Introduction into NVIDIA® Nsight™ Systems

HLRS
High-Performance Computing Center | Stuttgart

Dr. Momme Allalen | LRZ | 12.07.2022

PRACE Training

LRZ as part of the Gauss Centre for Supercomputing (GCS) and IT4Innovations belong to the 14 **PRACE Training Centres** that started in 2012-2017-2020:

- Barcelona Supercomputing Center (Spain)
- CINECA Consorzio Interuniversitario (Italy)
- CSC – IT Center for Science Ltd (Finland)
- EPCC at the University of Edinburgh (UK)
- Gauss Centre for Supercomputing (Germany)
- Maison de la Simulation (France)
- GRNET – Greek Research and Technology Network (Greece)
- ICHEC – Irish Centre for High-End Computing (Ireland)
- IT4I – National Supercomputing Center VSB Technical University of Ostrava (Czech Republic)
- SURFsara (The Netherlands)
- TU Wien – VSC Research Center (Austria)
- University ANTWERPEN – VSC & CÉCI (Belgium)
- University of Ljubljana – HPC Center Slovenia (Slovenia)
- Swedish National Infrastructure for Computing (SNIC) (Sweden)

Mission: Serve as **European hubs and key drivers of advanced high-quality training** for researchers working in the computational sciences.

http://www.training.prace-ri.eu/

DEEP LEARNING INSTITUTE

DLI Mission: Help the world to solve the most challenging problems using AI and deep learning

We help developers, data scientists and engineers to get started in architecting, optimizing, and deploying neural networks to solve real-world problems in diverse industries such as autonomous vehicles, healthcare, robotics, media & entertainment and game development.

CUDA® PROFILING TOOLS

nvvp: NVIDIA visual profiler
nvprof: tool to understand and optimize the performance of your CUDA, OpenACC or OpenMP applications,

Application level opportunities
Overall application performance
Overlap CPU and GPU work, identify the bottlenecks (CPU or GPU)
Overall GPU utilization and efficiency
-Overlap compute and memory copies
-Utilize compute and copy engines effectively.

Nsight Systems
Nsight Compute

Kernel level opportunities
- Use memory bandwidth efficiently
- Use compute resources efficiently
- Hide instruction and memory latency

There are more features, example for Dependency Analysis
Command: **nvprof --dependency-analysis --cpu-thread-tracing on ./executable_cuda**

NSIGHT PRODUCT FAMILY

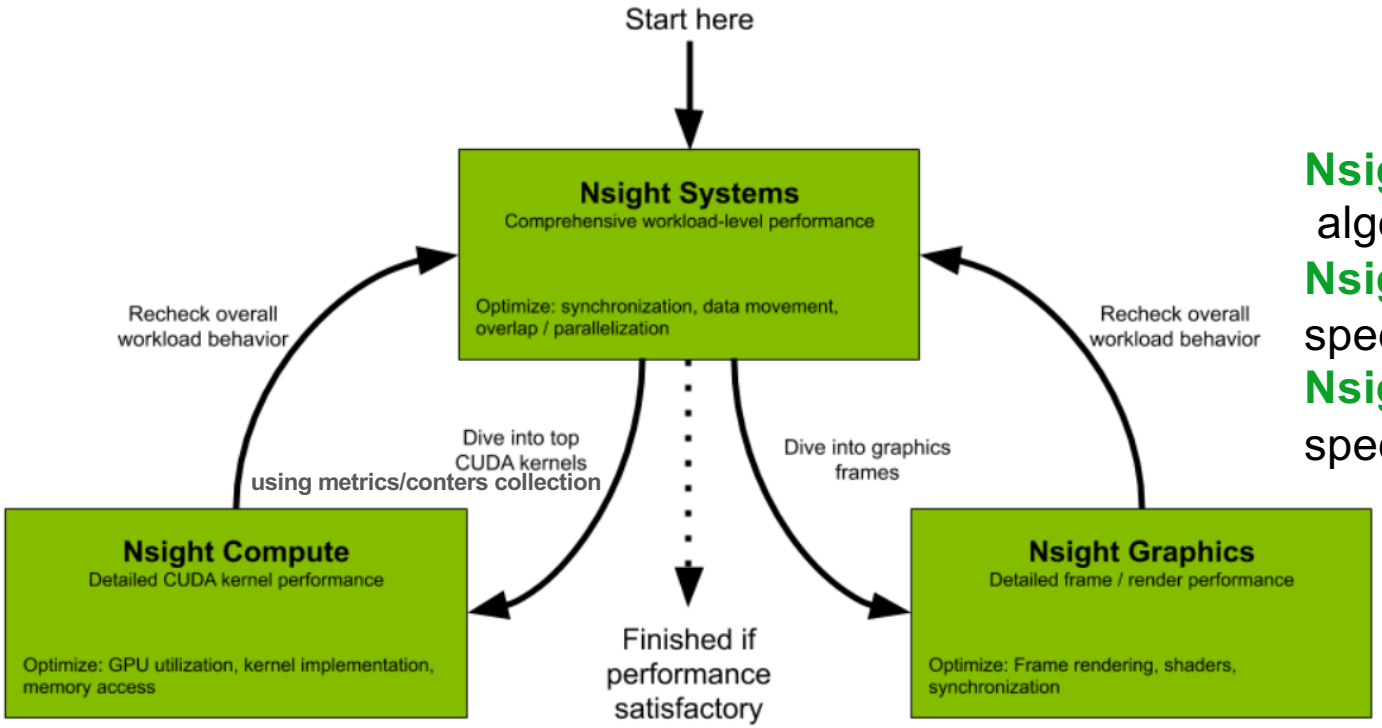


Figure 1. Flowchart describing working with new NVIDIA Nsight tools for performance optimization

