ALCF DEVELOPER SESSION

INTEL PERFORMANCE PROFILING TOOLS ON AURORA

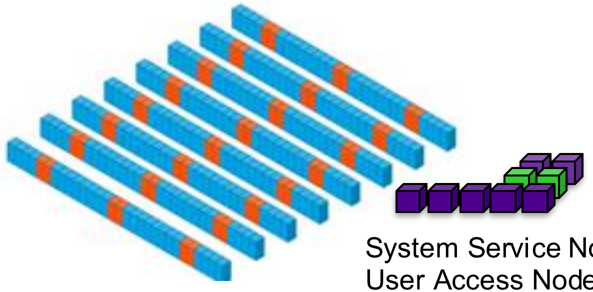
JAEHYUK KWACK
(jkwack@anl.gov)
ALCF Perf. Engr. Group

U.S. DEPARTMENT OF ENERGY
Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.

9/25/2024

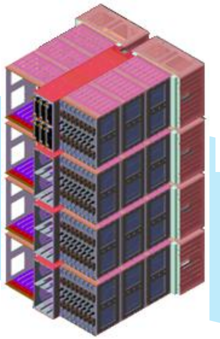
AURORA OVERVIEW

AURORA HIGH-LEVEL SYSTEM OVERVIEW

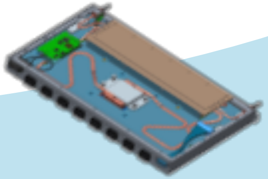


System Service Nodes (SSNs)
User Access Nodes (UANs)
DAOS Nodes (DNs)
Gateway Nodes (GNs)
IOF service, scalable library loading
DAOS <-> Lustre data mover

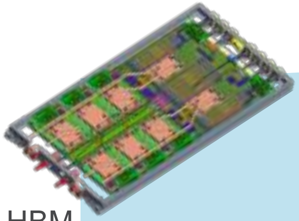
AURORA SYSTEM
166 Compute racks
10,624 Nodes
GPU: 8.16 PB HBM
CPU: 1.36 PB HBM, 10.9 PB DDR5
DAOS: 64 racks, 1024 nodes
230 PB (usable), 31 TB/s



COMPUTE RACK
64 Compute blades
32 Switch blades
GPU: 49.1 TB HBM
CPU: 8.2 TB HBM, 64 TB DDR5



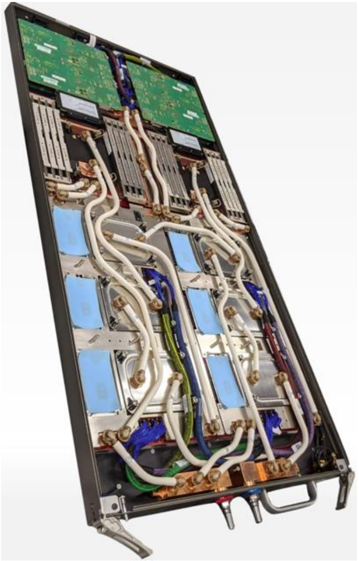
SWITCH BLADE
1 Slingshot switch
64 ports
Dragonfly topology

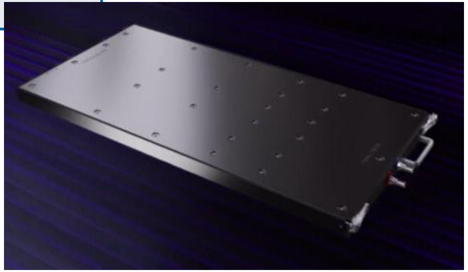


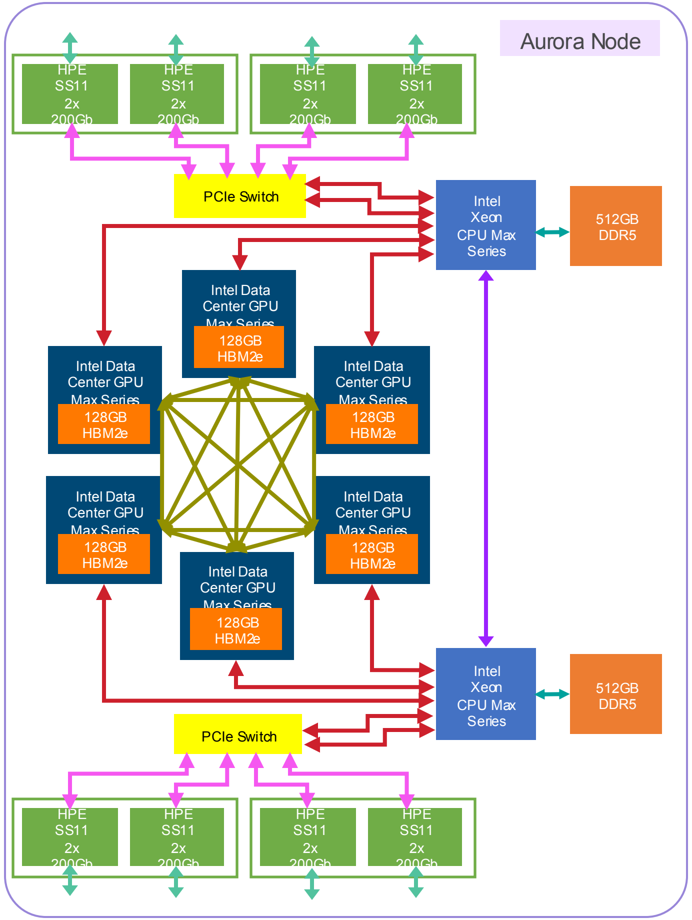
COMPUTE BLADE
2x Intel Xeon Max Series w HBM
6x Intel Data Center GPU Max Series
GPU: 768 GB HBM
CPU: 128 GB HBM, 1024 GB DDR5

AURORA EXASCALE COMPUTE BLADE

NODE CHARACTERISTICS	
6	GPU - Intel Data Center GPU Max Series (#)
2	CPU - Intel Xeon CPU Max Series (#)
768	GPU HBM Memory (GB)
19.66	Peak GPU HBM BW (TB/s)
128	CPU HBM Memory (GB)
2.87	Peak CPU HBM BW (TB/s)
1024	CPU DDR5 Memory (GB)
0.56	Peak CPU DDR5 BW (TB/s)
≧ 130	Peak Node DP FLOPS (TF)
200	Max Fabric Injection (GB/s)
8	NICs (#)







PROFILING TOOLS ON AURORA

PROFILING TOOLS ON AURORA

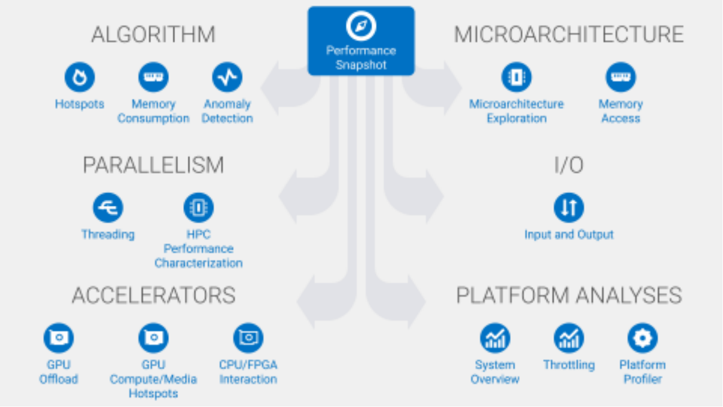
A list of popular profiling tools on Aurora

- Intel VTune
- Intel Advisor
- Intel APS (Application Performance Snapshot)
- Intel xpu-smi
- Intel unitrace

- THAPI from Argonne (https://github.com/argonne-lcf/THAPI)
- HPCToolkit from Rice University (https://hpctoolkit.org/)
- TAU from University of Oregon (https://www.paratools.com/tau)

INTEL VTUNE

Profile GPU Performance

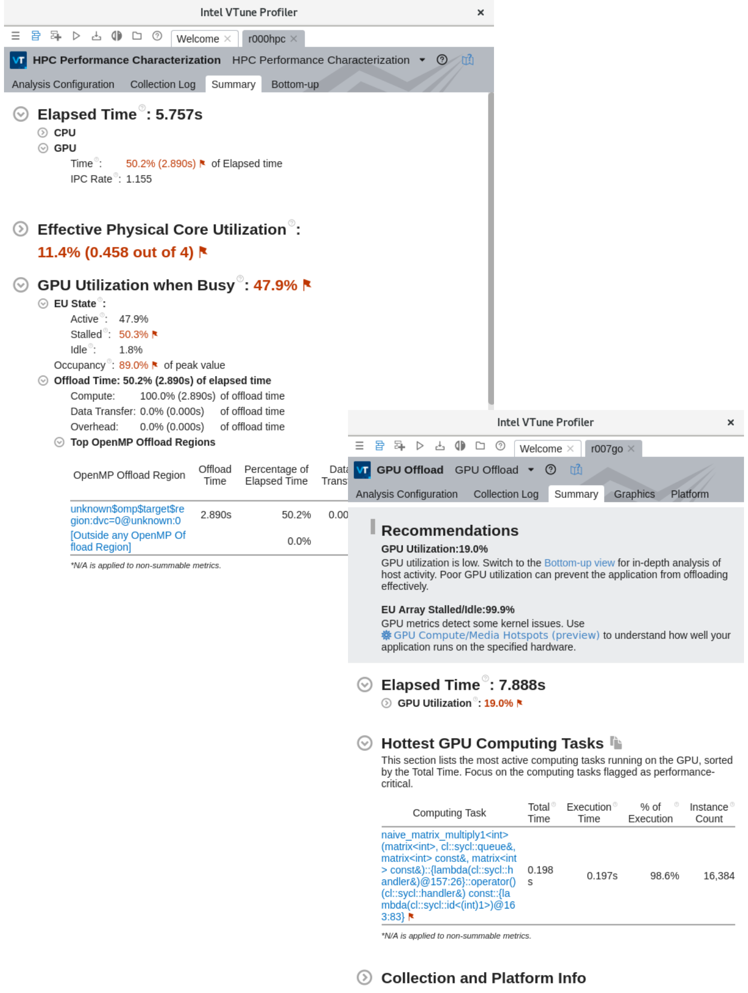


- Multi-GPU systems analysis
- GPU Offload cost profiling
  - CPU vs GPU boundness
  - Offload overhead & host-to-device traffic, GPU compute vs data transfer
  - GPU utilization and software queues
- GPU Hotspots analysis
  - XVE (Xe Vector Engine) and memory efficiency metrics, GPU occupancy limiting factors
  - Memory hierarchy diagram and throughput analysis
- Source level in-kernel profiling (need to build with "***-fdebug-info-for-profiling -gline-tables-only***")
  - Dynamic instruction count
  - Basic Block execution latency
  - Memory latency
  - HW-assisted stall sampling
  - Memory Access Analysis Tool (MAAT)

INTEL VTUNE

Analysis types for Intel GPUs

- HPC Performance Characterization
  - `$ <mpi launcher> <mpi param-s> vtune -collect hpc-performance -r <result_dir> <my app> <app param-s>`
  - Provides a different aspect of application performance
  - High level hardware information, CPU cores utilization, GPU stacks utilization including XVE HW metrics and top offload regions, CPU-side memory metrics, and CPU instruction statics
- GPU offload
  - `$ <mpi launcher> <mpi param-s> vtune -collect gpu-offload -r <result_dir> <my app> <app param-s>`
  - Serves studies of an application offload implementation and assesses its efficiency
  - Traces Level-zero and OpenCL API functions in oneAPI software stack; detects long latency host functions; shows time spent in data allocation and transfer function as well as kernel device time



Elapsed Time: 5.757s
CPU
GPU
Time: 50.2% (2.890s) of Elapsed time
IPC Rate: 1.155

Effective Physical Core Utilization: 11.4% (0.458 out of 4)

GPU Utilization when Busy: 47.9%
EU State:
Active: 47.9%
Stalled: 50.3%
Idle: 1.8%
Occupancy: 89.0% of peak value
Offload Time: 50.2% (2.890s) of elapsed time
Compute: 100.0% (2.890s) of offload time
Data Transfer: 0.0% (0.000s) of offload time
Overhead: 0.0% (0.000s) of offload time
Top OpenMP Offload Regions

OpenMP Offload Region	Offload Time	Percentage of Elapsed Time	Data Trans
unknown$omp$target$region:dvc=0@unknown:0	2.890s	50.2%	0.00
[Outside any OpenMP Offload Region]		0.0%	

*N/A is applied to non-summable metrics.

Recommendations
GPU Utilization:19.0%
GPU utilization is low. Switch to the Bottom-up view for in-depth analysis of host activity. Poor GPU utilization can prevent the application from offloading effectively.

EU Array Stalled/Idle:99.9%
GPU metrics detect some kernel issues. Use GPU Compute/Media Hotspots (preview) to understand how well your application runs on the specified hardware.

Elapsed Time: 7.888s
GPU Utilization: 19.0%

Hottest GPU Computing Tasks
This section lists the most active computing tasks running on the GPU, sorted by the Total Time. Focus on the computing tasks flagged as performance-critical.

