COMPREHENSIVE GENERAL LUMI COURSE
WARSAW, POLAND

INTRODUCTION TO OMNIPERF

GINA SITARAMAN, SUYASH TANDON, GEORGE MARKOMANOLIS, JONATHAN MADSEN, AUSTIN ELLIS, BOB ROBEY, XIAOMIN LU, NOAH WOLFE, SAMUEL ANTAO

JAKUB KURZAK - PRESENTER

ADVANCED MICRO DEVICES, INC.

AMD
together we advance_

slides on LUMI in /project/project_465000644/Slides/AMD/

hands-on exercises: https://hackmd.io/@sfantao/H1QU6xRR3

hands-on source code: /project/project_465000644/Exercises/AMD/HPCTrainingExamples/

OMNIPERF

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

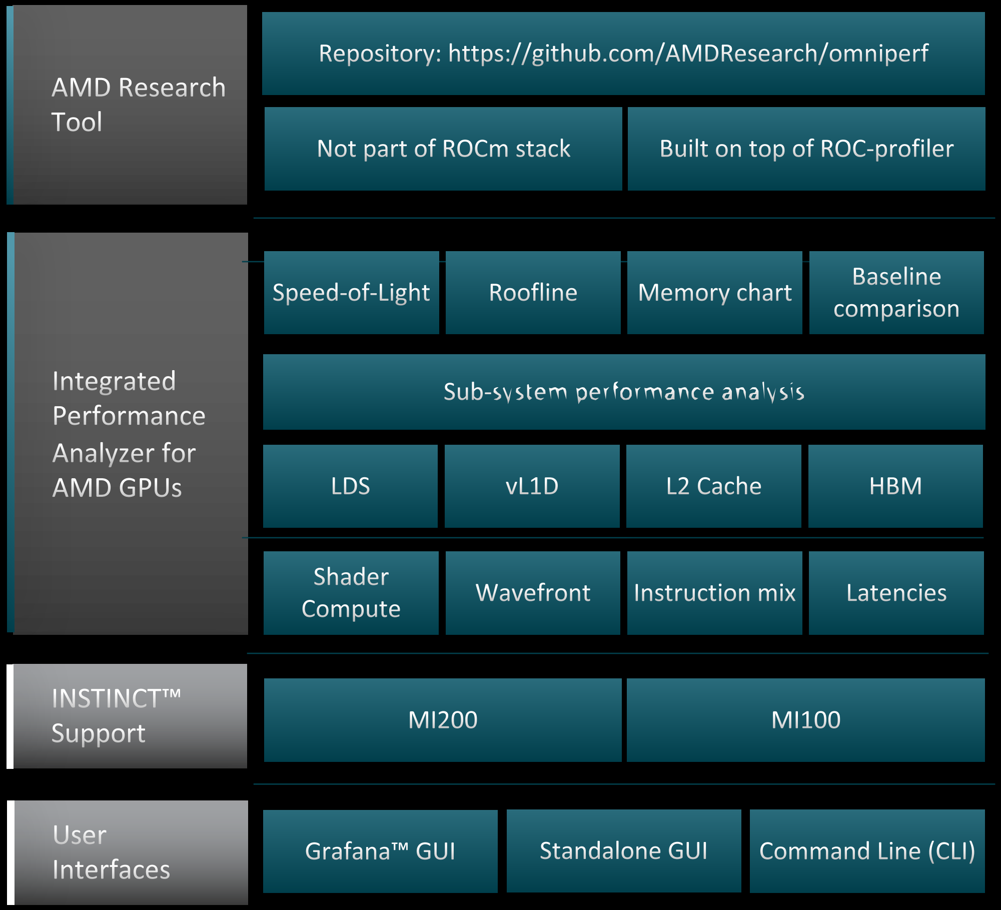
Omnitrace

- 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

Omniperf

- Performance Analysis
  - Automated collection of hardware counters
  - Analysis
  - Visualization
- Supports
  - Speed of Light
  - Memory chart
  - Rooflines
  - Kernel comparison
- Visualization
  - With Grafana™ or standalone GUI

ABOUT



AMD Research Tool

Repository: https://github.com/AMDResearch/omniperf

Not part of ROCm stack

Built on top of ROC-profiler

Integrated Performance Analyzer for AMD GPUs

Speed-of-Light

Roofline

Memory chart

Baseline comparison

Sub-system performance analysis

LDS

vL1D

L2 Cache

HBM

Shader Compute

Wavefront

Instruction mix

Latencies

INSTINCT™ Support

MI200

MI100

User Interfaces

Grafana™ GUI

Standalone GUI

Command Line (CLI)

- performance profiling tool for AMD Instinct(TM) GPUs
    - supporting MI100 and MI200 GPUs
    - MI300 support under development
    - Radeon™ RDNA™ on the radar
- not part of the ROCm software stack yet
- built on top of rocprof
- research project
- open source
- mostly Python™
- open to community contributions
- MIT license (permissive / copyleft)

online documentation: https://amdresearch.github.io/omniperf/

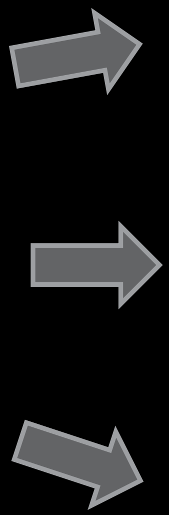
Github repository: https://github.com/AMDResearch/omniperf

- reporting problems: https://github.com/AMDResearch/omniperf/issues

COMPONENTS

omniperf profiling

- uses rocprof
- relies on application replay
- collects raw performance counters
- stores in CSV
- in addition, runs a set of MI200 microbenchmarks to acquire roofline data



omniperf Grafana™ Analyzer

- raw counters imported into MongoDB®
- visualization using the Grafana™ GUI
    - compiles performance metrics
    - provides visualization

omniperf standalone GUI Analyzer

- visualization without backend database

formatted text output (CLI)

- no visualization
- formatted console output

COMPONENTS

MongoDB®

- open-source NoSQL database system
- stores data in JSON-like documents
- scalable for handling large datasets
- well-suited for web applications
- https://www.mongodb.com/

Grafana™

- open-source and visualization platform
- widely used in DevOps and IT operations
- visualizes time-series data and metrics
- supports various data sources and plugins
- https://grafana.com/

INSTALLATION

OMNIPERF

```
# download a release from https://github.com/AMDResearch/omniperf/releases
wget https://github.com/AMDResearch/omniperf/releases/download/v1.0.10/omniperf-v1.0.10.tar.gz
tar xfz omniperf-v1.0.10.tar.gz
cd omniperf-1.0.10
```

```
# or clone the Github repository and checkout the dev branch
git clone https://github.com/AMDResearch/omniperf.git
cd omniperf
git checkout dev
```

```
# define top-level install path
export INSTALL_DIR=${HOME}/opt/omniperf/

# install python deps
python3 -m pip install -t ${INSTALL_DIR}/python-libs -r requirements.txt

# configure Omniperf for shared install
mkdir build
cd build cmake -DCMAKE_INSTALL_PREFIX=${INSTALL_DIR} \
               -DPYTHON_DEPS=${INSTALL_DIR}/python-libs \
               -DMOD_INSTALL_PATH=${INSTALL_DIR}/modulefiles ..

# install
make install
```

prerequisites

- ROCm (>=5.2.0)
- CMake (>=3.19)
- Python™ (>=3.7)

INSTALLATION

LMOD SETUP

```
# set up Lmod
sudo apt install lmod
source /etc/profile.d/lmod.sh

# add the omniperf module to Lmod
module use $INSTALL_DIR/modulefiles
module avail

# load the omniperf module
module load omniperf

# check if module loaded
which omniperf

# check if omniperf runs
omniperf --version
```

using Lmod

```
export PATH=$INSTALL_DIR/bin:$PATH
export PYTHONPATH=$INSTALL_DIR/python-libs
```

without Lmod

INSTALLATION

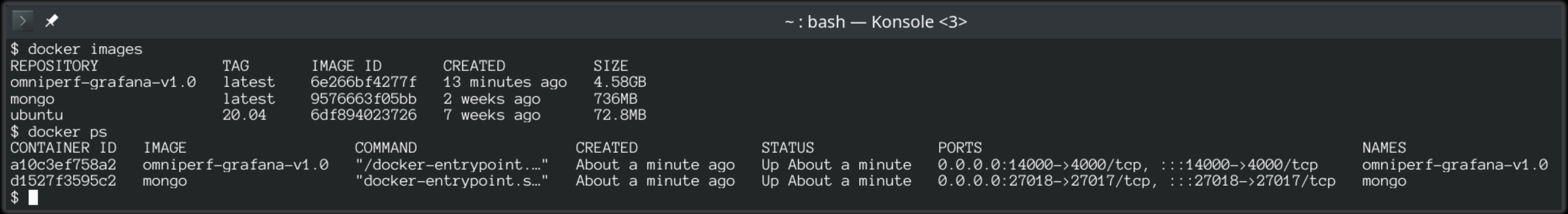
MONGODB® / GRAFANA™

```
# install MongoDB for your OS
# https://www.mongodb.com/download-center/database-tools/releases/archive
wget https://fastdl.mongodb.org/tools/db/mongodb-database-tools-ubuntu2204-x86_64-100.8.0.deb
sudo apt install ./mongodb-database-tools-ubuntu2204-x86_64-100.8.0.deb

# create local directories and bind them to Docker volumes
sudo mkdir -p /usr/local/persist && cd /usr/local/persist/
sudo mkdir -p grafana-storage mongodb
sudo docker volume create --driver local --opt type=none --opt device=/usr/local/persist/grafana-storage --opt o=bind grafana-storage
sudo docker volume create --driver local --opt type=none --opt device=/usr/local/persist/mongodb --opt o=bind grafana-mongo-db

# install Docker packages
sudo install docker.io
sudo install docker-compose

# go to the omniperf repository cloned from Github
# build and launch Docker containers for MongoDB and Grafana
sudo docker-compose build
sudo docker-compose up -d
```

INSTALLATION

MONGODB® / GRAFANA™

```
# list Docker images
docker images

# list active Docker containers
docker ps
```

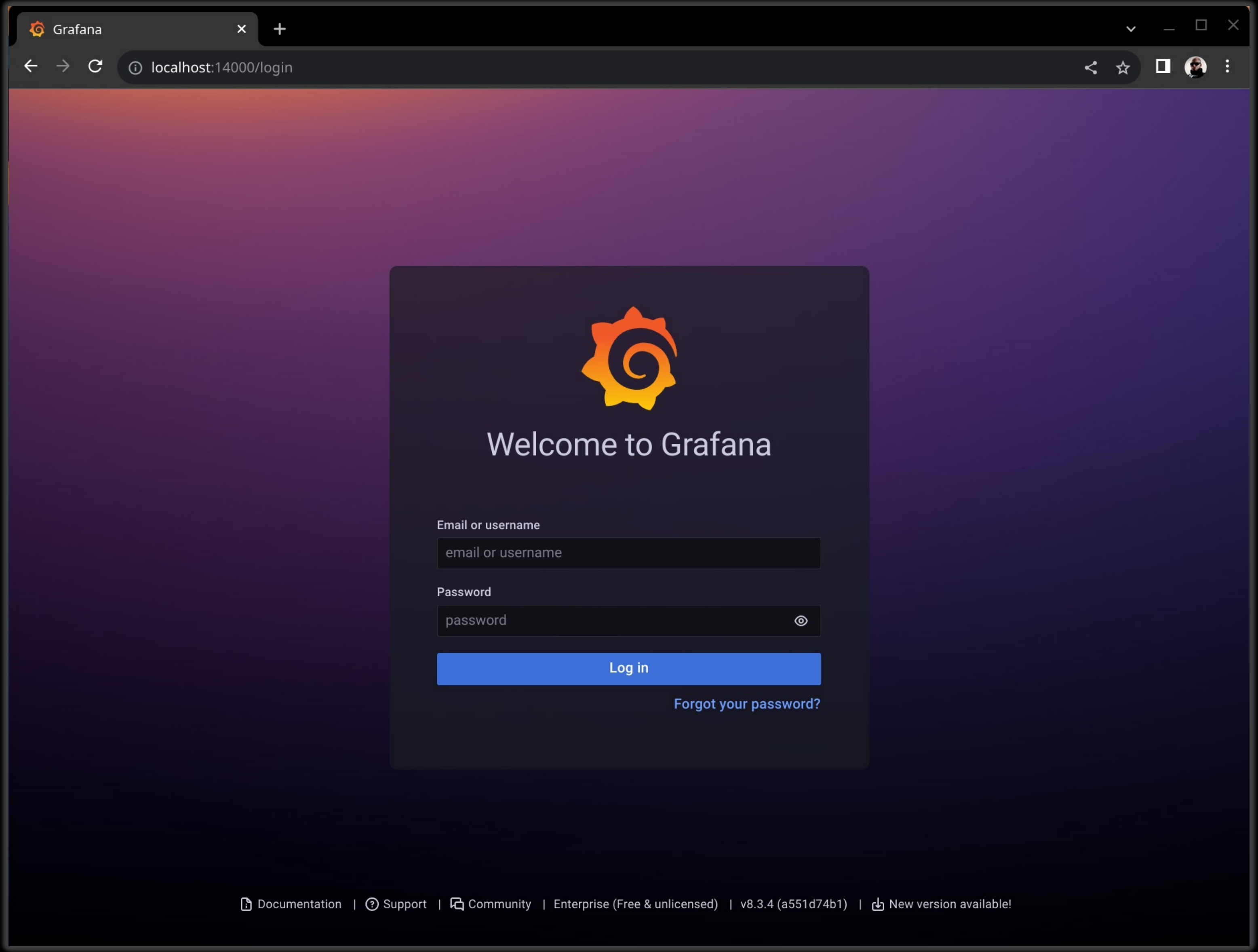


```
$ docker images
REPOSITORY             TAG       IMAGE ID       CREATED          SIZE
omniperf-grafana-v1.0  latest    6e266bf4277f   13 minutes ago   4.58GB
mongo                  latest    9576663f05bb   2 weeks ago      736MB
ubuntu                 20.04     6df894023726   7 weeks ago      72.8MB
$ docker ps
CONTAINER ID   IMAGE                   COMMAND                  CREATED              STATUS              PORTS                                           NAMES
a10c3ef758a2   omniperf-grafana-v1.0   "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:14000->4000/tcp, :::14000->4000/tcp     omniperf-grafana-v1.0
d1527f3595c2   mongo                   "docker-entrypoint.s…"   About a minute ago   Up About a minute   0.0.0.0:27018->27017/tcp, :::27018->27017/tcp   mongo
$ 
```

INSTALLATION

MONGODB® / GRAFANA™

- open a browser
- go to, e.g., http://localhost:14000/
- username: admin
- password: admin



Grafana
localhost:14000/login

Welcome to Grafana

Email or username

email or username

Password

password

Log in

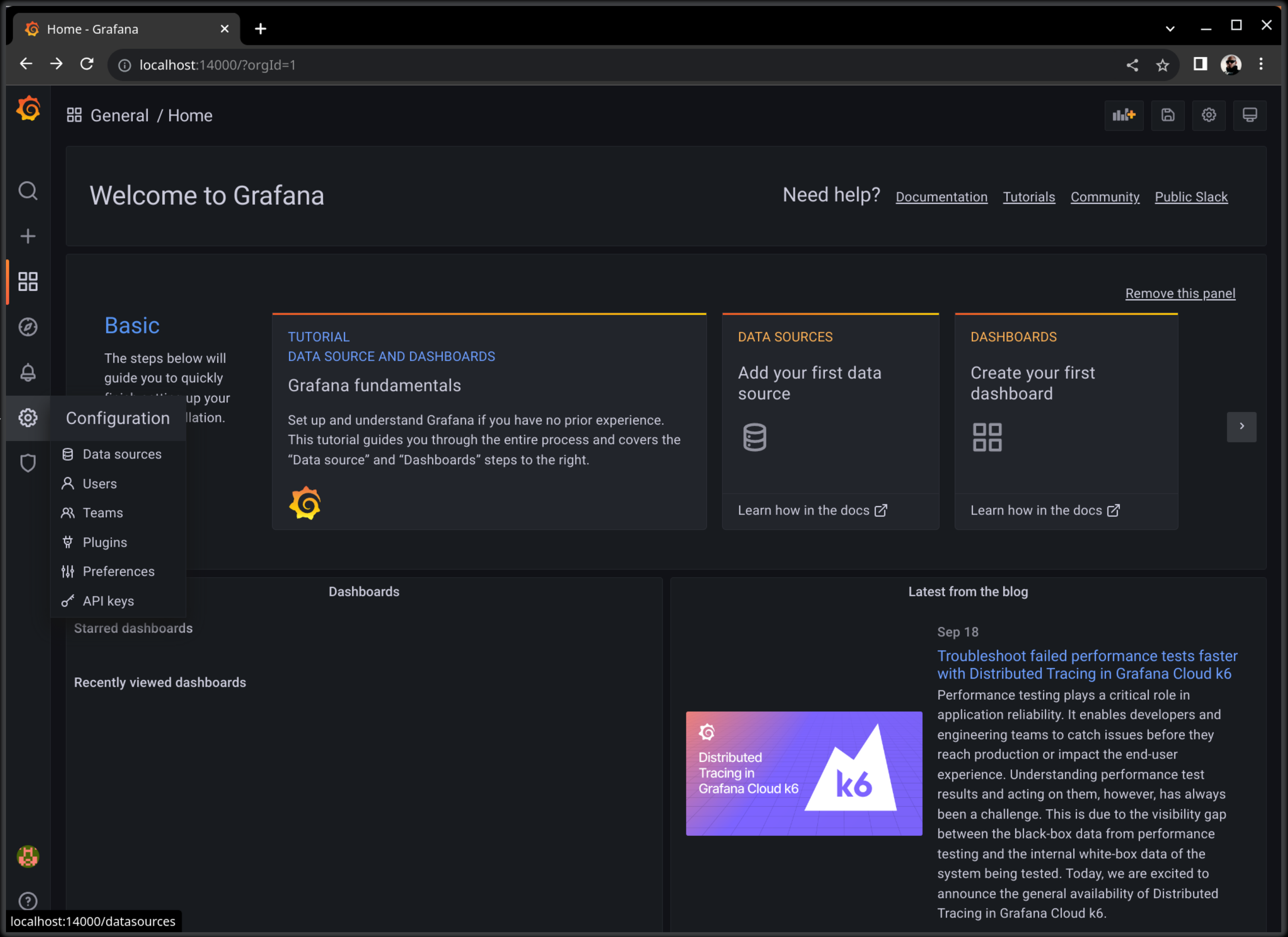
Forgot your password?

Documentation | Support | Community | Enterprise (Free & unlicensed) | v8.3.4 (a551d74b1) | New version available!

INSTALLATION

MONGODB® / GRAFANA™

go to settings



Home - Grafana

localhost:14000/?orgId=1

General / Home

Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Remove this panel

Basic

The steps below will guide you to quickly ... up your ...llation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the “Data source” and “Dashboards” steps to the right.

DATA SOURCES
Add your first data source

Learn how in the docs

DASHBOARDS
Create your first dashboard

Learn how in the docs

Dashboards

Starred dashboards

Recently viewed dashboards

Latest from the blog

Sep 18

Troubleshoot failed performance tests faster with Distributed Tracing in Grafana Cloud k6

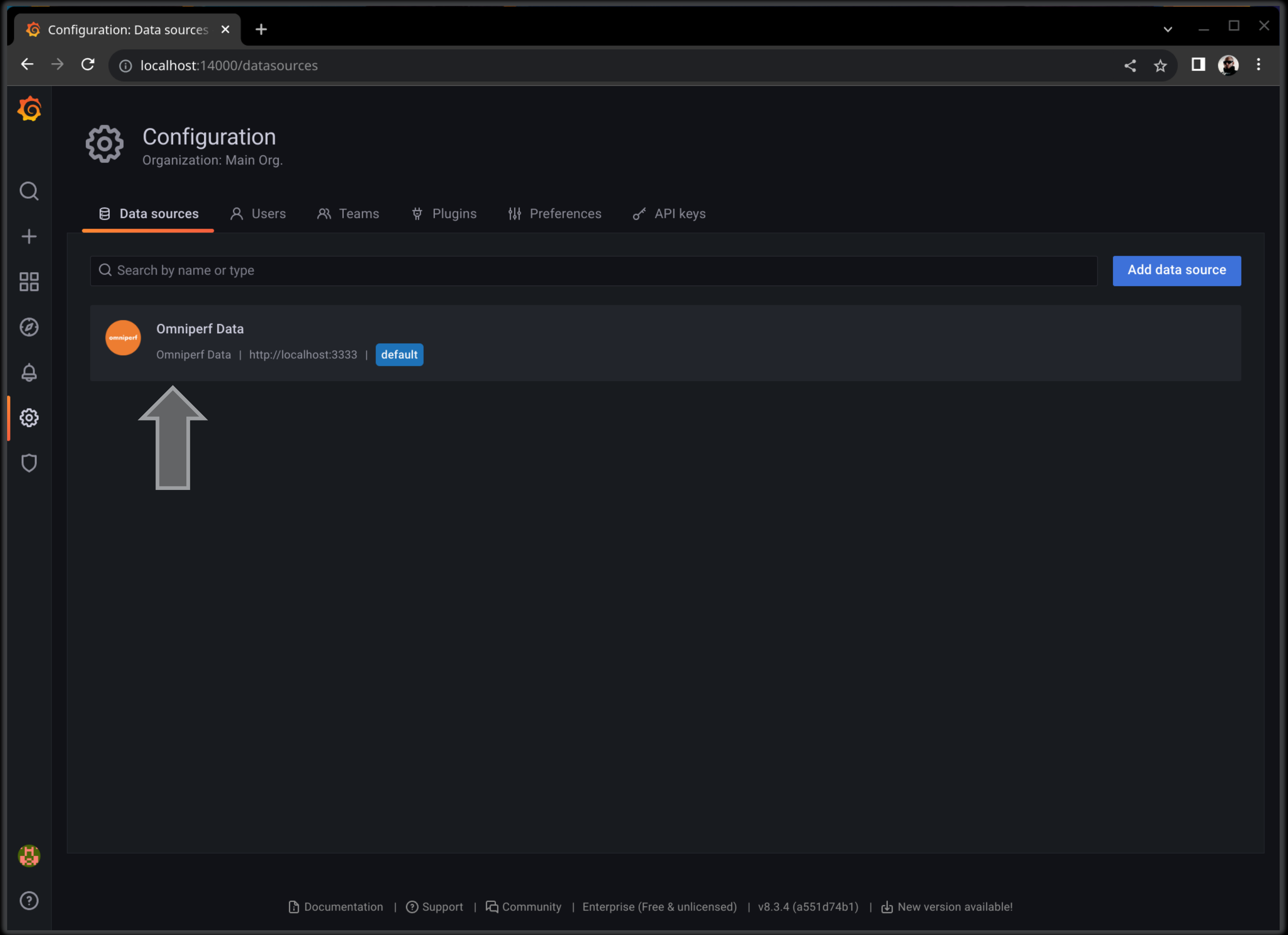
Performance testing plays a critical role in application reliability. It enables developers and engineering teams to catch issues before they reach production or impact the end-user experience. Understanding performance test results and acting on them, however, has always been a challenge. This is due to the visibility gap between the black-box data from performance testing and the internal white-box data of the system being tested. Today, we are excited to announce the general availability of Distributed Tracing in Grafana Cloud k6.

localhost:14000/datasources

INSTALLATION

MONGODB® / GRAFANA™

chose Omniperf Data


Configuration: Data sources

localhost:14000/datasources

Configuration

Organization: Main Org.