Nsight Systems – Analyze application algorithm system wide

Nsight Compute – Debug/&optomize specific CUDA kernels

Nsight Graphics – Debug/&optimize specific graphics workloads

nvprof replaced with **nsys –profile....**

https://developer.nvidia.com/nsight-systems

NVIDIA NSIGHT SYSTEMS

- Provides users with a more complete view of how their codes balance workload across multiple CPUs and GPUs
- Locate optimization opportunities, helps and allows to identify issues such as:
  - GPU starvation
  - Insufficient CPU parallelisation or pipelining
  - Unexpectedly expensive CPU or GPU algorithm
  - Unnecessary GPU synchronization

- The tool uses low overhead tracing and sampling techniques to collect process and thread activity and visualize millions of events on a very fast GUI timeline
- Correlates that data across CPU cores and GPU streams, allowing users to investigate bottlenecks.
- Multi-platform: Linux & Windows, x86-64, Tegra, Power, MacOSX (host only)

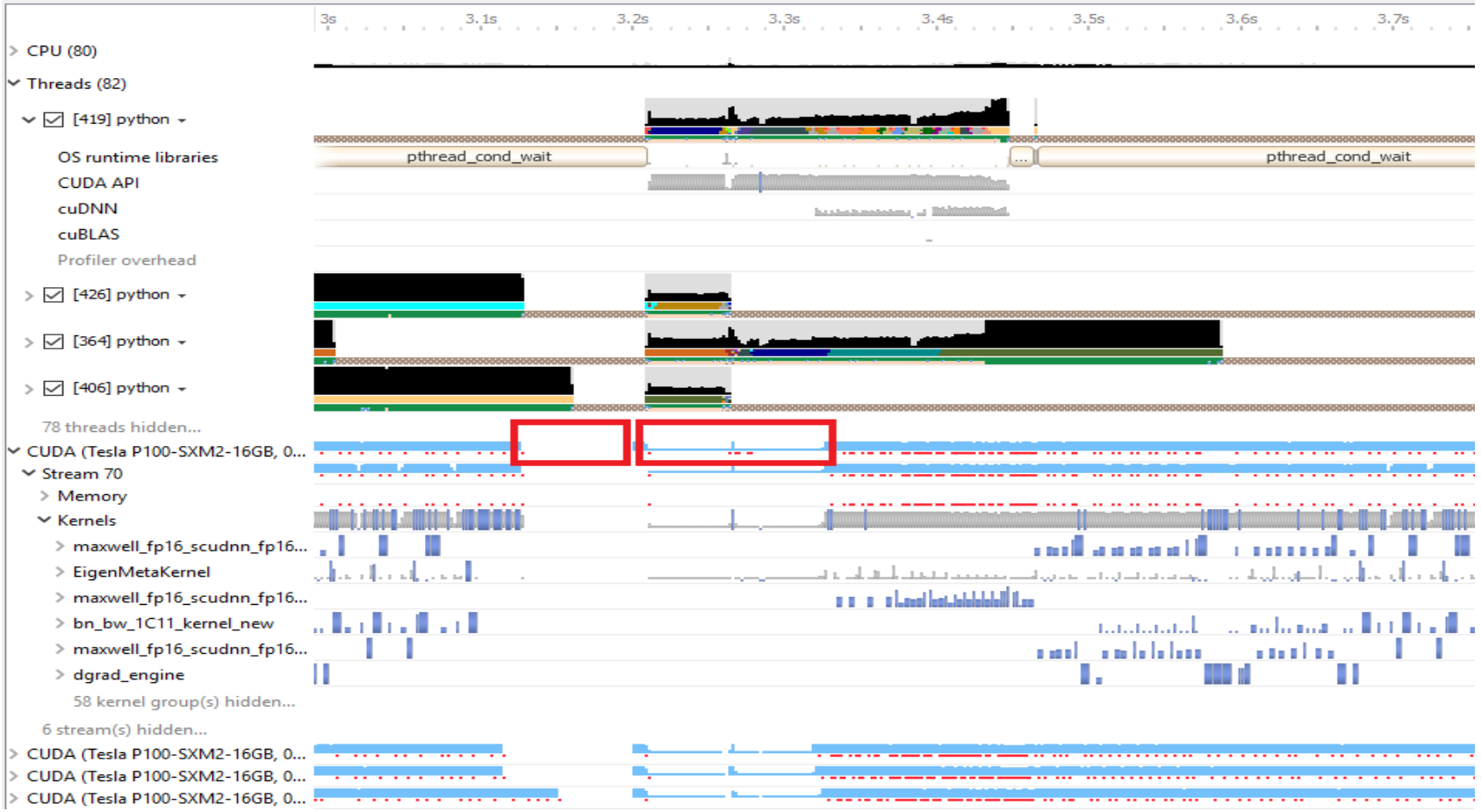
https://developer.nvidia.com/nsight-systems

