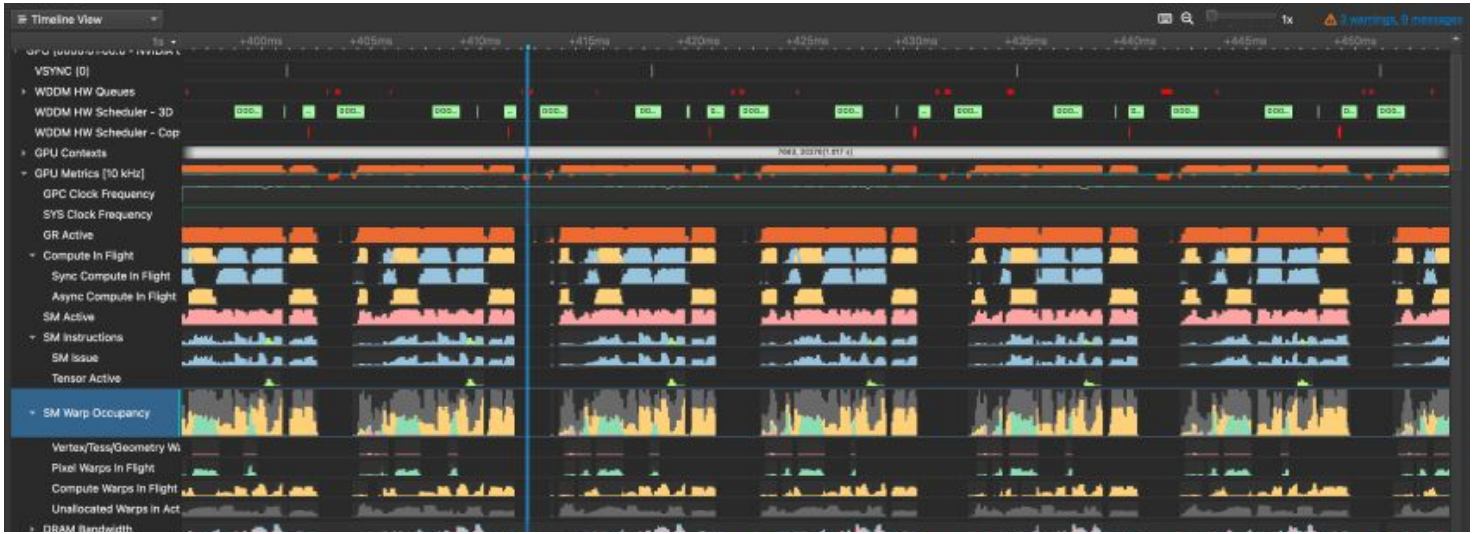
Profiling CPU & MPI Applications with Nsight Systems

