

# Profiling CPU & MPI Applications with Nsight Systems

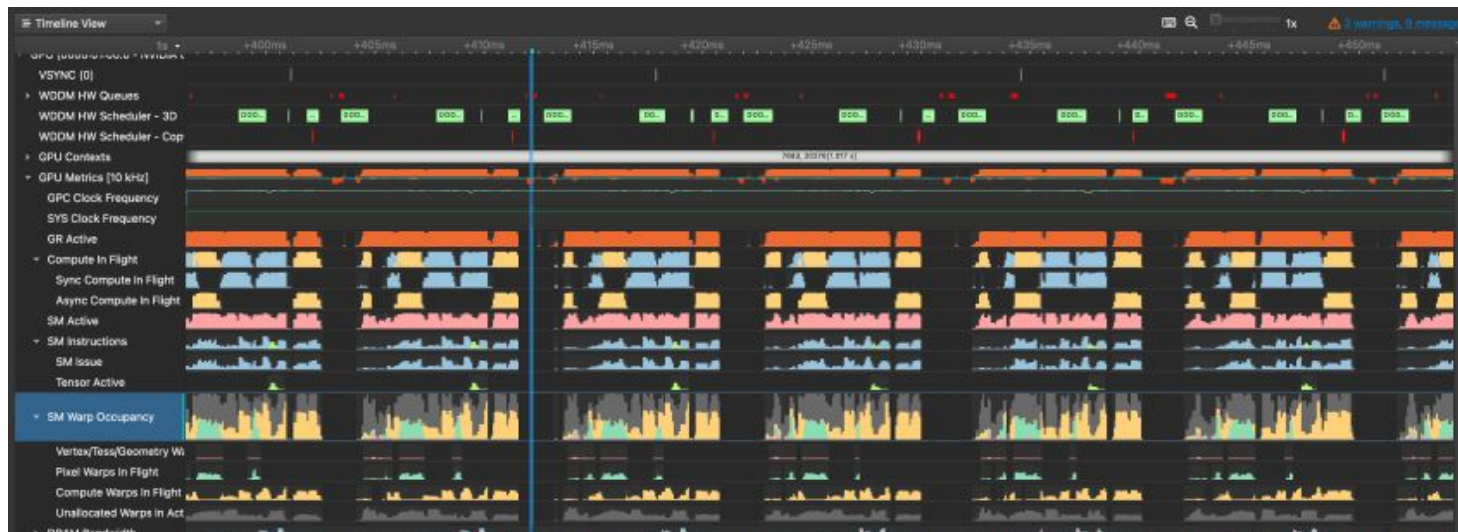
---

Yi Kuo @ PP24 Lab2

# NVIDIA Nsight Systems



- Provides an instinct timeline view of your program
- Developed by NVIDIA, mainly for analyzing GPU application performance
  - But is also very useful for CPU only applications!



# Prerequisites

- Download & Install Nsight Systems on your computer [here](#)

# Profiling non-MPI Applications with Nsight Systems

---

# Profiling non-MPI Applications with Nsight Systems

non-MPI: Single thread / Multi-thread (pthread / OpenMP) program

1. Load the nsys module
  - **module load nsys**
2. Add “nsys profile” in front of your running command (but after srun)
  - **srun -nl -cX nsys profile <nsys options> ./your\_program <program args>**
  - Generates a .nsys-rep file
3. Copy the .nsys-rep file to your computer
4. Open the report with Nsight Systems GUI on your computer

# Nsight Systems Profiling Options

- **-o <output.nsys-rep>**
- **--trace <events>**
  - Events to trace
  - Available options: **cuda**, **nvtx**, cublas, cublas-verbose, cusparse, cusparse-verbose, cudnn, cudla, cudla-verbose, cusolver, cusolver-verbose, opengl, opengl-annotations, openacc, **openmp**, **osrt**, **mpi**, nvvideo, vulkan, vulkan-annotations, dx11, dx11-annotations, dx12, dx12-annotations, openxr, openxr-annotations, oshmem, **ucx**, wddm, tegra-accelerators, python-gil, syscall, none
- **--start-later X**
  - Start profiling after X seconds
- **--duration Y**
  - Profile for Y seconds
- More options here:  
<https://docs.nvidia.com/nsight-systems/UserGuide/index.html#cli-profile-command-switch-options>