Command Line Options nsys

Command	Description
profile	A fully formed profiling description requiring and accepting no further input. The command switch options used (see below table) determine when the collection starts, stops, what collectors are used (e.g. API trace, IP sampling, etc.), what processes are monitored, etc.
start	Start a collection in interactive mode. The start command can be executed before or after a launch command.
stop	Stop a collection that was started in interactive mode. When executed, all active collections stop, the CLI process terminates but the application continues running.
cancel	Cancels an existing collection started in interactive mode. All data already collected in the current collection is discarded.
launch	In interactive mode, launches an application in an environment that supports the requested options. The launch command can be executed before or after a start command.
shutdown	Disconnects the CLI process from the launched application and forces the CLI process to exit. If a collection is pending or active, it is cancelled
export	Generates an export file from an existing .nsys-rep file. For more information about the exported formats see the /documentation/nsys-exporter directory in your Nsight Systems installation directory.
stats	Post process existing Nsight Systems result, either in .nsys-rep or SQLite format, to generate statistical information.
analyze	Post process existing Nsight Systems result, either in .nsys-rep or SQLite format, to generate expert systems report.
status	Reports on the status of a CLI-based collection or the suitability of the profiling environment.
sessions	Gives information about all sessions running on the system.

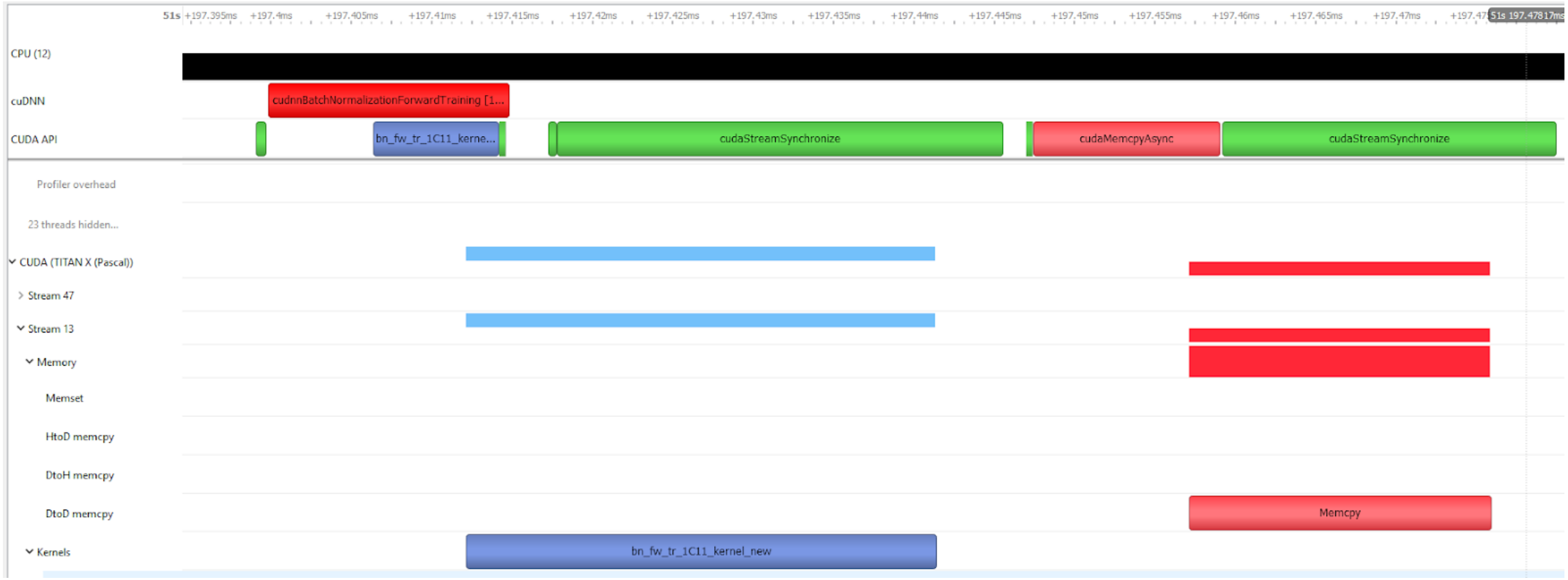
https://docs.nvidia.com/nsight-systems/UserGuide/index.html

GPU Starvation Investigations



https://developer.nvidia.com/nsight-systems

Unnecessary GPU Synchronization Calls



https://developer.nvidia.com/nsight-systems

NVIDIA NSIGHT SYSTEMS

- Support: **MPI**, **OpenACC**, **OpenMP**
- Complex data mining capabilities, enables to go beyond basic statistics.
- Support multiple simultaneous sessions.
- **MPI** trace feature enables to analyse when the threads are busy or blocked in long-running functions of the **MPI** standard, available on **OpenMPI** , **MPICH and NVShmem**.
- **OpenACC** trace enables to see where code has been offload and parallelized onto the GPU, which helps you to analyse the activities executing on the CPUs and GPUs in parallel.
- Tracing **OpenMP** code is available for compilers supporting **OpenMP 5** and **OMPT interface.** This capability enables tracing of the parallel regions of code that are distributed either across multiple threads or to the GPU.
- Provides support for **CUDA** graphs. To understand the execution of the source of **CUDA** kernels and execution of **CUDA** graphs, kernels can be correlated back through the graph lunch, instantiation, and all the way back to the node creation, to identify the origin of the kernel execution on the GPU.

