock | sec   |  0.000041 |  0.000002 |  0.000001 |  0.000004 |  0.000000 |  0.000001 |  100.0 |
| |0>>> |_pthread_create                                                                                                   |      1 |      1 | wall_clock | sec   |  0.023345 |  0.023345 |  0.023345 |  0.023345 |  0.000000 |  0.000000 |  100.0 |
| |1>>>   |_start_thread                                                                                                   | -      |      2 | -          | -     | -         | -         | -         | -         | -         | -         | -      |
| |0>>> |_hipDeviceGetName                                                                                                 |      1 |      1 | wall_clock | sec   |  0.001030 |  0.001030 |  0.001030 |  0.001030 |  0.000000 |  0.000000 |  100.0 |
| |0>>> |_hipMalloc                                                                                                        |   1076 |      1 | wall_clock | sec   |  0.019050 |  0.000018 |  0.000001 |  0.000583 |  0.000000 |  0.000046 |  100.0 |
| |0>>> |_hipMemcpy                                                                                                        |  92578 |      1 | wall_clock | sec   |  6.052626 |  0.000065 |  0.000001 |  0.181018 |  0.000000 |  0.000605 |   99.7 |
| |0>>>   |_mbind                                                                                                          |    146 |      2 | wall_clock | sec   |  0.000167 |  0.000001 |  0.000001 |  0.000003 |  0.000000 |  0.000001 |  100.0 |
| |0>>>   |_void gather_kernel_add<double>(double*, int, int, int const*, double const*, int, int const*, int, int cons... |  52100 |      2 | wall_clock | sec   |  0.001629 |  0.000000 |  0.000000 |  0.000006 |  0.000000 |  0.000000 |  100.0 |
| |0>>>   |_void scatter_kernel<double>(double*, int, int const*, double*, int, int const*, int, int const*, int const*)   |  52106 |      2 | wall_clock | sec   |  0.002148 |  0.000000 |  0.000000 |  0.000248 |  0.000000 |  0.000001 |  100.0 |
| |0>>>   |_void coef_generate_dxyz_kernel<double, 8, 1024>(double*, double*, double*, double*, double*, double*, doubl... |      1 |      2 | wall_clock | sec   |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  100.0 |
| |0>>>   |_void coef_generate_drst_kernel<double>(double*, double*, double*, double*, double*, double*, double*, doubl... |      3 |      2 | wall_clock | sec   |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  100.0 |
| |0>>>   |_void coef_generate_geo_kernel<double, 8, 1024>(double*, double*, double*, double*, double*, double*, double... |      1 |      2 | wall_clock | sec   |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  100.0 |
| |0>>>   |_void invcol1_kernel<double>(double*, int)                                                                      |    509 |      2 | wall_clock | sec   |  0.000016 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  100.0 |
| |0>>>   |_void glsum_kernel<double>(double const*, double*, int)                                                         |      3 |      2 | wall_clock | sec   |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  0.000000 |  100.0 |
| |0>>>   |_void reduce_kernel<double>(double*, int)                                                                       |  78705 |      2 | wall_clock | sec   |  0.003255 |  0.000000 |  0.000000 |  0.000001 |  0.000000 |  0.000000 |  100.0 |
```

User API

- Omnitrace provides an API to control the instrumentation

API Call	Description
int **omnitrace_user_start_trace(void)**	Enable tracing on this thread and all subsequently created threads
int **omnitrace_user_stop_trace(void)**	Disable tracing on this thread and all subsequently created threads
int **omnitrace_user_start_thread_trace(void)**	Enable tracing on this specific thread. Does not apply to subsequently created threads
int **omnitrace_user_stop_thread_trace(void)**	Disable tracing on this specific thread. Does not apply to subsequently created threads

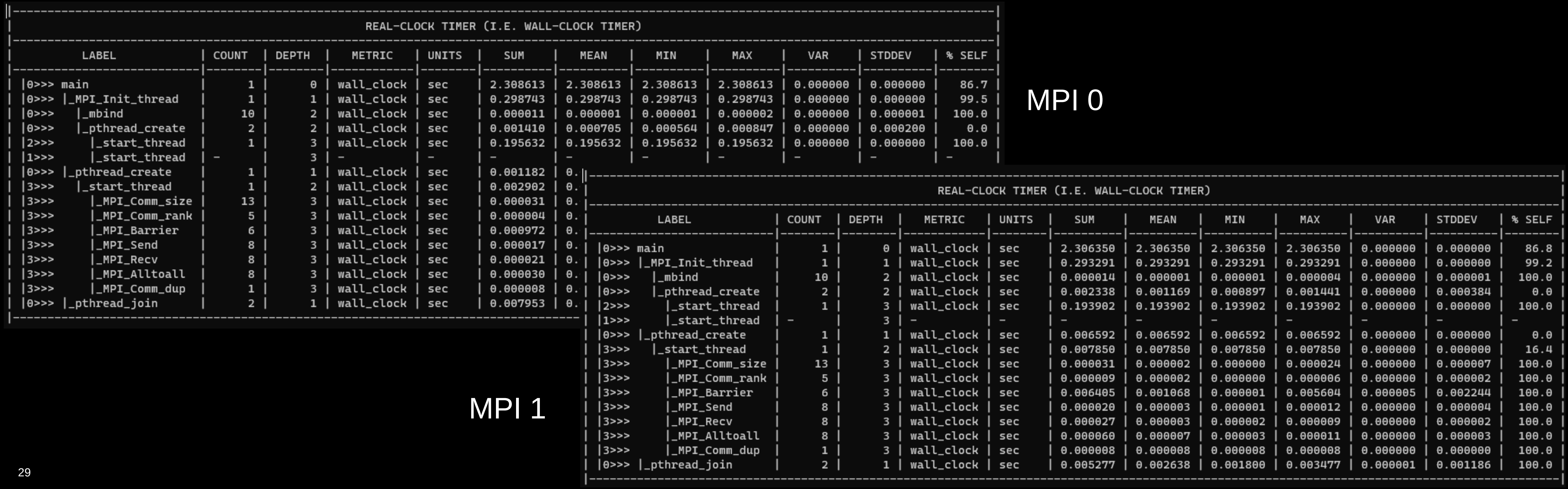
All the API calls: https://amdresearch.github.io/omnitrace/user_api.html

Profiling MPI-based applications

We use the example omnitrace/examples/mpi/mpi.cpp

Compile, create a instrumented binary and then run:

```
$ srun -n 1 omnitrace -o mpi.inst -- ./mpi
$ srun -n 2 ./mpi.inst
```



MPI 0

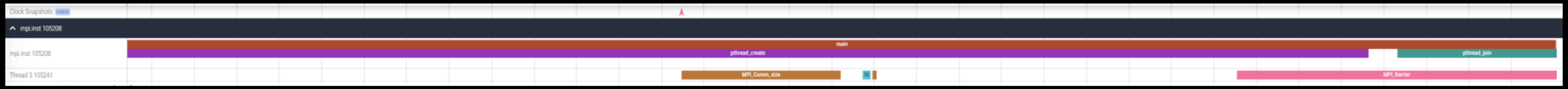
```
|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                    REAL-CLOCK TIMER (I.E. WALL-CLOCK TIMER)                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|          LABEL           | COUNT  | DEPTH  |   METRIC   | UNITS  |   SUM    |   MEAN   |   MIN    |   MAX    |   VAR    |  STDDEV  | % SELF |
|--------------------------|--------|--------|------------|--------|----------|----------|----------|----------|----------|----------|--------|
| |0>>> main               |      1 |      0 | wall_clock | sec    | 2.308613 | 2.308613 | 2.308613 | 2.308613 | 0.000000 | 0.000000 |   86.7 |
| |0>>> |_MPI_Init_thread  |      1 |      1 | wall_clock | sec    | 0.298743 | 0.298743 | 0.298743 | 0.298743 | 0.000000 | 0.000000 |   99.5 |
| |0>>>   |_mbind          |     10 |      2 | wall_clock | sec    | 0.000011 | 0.000001 | 0.000001 | 0.000002 | 0.000000 | 0.000001 |  100.0 |
| |0>>>   |_pthread_create |      2 |      2 | wall_clock | sec    | 0.001410 | 0.000705 | 0.000564 | 0.000847 | 0.000000 | 0.000200 |    0.0 |
| |2>>>     |_start_thread |      1 |      3 | wall_clock | sec    | 0.195632 | 0.195632 | 0.195632 | 0.195632 | 0.000000 | 0.000000 |  100.0 |
| |1>>>     |_start_thread | -      |      3 | -          | -      | -        | -        | -        | -        | -        | -        | -      |
| |0>>> |_pthread_create   |      1 |      1 | wall_clock | sec    | 0.001182 | 0.
| |3>>>   |_start_thread   |      1 |      2 | wall_clock | sec    | 0.002902 | 0.
| |3>>>     |_MPI_Comm_size |    13 |      3 | wall_clock | sec    | 0.000031 | 0.
| |3>>>     |_MPI_Comm_rank |     5 |      3 | wall_clock | sec    | 0.000004 | 0.
| |3>>>     |_MPI_Barrier  |      6 |      3 | wall_clock | sec    | 0.000972 | 0.
| |3>>>     |_MPI_Send     |      8 |      3 | wall_clock | sec    | 0.000017 | 0.
| |3>>>     |_MPI_Recv     |      8 |      3 | wall_clock | sec    | 0.000021 | 0.
| |3>>>     |_MPI_Alltoall |      8 |      3 | wall_clock | sec    | 0.000030 | 0.
| |3>>>     |_MPI_Comm_dup |      1 |      3 | wall_clock | sec    | 0.000008 | 0.
| |0>>> |_pthread_join     |      2 |      1 | wall_clock | sec    | 0.007953 | 0.
|---------------------------------------------------------------------------------------
```

MPI 1

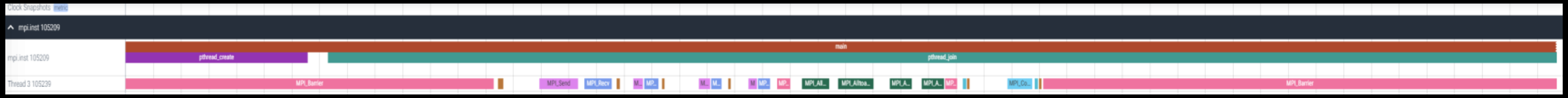
```
|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                    REAL-CLOCK TIMER (I.E. WALL-CLOCK TIMER)                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|          LABEL           | COUNT  | DEPTH  |   METRIC   | UNITS  |   SUM    |   MEAN   |   MIN    |   MAX    |   VAR    |  STDDEV  | % SELF |
|--------------------------|--------|--------|------------|--------|----------|----------|----------|----------|----------|----------|--------|
| |0>>> main               |      1 |      0 | wall_clock | sec    | 2.306350 | 2.306350 | 2.306350 | 2.306350 | 0.000000 | 0.000000 |   86.8 |
| |0>>> |_MPI_Init_thread  |      1 |      1 | wall_clock | sec    | 0.293291 | 0.293291 | 0.293291 | 0.293291 | 0.000000 | 0.000000 |   99.2 |
| |0>>>   |_mbind          |     10 |      2 | wall_clock | sec    | 0.000014 | 0.000001 | 0.000001 | 0.000004 | 0.000000 | 0.000001 |  100.0 |
| |0>>>   |_pthread_create |      2 |      2 | wall_clock | sec    | 0.002338 | 0.001169 | 0.000897 | 0.001441 | 0.000000 | 0.000384 |    0.0 |
| |2>>>     |_start_thread |      1 |      3 | wall_clock | sec    | 0.193902 | 0.193902 | 0.193902 | 0.193902 | 0.000000 | 0.000000 |  100.0 |
| |1>>>     |_start_thread | -      |      3 | -          | -      | -        | -        | -        | -        | -        | -        | -      |
| |0>>> |_pthread_create   |      1 |      1 | wall_clock | sec    | 0.006592 | 0.006592 | 0.006592 | 0.006592 | 0.000000 | 0.000000 |    0.0 |
| |3>>>   |_start_thread   |      1 |      2 | wall_clock | sec    | 0.007850 | 0.007850 | 0.007850 | 0.007850 | 0.000000 | 0.000000 |   16.4 |
| |3>>>     |_MPI_Comm_size |    13 |      3 | wall_clock | sec    | 0.000031 | 0.000002 | 0.000000 | 0.000024 | 0.000000 | 0.000007 |  100.0 |
| |3>>>     |_MPI_Comm_rank |     5 |      3 | wall_clock | sec    | 0.000009 | 0.000002 | 0.000000 | 0.000006 | 0.000000 | 0.000002 |  100.0 |
| |3>>>     |_MPI_Barrier  |      6 |      3 | wall_clock | sec    | 0.006405 | 0.001068 | 0.000001 | 0.005604 | 0.000005 | 0.002244 |  100.0 |
| |3>>>     |_MPI_Send     |      8 |      3 | wall_clock | sec    | 0.000020 | 0.000003 | 0.000001 | 0.000012 | 0.000000 | 0.000004 |  100.0 |
| |3>>>     |_MPI_Recv     |      8 |      3 | wall_clock | sec    | 0.000027 | 0.000003 | 0.000002 | 0.000009 | 0.000000 | 0.000002 |  100.0 |
| |3>>>     |_MPI_Alltoall |      8 |      3 | wall_clock | sec    | 0.000060 | 0.000007 | 0.000003 | 0.000011 | 0.000000 | 0.000003 |  100.0 |
| |3>>>     |_MPI_Comm_dup |      1 |      3 | wall_clock | sec    | 0.000008 | 0.000008 | 0.000008 | 0.000008 | 0.000000 | 0.000000 |  100.0 |
| |0>>> |_pthread_join     |      2 |      1 | wall_clock | sec    | 0.005277 | 0.002638 | 0.001800 | 0.003477 | 0.000001 | 0.001186 |  100.0 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|
```

Visualizing - one Perfetto per MPI process or combined

MPI 0



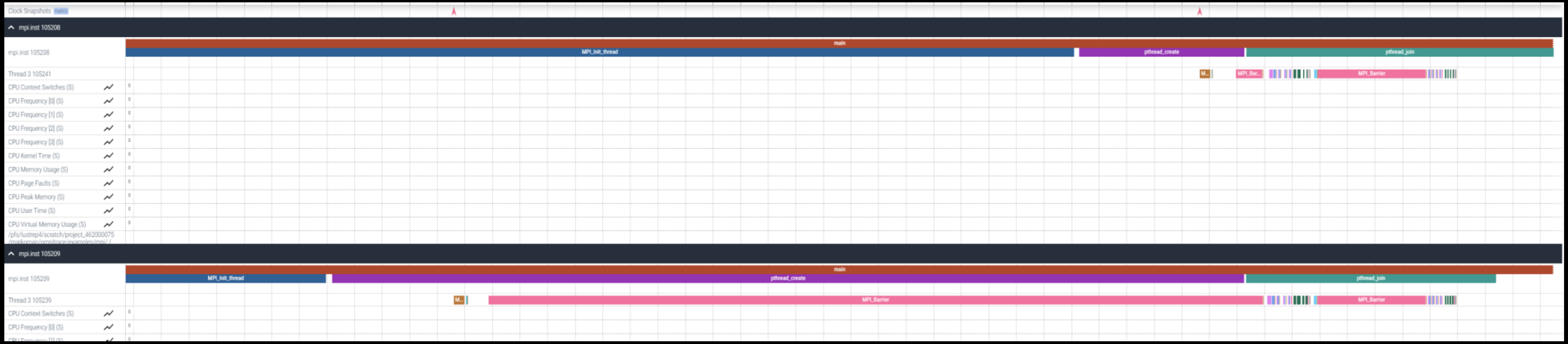
MPI 1



Merge Perfetto

Use the following command to merge and concatenate multiple traces:

```
$ cat perfetto-trace-0.proto perfetto-trace-1.proto > allprocesses.proto
```



OpenMP®

We use the example /omnitrace/examples/openmp/

Build the code with CMake:

```
$ cmake-B build
```

Use the openmp-lu binary, which can be executed with:

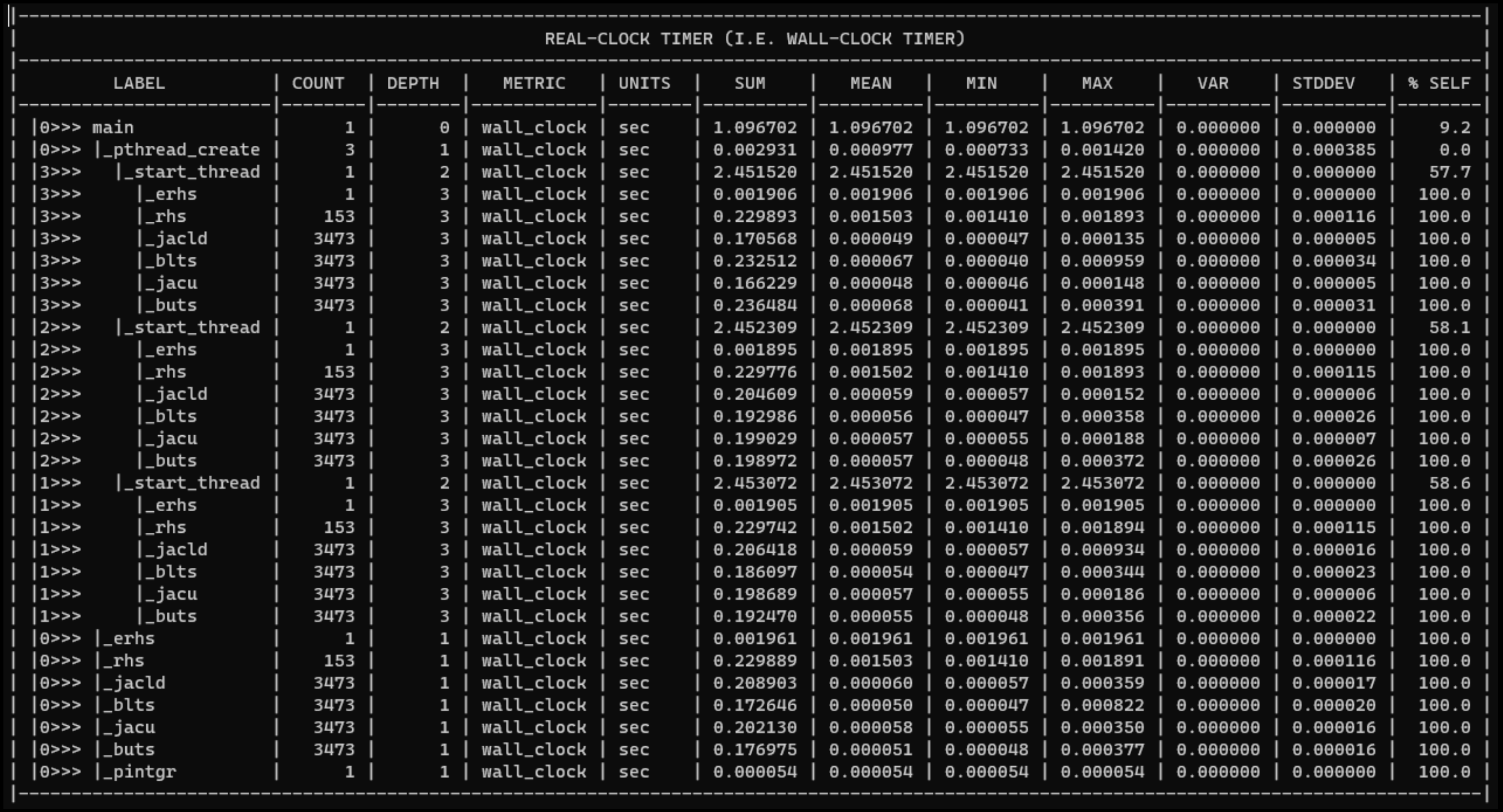
```
$ export OPENMP_NUM_THREADS=4
$ srun -n 1 -c 4 ./openmp-lu
```

Create a new instrumented binary:

```
$ srun -n 1 omnitrace -o openmp-lu.inst --
./openmp-lu
```

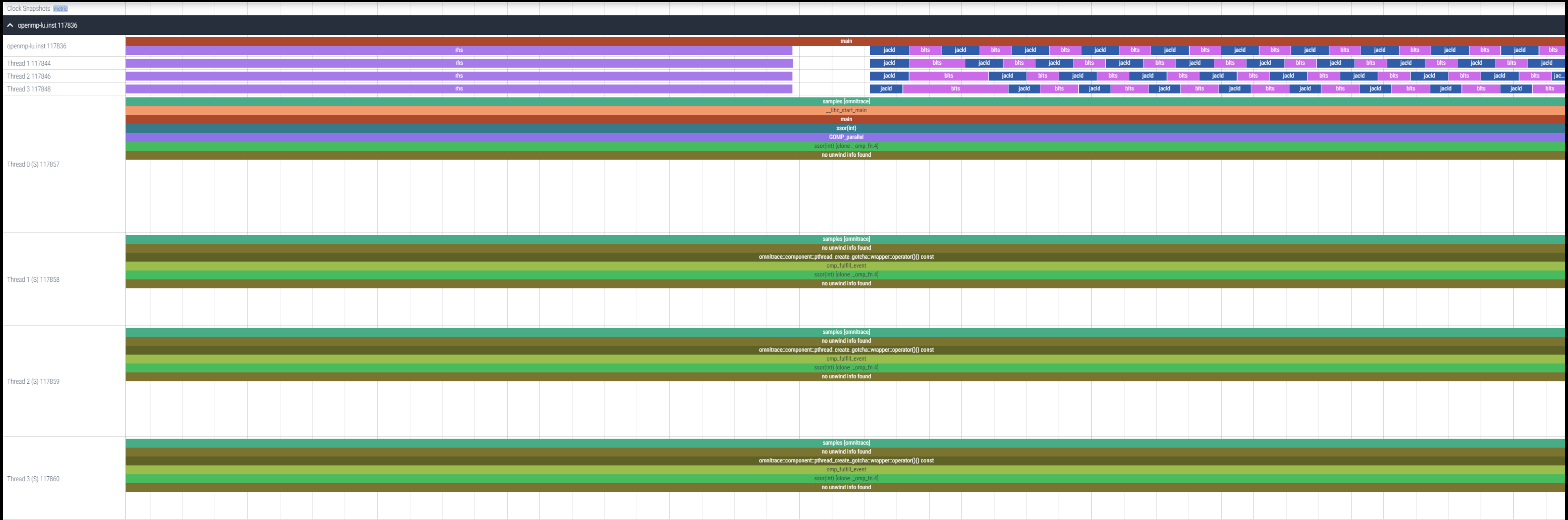
Execute the new binary:

```
$ srun -n 1 -c 4 ./openmp-lu.inst
```



```
|-----------------------------------------------------------------------------------------------------------------------------------------------|
|                                                     REAL-CLOCK TIMER (I.E. WALL-CLOCK TIMER)                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------|
|          LABEL           | COUNT  | DEPTH  |   METRIC   | UNITS  |   SUM    |   MEAN   |   MIN    |   MAX    |   VAR    |  STDDEV  | % SELF |
|--------------------------|--------|--------|------------|--------|----------|----------|----------|----------|----------|----------|--------|
| |0>>> main               |      1 |      0 | wall_clock | sec    | 1.096702 | 1.096702 | 1.096702 | 1.096702 | 0.000000 | 0.000000 |    9.2 |
| |0>>> |_pthread_create   |      3 |      1 | wall_clock | sec    | 0.002931 | 0.000977 | 0.000733 | 0.001420 | 0.000000 | 0.000385 |    0.0 |
| |3>>>   |_start_thread   |      1 |      2 | wall_clock | sec    | 2.451520 | 2.451520 | 2.451520 | 2.451520 | 0.000000 | 0.000000 |   57.7 |
| |3>>>     |_erhs         |      1 |      3 | wall_clock | sec    | 0.001906 | 0.001906 | 0.001906 | 0.001906 | 0.000000 | 0.000000 |  100.0 |
| |3>>>     |_rhs          |    153 |      3 | wall_clock | sec    | 0.229893 | 0.001503 | 0.001410 | 0.001893 | 0.000000 | 0.000116 |  100.0 |
| |3>>>     |_jacld        |   3473 |      3 | wall_clock | sec    | 0.170568 | 0.000049 | 0.000047 | 0.000135 | 0.000000 | 0.000005 |  100.0 |
| |3>>>     |_blts         |   3473 |      3 | wall_clock | sec    | 0.232512 | 0.000067 | 0.000040 | 0.000959 | 0.000000 | 0.000034 |  100.0 |
| |3>>>     |_jacu         |   3473 |      3 | wall_clock | sec    | 0.166229 | 0.000048 | 0.000046 | 0.000148 | 0.000000 | 0.000005 |  100.0 |
| |3>>>     |_buts         |   3473 |      3 | wall_clock | sec    | 0.236484 | 0.000068 | 0.000041 | 0.000391 | 0.000000 | 0.000031 |  100.0 |
| |2>>>   |_start_thread   |      1 |      2 | wall_clock | sec    | 2.452309 | 2.452309 | 2.452309 | 2.452309 | 0.000000 | 0.000000 |   58.1 |
| |2>>>     |_erhs         |      1 |      3 | wall_clock | sec    | 0.001895 | 0.001895 | 0.001895 | 0.001895 | 0.000000 | 0.000000 |  100.0 |
| |2>>>     |_rhs          |    153 |      3 | wall_clock | sec    | 0.229776 | 0.001502 | 0.001410 | 0.001893 | 0.000000 | 0.000115 |  100.0 |
| |2>>>     |_jacld        |   3473 |      3 | wall_clock | sec    | 0.204609 | 0.000059 | 0.000057 | 0.000152 | 0.000000 | 0.000006 |  100.0 |
| |2>>>     |_blts         |   3473 |      3 | wall_clock | sec    | 0.192986 | 0.000056 | 0.000047 | 0.000358 | 0.000000 | 0.000026 |  100.0 |
| |2>>>     |_jacu         |   3473 |      3 | wall_clock | sec    | 0.199029 | 0.000057 | 0.000055 | 0.000188 | 0.000000 | 0.000007 |  100.0 |
| |2>>>     |_buts         |   3473 |      3 | wall_clock | sec    | 0.198972 | 0.000057 | 0.000048 | 0.000372 | 0.000000 | 0.000026 |  100.0 |
| |1>>>   |_start_thread   |      1 |      2 | wall_clock | sec    | 2.453072 | 2.453072 | 2.453072 | 2.453072 | 0.000000 | 0.000000 |   58.6 |
| |1>>>     |_erhs         |      1 |      3 | wall_clock | sec    | 0.001905 | 0.001905 | 0.001905 | 0.001905 | 0.000000 | 0.000000 |  100.0 |
| |1>>>     |_rhs          |    153 |      3 | wall_clock | sec    | 0.229742 | 0.001502 | 0.001410 | 0.001894 | 0.000000 | 0.000115 |  100.0 |
| |1>>>     |_jacld        |   3473 |      3 | wall_clock | sec    | 0.206418 | 0.000059 | 0.000057 | 0.000934 | 0.000000 | 0.000016 |  100.0 |
| |1>>>     |_blts         |   3473 |      3 | wall_clock | sec    | 0.186097 | 0.000054 | 0.000047 | 0.000344 | 0.000000 | 0.000023 |  100.0 |
| |1>>>     |_jacu         |   3473 |      3 | wall_clock | sec    | 0.198689 | 0.000057 | 0.000055 | 0.000186 | 0.000000 | 0.000006 |  100.0 |
| |1>>>     |_buts         |   3473 |      3 | wall_clock | sec    | 0.192470 | 0.000055 | 0.000048 | 0.000356 | 0.000000 | 0.000022 |  100.0 |
| |0>>> |_erhs             |      1 |      1 | wall_clock | sec    | 0.001961 | 0.001961 | 0.001961 | 0.001961 | 0.000000 | 0.000000 |  100.0 |
| |0>>> |_rhs              |    153 |      1 | wall_clock | sec    | 0.229889 | 0.001503 | 0.001410 | 0.001891 | 0.000000 | 0.000116 |  100.0 |
| |0>>> |_jacld            |   3473 |      1 | wall_clock | sec    | 0.208903 | 0.000060 | 0.000057 | 0.000359 | 0.000000 | 0.000017 |  100.0 |
| |0>>> |_blts             |   3473 |      1 | wall_clock | sec    | 0.172646 | 0.000050 | 0.000047 | 0.000822 | 0.000000 | 0.000020 |  100.0 |
| |0>>> |_jacu             |   3473 |      1 | wall_clock | sec    | 0.202130 | 0.000058 | 0.000055 | 0.000350 | 0.000000 | 0.000016 |  100.0 |
| |0>>> |_buts             |   3473 |      1 | wall_clock | sec    | 0.176975 | 0.000051 | 0.000048 | 0.000377 | 0.000000 | 0.000016 |  100.0 |
| |0>>> |_pintgr           |      1 |      1 | wall_clock | sec    | 0.000054 | 0.000054 | 0.000054 | 0.000054 | 0.000000 | 0.000000 |  100.0 |
|-----------------------------------------------------------------------------------------------------------------------------------------------|
```

OpenMP® visualization



Python™

The omnitrace Python package is installed in /path/omnitrace_install/lib/pythonX.Y/site-packages/omnitrace

Setup the environment:

```
$ export PYTHONPATH=/path/omnitrace/lib/python/site-packages/:${PYTHONPATH}
```

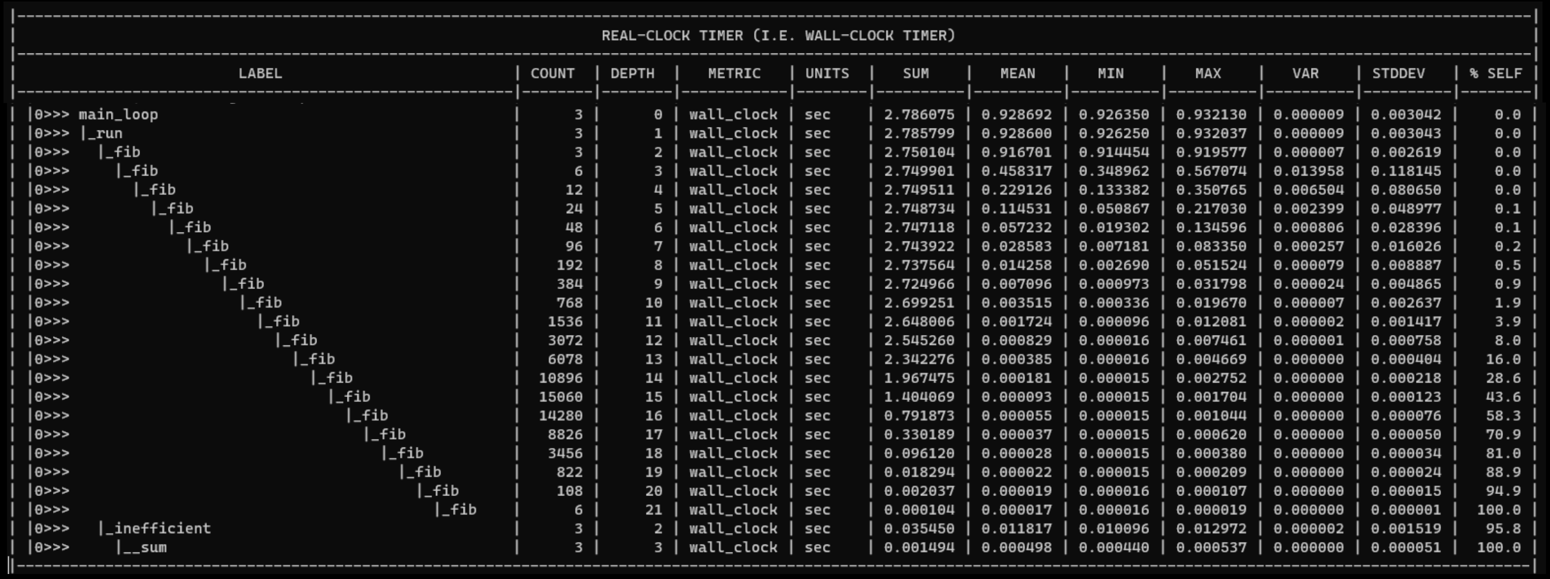
We use the Fibonacci example in: omnitrace/examples/python/source.py

Execute the python program with:

```
$ omnitrace-python ./external.py
```

Profiled data is dumped in output directory

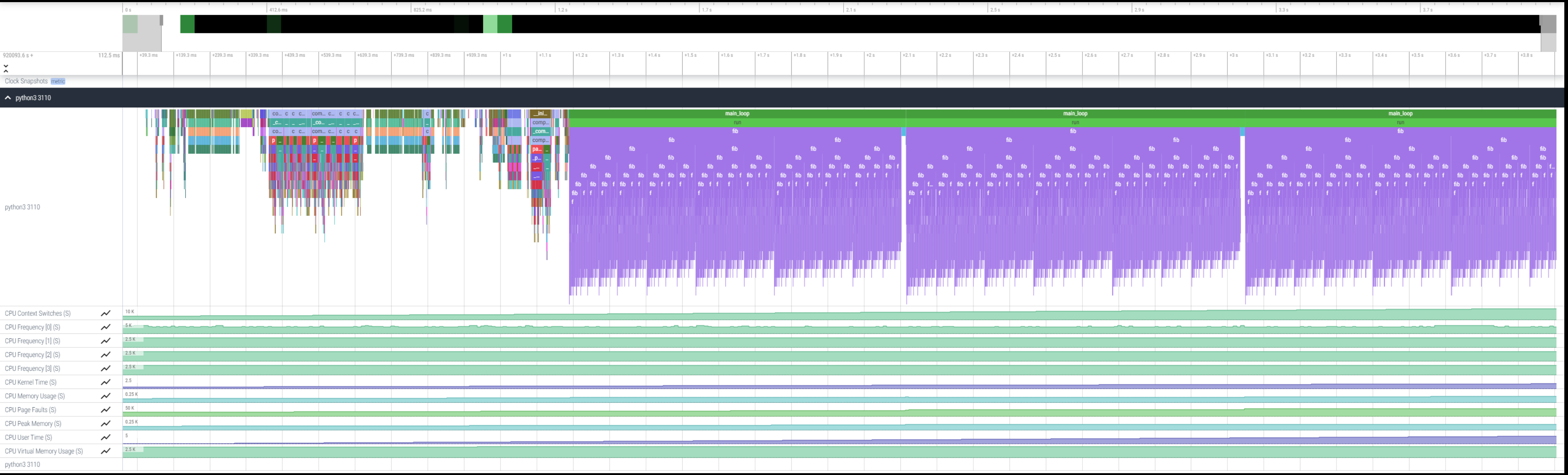
```
$ cat omnitrace-source-output/timestamp/wall_clock.txt
```



```
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                            REAL-CLOCK TIMER (I.E. WALL-CLOCK TIMER)                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          LABEL                           | COUNT  | DEPTH  |   METRIC   | UNITS  |   SUM    |   MEAN   |   MIN    |   MAX    |   VAR    |  STDDEV  | % SELF |
|----------------------------------------------------------|--------|--------|------------|--------|----------|----------|----------|----------|----------|----------|--------|
| |0>>> main_loop                                          |      3 |      0 | wall_clock | sec    | 2.786075 | 0.928692 | 0.926350 | 0.932130 | 0.000009 | 0.003042 |    0.0 |
| |0>>> |_run                                              |      3 |      1 | wall_clock | sec    | 2.785799 | 0.928600 | 0.926250 | 0.932037 | 0.000009 | 0.003043 |    0.0 |
| |0>>>   |_fib                                            |      3 |      2 | wall_clock | sec    | 2.750104 | 0.916701 | 0.914454 | 0.919577 | 0.000007 | 0.002619 |    0.0 |
| |0>>>     |_fib                                          |      6 |      3 | wall_clock | sec    | 2.749901 | 0.458317 | 0.348962 | 0.567074 | 0.013958 | 0.118145 |    0.0 |
| |0>>>       |_fib                                        |     12 |      4 | wall_clock | sec    | 2.749511 | 0.229126 | 0.133382 | 0.350765 | 0.006504 | 0.080650 |    0.0 |
| |0>>>         |_fib                                      |     24 |      5 | wall_clock | sec    | 2.748734 | 0.114531 | 0.050867 | 0.217030 | 0.002399 | 0.048977 |    0.1 |
| |0>>>           |_fib                                    |     48 |      6 | wall_clock | sec    | 2.747118 | 0.057232 | 0.019302 | 0.134596 | 0.000806 | 0.028396 |    0.1 |
| |0>>>             |_fib                                  |     96 |      7 | wall_clock | sec    | 2.743922 | 0.028583 | 0.007181 | 0.083350 | 0.000257 | 0.016026 |    0.2 |
| |0>>>               |_fib                                |    192 |      8 | wall_clock | sec    | 2.737564 | 0.014258 | 0.002690 | 0.051524 | 0.000079 | 0.008887 |    0.5 |
| |0>>>                 |_fib                              |    384 |      9 | wall_clock | sec    | 2.724966 | 0.007096 | 0.000973 | 0.031798 | 0.000024 | 0.004865 |    0.9 |
| |0>>>                   |_fib                            |    768 |     10 | wall_clock | sec    | 2.699251 | 0.003515 | 0.000336 | 0.019670 | 0.000007 | 0.002637 |    1.9 |
| |0>>>                     |_fib                          |   1536 |     11 | wall_clock | sec    | 2.648006 | 0.001724 | 0.000096 | 0.012081 | 0.000002 | 0.001417 |    3.9 |
| |0>>>                       |_fib                        |   3072 |     12 | wall_clock | sec    | 2.545260 | 0.000829 | 0.000016 | 0.007461 | 0.000001 | 0.000758 |    8.0 |
| |0>>>                         |_fib                      |   6078 |     13 | wall_clock | sec    | 2.342276 | 0.000385 | 0.000016 | 0.004669 | 0.000000 | 0.000404 |   16.0 |
| |0>>>                           |_fib                    |  10896 |     14 | wall_clock | sec    | 1.967475 | 0.000181 | 0.000015 | 0.002752 | 0.000000 | 0.000218 |   28.6 |
| |0>>>                             |_fib                  |  15060 |     15 | wall_clock | sec    | 1.404069 | 0.000093 | 0.000015 | 0.001704 | 0.000000 | 0.000123 |   43.6 |
| |0>>>                               |_fib                |  14280 |     16 | wall_clock | sec    | 0.791873 | 0.000055 | 0.000015 | 0.001044 | 0.000000 | 0.000076 |   58.3 |
| |0>>>                                 |_fib              |   8826 |     17 | wall_clock | sec    | 0.330189 | 0.000037 | 0.000015 | 0.000620 | 0.000000 | 0.000050 |   70.9 |
| |0>>>                                   |_fib            |   3456 |     18 | wall_clock | sec    | 0.096120 | 0.000028 | 0.000015 | 0.000380 | 0.000000 | 0.000034 |   81.0 |
| |0>>>                                     |_fib          |    822 |     19 | wall_clock | sec    | 0.018294 | 0.000022 | 0.000015 | 0.000209 | 0.000000 | 0.000024 |   88.9 |
| |0>>>                                       |_fib        |    108 |     20 | wall_clock | sec    | 0.002037 | 0.000019 | 0.000016 | 0.000107 | 0.000000 | 0.000015 |   94.9 |
| |0>>>                                         |_fib      |      6 |     21 | wall_clock | sec    | 0.000104 | 0.000017 | 0.000016 | 0.000019 | 0.000000 | 0.000001 |  100.0 |
| |0>>>   |_inefficient                                    |      3 |      2 | wall_clock | sec    | 0.035450 | 0.011817 | 0.010096 | 0.012972 | 0.000002 | 0.001519 |   95.8 |
| |0>>>     |__sum                                         |      3 |      3 | wall_clock | sec    | 0.001494 | 0.000498 | 0.000440 | 0.000537 | 0.000000 | 0.000051 |  100.0 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
```

Python documentation: https://amdresearch.github.io/omnitrace/python.html

Visualizing Python™ Perfetto tracing



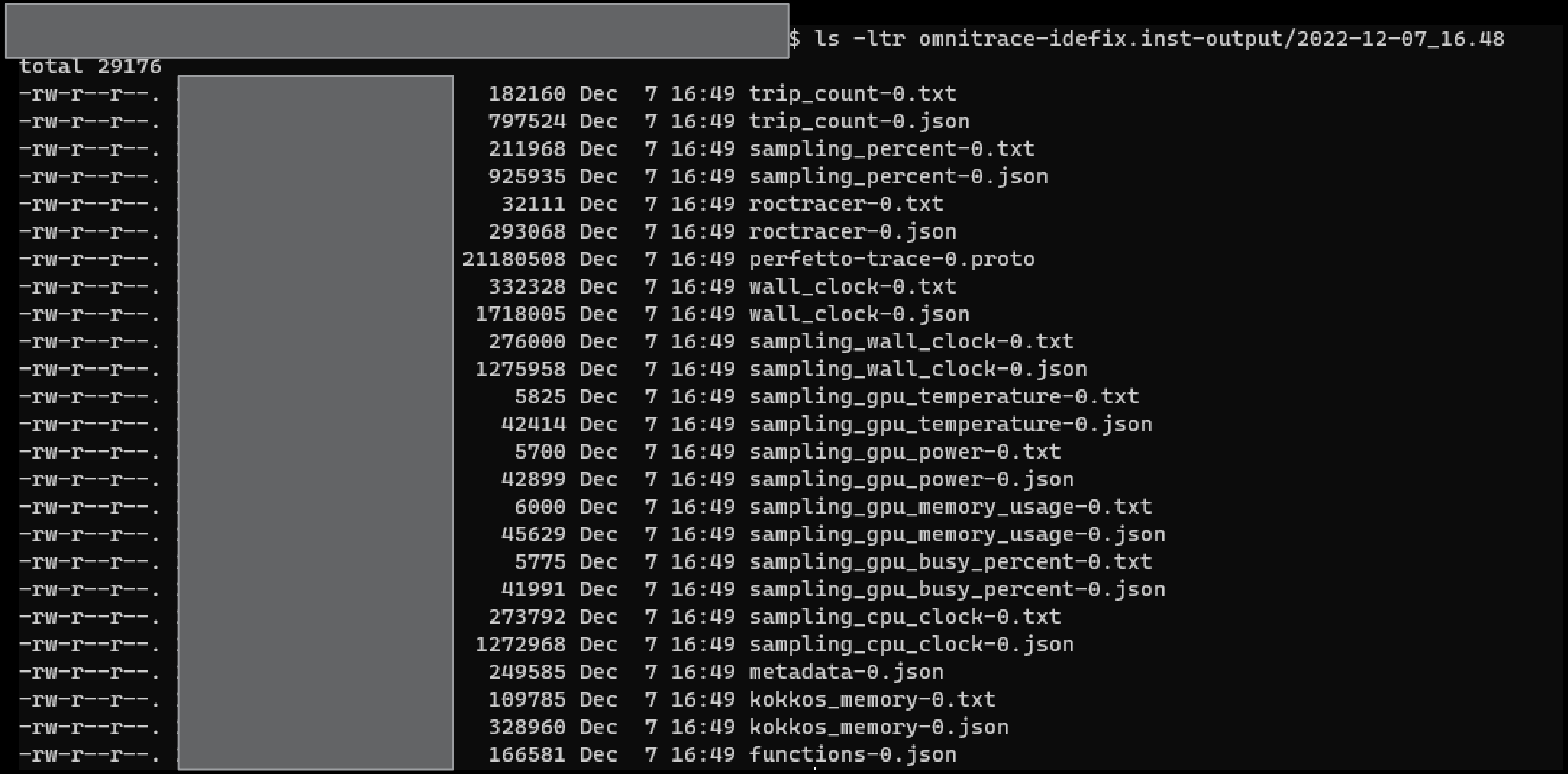
AMD
together we advance_

Kokkos

Omnitrace can instrument Kokkos applications too

Edit the $HOME/.omnitrace.cfg file and enable omnitrace:

```
...
OMNITRACE_USE_KOKKOSP = true
...
```

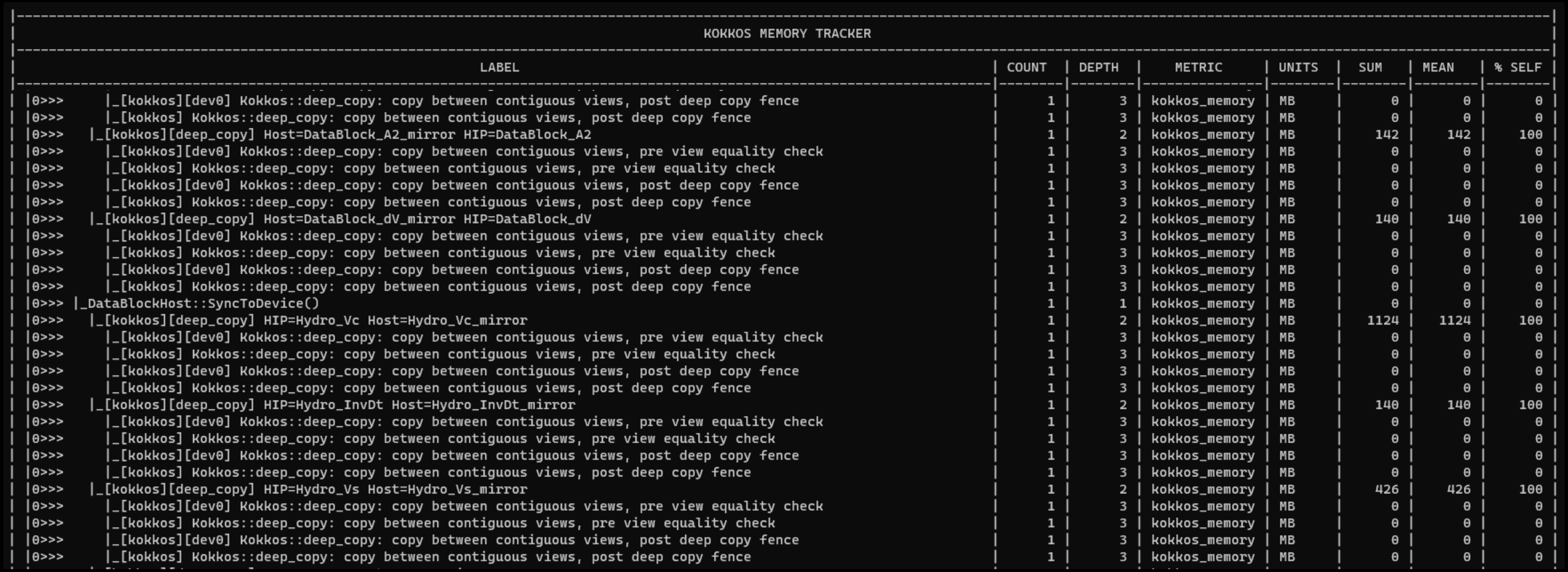


```
[redacted]$ ls -ltr omnitrace-idefix.inst-output/2022-12-07_16.48
total 29176
-rw-r--r--. [redacted]  182160 Dec  7 16:49 trip_count-0.txt
-rw-r--r--.             797524 Dec  7 16:49 trip_count-0.json
-rw-r--r--.             211968 Dec  7 16:49 sampling_percent-0.txt
-rw-r--r--.             925935 Dec  7 16:49 sampling_percent-0.json
-rw-r--r--.              32111 Dec  7 16:49 roctracer-0.txt
-rw-r--r--.             293068 Dec  7 16:49 roctracer-0.json
-rw-r--r--.           21180508 Dec  7 16:49 perfetto-trace-0.proto
-rw-r--r--.             332328 Dec  7 16:49 wall_clock-0.txt
-rw-r--r--.            1718005 Dec  7 16:49 wall_clock-0.json
-rw-r--r--.             276000 Dec  7 16:49 sampling_wall_clock-0.txt
-rw-r--r--.            1275958 Dec  7 16:49 sampling_wall_clock-0.json
-rw-r--r--.               5825 Dec  7 16:49 sampling_gpu_temperature-0.txt
-rw-r--r--.              42414 Dec  7 16:49 sampling_gpu_temperature-0.json
-rw-r--r--.               5700 Dec  7 16:49 sampling_gpu_power-0.txt
-rw-r--r--.              42899 Dec  7 16:49 sampling_gpu_power-0.json
-rw-r--r--.               6000 Dec  7 16:49 sampling_gpu_memory_usage-0.txt
-rw-r--r--.              45629 Dec  7 16:49 sampling_gpu_memory_usage-0.json
-rw-r--r--.               5775 Dec  7 16:49 sampling_gpu_busy_percent-0.txt
-rw-r--r--.              41991 Dec  7 16:49 sampling_gpu_busy_percent-0.json
-rw-r--r--.             273792 Dec  7 16:49 sampling_cpu_clock-0.txt
-rw-r--r--.            1272968 Dec  7 16:49 sampling_cpu_clock-0.json
-rw-r--r--.             249585 Dec  7 16:49 metadata-0.json
-rw-r--r--.             109785 Dec  7 16:49 kokkos_memory-0.txt
-rw-r--r--.             328960 Dec  7 16:49 kokkos_memory-0.json
-rw-r--r--.             166581 Dec  7 16:49 functions-0.json
```

AMD
together we advance_

Kokkos

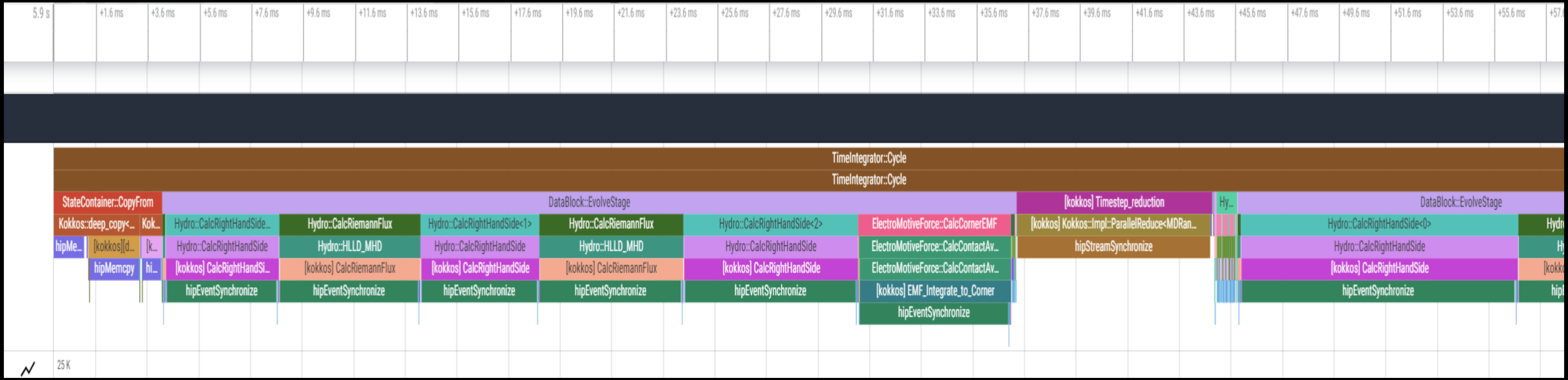
```
$ cat kokkos_memory0.txt
```



```
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                     KOKKOS MEMORY TRACKER                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                          LABEL                                                          | COUNT  | DEPTH  |    METRIC     | UNITS  | SUM    | MEAN   | % SELF |
|-------------------------------------------------------------------------------------------------------------------------|--------|--------|---------------|--------|--------|--------|--------|
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                       |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                             |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>   |_[kokkos][deep_copy] Host=DataBlock_A2_mirror HIP=DataBlock_A2                                                 |      1 |      2 | kokkos_memory | MB     |    142 |    142 |    100 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, pre view equality check                    |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, pre view equality check                          |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                       |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                             |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>   |_[kokkos][deep_copy] Host=DataBlock_dV_mirror HIP=DataBlock_dV                                                 |      1 |      2 | kokkos_memory | MB     |    140 |    140 |    100 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, pre view equality check                    |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, pre view equality check                          |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                       |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                             |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>> |_DataBlockHost::SyncToDevice()                                                                                   |      1 |      1 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>   |_[kokkos][deep_copy] HIP=Hydro_Vc Host=Hydro_Vc_mirror                                                         |      1 |      2 | kokkos_memory | MB     |   1124 |   1124 |    100 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, pre view equality check                    |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, pre view equality check                          |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                       |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                             |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>   |_[kokkos][deep_copy] HIP=Hydro_InvDt Host=Hydro_InvDt_mirror                                                   |      1 |      2 | kokkos_memory | MB     |    140 |    140 |    100 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, pre view equality check                    |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, pre view equality check                          |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                       |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                             |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>   |_[kokkos][deep_copy] HIP=Hydro_Vs Host=Hydro_Vs_mirror                                                         |      1 |      2 | kokkos_memory | MB     |    426 |    426 |    100 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, pre view equality check                    |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, pre view equality check                          |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos][dev0] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                       |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
| |0>>>     |_[kokkos] Kokkos::deep_copy: copy between contiguous views, post deep copy fence                             |      1 |      3 | kokkos_memory | MB     |      0 |      0 |      0 |
```

Visualizing Kokkos with Perfetto trace

- Visualize perfetto-trace-0.proto (with sampling enabled)



Omnitrace-sample

- For easy usage of Omnitrace there is also the omnitrace-sample that does sampling with less overhead.
- It provides less overhead but you need to be sure that you do not miss information
- Not all the declarations of a cfg file apply, for example to use hardware counters, ou need to execute the following command:

```
srun -n 1 omnitrace-sample -TPHD -G
"GPUBusy:device=0,Wavefronts:device=0,VALUBusy:device=0,L2CacheHit:device=0,MemUnitBusy:device=0” -- ./binary
```

See omnitrace-sample -h for more information

Tips & Tricks

- My Perfetto timeline seems weird how can I check the clock skew?
  - OMNITRACE_VERBOSE equal to 1 or higher for verbose mode and it will print the timestamp skew
- Omnitrace takes too long time in the finalization, how to check which part takes a lot of time?
  - Use OMNITRACE_VERBOSE equal to 1 or higher for verbose mode
- It takes too long time to map rocm-smi samples to the kernels
  - Use temporarily OMNITRACE_USE_ROCM_SMI=OFF
- If you are doing binary rewriting and you do not get information about kernels, declare:
  - HSA_TOOLS_LIB=libomnitrace.so in the environment and be sure that OMNITRACE_USE_ROCTRACER=ON in the cfg file
- My HIP application hangs in different points, what to do?
  - Try to set HSA_ENABLE_INTERRUPT=0 in the environment, this handles different how HIP is notified that GPU kernels completed
- It is preferred to use binary rewriting for MPI applications, in order to write one file per MPI process, and not aggregated, use: OMNITRACE_USE_PID=ON
- My Perfetto trace is too big, can I decrease it?
  - Yes, with v1.7.3 and later declare OMNITRACE_PERFETTO_ANNOTATIONS to false.
- Full documentation: https://amdresearch.github.io/omnitrace/

Omniperf

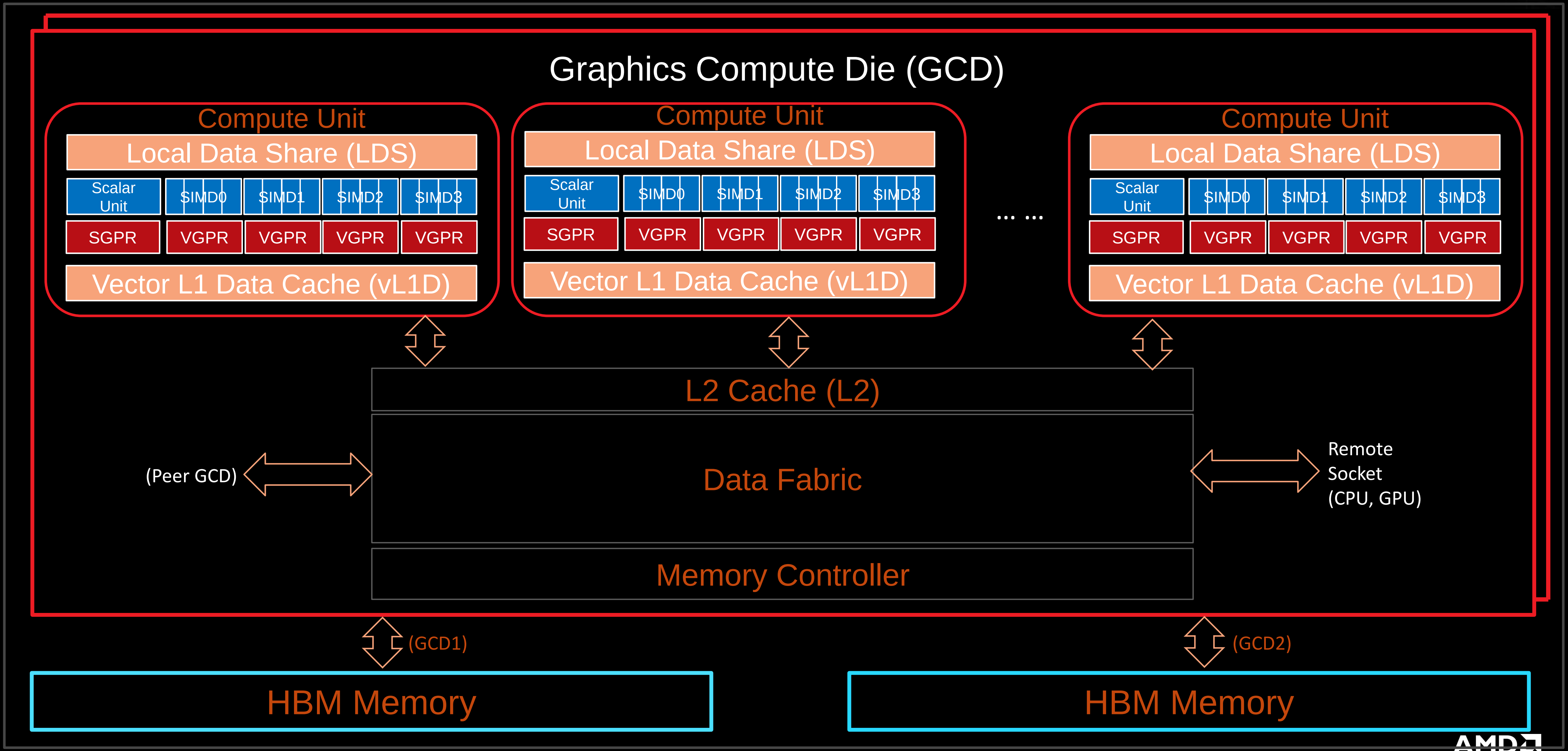
AMD

Omniperf

- The Omniperf executes the code as many times required based on the job submission
- Without specific option the application will be executed many times with various hardware counters (more than 100), so this can take long time. It does not mean that all the counters will provide useful data for a specific code.
- There are various options for filtering (kernel, metric) even to execute mainly for roofline analysis, roofline is supported only for MI200 GPU series.
- There are many data per metric/HW and we will show a few, Omniperf provides tables for every metric
- With Omniperf first we profile, then we analyze and then we can import to database or visualize with standalone GUI
- The Omniperf targets MI100 and MI200 and later future generation AMD GPUs

- For problems, create an issue here: https://github.com/AMDResearch/omniperf/issues

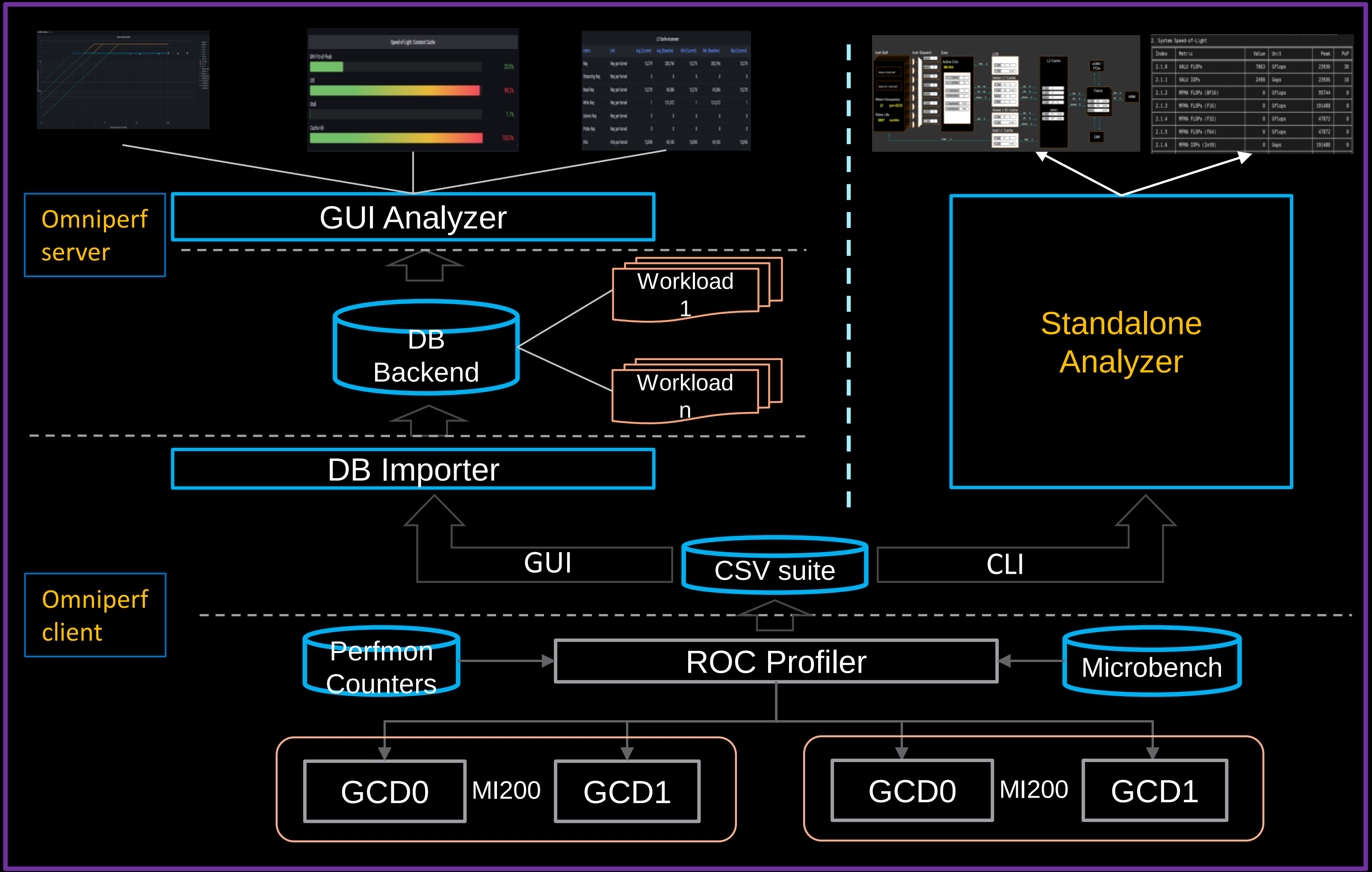
Overview - AMD Instinct™ MI200 Architecture



together we advance_

Performance Analysis on MI200 GPUs - Omniperf

- Opensource github repos
  - *https://github.com/AMDResearch/omniperf*
- Built on top of ROC Profiler
- Integrated Performance Analyzer for AMD GPUs
  - Roofline Analyzer
  - Mem Chart Analyzer
  - Speed-of-Light
  - Baseline Comparison
  - Shared Workload Database
  - Flexible Filtering and Normalization
  - Comprehensive Profiling
    - Wavefront Dispatching
    - Shader Compute
    - Local Data Share (LDS) Accesses
    - L1/L2 Cache Accesses
    - HBM Accesses
- User Interfaces
  - Grafana™ Based GUI
  - Standalone GUI



Empirical Hierarchical Roofline on MI200 - Perfmon Counters

- Weight
  - ADD: 1
  - MUL: 1
  - FMA: 2
  - Transcendental: 1
- FLOP Count
  - VALU: derived from VALU math instructions (assuming 64 active threads)
  - MFMA: count FLOP directly, in unit of 512
- Transcendental Instructions (7 in total)
  - e^x, $\log(x)$: F16, F32
  - $\frac{1}{x}$, $\sqrt{x}$, $\frac{1}{\sqrt{x}}$: F16, F32, F64
  - $\sin x$, $\cos x$: F16, F32
- Profiling Overhead
  - Require 3 application replays

v_rcp_f64_e32 v[4:5], v[2:3]

v_sin_f32_e32 v2, v2

v_cos_f32_e32 v2, v2

v_rsq_f64_e32 v[6:7], v[2:3]

v_sqrt_f32_e32 v3, v2

v_log_f32_e32 v2, v2

v_exp_f32_e32 v2, v2

ID	HW Counter	Category
1	SQ_INSTS_VALU_ADD_F16	FLOP counter
2	SQ_INSTS_VALU_MUL_F16	FLOP counter
3	SQ_INSTS_VALU_FMA_F16	FLOP counter
4	SQ_INSTS_VALU_TRANS_F16	FLOP counter
5	SQ_INSTS_VALU_ADD_F32	FLOP counter
6	SQ_INSTS_VALU_MUL_F32	FLOP counter
7	SQ_INSTS_VALU_FMA_F32	FLOP counter
8	SQ_INSTS_VALU_TRANS_F32	FLOP counter
9	SQ_INSTS_VALU_ADD_F64	FLOP counter
10	SQ_INSTS_VALU_MUL_F64	FLOP counter
11	SQ_INSTS_VALU_FMA_F64	FLOP counter
12	SQ_INSTS_VALU_TRANS_F64	FLOP counter
13	SQ_INSTS_VALU_INT32	IOP counter
14	SQ_INSTS_VALU_INT64	IOP counter
15	SQ_INSTS_VALU_MFMA_MOPS_I8	IOP counter
16	SQ_INSTS_VALU_MFMA_MOPS_F16	FLOP counter
17	SQ_INSTS_VALU_MFMA_MOPS_BF16	FLOP counter
18	SQ_INSTS_VALU_MFMA_MOPS_F32	FLOP counter
19	SQ_INSTS_VALU_MFMA_MOPS_F64	FLOP counter
20	SQ_LDS_IDX_ACTIVE	LDS Bandwidth
21	SQ_LDS_BANK_CONFLICT	LDS Bandwidth
22	TCP_TOTAL_CACHE_ACCESSES_sum	vL1D Bandwidth
23	TCP_TCC_WRITE_REQ_sum	L2 Bandwidth
24	TCP_TCC_ATOMIC_WITH_RET_REQ_sum	L2 Bandwidth
25	TCP_TCC_ATOMIC_WITHOUT_RET_REQ_sum	L2 Bandwidth
26	TCP_TCC_READ_REQ_sum	L2 Bandwidth
27	TCC_EA_RDREQ_sum	HBM Bandwidth
28	TCC_EA_RDREQ_32B_sum	HBM Bandwidth
29	TCC_EA_WRREQ_sum	HBM Bandwidth
30	TCC_EA_WRREQ_64B_sum	HBM Bandwidth

Empirical Hierarchical Roofline on MI200 - Arithmetic

Total_FLOP = 64 ∗ (SQ_INSTS_VALU_ADD_F16 + SQ_INSTS_VALU_MUL_F16 + SQ_INSTS_VALU_TRANS_F16 + 2 ∗ SQ_INSTS_VALU_FMA_F16)
+ 64 ∗ (SQ_INSTS_VALU_ADD_F32 + SQ_INSTS_VALU_MUL_F32 + SQ_INSTS_VALU_TRANS_F32 + 2 ∗ SQ_INSTS_VALU_FMA_F32)
+ 64 ∗ (SQ_INSTS_VALU_ADD_F64 + SQ_INSTS_VALU_MUL_F64 + SQ_INSTS_VALU_TRANS_F64 + 2 ∗ SQ_INSTS_VALU_FMA_F64)
+ 512 * SQ_INSTS_VALU_MFMA_MOPS_F16
+ 512 * SQ_INSTS_VALU_MFMA_MOPS_BF16
+ 512 * SQ_INSTS_VALU_MFMA_MOPS_F32
+ 512 * SQ_INSTS_VALU_MFMA_MOPS_F64

Total_IOP = 64 ∗ (SQ_INSTS_VALU_INT32 + SQ_INSTS_VALU_INT64)

LDS_{BW} = 32 ∗ 4 ∗ (SQ_LDS_IDX_ACTIVE – SQ_LDS_BANK_CONFLICT)

$vL1D_{BW}$ = 64 ∗ TCP_TOTAL_CACHE_ACCESSES_sum

$L2_{BW}$ = 64 ∗ TCP_TCC_READ_REQ_sum
+ 64 * TCP_TCC_WRITE_REQ_sum
+ 64 * (TCP_TCC_ATOMIC_WITH_RET_REQ_sum + TCP_TCC_ATOMIC_WITHOUT_RET_REQ_sum)

HBM_{BW} = 32 * TCC_EA_RDREQ_32B_sum + 64 * (TCC_EA_RDREQ_sum - TCC_EA_RDREQ_32B_sum)
+ 32 * (TCC_EA_WRREQ_sum – TCC_EA_WRREQ_64B_sum) + 64 * TCC_EA_WRREQ_64B_sum

$$AI_{LDS} \frac{TOTAL\_FLOP}{LDS_{BW}}$$

$$AI_{vL1D} \frac{TOTAL\_FLOP}{vL1D_{BW}}$$

$$AI_{L2} \frac{TOTAL\_FLOP}{L2_{BW}}$$

$$AI_{HBM} = \frac{TOTAL\_FLOP}{HBM_{BW}}$$

*** All calculations are subject to change without notice***

Omniperf features

Omniperf Features	
MI200 support	Roofline Analysis Panel (*Supported on MI200 only, SLES 15 SP3 or RHEL8*)
MI100 support	Command Processor (CP) Panel
Standalone GUI Analyzer	Shader Processing Input (SPI) Panel
Grafana/MongoDB GUI Analyzer	Wavefront Launch Panel
Dispatch Filtering	Compute Unit - Instruction Mix Panel
Kernel Filtering	Compute Unit - Pipeline Panel
GPU ID Filtering	Local Data Share (LDS) Panel
Baseline Comparison	Instruction Cache Panel
Multi-Normalizations	Scalar L1D Cache Panel
System Info Panel	Texture Addresser and Data Panel
System Speed-of-Light Panel	Vector L1D Cache Panel
Kernel Statistic Panel	L2 Cache Panel
Memory Chart Analysis Panel	L2 Cache (per-Channel) Panel

Client-side installation (if required)

- Download the latest version from here: https://github.com/AMDResearch/omniperf/releases

```
wget https://github.com/AMDResearch/omniperf/releases/download/v1.0.4/omniperf-1.0.4.tar.gz

tar zxvf omniperf-1.0.4.tar.gz

cd omniperf-1.0.4/
python3 -m pip install -t ${INSTALL_DIR}/python-libs -r requirements.txt
mkdir build
cd build
export PYTHONPATH=$INSTALL_DIR/python-libs:$PYTHONPATH
cmake -DCMAKE_INSTALL_PREFIX=${INSTALL_DIR}/1.0.4 \
        -DPYTHON_DEPS=${INSTALL_DIR}/python-libs \
        -DMOD_INSTALL_PATH=${INSTALL_DIR}/modulefiles ..
make install
export PATH=$INSTALL_DIR/1.0.4/bin:$PATH
```

Omniperf modes

- Profiling

```
omniperf profile  -n workload_name [profile options] [roofline options] --
<profile_cmd>
```

- Analysis

```
omniperf analyze -p workloads/workload_name/mi200/
```

- GUI import

```
omniperf database --import [CONNECTION OPTIONS]
```

- GUI standalone

```
omniperf analyze -p workloads/workload_name/mi200/ --gui
```

Then follow the instructions to open the web page for the GUI

Omniperf Profiling

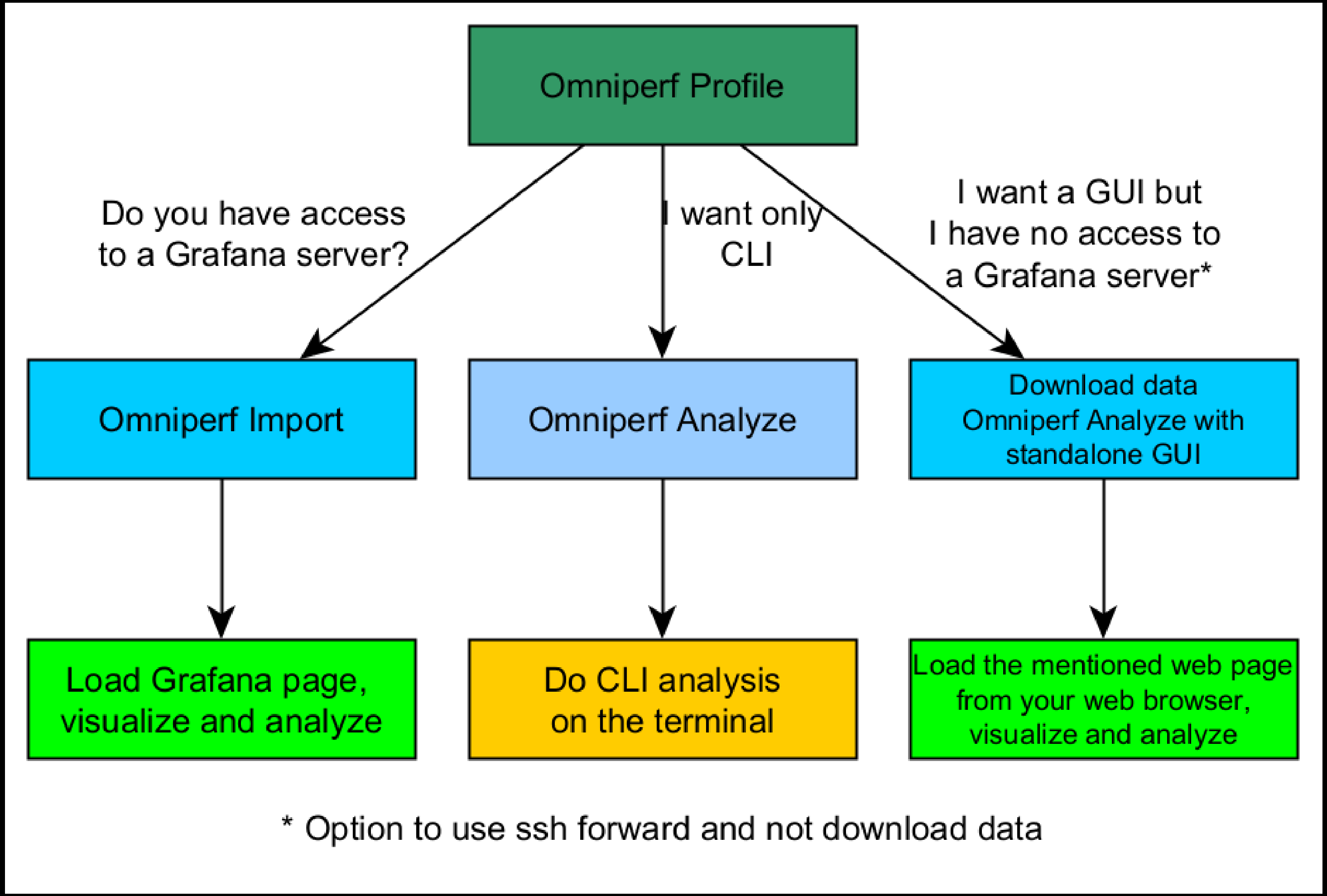
- We use the example sample/vcopy.cpp from the Omniperf installation folder (cp omniperf/1.0.4/share/sample/vcopy.cpp .)
- Compile with hipcc, let’s call the binary vcopy
- Load Omniperf module
- Profiling with the default set pf data for all kernels, execute:

```
srun -n 1 --gpus 1 omniperf profile -n vcopy_all -- ./vcopy 1048576 256
…
-------------
Profile only
-------------

omniperf ver:  1.0.4
Path:  /pfs/lustrep4/scratch/project_462000075/markoman/omniperf-1.0.4/build/workloads
Target:  mi200
Command:  ./vcopy 1048576 256
Kernel Selection:  None
Dispatch Selection:  None
IP Blocks: All
```

In this case we call the workload name “vcopy_all” and after the “--” everything is about the application we execute. In this case, the application will be executed many times for collecting different metrics, if the application takes significant time to run once, then this could b not the optimum approach.

At the end of the execution, we have a folder workloads/vcopy_all/mi200/
You can see all the options with the command `omniperf profile --help`

Omniperf workflows


Omniperf Profile
Do you have access to a Grafana server?
I want only CLI
I want a GUI but I have no access to a Grafana server*
Omniperf Import
Omniperf Analyze
Download data Omniperf Analyze with standalone GUI
Load Grafana page, visualize and analyze
Do CLI analysis on the terminal
Load the mentioned web page from your web browser, visualize and analyze

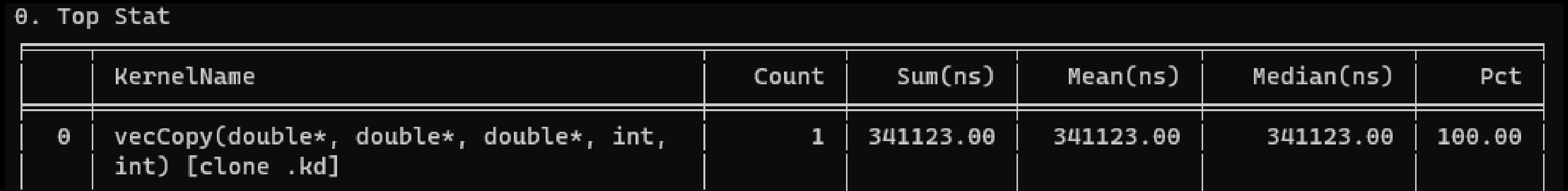
* Option to use ssh forward and not download data

The installed Omniperf version on LUMI, has disabled the --gui option, so in order to visualize, better to download the data on your laptop and install the Omniperf version with GUI support or use Grafana.

Omniperf Analyze

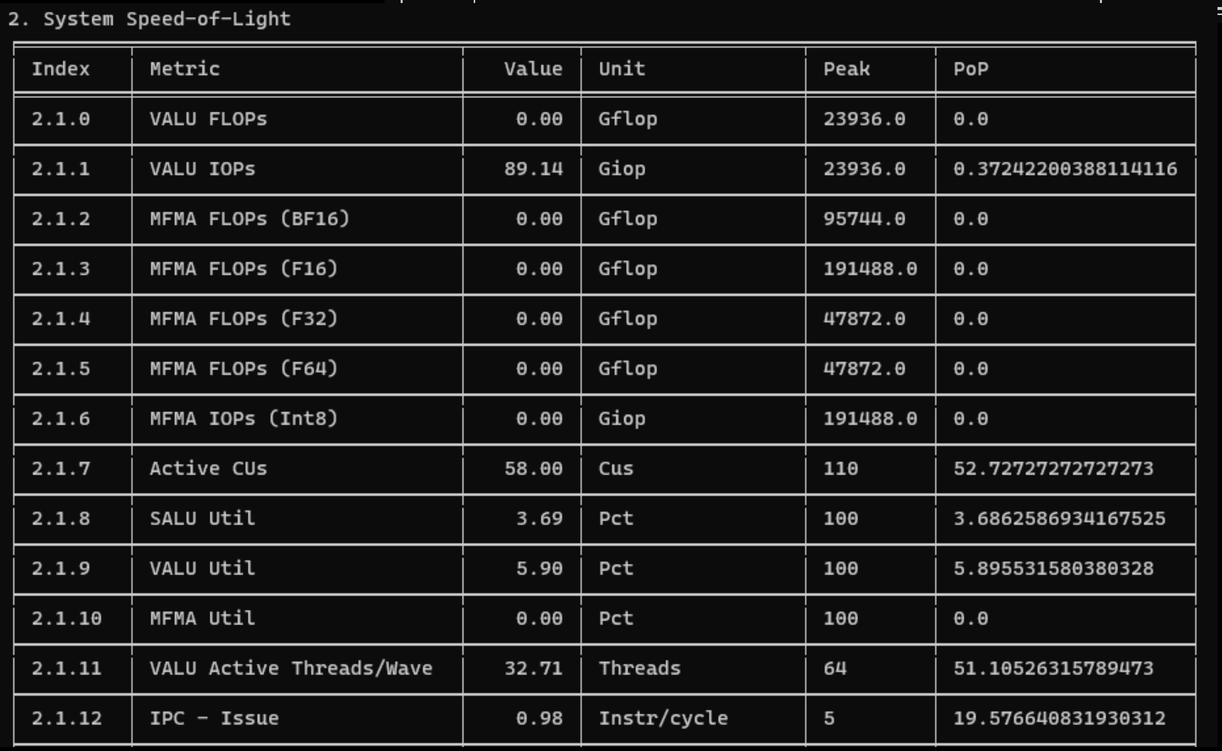
- We use the example sample/vcopy.cpp from the Omniperf installation folder