# Profiling MPI Applications with Nsight Systems

---

# Profiling MPI Applications with Nsight Systems

1. Load the nsys module & MPI module
  - `module load nsys`
  - `module load openmpi` or `module load mpi`
2. Create a wrapper script for each process (on the next page)
3. Run the wrapper script
  - `srun -nX ./wrapper.sh ./your_program <program args>`
  - Generates X .nsys-rep files
4. Copy the X .nsys-rep files to your computer
5. Open the report with Nsight Systems GUI on your computer with Multi-report view



# MPI Wrapper Script (wrapper.sh)

```
#!/bin/bash

mkdir -p nsys_reports

# Output to ./nsys_reports/rank_${N}.nsys-rep
nsys profile \
  -o "./nsys_reports/rank_${PMI_RANK}.nsys-rep" \
  --mpi-impl openmpi \
  --trace mpi,ucx,osrt \
  $@
```

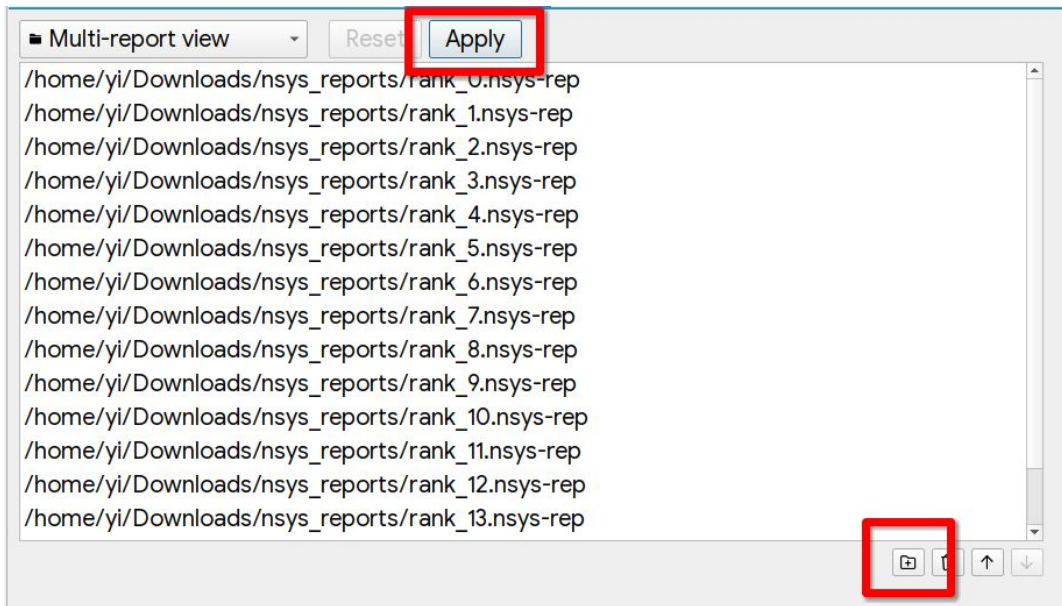
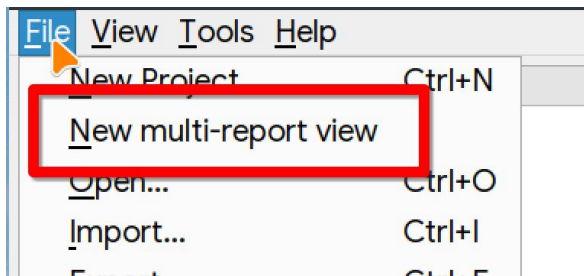
Remember to **chmod +x wrapper.sh** !