NVIDIA NSIGHT COMPUTE (ncu)

Interactive CUDA Kernel profiler

Targeted metric sections for various performance aspects (Debug/&Profile)
API debugging via a user interface command line tool.

Very high freq GPU perf counter, customizable data collection and presentation (tables, charts ..,)

Python-based rules for guided analysis (or postprocessing)

Provides a customizable and data-driven user interface and metric collection and can be extended with analysis scripts for post-processing results.

https://developer.nvidia.com/nsight-compute

NVIDIA NSIGHT COMPUTE
Important Features

- Result comparison across one or multiple reports within the tool
- Graphical profile report
- Interactive kernel profiler and API debugger: debugging CPU and GPU simultaneously and capable of handling thousands of simultaneous threads.
- Fast data collection
- GUI and command line interface
- Fully customizable reports and analysis rules.

https://developer.nvidia.com/nsight-compute

Nsight Compute Feature Spotlight in CUDA Toolkit 11 and A100

- **Roofline Analysis**

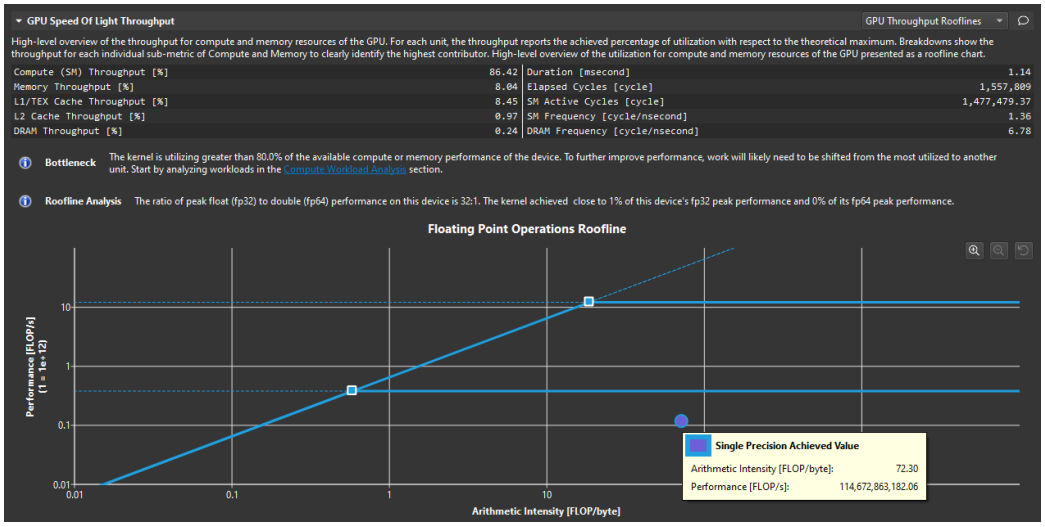
  Arithmetic intensity= Compute/Memory

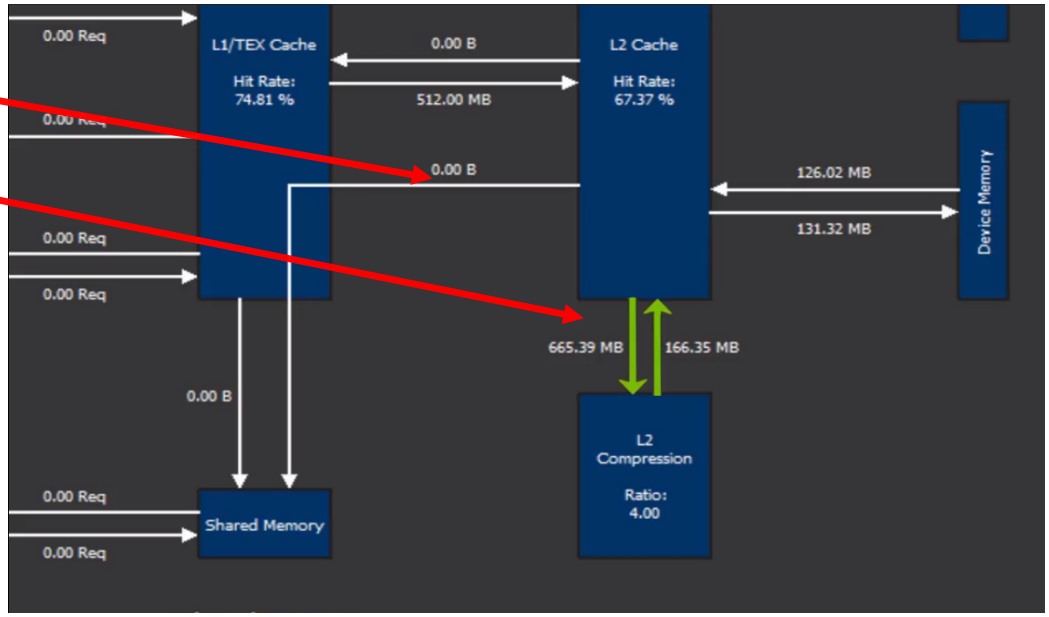
  FLOPS = Floating Points Ops/Second

- **Asynchronous copy**

- **Sparse Data Compression**

Shows the amount of data compressed through this feature and the compression ratio, helps on kernels with bandwidth or cache issues.





https://developer.nvidia.com/nsight-compute