Computing Task	Total Time	Execution Time	% of Execution	Instance Count
naive_matrix_multiply1<int>(matrix<int>, cl::sycl::queue&, matrix<int> const&, matrix<int> const&)::{lambda(cl::sycl::handler&)@157:26}::operator()(cl::sycl::handler&) const::{lambda(cl::sycl::id<(int)1>)@163:83}	0.198s	0.197s	98.6%	16,384

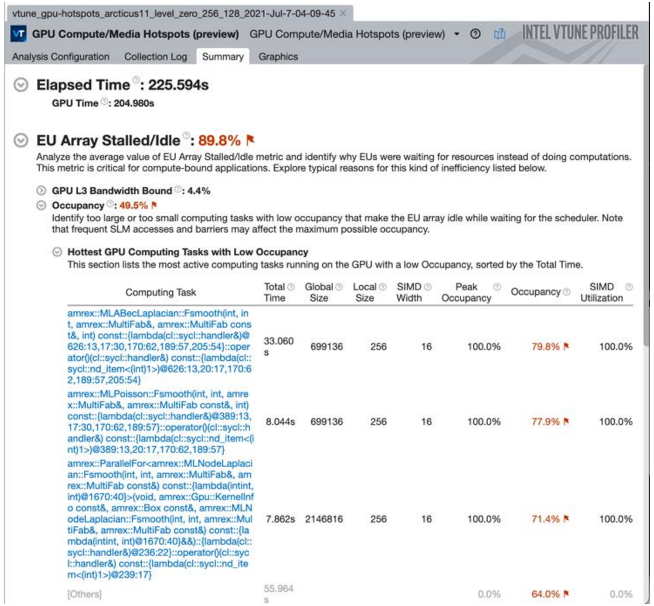
*N/A is applied to non-summable metrics.

Collection and Platform Info

INTEL VTUNE

Analysis types for Intel GPUs

- GPU Compute/Media Hotspots
  - `$ <mpi launcher> <mpi param-s> vtune -collect gpu-hotspots -r <result_dir> <my app> <app param-s>`
  - The most accurate analysis in tracing kernels on GPU
  - Allows to analyze the most time-consuming GPU kernels, characterize GPU usage based on GPU hardware metrics, identify performance issues caused by memory latency or inefficient kernel algorithms, and analyze GPU instruction frequency per certain instruction types.


vtune_gpu-hotspots_arcticus11_level_zero_256_128_2021-Jul-7-04-09-45
GPU Compute/Media Hotspots (preview)
GPU Compute/Media Hotspots (preview)
INTEL VTUNE PROFILER
Analysis Configuration
Collection Log
Summary
Graphics

Elapsed Time: 225.594s

GPU Time: 204.980s

EU Array Stalled/Idle: 89.8%

Analyze the average value of EU Array Stalled/Idle metric and identify why EUs were waiting for resources instead of doing computations. This metric is critical for compute-bound applications. Explore typical reasons for this kind of inefficiency listed below.

GPU L3 Bandwidth Bound: 4.4%

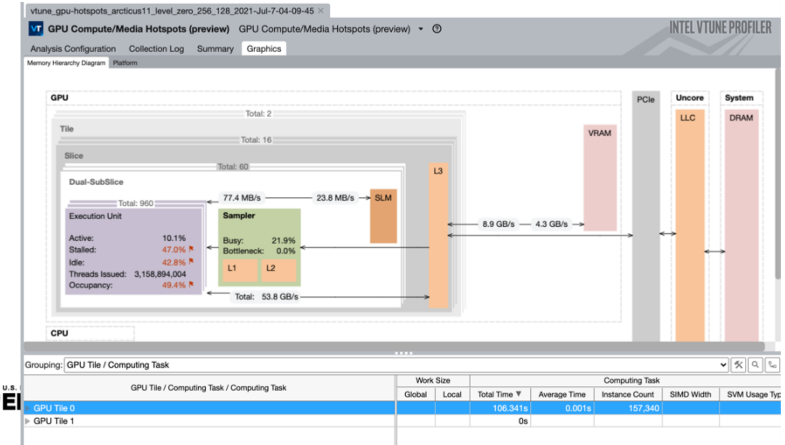
Occupancy: 49.5%

Identify too large or too small computing tasks with low occupancy that make the EU array idle while waiting for the scheduler. Note that frequent SLM accesses and barriers may affect the maximum possible occupancy.

Hottest GPU Computing Tasks with Low Occupancy

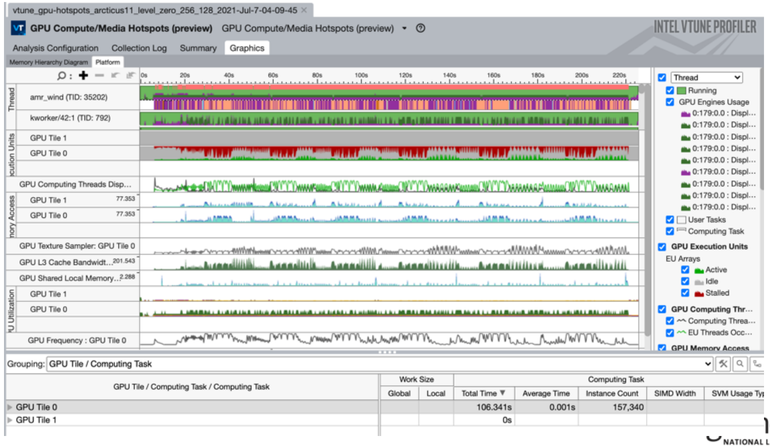
This section lists the most active computing tasks running on the GPU with a low Occupancy, sorted by the Total Time.

Computing Task	Total Time	Global Size	Local Size	SIMD Width	Peak Occupancy	Occupancy	SIMD Utilization
amrex::MLABecLaplacian::Fsmooth(int, int, amrex::MultiFab&, amrex::MultiFab const&, int) const::{lambda(cl::sycl::handler&)@626:13,17:30,170:62,189:57,205:54}::operator()(cl::sycl::handler&) const::{lambda(cl::sycl::nd_item<(int)1>)@626:13,20:17,170:62,189:57,205:54}	33.060s	699136	256	16	100.0%	79.8%	100.0%
amrex::MLPoisson::Fsmooth(int, int, amrex::MultiFab&, amrex::MultiFab const&, int) const::{lambda(cl::sycl::handler&)@389:13,17:30,170:62,189:57}::operator()(cl::sycl::handler&) const::{lambda(cl::sycl::nd_item<(int)1>)@389:13,20:17,170:62,189:57}	8.044s	699136	256	16	100.0%	77.9%	100.0%
amrex::ParallelFor<amrex::MLNodeLaplacian::Fsmooth(int, int, amrex::MultiFab&, amrex::MultiFab const&) const::{lambda(intint, int)@1670:40}>(void, amrex::Gpu::KernelInfo const&, amrex::Box const&, amrex::MLNodeLaplacian::Fsmooth(int, int, amrex::MultiFab&, amrex::MultiFab const&) const::{lambda(intint, int)@1670:40}&&)::{lambda(cl::sycl::handler&)@236:22}::operator()(cl::sycl::handler&) const::{lambda(cl::sycl::nd_item<(int)1>)@239:17}	7.862s	2146816	256	16	100.0%	71.4%	100.0%
[Others]	55.964s				0.0%	64.0%	0.0%



Grouping: GPU Tile / Computing Task

GPU Tile / Computing Task / Computing Task	Work Size		Computing Task				
	Global	Local	Total Time ▼	Average Time	Instance Count	SIMD Width	SVM Usage Typ
GPU Tile 0			106.341s	0.001s	157,340		
GPU Tile 1			0s				



Grouping: GPU Tile / Computing Task

GPU Tile / Computing Task / Computing Task	Work Size		Computing Task				
	Global	Local	Total Time ▼	Average Time	Instance Count	SIMD Width	SVM Usage Typ
GPU Tile 0			106.341s	0.001s	157,340		
GPU Tile 1			0s				

VTune server for pre-collected results on Aurora via SSH terminal

- Step1: Open a new terminal and log into Aurora login node (no X11 forwarding required)

```
$ ssh <username>@bastion.alcf.anl.gov
$ ssh <username>@login.aurora.alcf.anl.gov
```

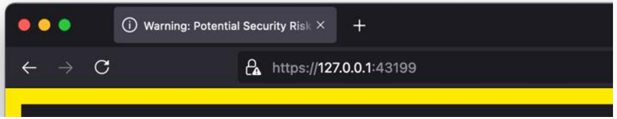
- Step2: Start VTune server on an Aurora login node after loading oneapi module and setting corresponding environmental variables for VTune

```
$ module load oneapi
$ vtune-backend --data-directory=<location of precollected VTune results>
```

- Step3: Open a new terminal with SSH port forwarding enabled (need 2 hops):

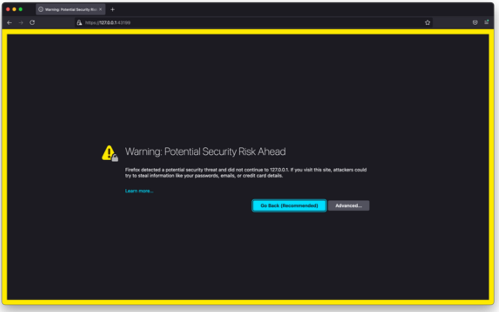
```
$ ssh -L 127.0.0.1:<port printed by vtune-backend>:127.0.0.1:<port printed by vtune-backend>
<username>@bastion.alcf.anl.gov
$ ssh -L 127.0.0.1:<port printed by vtune-backend>:127.0.0.1:<port printed by vtune-backend>
<username>@login.aurora.alcf.anl.gov
```

- Step4: Open the URL printed by VTune server in your local web browser (e.g., firefox on your laptop)



Warning: Potential Security Risk ×

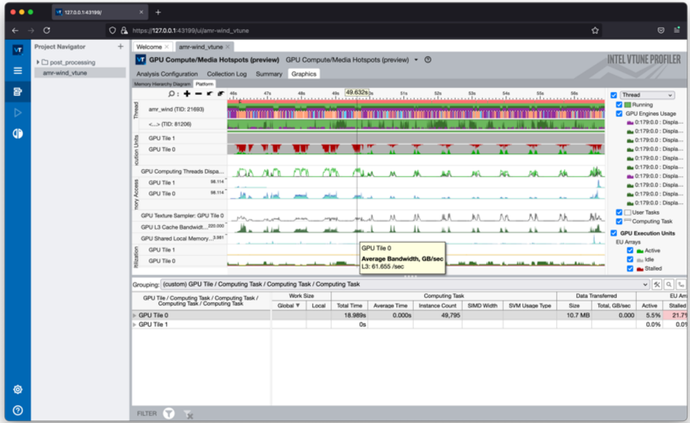
https://127.0.0.1:43199



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to 127.0.0.1. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

Learn more...

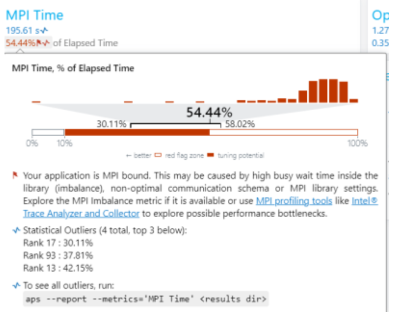


https://127.0.0.1:43199/ui/amr-wind_vtune

INTEL APS

VTune Profiler's Application Performance Snapshot

- Provides an aggregated view of an application at scale, designed for large MPI workloads.
- Captures performance aspects of compute intensive applications
  - MPI and OpenMP usage and imbalance,
  - CPU and GPU utilization
  - CPU stalls due to memory accesses, vectorization, I/O, and memory footprint
- Command lines
  - `$ <mpi launcher> <mpi param-s> aps -r <result_dir> <my app> [<app param-s>]`
  - `$ aps-report <result_dir>`



MPI Time
195.61 s
54.44% of Elapsed Time

Op
1.27
0.35

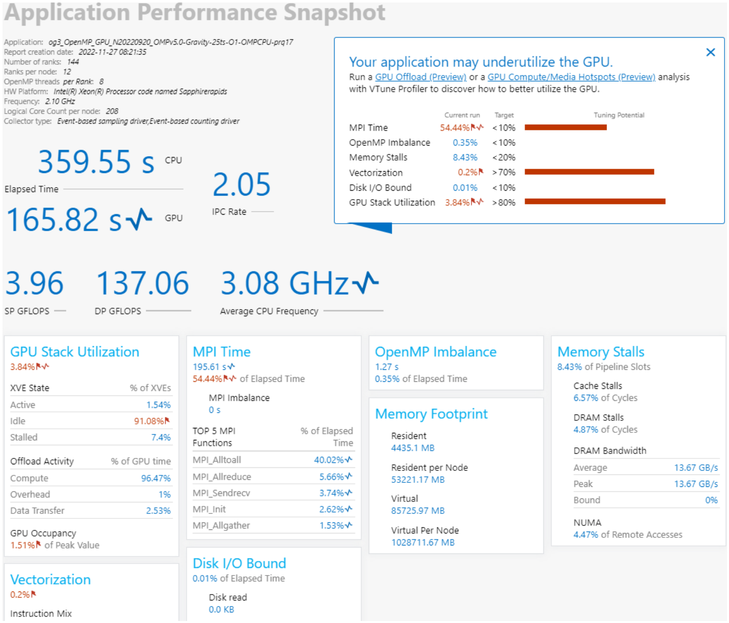
MPI Time, % of Elapsed Time

Your application is MPI bound. This may be caused by high busy wait time inside the library (imbalance), non-optimal communication schema or MPI library settings. Explore the MPI Imbalance metric if it is available or use MPI profiling tools like Intel® Trace Analyzer and Collector to explore possible performance bottlenecks.