Data sources
Users
Teams
Plugins
Preferences
API keys
Search by name or type
Add data source

Omniperf Data

Omniperf Data | http://localhost:3333 | default

Documentation | Support | Community | Enterprise (Free & unlicensed) | v8.3.4 (a551d74b1) | New version available!

INSTALLATION

MONGODB® / GRAFANA™

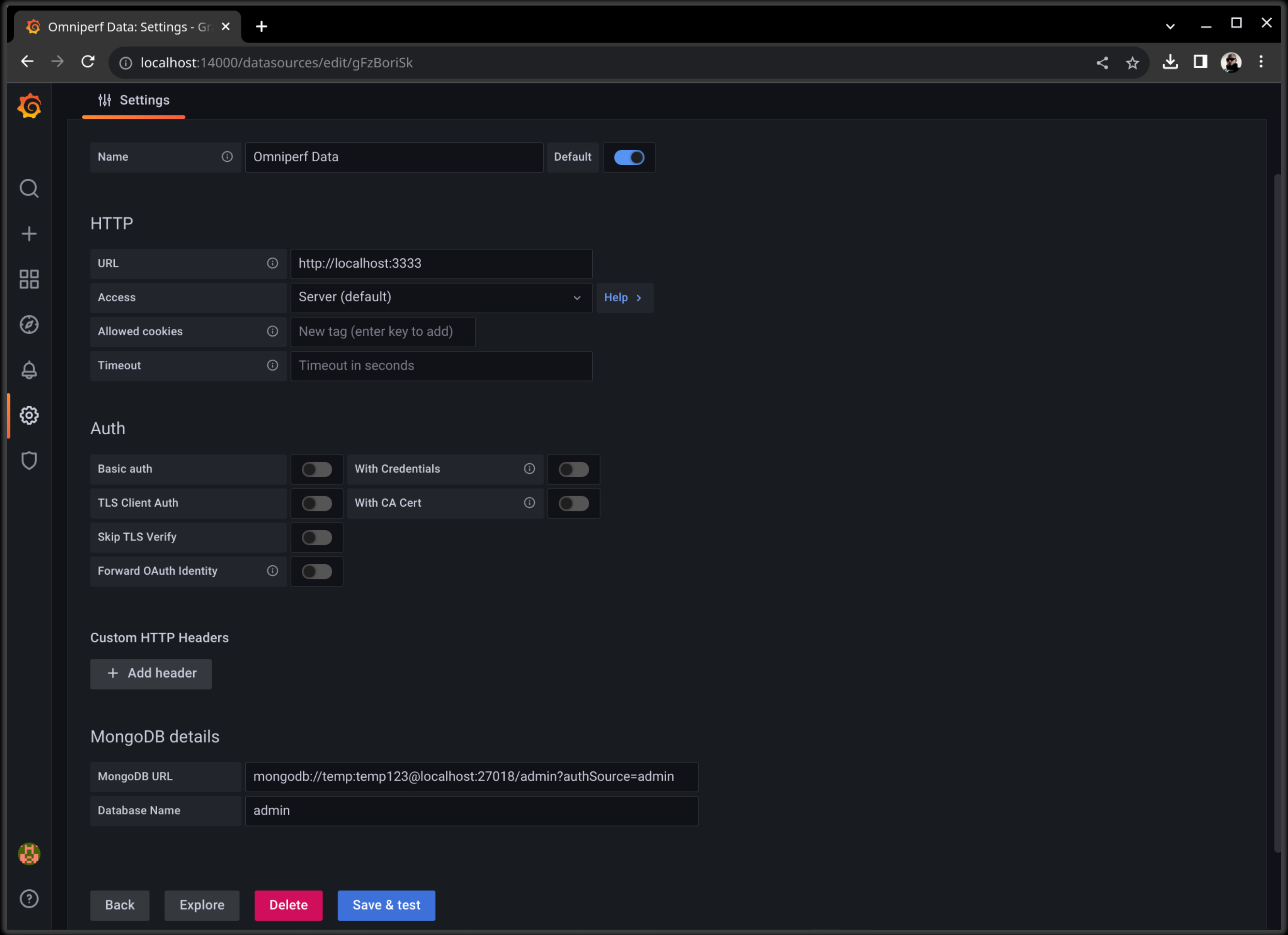
configure connection settings

- URL: http://localhost:3333
- MongoDB® URL:

  mongodb://temp:temp123@<host-ip>:27018/admin?authSource=admin
- Database Name: admin

get the <host-ip> from, e.g., ifconfig (should start with 10)

Save & Test



Omniperf Data: Settings - Gr

localhost:14000/datasources/edit/gFzBoriSk

Settings

Name: Omniperf Data
Default: on

HTTP

URL: http://localhost:3333
Access: Server (default)
Allowed cookies: New tag (enter key to add)
Timeout: Timeout in seconds

Auth

Basic auth: off
With Credentials: off
TLS Client Auth: off
With CA Cert: off
Skip TLS Verify: off
Forward OAuth Identity: off

Custom HTTP Headers

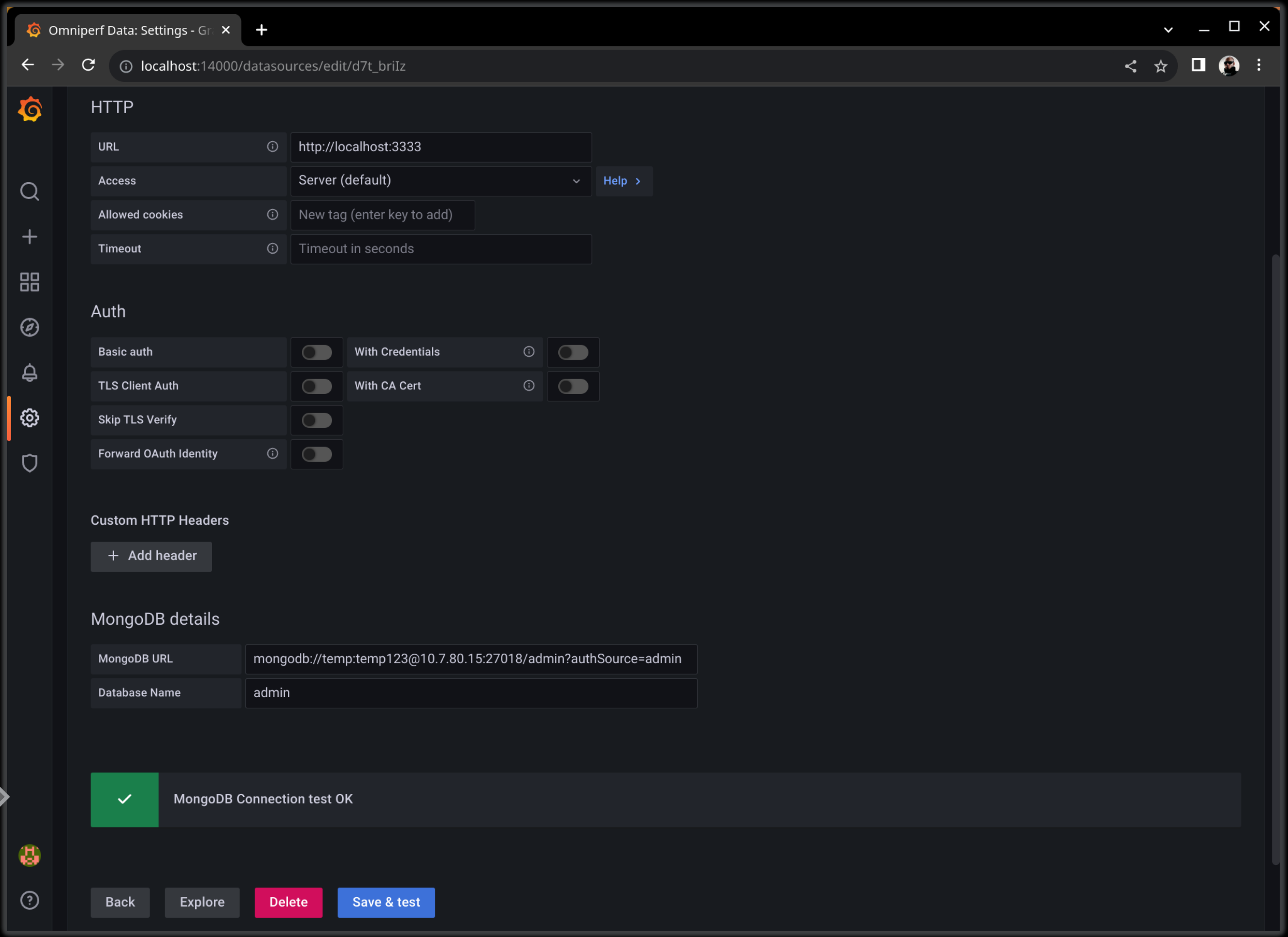
MongoDB details

MongoDB URL: mongodb://temp:temp123@localhost:27018/admin?authSource=admin
Database Name: admin

AMD
together we advance_

INSTALLATION

MONGODB® / GRAFANA™



Omniperf Data: Settings - Gr
localhost:14000/datasources/edit/d7t_brilz

HTTP

URL: http://localhost:3333

Access: Server (default)

Allowed cookies: New tag (enter key to add)

Timeout: Timeout in seconds

Auth

Basic auth: off

With Credentials: off

TLS Client Auth: off

With CA Cert: off

Skip TLS Verify: off

Forward OAuth Identity: off

Custom HTTP Headers

MongoDB details

MongoDB URL: mongodb://temp:temp123@10.7.80.15:27018/admin?authSource=admin

Database Name: admin

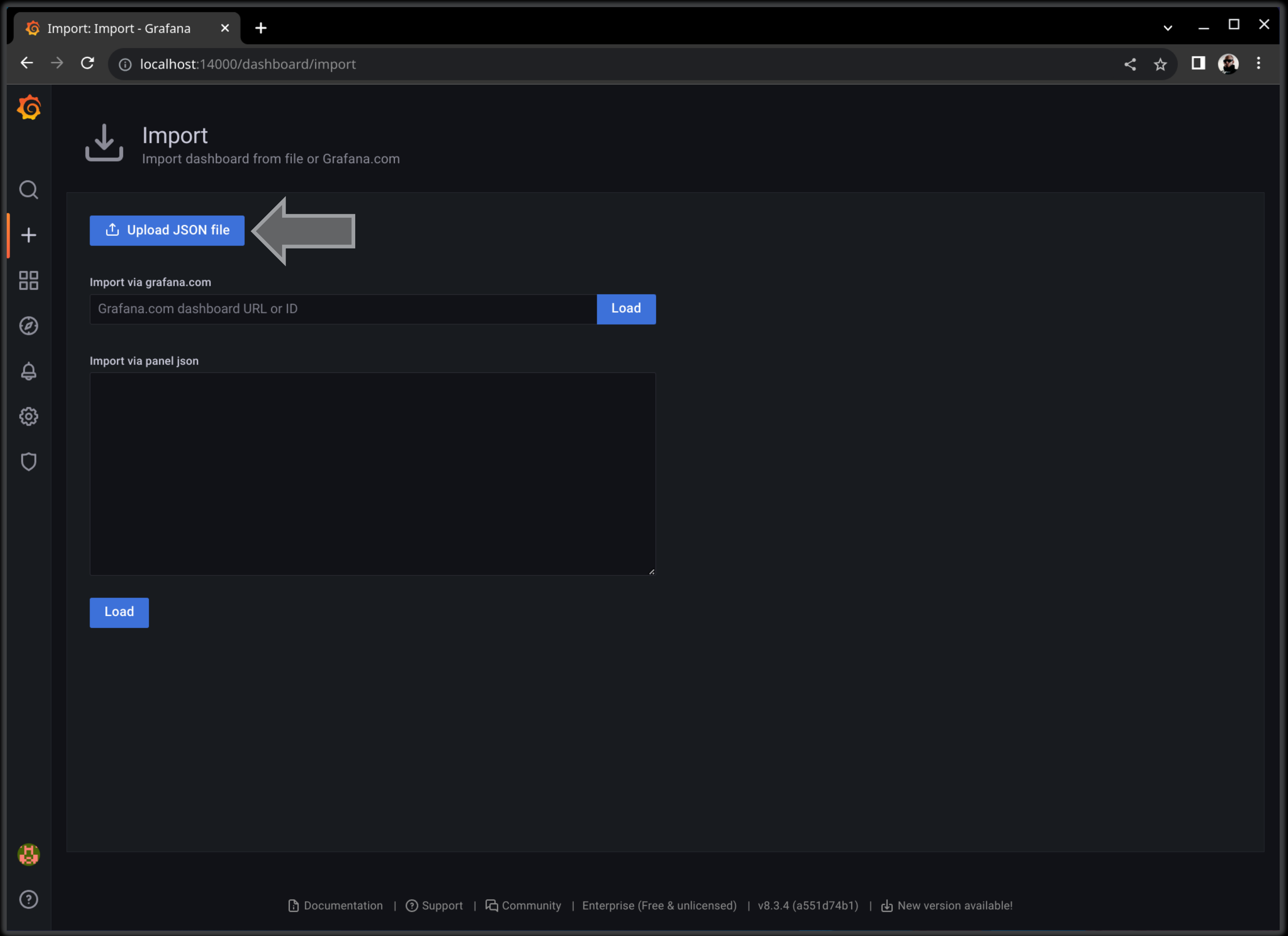
✓ MongoDB Connection test OK

if all goes well

INSTALLATION

MONGODB® / GRAFANA™

- go to Create (+) → Import
- click "Upload JSON file"
- import the dashboard file from the omniperf repository /dashboards/Omniperf_v1.0.8_pub.json

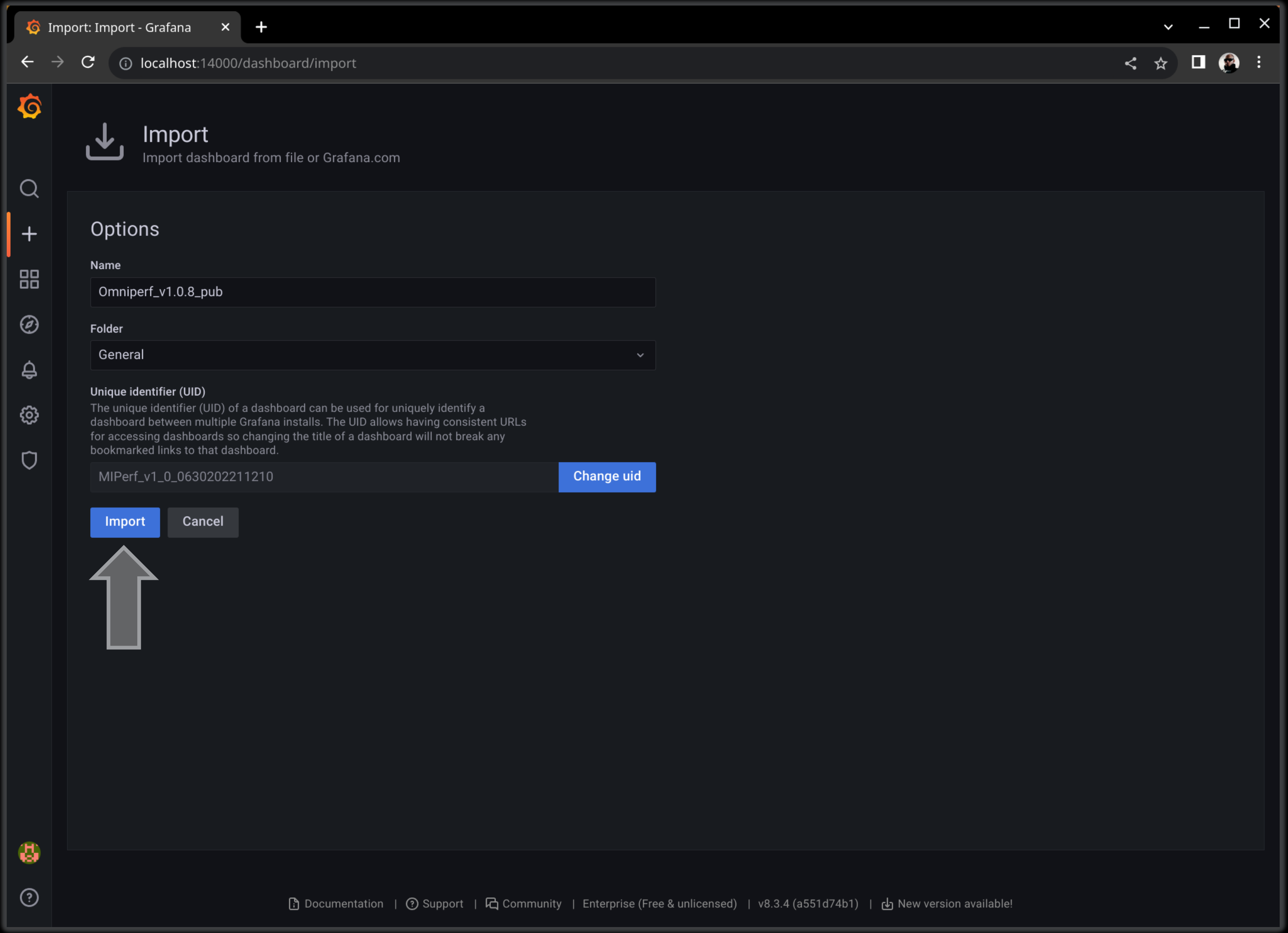

Import: Import - Grafana
localhost:14000/dashboard/import
Import
Import dashboard from file or Grafana.com
Upload JSON file
Import via grafana.com
Grafana.com dashboard URL or ID
Load
Import via panel json
Load
Documentation | Support | Community | Enterprise (Free & unlicensed) | v8.3.4 (a551d74b1) | New version available!

AMD
together we advance_

INSTALLATION

MONGODB® / GRAFANA™

- go to Create (+) → Import
- click “Upload JSON file”
- import the dashboard file from the omniperf repository /dashboards/Omniperf_v1.0.8_pub.json



Import: Import - Grafana

localhost:14000/dashboard/import

Import

Import dashboard from file or Grafana.com

Options

Name: Omniperf_v1.0.8_pub

Folder: General

Unique identifier (UID)

The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

MIPerf_v1_0_0630202211210

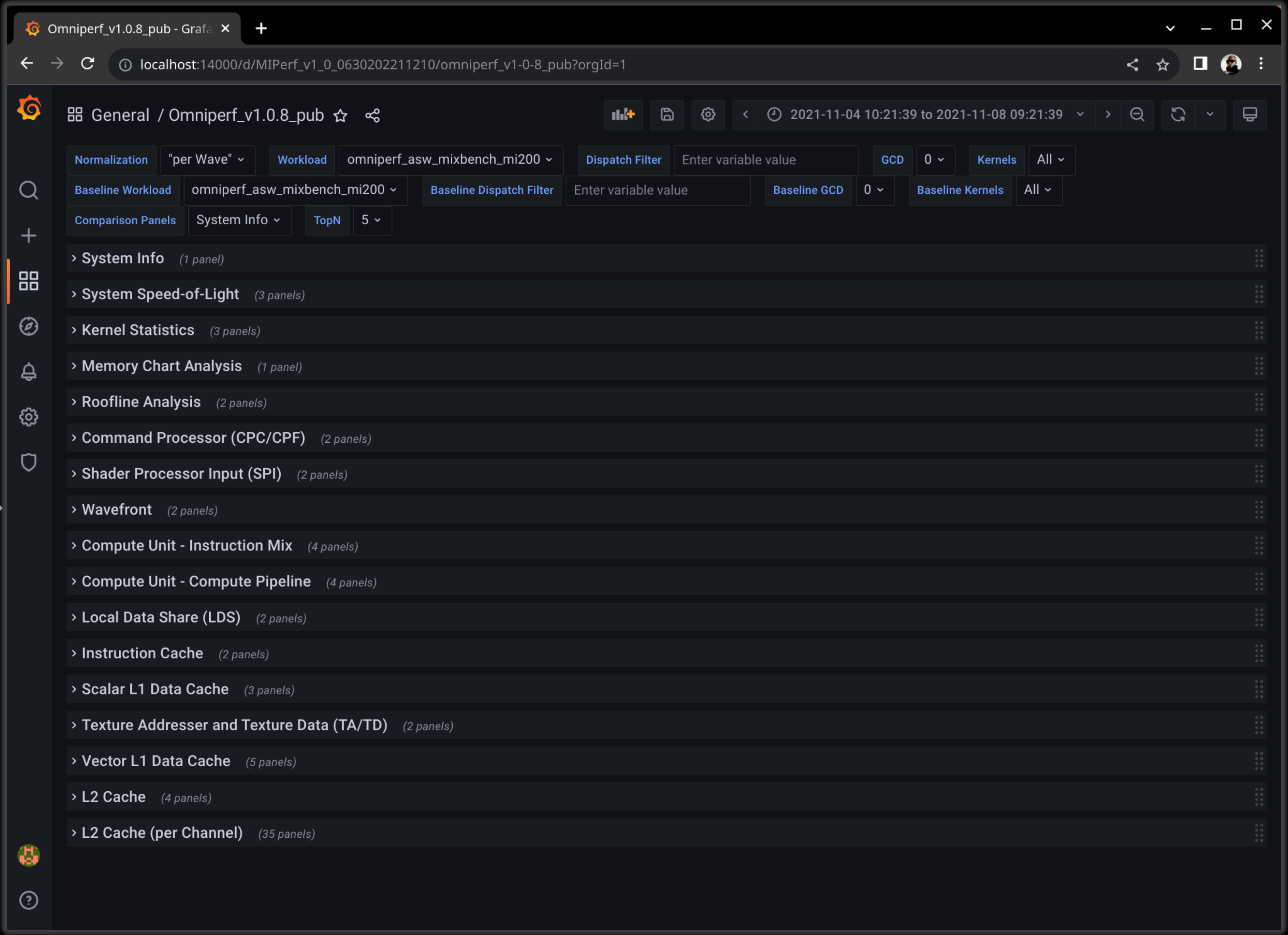
Change uid
Import
Cancel
Documentation | Support | Community | Enterprise (Free & unlicensed) | v8.3.4 (a551d74b1) | New version available!