NVIDIA® Tools Extension SDK (NVTX)

- C-based Application Programming Interface (API) for annotating events, code ranges, and resources in your applications
- Codes which integrate NVTX can use NVIDIA Nsight, Tegra System Profiler, and Visual Profiler to capture and visualize these events and ranges.

```
[allalen1@jwlogin22 v2]$ ncu -h | grep nvtx
  --nvtx                                Enable NVTX support.
  --nvtx-include arg                    Adds include statement to the NVTX filter, which allows selecting kernels to
  --nvtx-exclude arg                    Adds exclude statement to the NVTX filter, which allows selecting kernels to
  --print-nvtx-rename arg (=none)        Select how NVTX should be used for renaming:
                                           per-nvtx
Usage of --nvtx-include and --nvtx-exclude:
  ncu --nvtx --nvtx-include "Domain A@Range A"
  ncu --nvtx --nvtx-exclude "Range A]"
  ncu --nvtx --nvtx-include "Range A" --nvtx-exclude "Range B"
```

https://docs.nvidia.com/nsight-visual-studio-edition/nvtx/index.html

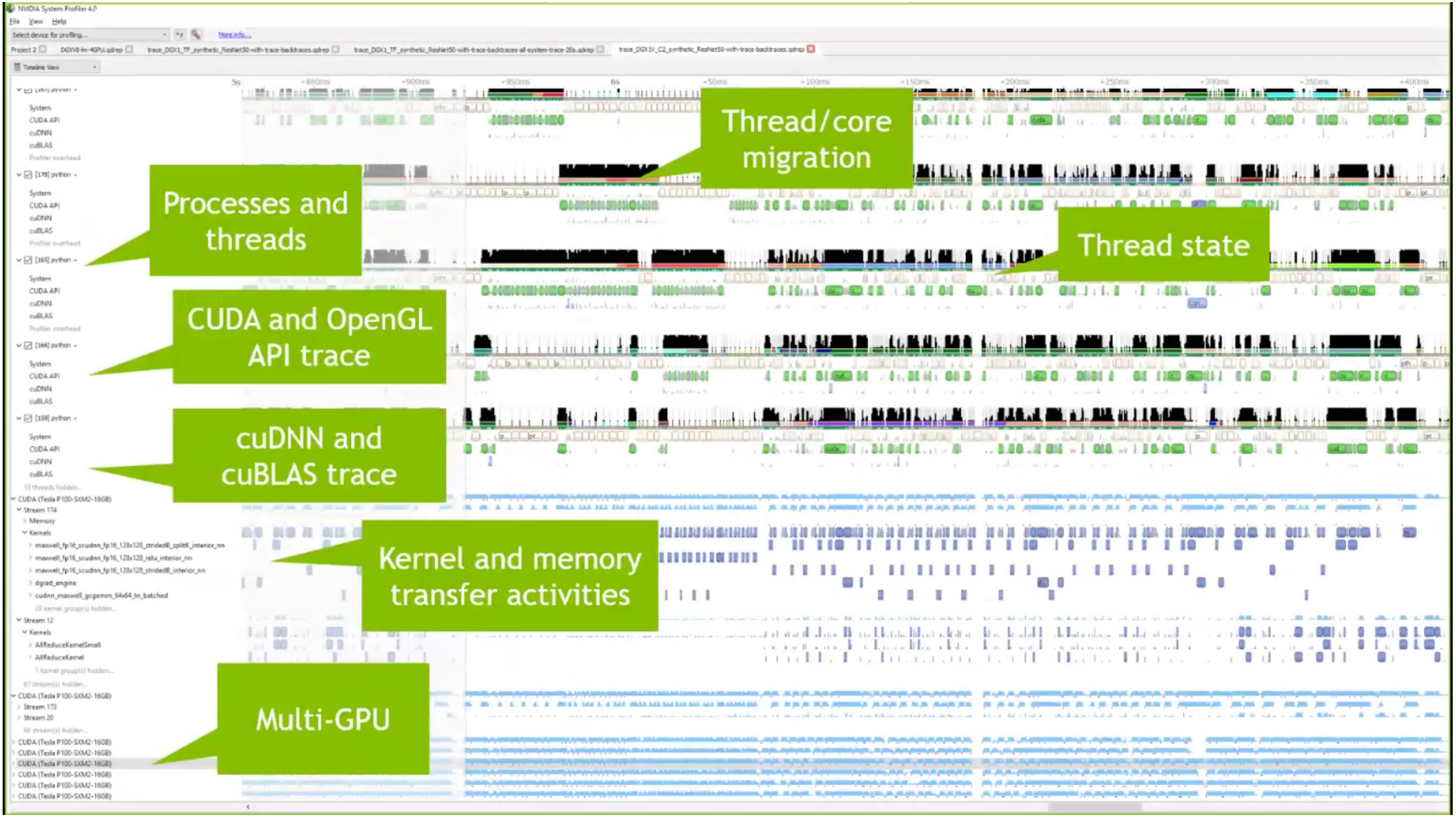
NVIDIA® Tools Extension SDK (NVTX)

```c
#include <nvToolsExt.h>
#include <sys/syscall.h>
#include <unistd.h>

static void wait(int seconds) {
    nvtxRangePush(__FUNCTION__);
    nvtxMark("Waiting...");
    sleep(seconds);
    nvtxRangePop();
}

int main(void) {
    nvtxNameOsThread(syscall(SYS_gettid), "Main Thread");
    nvtxRangePush(__FUNCTION__);
    wait(1);
    nvtxRangePop();
}
```

