oneAPI

LRZ Workshop – 10.11.2022

Application Profiling with Intel® VTune™ and PTI-GPU

Heinrich Bockhorst

intel®

Agenda

- PTI-GPU tools
- onetrace tool from PTI-GPU
- VTune overview
- Command lines – Playbook
- VTune GPU analysis using GROMACS
- Documentation
- Demo – Hands on

Profiling Tools Interfaces for GPU (PTI for GPU)

- Light weight Open Source tools for GPU profiling (MIT license)
- Github initiated by Intel® engineers but with contributions from community: https://github.com/intel/pti-gpu
- Profiling for Intel® GPUs using OpenCL and Level Zero runtime
- OpenCL*, SYCL*, and OpenMP* offloading supported
- Snapshots taken from Intel® DevCloud using Intel® Iris® Xᵉ GPUs

sysmon

- Linux static and dynamic information about GPU activity
- Can be used like Linux „top“

```
u48655@s011-n001: ~
Every 1.0s: /data/comp/sysmon                                   s011-n001: Fri May 20 11:23:16 2022

================================================================================
GPU 0: Intel(R) Iris(R) Xe MAX Graphics [0x4905]    PCI Bus: 0000:69:00.0
Vendor: Intel(R) Corporation    Driver Version: 1.2.21786    Subdevices: 0
EU Count: 96    Threads Per EU: 7    EU SIMD Width: 8    Total Memory(MB): 7714.0
Core Frequency(MHz): 1550.0 of 1550.0    Core Temperature(C): unknown
================================================================================
Running Processes: 2
     PID,  Device Memory Used(MB),  Shared Memory Used(MB),  GPU Engines, Executable
  221409,                  15.9,                     0.0,       3D;DMA, /home/u48655/Software/GROMACS/2022/gromacs-master-22.02.03/oneapi/2022.1.2SIMD8USM/bin/gmx
  221449,                   0.2,                     0.0,      UNKNOWN, /data/comp/sysmon
================================================================================
GPU 1: Intel(R) Iris(R) Xe MAX Graphics [0x4905]    PCI Bus: 0000:1b:00.0
Vendor: Intel(R) Corporation    Driver Version: 1.2.21786    Subdevices: 0
EU Count: 96    Threads Per EU: 7    EU SIMD Width: 8    Total Memory(MB): 7714.0
Core Frequency(MHz): 300.0 of 1550.0    Core Temperature(C): unknown
================================================================================
Running Processes: 1
     PID,  Device Memory Used(MB),  Shared Memory Used(MB),  GPU Engines, Executable
  221449,                   0.2,                     0.0,      UNKNOWN, /data/comp/sysmon
```

onetrace

- Profile OpenCL* and Level Zero backend.
- For OpenCL, SYCL* and OpenMP* offloading applications.

usage: onetrace [options] <application> <args>

Options:

--host-timing [-h]	host API calls statistics
--device-timing [-d]	kernel execution timing
--chrome-kernel-timeline	generates trace for chrome://tracing

...

onetrace host statistics (GROMACS SYCL)

- $ onetrace –h gmx <gmx args>

/cygdrive/c/Users/hbockhor/OneDrive - Intel Corporation/Training/2022/22.05.30_ISC_MTA/RESULTS_onetrace

```
=== API Timing Results: ===

            Total Execution Time (ns):          8849359641
    Total API Time for L0 backend (ns):          5083012588

== L0 Backend: ==

                          Function,       Calls,          Time (ns),  Time (%),       Average (ns),          Min (ns),          Max (ns)
             zeCommandQueueSynchronize,        2009,        3697535106,     72.74,         1840485,             164,            7145738
                      zeModuleCreate,           7,        1030797983,     20.28,       147256854,        83120536,          227200191
    zeCommandQueueExecuteCommandLists,        4226,         146236830,      2.88,           34604,           14481,             104637
        zeCommandListAppendMemoryCopy,        3172,          87847894,      1.73,           27694,           11550,             840112
                  zeCommandListReset,         4226,          37004797,      0.73,            8756,             554,              36072
                 zeCommandListCreate,           8,          27893013,      0.55,         3486626,          226640,           13670348
      zeCommandListAppendLaunchKernel,        2811,          13505254,      0.27,            4804,             926,              17791
            zeKernelSetArgumentValue,        35215,           9806280,      0.19,             278,              31,               9071
                     zeEventCreate,           5983,           6979341,      0.14,            1166,             364,              54293
                zeCommandListClose,           4226,           3455592,      0.07,             817,             140,              15018
                  zeMemAllocDevice,             11,           3054949,      0.06,          277722,           56756,            1191462
                     zeFenceReset,           4226,           3037219,      0.06,             718,             536,              24105
               zeKernelSetGroupSize,         2811,           2867434,      0.06,            1020,             141,               4410
```

onetrace device statistics (GROMACS SYCL)