INSTALLATION

MONGODB® / GRAFANA™

if all goes well

tabs of the imported dashboard



Omniperf_v1.0.8_pub - Grafa

localhost:14000/d/MIPerf_v1_0_0630202211210/omniperf_v1-0-8_pub?orgId=1

General / Omniperf_v1.0.8_pub

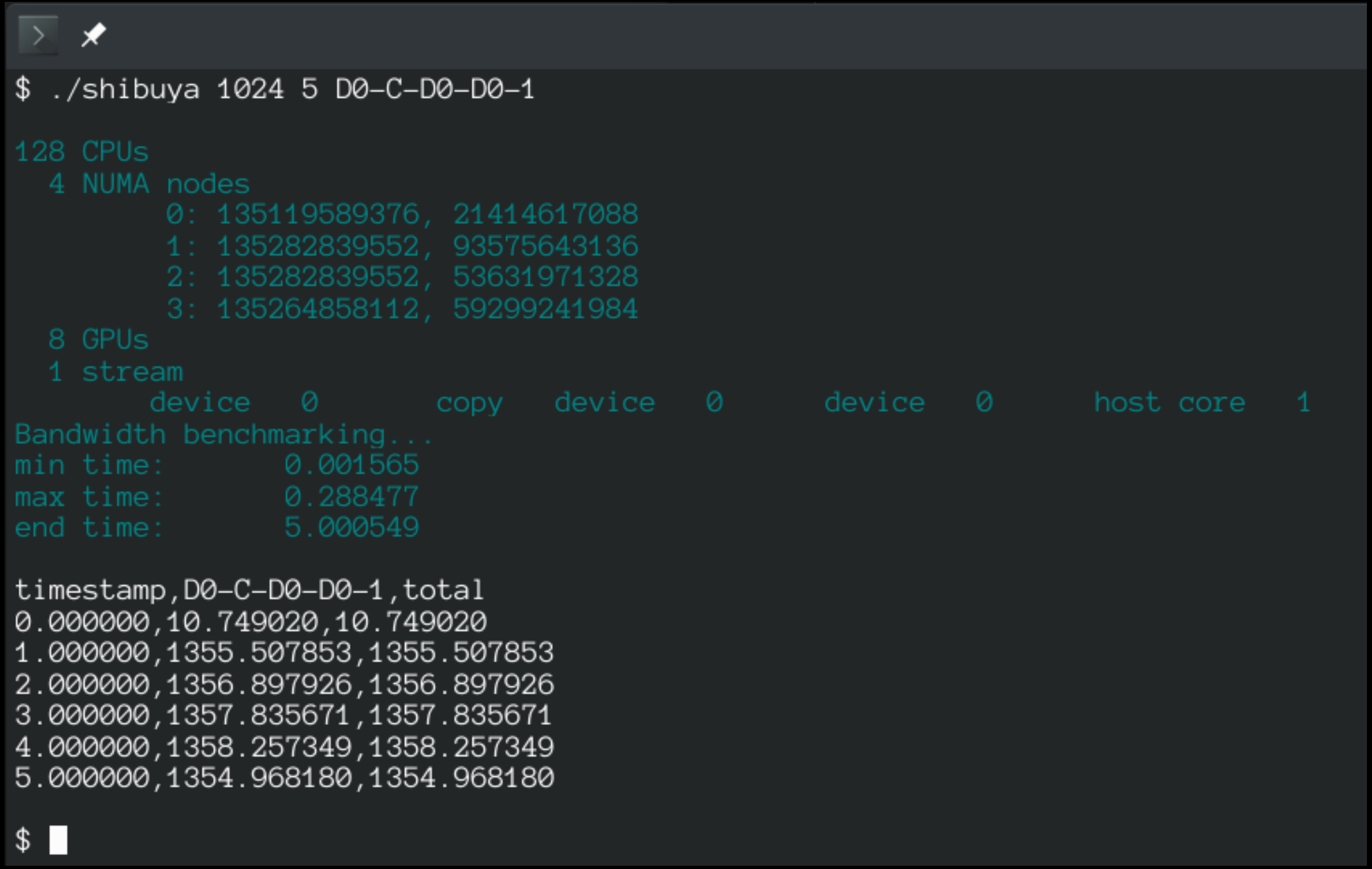
2021-11-04 10:21:39 to 2021-11-08 09:21:39

Normalization: "per Wave"
Workload: omniperf_asw_mixbench_mi200
Dispatch Filter: Enter variable value
GCD: 0
Kernels: All
Baseline Workload: omniperf_asw_mixbench_mi200
Baseline Dispatch Filter: Enter variable value
Baseline GCD: 0
Baseline Kernels: All
Comparison Panels: System Info
TopN: 5

- System Info (1 panel)
- System Speed-of-Light (3 panels)
- Kernel Statistics (3 panels)
- Memory Chart Analysis (1 panel)
- Roofline Analysis (2 panels)
- Command Processor (CPC/CPF) (2 panels)
- Shader Processor Input (SPI) (2 panels)
- Wavefront (2 panels)
- Compute Unit - Instruction Mix (4 panels)
- Compute Unit - Compute Pipeline (4 panels)
- Local Data Share (LDS) (2 panels)
- Instruction Cache (2 panels)
- Scalar L1 Data Cache (3 panels)
- Texture Addresser and Texture Data (TA/TD) (2 panels)
- Vector L1 Data Cache (5 panels)
- L2 Cache (4 panels)
- L2 Cache (per Channel) (35 panels)

BANDWIDTH BENCHMARK

SHIBUYA STREAM



```
$ ./shibuya 1024 5 D0-C-D0-D0-1

128 CPUs
  4 NUMA nodes
         0: 135119589376, 21414617088
         1: 135282839552, 93575643136
         2: 135282839552, 53631971328
         3: 135264858112, 59299241984
  8 GPUs
  1 stream
        device   0        copy   device   0       device   0       host core   1
Bandwidth benchmarking...
min time:       0.001565
max time:       0.288477
end time:       5.000549

timestamp,D0-C-D0-D0-1,total
0.000000,10.749020,10.749020
1.000000,1355.507853,1355.507853
2.000000,1356.897926,1356.897926
3.000000,1357.835671,1357.835671
4.000000,1358.257349,1358.257349
5.000000,1354.968180,1354.968180

$ █
```

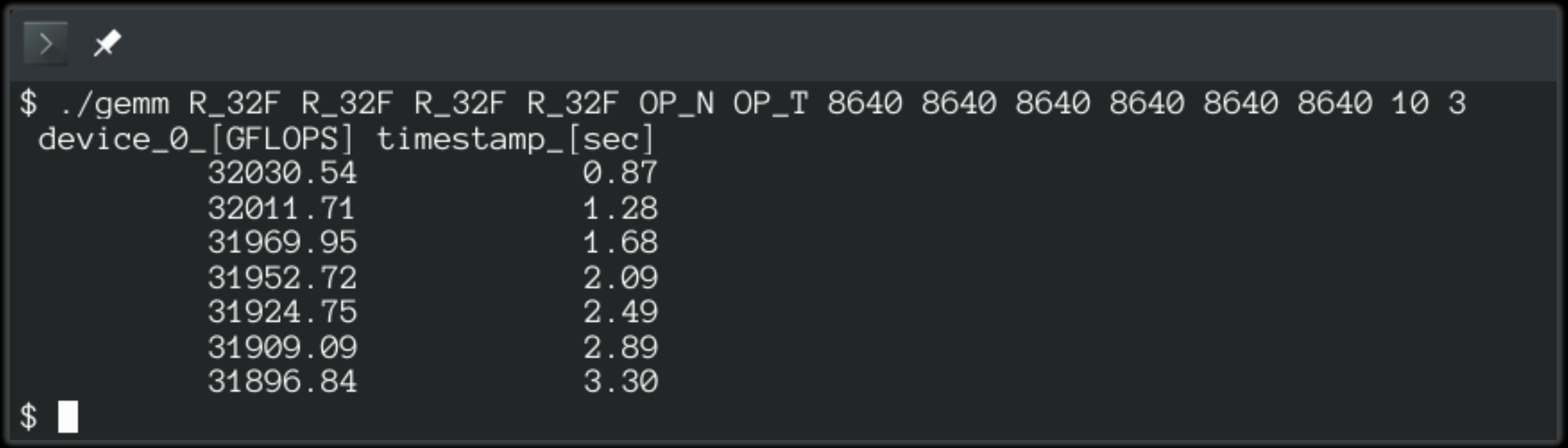
- https://github.com/AMD-HPC/ShibuyaStream
- bandwidth benchmark
  - GPU↔HBM
  - CPU↔DDR
  - GPU↔DDR
  - CPU↔HBM
- BabelStream workloads (copy, add, triad, ...)

- one MI250X GCD
- ~1.3 TBPS

COMPUTE BENCHMARK

CORAL GEMM

- https://github.com/AMD-HPC/CoralGemm
- matrix multiplication benchmark
- all precisions including FP16 and BF16
- complete GEMM API, including batched and strided
- runs by default on all devices in the system



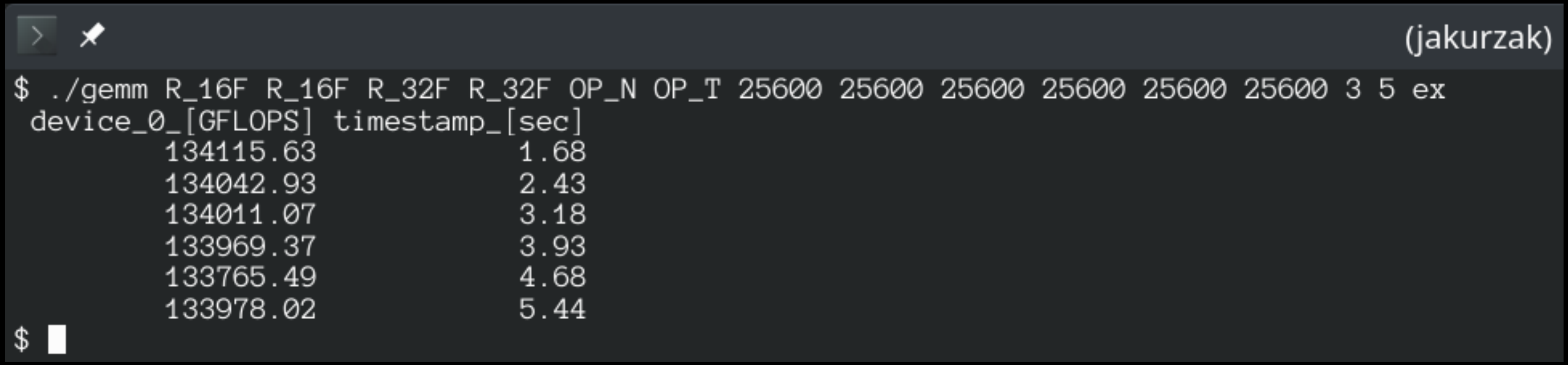
```
$ ./gemm R_32F R_32F R_32F R_32F OP_N OP_T 8640 8640 8640 8640 8640 8640 10 3
 device_0_[GFLOPS] timestamp_[sec]
          32030.54            0.87
          32011.71            1.28
          31969.95            1.68
          31952.72            2.09
          31924.75            2.49
          31909.09            2.89
          31896.84            3.30
$ █
```

- one MI250X GCD
- ~32 TFLOPS in FP32

COMPUTE BENCHMARK

CORAL GEMM

- https://github.com/AMD-HPC/CoralGemm
- matrix multiplication benchmark
- all precisions including FP16 and BF16
- complete GEMM API, including batched and strided
- runs by default on all devices in the system



```
$ ./gemm R_16F R_16F R_32F R_32F OP_N OP_T 25600 25600 25600 25600 25600 25600 3 5 ex
 device_0_[GFLOPS] timestamp_[sec]
         134115.63            1.68
         134042.93            2.43
         134011.07            3.18
         133969.37            3.93
         133765.49            4.68
         133978.02            5.44
$ 
```

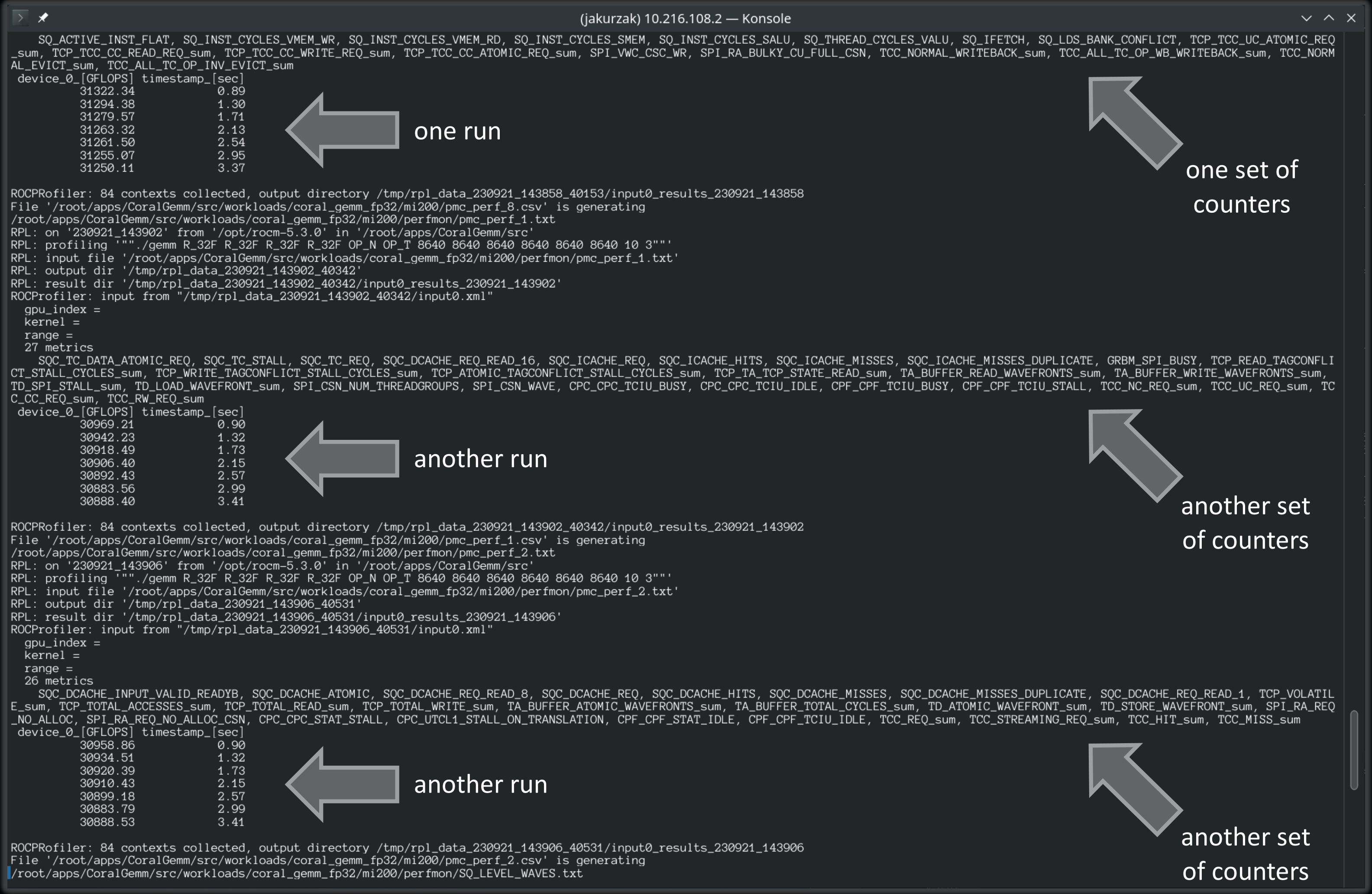
- one MI250X GCD
- ~134 TFLOPS in FP16

COMPUTE BENCHMARK

CORAL GEMM

```
omniperf profile -n coral_gemm_fp16 -- ./gemm ...
```

- profile the workload
- collect performance counters
- replay to collect many counters
- filtering options available