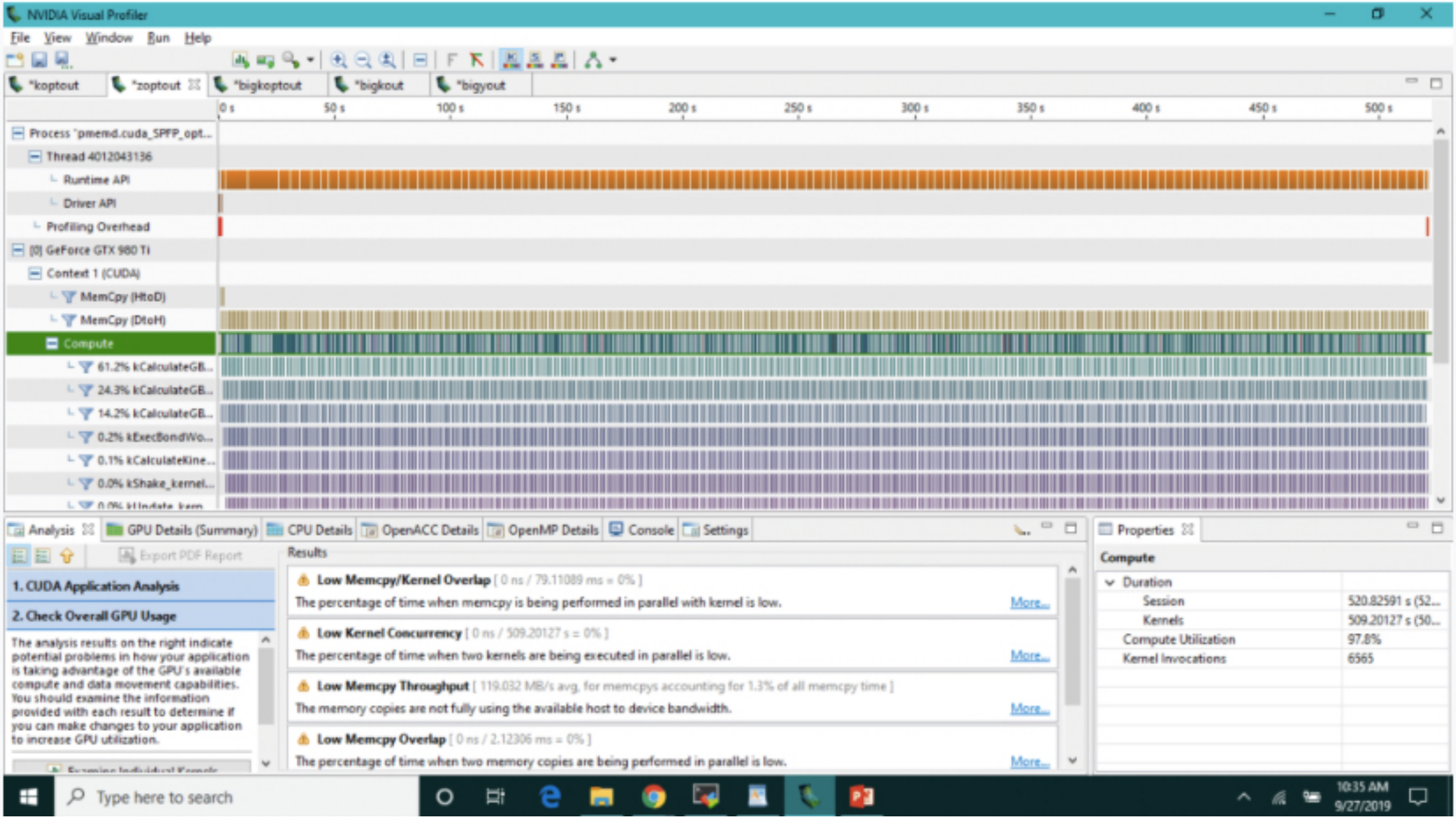
nsys profile –t nvtx --stats=true …
Or for Julia code:
nsys profile -t nvtx,cuda -o output_file.qdrep
julia --project=../../ script.jl