- $ onetrace --demangle –d gmx <gmx args>

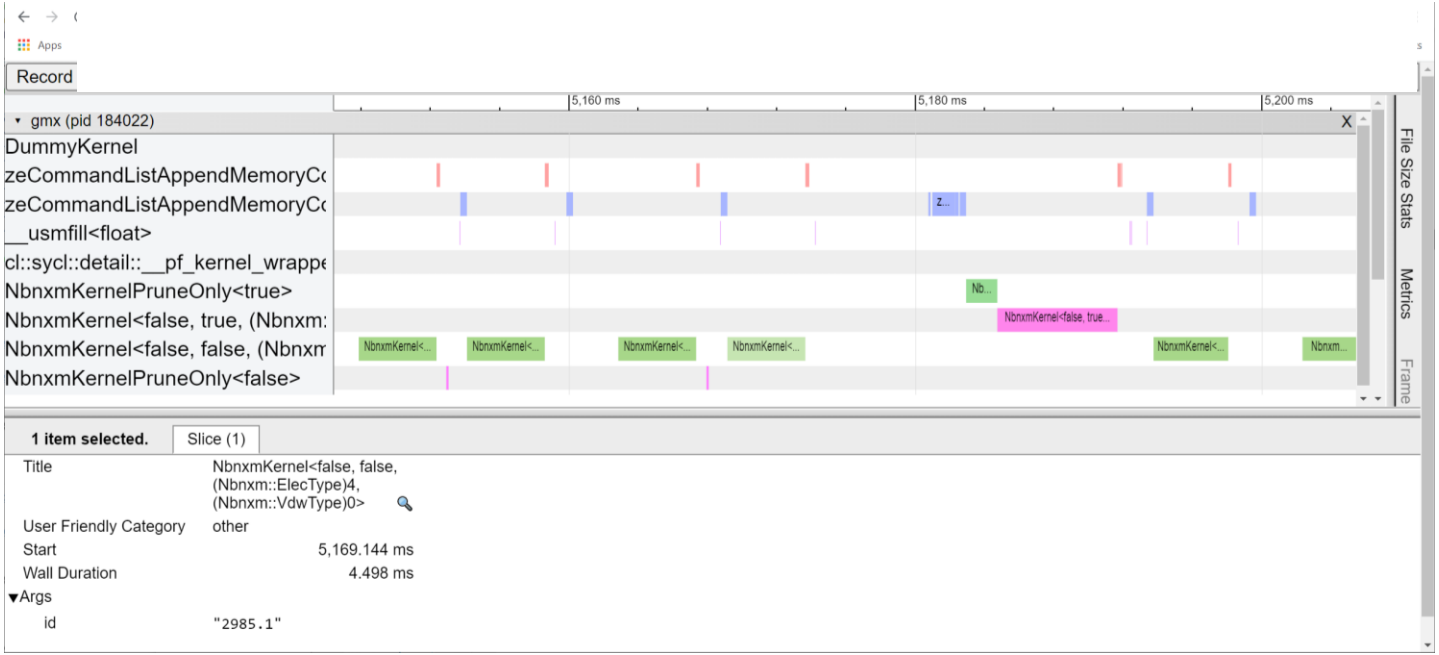
/cygdrive/c/Users/hbockhor/OneDrive - Intel Corporation/Training/2022/22.05.30_ISC_MTA/RESULTS_onetrace

```
=== Device Timing Results: ===

                Total Execution Time (ns):          9250282340
    Total Device Time for L0 backend (ns):          5144159294

== L0 Backend: ==

                                                    Kernel,       Calls,         Time (ns),    Time (%),       Average (ns),         Min (ns),         Max (ns)
NbnxmKernel<false, false, (Nbnxm::ElecType)4, (Nbnxm::VdwType)0>,         990,      4374811057,       85.04,         4419001,      4290416,      4946458
                                zeCommandListAppendMemoryCopy(M2D),        2047,       393601766,        7.65,          192282,         3229,      1283489
                                zeCommandListAppendMemoryCopy(D2M),        1125,       183165569,        3.56,          162813,          520,       190104
 NbnxmKernel<false, true, (Nbnxm::ElecType)4, (Nbnxm::VdwType)0>,          11,        77472702,        1.51,         7042972,      6985937,      7131562
                                        NbnxmKernelPruneOnly<false>,       490,        67430948,        1.31,          137614,       130416,       151979
                                                    __usmfill<float>,      1307,        27828716,        0.54,           21292,         1666,        42500
                                         NbnxmKernelPruneOnly<true>,          11,        19808433,        0.39,         1800766,      1798125,      1804687
       cl::sycl::detail::__pf_kernel_wrapper<__usmfill<float> >,            1,           35833,        0.00,           35833,        35833,        35833
                                                    DummyKernel,          1,            4270,        0.00,            4270,         4270,         4270
```

More information available by adding –v (verbose) option

onetrace kernel timeline

- $ onetrace --demangle --chrome-kernel-timeline <gmx> <args>



PTI-GPU advantages

- No aditional drivers necessary
- Low overhead < 3%
- Traces in json format to be loded by Google Chrome: chrome://tracing

VTune™ -- Short Overview

Optimize Performance

Intel® VTune™ Profiler

Get the Right Data to Find Bottlenecks

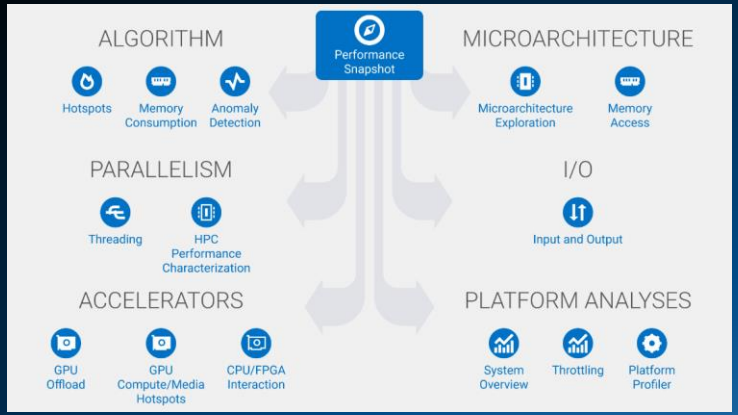
- A suite of profiling for CPU, GPU, FPGA, threading, memory, cache, storage, offload, power…
- DPC++, C, C++, Fortran, Python*, Go*, Java*, or a mix
- Linux, Windows, FreeBSD, Android, Yocto and more

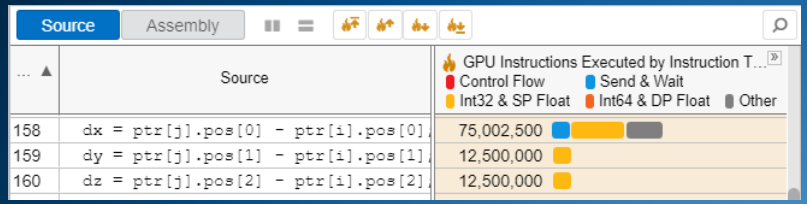
Analyze Data Faster

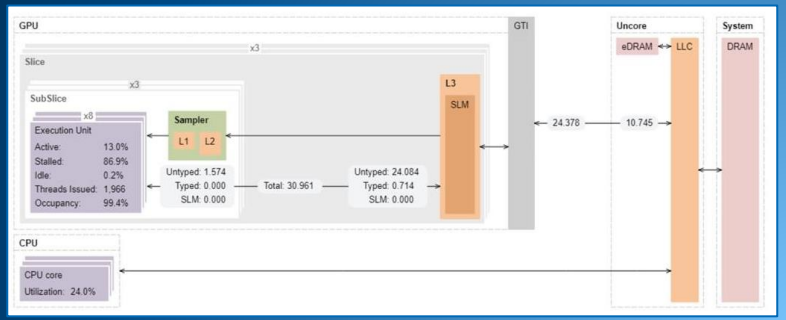
- See data on your source, in architecture diagrams, as a histogram, on a timeline…
- Filter and organize data to find answers