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 -n1 -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 \
     $@
```

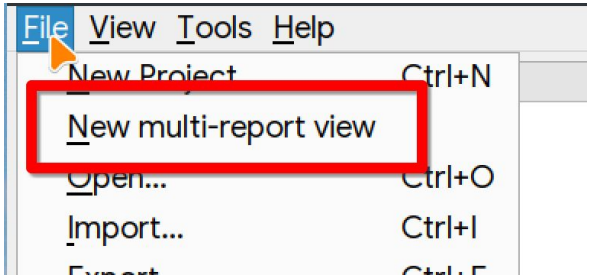
Remember to **`chmod +x wrapper.sh`**!

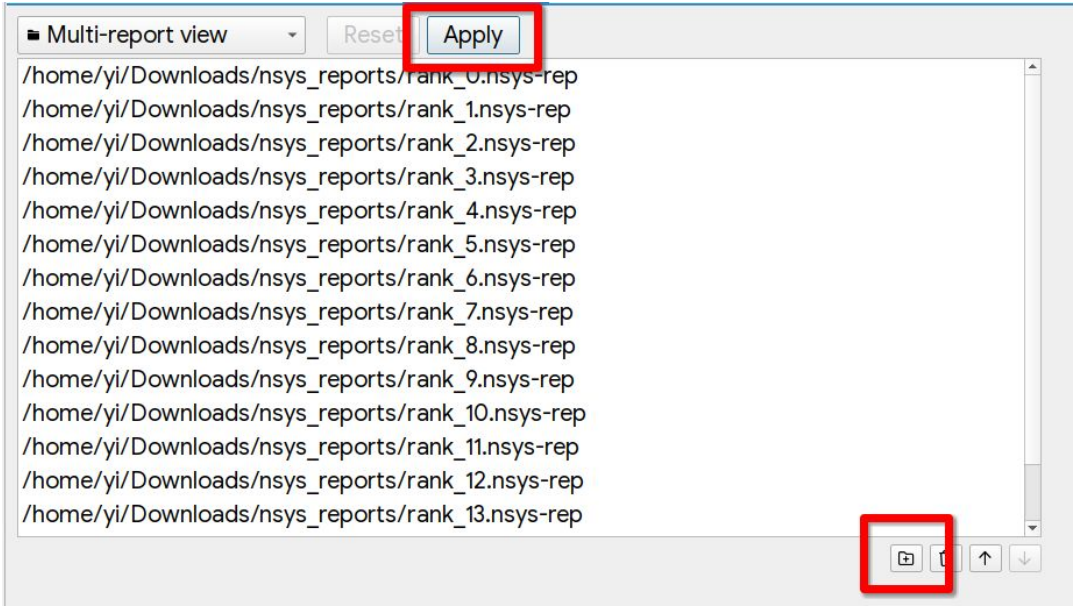
Nsight Systems Profiling Options

- **`-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

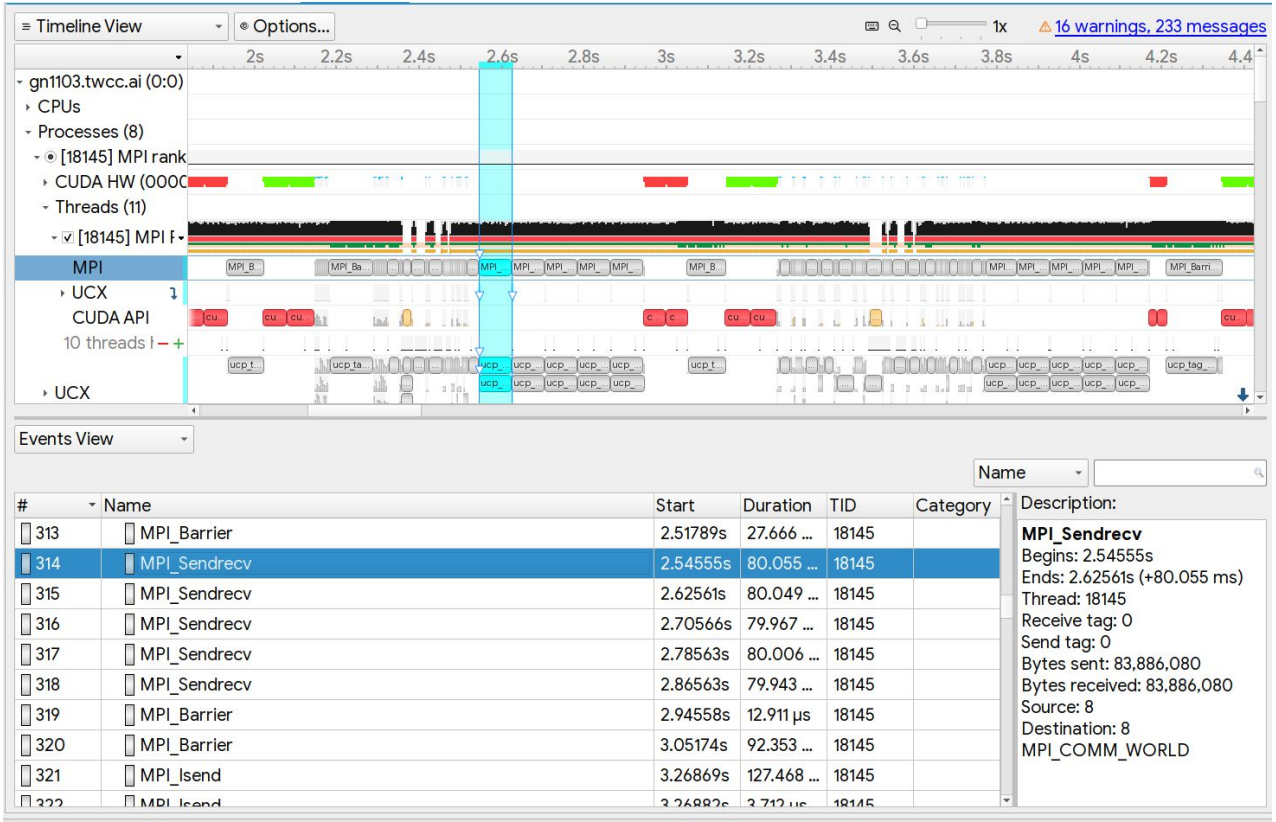
Multi-report View

- Download the reports to your local machine



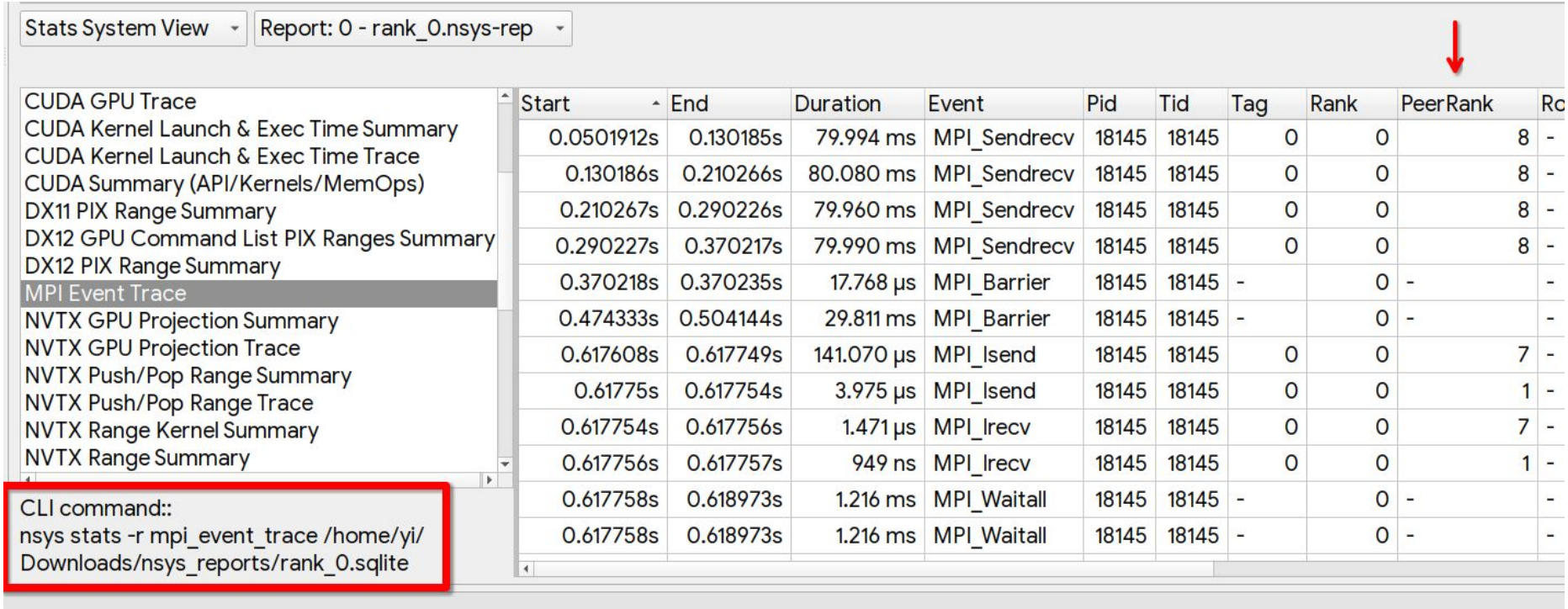


Timeline View & Events View



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

Stats System View - MPI Event Trace



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                83.886
 130,185,664  210,266,030     80,080,366  MPI_Sendrecv  18,145  18,145    0     0         8                83.886
 210,266,712  290,226,430     79,959,718  MPI_Sendrecv  18,145  18,145    0     0         8                83.886
 290,227,097  370,216,762     79,989,665  MPI_Sendrecv  18,145  18,145    0     0         8                83.886
 370,217,549  370,235,317         17,768  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,,
370217549,370235317,17768,MPI_Barrier,18145,18145,,,,,,,
```