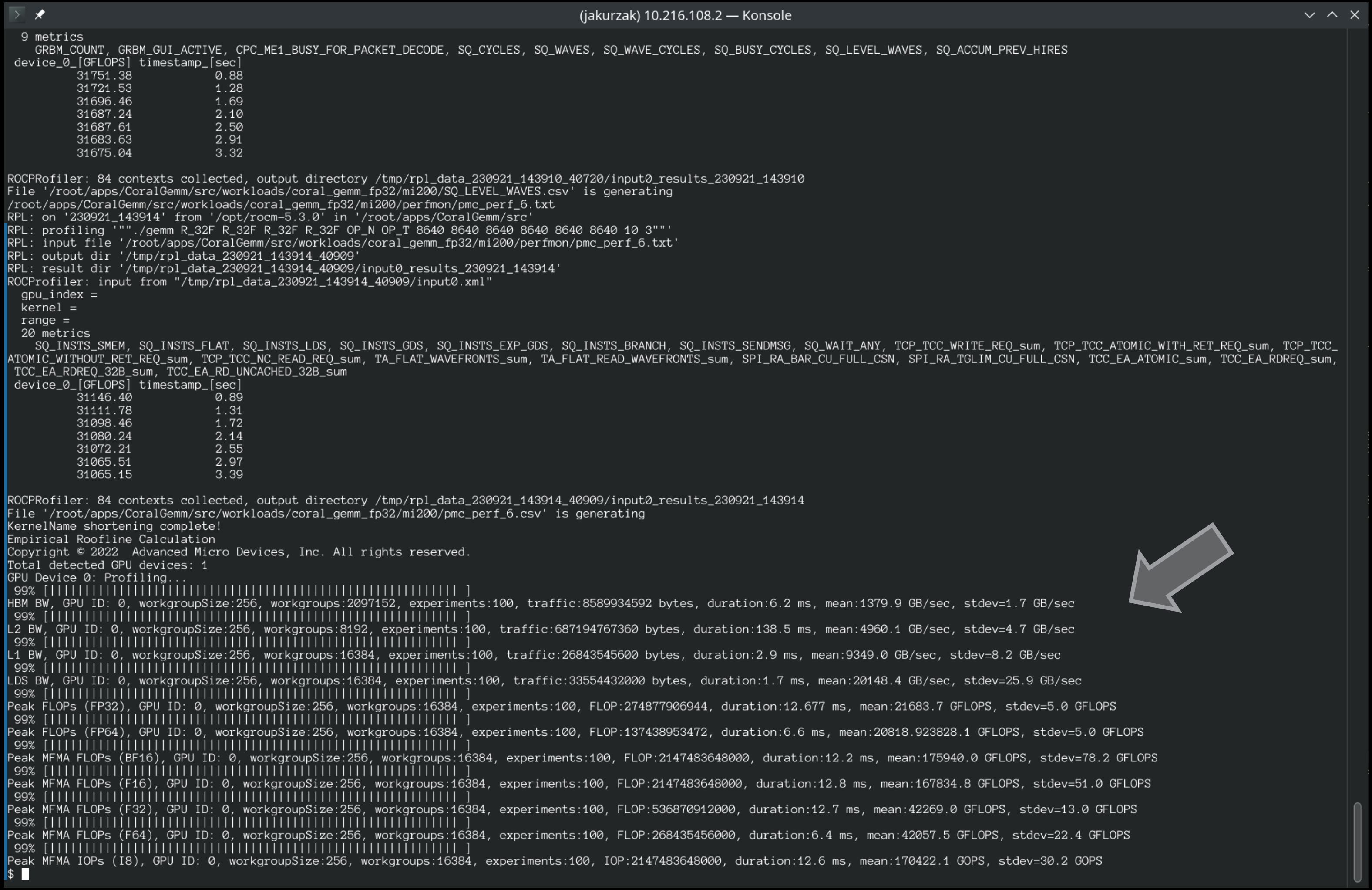


```
    SQ_ACTIVE_INST_FLAT, SQ_INST_CYCLES_VMEM_WR, SQ_INST_CYCLES_VMEM_RD, SQ_INST_CYCLES_SMEM, SQ_INST_CYCLES_SALU, SQ_THREAD_CYCLES_VALU, SQ_IFETCH, SQ_LDS_BANK_CONFLICT, TCP_TCC_UC_ATOMIC_REQ
_sum, TCP_TCC_CC_READ_REQ_sum, TCP_TCC_CC_WRITE_REQ_sum, TCP_TCC_CC_ATOMIC_REQ_sum, SPI_VWC_CSC_WR, SPI_RA_BULKY_CU_FULL_CSN, TCC_NORMAL_WRITEBACK_sum, TCC_ALL_TC_OP_WB_WRITEBACK_sum, TCC_NORM
AL_EVICT_sum, TCC_ALL_TC_OP_INV_EVICT_sum
 device_0_[GFLOPS] timestamp_[sec]
          31322.34           0.89
          31294.38           1.30
          31279.57           1.71
          31263.32           2.13
          31261.50           2.54
          31255.07           2.95
          31250.11           3.37

ROCPRofiler: 84 contexts collected, output directory /tmp/rpl_data_230921_143858_40153/input0_results_230921_143858
File '/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/pmc_perf_8.csv' is generating
/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/perfmon/pmc_perf_1.txt
RPL: on '230921_143902' from '/opt/rocm-5.3.0' in '/root/apps/CoralGemm/src'
RPL: profiling '""./gemm R_32F R_32F R_32F R_32F OP_N OP_T 8640 8640 8640 8640 8640 8640 10 3""'
RPL: input file '/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/perfmon/pmc_perf_1.txt'
RPL: output dir '/tmp/rpl_data_230921_143902_40342'
RPL: result dir '/tmp/rpl_data_230921_143902_40342/input0_results_230921_143902'
ROCProfiler: input from "/tmp/rpl_data_230921_143902_40342/input0.xml"
  gpu_index =
  kernel =
  range =
  27 metrics
    SQC_TC_DATA_ATOMIC_REQ, SQC_TC_STALL, SQC_TC_REQ, SQC_DCACHE_REQ_READ_16, SQC_ICACHE_REQ, SQC_ICACHE_HITS, SQC_ICACHE_MISSES, SQC_ICACHE_MISSES_DUPLICATE, GRBM_SPI_BUSY, TCP_READ_TAGCONFLI
CT_STALL_CYCLES_sum, TCP_WRITE_TAGCONFLICT_STALL_CYCLES_sum, TCP_ATOMIC_TAGCONFLICT_STALL_CYCLES_sum, TCP_TA_TCP_STATE_READ_sum, TA_BUFFER_READ_WAVEFRONTS_sum, TA_BUFFER_WRITE_WAVEFRONTS_sum,
TD_SPI_STALL_sum, TD_LOAD_WAVEFRONT_sum, SPI_CSN_NUM_THREADGROUPS, SPI_CSN_WAVE, CPC_CPC_TCIU_BUSY, CPC_CPC_TCIU_IDLE, CPF_CPF_TCIU_BUSY, CPF_CPF_TCIU_STALL, TCC_NC_REQ_sum, TCC_UC_REQ_sum, TC
C_CC_REQ_sum, TCC_RW_REQ_sum
 device_0_[GFLOPS] timestamp_[sec]
          30969.21           0.90
          30942.23           1.32
          30918.49           1.73
          30906.40           2.15
          30892.43           2.57
          30883.56           2.99
          30888.40           3.41

ROCPRofiler: 84 contexts collected, output directory /tmp/rpl_data_230921_143902_40342/input0_results_230921_143902
File '/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/pmc_perf_1.csv' is generating
/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/perfmon/pmc_perf_2.txt
RPL: on '230921_143906' from '/opt/rocm-5.3.0' in '/root/apps/CoralGemm/src'
RPL: profiling '""./gemm R_32F R_32F R_32F R_32F OP_N OP_T 8640 8640 8640 8640 8640 8640 10 3""'
RPL: input file '/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/perfmon/pmc_perf_2.txt'
RPL: output dir '/tmp/rpl_data_230921_143906_40531'
RPL: result dir '/tmp/rpl_data_230921_143906_40531/input0_results_230921_143906'
ROCProfiler: input from "/tmp/rpl_data_230921_143906_40531/input0.xml"
  gpu_index =
  kernel =
  range =
  26 metrics
    SQC_DCACHE_INPUT_VALID_READYB, SQC_DCACHE_ATOMIC, SQC_DCACHE_REQ_READ_8, SQC_DCACHE_REQ, SQC_DCACHE_HITS, SQC_DCACHE_MISSES, SQC_DCACHE_MISSES_DUPLICATE, SQC_DCACHE_REQ_READ_1, TCP_VOLATIL
E_sum, TCP_TOTAL_ACCESSES_sum, TCP_TOTAL_READ_sum, TCP_TOTAL_WRITE_sum, TA_BUFFER_ATOMIC_WAVEFRONTS_sum, TA_BUFFER_TOTAL_CYCLES_sum, TD_ATOMIC_WAVEFRONT_sum, TD_STORE_WAVEFRONT_sum, SPI_RA_REQ
_NO_ALLOC, SPI_RA_REQ_NO_ALLOC_CSN, CPC_CPC_STAT_STALL, CPC_UTCL1_STALL_ON_TRANSLATION, CPF_CPF_STAT_IDLE, CPF_CPF_TCIU_IDLE, TCC_REQ_sum, TCC_STREAMING_REQ_sum, TCC_HIT_sum, TCC_MISS_sum
 device_0_[GFLOPS] timestamp_[sec]
          30958.86           0.90
          30934.51           1.32
          30920.39           1.73
          30910.43           2.15
          30899.18           2.57
          30883.79           2.99
          30888.53           3.41

ROCPRofiler: 84 contexts collected, output directory /tmp/rpl_data_230921_143906_40531/input0_results_230921_143906
File '/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/pmc_perf_2.csv' is generating
/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/perfmon/SQ_LEVEL_WAVES.txt
```

COMPUTE BENCHMARK

CORAL GEMM

```
omniperf profile -n coral_gemm_fp16 -- ./gemm ...
```



```
  9 metrics
    GRBM_COUNT, GRBM_GUI_ACTIVE, CPC_ME1_BUSY_FOR_PACKET_DECODE, SQ_CYCLES, SQ_WAVES, SQ_WAVE_CYCLES, SQ_BUSY_CYCLES, SQ_LEVEL_WAVES, SQ_ACCUM_PREV_HIRES
 device_0_[GFLOPS] timestamp_[sec]
          31751.38            0.88
          31721.53            1.28
          31696.46            1.69
          31687.24            2.10
          31687.61            2.50
          31683.63            2.91
          31675.04            3.32

ROCPRofiler: 84 contexts collected, output directory /tmp/rpl_data_230921_143910_40720/input0_results_230921_143910
File '/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/SQ_LEVEL_WAVES.csv' is generating
/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/perfmon/pmc_perf_6.txt
RPL: on '230921_143914' from '/opt/rocm-5.3.0' in '/root/apps/CoralGemm/src'
RPL: profiling '""./gemm R_32F R_32F R_32F R_32F OP_N OP_T 8640 8640 8640 8640 8640 8640 10 3""'
RPL: input file '/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/perfmon/pmc_perf_6.txt'
RPL: output dir '/tmp/rpl_data_230921_143914_40909'
RPL: result dir '/tmp/rpl_data_230921_143914_40909/input0_results_230921_143914'
ROCProfiler: input from "/tmp/rpl_data_230921_143914_40909/input0.xml"
  gpu_index =
  kernel =
  range =
  20 metrics
    SQ_INSTS_SMEM, SQ_INSTS_FLAT, SQ_INSTS_LDS, SQ_INSTS_GDS, SQ_INSTS_EXP_GDS, SQ_INSTS_BRANCH, SQ_INSTS_SENDMSG, SQ_WAIT_ANY, TCP_TCC_WRITE_REQ_sum, TCP_TCC_ATOMIC_WITH_RET_REQ_sum, TCP_TCC_
ATOMIC_WITHOUT_RET_REQ_sum, TCP_TCC_NC_READ_REQ_sum, TA_FLAT_WAVEFRONTS_sum, TA_FLAT_READ_WAVEFRONTS_sum, SPI_RA_BAR_CU_FULL_CSN, SPI_RA_TGLIM_CU_FULL_CSN, TCC_EA_ATOMIC_sum, TCC_EA_RDREQ_sum,
 TCC_EA_RDREQ_32B_sum, TCC_EA_RD_UNCACHED_32B_sum
 device_0_[GFLOPS] timestamp_[sec]
          31146.40            0.89
          31111.78            1.31
          31098.46            1.72
          31080.24            2.14
          31072.21            2.55
          31065.51            2.97
          31065.15            3.39

ROCPRofiler: 84 contexts collected, output directory /tmp/rpl_data_230921_143914_40909/input0_results_230921_143914
File '/root/apps/CoralGemm/src/workloads/coral_gemm_fp32/mi200/pmc_perf_6.csv' is generating
KernelName shortening complete!
Empirical Roofline Calculation
Copyright © 2022  Advanced Micro Devices, Inc. All rights reserved.
Total detected GPU devices: 1
GPU Device 0: Profiling...
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
HBM BW, GPU ID: 0, workgroupSize:256, workgroups:2097152, experiments:100, traffic:8589934592 bytes, duration:6.2 ms, mean:1379.9 GB/sec, stdev=1.7 GB/sec
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
L2 BW, GPU ID: 0, workgroupSize:256, workgroups:8192, experiments:100, traffic:687194767360 bytes, duration:138.5 ms, mean:4960.1 GB/sec, stdev=4.7 GB/sec
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
L1 BW, GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, traffic:26843545600 bytes, duration:2.9 ms, mean:9349.0 GB/sec, stdev=8.2 GB/sec
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
LDS BW, GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, traffic:33554432000 bytes, duration:1.7 ms, mean:20148.4 GB/sec, stdev=25.9 GB/sec
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
Peak FLOPs (FP32), GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, FLOP:274877906944, duration:12.677 ms, mean:21683.7 GFLOPS, stdev=5.0 GFLOPS
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
Peak FLOPs (FP64), GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, FLOP:137438953472, duration:6.6 ms, mean:20818.923828.1 GFLOPS, stdev=5.0 GFLOPS
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
Peak MFMA FLOPs (BF16), GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, FLOP:2147483648000, duration:12.2 ms, mean:175940.0 GFLOPS, stdev=78.2 GFLOPS
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
Peak MFMA FLOPs (F16), GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, FLOP:2147483648000, duration:12.8 ms, mean:167834.8 GFLOPS, stdev=51.0 GFLOPS
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
Peak MFMA FLOPs (F32), GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, FLOP:536870912000, duration:12.7 ms, mean:42269.0 GFLOPS, stdev=13.0 GFLOPS
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
Peak MFMA FLOPs (F64), GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, FLOP:268435456000, duration:6.4 ms, mean:42057.5 GFLOPS, stdev=22.4 GFLOPS
 99% [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| ]
Peak MFMA IOPs (I8), GPU ID: 0, workgroupSize:256, workgroups:16384, experiments:100, IOP:2147483648000, duration:12.6 ms, mean:170422.1 GOPS, stdev=30.2 GOPS
$ 
```

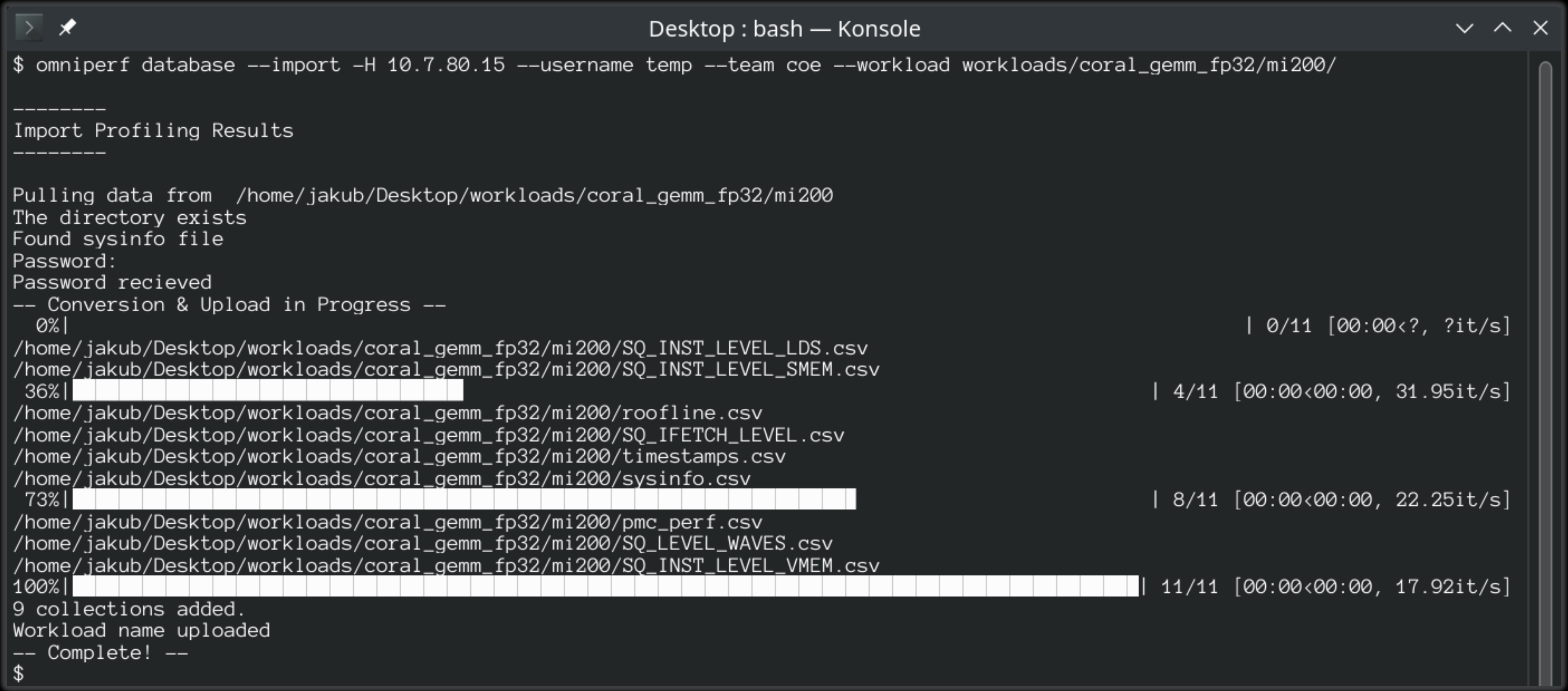
at the end

- build the roofline model of the device
- run synthetic benchmarks to measure
  - HBM bandwidth
  - L2 bandwidth
  - L1 bandwidth
  - LDS bandwidth
  - FLOPS
  - MFMA FLOPS

DATABASE IMPORT

MONGODB® / GRAFANA™

```
omniperf database --import -H 10.7.80.15 --username temp --team coe --workload workloads/coral_gemm_fp32/mi200
```

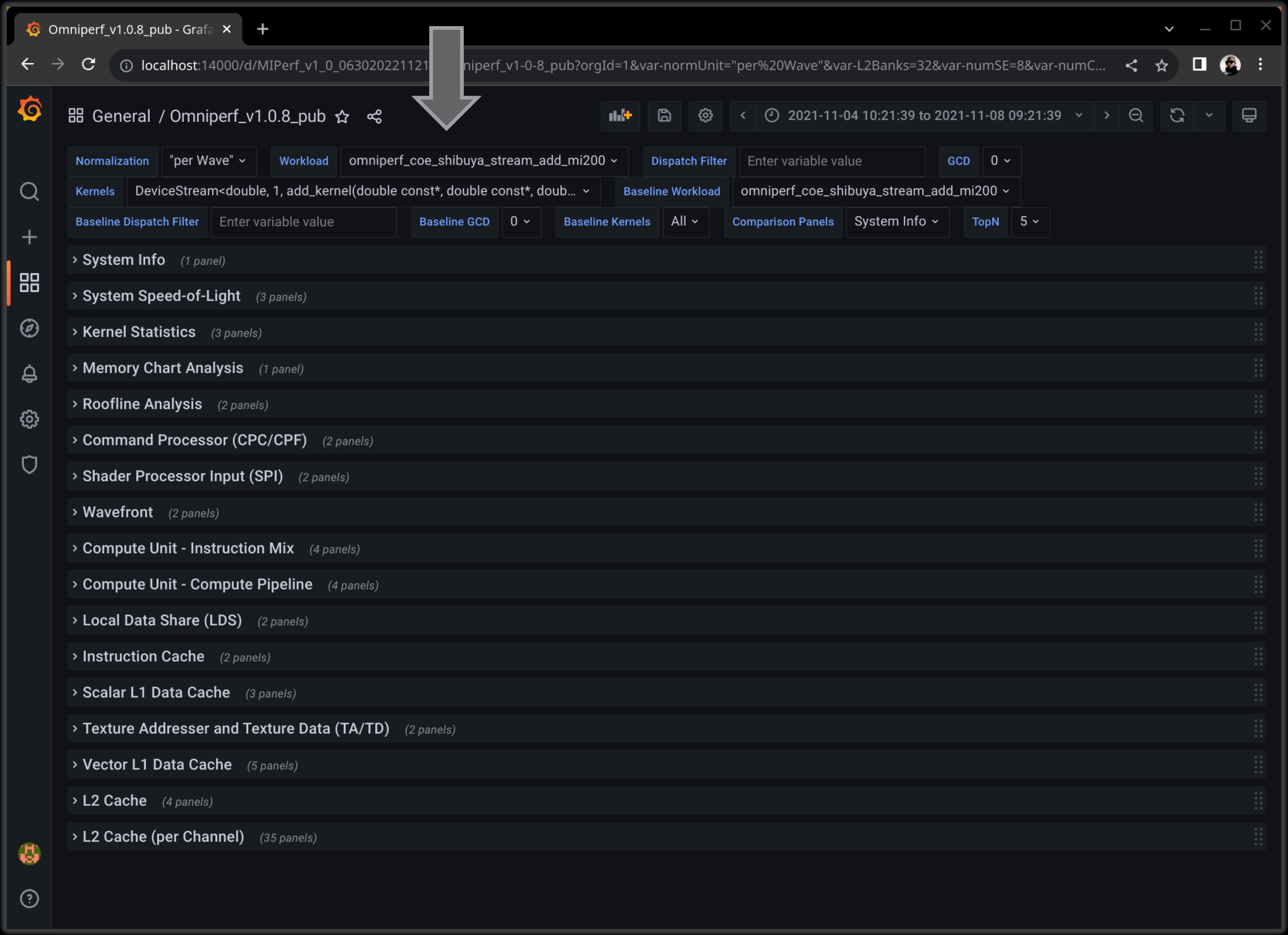


```
$ omniperf database --import -H 10.7.80.15 --username temp --team coe --workload workloads/coral_gemm_fp32/mi200/

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

Pulling data from  /home/jakub/Desktop/workloads/coral_gemm_fp32/mi200
The directory exists
Found sysinfo file
Password:
Password recieved
-- Conversion & Upload in Progress --
  0%|                                                                                           | 0/11 [00:00<?, ?it/s]
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/SQ_INST_LEVEL_LDS.csv
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/SQ_INST_LEVEL_SMEM.csv
 36%|█████████████████████████████████                                                          | 4/11 [00:00<00:00, 31.95it/s]
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/roofline.csv
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/SQ_IFETCH_LEVEL.csv
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/timestamps.csv
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/sysinfo.csv
 73%|██████████████████████████████████████████████████████████████████                         | 8/11 [00:00<00:00, 22.25it/s]
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/pmc_perf.csv
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/SQ_LEVEL_WAVES.csv
/home/jakub/Desktop/workloads/coral_gemm_fp32/mi200/SQ_INST_LEVEL_VMEM.csv
100%|███████████████████████████████████████████████████████████████████████████████████████████| 11/11 [00:00<00:00, 17.92it/s]
9 collections added.
Workload name uploaded
-- Complete! --
$
```

GRAFANA™

SHIBUYA STREAM

- go to Grafana™
- select workload (shibuya add)
- select kernel (add_kernel)
- explore tabs