Statistical Outliers (4 total, top 3 below):
Rank 17 : 30.11%
Rank 93 : 37.81%
Rank 13 : 42.15%

To see all outliers, run:

```
aps --report --metrics='MPI Time' <results dir>
```



Application Performance Snapshot

Application: *og3_OpenMP_GPU_N20220920_OMPv5.0-Gravity-25ts-O1-OMPCPU-prq17*
Report creation date: *2022-11-27 08:21:35*
Number of ranks: *144*
Ranks per node: *12*
OpenMP threads *per Rank: 8*
HW Platform: *Intel(R) Xeon(R) Processor code named Sapphirerapids*
Frequency: *2.10 GHz*
Logical Core Count per node: *208*
Collector type: *Event-based sampling driver,Event-based counting driver*

359.55 s CPU
Elapsed Time
165.82 s GPU

2.05
IPC Rate

3.96
SP GFLOPS

137.06
DP GFLOPS

3.08 GHz
Average CPU Frequency

Your application may underutilize the GPU.
Run a GPU Offload (Preview) or a GPU Compute/Media Hotspots (Preview) analysis with VTune Profiler to discover how to better utilize the GPU.

	Current run	Target	Tuning Potential
MPI Time	54.44%	<10%	
OpenMP Imbalance	0.35%	<10%	
Memory Stalls	8.43%	<20%	
Vectorization	0.2%	>70%	
Disk I/O Bound	0.01%	<10%	
GPU Stack Utilization	3.84%	>80%	

GPU Stack Utilization
3.84%

XVE State	% of XVEs
Active	1.54%
Idle	91.08%
Stalled	7.4%

Offload Activity	% of GPU time
Compute	96.47%
Overhead	1%
Data Transfer	2.53%

GPU Occupancy
1.51% of Peak Value

Vectorization
0.2%

Instruction Mix

MPI Time
195.61 s
54.44% of Elapsed Time

MPI Imbalance
0 s

TOP 5 MPI Functions	% of Elapsed Time
MPI_Alltoall	40.02%
MPI_Allreduce	5.66%
MPI_Sendrecv	3.74%
MPI_Init	2.62%
MPI_Allgather	1.53%

Disk I/O Bound
0.01% of Elapsed Time

Disk read
0.0 KB

OpenMP Imbalance
1.27 s
0.35% of Elapsed Time

Memory Footprint

Resident
4435.1 MB

Resident per Node
53221.17 MB

Virtual
85725.97 MB

Virtual Per Node
1028711.67 MB

Memory Stalls
8.43% of Pipeline Slots

Cache Stalls
6.57% of Cycles

DRAM Stalls
4.87% of Cycles

DRAM Bandwidth

Average	13.67 GB/s
Peak	13.67 GB/s
Bound	0%

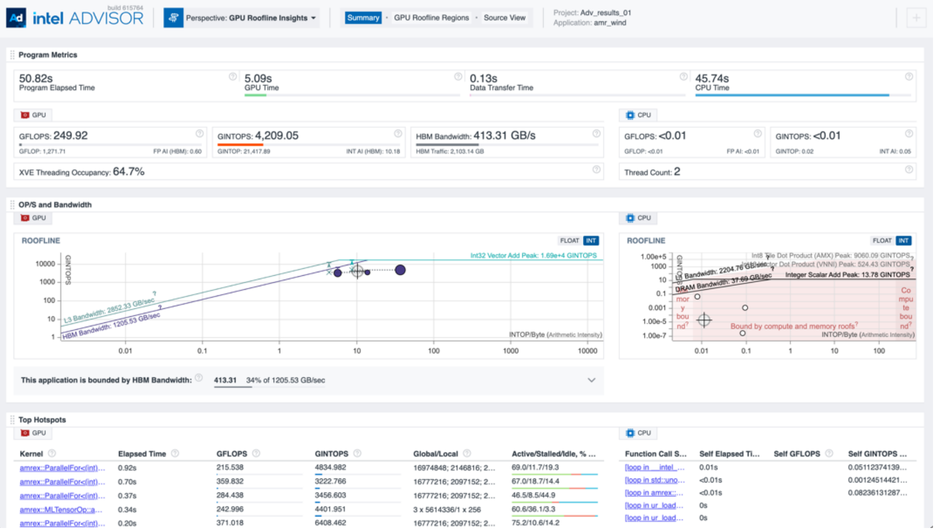
NUMA
4.47% of Remote Accesses

INTEL ADVISOR

Overview

- A design and analysis tool for developing performant code
  - Performant CPU Code: Design your application for efficient threading, vectorization, and memory use.
  - Efficient GPU Offload: Identify parts of the code that can be profitably offloaded. Optimize the code for compute and memory.
  - Flow Graph Design and Analysis: Create, visualize, and analyze task and dependency computation for heterogeneous algorithms.

Summary of CPU/GPU analysis



Project: Adv_results_01
Application: amr_wind

Program Metrics

50.82s Program Elapsed Time

5.09s GPU Time

0.13s Data Transfer Time

45.74s CPU Time

GPU

GFLOPS: 249.92 — GFLOP: 1,271.71 — FP AI (HBM): 0.60

GINTOPS: 4,209.05 — GINTOP: 21,417.89 — INT AI (HBM): 10.18

HBM Bandwidth: 413.31 GB/s — HBM Traffic: 2,103.14 GB

XVE Threading Occupancy: 64.7%

CPU

GFLOPS: <0.01 — GFLOP: <0.01 — FP AI: <0.01

GINTOPS: <0.01 — GINTOP: 0.02 — INT AI: 0.05

Thread Count: 2

OP/S and Bandwidth

GPU
ROOFLINE
FLOAT
INT
GINTOPS
Int32 Vector Add Peak: 1.69e+4 GINTOPS
L3 Bandwidth: 2852.33 GB/sec
HBM Bandwidth: 1205.53 GB/sec
10000
1000
100
10
1
0.01
0.1
1
10
100
1000
10000
INTOP/Byte (Arithmetic Intensity)
CPU
ROOFLINE
FLOAT
INT
Int8 Tile Dot Product (AMX) Peak: 9060.09 GINTOPS
Int8 Vector Dot Product (VNNI) Peak: 524.43 GINTOPS
L3 Bandwidth: 2204.76 GB/sec
DRAM Bandwidth: 37.69 GB/sec
Integer Scalar Add Peak: 13.78 GINTOPS
memory bound
Compute bound
Bound by compute and memory roofs
1.00e+5
1000
10
0.1
0.001
1.00e-5
1.00e-7
0.01
0.1
1
10
100
INTOP/Byte (Arithmetic Intensity)

This application is bounded by HBM Bandwidth: 413.31 34% of 1205.53 GB/sec

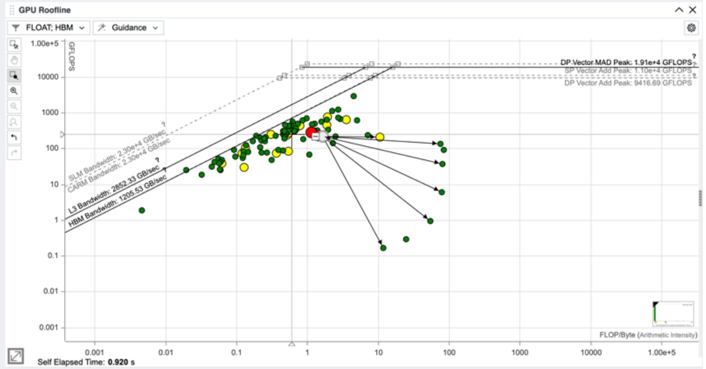
Top Hotspots

GPU

Kernel	Elapsed Time	GFLOPS	GINTOPS	Global/Local	Active/Stalled/Idle, % ...
amrex::ParallelFor<(int)...	0.92s	215.538	4834.982	16974848; 2146816; 2...	69.0/11.7/19.3
amrex::ParallelFor<(int)...	0.70s	359.832	3222.766	16777216; 2097152; 2...	67.0/18.7/14.4
amrex::ParallelFor<(int)...	0.37s	284.438	3456.603	16777216; 2097152; 2...	46.5/8.5/44.9
amrex::MLTensorOp::a...	0.34s	242.996	4401.951	3 x 5614336/1 x 256	60.6/36.1/3.3
amrex::ParallelFor<(int)...	0.20s	371.018	6408.462	16777216; 2097152; 2...	75.2/10.6/14.2

CPU

Function Call S...	Self Elapsed Ti...	Self GFLOPS	Self GINTOPS ...
[loop in __intel_...	0.01s		0.05112374139...
[loop in std::uno...	<0.01s		0.00124514421...
[loop in amrex::...	<0.01s		0.08236131287...
[loop in ur_load...	0s		
[loop in ur_load...	0s		



Self Elapsed Time: **0.920** s

GPU Roofline
w/ memory hierarchy

INTEL ADVISOR

Advisor Roofline analysis

- Advisor version on Aurora

```
$ advisor --version
Intel(R) Advisor 2024.2.1 (build 615624) Command Line Tool
Copyright (C) 2009-2024 Intel Corporation. All rights reserved.
```

- Step1: Setting the environments

```
$ module load oneapi
$ export PRJ=<your_project_dir>
```

- Step 2-a: Collecting the GPU Roofline data on a single GPU (Survey analysis and Trip Count with FLOP analysis)

```
$ advisor --collect=roofline --profile-gpu --project-dir=$PRJ -- <your_executable> <your_arguments>
```

- Step 2-b: Collecting the GPU Roofline data on one of MPI ranks(Survey analysis and Trip Count with FLOP analysis)

```
$ mpiexec -n 1 gpu_tile_compact.sh advisor --collect=survey --profile-gpu --project-dir=$PRJ --
<your_executable> <your_arguments> : -n 1 gpu_tile_compact.sh <your_executable> <your_arguments>
$ mpirun -n 1 gpu_tile_compact.sh advisor --collect=tripcounts --profile-gpu --flop --no-trip-counts --
project-dir=$PRJ -- <your_executable> <your_arguments> : -n 1 gpu_tile_compact.sh <your_executable>
<your_arguments>
```

- Step 3-a: Generate a GPU Roofline report, and then review the HTML report

```
$ advisor --report=all --project-dir=$PRJ --report-output=${PRJ}/roofline_all.html
```

- Step 3-b: Download the project folder to your laptop and open it with Advisor Client
  - https://www.intel.com/content/www/us/en/developer/articles/tool/oneapi-standalone-components.html#advisor

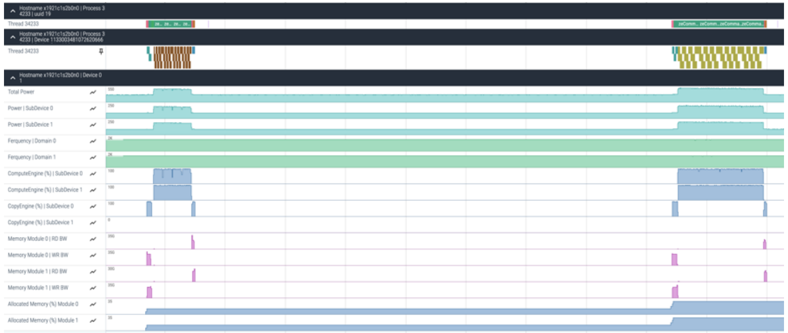
ANL Developers/PoCs: Brice Videau, Thomas Applencourt, Solomon Bekele

THAPI: TRACING HETEROGENEOUS APIs

A lightweight tool for tracing and sampling

- Overview
  - THAPI is a portable, programming model-centric tracing framework for heterogeneous systems.
    - OpenCL, L0, Cuda, HIP, OMPT, MPI
  - Two Components:
    - Tracing Events
      - LTTng based tracing
    - Parsing of the trace
      - Babeltrace2 library and tools

- Device Sampling
  - Ability to sample device telemetry with API tracing
    - Holistic view of system performance and behavior
    - Help understand H/W behavior in application context
    - Power/energy optimization
    - Resource management
    - Improves Debugging
  - Power, Fabric and Memory Traffic, Engine Activities
  - Timeline Visualization
    - Perfetto