```
srun -n 1 --gpus 1 omniperf analyze -p workloads/vcopy_all/mi200/ &>
vcopy_analyze.txt
```



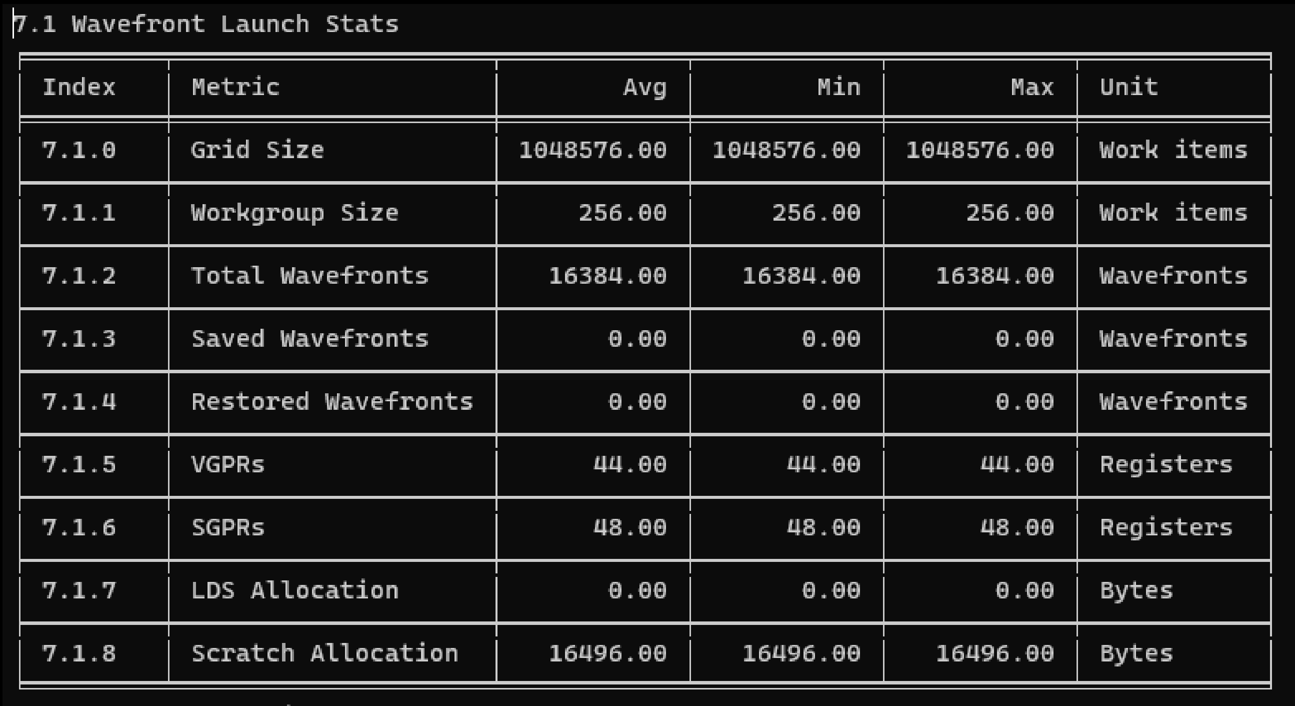
0. Top Stat

	KernelName	Count	Sum(ns)	Mean(ns)	Median(ns)	Pct
0	vecCopy(double*, double*, double*, int, int) [clone .kd]	1	341123.00	341123.00	341123.00	100.00



2. System Speed-of-Light

Index	Metric	Value	Unit	Peak	PoP
2.1.0	VALU FLOPs	0.00	Gflop	23936.0	0.0
2.1.1	VALU IOPs	89.14	Giop	23936.0	0.37242200388114116
2.1.2	MFMA FLOPs (BF16)	0.00	Gflop	95744.0	0.0
2.1.3	MFMA FLOPs (F16)	0.00	Gflop	191488.0	0.0
2.1.4	MFMA FLOPs (F32)	0.00	Gflop	47872.0	0.0
2.1.5	MFMA FLOPs (F64)	0.00	Gflop	47872.0	0.0
2.1.6	MFMA IOPs (Int8)	0.00	Giop	191488.0	0.0
2.1.7	Active CUs	58.00	Cus	110	52.72727272727273
2.1.8	SALU Util	3.69	Pct	100	3.6862586934167525
2.1.9	VALU Util	5.90	Pct	100	5.895531580380328
2.1.10	MFMA Util	0.00	Pct	100	0.0
2.1.11	VALU Active Threads/Wave	32.71	Threads	64	51.10526315789473
2.1.12	IPC - Issue	0.98	Instr/cycle	5	19.576640831930312



7.1 Wavefront Launch Stats

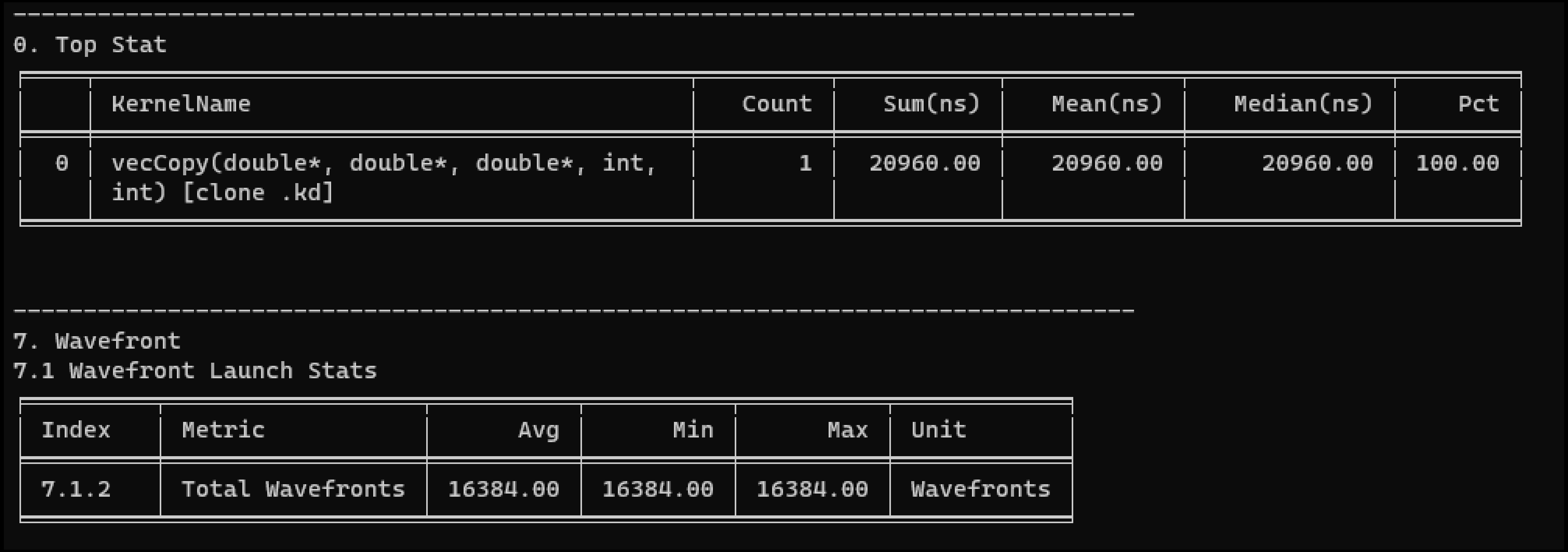
Index	Metric	Avg	Min	Max	Unit
7.1.0	Grid Size	1048576.00	1048576.00	1048576.00	Work items
7.1.1	Workgroup Size	256.00	256.00	256.00	Work items
7.1.2	Total Wavefronts	16384.00	16384.00	16384.00	Wavefronts
7.1.3	Saved Wavefronts	0.00	0.00	0.00	Wavefronts
7.1.4	Restored Wavefronts	0.00	0.00	0.00	Wavefronts
7.1.5	VGPRs	44.00	44.00	44.00	Registers
7.1.6	SGPRs	48.00	48.00	48.00	Registers
7.1.7	LDS Allocation	0.00	0.00	0.00	Bytes
7.1.8	Scratch Allocation	16496.00	16496.00	16496.00	Bytes

Omniperf Analyze (II)

- Execute omniperf analyze –h to see various options
- Use specific IP block (-b)
- Top kernel:

```
srun -n 1 --gpus 1 omniperf analyze -p workloads/vcopy_all/mi200/ -b 0
```

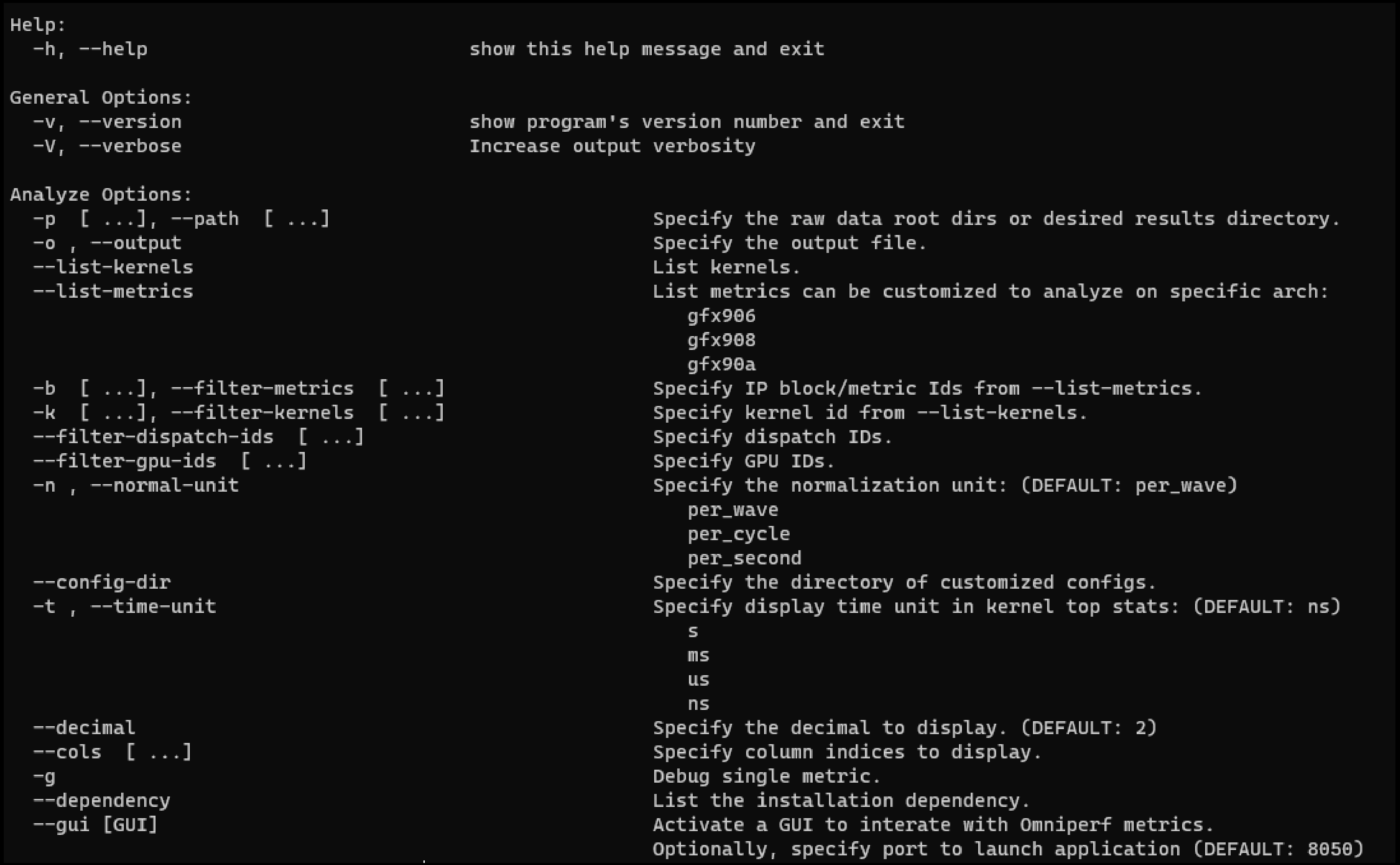
- IP Block of wavefronts: `srun -n 1 --gpus 1 omniperf analyze -p workloads/vcopy_all/mi200/ -b 7.1.2`



```
--------------------------------------------------------------------------------
0. Top Stat
╒════╤══════════════════════════════════════════╤═════════╤═══════════╤════════════╤══════════════╤════════╕
│    │ KernelName                               │   Count │   Sum(ns) │  Mean(ns)  │   Median(ns) │    Pct │
╞════╪══════════════════════════════════════════╪═════════╪═══════════╪════════════╪══════════════╪════════╡
│  0 │ vecCopy(double*, double*, double*, int,  │       1 │  20960.00 │  20960.00  │     20960.00 │ 100.00 │
│    │ int) [clone .kd]                         │         │           │            │              │        │
╘════╧══════════════════════════════════════════╧═════════╧═══════════╧════════════╧══════════════╧════════╛


--------------------------------------------------------------------------------
7. Wavefront
7.1 Wavefront Launch Stats
╒═════════╤══════════════════╤══════════╤══════════╤══════════╤════════════╕
│ Index   │ Metric           │      Avg │      Min │      Max │ Unit       │
╞═════════╪══════════════════╪══════════╪══════════╪══════════╪════════════╡
│ 7.1.2   │ Total Wavefronts │ 16384.00 │ 16384.00 │ 16384.00 │ Wavefronts │
╘═════════╧══════════════════╧══════════╧══════════╧══════════╧════════════╛
```

Omniperf Analyze (III)

omniperf analyze -h



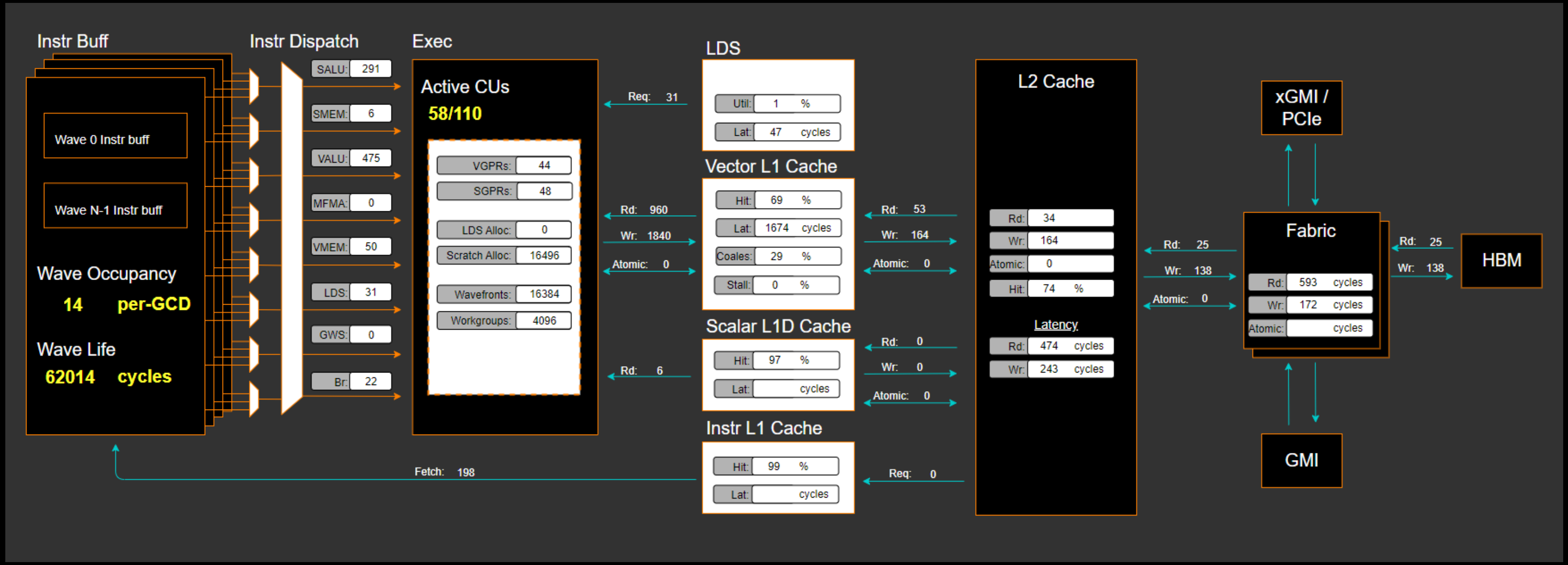
```
Help:
  -h, --help                            show this help message and exit

General Options:
  -v, --version                         show program's version number and exit
  -V, --verbose                         Increase output verbosity

Analyze Options:
  -p  [ ...], --path  [ ...]                            Specify the raw data root dirs or desired results directory.
  -o , --output                                         Specify the output file.
  --list-kernels                                        List kernels.
  --list-metrics                                        List metrics can be customized to analyze on specific arch:
                                                           gfx906
                                                           gfx908
                                                           gfx90a
  -b  [ ...], --filter-metrics  [ ...]                  Specify IP block/metric Ids from --list-metrics.
  -k  [ ...], --filter-kernels  [ ...]                  Specify kernel id from --list-kernels.
  --filter-dispatch-ids  [ ...]                         Specify dispatch IDs.
  --filter-gpu-ids  [ ...]                              Specify GPU IDs.
  -n , --normal-unit                                    Specify the normalization unit: (DEFAULT: per_wave)
                                                           per_wave
                                                           per_cycle
                                                           per_second
  --config-dir                                          Specify the directory of customized configs.
  -t , --time-unit                                      Specify display time unit in kernel top stats: (DEFAULT: ns)
                                                           s
                                                           ms
                                                           us
                                                           ns
  --decimal                                             Specify the decimal to display. (DEFAULT: 2)
  --cols  [ ...]                                        Specify column indices to display.
  -g                                                    Debug single metric.
  --dependency                                          List the installation dependency.
  --gui [GUI]                                           Activate a GUI to interate with Omniperf metrics.
                                                        Optionally, specify port to launch application (DEFAULT: 8050)
```

Omniperf Analyze with standalone GUI

- Download the data on your computer (workloads/vcopy_all/), install Omniperf without ROCm, and execute:

```
omniperf analyze -p workloads/vcopy_all/mi200/ --gui
```

Open web page http://IP:8050/



Instr Buff
Wave 0 Instr buff
Wave N-1 Instr buff
Wave Occupancy
14 per-GCD
Wave Life
62014 cycles

Instr Dispatch
SALU: 291
SMEM: 6
VALU: 475
MFMA: 0
VMEM: 50
LDS: 31
GWS: 0
Br: 22

Exec
Active CUs
58/110
VGPRs: 44
SGPRs: 48
LDS Alloc: 0
Scratch Alloc: 16496
Wavefronts: 16384
Workgroups: 4096

Req: 31
Rd: 960
Wr: 1840
Atomic: 0
Rd: 6
Fetch: 198

LDS
Util: 1 %
Lat: 47 cycles

Vector L1 Cache
Hit: 69 %
Lat: 1674 cycles
Coales: 29 %
Stall: 0 %

Scalar L1D Cache
Hit: 97 %
Lat: cycles

Instr L1 Cache
Hit: 99 %
Lat: cycles

Rd: 53
Wr: 164
Atomic: 0
Rd: 0
Wr: 0
Atomic: 0
Req: 0

L2 Cache
Rd: 34
Wr: 164
Atomic: 0
Hit: 74 %
Latency
Rd: 474 cycles
Wr: 243 cycles

Rd: 25
Wr: 138
Atomic: 0

xGMI / PCIe

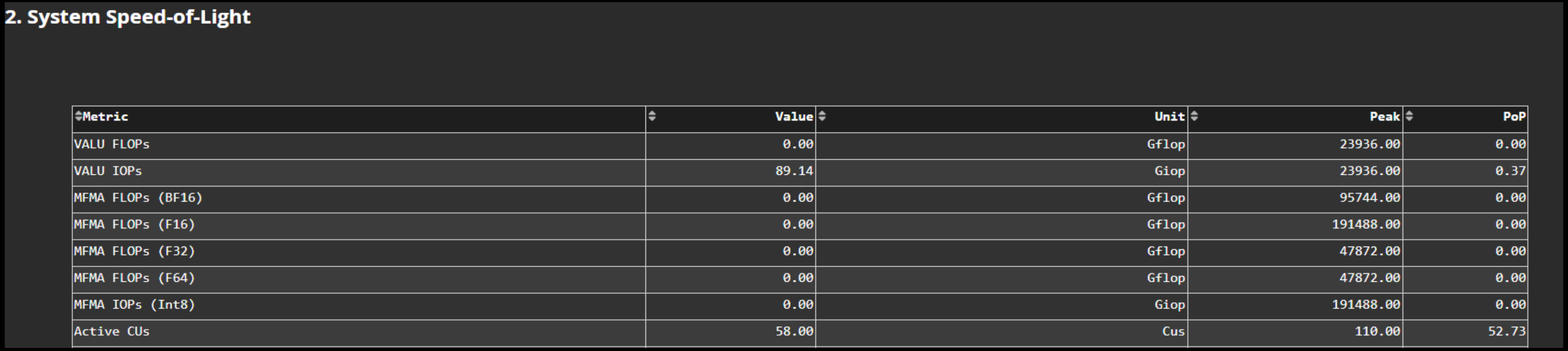
Fabric
Rd: 593 cycles
Wr: 172 cycles
Atomic: cycles

GMI

Rd: 25
Wr: 138

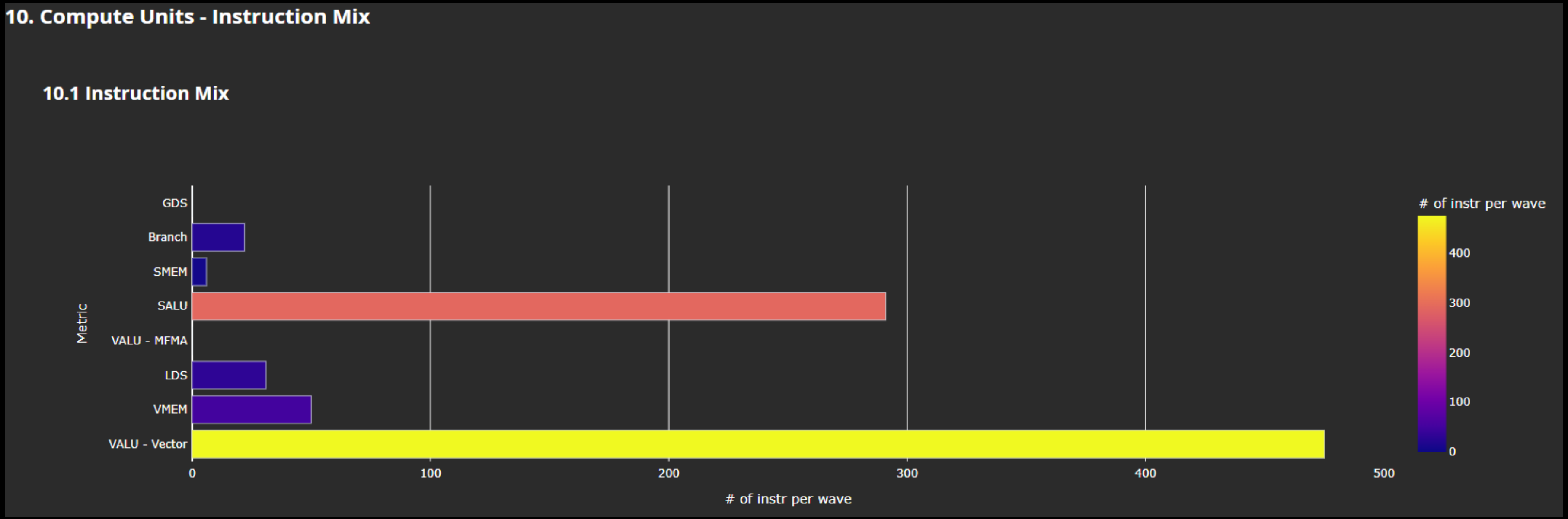
HBM

Omniperf Analyze with standalone GUI (II)



2. System Speed-of-Light

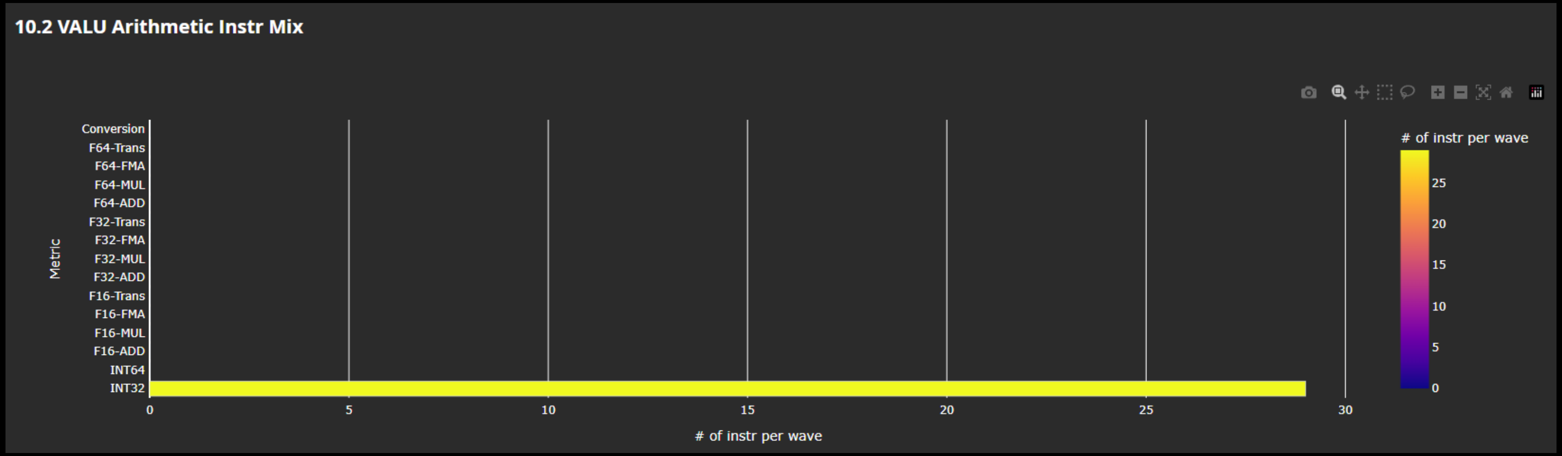
Metric	Value	Unit	Peak	PoP
VALU FLOPs	0.00	Gflop	23936.00	0.00
VALU IOPs	89.14	Giop	23936.00	0.37
MFMA FLOPs (BF16)	0.00	Gflop	95744.00	0.00
MFMA FLOPs (F16)	0.00	Gflop	191488.00	0.00
MFMA FLOPs (F32)	0.00	Gflop	47872.00	0.00
MFMA FLOPs (F64)	0.00	Gflop	47872.00	0.00
MFMA IOPs (Int8)	0.00	Giop	191488.00	0.00
Active CUs	58.00	Cus	110.00	52.73

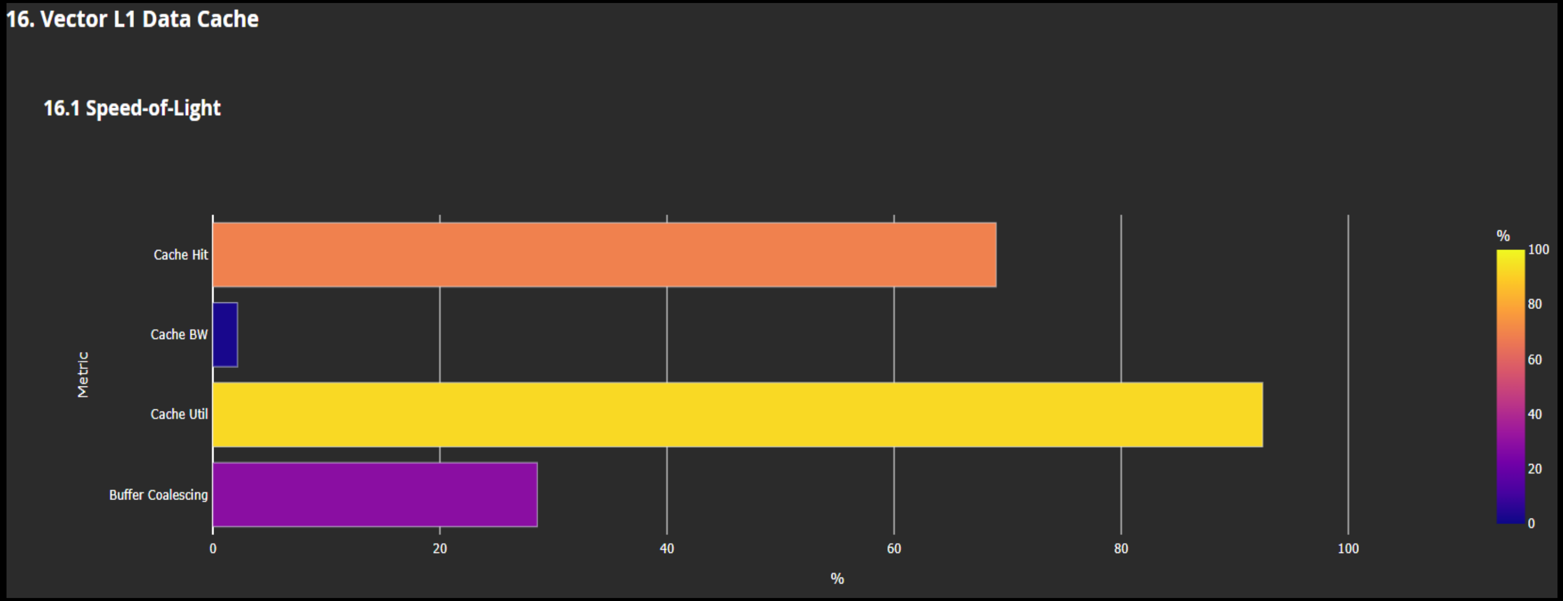


10. Compute Units - Instruction Mix

10.1 Instruction Mix

Omniperf Analyze with standalone GUI (III)



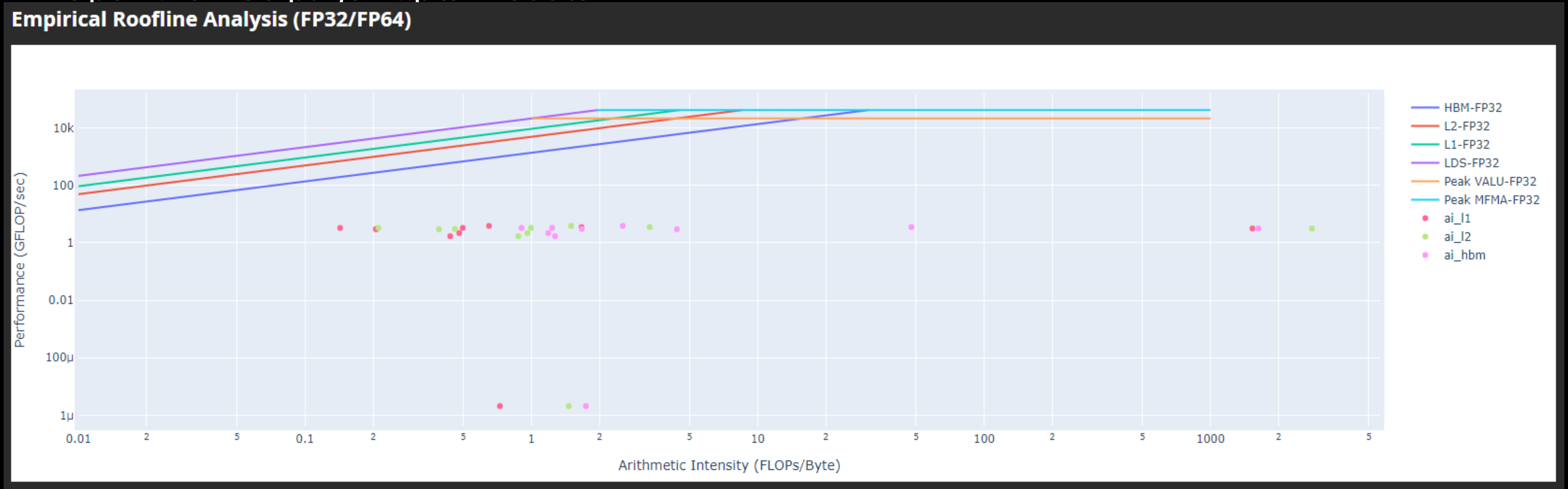


Roofline Analysis

- Profile with roofline:

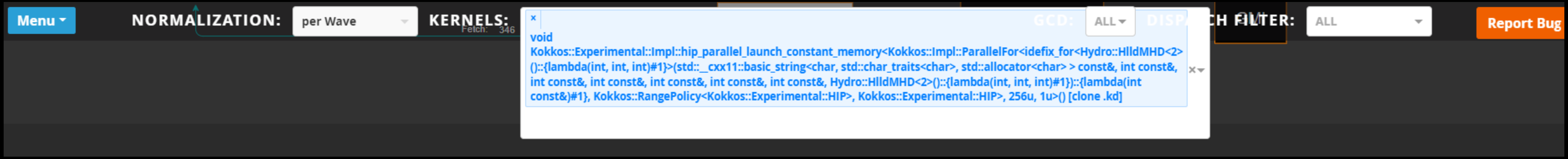
```
srun -n 1 --gpus 1 omniperf profile -n roofline_case_app --roof-only --
./app
```

- Prepare GUI:
  Copy the workload to your computer
  Execute: omniperf analyze -p workloads/roofline_case_app/mi200/ --gui
  Open the web page http://IP:8050/

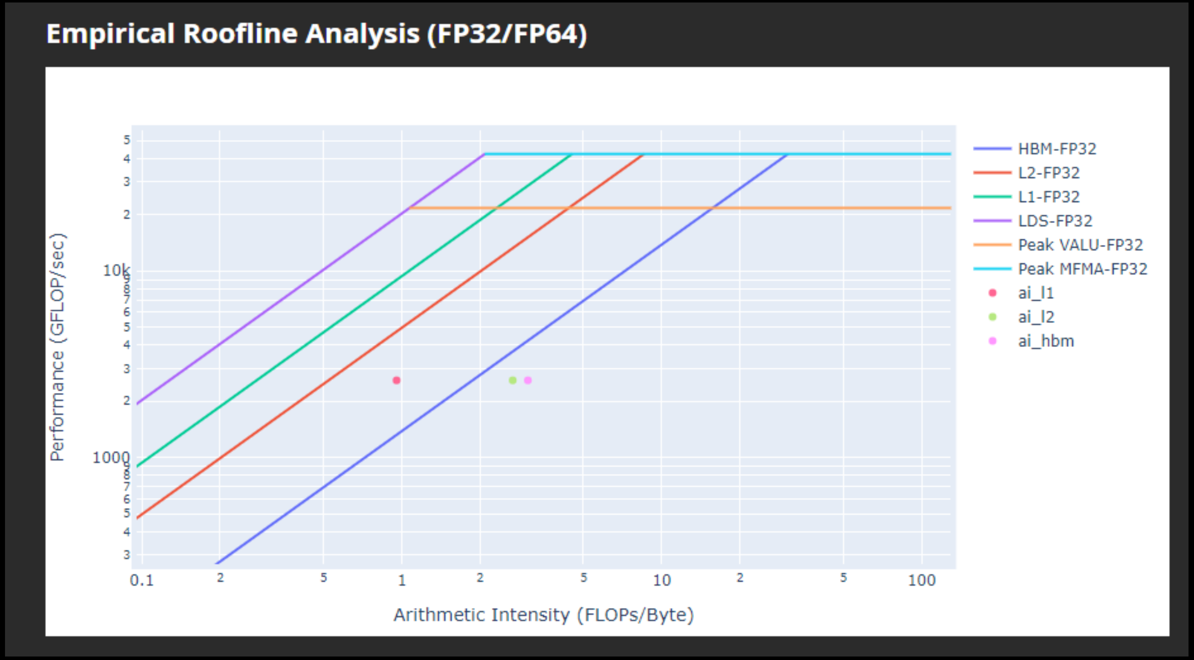


Empirical Roofline Analysis (FP32/FP64)

Roofline Analysis – Kokkos code



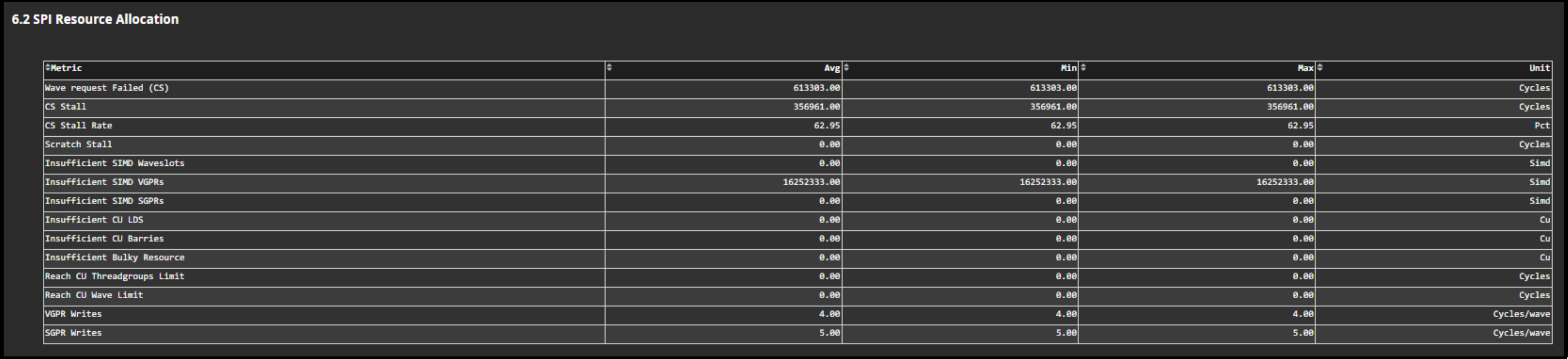
```
void
Kokkos::Experimental::Impl::hip_parallel_launch_constant_memory<Kokkos::Impl::ParallelFor<idefix_for<Hydro::HlldMHD<2>
()::{lambda(int, int, int)#1}>(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, int const&,
int const&, int const&, int const&, int const&, int const&, Hydro::HlldMHD<2>()::{lambda(int, int, int)#1}):::{lambda(int
const&)#1}, Kokkos::RangePolicy<Kokkos::Experimental::HIP>, Kokkos::Experimental::HIP>, 256u, 1u>() [clone .kd]
```


Empirical Roofline Analysis (FP32/FP64)

- Roofline: the first-step characterization of workload performance