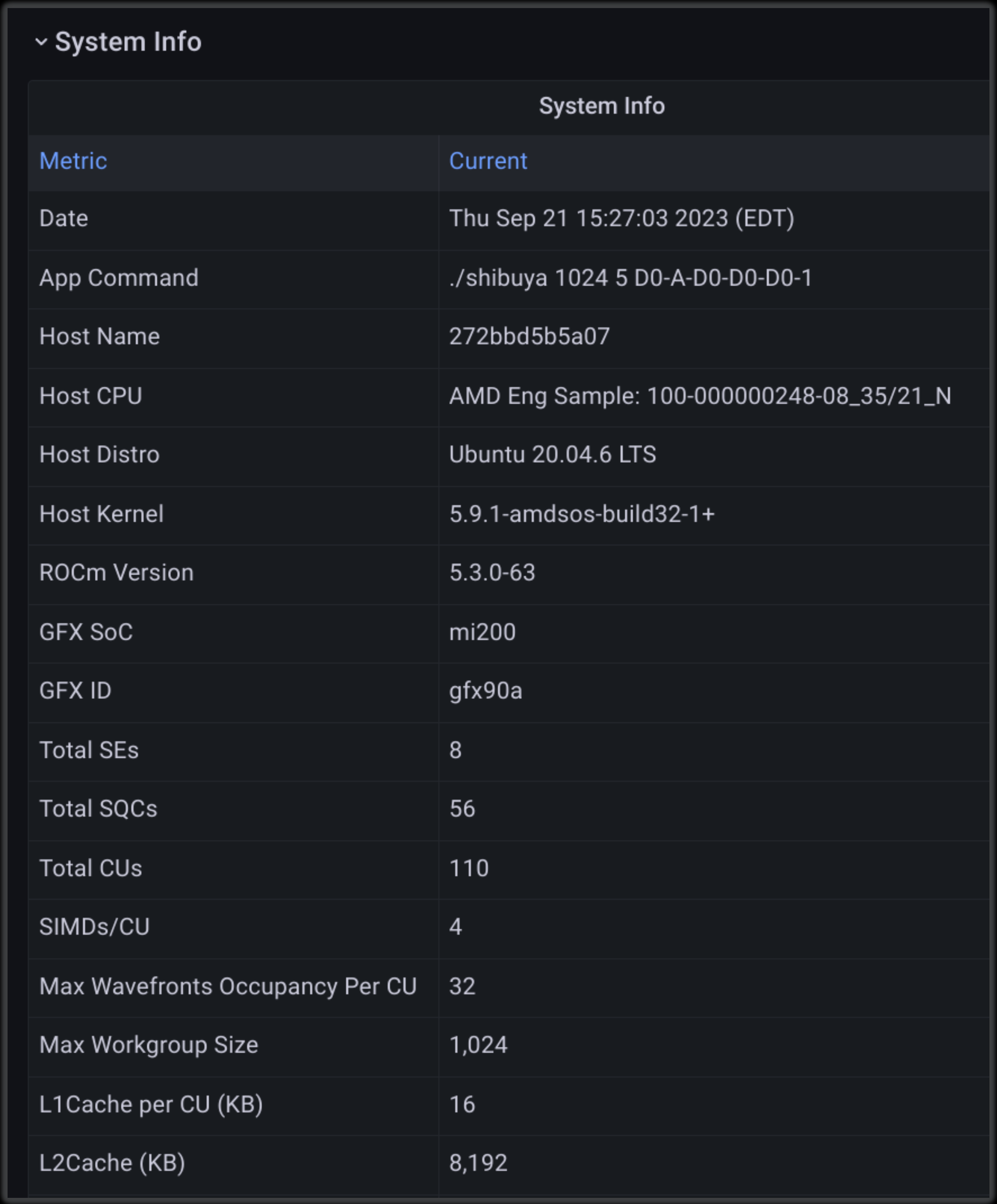

Omniperf_v1.0.8_pub - Grafa
localhost:14000/d/MIPerf_v1_0_063020221121 niperf_v1-0-8_pub?orgId=1&var-normUnit="per%20Wave"&var-L2Banks=32&var-numSE=8&var-numC...
General / Omniperf_v1.0.8_pub
2021-11-04 10:21:39 to 2021-11-08 09:21:39

Normalization "per Wave"
Workload omniperf_coe_shibuya_stream_add_mi200
Dispatch Filter Enter variable value
GCD 0
Kernels DeviceStream<double, 1, add_kernel(double const*, double const*, doub...
Baseline Workload omniperf_coe_shibuya_stream_add_mi200
Baseline Dispatch Filter Enter variable value
Baseline GCD 0
Baseline Kernels All
Comparison Panels System Info
TopN 5

- System Info (1 panel)
- System Speed-of-Light (3 panels)
- Kernel Statistics (3 panels)
- Memory Chart Analysis (1 panel)
- Roofline Analysis (2 panels)
- Command Processor (CPC/CPF) (2 panels)
- Shader Processor Input (SPI) (2 panels)
- Wavefront (2 panels)
- Compute Unit - Instruction Mix (4 panels)
- Compute Unit - Compute Pipeline (4 panels)
- Local Data Share (LDS) (2 panels)
- Instruction Cache (2 panels)
- Scalar L1 Data Cache (3 panels)
- Texture Addresser and Texture Data (TA/TD) (2 panels)
- Vector L1 Data Cache (5 panels)
- L2 Cache (4 panels)
- L2 Cache (per Channel) (35 panels)

GRAFANA™

SHIBUYA STREAM



⌄ System Info

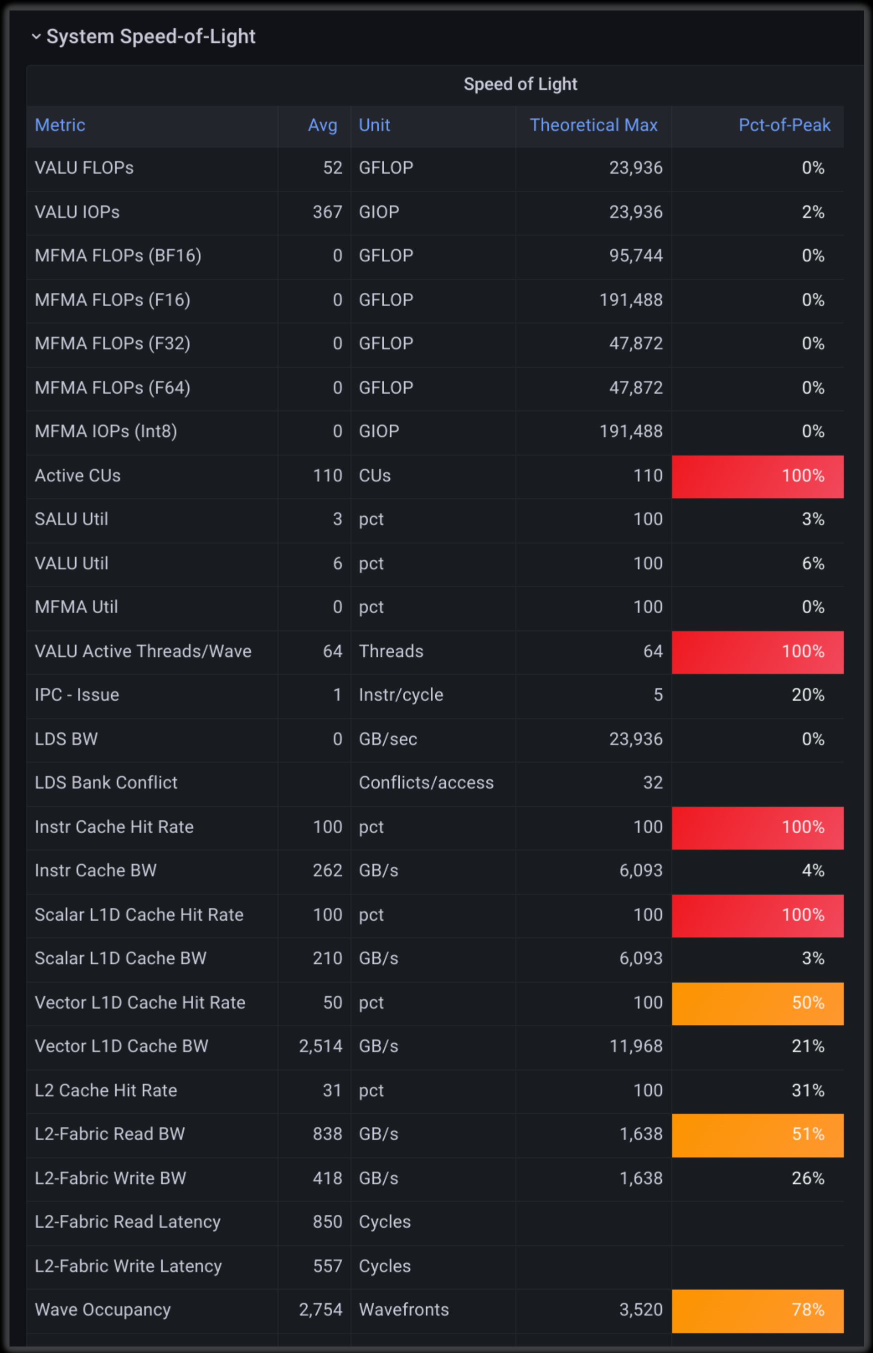
System Info

Metric	Current
Date	Thu Sep 21 15:27:03 2023 (EDT)
App Command	./shibuya 1024 5 D0-A-D0-D0-D0-1
Host Name	272bbd5b5a07
Host CPU	AMD Eng Sample: 100-000000248-08_35/21_N
Host Distro	Ubuntu 20.04.6 LTS
Host Kernel	5.9.1-amdsos-build32-1+
ROCm Version	5.3.0-63
GFX SoC	mi200
GFX ID	gfx90a
Total SEs	8
Total SQCs	56
Total CUs	110
SIMDs/CU	4
Max Wavefronts Occupancy Per CU	32
Max Workgroup Size	1,024
L1Cache per CU (KB)	16
L2Cache (KB)	8,192

- time & date
- command
- OS version
- ROCm version
- hardware specs
  - no. SEs
  - no. CUs
  - no. SIMDs/CU
  - cache sizes
  - etc.

GRAFANA™

SHIBUYA STREAM



⌄ System Speed-of-Light

Speed of Light

Metric	Avg	Unit	Theoretical Max	Pct-of-Peak
VALU FLOPs	52	GFLOP	23,936	0%
VALU IOPs	367	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	110	CUs	110	100%
SALU Util	3	pct	100	3%
VALU Util	6	pct	100	6%
MFMA Util	0	pct	100	0%
VALU Active Threads/Wave	64	Threads	64	100%
IPC - Issue	1	Instr/cycle	5	20%
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	262	GB/s	6,093	4%
Scalar L1D Cache Hit Rate	100	pct	100	100%
Scalar L1D Cache BW	210	GB/s	6,093	3%
Vector L1D Cache Hit Rate	50	pct	100	50%
Vector L1D Cache BW	2,514	GB/s	11,968	21%
L2 Cache Hit Rate	31	pct	100	31%
L2-Fabric Read BW	838	GB/s	1,638	51%
L2-Fabric Write BW	418	GB/s	1,638	26%
L2-Fabric Read Latency	850	Cycles		
L2-Fabric Write Latency	557	Cycles		
Wave Occupancy	2,754	Wavefronts	3,520	78%

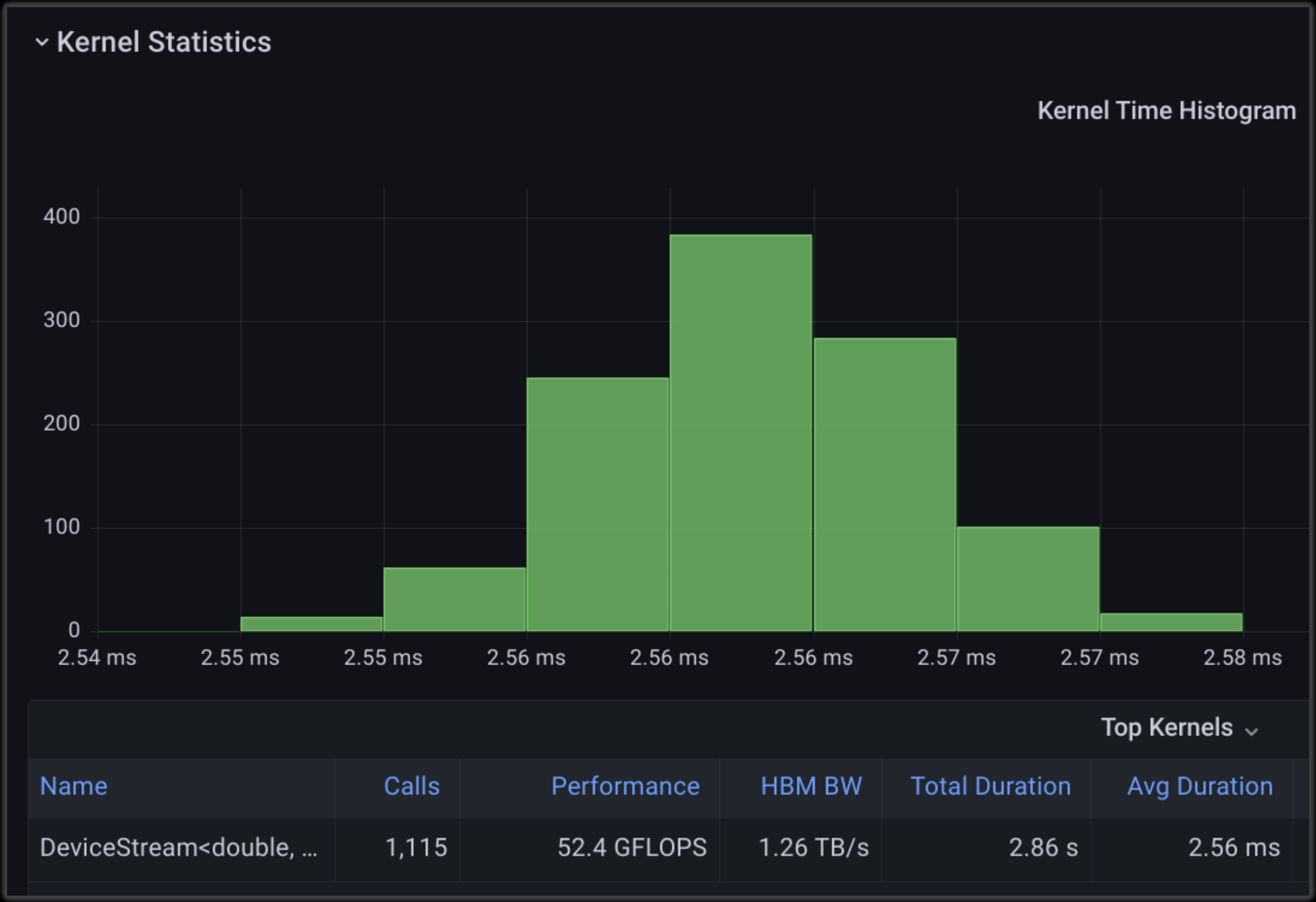
hardware utilization

- CUs
- threads/wave
- cache hit rates
- etc.

GRAFANA™

SHIBUYA STREAM

- kernel time histogram
- Gaussian distribution
- 2.55 – 2.58 ms
- number of calls
- GFLOPS
- bandwidth
- total duration
- average duration

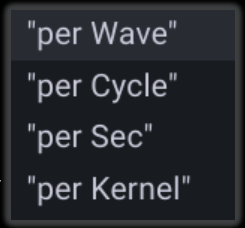


Name	Calls	Performance	HBM BW	Total Duration	Avg Duration
DeviceStream<double, ...	1,115	52.4 GFLOPS	1.26 TB/s	2.86 s	2.56 ms

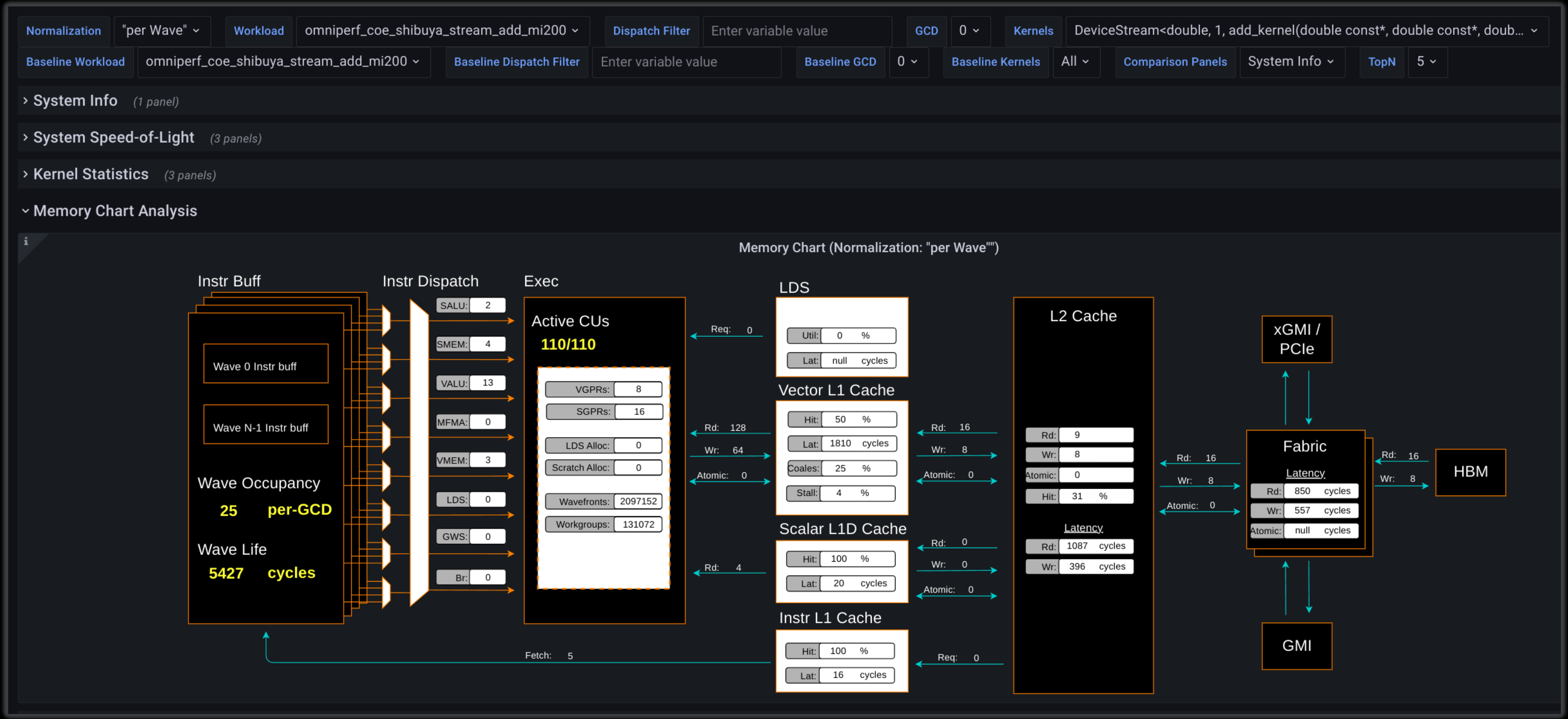
GRAFANA™

SHIBUYA STREAM

- traffic across the memory system
- normalized per wave / cycle / sec / kernel



"per Wave"
"per Cycle"
"per Sec"
"per Kernel"



Normalization "per Wave" | Workload omniperf_coe_shibuya_stream_add_mi200 | Dispatch Filter Enter variable value | GCD 0 | Kernels DeviceStream<double, 1, add_kernel(double const*, double const*, doub…

Baseline Workload omniperf_coe_shibuya_stream_add_mi200 | Baseline Dispatch Filter Enter variable value | Baseline GCD 0 | Baseline Kernels All | Comparison Panels System Info | TopN 5

› System Info *(1 panel)*

› System Speed-of-Light *(3 panels)*

› Kernel Statistics *(3 panels)*

⌄ Memory Chart Analysis

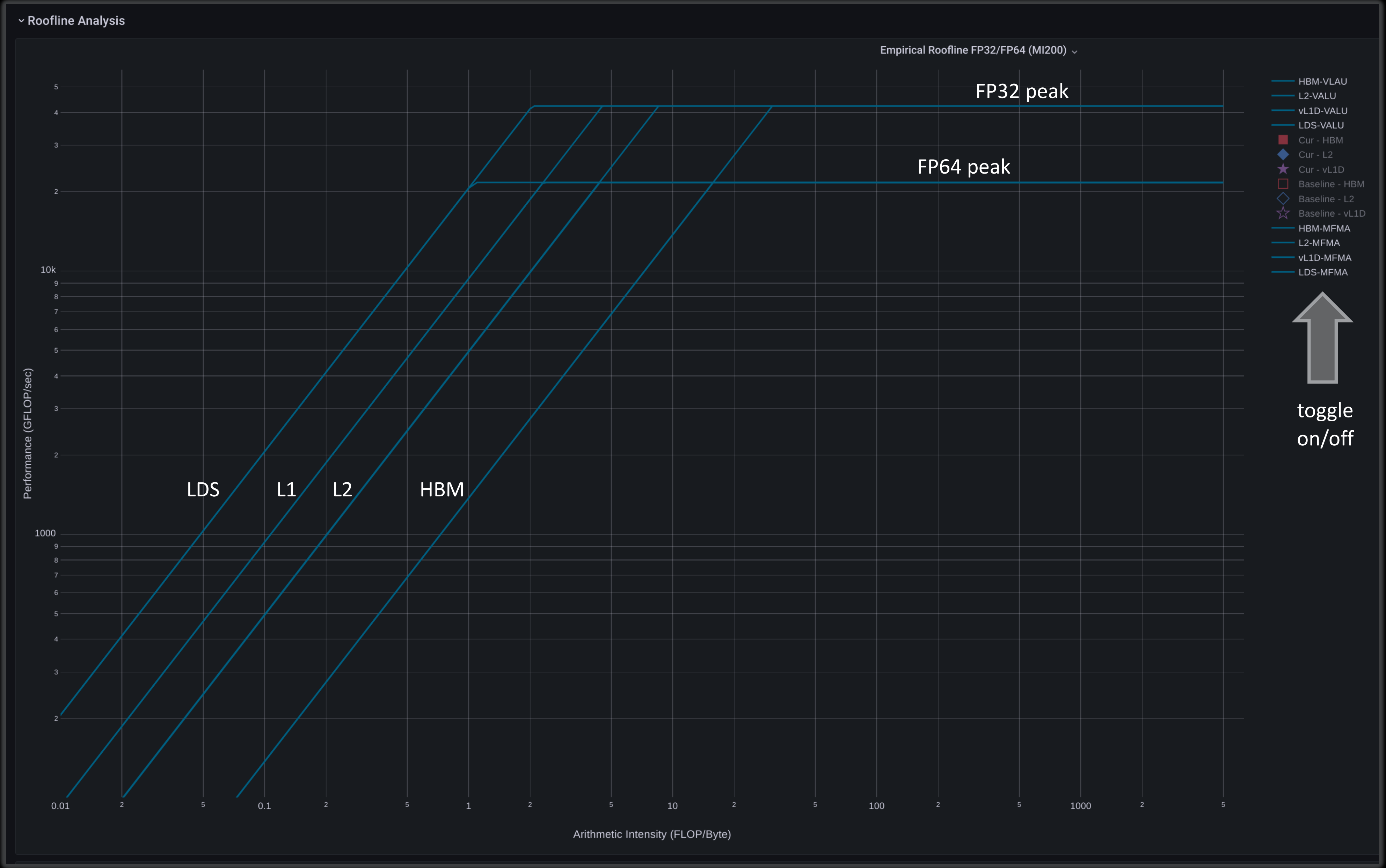
Memory Chart (Normalization: "per Wave"")