Hostname x1921c1s2b0n0 | Process 34233 | uuid 19
Thread 34233
Hostname x1921c1s2b0n0 | Process 34233 | Device 1133003481072620666
Thread 34233
Hostname x1921c1s2b0n0 | Device 0 1
Total Power
Power | SubDevice 0
Power | SubDevice 1
Ferquency | Domain 0
Ferquency | Domain 1
ComputeEngine (%) | SubDevice 0
ComputeEngine (%) | SubDevice 1
CopyEngine (%) | SubDevice 0
CopyEngine (%) | SubDevice 1
Memory Module 0 | RD BW
Memory Module 0 | WR BW
Memory Module 1 | RD BW
Memory Module 1 | WR BW
Allocated Memory (%) Module 0
Allocated Memory (%) Module 1

ADVISOR & VTUNE EXAMPLES ON AURORA

WARMING-UP

Advisor on Aurora

- `$ advisor -version`
- `$ advisor -help`
- `$ advisor -help collect`

VTune on Aurora

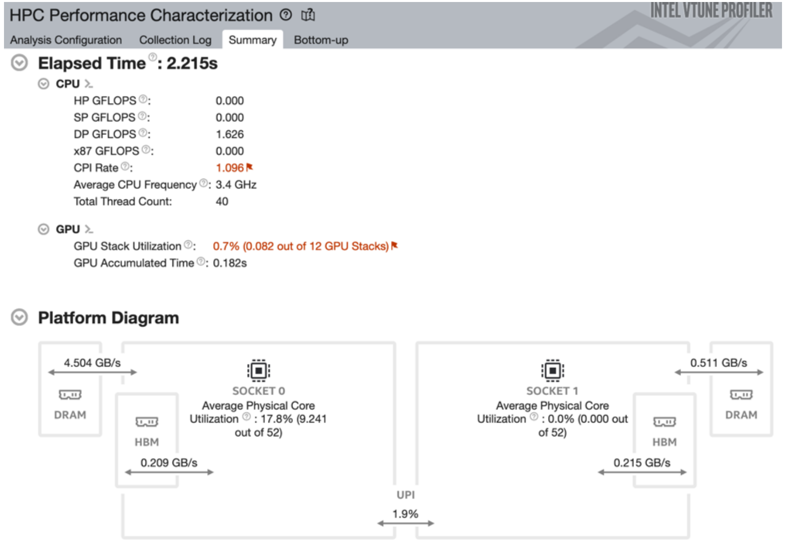
- `$ vtune -version`
- `$ vtune -help`
- `$ vtune -help collect`
- `$ vtune -help collect gpu-hotspots`
- `Vtune server for post-processing via ssh forwarding (i.e., $ vtune-backend)`

APS on Aurora

- `$ aps –version`
- `$ aps -help`

VTUNE HPC-PERFORMANCE ANALYSIS

```
$ mpiexec -n 12 -ppn 12 gpu_tile_compact.sh vtune -collect hpc-performance -r VTune_hpc-performance
./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```



HPC Performance Characterization
INTEL VTUNE PROFILER
Analysis Configuration
Collection Log
Summary
Bottom-up

Elapsed Time: 2.215s

CPU

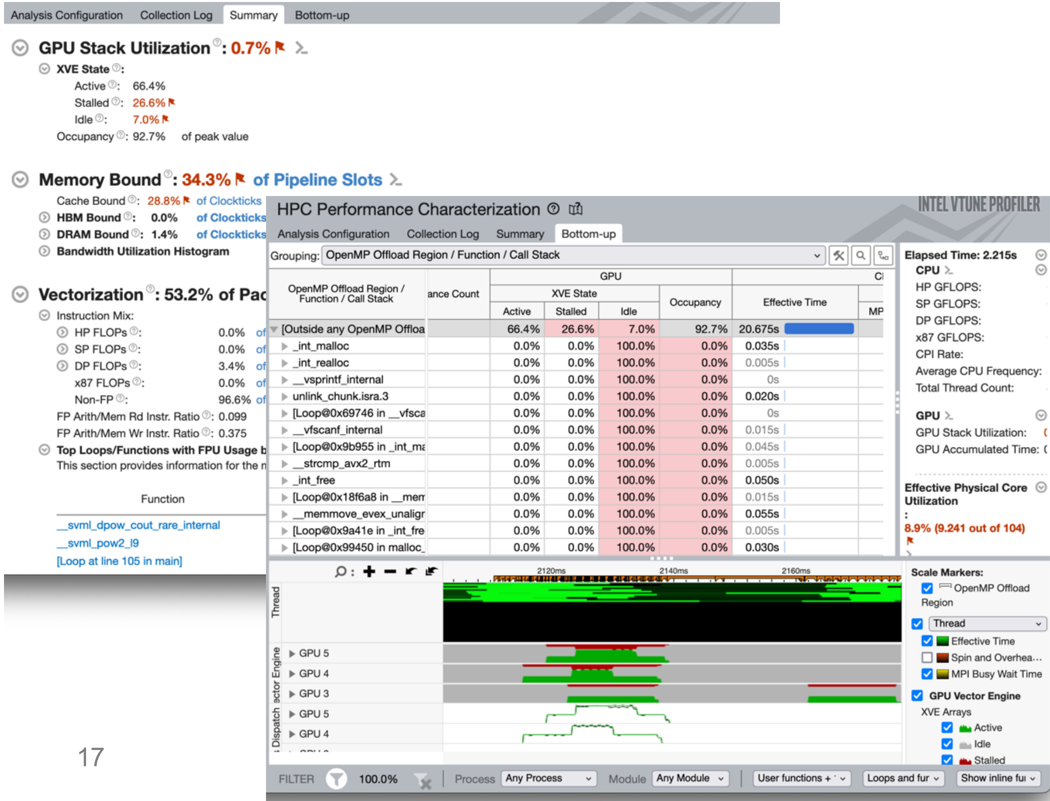
HP GFLOPS:	0.000
SP GFLOPS:	0.000
DP GFLOPS:	1.626
x87 GFLOPS:	0.000
CPI Rate:	1.096
Average CPU Frequency:	3.4 GHz
Total Thread Count:	40

GPU

GPU Stack Utilization:	0.7% (0.082 out of 12 GPU Stacks)
GPU Accumulated Time:	0.182s

Platform Diagram

4.504 GB/s
DRAM
SOCKET 0
Average Physical Core Utilization : 17.8% (9.241 out of 52)
HBM
0.209 GB/s
UPI
1.9%
SOCKET 1
Average Physical Core Utilization : 0.0% (0.000 out of 52)
HBM
0.215 GB/s
0.511 GB/s
DRAM



GPU Stack Utilization: 0.7%

XVE State:

Active:	66.4%
Stalled:	26.6%
Idle:	7.0%
Occupancy:	92.7% of peak value

Memory Bound: 34.3% of Pipeline Slots

Cache Bound:	28.8%	of Clockticks
HBM Bound:	0.0%	of Clockticks
DRAM Bound:	1.4%	of Clockticks
Bandwidth Utilization Histogram		

Vectorization: 53.2% of Pac

Instruction Mix:

HP FLOPs:	0.0%	of
SP FLOPs:	0.0%	of
DP FLOPs:	3.4%	of
x87 FLOPs:	0.0%	of
Non-FP:	96.6%	of
FP Arith/Mem Rd Instr. Ratio:	0.099	
FP Arith/Mem Wr Instr. Ratio:	0.375	

Top Loops/Functions with FPU Usage b

This section provides information for the n

Function
__svml_dpow_cout_rare_internal
__svml_pow2_l9
[Loop at line 105 in main]

Grouping: OpenMP Offload Region / Function / Call Stack

OpenMP Offload Region / Function / Call Stack	ance Count	GPU					CI
		XVE State			Occupancy	Effective Time	
		Active	Stalled	Idle			MP
[Outside any OpenMP Offloa		66.4%	26.6%	7.0%	92.7%	20.675s	
_int_malloc		0.0%	0.0%	100.0%	0.0%	0.035s	
_int_realloc		0.0%	0.0%	100.0%	0.0%	0.005s	
__vsprintf_internal		0.0%	0.0%	100.0%	0.0%	0s	
unlink_chunk.isra.3		0.0%	0.0%	100.0%	0.0%	0.020s	
[Loop@0x69746 in __vfsca		0.0%	0.0%	100.0%	0.0%	0s	
__vfscanf_internal		0.0%	0.0%	100.0%	0.0%	0.015s	
[Loop@0x9b955 in _int_ma		0.0%	0.0%	100.0%	0.0%	0.045s	
__strcmp_avx2_rtm		0.0%	0.0%	100.0%	0.0%	0.005s	
_int_free		0.0%	0.0%	100.0%	0.0%	0.050s	
[Loop@0x18f6a8 in __mem		0.0%	0.0%	100.0%	0.0%	0.015s	
__memmove_evex_unaligr		0.0%	0.0%	100.0%	0.0%	0.055s	
[Loop@0x9a41e in _int_fre		0.0%	0.0%	100.0%	0.0%	0.005s	
[Loop@0x99450 in malloc_		0.0%	0.0%	100.0%	0.0%	0.030s	

Elapsed Time: 2.215s

CPU

HP GFLOPS:

SP GFLOPS:

DP GFLOPS:

x87 GFLOPS:

CPI Rate:

Average CPU Frequency:

Total Thread Count:

GPU

GPU Stack Utilization: (

GPU Accumulated Time: (

Effective Physical Core Utilization

:

8.9% (9.241 out of 104)

Scale Markers:

☑ OpenMP Offload Region

☑ Thread

☑ Effective Time

☐ Spin and Overhea…

☑ MPI Busy Wait Time

☑ **GPU Vector Engine**

XVE Arrays

☑ Active

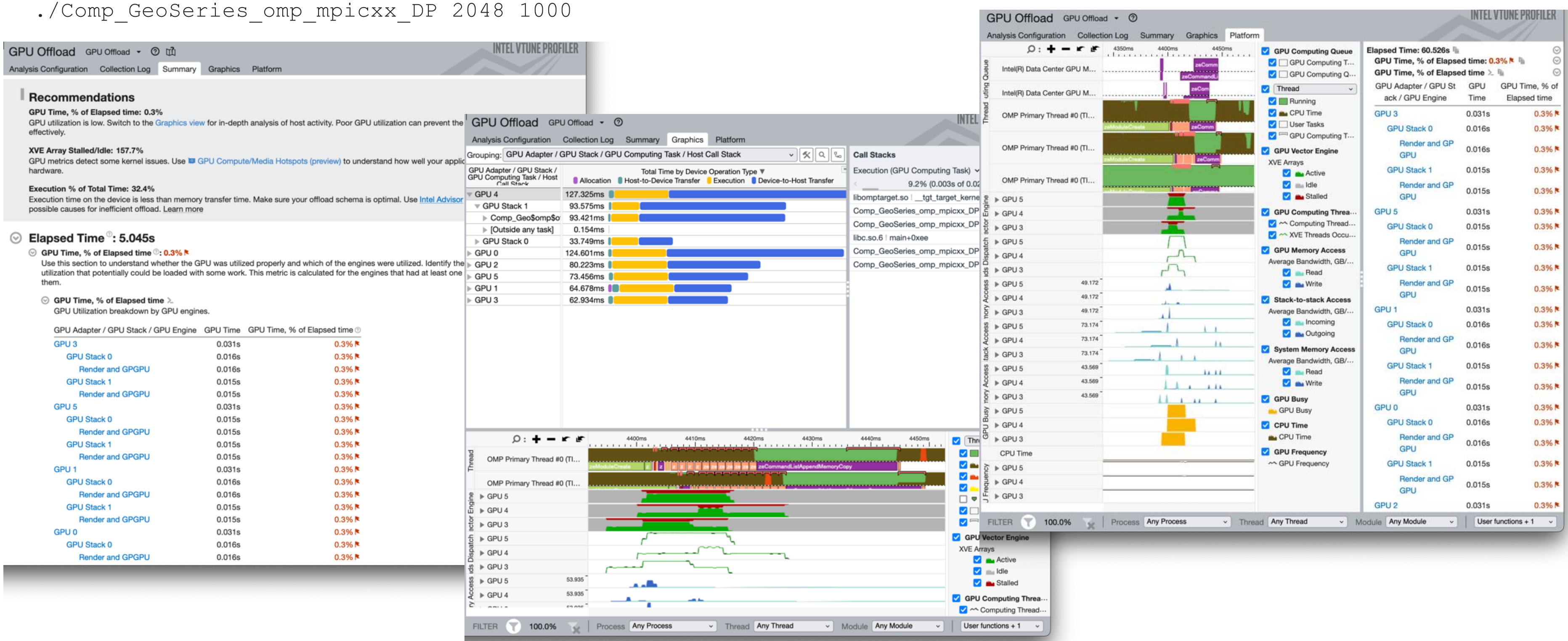
☑ Idle

☑ Stalled

FILTER 100.0% Process Any Process Module Any Module User functions + ˙ Loops and fur Show inline fu

VTUNE GPU-OFFLOAD ANALYSIS

```
$ mpiexec -n 12 -ppn 12 gpu_tile_compact.sh vtune -collect gpu-offload -r VTune_gpu-offload
./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```



GPU Offload — GPU Offload — Analysis Configuration | Collection Log | Summary | Graphics | Platform — INTEL VTUNE PROFILER

Recommendations

GPU Time, % of Elapsed time: 0.3%
GPU utilization is low. Switch to the Graphics view for in-depth analysis of host activity. Poor GPU utilization can prevent the effectively.