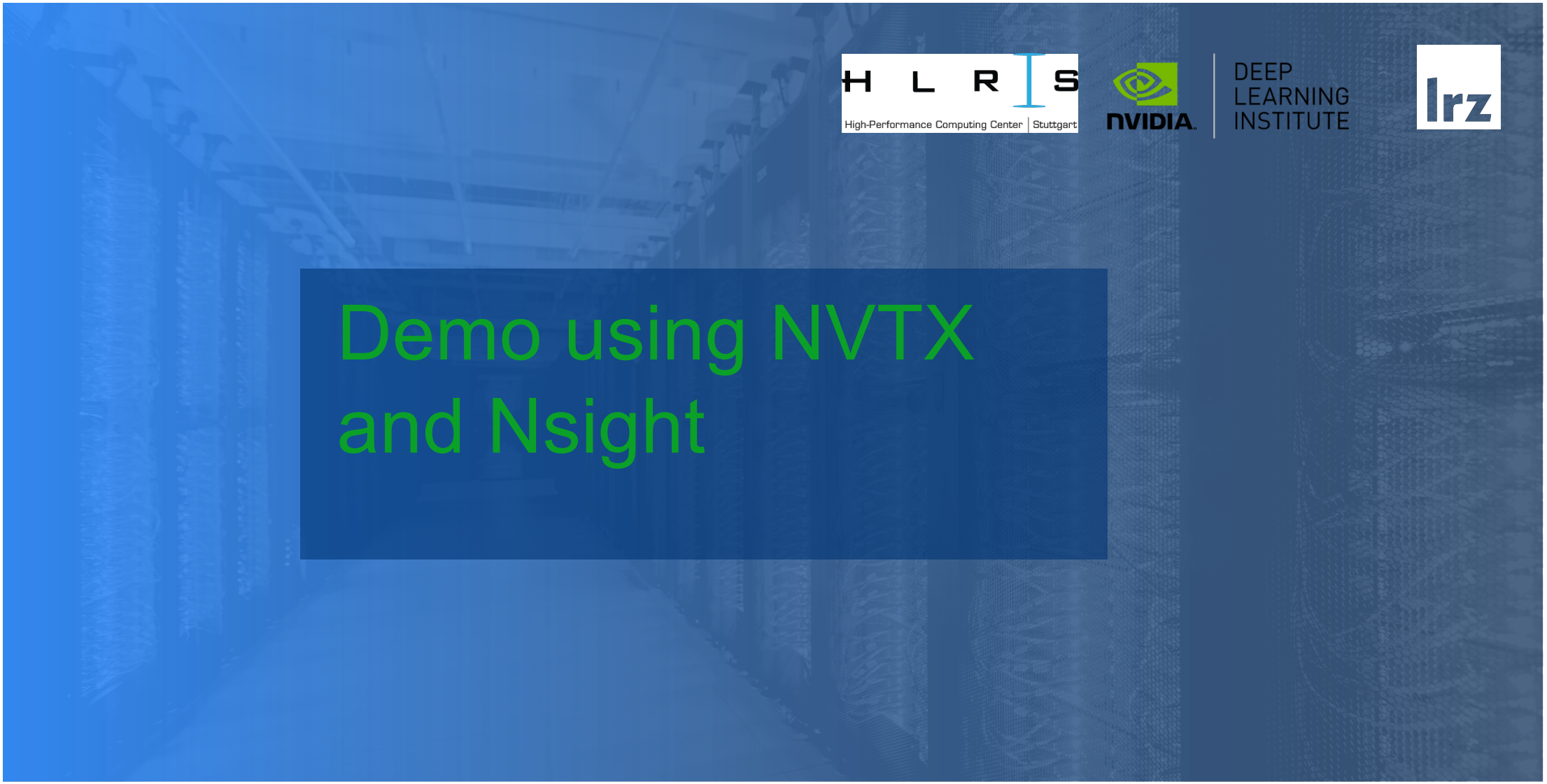
https://docs.nvidia.com/nsight-visual-studio-edition/2020.1/nvtx/index.html

https://docs.nvidia.com/nsight-visual-studio-edition/nvtx/index.html


Thread/core migration
Processes and threads
Thread state
CUDA and OpenGL API trace
cuDNN and cuBLAS trace
Kernel and memory transfer activities
Multi-GPU