Work Your Way

- User interface or command line
- Profile locally and remotely
- Install as an application
- Install as a server accessible with a web browser







Find Answers Fast

Intel® VTune™ Profiler

Adjust Data Grouping

Function / Call Stack

Source Function / Function / Call Stack

Sync Object / Function / Call Stack

Sync Object / Thread / Function / Call Stack

... (Partial list shown)

Double Click Function to View Source

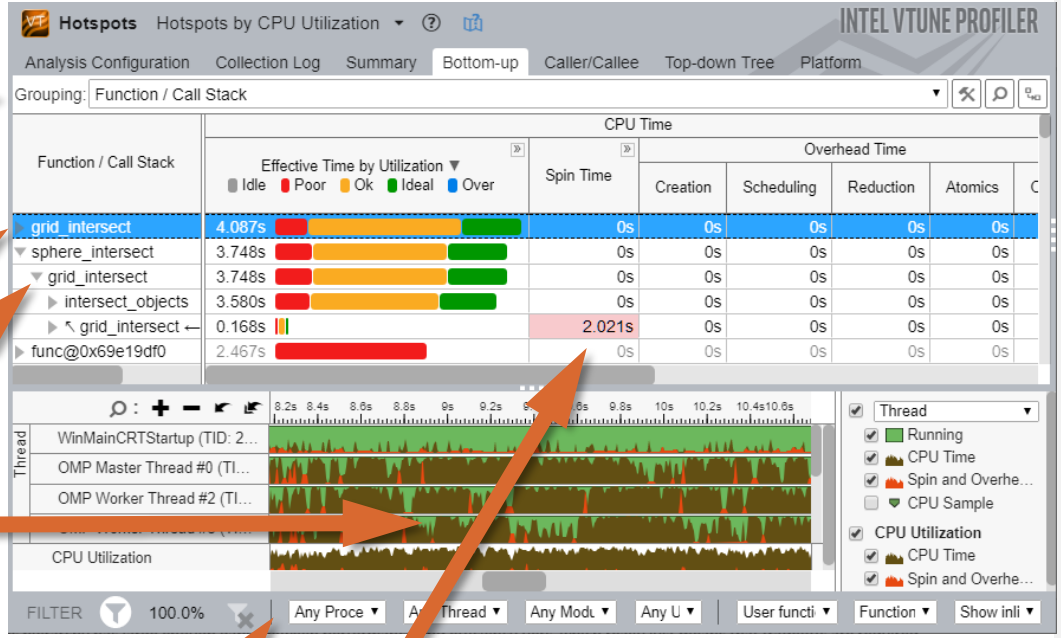
Click [▶] for Call Stack

Filter by Timeline Selection (or by Grid Selection)

Zoom In And Filter On Selection

Filter In by Selection

Remove All Filters

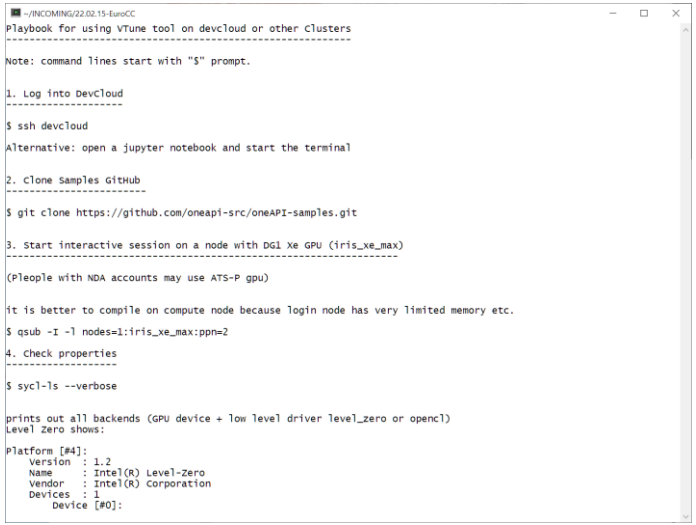


Filter by Process & Other Controls

Tuning Opportunities Shown in Pink. Hover for Tips

VTune Playbook – for this session

- Easy access to command lines
- Nbody Example for DevCloud – please try



Playbooks on DevCloud:
`/data/comp/workshop/Vtune_playbook.txt`

How to start an Analysis

- VTune offers different analysis types with additional “knobs”
- Application Performance Snapshot (VTune or standalone)
- APS does first analysis and guides to the right VTune analysis or a different tool available in the oneAPI toolkits.
- APS analyses HW counters, OpenMP and MPI
- APS can be used for MPI and/or OpenMP only to avoid too much data being collected