XVE Array Stalled/Idle: 157.7%
GPU metrics detect some kernel issues. Use GPU Compute/Media Hotspots (preview) to understand how well your applic hardware.

Execution % of Total Time: 32.4%
Execution time on the device is less than memory transfer time. Make sure your offload schema is optimal. Use Intel Advisor possible causes for inefficient offload. Learn more

Elapsed Time: 5.045s

GPU Time, % of Elapsed time: 0.3%
Use this section to understand whether the GPU was utilized properly and which of the engines were utilized. Identify the utilization that potentially could be loaded with some work. This metric is calculated for the engines that had at least one them.

GPU Time, % of Elapsed time
GPU Utilization breakdown by GPU engines.

GPU Adapter / GPU Stack / GPU Engine	GPU Time	GPU Time, % of Elapsed time
GPU 3	0.031s	0.3%
GPU Stack 0	0.016s	0.3%
Render and GPGPU	0.016s	0.3%
GPU Stack 1	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU 5	0.031s	0.3%
GPU Stack 0	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU Stack 1	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU 1	0.031s	0.3%
GPU Stack 0	0.016s	0.3%
Render and GPGPU	0.016s	0.3%
GPU Stack 1	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU 0	0.031s	0.3%
GPU Stack 0	0.016s	0.3%
Render and GPGPU	0.016s	0.3%

GPU Offload — GPU Offload — Analysis Configuration | Collection Log | Summary | Graphics | Platform

Grouping: GPU Adapter / GPU Stack / GPU Computing Task / Host Call Stack

GPU Adapter / GPU Stack / GPU Computing Task / Host Call Stack	Total Time by Device Operation Type (Allocation, Host-to-Device Transfer, Execution, Device-to-Host Transfer)
GPU 4	127.325ms
GPU Stack 1	93.575ms
Comp_Geo$omp$o	93.421ms
[Outside any task]	0.154ms
GPU Stack 0	33.749ms
GPU 0	124.601ms
GPU 2	80.223ms
GPU 5	73.456ms
GPU 1	64.678ms
GPU 3	62.934ms

Call Stacks
Execution (GPU Computing Task)
9.2% (0.003s of 0.02

- libomptarget.so ! __tgt_target_kerne
- Comp_GeoSeries_omp_mpicxx_DP
- Comp_GeoSeries_omp_mpicxx_DP
- libc.so.6 ! main+0xee
- Comp_GeoSeries_omp_mpicxx_DP
- Comp_GeoSeries_omp_mpicxx_DP

FILTER 100.0% | Process: Any Process | Thread: Any Thread | Module: Any Module | User functions + 1

GPU Offload — GPU Offload — Analysis Configuration | Collection Log | Summary | Graphics | Platform — INTEL VTUNE PROFILER

Elapsed Time: 60.526s
GPU Time, % of Elapsed time: 0.3%
GPU Time, % of Elapsed time

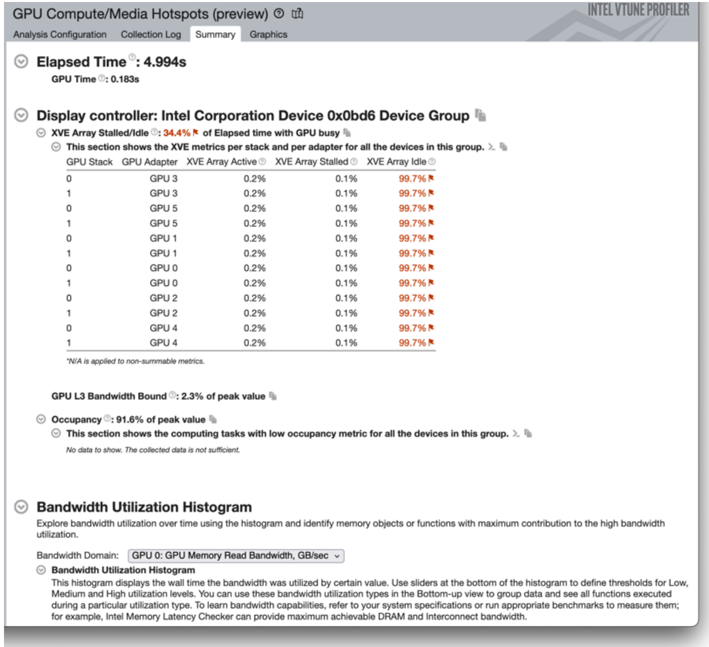
GPU Adapter / GPU Stack / GPU Engine	GPU Time	GPU Time, % of Elapsed time
GPU 3	0.031s	0.3%
GPU Stack 0	0.016s	0.3%
Render and GPGPU	0.016s	0.3%
GPU Stack 1	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU 5	0.031s	0.3%
GPU Stack 0	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU Stack 1	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU 1	0.031s	0.3%
GPU Stack 0	0.016s	0.3%
Render and GPGPU	0.016s	0.3%
GPU Stack 1	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU 0	0.031s	0.3%
GPU Stack 0	0.016s	0.3%
Render and GPGPU	0.016s	0.3%
GPU Stack 1	0.015s	0.3%
Render and GPGPU	0.015s	0.3%
GPU 2	0.031s	0.3%

FILTER 100.0% | Process: Any Process | Thread: Any Thread | Module: Any Module | User functions + 1

U.S. DEPARTMENT OF ENERGY — Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

VTUNE GPU-HOTSPOTS ANALYSIS

```
$ mpiexec -n 12 -ppn 12 gpu_tile_compact.sh vtune -collect gpu-hotspots -r VTune_gpu-hotspots
./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```



Elapsed Time: 4.994s

GPU Time: 0.183s

Display controller: Intel Corporation Device 0x0bd6 Device Group

XVE Array Stalled/Idle: 34.4% of Elapsed time with GPU busy

This section shows the XVE metrics per stack and per adapter for all the devices in this group.

GPU Stack	GPU Adapter	XVE Array Active	XVE Array Stalled	XVE Array Idle
0	GPU 3	0.2%	0.1%	99.7%
1	GPU 3	0.2%	0.1%	99.7%
0	GPU 5	0.2%	0.1%	99.7%
1	GPU 5	0.2%	0.1%	99.7%
0	GPU 1	0.2%	0.1%	99.7%
1	GPU 1	0.2%	0.1%	99.7%
0	GPU 0	0.2%	0.1%	99.7%
1	GPU 0	0.2%	0.1%	99.7%
0	GPU 2	0.2%	0.1%	99.7%
1	GPU 2	0.2%	0.1%	99.7%
0	GPU 4	0.2%	0.1%	99.7%
1	GPU 4	0.2%	0.1%	99.7%

*N/A is applied to non-summable metrics.

GPU L3 Bandwidth Bound: 2.3% of peak value

Occupancy: 91.6% of peak value

This section shows the computing tasks with low occupancy metric for all the devices in this group.

No data to show. The collected data is not sufficient.

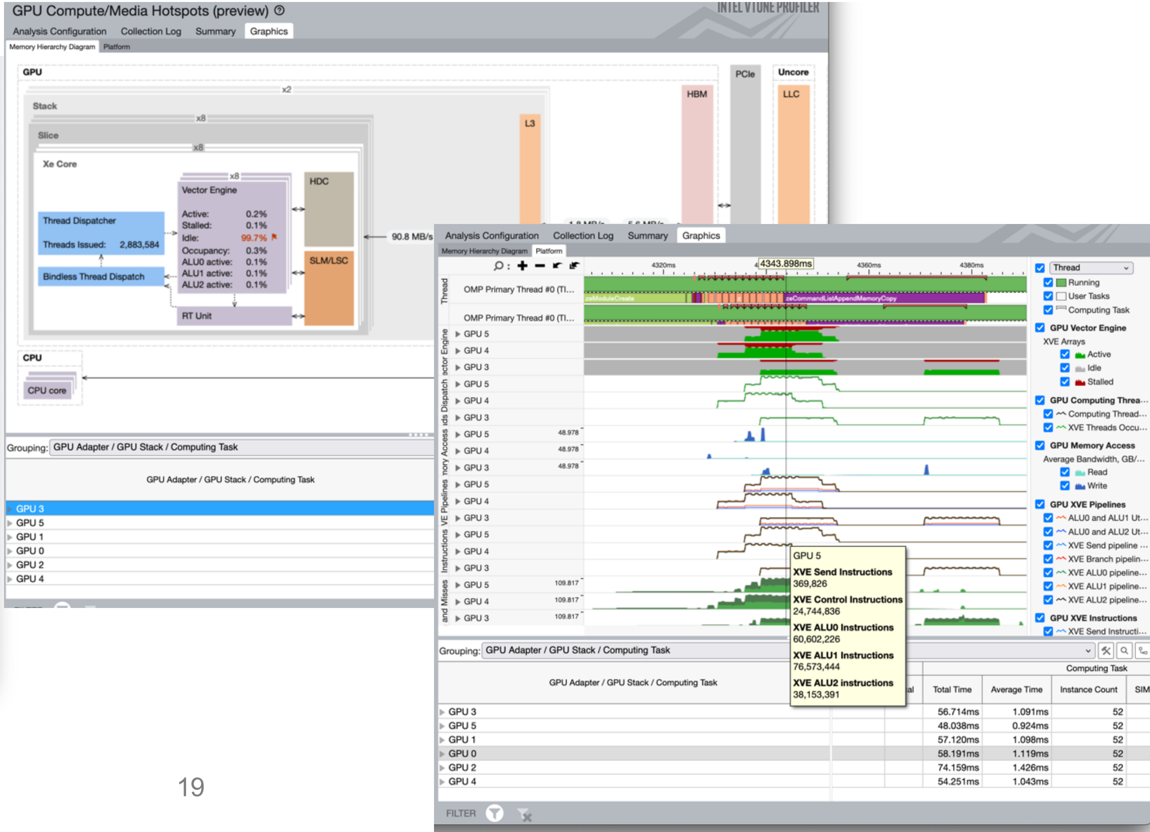
Bandwidth Utilization Histogram

Explore bandwidth utilization over time using the histogram and identify memory objects or functions with maximum contribution to the high bandwidth utilization.

Bandwidth Domain: GPU 0: GPU Memory Read Bandwidth, GB/sec

Bandwidth Utilization Histogram

This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed during a particular utilization type. To learn bandwidth capabilities, refer to your system specifications or run appropriate benchmarks to measure them; for example, Intel Memory Latency Checker can provide maximum achievable DRAM and Interconnect bandwidth.



Threads Issued: 2,883,584

Vector Engine	
Active:	0.2%
Stalled:	0.1%
Idle:	99.7%
Occupancy:	0.3%
ALU0 active:	0.1%
ALU1 active:	0.1%
ALU2 active:	0.1%

Grouping: GPU Adapter / GPU Stack / Computing Task

GPU Adapter / GPU Stack / Computing Task
GPU 3
GPU 5
GPU 1
GPU 0
GPU 2
GPU 4

GPU 5

XVE Send Instructions 369,826

XVE Control Instructions 24,744,836

XVE ALU0 Instructions 60,602,226

XVE ALU1 Instructions 76,573,444

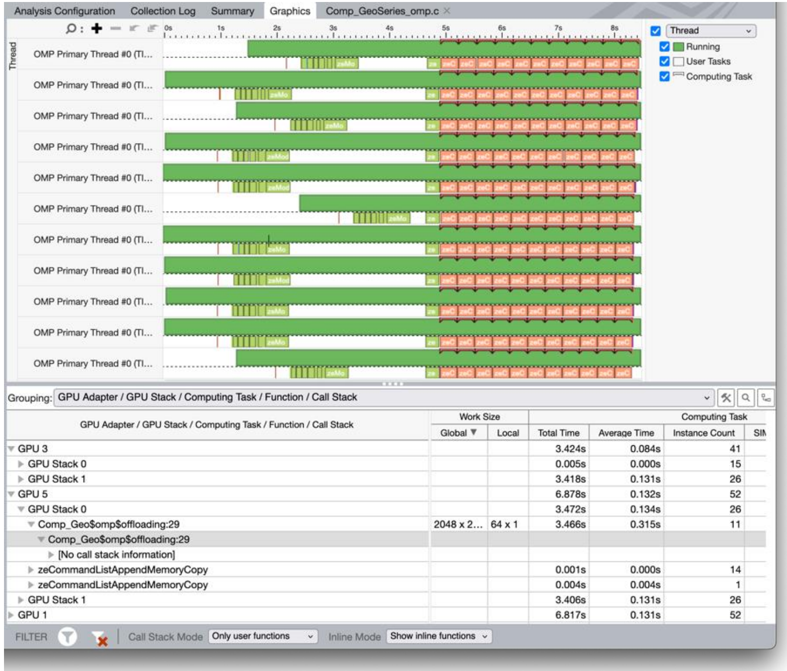
XVE ALU2 instructions 38,153,391

Grouping: GPU Adapter / GPU Stack / Computing Task

GPU Adapter / GPU Stack / Computing Task	Computing Task Total Time	Average Time	Instance Count
GPU 3	56.714ms	1.091ms	52
GPU 5	48.038ms	0.924ms	52
GPU 1	57.120ms	1.098ms	52
GPU 0	58.191ms	1.119ms	52
GPU 2	74.159ms	1.426ms	52
GPU 4	54.251ms	1.043ms	52

VTUNE INSTRUCTION COUNT ANALYSIS

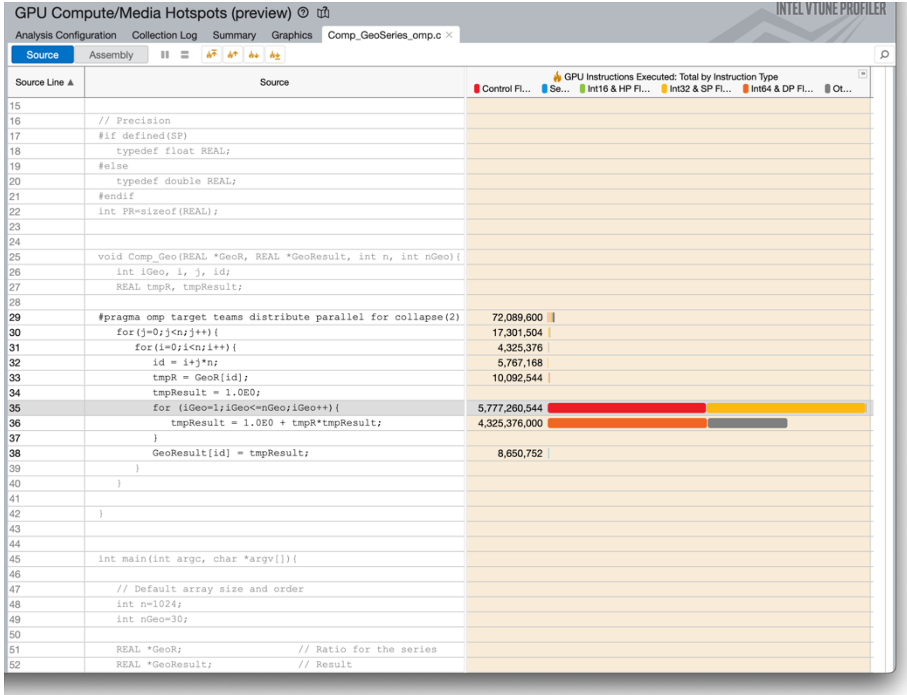
```
$ mpiexec -n 12 -ppn 12 gpu_tile_compact.sh vtune -collect gpu-hotspots -knob characterization-mode=instruction-count -r VTune_inst-count ./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```