AMD
together we advance_

GRAFANA™

ROOFLINE

performance (GFLOPS)



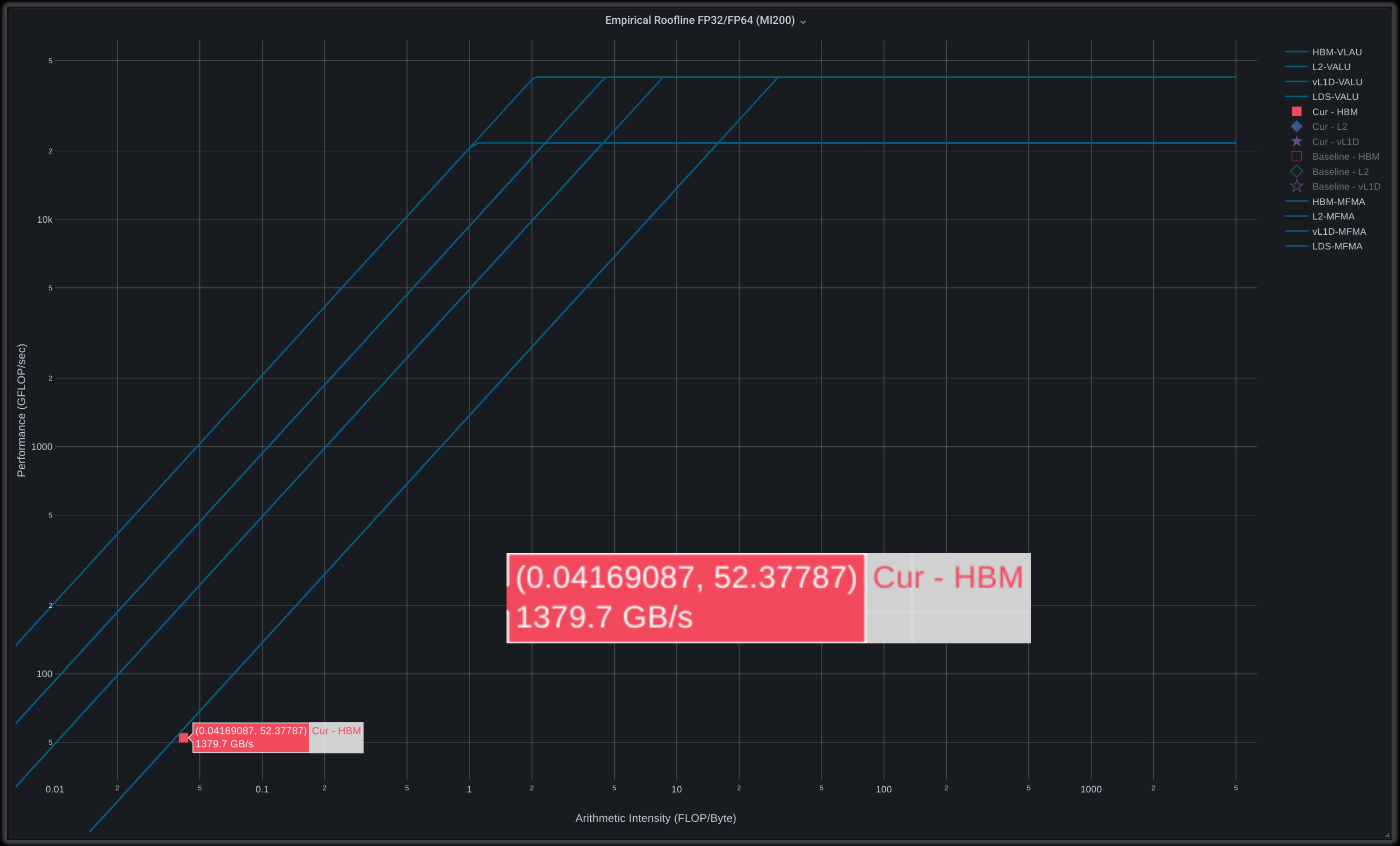
arithmetic intensity (FLOPS/byte)

- performance vs arithmetic intensity
- check if kernels are BW or FP limited

ROOFLINE

SHIBUYA STREAM

- add in HBM
- c[i]=a[i]+b[i]
- 24 bytes/FLOP
- 0.04 FLOPS/byte
- HBM-bound
- 52 GFLOPS

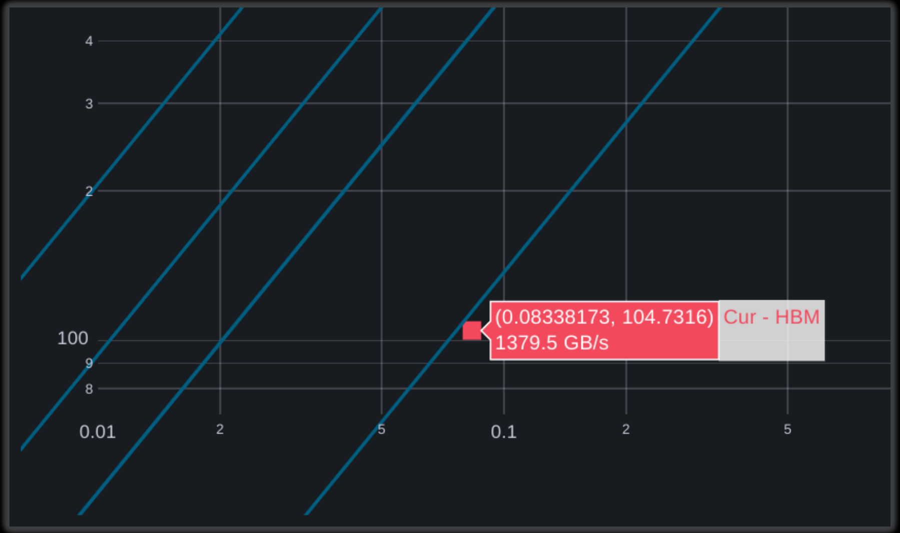


AMD
together we advance_

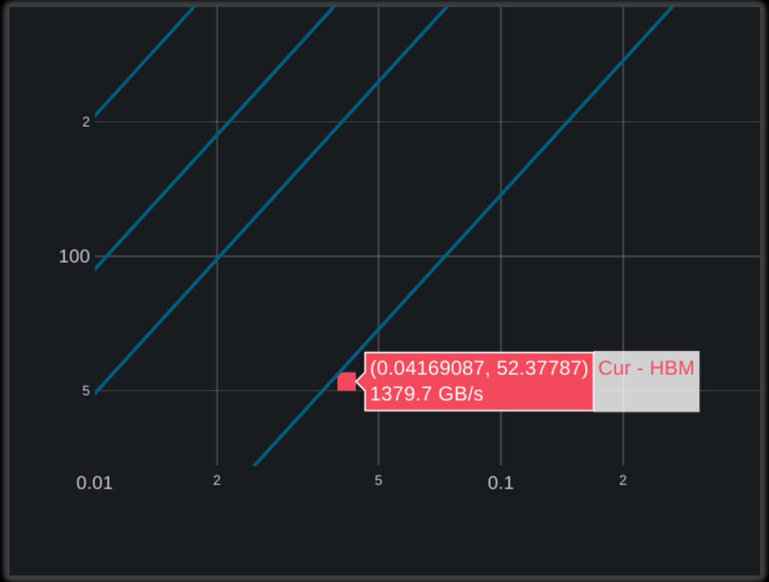
ROOFLINE

SHIBUYA STREAM

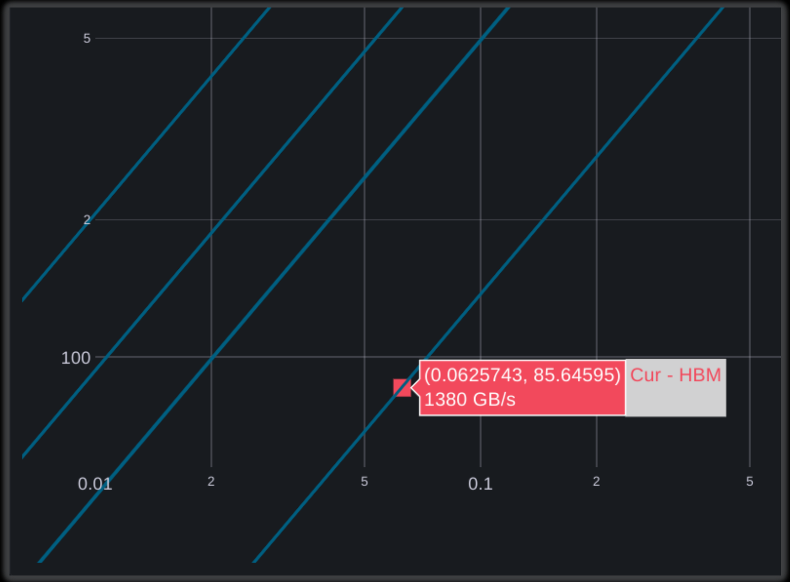
- triad
- c[i]=alpha*a[i]+b[i]
- 0.08 FLOPS/byte
- 104 GFLOPS



- add
- c[i]=a[i]+b[i]
- 0.04 FLOPS/byte
- 52 GFLOPS



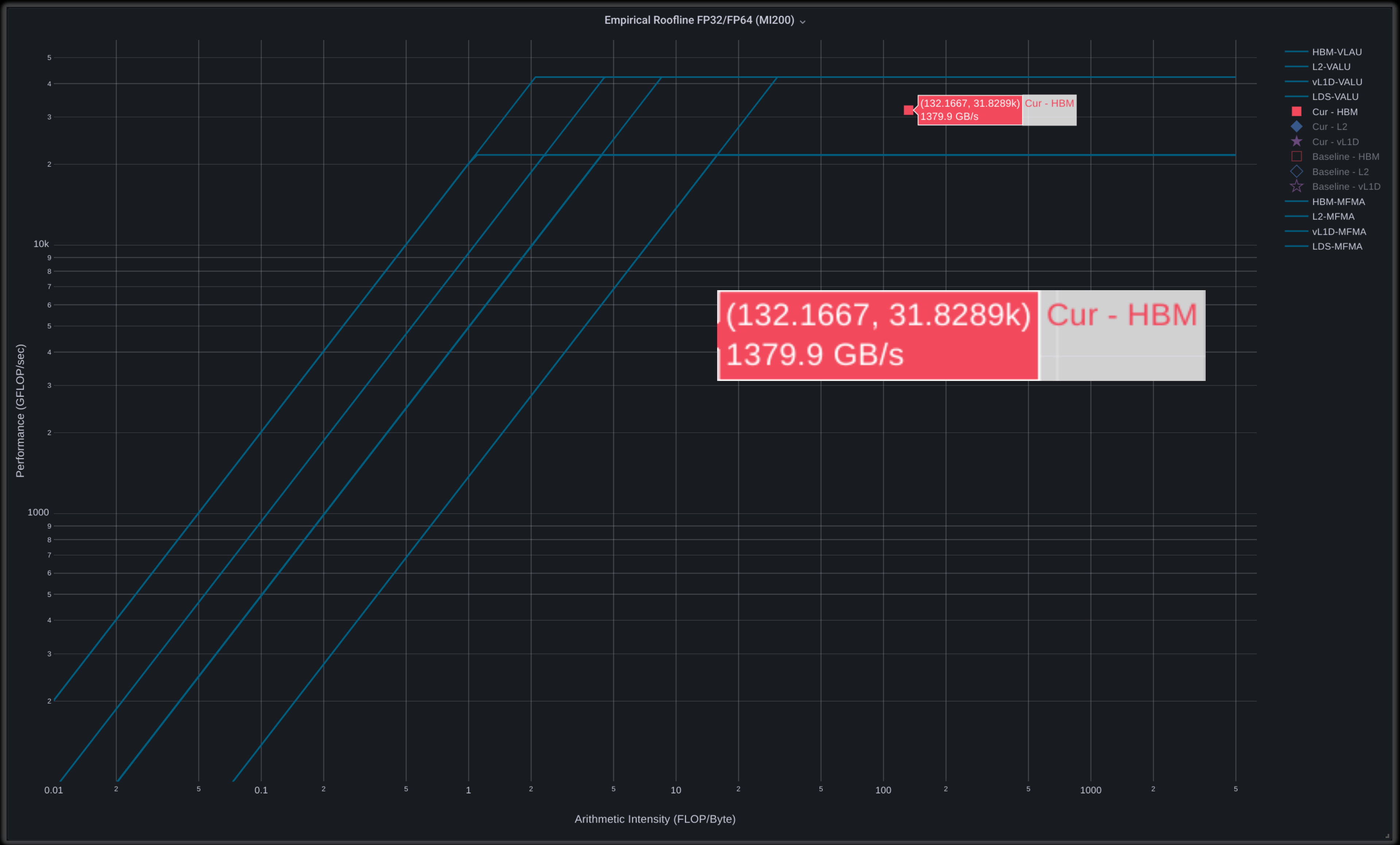
- multiply
- b[i]=alpha*a[i]
- 0.06 FLOPS/byte
- 85 GFLOPS



ROOFLINE

CORAL GEMM

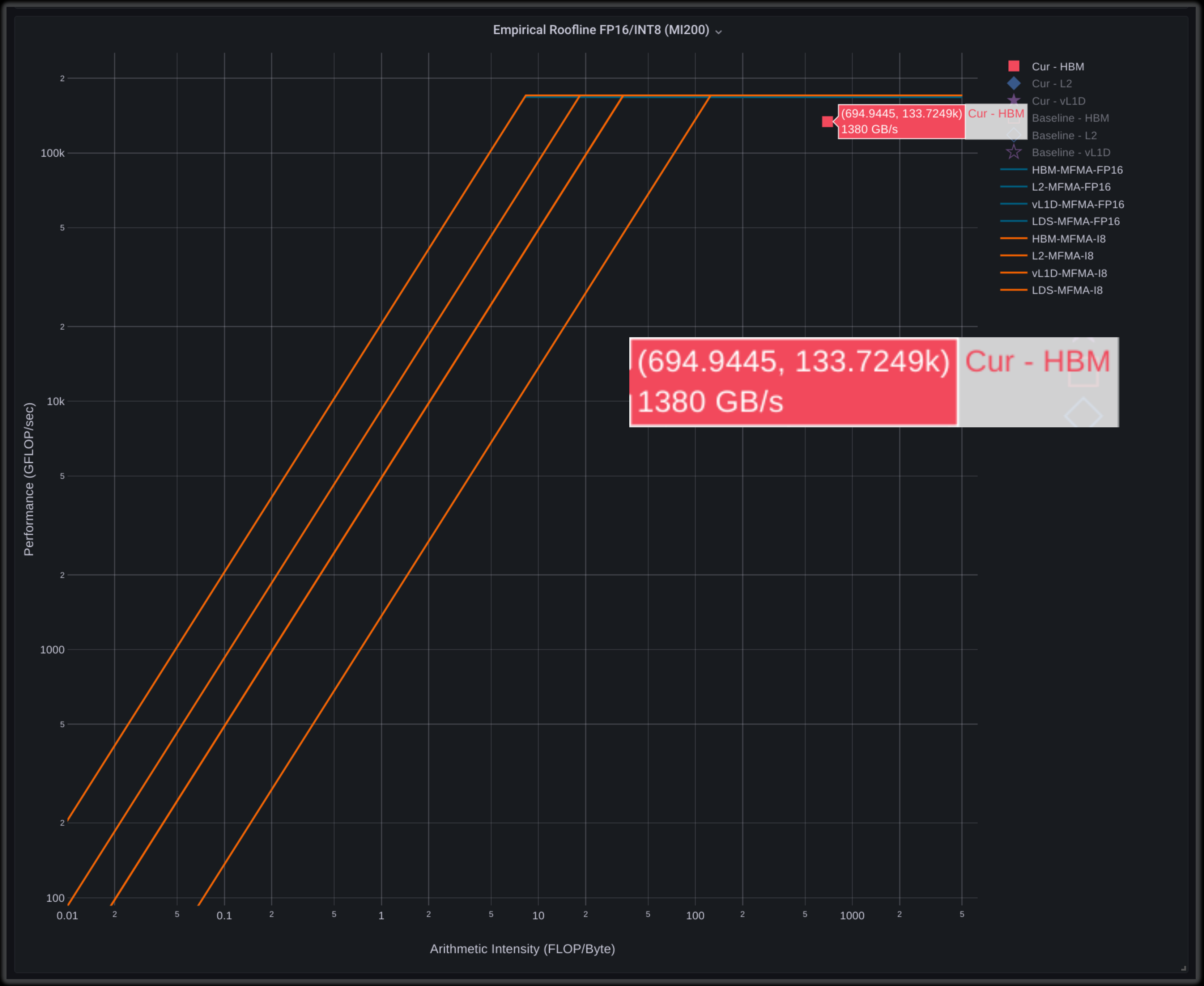
- GEMM in FP32
- 132 FLOPS/byte
- ~32 TFLOPS
- compute-bound



ROOFLINE

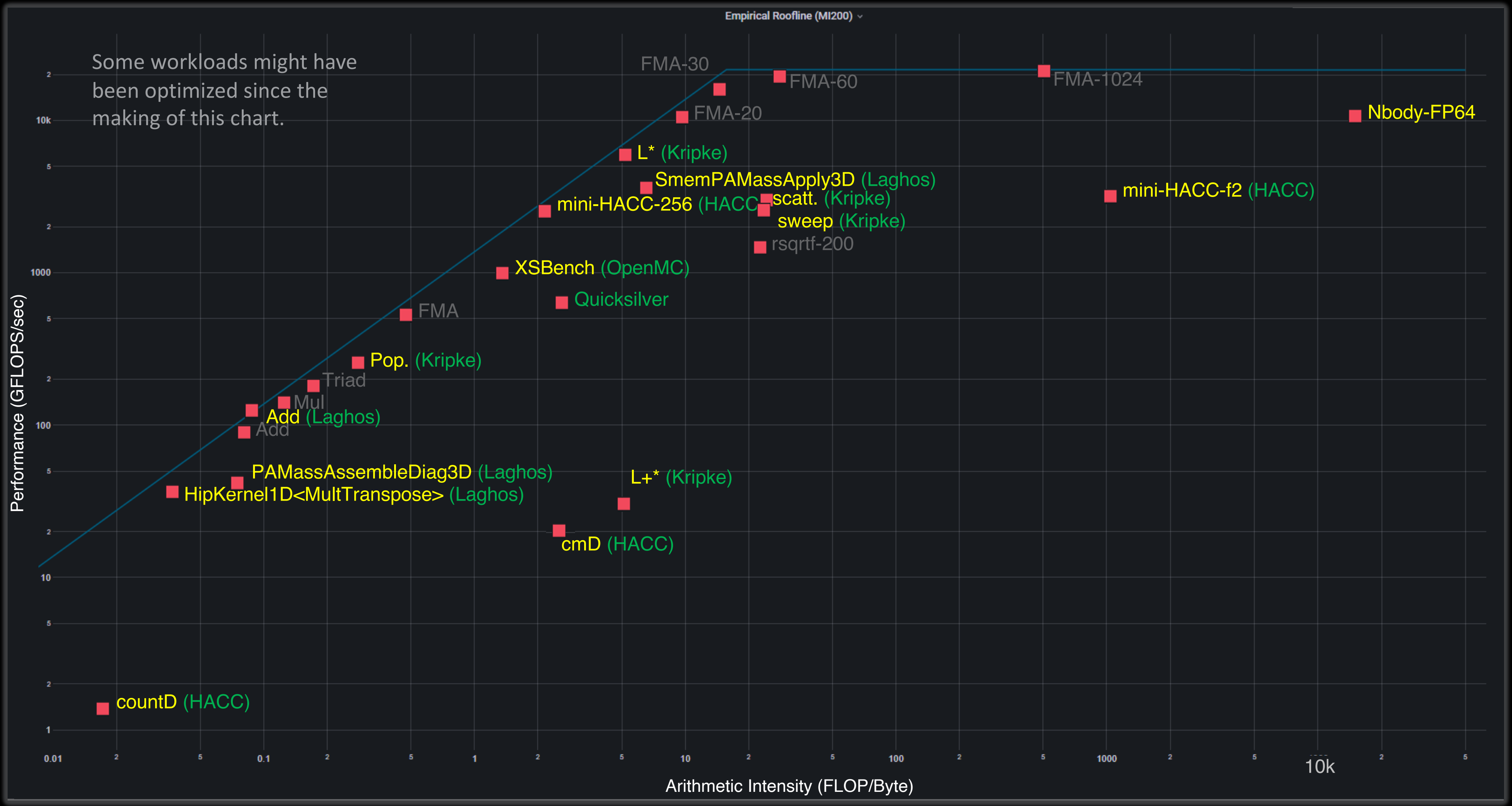
CORAL GEMM

- GEMM in FP16
- 695 FLOPS/byte
- ~134 TFLOPS
- compute-bound



AMD
together we advance_

ROOFLINE ANALYSIS



Some workloads might have been optimized since the making of this chart.