APS usage

- Help menu

```
$ aps –help
```

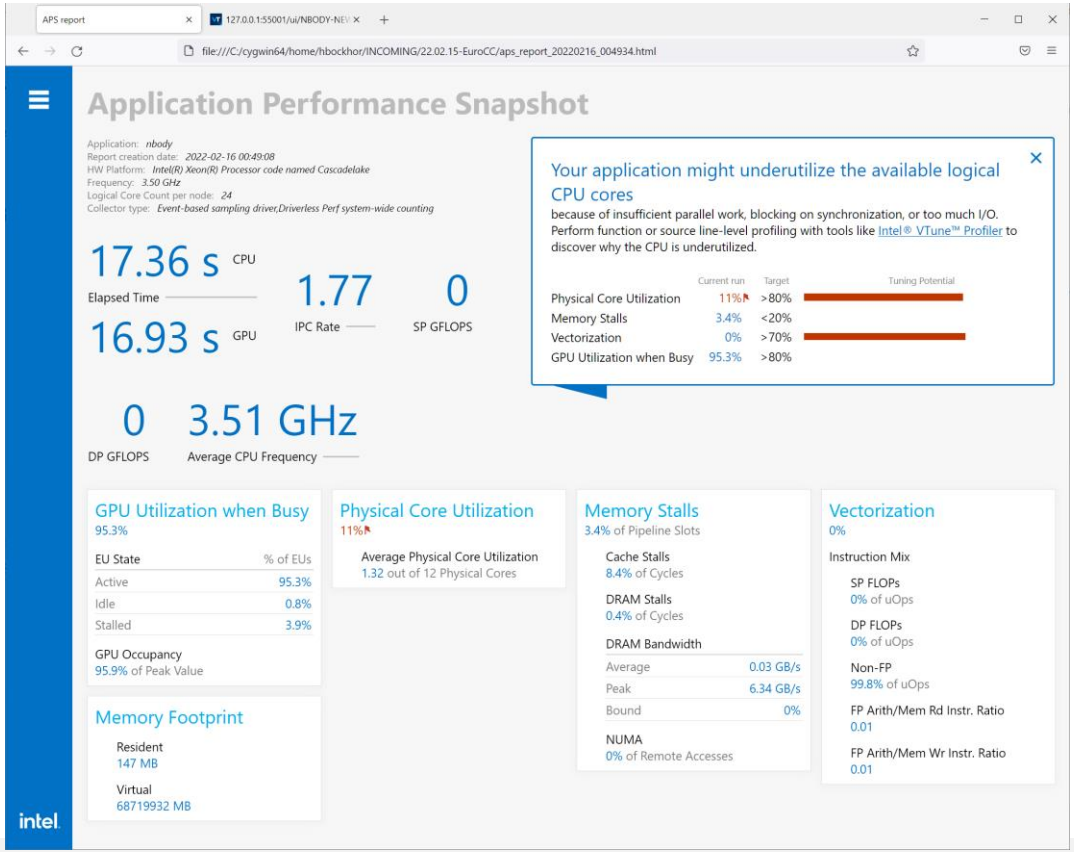
- Run with APS

```
$ aps <application> <app paramter>
```

- Run with MPI (Intel MPI or compatible)

```
$ mpirun –n <N> aps  -- <application> < app parameter>
```

APS HTML output (nbody)



Intel® VTune™ Profiler
GROMACS SYCL version

VTune GPU analysis

- Host API and GPU kernels
- Detailed source view on Kernels
- Bandwidth data with Hierarchical Memory Diagram
- Estimate for timing per source line (basic block timing)
- Estimate for memory latency per source line

SYCL version of GROMACS

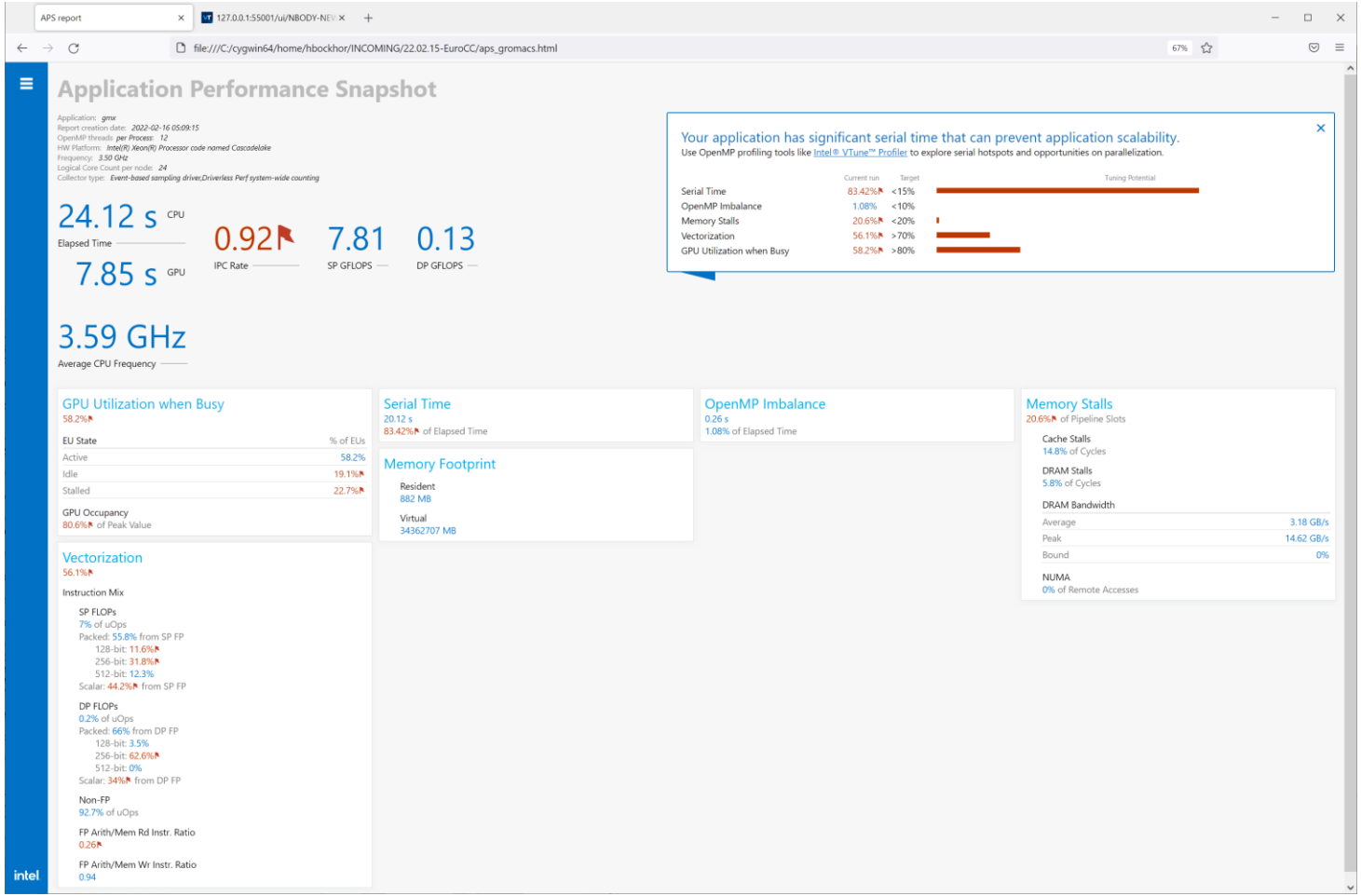
- Clone master branch:
  $ git clone https://gitlab.com/gromacs/gromacs.git