Grouping: GPU Adapter / GPU Stack / Computing Task / Function / Call Stack

GPU Adapter / GPU Stack / Computing Task / Function / Call Stack	Work Size		Computing Task			
	Global ▼	Local	Total Time	Average Time	Instance Count	SIN
▼ GPU 3			3.424s	0.084s	41	
▶ GPU Stack 0			0.005s	0.000s	15	
▶ GPU Stack 1			3.418s	0.131s	26	
▼ GPU 5			6.878s	0.132s	52	
▼ GPU Stack 0			3.472s	0.134s	26	
▼ Comp_Geo$omp$offloading:29	2048 x 2...	64 x 1	3.466s	0.315s	11	
▼ Comp_Geo$omp$offloading:29						
▶ [No call stack information]						
▶ zeCommandListAppendMemoryCopy			0.001s	0.000s	14	
▶ zeCommandListAppendMemoryCopy			0.004s	0.004s	1	
▶ GPU Stack 1			3.406s	0.131s	26	
▶ GPU 1			6.817s	0.131s	52	

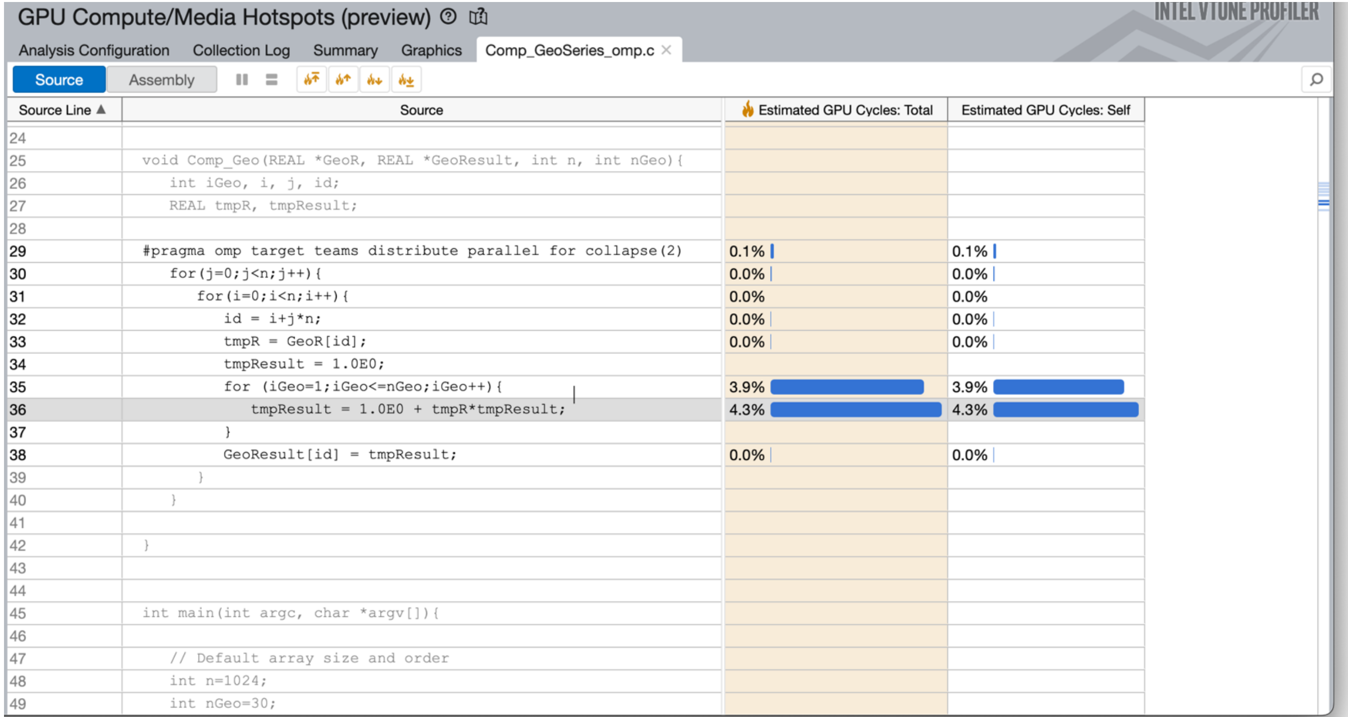
FILTER | Call Stack Mode: Only user functions | Inline Mode: Show inline functions



Source Line ▲	Source	GPU Instructions Executed: Total by Instruction Type Control Fl... Se... Int16 & HP Fl... Int32 & SP Fl... Int64 & DP Fl... Ot...
15		
16	`// Precision`	
17	`#if defined(SP)`	
18	`typedef float REAL;`	
19	`#else`	
20	`typedef double REAL;`	
21	`#endif`	
22	`int PR=sizeof(REAL);`	
23		
24		
25	`void Comp_Geo(REAL *GeoR, REAL *GeoResult, int n, int nGeo){`	
26	`int iGeo, i, j, id;`	
27	`REAL tmpR, tmpResult;`	
28		
29	`#pragma omp target teams distribute parallel for collapse(2)`	72,089,600
30	`for(j=0;j<n;j++){`	17,301,504
31	`for(i=0;i<n;i++){`	4,325,376
32	`id = i+j*n;`	5,767,168
33	`tmpR = GeoR[id];`	10,092,544
34	`tmpResult = 1.0E0;`	
35	`for (iGeo=1;iGeo<=nGeo;iGeo++){`	5,777,260,544
36	`tmpResult = 1.0E0 + tmpR*tmpResult;`	4,325,376,000
37	`}`	
38	`GeoResult[id] = tmpResult;`	8,650,752
39	`}`	
40	`}`	
41		
42	`}`	
43		
44		
45	`int main(int argc, char *argv[]){`	
46		
47	`// Default array size and order`	
48	`int n=1024;`	
49	`int nGeo=30;`	
50		
51	`REAL *GeoR;                // Ratio for the series`	
52	`REAL *GeoResult;           // Result`	

VTUNE SOURCE ANALYSIS

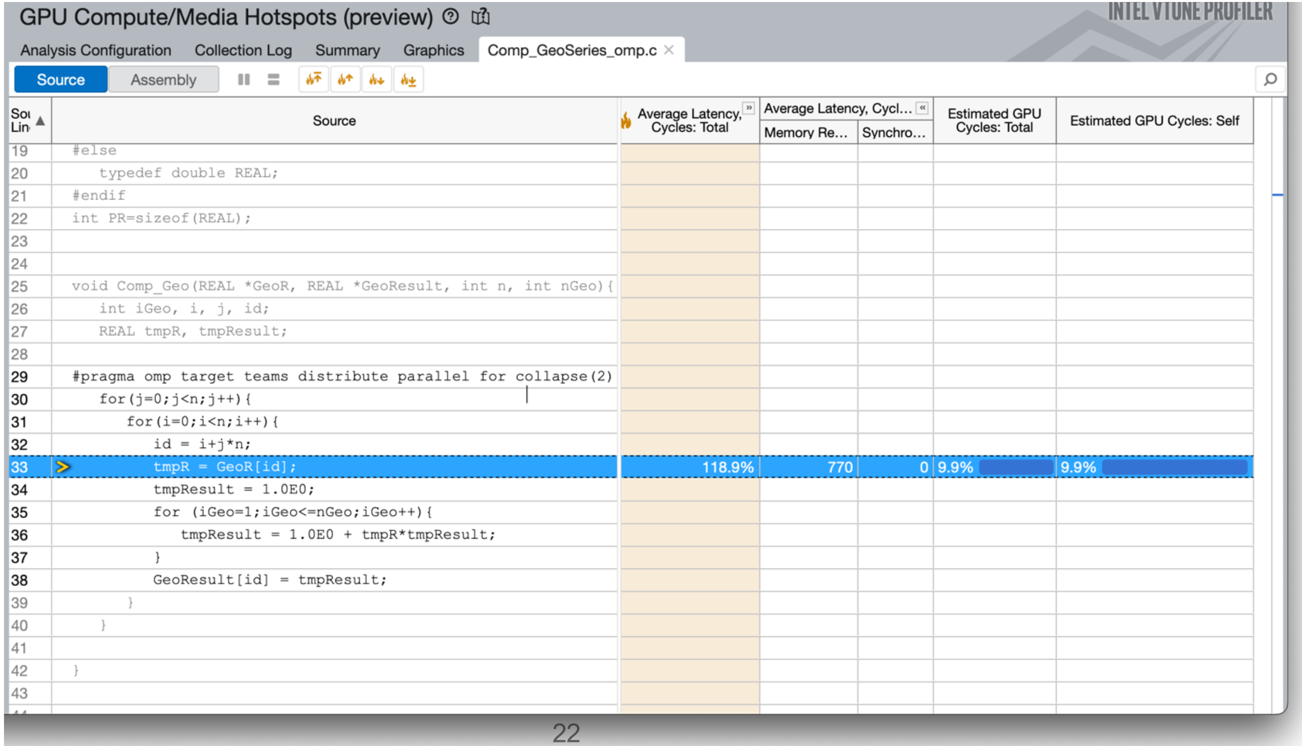
```
$ mpiexec -n 12 -ppn 12 gpu_tile_compact.sh vtune -collect gpu-hotspots -knob profiling-mode=source-analysis -r
VTune_source ./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```



Source Line	Source	Estimated GPU Cycles: Total	Estimated GPU Cycles: Self
24			
25	`void Comp_Geo(REAL *GeoR, REAL *GeoResult, int n, int nGeo){`		
26	`int iGeo, i, j, id;`		
27	`REAL tmpR, tmpResult;`		
28			
29	`#pragma omp target teams distribute parallel for collapse(2)`	0.1%	0.1%
30	`for(j=0;j<n;j++){`	0.0%	0.0%
31	`for(i=0;i<n;i++){`	0.0%	0.0%
32	`id = i+j*n;`	0.0%	0.0%
33	`tmpR = GeoR[id];`	0.0%	0.0%
34	`tmpResult = 1.0E0;`		
35	`for (iGeo=1;iGeo<=nGeo;iGeo++){`	3.9%	3.9%
36	`tmpResult = 1.0E0 + tmpR*tmpResult;`	4.3%	4.3%
37	`}`		
38	`GeoResult[id] = tmpResult;`	0.0%	0.0%
39	`}`		
40	`}`		
41			
42	`}`		
43			
44			
45	`int main(int argc, char *argv[]){`		
46			
47	`// Default array size and order`		
48	`int n=1024;`		
49	`int nGeo=30;`		

VTUNE MEMORY LATENCY ANALYSIS

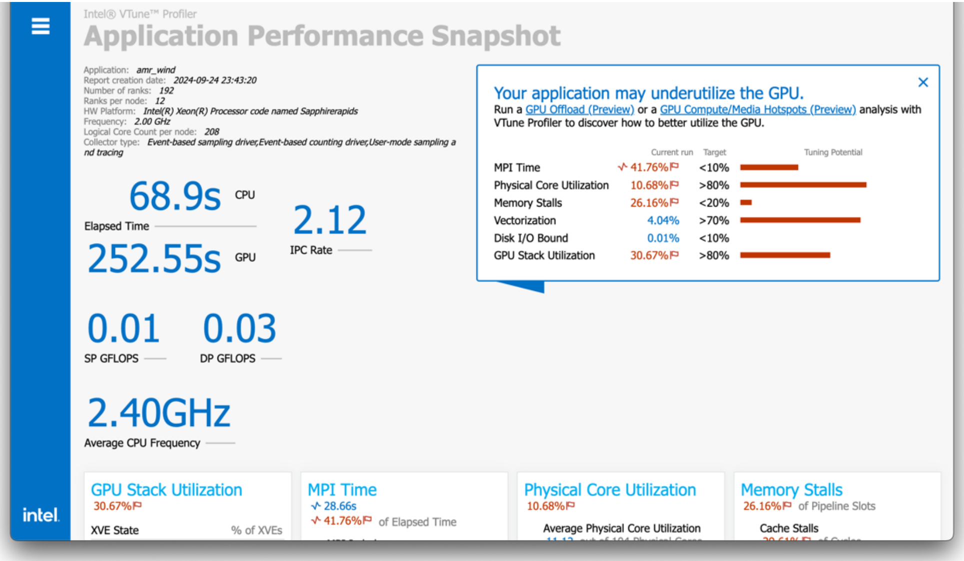
```
$ mpiexec -n 12 -ppn 12 gpu_tile_compact.sh vtune -collect gpu-hotspots -knob profiling-mode=source-analysis -knob source-analysis=mem-latency -r VTune_mem-latency ./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```