# Nsight Systems Profiling Options

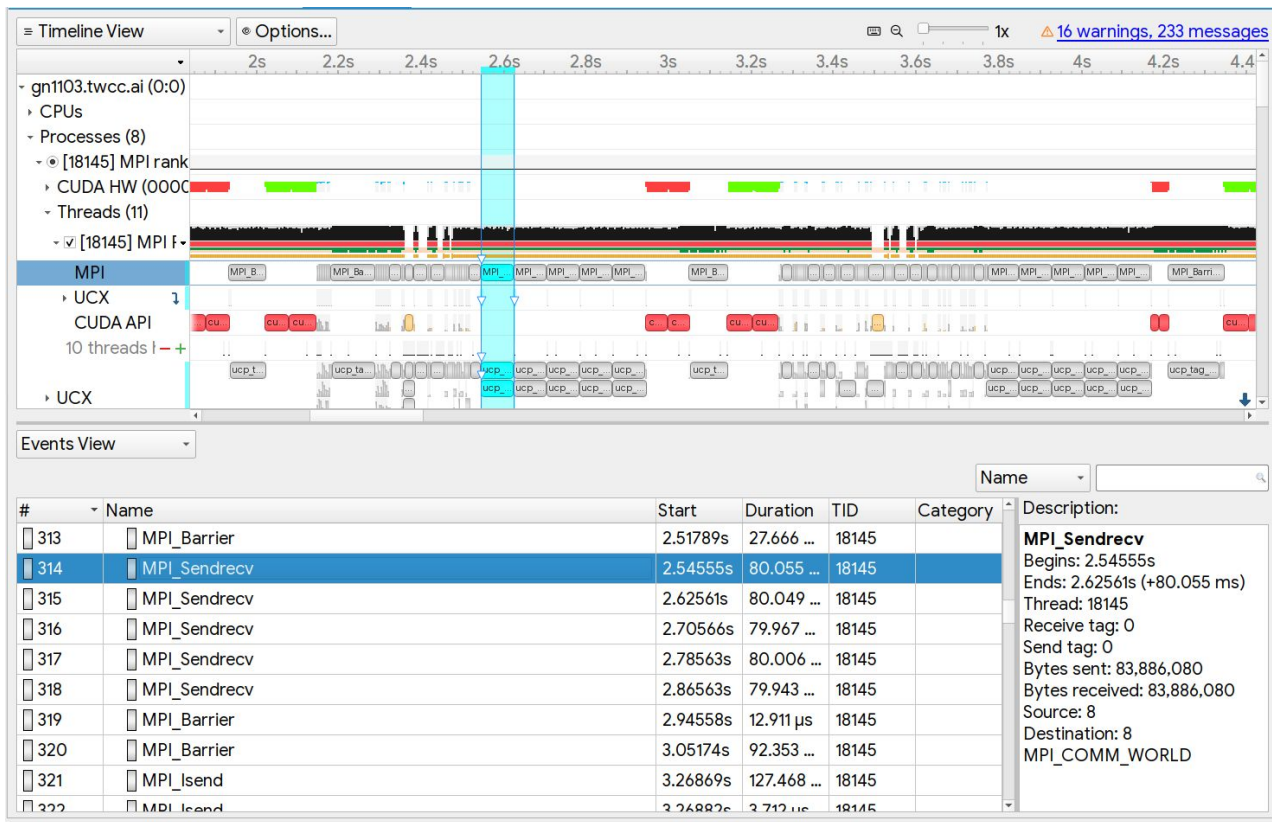
- **-o <output.nsys-rep>**
- **--trace <events>**
  - Events to trace
  - Available options: **cuda**, **nvtx**, cublas, cublas-verbose, cusparse, cusparse-verbose, cudnn, cudla, cudla-verbose, cusolver, cusolver-verbose, opengl, opengl-annotations, openacc, **openmp**, **osrt**, **mpi**, nvvideo, vulkan, vulkan-annotations, dx11, dx11-annotations, dx12, dx12-annotations, openxr, openxr-annotations, oshmem, **ucx**, wddm, tegra-accelerators, python-gil, syscall, none
- **--start-later X**
  - Start profiling after X seconds
- **--duration Y**
  - Profile for Y seconds
- **--mpi-impl <MPI implementation>**
  - **openmpi** for OpenMPI
  - **mpich** for Intel MPI
- More options here:  
<https://docs.nvidia.com/nsight-systems/UserGuide/index.html#cli-profile-command-switch-options>