  - Workload characterization
    - Compute bound
    - Memory bound
    - Performance margin
    - L1/L2 cache accesses
- Thorough SoC perf analysis for each subsystem to identify bottlenecks
  - HBM
  - L1/L2
  - LDS
  - Shader compute
  - Wavefront dispatch
- Omniperf tooling support
  - Roofline plot (float, integer)
  - Baseline roofline comparison
  - Kernel statistics

SPI Resource Allocation

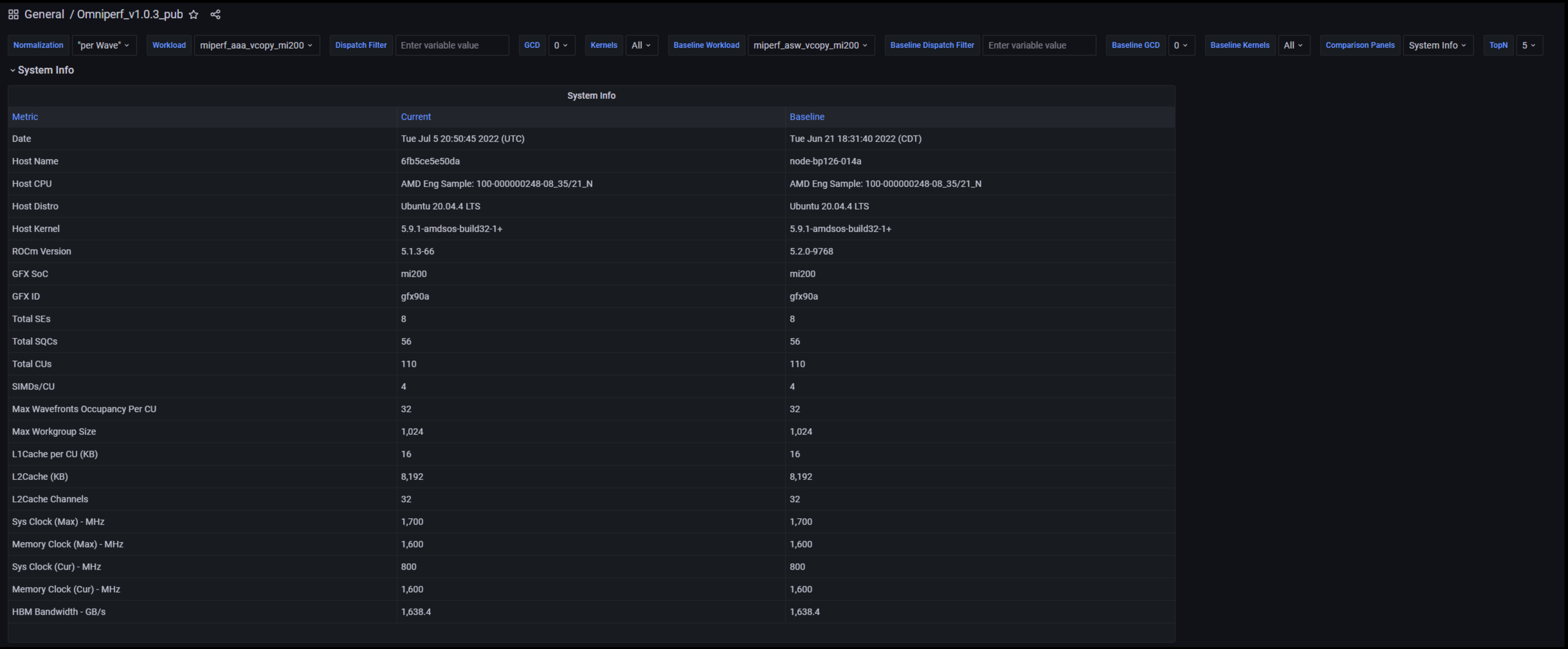
- Dispatch Bound
  - Wavefront dispatching failure due to resources limitation
    - Wavefront slots
    - VGPR
    - SGPR
    - LDS allocation
    - Barriers
    - Etc.
  - Omniperf tooling support
    - Shader Processor Input (SPI) metrics



6.2 SPI Resource Allocation

Metric	Avg	Min	Max	Unit
Wave request Failed (CS)	613303.00	613303.00	613303.00	Cycles
CS Stall	356961.00	356961.00	356961.00	Cycles
CS Stall Rate	62.95	62.95	62.95	Pct
Scratch Stall	0.00	0.00	0.00	Cycles
Insufficient SIMD Waveslots	0.00	0.00	0.00	Simd
Insufficient SIMD VGPRs	16252333.00	16252333.00	16252333.00	Simd
Insufficient SIMD SGPRs	0.00	0.00	0.00	Simd
Insufficient CU LDS	0.00	0.00	0.00	Cu
Insufficient CU Barries	0.00	0.00	0.00	Cu
Insufficient Bulky Resource	0.00	0.00	0.00	Cu
Reach CU Threadgroups Limit	0.00	0.00	0.00	Cycles
Reach CU Wave Limit	0.00	0.00	0.00	Cycles
VGPR Writes	4.00	4.00	4.00	Cycles/wave
SGPR Writes	5.00	5.00	5.00	Cycles/wave

Grafana – System Info



General / Omniperf_v1.0.3_pub

Normalization "per Wave" · Workload miperf_aaa_vcopy_mi200 · Dispatch Filter Enter variable value · GCD 0 · Kernels All · Baseline Workload miperf_asw_vcopy_mi200 · Baseline Dispatch Filter Enter variable value · Baseline GCD 0 · Baseline Kernels All · Comparison Panels System Info · TopN 5

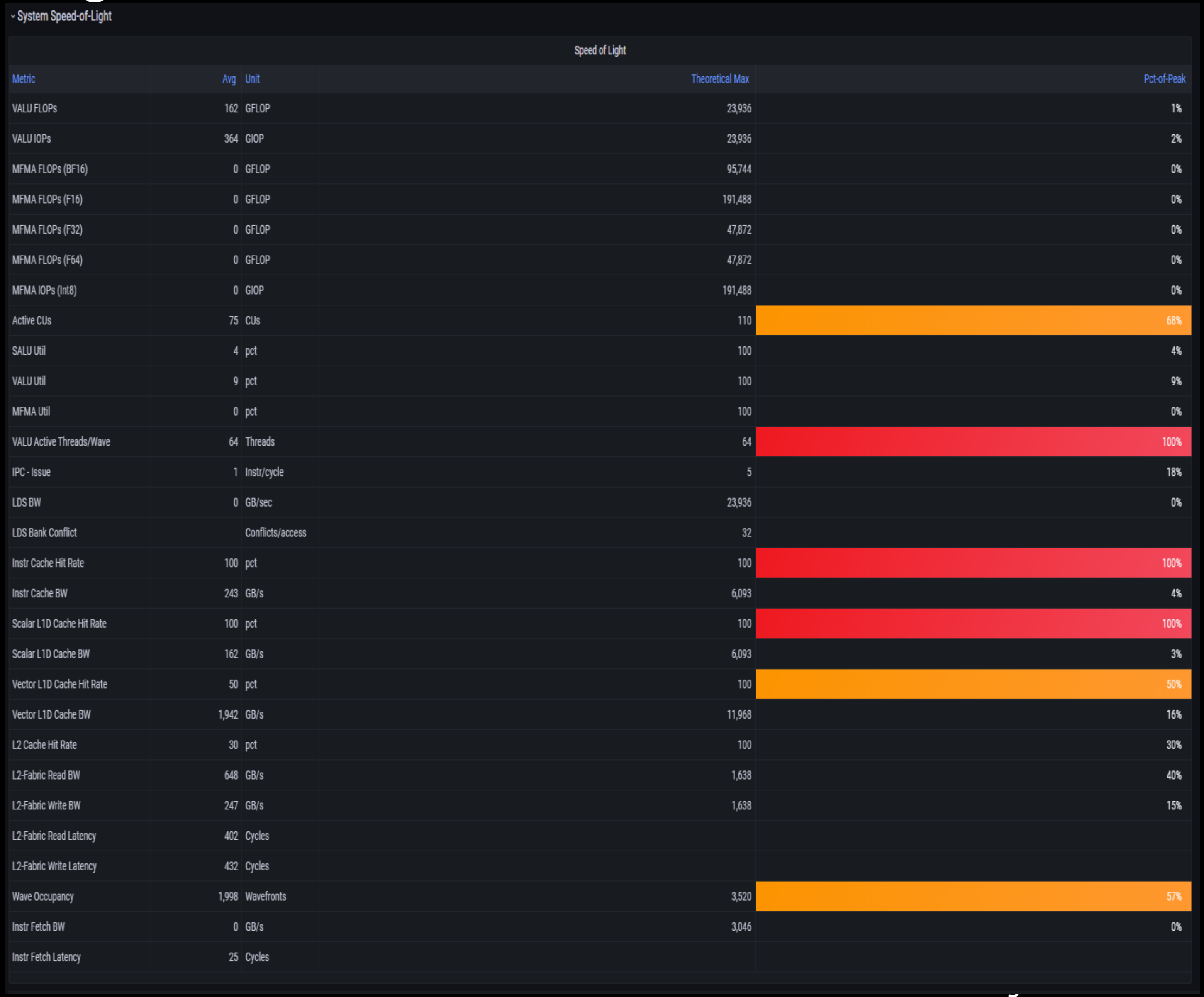
System Info

System Info

Metric	Current	Baseline
Date	Tue Jul 5 20:50:45 2022 (UTC)	Tue Jun 21 18:31:40 2022 (CDT)
Host Name	6fb5ce5e50da	node-bp126-014a
Host CPU	AMD Eng Sample: 100-000000248-08_35/21_N	AMD Eng Sample: 100-000000248-08_35/21_N
Host Distro	Ubuntu 20.04.4 LTS	Ubuntu 20.04.4 LTS
Host Kernel	5.9.1-amdsos-build32-1+	5.9.1-amdsos-build32-1+
ROCm Version	5.1.3-66	5.2.0-9768
GFX SoC	mi200	mi200
GFX ID	gfx90a	gfx90a
Total SEs	8	8
Total SQCs	56	56
Total CUs	110	110
SIMDs/CU	4	4
Max Wavefronts Occupancy Per CU	32	32
Max Workgroup Size	1,024	1,024
L1Cache per CU (KB)	16	16
L2Cache (KB)	8,192	8,192
L2Cache Channels	32	32
Sys Clock (Max) - MHz	1,700	1,700
Memory Clock (Max) - MHz	1,600	1,600
Sys Clock (Cur) - MHz	800	800
Memory Clock (Cur) - MHz	1,600	1,600
HBM Bandwidth - GB/s	1,638.4	1,638.4

Grafana – System Speed-of-Light