Sou Lin ▲	Source	Average Latency, Cycles: Total	Average Latency, Cycl…		Estimated GPU Cycles: Total	Estimated GPU Cycles: Self
			Memory Re…	Synchro…		
19	`#else`					
20	`typedef double REAL;`					
21	`#endif`					
22	`int PR=sizeof(REAL);`					
23						
24						
25	`void Comp_Geo(REAL *GeoR, REAL *GeoResult, int n, int nGeo){`					
26	`int iGeo, i, j, id;`					
27	`REAL tmpR, tmpResult;`					
28						
29	`#pragma omp target teams distribute parallel for collapse(2)`					
30	`for(j=0;j<n;j++){`					
31	`for(i=0;i<n;i++){`					
32	`id = i+j*n;`					
33	`tmpR = GeoR[id];`	118.9%	770	0	9.9%	9.9%
34	`tmpResult = 1.0E0;`					
35	`for (iGeo=1;iGeo<=nGeo;iGeo++){`					
36	`tmpResult = 1.0E0 + tmpR*tmpResult;`					
37	`}`					
38	`GeoResult[id] = tmpResult;`					
39	`}`					
40	`}`					
41						
42	`}`					
43						

APS ANALYSIS

```
$ mpiexec -n 192 gpu_tile_compact.sh aps -r APS_NMPI192 ./amr_wind
../test/test_files/abl_godunov/abl_godunov.inp
```



Intel® VTune™ Profiler

Application Performance Snapshot

Application: *amr_wind*
Report creation date: *2024-09-24 23:43:20*
Number of ranks: *192*
Ranks per node: *12*
HW Platform: *Intel(R) Xeon(R) Processor code named Sapphirerapids*
Frequency: *2.00 GHz*
Logical Core Count per node: *208*
Collector type: *Event-based sampling driver,Event-based counting driver,User-mode sampling and tracing*

68.9s CPU
252.55s GPU
Elapsed Time

2.12
IPC Rate

0.01
SP GFLOPS

0.03
DP GFLOPS

2.40GHz
Average CPU Frequency

Your application may underutilize the GPU.
Run a GPU Offload (Preview) or a GPU Compute/Media Hotspots (Preview) analysis with VTune Profiler to discover how to better utilize the GPU.

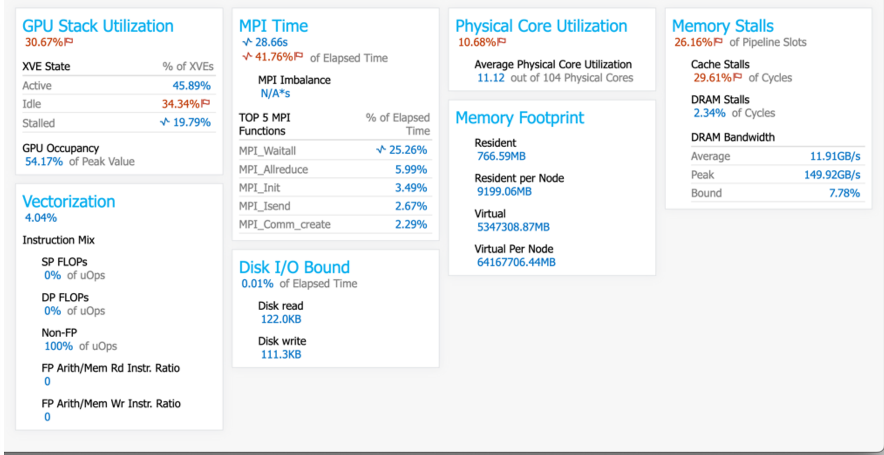
	Current run	Target	Tuning Potential
MPI Time	41.76%	<10%	
Physical Core Utilization	10.68%	>80%	
Memory Stalls	26.16%	<20%	
Vectorization	4.04%	>70%	
Disk I/O Bound	0.01%	<10%	
GPU Stack Utilization	30.67%	>80%	

GPU Stack Utilization
30.67%
XVE State % of XVEs

MPI Time
28.66s
41.76% of Elapsed Time

Physical Core Utilization
10.68%
Average Physical Core Utilization

Memory Stalls
26.16% of Pipeline Slots
Cache Stalls



GPU Stack Utilization
30.67%

XVE State	% of XVEs
Active	45.89%
Idle	34.34%
Stalled	19.79%

GPU Occupancy
54.17% of Peak Value

Vectorization
4.04%

Instruction Mix

SP FLOPs
0% of uOps

DP FLOPs
0% of uOps

Non-FP
100% of uOps

FP Arith/Mem Rd Instr. Ratio
0

FP Arith/Mem Wr Instr. Ratio
0

MPI Time
28.66s
41.76% of Elapsed Time

MPI Imbalance
N/A*s

TOP 5 MPI Functions	% of Elapsed Time
MPI_Waitall	25.26%
MPI_Allreduce	5.99%
MPI_Init	3.49%
MPI_Isend	2.67%
MPI_Comm_create	2.29%

Disk I/O Bound
0.01% of Elapsed Time

Disk read
122.0KB

Disk write
111.3KB

Physical Core Utilization
10.68%

Average Physical Core Utilization
11.12 out of 104 Physical Cores

Memory Footprint

Resident
766.59MB

Resident per Node
9199.06MB

Virtual
5347308.87MB

Virtual Per Node
64167706.44MB

Memory Stalls
26.16% of Pipeline Slots

Cache Stalls
29.61% of Cycles

DRAM Stalls
2.34% of Cycles

DRAM Bandwidth

Average	11.91GB/s
Peak	149.92GB/s
Bound	7.78%

```
$ aps-report --metrics=? APS_NMPI192_MaxGrid256_MaxStep13_jobid810828
| Available Metrics:
|--------------------
Elapsed Time
MPI Time
MPI Time
MPI Imbalance
MPI Hotspot 1 - MPI_Waitall
MPI Hotspot 1 - MPI_Waitall
MPI Hotspot 2 - MPI_Allreduce
MPI Hotspot 2 - MPI_Allreduce
MPI Hotspot 3 - MPI_Init
MPI Hotspot 3 - MPI_Init
MPI Hotspot 4 - MPI_Isend
MPI Hotspot 4 - MPI_Isend
MPI Hotspot 5 - MPI_Comm_create
MPI Hotspot 5 - MPI_Comm_create
Disk I/O Bound
Disk I/O Bound
Disk read
Disk write
Resident Memory Usage per Rank
Resident Memory Usage per Node
Virtual Memory Usage per Rank
Virtual Memory Usage per Node
Instructions Per Cycle Rate
Average CPU Frequency
Physical Core Utilization
Average Physical Core Utilization
Memory Stalls
Cache Stalls
DRAM Stalls
Average DRAM Bandwidth
DRAM Bandwidth Peak
DRAM Bandwidth Average
DRAM Bandwidth Bound
SP GFLOPS
DP GFLOPS
Vectorization
SP FLOPs
DP FLOPs
Non-FP
FP Arith/Mem Rd Instr. Ratio
FP Arith/Mem Wr Instr. Ratio
GPU Accumulated Time
GPU Stack Utilization
XVE State: Active
XVE State: Stalled
XVE State: Idle
GPU Occupancy
GPU Inbound PCIe Read
GPU Inbound PCIe Write
GPU Outbound PCIe Read
GPU Outbound PCIe Write
Network Controller Inbound PCIe Read
Network Controller Inbound PCIe Write
Network Controller Outbound PCIe Read
Network Controller Outbound PCIe Write
Inbound PCIe Read Per Device
Inbound PCIe Write Per Device
Outbound PCIe Read Per Device
Outbound PCIe Write Per Device
GPU Accumulated Time Per Device
GPU Stack Utilization Per Device
XVE State: Active Per Device
XVE State: Stalled Per Device
XVE State: Idle Per Device
GPU Occupancy Per Device
```

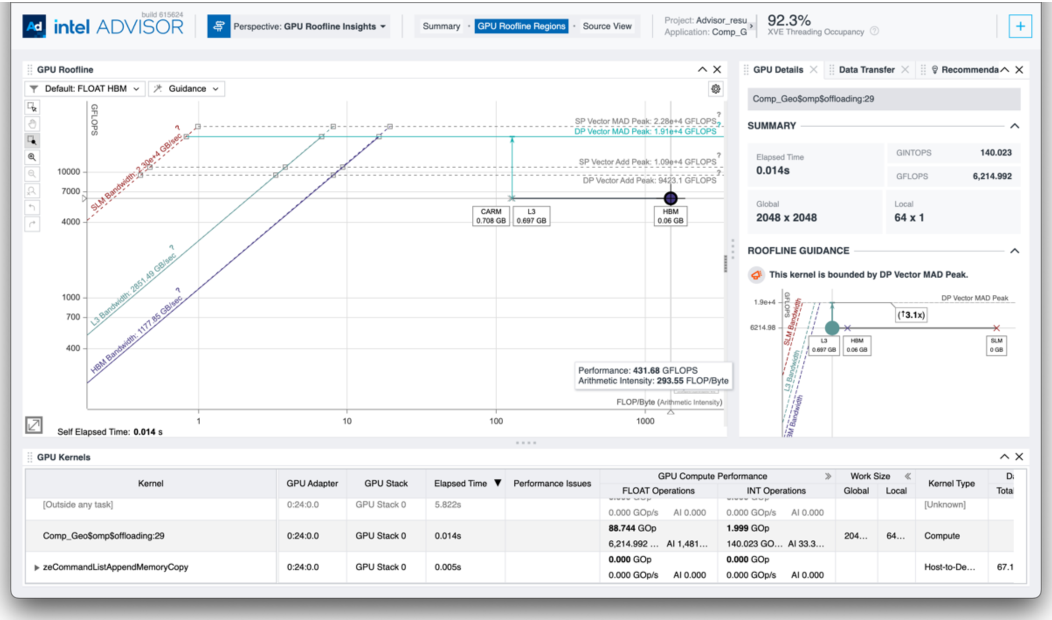
```
jkwack@aurora-uan-0009:/lus/flare/projects/Aurora_deployment/jkwack/ExaWind_flare/amr-wind/JK_build_2024.07.30.002_w_mpi> aps-report
APS_NMPI192_MaxGrid256_MaxStep13_jobid810828 --metrics="MPI Time, GPU Occupancy"

| Metric Table%

|-----------------------------------------------------------------------------------------

Metric Name                                  Node Name           Rank   Metric Value   Outlier Type

MPI Time, s                              x4305c3s2b0n0            135        30.0779           None

MPI Time, s                              x4305c3s3b0n0             72         30.037           None

MPI Time, s                              x4305c4s3b0n0             79        30.0068           None

MPI Time, s                              x4305c3s2b0n0             71         30.004           None

MPI Time, s                              x4305c4s1b0n0            109        29.9483           None

MPI Time, s                              x4305c2s4b0n0            130         29.946           None

MPI Time, s                              x4305c3s2b0n0            103        29.9379           None

MPI Time, s                              x4305c3s2b0n0            167        29.9315           None

MPI Time, s                              x4305c2s7b0n0             53        29.9133           None

MPI Time, s                              x4305c3s3b0n0            136        29.9061           None

MPI Time, s                              x4305c3s1b0n0             70         29.901           None

……

GPU Occupancy, % of Peak Value           x4305c4s2b0n0            N/A           54.4           None

GPU Occupancy, % of Peak Value           x4305c4s1b0n0            N/A           54.4           None

GPU Occupancy, % of Peak Value           x4305c4s0b0n0            N/A           54.4           None

GPU Occupancy, % of Peak Value           x4305c3s6b0n0            N/A           54.4           None

GPU Occupancy, % of Peak Value           x4305c4s3b0n0            N/A           54.3           None

GPU Occupancy, % of Peak Value           x4305c3s3b0n0            N/A           54.3           None

GPU Occupancy, % of Peak Value           x4305c3s5b0n0            N/A           54.2           None
```