# Multi-report View

- Download the reports to your local machine



## Timeline View & Events View



Right click on a track > click Show in Events View to view in Events View

# Stats System View - MPI Event Trace

Stats System View

Report: 0 - rank\_0.nsys-rep



CUDA GPU Trace

CUDA Kernel Launch & Exec Time Summary

CUDA Kernel Launch & Exec Time Trace

CUDA Summary (API/Kernels/MemOps)

DX11 PIX Range Summary

DX12 GPU Command List PIX Ranges Summary

DX12 PIX Range Summary

MPI Event Trace

NVTX GPU Projection Summary

NVTX GPU Projection Trace

NVTX Push/Pop Range Summary

NVTX Push/Pop Range Trace

NVTX Range Kernel Summary

NVTX Range Summary

CLI command::

```
nsys stats -r mpi_event_trace /home/yi/  
Downloads/nsys_reports/rank_0.sqlite
```

Start	End	Duration	Event	Pid	Tid	Tag	Rank	PeerRank	Ro
0.0501912s	0.130185s	79.994 ms	MPI_Sendrecv	18145	18145	0	0	8	-
0.130186s	0.210266s	80.080 ms	MPI_Sendrecv	18145	18145	0	0	8	-
0.210267s	0.290226s	79.960 ms	MPI_Sendrecv	18145	18145	0	0	8	-
0.290227s	0.370217s	79.990 ms	MPI_Sendrecv	18145	18145	0	0	8	-
0.370218s	0.370235s	17.768 µs	MPI_Barrier	18145	18145	-	0	-	-
0.474333s	0.504144s	29.811 ms	MPI_Barrier	18145	18145	-	0	-	-
0.617608s	0.617749s	141.070 µs	MPI_Isend	18145	18145	0	0	7	-
0.61775s	0.617754s	3.975 µs	MPI_Isend	18145	18145	0	0	1	-
0.617754s	0.617756s	1.471 µs	MPI_Irecv	18145	18145	0	0	7	-
0.617756s	0.617757s	949 ns	MPI_Irecv	18145	18145	0	0	1	-
0.617758s	0.618973s	1.216 ms	MPI_Waitall	18145	18145	-	0	-	-
0.617758s	0.618973s	1.216 ms	MPI_Waitall	18145	18145	-	0	-	-

# Export Stats

```
$ nsys stats -r mpi_event_trace <.sqlite or .nsys-rep>
```

```
$ nsys stats -r mpi_event_trace --format csv <.sqlite or .nsys-rep>
```

```
> nsys stats -r mpi_event_trace /home/yi/Downloads/nsys_reports/rank_0.sqlite
Processing [/home/yi/Downloads/nsys_reports/rank_0.sqlite] with [/opt/nvidia/nsight-systems/2023.3.1/host-linux-x64/reports/mpi_event_trace.py]...

** MPI Event Trace (mpi_event_trace):
```

Start (ns)	End (ns)	Duration (ns)	Event	Pid	Tid	Tag	Rank	PeerRank	RootRank	Size (MB)	CollSendSize (MB)	CollRecvSize (MB)
50,191,228	130,184,824	79,993,596	MPI_Sendrecv	18,145	18,145	0	0	8	8	83.886		
130,185,664	210,266,030	80,080,366	MPI_Sendrecv	18,145	18,145	0	0	8	8	83.886		
210,266,712	290,226,430	79,959,718	MPI_Sendrecv	18,145	18,145	0	0	8	8	83.886		
290,227,097	370,216,762	79,989,665	MPI_Sendrecv	18,145	18,145	0	0	8	8	83.886		
370,217,549	370,225,217	7,768	MPI_Barrier	18,145	18,145		0					

```
> nsys stats -r mpi_event_trace --format csv /home/yi/Downloads/nsys_reports/rank_0.sqlite
Processing [/home/yi/Downloads/nsys_reports/rank_0.sqlite] with [/opt/nvidia/nsight-systems/2023.3.1/host-linux-x64/reports/mpi_event_trace.py]...
Start (ns),End (ns),Duration (ns),Event,Pid,Tid,Tag,Rank,PeerRank,RootRank,Size (MB),CollSendSize (MB),CollRecvSize (MB)
50191228,130184824,79993596,MPI_Sendrecv,18145,18145,0,0,8,,83.886,,
130185664,210266030,80080366,MPI_Sendrecv,18145,18145,0,0,8,,83.886,,
210266712,290226430,79959718,MPI_Sendrecv,18145,18145,0,0,8,,83.886,,
290227097,370216762,79989665,MPI_Sendrecv,18145,18145,0,0,8,,83.886,,
370217549,370225217,7768,MPI_Barrier,18145,18145,,0,,,
```