synthetic workload

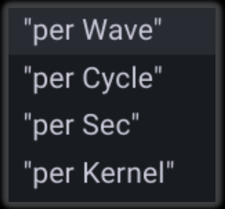
proxy app

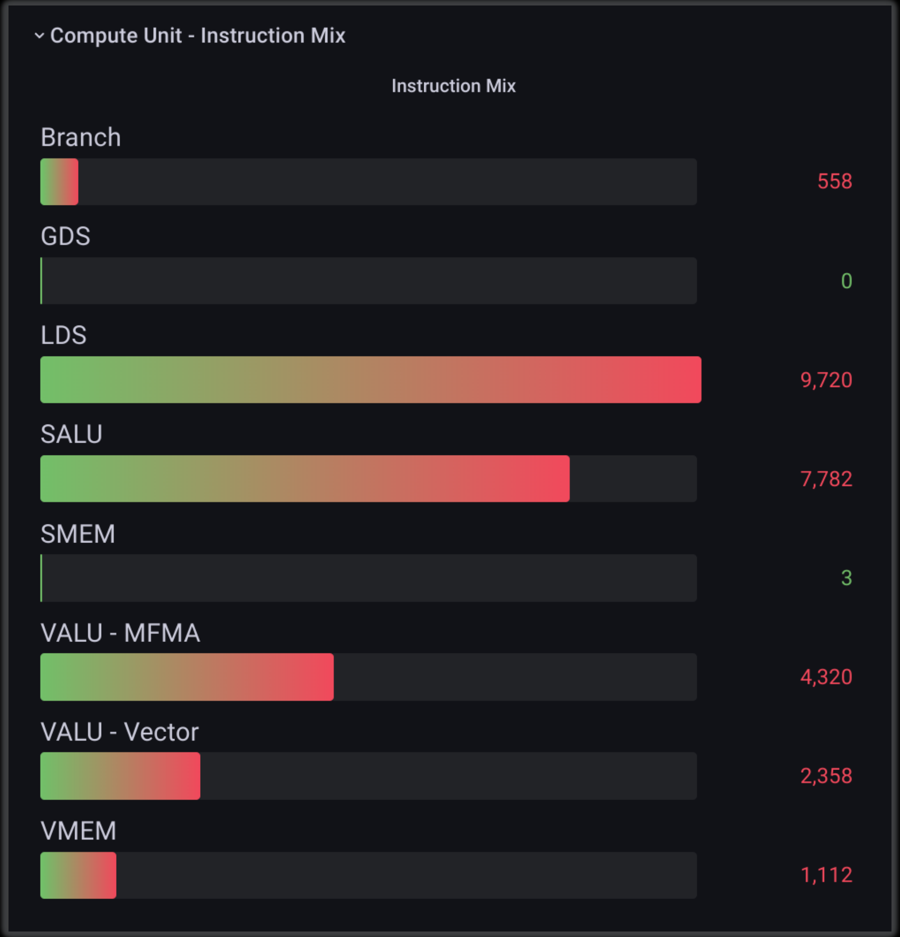
full app

courtesy Noah Wolfe

INSTRUCTION MIX

CORAL GEMM FP32

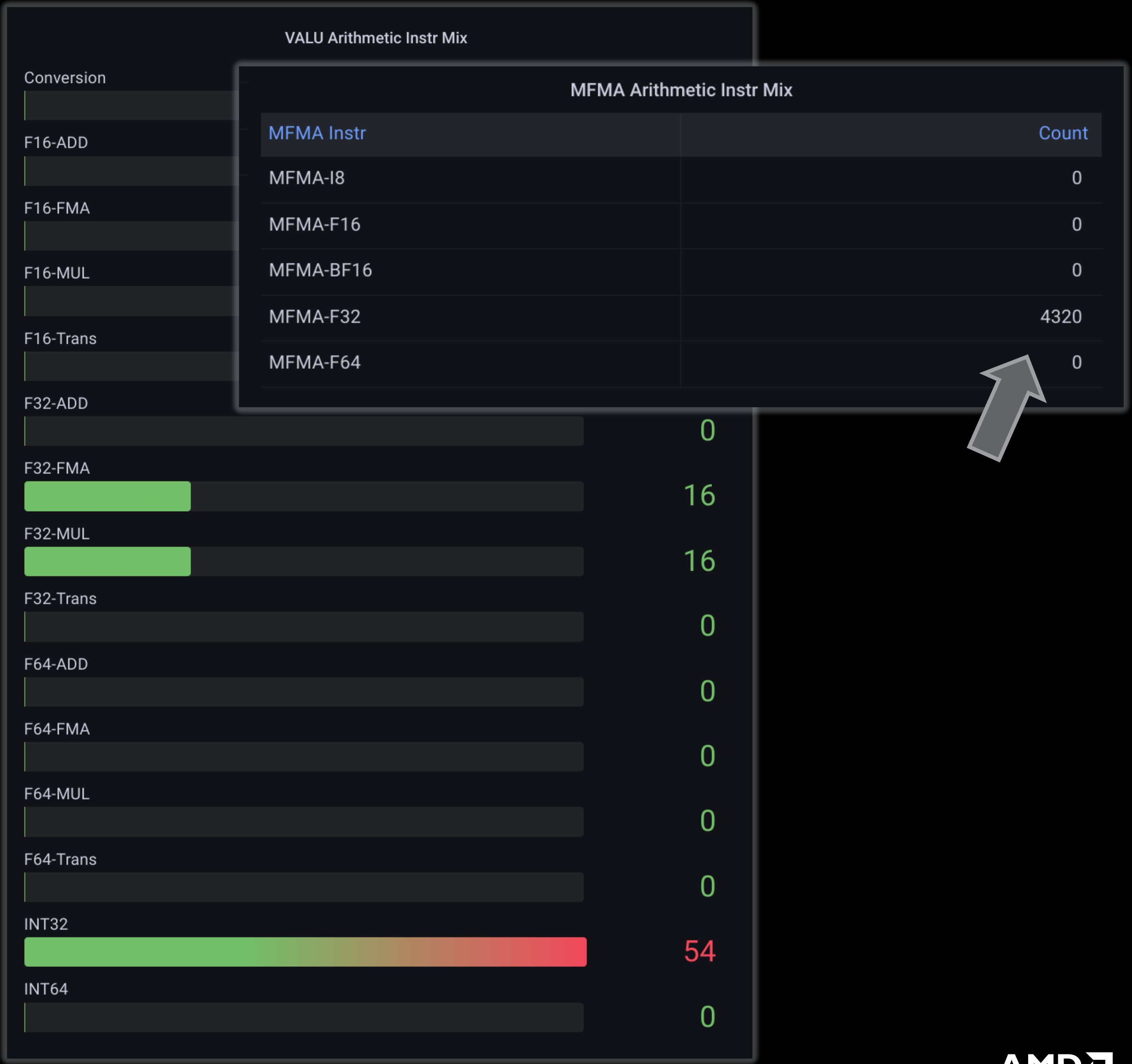




Compute Unit - Instruction Mix

Instruction Mix

Branch	558
GDS	0
LDS	9,720
SALU	7,782
SMEM	3
VALU - MFMA	4,320
VALU - Vector	2,358
VMEM	1,112



VALU Arithmetic Instr Mix

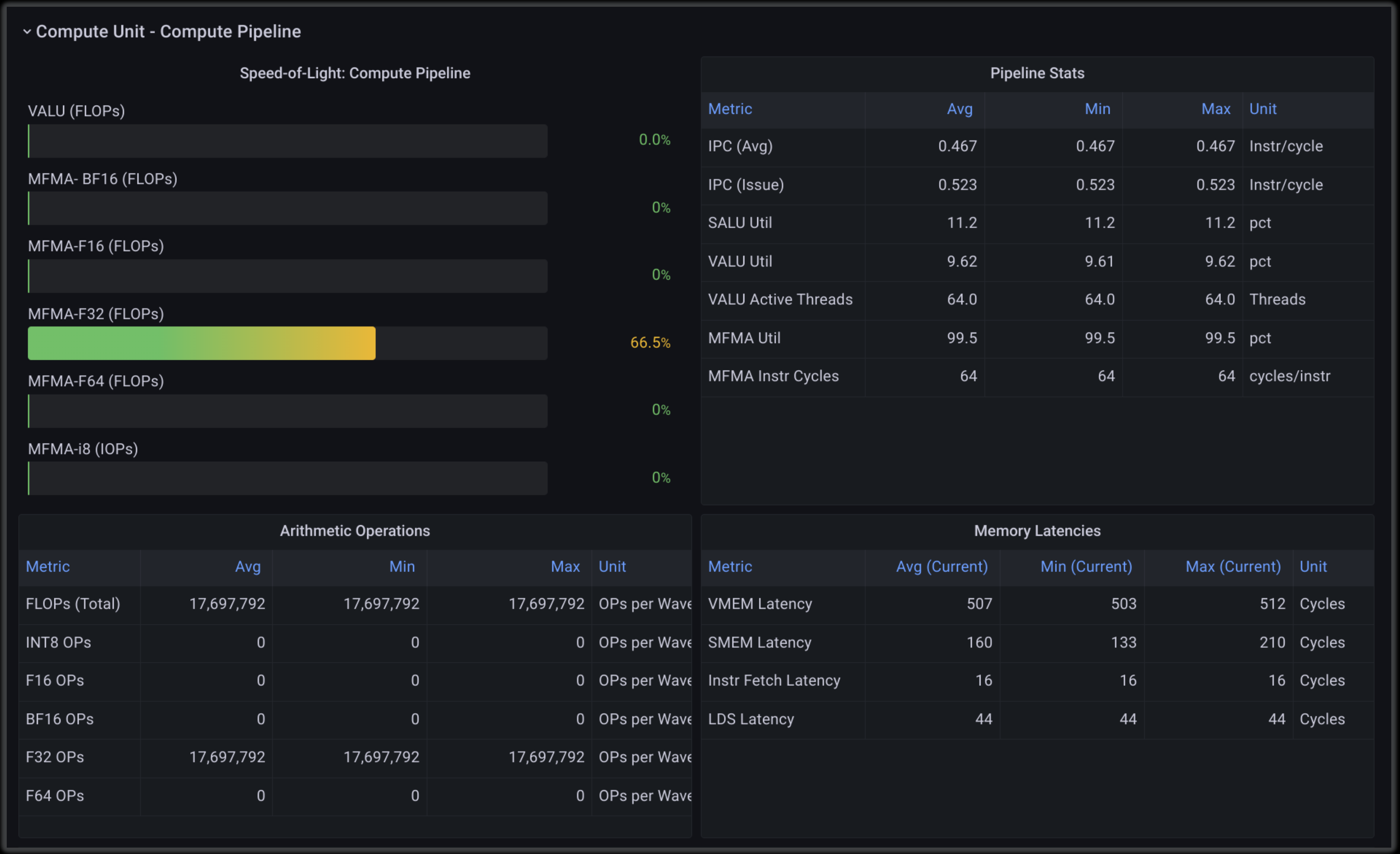
Conversion	
F16-ADD	
F16-FMA	
F16-MUL	
F16-Trans	
F32-ADD	0
F32-FMA	16
F32-MUL	16
F32-Trans	0
F64-ADD	0
F64-FMA	0
F64-MUL	0
F64-Trans	0
INT32	54
INT64	0

MFMA Arithmetic Instr Mix

MFMA Instr	Count
MFMA-I8	0
MFMA-F16	0
MFMA-BF16	0
MFMA-F32	4320
MFMA-F64	0

COMPUTE PUPELINE

CORAL GEMM FP32



Compute Unit - Compute Pipeline

Speed-of-Light: Compute Pipeline

VALU (FLOPs)	0.0%
MFMA- BF16 (FLOPs)	0%
MFMA-F16 (FLOPs)	0%
MFMA-F32 (FLOPs)	66.5%
MFMA-F64 (FLOPs)	0%
MFMA-i8 (IOPs)	0%

Pipeline Stats

Metric	Avg	Min	Max	Unit
IPC (Avg)	0.467	0.467	0.467	Instr/cycle
IPC (Issue)	0.523	0.523	0.523	Instr/cycle
SALU Util	11.2	11.2	11.2	pct
VALU Util	9.62	9.61	9.62	pct
VALU Active Threads	64.0	64.0	64.0	Threads
MFMA Util	99.5	99.5	99.5	pct
MFMA Instr Cycles	64	64	64	cycles/instr

Arithmetic Operations

Metric	Avg	Min	Max	Unit
FLOPs (Total)	17,697,792	17,697,792	17,697,792	OPs per Wave
INT8 OPs	0	0	0	OPs per Wave
F16 OPs	0	0	0	OPs per Wave
BF16 OPs	0	0	0	OPs per Wave
F32 OPs	17,697,792	17,697,792	17,697,792	OPs per Wave
F64 OPs	0	0	0	OPs per Wave

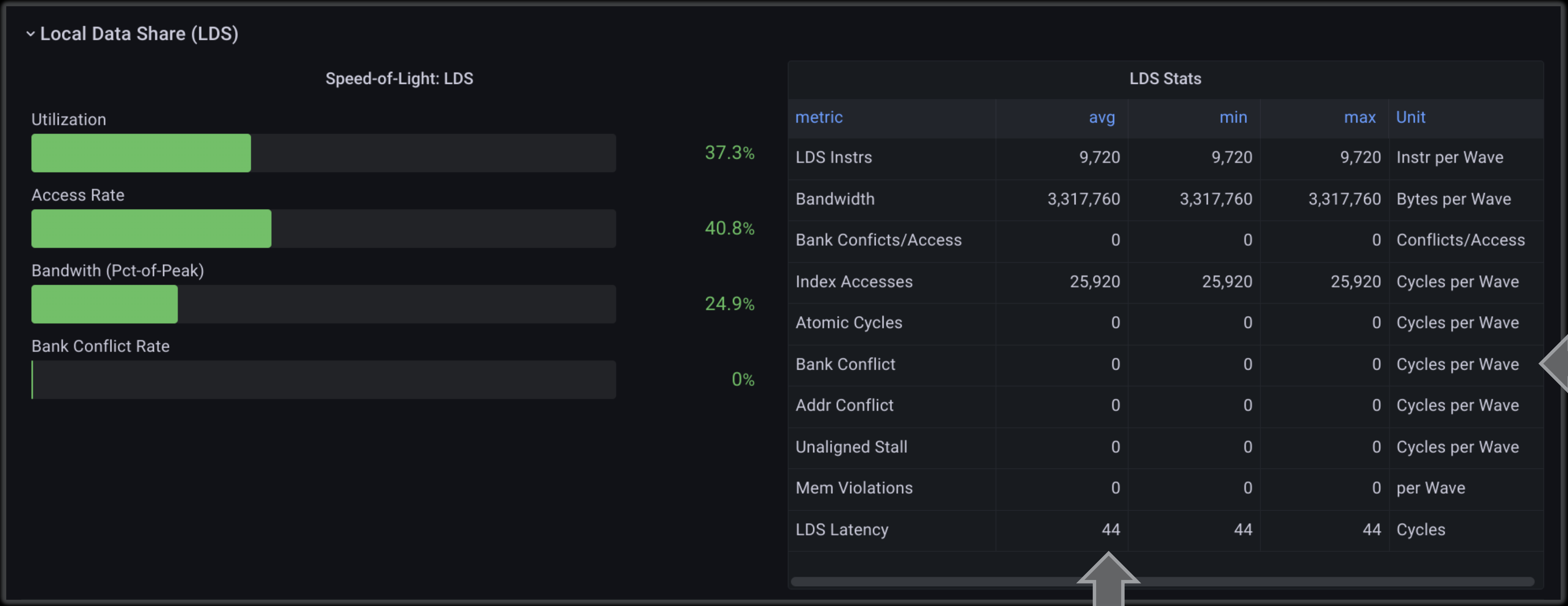
Memory Latencies

Metric	Avg (Current)	Min (Current)	Max (Current)	Unit
VMEM Latency	507	503	512	Cycles
SMEM Latency	160	133	210	Cycles
Instr Fetch Latency	16	16	16	Cycles
LDS Latency	44	44	44	Cycles

- pipeline utilization
- memory latencies
- etc.

LOCAL DATA SHARE

CORAL GEMM FP32



⌄ Local Data Share (LDS)

Speed-of-Light: LDS

Utilization 37.3%

Access Rate 40.8%

Bandwith (Pct-of-Peak) 24.9%

Bank Conflict Rate 0%

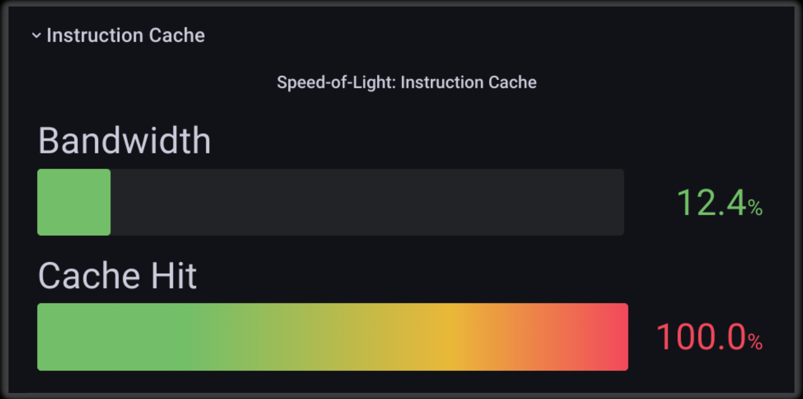
LDS Stats

metric	avg	min	max	Unit
LDS Instrs	9,720	9,720	9,720	Instr per Wave
Bandwidth	3,317,760	3,317,760	3,317,760	Bytes per Wave
Bank Conficts/Access	0	0	0	Conflicts/Access
Index Accesses	25,920	25,920	25,920	Cycles per Wave
Atomic Cycles	0	0	0	Cycles per Wave
Bank Conflict	0	0	0	Cycles per Wave
Addr Conflict	0	0	0	Cycles per Wave
Unaligned Stall	0	0	0	Cycles per Wave
Mem Violations	0	0	0	per Wave
LDS Latency	44	44	44	Cycles

- LDS bank conflicts
- LDS latency
- etc.