- Workload: https://www.mpinat.mpg.de/632209/benchMEM.zip

```
SRC=$HOME/gromacs
PRE=$HOME/GROMACS/

 cmake $SRC                      \
    -DCMAKE_INSTALL_PREFIX=$PRE  \
    -DGMX_OPENMP=ON              \
    -DGMX_OPENMP_MAX_THREADS=128\
    -DGMX_GPU=SYCL               \
    -DGMX_FFT_LIBRARY=mkl        \
    -DCMAKE_C_COMPILER=icx        \
    -DCMAKE_CXX_COMPILER=icpx
```

- Run:

```
${EXE_DIR}/gmx mdrun -ntmpi $RANKS -ntomp $OMP_NUM_THREADS -nb $MODE -pme cpu -s benchMEM.tpr  -nsteps $steps
```


APS report
127.0.0.1:55001/ui/NBODY-NEW
file:///C:/cygwin64/home/hbockhor/INCOMING/22.02.15-EuroCC/aps_gromacs.html
67%
Application Performance Snapshot
Application: gmx
Report creation date: 2022-02-16 05:09:15
OpenMP threads per Process: 12
HW Platform: Intel(R) Xeon(R) Processor code named Cascadelake
Frequency: 3.50 GHz
Logical Core Count per node: 24
Collector type: Event-based sampling driver,Driverless Perf system-wide counting
24.12 s CPU
Elapsed Time
7.85 s GPU
0.92
IPC Rate
7.81
SP GFLOPS
0.13
DP GFLOPS
3.59 GHz
Average CPU Frequency
Your application has significant serial time that can prevent application scalability.
Use OpenMP profiling tools like Intel® VTune™ Profiler to explore serial hotspots and opportunities on parallelization.
Current run
Target
Tuning Potential
Serial Time 83.42% <15%
OpenMP Imbalance 1.08% <10%
Memory Stalls 20.6% <20%
Vectorization 56.1% >70%
GPU Utilization when Busy 58.2% >80%
GPU Utilization when Busy
58.2%
EU State
% of EUs
Active 58.2%
Idle 19.1%
Stalled 22.7%
GPU Occupancy
80.6% of Peak Value
Serial Time
20.12 s
83.42% of Elapsed Time
OpenMP Imbalance
0.26 s
1.08% of Elapsed Time
Memory Stalls
20.6% of Pipeline Slots
Cache Stalls
14.8% of Cycles
DRAM Stalls
5.8% of Cycles
DRAM Bandwidth
Average 3.18 GB/s
Peak 14.62 GB/s
Bound 0%
NUMA
0% of Remote Accesses
Memory Footprint
Resident
882 MB
Virtual
34362707 MB
Vectorization
56.1%
Instruction Mix
SP FLOPs
7% of uOps
Packed: 55.8% from SP FP
128-bit: 11.6%
256-bit: 31.8%
512-bit: 12.3%
Scalar: 44.2% from SP FP
DP FLOPs
0.2% of uOps
Packed: 66% from DP FP
128-bit: 3.5%
256-bit: 62.6%
512-bit: 0%
Scalar: 34% from DP FP
Non-FP
92.7% of uOps
FP Arith/Mem Rd Instr. Ratio
0.26
FP Arith/Mem Wr Instr. Ratio
0.94
intel

Command lines used

- Basic gpu analysis:

```
vtune –collect gpu-hotspots –r <your-result-dir> -- <executable><args>
```

- Full instrumentation:

```
vtune -c gpu-hotspots -knob characterization-mode=instruction-count \
                          –r <your-result-dir> -- <executable><args>
```

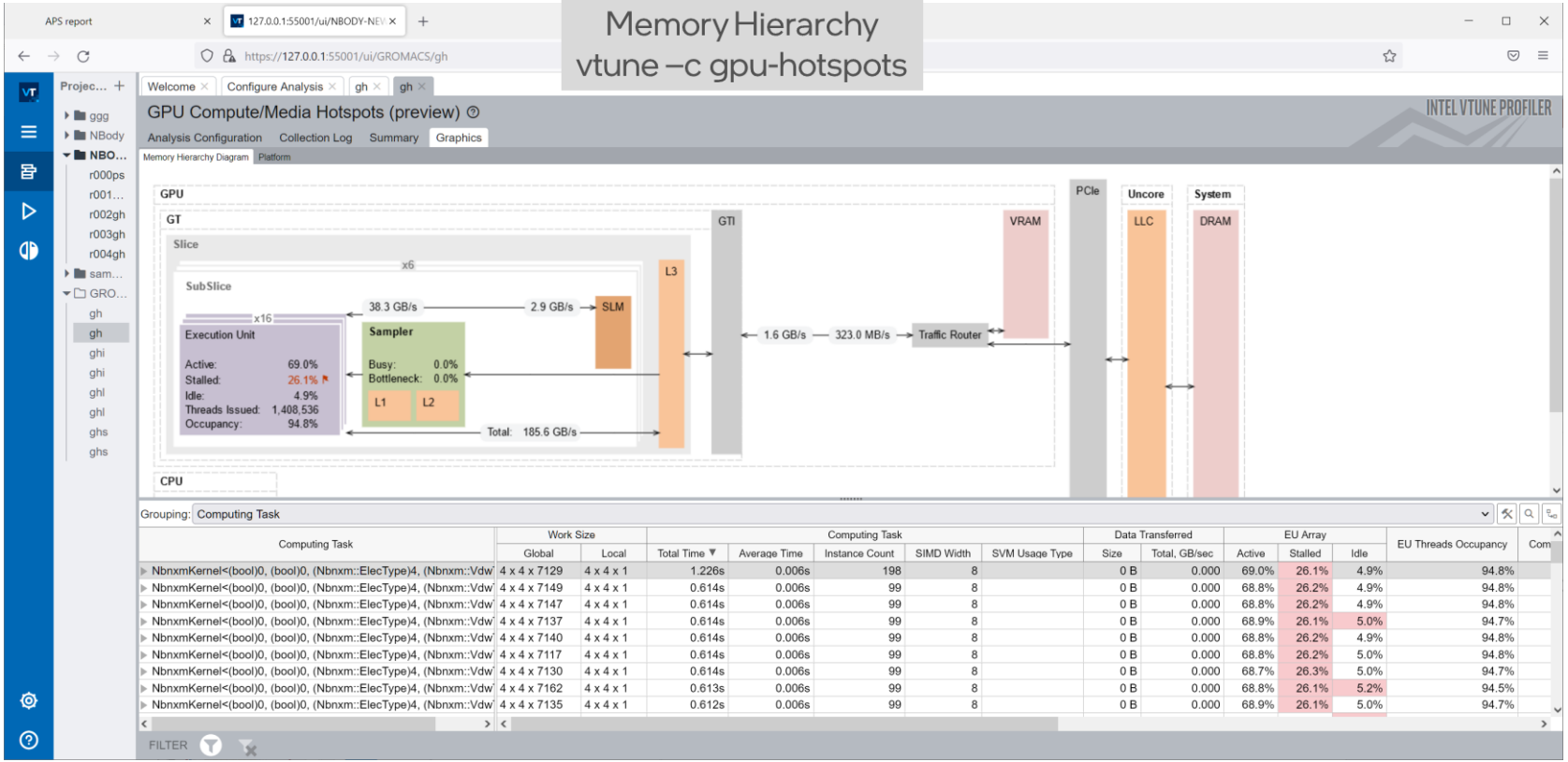
- Source Instrumentation with timing of basic blocks:

```
vtune -c gpu-hotspots -knob profiling-mode=source-analysis\
                           –r <your-result-dir> -- <executable><args>
```