Add your own traces using NVTX

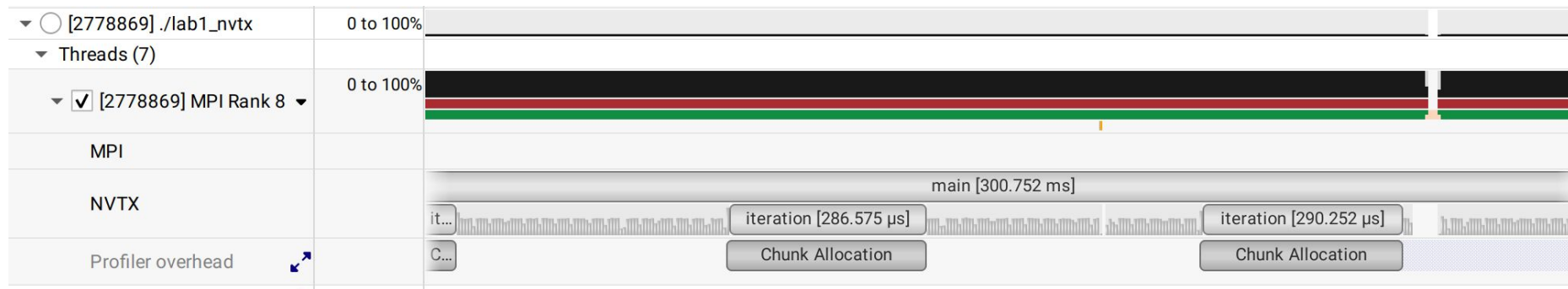
---

# NVTX

- Add your own ranges & show up on Nsight Systems
- Usage: <https://github.com/NVIDIA/NVTX/blob/release-v3/c/README.md>
- **#include <nvtx3/nvToolsExt.h>**
- **nvtxRangePush("My Range");**
- **nvtxRangePop();**
- Adding Colors
  - **nvtxEventAttributes\_t eventAttrib = {0};**
  - **eventAttrib.colorType = NVTX\_COLOR\_ARGB;**
  - **eventAttrib.color = COLOR\_GREEN;**
  - **eventAttrib.messageType = NVTX\_MESSAGE\_TYPE\_ASCII;**
  - **eventAttrib.message.ascii = "My Range";**
  - **nvtxRangePushEx(&eventAttrib);**
  - **nvtxRangePop();**



# Timeline View with NVTX



# Stats System View - NVTX Push/Pop Range Summary

Stats System View

Report: 0 - rank\_0.nsys-rep

MPI Event Trace	Time	Range	Total Time	Instances	Avg	Med	Min	Max	StdDev
NVTX GPU Projection Summary	99.0%	:main	296.683 ms	1	296.683 ms	296.683 ms	296.683 ms	296.683 ms	0 ns
NVTX GPU Projection Trace									
NVTX Push/Pop Range Summary	1.0%	:iteration	2.865 ms	10000	286 ns	183 ns	177 ns	331.677 µs	5.244 µs
NVTX Push/Pop Range Trace									
NVTX Range Kernel Summary									
NVTX Range Summary									
NVTX Start/End Range Summary									
Network Devices Congestion									
NvVideo API Summary									
OS Runtime Summary									
OpenACC Summary									

CLI command::  
nsys stats -r nvtx\_pushpop\_sum "/  
home/yi/tmp/nsys\_reports/  
rank\_0.sqlite"

# Tips

- If your program takes time to run, **be sure to set --start-after and --duration!**
  - Otherwise, the size will be very big & takes forever to open in GUI!
  - You only need to take a **sample** of how your program is running
  - **A recommended duration value is < 10s**
- You can export the stats, analysing it meaningfully and plot it using Google Sheets or Excel to put it in your report
  - Measuring I/O, Compute, Communication times
  - Load balance of thread/ranks
  - ... etc.