```
$omniperf database --import -H pavii1 -u amd -t asw -w
workloads/vcopy_demo/mi200/
ROC Profiler:  /usr/bin/rocprof

--------
Import Profiling Results
--------

Pulling data from  /root/test/workloads/vcopy_demo/mi200
The directory exists
Found sysinfo file
KernelName shortening enabled
Kernel name verbose level: 2
Password:
Password recieved
-- Conversion & Upload in Progress –
… …
9 collections added.
Workload name uploaded
-- Complete! --
```

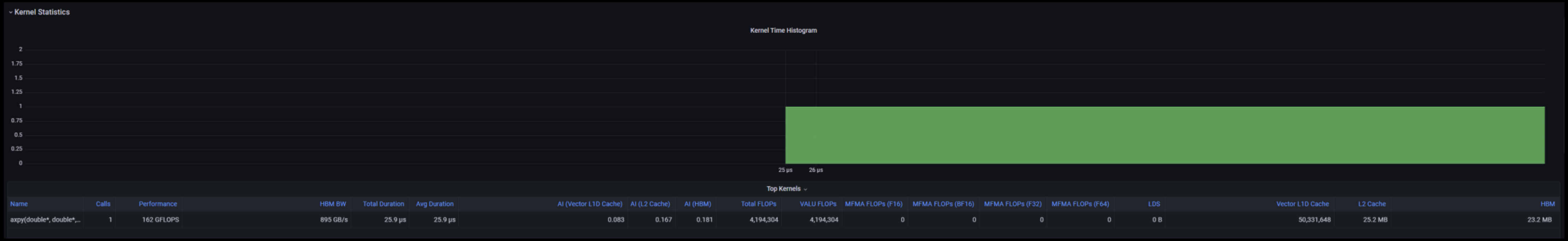


⌄ System Speed-of-Light

Speed of Light

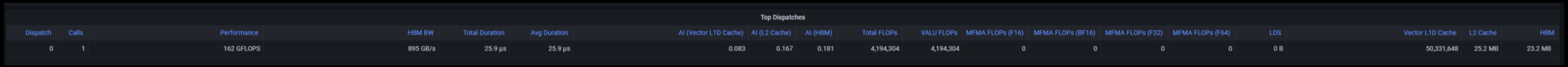
Metric	Avg	Unit	Theoretical Max	Pct-of-Peak
VALU FLOPs	162	GFLOP	23,936	1%
VALU IOPs	364	GIOP	23,936	2%
MFMA FLOPs (BF16)	0	GFLOP	95,744	0%
MFMA FLOPs (F16)	0	GFLOP	191,488	0%
MFMA FLOPs (F32)	0	GFLOP	47,872	0%
MFMA FLOPs (F64)	0	GFLOP	47,872	0%
MFMA IOPs (Int8)	0	GIOP	191,488	0%
Active CUs	75	CUs	110	68%
SALU Util	4	pct	100	4%
VALU Util	9	pct	100	9%
MFMA Util	0	pct	100	0%
VALU Active Threads/Wave	64	Threads	64	100%
IPC - Issue	1	Instr/cycle	5	18%
LDS BW	0	GB/sec	23,936	0%
LDS Bank Conflict		Conflicts/access	32	
Instr Cache Hit Rate	100	pct	100	100%
Instr Cache BW	243	GB/s	6,093	4%
Scalar L1D Cache Hit Rate	100	pct	100	100%
Scalar L1D Cache BW	162	GB/s	6,093	3%
Vector L1D Cache Hit Rate	50	pct	100	50%
Vector L1D Cache BW	1,942	GB/s	11,968	16%
L2 Cache Hit Rate	30	pct	100	30%
L2-Fabric Read BW	648	GB/s	1,638	40%
L2-Fabric Write BW	247	GB/s	1,638	15%
L2-Fabric Read Latency	402	Cycles		
L2-Fabric Write Latency	432	Cycles		
Wave Occupancy	1,998	Wavefronts	3,520	57%
Instr Fetch BW	0	GB/s	3,046	0%
Instr Fetch Latency	25	Cycles		

Grafana- Kernel Statistics



Top Kernels ⌄

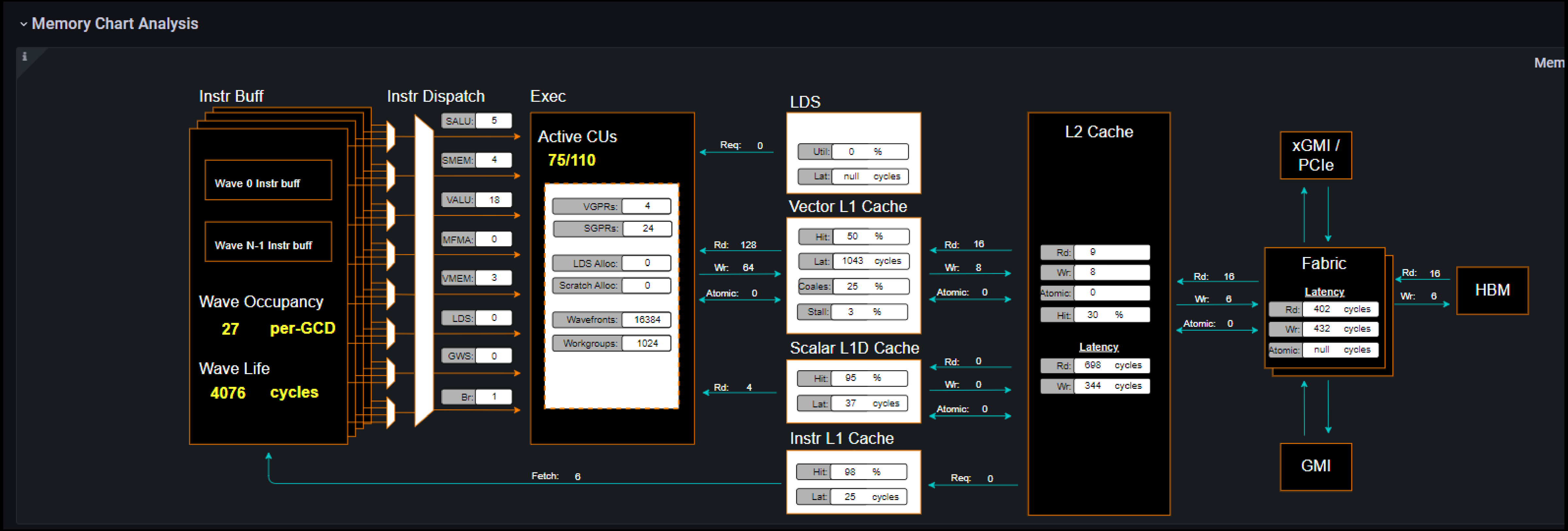
Name	Calls	Performance	HBM BW	Total Duration	Avg Duration	AI (Vector L1D Cache)	AI (L2 Cache)	AI (HBM)	Total FLOPs	VALU FLOPs	MFMA FLOPs (F16)	MFMA FLOPs (BF16)	MFMA FLOPs (F32)	MFMA FLOPs (F64)	LDS	Vector L1D Cache	L2 Cache	HBM
axpy(double*, double*,...	1	162 GFLOPS	895 GB/s	25.9 µs	25.9 µs	0.083	0.167	0.181	4,194,304	4,194,304	0	0	0	0	0 B	50,331,648	25.2 MB	23.2 MB



Top Dispatches

Dispatch	Calls	Performance	HBM BW	Total Duration	Avg Duration	AI (Vector L1D Cache)	AI (L2 Cache)	AI (HBM)	Total FLOPs	VALU FLOPs	MFMA FLOPs (F16)	MFMA FLOPs (BF16)	MFMA FLOPs (F32)	MFMA FLOPs (F64)	LDS	Vector L1D Cache	L2 Cache	HBM
0	1	162 GFLOPS	895 GB/s	25.9 µs	25.9 µs	0.083	0.167	0.181	4,194,304	4,194,304	0	0	0	0	0 B	50,331,648	25.2 MB	23.2 MB

Grafana – Memory Chart Analysis



˅ Memory Chart Analysis

Mem

Instr Buff

Wave 0 Instr buff

Wave N-1 Instr buff

Wave Occupancy
27 per-GCD

Wave Life
4076 cycles

Instr Dispatch

SALU: 5
SMEM: 4
VALU: 18
MFMA: 0
VMEM: 3
LDS: 0
GWS: 0
Br: 1

Exec

Active CUs
75/110

VGPRs: 4
SGPRs: 24
LDS Alloc: 0
Scratch Alloc: 0
Wavefronts: 16384
Workgroups: 1024

Req: 0
Rd: 128
Wr: 64
Atomic: 0
Rd: 4
Fetch: 6

LDS
Util: 0 %
Lat: null cycles

Vector L1 Cache
Hit: 50 %
Lat: 1043 cycles
Coales: 25 %
Stall: 3 %

Scalar L1D Cache
Hit: 95 %
Lat: 37 cycles

Instr L1 Cache
Hit: 98 %
Lat: 25 cycles

Rd: 16
Wr: 8
Atomic: 0
Rd: 0
Wr: 0
Atomic: 0
Req: 0

L2 Cache
Rd: 9
Wr: 8
Atomic: 0
Hit: 30 %

Latency
Rd: 698 cycles
Wr: 344 cycles

Rd: 16
Wr: 6
Atomic: 0

xGMI / PCIe

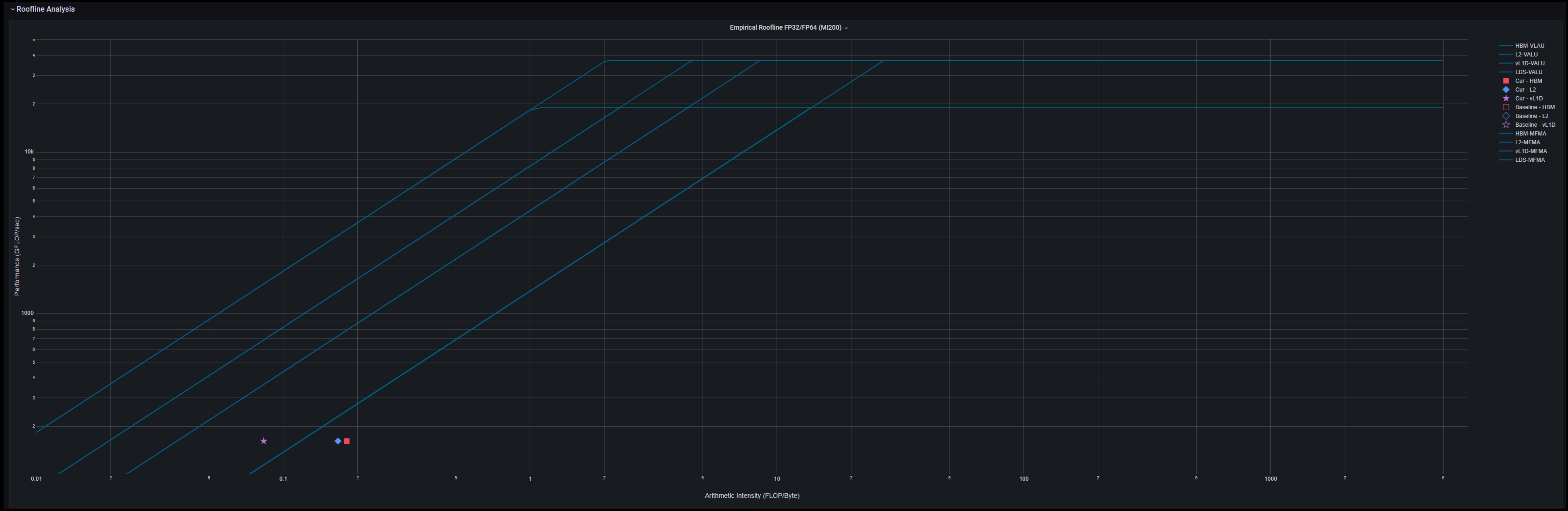
Fabric
Latency
Rd: 402 cycles
Wr: 432 cycles
Atomic: null cycles

GMI

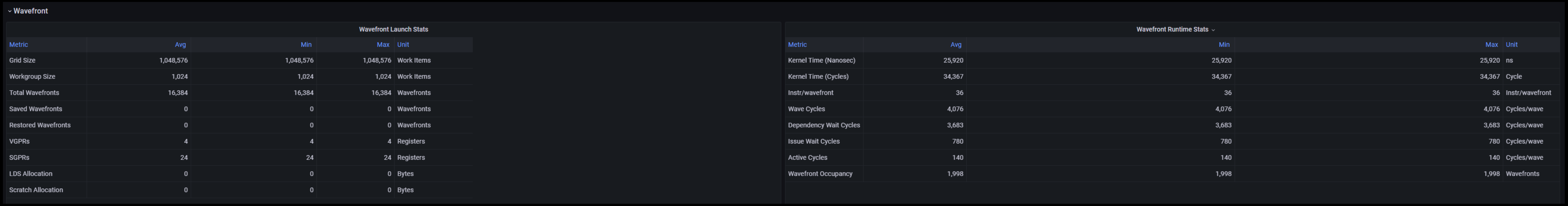
Rd: 16
Wr: 6

HBM

Grafana - Roofline



Grafana – Wavefront & Compute Unit



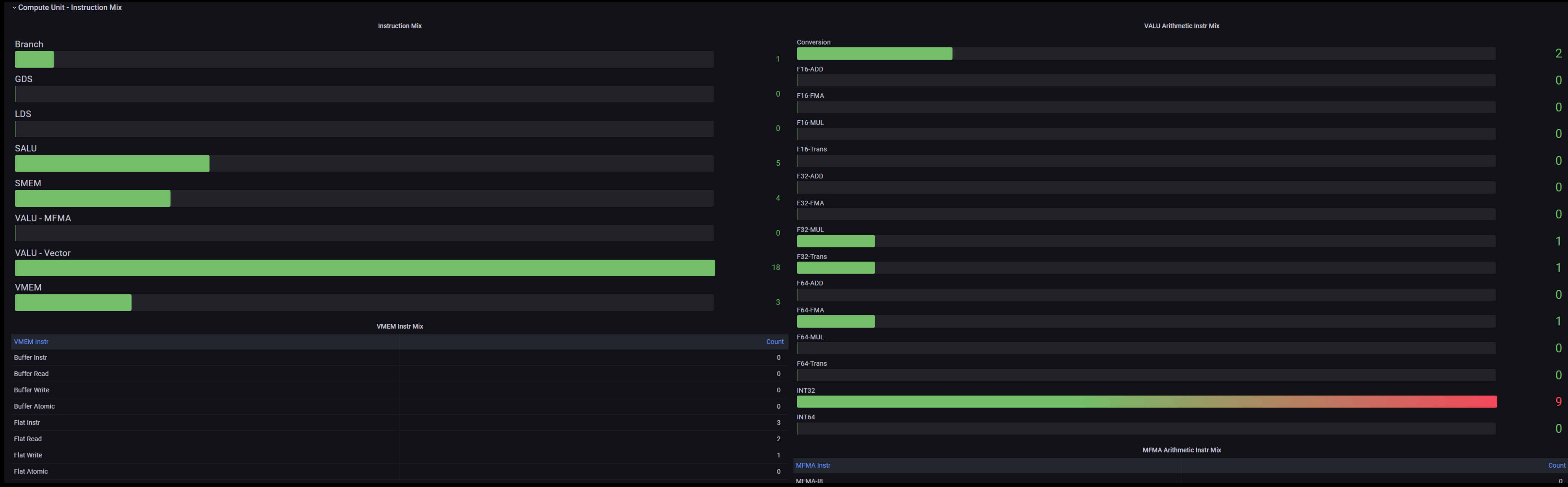
Wavefront

Wavefront Launch Stats

Metric	Avg	Min	Max	Unit
Grid Size	1,048,576	1,048,576	1,048,576	Work Items
Workgroup Size	1,024	1,024	1,024	Work Items
Total Wavefronts	16,384	16,384	16,384	Wavefronts
Saved Wavefronts	0	0	0	Wavefronts
Restored Wavefronts	0	0	0	Wavefronts
VGPRs	4	4	4	Registers
SGPRs	24	24	24	Registers
LDS Allocation	0	0	0	Bytes
Scratch Allocation	0	0	0	Bytes

Wavefront Runtime Stats

Metric	Avg	Min	Max	Unit
Kernel Time (Nanosec)	25,920	25,920	25,920	ns
Kernel Time (Cycles)	34,367	34,367	34,367	Cycle
Instr/wavefront	36	36	36	Instr/wavefront
Wave Cycles	4,076	4,076	4,076	Cycles/wave
Dependency Wait Cycles	3,683	3,683	3,683	Cycles/wave
Issue Wait Cycles	780	780	780	Cycles/wave
Active Cycles	140	140	140	Cycles/wave
Wavefront Occupancy	1,998	1,998	1,998	Wavefronts



Compute Unit - Instruction Mix

Instruction Mix

Branch	1
GDS	0
LDS	0
SALU	5
SMEM	4
VALU - MFMA	0
VALU - Vector	18
VMEM	3

VMEM Instr Mix

VMEM Instr	Count
Buffer Instr	0
Buffer Read	0
Buffer Write	0
Buffer Atomic	0
Flat Instr	3
Flat Read	2
Flat Write	1
Flat Atomic	0

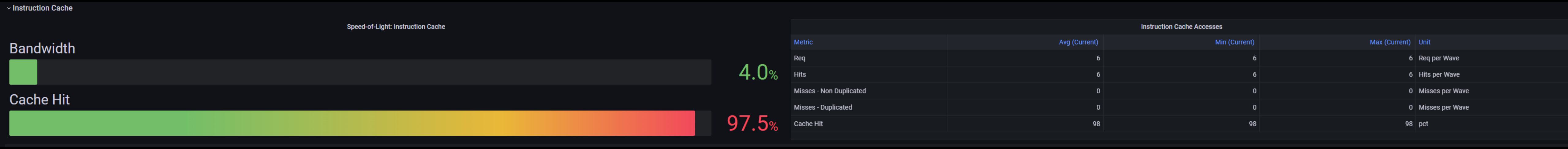
VALU Arithmetic Instr Mix

Conversion	2
F16-ADD	0
F16-FMA	0
F16-MUL	0
F16-Trans	0
F32-ADD	0
F32-FMA	0
F32-MUL	1
F32-Trans	1
F64-ADD	0
F64-FMA	1
F64-MUL	0
F64-Trans	0
INT32	9
INT64	0

MFMA Arithmetic Instr Mix

MFMA Instr	Count
MFMA-I8	0

Grafana – Instruction Cache & Scalar L1 Data Cache

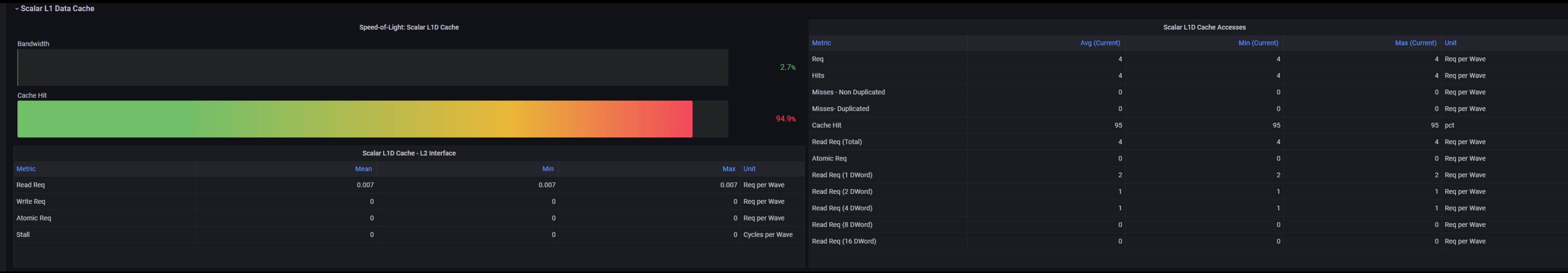


˅ Instruction Cache

Speed-of-Light: Instruction Cache

Instruction Cache Accesses

Metric	Avg (Current)	Min (Current)	Max (Current)	Unit
Req	6	6	6	Req per Wave
Hits	6	6	6	Hits per Wave
Misses - Non Duplicated	0	0	0	Misses per Wave
Misses - Duplicated	0	0	0	Misses per Wave
Cache Hit	98	98	98	pct



˅ Scalar L1 Data Cache

Speed-of-Light: Scalar L1D Cache

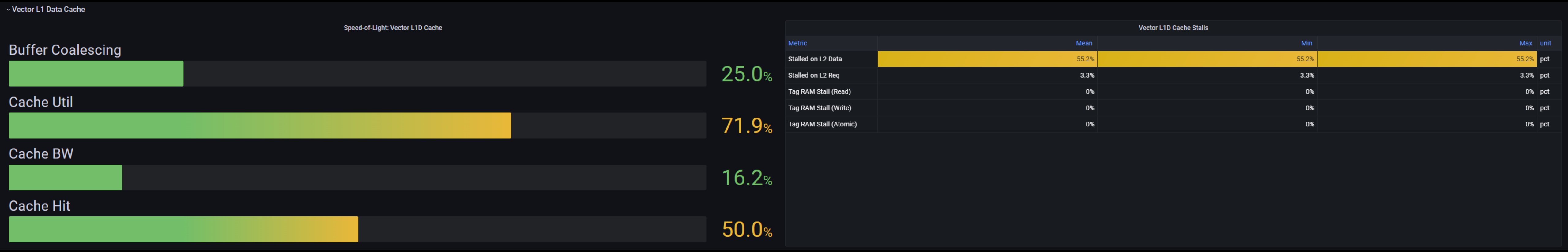
Scalar L1D Cache - L2 Interface

Metric	Mean	Min	Max	Unit
Read Req	0.007	0.007	0.007	Req per Wave
Write Req	0	0	0	Req per Wave
Atomic Req	0	0	0	Req per Wave
Stall	0	0	0	Cycles per Wave

Scalar L1D Cache Accesses

Metric	Avg (Current)	Min (Current)	Max (Current)	Unit
Req	4	4	4	Req per Wave
Hits	4	4	4	Req per Wave
Misses - Non Duplicated	0	0	0	Req per Wave
Misses- Duplicated	0	0	0	Req per Wave
Cache Hit	95	95	95	pct
Read Req (Total)	4	4	4	Req per Wave
Atomic Req	0	0	0	Req per Wave
Read Req (1 DWord)	2	2	2	Req per Wave
Read Req (2 DWord)	1	1	1	Req per Wave
Read Req (4 DWord)	1	1	1	Req per Wave
Read Req (8 DWord)	0	0	0	Req per Wave
Read Req (16 DWord)	0	0	0	Req per Wave

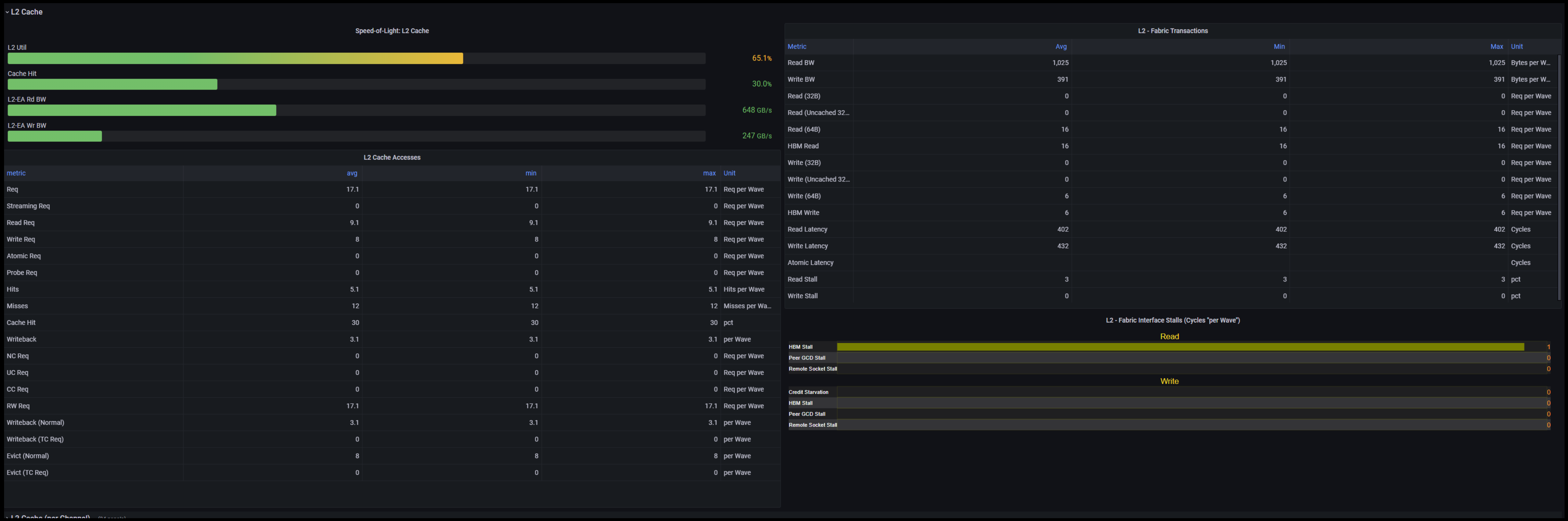
Grafana – Vector L1 Data Cache



Vector L1D Cache Stalls

Metric	Mean	Min	Max	unit
Stalled on L2 Data	55.2%	55.2%	55.2%	pct
Stalled on L2 Req	3.3%	3.3%	3.3%	pct
Tag RAM Stall (Read)	0%	0%	0%	pct
Tag RAM Stall (Write)	0%	0%	0%	pct
Tag RAM Stall (Atomic)	0%	0%	0%	pct

Grafana – L2 Cache



˅ L2 Cache

Speed-of-Light: L2 Cache

L2 Util	65.1%
Cache Hit	30.0%
L2-EA Rd BW	648 GB/s
L2-EA Wr BW	247 GB/s

L2 Cache Accesses

metric	avg	min	max	Unit
Req	17.1	17.1	17.1	Req per Wave
Streaming Req	0	0	0	Req per Wave
Read Req	9.1	9.1	9.1	Req per Wave
Write Req	8	8	8	Req per Wave
Atomic Req	0	0	0	Req per Wave
Probe Req	0	0	0	Req per Wave
Hits	5.1	5.1	5.1	Hits per Wave
Misses	12	12	12	Misses per Wa...
Cache Hit	30	30	30	pct
Writeback	3.1	3.1	3.1	per Wave
NC Req	0	0	0	Req per Wave
UC Req	0	0	0	Req per Wave
CC Req	0	0	0	Req per Wave
RW Req	17.1	17.1	17.1	Req per Wave
Writeback (Normal)	3.1	3.1	3.1	per Wave
Writeback (TC Req)	0	0	0	per Wave
Evict (Normal)	8	8	8	per Wave
Evict (TC Req)	0	0	0	per Wave

L2 - Fabric Transactions

Metric	Avg	Min	Max	Unit
Read BW	1,025	1,025	1,025	Bytes per W...
Write BW	391	391	391	Bytes per W...
Read (32B)	0	0	0	Req per Wave
Read (Uncached 32...	0	0	0	Req per Wave
Read (64B)	16	16	16	Req per Wave
HBM Read	16	16	16	Req per Wave
Write (32B)	0	0	0	Req per Wave
Write (Uncached 32...	0	0	0	Req per Wave
Write (64B)	6	6	6	Req per Wave
HBM Write	6	6	6	Req per Wave
Read Latency	402	402	402	Cycles
Write Latency	432	432	432	Cycles
Atomic Latency				Cycles
Read Stall	3	3	3	pct
Write Stall	0	0	0	pct

L2 - Fabric Interface Stalls (Cycles "per Wave")

Read

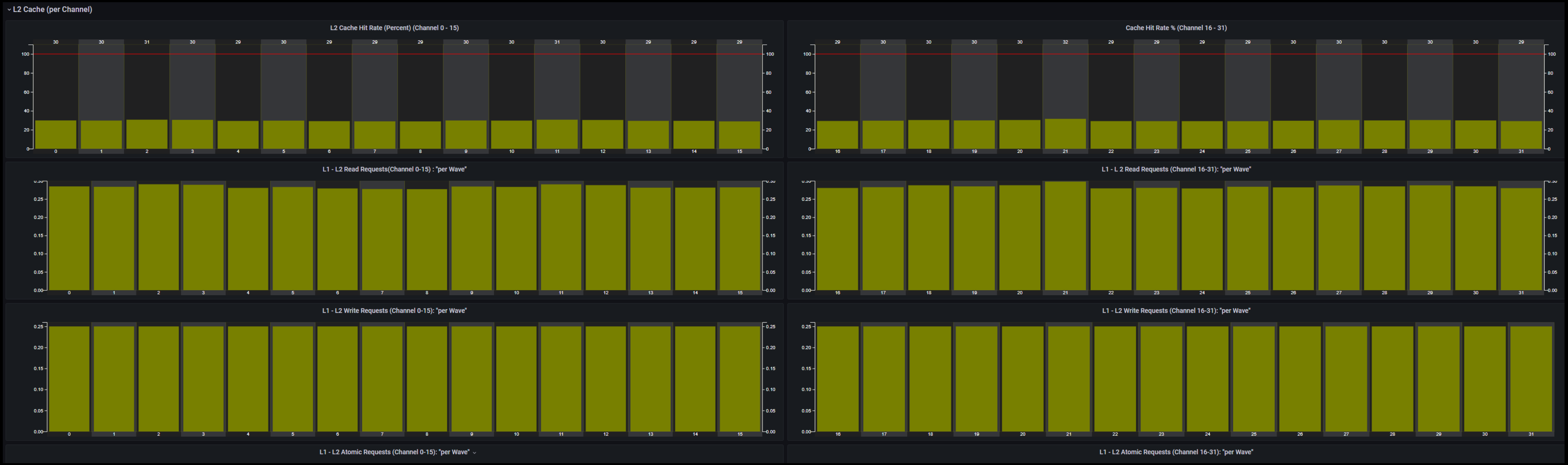
HBM Stall	1
Peer GCD Stall	0
Remote Socket Stall	0

Write

Credit Starvation	0
HBM Stall	0
Peer GCD Stall	0
Remote Socket Stall	0

› L2 Cache (per Channel) (24 panels)

Grafana – L2 Cache (per Channel)


L2 Cache (per Channel)
L2 Cache Hit Rate (Percent) (Channel 0 - 15)
30
30
31
30
29
30
29
29
29
30
30
31
30
29
29
29
0
20
40
60
80
100
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Cache Hit Rate % (Channel 16 - 31)
29
30
30
30
30
32
29
29
29
29
30
30
30
30
30
29
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
L1 - L2 Read Requests(Channel 0-15) : "per Wave"
0.00
0.05
0.10
0.15
0.20
0.25
L1 - L 2 Read Requests (Channel 16-31): "per Wave"
L1 - L2 Write Requests (Channel 0-15): "per Wave"
L1 - L2 Write Requests (Channel 16-31): "per Wave"
L1 - L2 Atomic Requests (Channel 0-15): "per Wave"
L1 - L2 Atomic Requests (Channel 16-31): "per Wave"

AMD
together we advance_