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

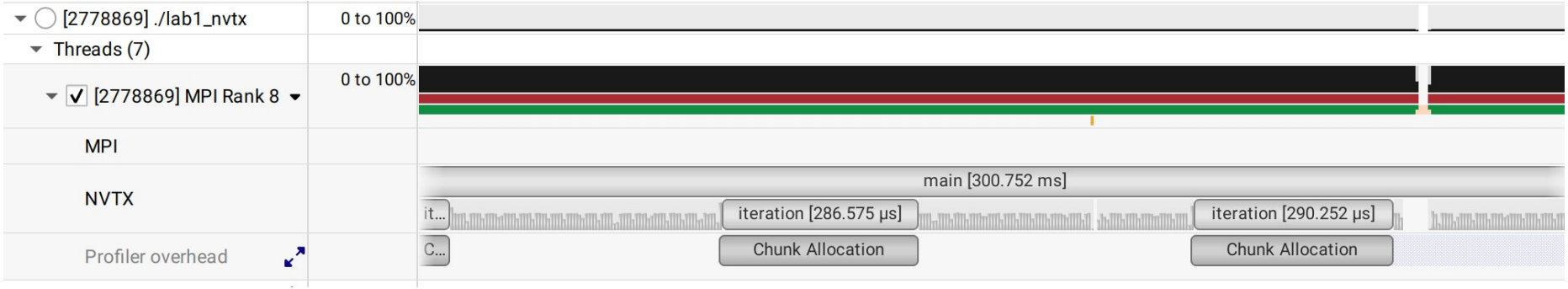
- `#include <nvtx3/nvToolsExt.h>`

- `nvtxRangePush("My Range");`
- `nvtxRangePop();`

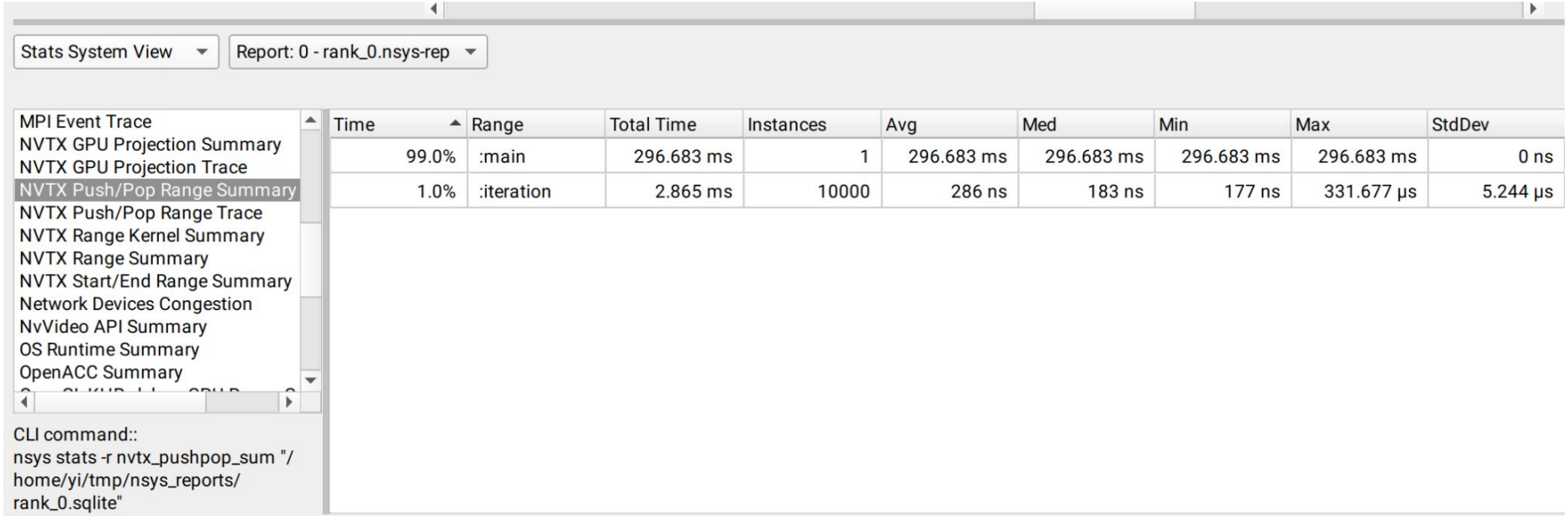
- Adding Colors
  - `nvtxEventAttributes_t eventAttrib = {0};`
  - `eventAttrib.colorType = NVTX_COLOR_ARGB;`
  - `eventAttrib.color = COLOR_GREEN;`
  - `eventAttrib.messageType = NVTX_MESSAGE_TYPE_ASCII;`
  - `eventAttrib.message.ascii = "My Range";`

  - `nvtxRangePushEx(&eventAttrib);`
  - `nvtxRangePop();`

Timeline View with NVTX



Stats System View - NVTX Push/Pop Range Summary



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.