ADVISOR ROOFLINE ANALYSIS

```
$ mpiexec -n 1 -ppn 12 gpu_tile_compact.sh advisor --collect=survey --profile-gpu --project-dir=Advisor_results --
./Comp_GeoSeries_omp_mpicxx_DP 2048 1000 : -n 11 -ppn 12 gpu_tile_compact.sh ./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```

```
$ mpiexec -n 1 -ppn 12 gpu_tile_compact.sh advisor --collect=tripcounts --profile-gpu --flop --no-trip-counts --project-
dir=Advisor_results -- ./Comp_GeoSeries_omp_mpicxx_DP 2048 1000 : -n 11 -ppn 12 gpu_tile_compact.sh
./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```

```
$ advisor --report=all --project-dir=Advisor_results --report-output=Advisor_results/roofline_all.html
```



Project: Advisor_resu
Application: Comp_G

Performance: 431.68 GFLOPS
Arithmetic Intensity: 293.55 FLOP/Byte

Self Elapsed Time: 0.014 s

Comp_Geo$omp$offloading:29

SUMMARY

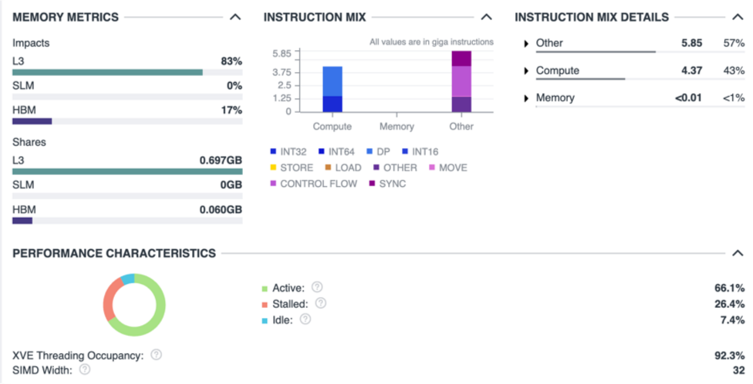
Elapsed Time 0.014s	GINTOPS 140.023 GFLOPS 6,214.992
Global 2048 x 2048	Local 64 x 1

ROOFLINE GUIDANCE

This kernel is bounded by DP Vector MAD Peak.

GPU Kernels

Kernel	GPU Adapter	GPU Stack	Elapsed Time ▼	Performance Issues	GPU Compute Performance		Work Size		Kernel Type	D
					FLOAT Operations	INT Operations	Global	Local		Tota
[Outside any task]	0:24:0.0	GPU Stack 0	5.822s		0.000 GOp/s AI 0.000	0.000 GOp/s AI 0.000			[Unknown]	
Comp_Geo$omp$offloading:29	0:24:0.0	GPU Stack 0	0.014s		88.744 GOp 6,214.992 ... AI 1,481...	1.999 GOp 140.023 GO... AI 33.3...	204...	64...	Compute	
▸ zeCommandListAppendMemoryCopy	0:24:0.0	GPU Stack 0	0.005s		0.000 GOp 0.000 GOp/s AI 0.000	0.000 GOp 0.000 GOp/s AI 0.000			Host-to-De...	67.1



MEMORY METRICS

Impacts

L3	83%
SLM	0%
HBM	17%

Shares

L3	0.697GB
SLM	0GB
HBM	0.060GB

INSTRUCTION MIX

All values are in giga instructions

INSTRUCTION MIX DETAILS

▸ Other	5.85	57%
▸ Compute	4.37	43%
▸ Memory	<0.01	<1%

PERFORMANCE CHARACTERISTICS

Active:	66.1%
Stalled:	26.4%
Idle:	7.4%
XVE Threading Occupancy:	92.3%
SIMD Width:	32

THAPI / IPROF ON AURORA

A lightweight tool for tracing

```
$ module load thapi
$ mpirun -n 12 -ppn 12 gpu_tile_compact.sh iprof ./Comp_GeoSeries_omp_mpicxx_DP 2048 1000
```

```
THAPI: Trace location: /home/jkwack/thapi-traces/thapi_aggreg--2024-09-25T06:27:15-05:00
BACKEND_MPI | 1 Hostnames | 12 Processes | 12 Threads |

         Name |      Time | Time(%) | Calls |  Average |      Min |     Max |
     MPI_Init |    16.76s |  98.13% |    12 |    1.40s | 434.72ms |   2.48s |
 MPI_Finalize |  219.50ms |   1.29% |    12 |  18.29ms |  17.48ms | 22.41ms |
   MPI_Reduce |   99.94ms |   0.59% |    96 |   1.04ms |    908ns | 19.77ms |
MPI_Comm_rank |   13.49us |   0.00% |    12 |   1.12us |    504ns |  3.27us |
MPI_Comm_size |    7.60us |   0.00% |    12 | 633.67ns |    485ns |   733ns |
        Total |    17.08s | 100.00% |   144 |
……
```

```
BACKEND_OMP | 1 Hostnames | 12 Processes | 12 Threads |

                               Name |     Time | Time(%) | Calls | Average |    Min |    Max |
                ompt_target_exit_data | 468.50ms |  39.62% |    12 | 39.04ms | 31.71ms | 45.23ms |
ompt_target_data_transfer_from_device | 460.39ms |  38.93% |    12 | 38.37ms | 31.12ms | 44.28ms |
                          ompt_target | 191.37ms |  16.18% |   132 |  1.45ms |  1.30ms |  9.28ms |
               ompt_target_enter_data |  30.59ms |   2.59% |    12 |  2.55ms |  1.84ms |  3.58ms |
  ompt_target_data_transfer_to_device |  20.24ms |   1.71% |    12 |  1.69ms |  1.19ms |  2.46ms |
               ompt_target_submit_emi |   9.36ms |   0.79% |   132 | 70.90us |  7.16us |  1.13ms |
               ompt_target_data_alloc |   1.33ms |   0.11% |    24 | 55.28us | 24.86us | 93.15us |
              ompt_target_data_delete | 775.82us |   0.07% |    24 | 32.33us |  5.08us | 67.70us |
                                Total |    1.18s | 100.00% |   360 |

BACKEND_ZE | 1 Hostnames | 12 Processes | 12 Threads |

                               Name |     Time | Time(%) | Calls |  Average |      Min |      Max | Error |
                       zeModuleCreate |    2.72s |  77.65% |   132 |  20.61ms | 103.89us | 224.80ms |     0 |
        zeCommandListAppendMemoryCopy | 483.38ms |  13.80% |   180 |   2.69ms |   9.00us |  44.28ms |     0 |
              zeEventHostSynchronize | 197.31ms |   5.63% |   312 | 632.41us |    108ns |   9.25ms |     0 |
                        zeEventCreate |  28.86ms |   0.82% | 49920 | 578.08ns |    223ns | 146.23us |     0 |
         zeCommandListCreateImmediate |  22.23ms |   0.63% |    24 | 926.29us |  55.87us |   2.90ms |     0 |
                      zeModuleDestroy |  10.31ms |   0.29% |   132 |  78.08us |   5.86us | 478.02us |     0 |
                       zeEventDestroy |   9.04ms |   0.26% | 49920 | 181.14ns |    108ns |  23.10us |     0 |
          zeContextMakeMemoryResident |   7.99ms |   0.23% |    84 |  95.15us |   5.18us | 610.76us |     0 |
       zeCommandListAppendLaunchKernel |   7.28ms |   0.21% |   132 |  55.12us |   6.57us | 636.24us |     0 |
                 zeCommandQueueCreate |   3.23ms |   0.09% |    12 | 268.84us | 232.46us | 299.16us |     0 |
                     zeMemAllocDevice |   2.55ms |   0.07% |    84 |  30.41us |  13.05us |  69.45us |     0 |
zeDriverGetExtensionFunctionAddress |   2.00ms |   0.06% |   132 |  16.69us |    289ns | 233.68us |    12 |
                       zeKernelCreate |   1.83ms |   0.05% |  1752 |   1.04us |    684ns |  12.13us |     0 |
……
```

```
Device profiling | 1 Hostnames | 12 Processes | 12 Threads | 12 Devices | 12 Subdevices |

                              Name |     Time | Time(%) | Calls |  Average |    Min |     Max |
zeCommandListAppendMemoryCopy(D2M) | 203.59ms |  51.65% |    12 |  16.97ms | 6.03ms | 30.29ms |
                      Comp_Geo_l29 | 172.75ms |  43.83% |   132 |   1.31ms | 1.29ms |  1.40ms |
zeCommandListAppendMemoryCopy(M2D) |  17.58ms |   4.46% |    96 | 183.11us |   80ns |  2.10ms |
zeCommandListAppendMemoryCopy(S2M) | 217.76us |   0.06% |    48 |   4.54us | 1.28us | 16.48us |
zeCommandListAppendMemoryCopy(M2M) |  24.40us |   0.01% |    12 |   2.03us | 1.36us |  2.80us |
zeCommandListAppendMemoryCopy(M2S) |    960ns |   0.00% |    12 |  80.00ns |   80ns |    80ns |
                             Total | 394.16ms | 100.00% |   312 |

Explicit memory traffic (BACKEND_MPI) | 1 Hostnames | 12 Processes | 12 Threads |

      Name | Byte | Byte(%) | Calls | Average | Min | Max |
MPI_Reduce | 768B | 100.00% |    96 |   8.00B |  8B |  8B |
     Total | 768B | 100.00% |    96 |

Explicit memory traffic (BACKEND_OMP) | 1 Hostnames | 12 Processes | 12 Threads |

                                 Name |     Byte | Byte(%) | Calls | Average |     Min |     Max |
               ompt_target_data_alloc | 805.31MB |  50.00% |    24 | 33.55MB | 33.55MB | 33.55MB |
  ompt_target_data_transfer_to_device | 402.65MB |  25.00% |    12 | 33.55MB | 33.55MB | 33.55MB |
ompt_target_data_transfer_from_device | 402.65MB |  25.00% |    12 | 33.55MB | 33.55MB | 33.55MB |
              ompt_target_data_delete |       0B |   0.00% |    24 |   0.00B |      0B |      0B |
                                Total |   1.61GB | 100.00% |    72 |

……
```

```
Explicit memory traffic (BACKEND_ZE) | 1 Hostnames | 12 Processes | 12 Threads |

                              Name |     Byte | Byte(%) | Calls | Average |     Min |     Max |
       zeContextMakeMemoryResident | 845.41MB |  51.21% |    84 | 10.06MB |      8B | 33.55MB |
zeCommandListAppendMemoryCopy(M2D) | 402.71MB |  24.40% |    96 |  4.19MB |      4B | 33.55MB |
zeCommandListAppendMemoryCopy(D2M) | 402.65MB |  24.39% |    12 | 33.55MB | 33.55MB | 33.55MB |
zeCommandListAppendMemoryCopy(S2M) |   2.40kB |   0.00% |    48 |  50.00B |      8B |    144B |
zeCommandListAppendMemoryCopy(M2S) |     768B |   0.00% |    12 |  64.00B |     64B |     64B |
zeCommandListAppendMemoryCopy(M2M) |     684B |   0.00% |    12 |  57.00B |     57B |     57B |
                             Total |   1.65GB | 100.00% |   264 |
```

KNOWN ISSUES

w/ the latest SDK (oneapi/eng-compiler/2024.07.30.002)

- Issue 1:
  - Symptom: got the following error message at the end
    - Exception: 0xb, Segmentation&nbsp;fault
    - Module: libswip.so
  - Workaround: Before running Advisor, VTune, and APS, do the following:
    - export SWIP_NULL_SOCKET=1
  - W/ 2025.0 version, the issue will be gone

- Issue 2:
  - Symptom: got the following error message after finishing collection
    - vtune: Error: Cannot stop collection of GPU events
  - Workaround: it is a false alarm. You can ignore it.
  - Intel is investigating this issue now.

- Any other issues: please send an email to support@alcf.anl.gov or jkwack@anl.gov (JaeHyuk Kwack)

THANKS!