CACHES

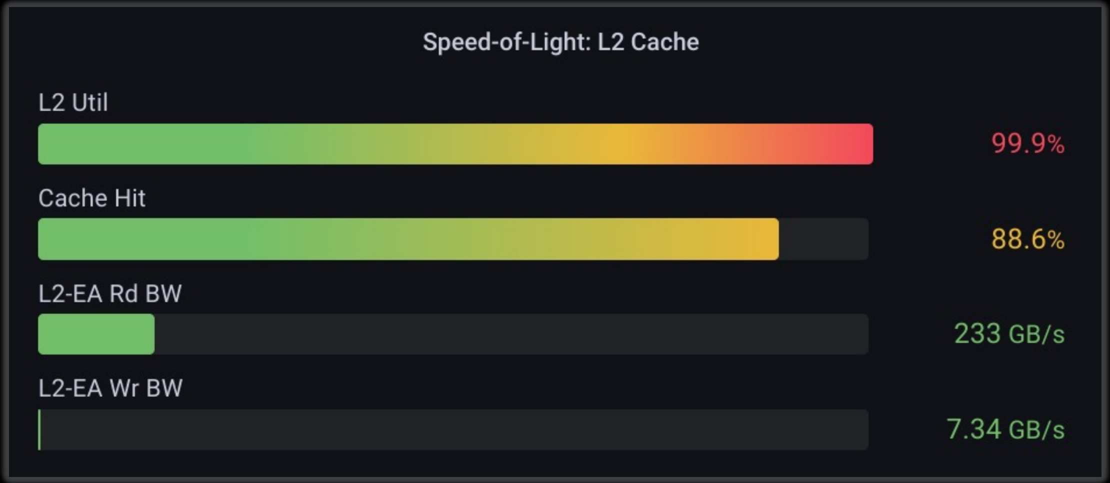
CORAL GEMM FP32



⌄ Instruction Cache

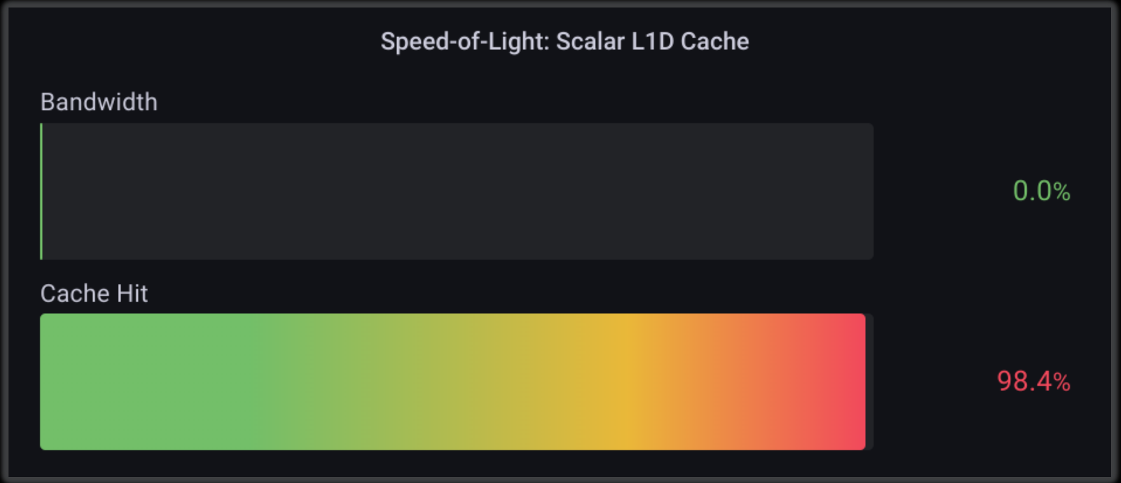
Speed-of-Light: Instruction Cache

Bandwidth	12.4%
Cache Hit	100.0%



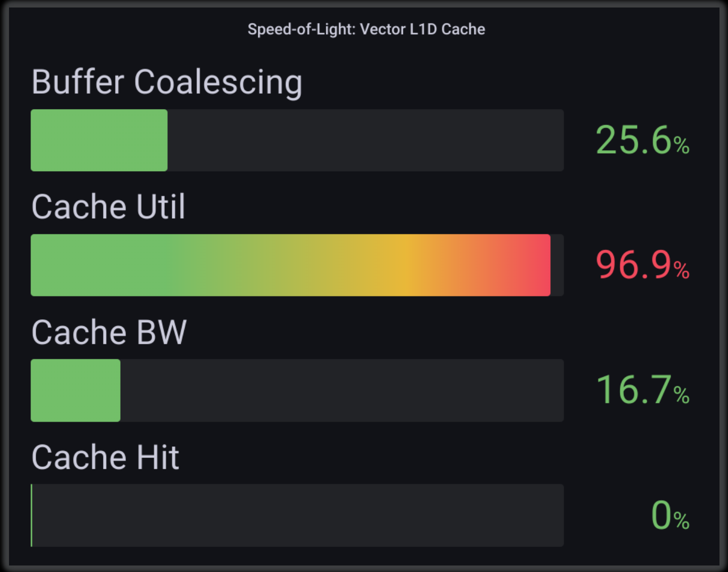
Speed-of-Light: L2 Cache

L2 Util	99.9%
Cache Hit	88.6%
L2-EA Rd BW	233 GB/s
L2-EA Wr BW	7.34 GB/s



Speed-of-Light: Scalar L1D Cache

Bandwidth	0.0%
Cache Hit	98.4%



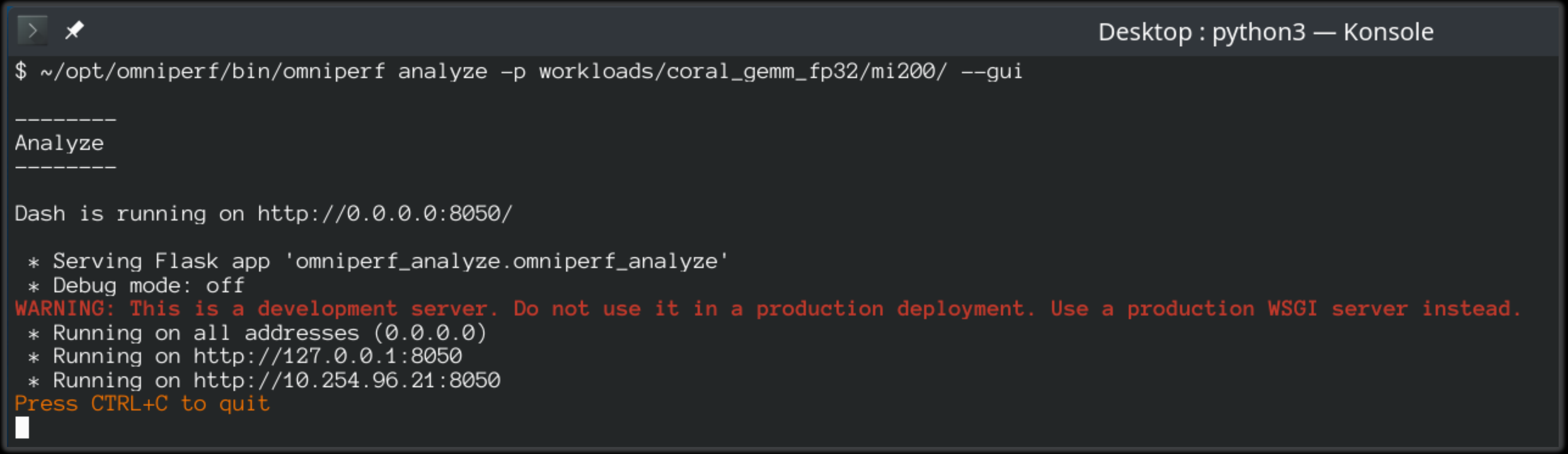
Speed-of-Light: Vector L1D Cache

Buffer Coalescing	25.6%
Cache Util	96.9%
Cache BW	16.7%
Cache Hit	0%

- instruction cache
- scalar L1 data
- vector L1 data
- L2

STAND-ALONE GUI

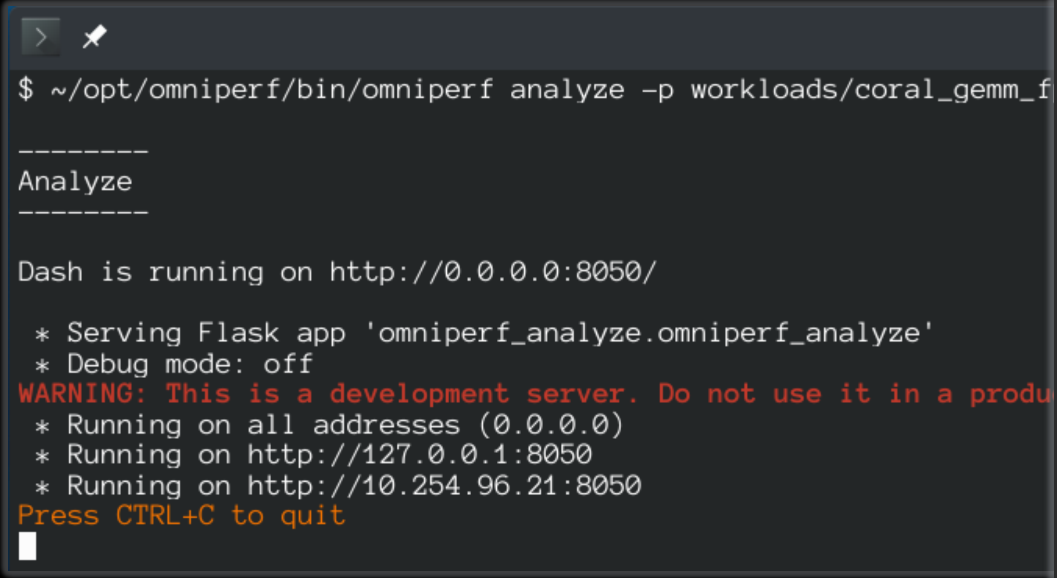
```
omniperf analyze -p workloads/coral_gemm_fp32/mi200/ --gui
```



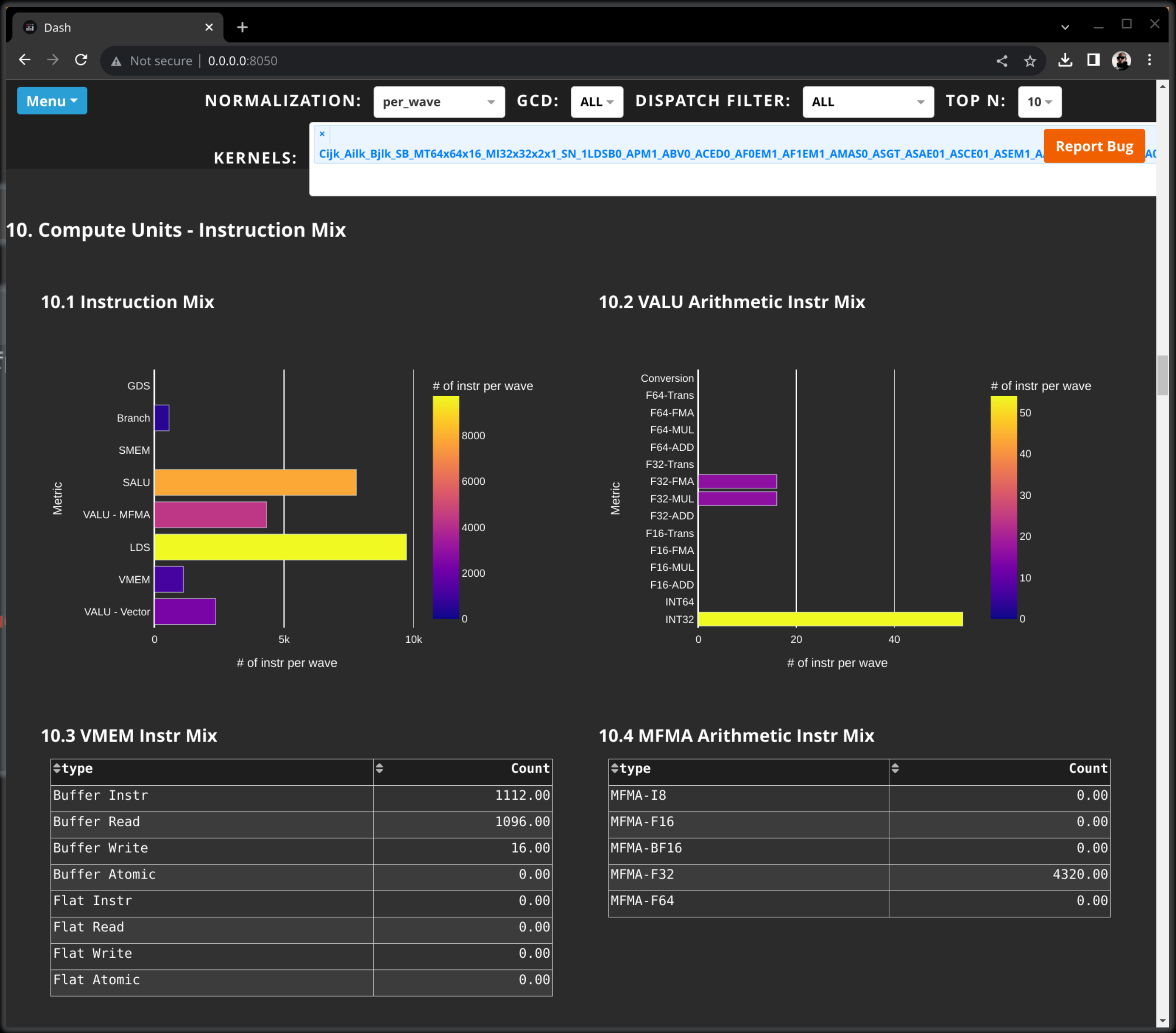
```
$ ~/opt/omniperf/bin/omniperf analyze -p workloads/coral_gemm_fp32/mi200/ --gui

--------
Analyze
--------

Dash is running on http://0.0.0.0:8050/

 * Serving Flask app 'omniperf_analyze.omniperf_analyze'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8050
 * Running on http://10.254.96.21:8050
Press CTRL+C to quit
```

STAND-ALONE GUI

```
omniperf analyze -p workloads/coral_gemm_fp32/mi200/ --gui
```



```
$ ~/opt/omniperf/bin/omniperf analyze -p workloads/coral_gemm_f

--------
Analyze
--------

Dash is running on http://0.0.0.0:8050/

 * Serving Flask app 'omniperf_analyze.omniperf_analyze'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a produ
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8050
 * Running on http://10.254.96.21:8050
Press CTRL+C to quit
```



NORMALIZATION: per_wave   GCD: ALL   DISPATCH FILTER: ALL   TOP N: 10

KERNELS: Cijk_Ailk_Bjlk_SB_MT64x64x16_MI32x32x2x1_SN_1LDSB0_APM1_ABV0_ACED0_AF0EM1_AF1EM1_AMAS0_ASGT_ASAE01_ASCE01_ASEM1_A

10. Compute Units - Instruction Mix

10.1 Instruction Mix

10.2 VALU Arithmetic Instr Mix

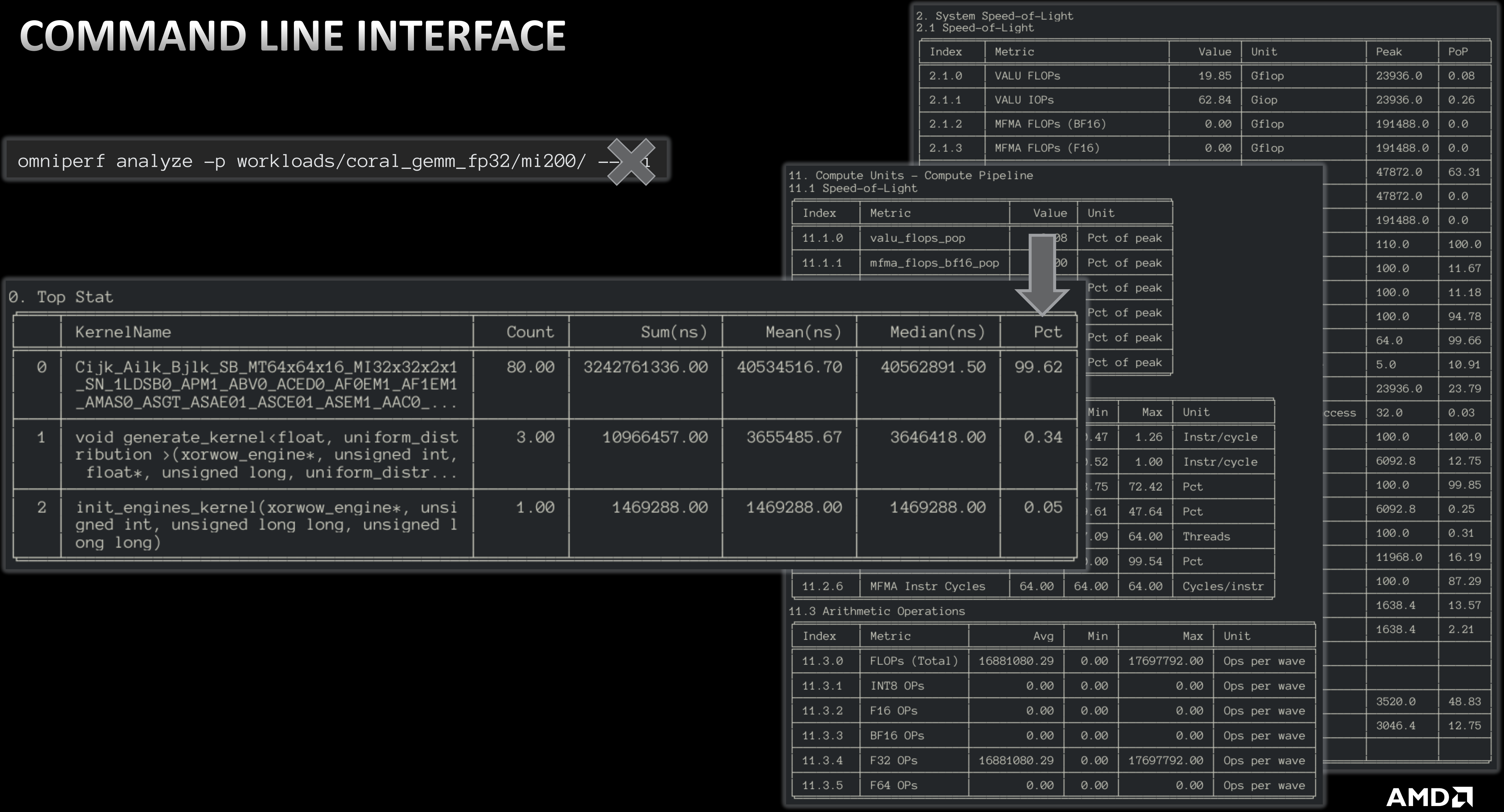
10.3 VMEM Instr Mix

type	Count
Buffer Instr	1112.00
Buffer Read	1096.00
Buffer Write	16.00
Buffer Atomic	0.00
Flat Instr	0.00
Flat Read	0.00
Flat Write	0.00
Flat Atomic	0.00

10.4 MFMA Arithmetic Instr Mix

type	Count
MFMA-I8	0.00
MFMA-F16	0.00
MFMA-BF16	0.00
MFMA-F32	4320.00
MFMA-F64	0.00

COMMAND LINE INTERFACE



```
omniperf analyze -p workloads/coral_gemm_fp32/mi200/ --gui
```

```
0. Top Stat
╒════╤══════════════════════════════════════════╤═════════╤═══════════════╤══════════════╤══════════════╤════════╕
│    │ KernelName                               │   Count │       Sum(ns) │     Mean(ns) │   Median(ns) │    Pct │
╞════╪══════════════════════════════════════════╪═════════╪═══════════════╪══════════════╪══════════════╪════════╡
│  0 │ Cijk_Ailk_Bjlk_SB_MT64x64x16_MI32x32x2x1 │   80.00 │ 3242761336.00 │  40534516.70 │  40562891.50 │  99.62 │
│    │ _SN_1LDSB0_APM1_ABV0_ACED0_AF0EM1_AF1EM1 │         │               │              │              │        │
│    │ _AMAS0_ASGT_ASAE01_ASCE01_ASEM1_AAC0_... │         │               │              │              │        │
├────┼──────────────────────────────────────────┼─────────┼───────────────┼──────────────┼──────────────┼────────┤
│  1 │ void generate_kernel<float, uniform_dist │    3.00 │   10966457.00 │   3655485.67 │   3646418.00 │   0.34 │
│    │ ribution >(xorwow_engine*, unsigned int, │         │               │              │              │        │
│    │  float*, unsigned long, uniform_distr... │         │               │              │              │        │
├────┼──────────────────────────────────────────┼─────────┼───────────────┼──────────────┼──────────────┼────────┤
│  2 │ init_engines_kernel(xorwow_engine*, unsi │    1.00 │    1469288.00 │   1469288.00 │   1469288.00 │   0.05 │
│    │ gned int, unsigned long long, unsigned l │         │               │              │              │        │
│    │ ong long)                                │         │               │              │              │        │
╘════╧══════════════════════════════════════════╧═════════╧═══════════════╧══════════════╧══════════════╧════════╛
```

```
2. System Speed-of-Light
2.1 Speed-of-Light
│ Index │ Metric              │  Value │ Unit  │ Peak     │ PoP   │
│ 2.1.0 │ VALU FLOPs          │  19.85 │ Gflop │ 23936.0  │ 0.08  │
│ 2.1.1 │ VALU IOPs           │  62.84 │ Giop  │ 23936.0  │ 0.26  │
│ 2.1.2 │ MFMA FLOPs (BF16)   │   0.00 │ Gflop │ 191488.0 │ 0.0   │
│ 2.1.3 │ MFMA FLOPs (F16)    │   0.00 │ Gflop │ 191488.0 │ 0.0   │
                                               │ 47872.0  │ 63.31 │
                                               │ 47872.0  │ 0.0   │
                                               │ 191488.0 │ 0.0   │
                                               │ 110.0    │ 100.0 │
                                               │ 100.0    │ 11.67 │
                                               │ 100.0    │ 11.18 │
                                               │ 100.0    │ 94.78 │
                                               │ 64.0     │ 99.66 │
                                               │ 5.0      │ 10.91 │
                                               │ 23936.0  │ 23.79 │
                                        ccess  │ 32.0     │ 0.03  │
                                               │ 100.0    │ 100.0 │
                                               │ 6092.8   │ 12.75 │
                                               │ 100.0    │ 99.85 │
                                               │ 6092.8   │ 0.25  │
                                               │ 100.0    │ 0.31  │
                                               │ 11968.0  │ 16.19 │
                                               │ 100.0    │ 87.29 │
                                               │ 1638.4   │ 13.57 │
                                               │ 1638.4   │ 2.21  │
                                               │          │       │
                                               │          │       │
                                               │ 3520.0   │ 48.83 │
                                               │ 3046.4   │ 12.75 │
                                               │          │       │
```

```
11. Compute Units - Compute Pipeline
11.1 Speed-of-Light
│ Index  │ Metric             │ Value │ Unit        │
│ 11.1.0 │ valu_flops_pop     │    08 │ Pct of peak │
│ 11.1.1 │ mfma_flops_bf16_pop│    00 │ Pct of peak │
                                      │ Pct of peak │
                                      │ Pct of peak │
                                      │ Pct of peak │
                                      │ Pct of peak │

                              │ Min   │   Max │ Unit         │
                              │ .47   │  1.26 │ Instr/cycle  │
                              │ .52   │  1.00 │ Instr/cycle  │
                              │ .75   │ 72.42 │ Pct          │
                              │ .61   │ 47.64 │ Pct          │
                              │ .09   │ 64.00 │ Threads      │
                              │ .00   │ 99.54 │ Pct          │
│ 11.2.6 │ MFMA Instr Cycles  │ 64.00 │ 64.00 │ 64.00 │ Cycles/instr │

11.3 Arithmetic Operations
│ Index  │ Metric       │         Avg │  Min │         Max │ Unit         │
│ 11.3.0 │ FLOPs (Total)│ 16881080.29 │ 0.00 │ 17697792.00 │ Ops per wave │
│ 11.3.1 │ INT8 OPs     │        0.00 │ 0.00 │        0.00 │ Ops per wave │
│ 11.3.2 │ F16 OPs      │        0.00 │ 0.00 │        0.00 │ Ops per wave │
│ 11.3.3 │ BF16 OPs     │        0.00 │ 0.00 │        0.00 │ Ops per wave │
│ 11.3.4 │ F32 OPs      │ 16881080.29 │ 0.00 │ 17697792.00 │ Ops per wave │
│ 11.3.5 │ F64 OPs      │        0.00 │ 0.00 │        0.00 │ Ops per wave │
```

AMD
together we advance_

OTHER CAPABILITIES

- filtering
  - by kernel
  - by dispatch
  - by IP block
- roofline only

QUESTIONS?

https://amdresearch.github.io/omniperf/

```
# general usage help
omniperf --help

# profiling usage help
omniperf profile --help

# database interactions help
omniperf database --help

# analysis help
omniperf analyze --help
```

DISCLAIMERS

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Third-party content is licensed to you directly by the third party that owns the content and is not licensed to you by AMD. ALL LINKED THIRD-PARTY CONTENT IS PROVIDED "AS IS" WITHOUT A WARRANTY OF ANY KIND. USE OF SUCH THIRD-PARTY CONTENT IS DONE AT YOUR SOLE DISCRETION AND UNDER NO CIRCUMSTANCES WILL AMD BE LIABLE TO YOU FOR ANY THIRD-PARTY CONTENT. YOU ASSUME ALL RISK AND ARE SOLELY RESPONSIBLE FOR ANY DAMAGES THAT MAY ARISE FROM YOUR USE OF THIRD-PARTY CONTENT.

ATTRIBUTIONS

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc.

Git and the Git logo are either registered trademarks or trademarks of Software Freedom Conservancy, Inc., corporate home of the Git Project, in the United States and/or other countries.

MongoDB is a registered trademark of MongoDB, Inc.

Python is a registered trademark of the PSF.

AMD