File Help
ubuntu@10.20.13.72
Device is ready More info...
Project 1
Report 1
Project 1
Report 1
Function Table View
99.6s 99.8s 100s 100.2s 100.29s 100.4s 100.6s 100.8s 101s 101.2s 101.4s 101.6s 101.8s 102s
CPU (4)
Core 0
Core 1
Core 2
Core 3
Frequency
CPU (1912.5 ...
Threads (4)
[6671] kfusio...
[6713] CUPTI...
[6712] [TSP]
[6714] kfusio...
CUDA (GM20B)
Default Stream
Memory
Memset
HtoH ...
Kernels
gener...
initVol...
mm2...
bilater...
halfSa...
depth...
vertex...
trackK...
reduce
raycast...
Blocked
Duration: 55.942ms
Call stack at 100.255s
[kernel.kallsyms]!kfree
libcuda.so.1.1!Func at 0xf64099f8, len 188
libcuda.so.1.1!Func at 0xf6382288, len 764
libcuda.so.1.1!Func at 0xf6386a30, len 592
libcuda.so.1.1!Func at 0xf62c3dd0, len 300
libcuda.so.1.1!Func at 0xf62c3fe4, len 56
libcuda.so.1.1!Func at 0xf62b224c, len 152
libcuda.so.1.1!cuCtxSynchronize
libToolsInjection32.so!Func at 0xf5e8f330, len 128
libToolsInjection32.so!Func at 0xf5e93dfc, len 116
libToolsInjection32.so!Func at 0xf5e99eb4, len 2736
[Unknown]!0xf5e99cf0
libcuda.so.1.1!Func at 0xf644c47c, len 96
libcuda.so.1.1!cuLaunchKernel
libcudart.so.6.5.51!Func at 0xf702b73c, len 436
libcudart.so.6.5.51!cudaLaunch
kfusion-benchmark-cuda!
__device_stub__Z13raycastKernel5ImageI6float33RefES2_6Volume7Matrix4ffff(...)
kfusion-benchmark-cuda!Kfusion::raycasting(...)
kfusion-benchmark-cuda!main
libc-2.19.so!__libc_start_main
kfusion-benchmark-cuda!0xecc5
Top-Down View
Filter...
51.53% of data is shown due to applied filters.
Search...
Symbol Name
Self, %
App, %
Total, %
Module Name
0xecc5 62.22 62.22 /home/ubuntu/sdcard/slambench-demo/build/kfusion/kfusion-benchmark-cuda
0xf6f4320d 30.50 /lib/arm-linux-gnueabihf/libc-2.19.so
[Broken backtraces] 0.02 7.27 [Broken backtraces]


NVIDIA Visual Profiler
File View Window Run Help
*koptout
*zoptout
*bigkoptout
*bigkout
*bigyout
0 s
50 s
100 s
150 s
200 s
250 s
300 s
350 s
400 s
450 s
500 s
Process "pmemd.cuda_SPFP_opt...
Thread 4012043136
Runtime API
Driver API
Profiling Overhead
[0] GeForce GTX 980 Ti
Context 1 (CUDA)
MemCpy (HtoD)
MemCpy (DtoH)
Compute
61.2% kCalculateGB...
24.3% kCalculateGB...
14.2% kCalculateGB...
0.2% kExecBondWo...
0.1% kCalculateKine...
0.0% kShake_kernel...
Analysis
GPU Details (Summary)
CPU Details
OpenACC Details
OpenMP Details
Console
Settings
Export PDF Report
1. CUDA Application Analysis
2. Check Overall GPU Usage
The analysis results on the right indicate potential problems in how your application is taking advantage of the GPU's available compute and data movement capabilities. You should examine the information provided with each result to determine if you can make changes to your application to increase GPU utilization.
Results
Low Memcpy/Kernel Overlap [0 ns / 79.11089 ms = 0%]
The percentage of time when memcpy is being performed in parallel with kernel is low.
More...
Low Kernel Concurrency [0 ns / 509.20127 s = 0%]
The percentage of time when two kernels are being executed in parallel is low.
More...
Low Memcpy Throughput [119.032 MB/s avg, for memcpys accounting for 1.3% of all memcpy time]
The memory copies are not fully using the available host to device bandwidth.
More...
Low Memcpy Overlap [0 ns / 2.12306 ms = 0%]
The percentage of time when two memory copies are being performed in parallel is low.
More...
Properties
Compute
Duration
Session 520.82591 s (52...
Kernels 509.20127 s (50...
Compute Utilization 97.8%
Kernel Invocations 6565
Type here to search
10:35 AM
9/27/2019


H L R S
High-Performance Computing Center | Stuttgart
NVIDIA
DEEP
LEARNING
INSTITUTE
lrz
Demo using NVTX
and Nsight

```julia
using BenchmarkTools
using CUDA

using QXContexts

function main(args)
    file_path  = @__DIR__
    dsl_file   = joinpath(dirname(dirname(file_path)), "examples/ghz/ghz_5.qx")
    input_file = joinpath(dirname(dirname(file_path)), "examples/ghz/ghz_5.jld2")

    cg, _ = parse_dsl_files(dsl_file, input_file)

    # get time on gpu
    ctx_gpu = QXContext{CuArray{ComplexF32}}(cg)
    set_open_bonds!(ctx_gpu)
    # run to ensure all is precompiled
    t = NVTX.@range "Warm up" begin @elapsed ctx_gpu() end
    @info "GPU warmup ran in $t"
    CUDA.@profile NVTX.@range "Run iteration" begin
        ctx_gpu()
    end
    nothing
end

main(ARGS)
```

DEEP
LEARNING
INSTITUTE


THANK YOU

Instructor: Dr. Momme Allalen

www.nvidia.com/dli