- Source Instrumentation with only memory inst. timed:

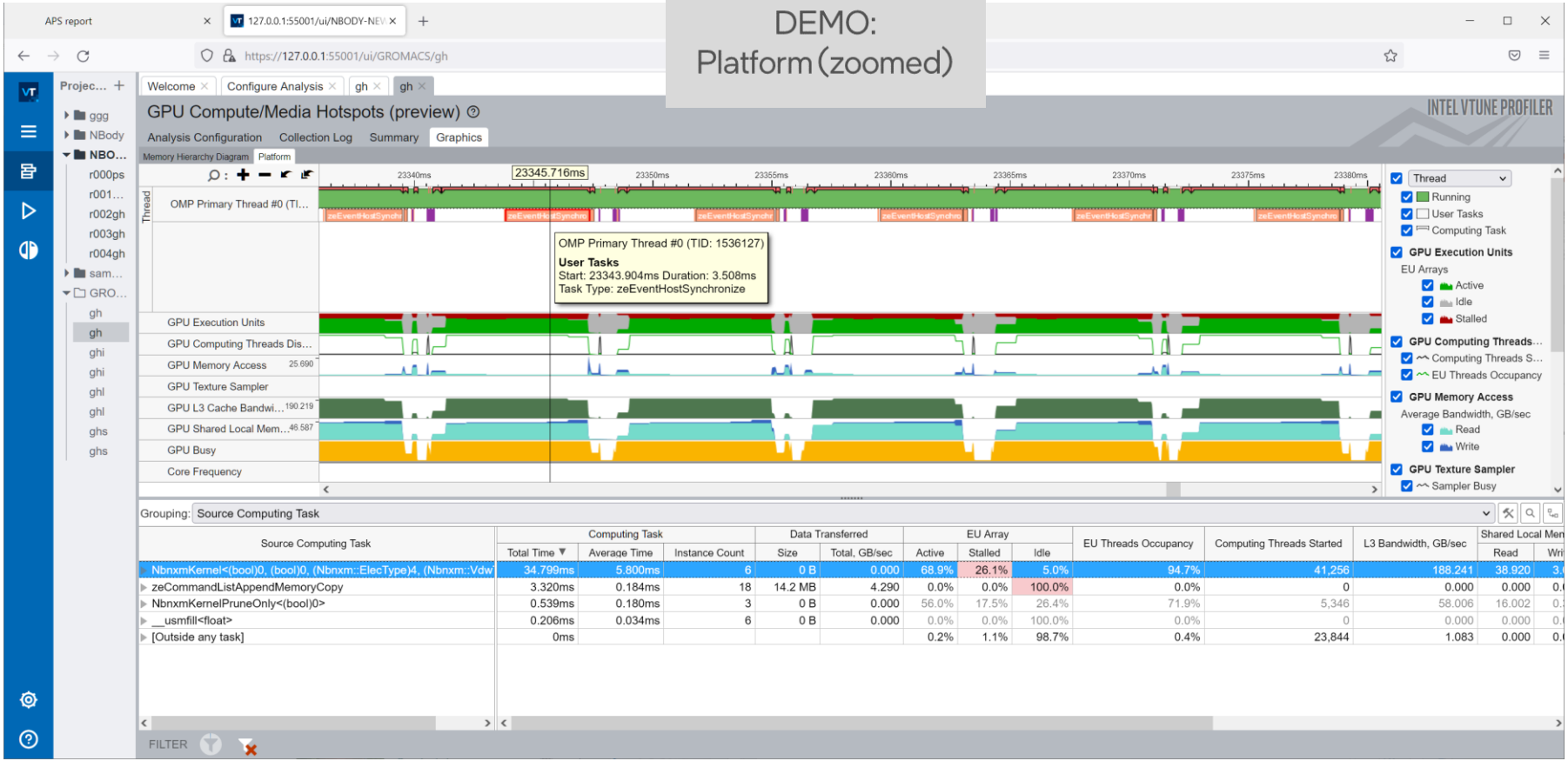
```
vtune -c gpu-hotspots -knob profiling-mode=source-analysis \
                      -knob source-analysis=mem-latency –r …
```

DEMO:
Memory Hierarchy
vtune –c gpu-hotspots

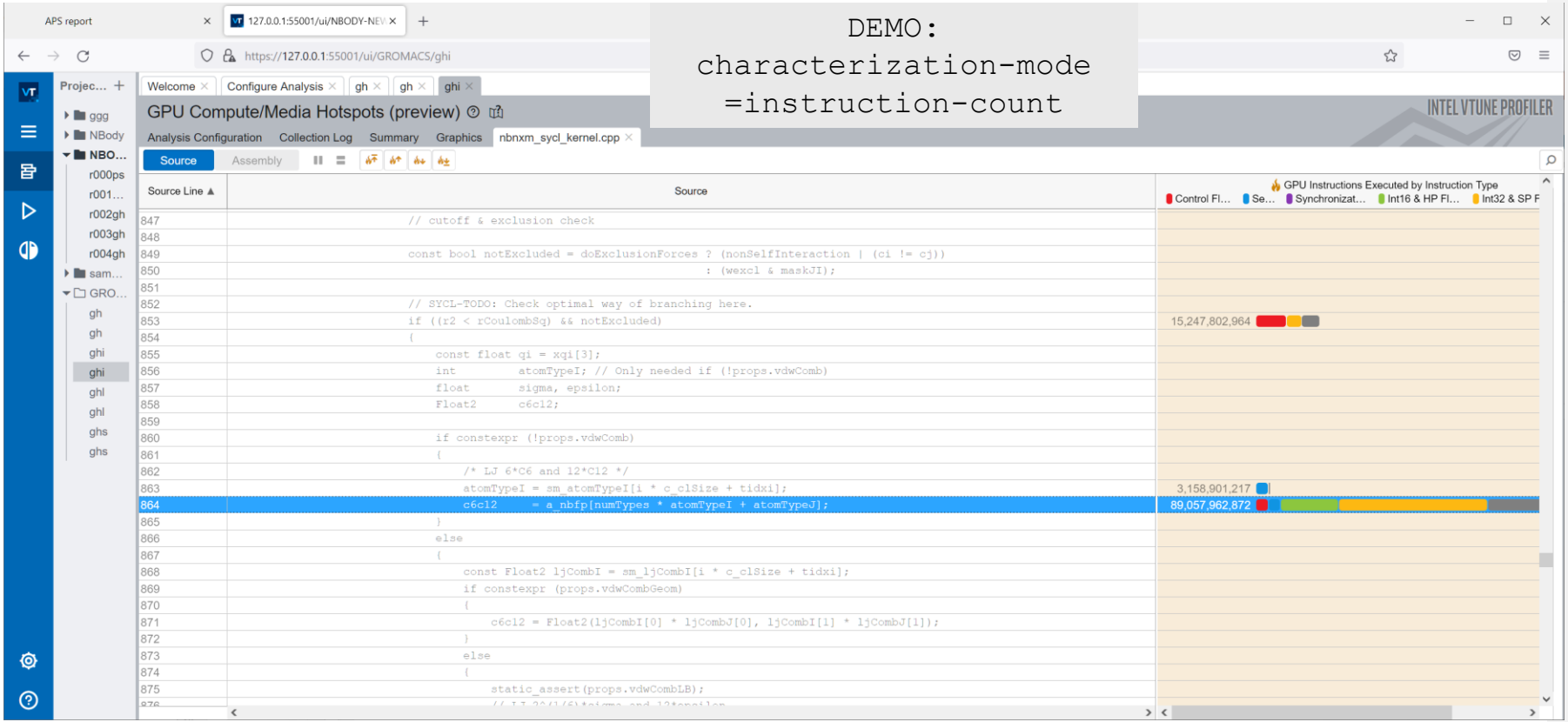


Computing Task	Work Size		Computing Task					Data Transferred		EU Array			EU Threads Occupancy	Com
	Global	Local	Total Time ▼	Average Time	Instance Count	SIMD Width	SVM Usage Type	Size	Total, GB/sec	Active	Stalled	Idle		
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7129	4 x 4 x 1	1.226s	0.006s	198	8		0 B	0.000	69.0%	26.1%	4.9%	94.8%	
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7149	4 x 4 x 1	0.614s	0.006s	99	8		0 B	0.000	68.8%	26.2%	4.9%	94.8%	
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7147	4 x 4 x 1	0.614s	0.006s	99	8		0 B	0.000	68.8%	26.2%	4.9%	94.8%	
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7137	4 x 4 x 1	0.614s	0.006s	99	8		0 B	0.000	68.9%	26.1%	5.0%	94.7%	
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7140	4 x 4 x 1	0.614s	0.006s	99	8		0 B	0.000	68.8%	26.2%	4.9%	94.8%	
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7117	4 x 4 x 1	0.614s	0.006s	99	8		0 B	0.000	68.8%	26.2%	5.0%	94.8%	
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7130	4 x 4 x 1	0.614s	0.006s	99	8		0 B	0.000	68.7%	26.3%	5.0%	94.7%	
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7162	4 x 4 x 1	0.613s	0.006s	99	8		0 B	0.000	68.8%	26.1%	5.2%	94.5%	
NbnxmKernel<(bool)0, (bool)0, (Nbnxm::ElecType)4, (Nbnxm::Vdw	4 x 4 x 7135	4 x 4 x 1	0.612s	0.006s	99	8		0 B	0.000	68.9%	26.1%	5.0%	94.7%	

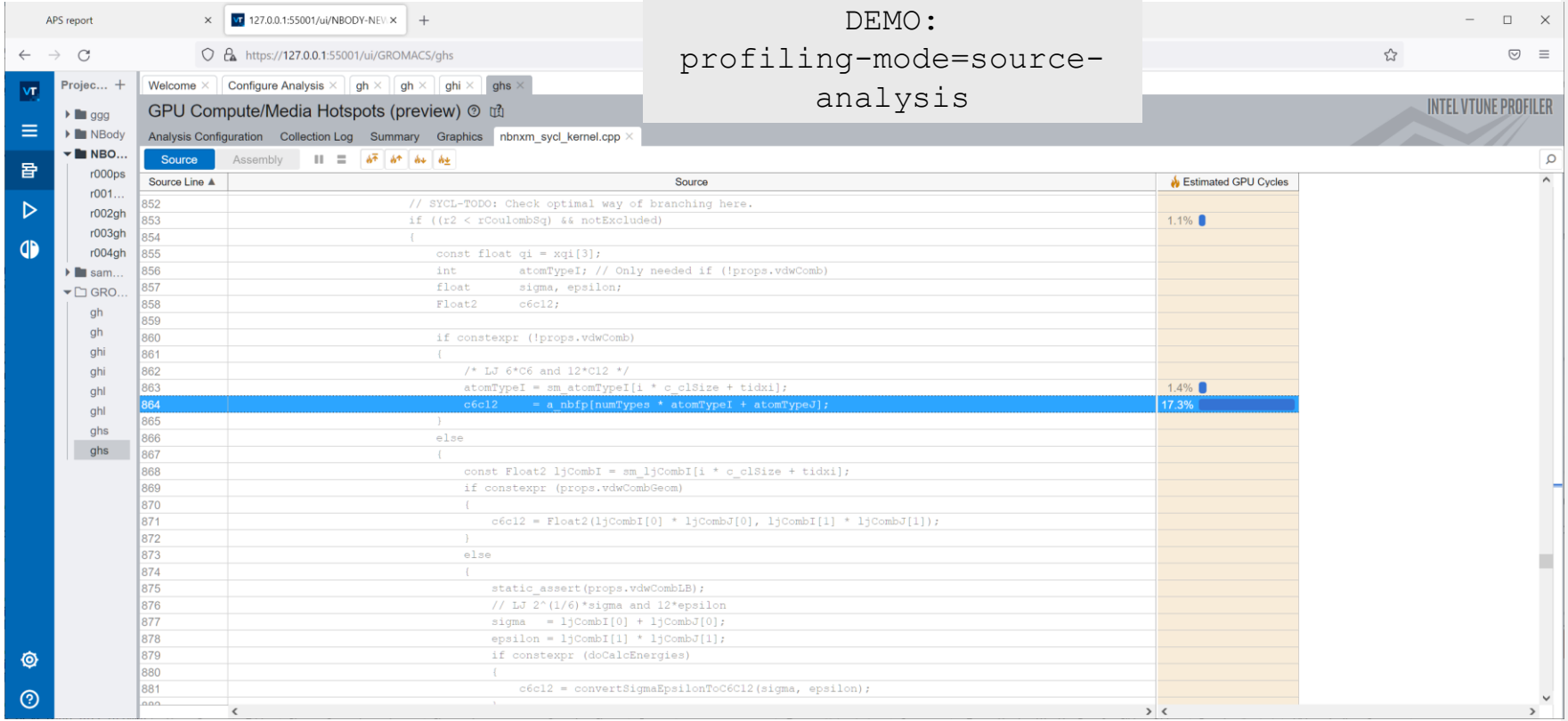
DEMO: Platform (zoomed)



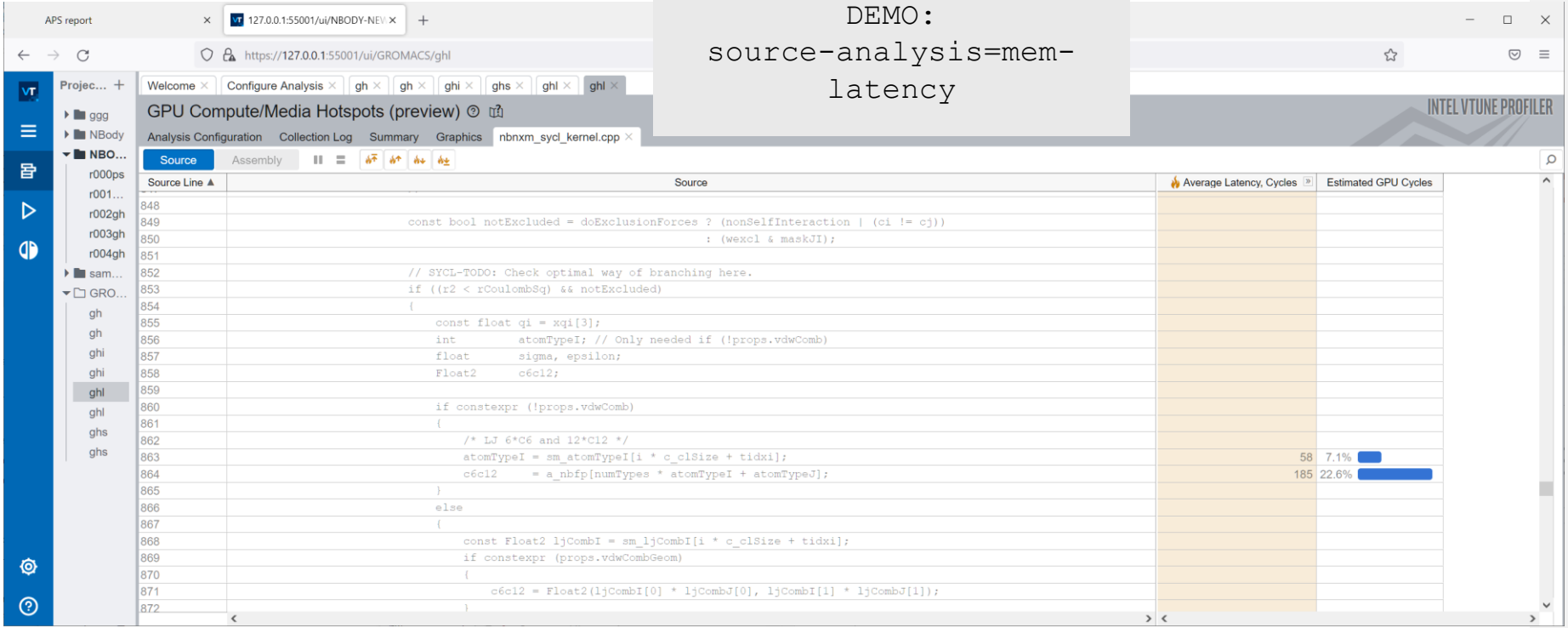
DEMO:
characterization-mode
=instruction-count



DEMO:
profiling-mode=source-analysis



DEMO:
source-analysis=mem-latency



Documentation

VTune cookbooks:

https://www.intel.com/content/www/us/en/develop/documentation/vtune-cookbook/top.html

Vtune on DevCloud:

https://www.intel.com/content/www/us/en/develop/documentation/vtune-cookbook/top/configuration-recipes/using-vtune-server-with-vs-code-intel-devcloud.html

GPU profiling:

https://www.intel.com/content/www/us/en/develop/documentation/vtune-help/top/analyze-performance/accelerators-group/gpu-compute-media-hotspots-analysis.html

Notices & Disclaimers

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.


intel