

Roofline on NVIDIA GPUs Hackathon

Charlene Yang
NERSC, July 8 2020

<https://www.nersc.gov/users/training/events/roofline-on-nvidia-gpus-hackathon/>



- **Zoom link:** <https://lbnl.zoom.us/j/94652290933> (Breakout rooms available)
- **Slack workspace:**
 - https://join.slack.com/t/roofline-hack-2020/shared_invite/zt-fectnuhv-cFx8kdHdo5oMGbzLPL1KRQ
- **Training accounts:** <https://iris.nersc.gov/train> (Training code **bmOe**)
- **Nsight GUI:**
 - NoMachine: <https://docs.nersc.gov/connect/nx/>
 - Or, download the laptop version of Nsight Compute 2020.1 and Nsight Systems 2020.3:
 - <https://developer.nvidia.com/tools-overview>
- **Bring a code of your own or use our example code**
 - <https://gitlab.com/NERSC/roofline-on-nvidia-gpus/>

Agenda



Time	Topic	Presenter
08:50 - 09:00 am	Logistics	Charlene Yang
09:00 - 10:00 am	Introduction to Roofline Performance Model	Sam Williams
10:00 - 10:30 am	Mechanism behind Roofline Data Collection	Charlene Yang
10:30 - 10:45 am	Break	
10:45 - 11:15 am	Roofline Analysis with Nsight Compute	Max Katz
11:15 - 12:15 pm	Demo of a Real World HPC Example	Max Katz
12:15 - 13:15 pm	Lunch	
13:15 - 16:30 pm	Applying Roofline to Your Own Code (Hands-on)	All attendees
16:30 - 17:00 pm	Summary of your Experience Today	All attendees

Instructions on Cori GPU



- Training accounts expire on July 22, and there is a scheduled maintenance from July 10-14.
- Please move important files off Cori as soon as you can!

```
$ ssh username@cori.nersc.gov
```

```
$ enter password
```

- Nsight Compute 2020.1.0 and Nsight Systems 2020.2.5.8 are available through

```
$ module load cuda/11.0.167
```

- Our reservation: roofline_hack, 6 GPU nodes on Cori, 9am - 6pm, July 8

```
$ module load esslurm
```

```
$ salloc --reservation roofline_hack -C gpu -q shared --gres=gpu:1 -t  
00:30:00
```

Instructions on Cori GPU



Nsight Systems [\[https://docs.nvidia.com/nsight-systems/\]](https://docs.nvidia.com/nsight-systems/)

- ```
$ srun -n 1 nsys profile -o output ./app # produces output.qdrep
```
- `--stats=true` # generate summary statistics after the collection (default false)
  - `-t openacc` # trace OpenACC API (default cuda,opengl,nvtx,osrt)
  - `-f` # force overwrite of existing result files (default false)

## Nsight Compute [\[https://docs.nvidia.com/nsight-compute/\]](https://docs.nvidia.com/nsight-compute/)

```
$ srun -n1 nv-nsight-cu-cli --set default -o output ./app # detailed, full
```

```
cjyang@cgpu05:~> nv-nsight-cu-cli --list-sets

Identifier Sections Enabled Estimated Metrics

default LaunchStats, Occupancy, SpeedOfLight yes 35
detailed ComputeWorkloadAnalysis, InstructionStats, LaunchStats, MemoryWorkloadAnalysis, Occupancy, SchedulerStats, SourceCounters, SpeedOfLight, SpeedOfLight_RooflineChart, WarpStateStats no 157
full ComputeWorkloadAnalysis, InstructionStats, LaunchStats, MemoryWorkloadAnalysis, MemoryWorkloadAnalysis_Chart, MemoryWorkloadAnalysis_Tables, Occupancy, SchedulerStats, SourceCounters, SpeedOfLight, SpeedOfLight_RooflineChart, WarpStateStats no 162
source SourceCounters no 47
```



# Instructions on Cori GPU



- Be mindful of the profiling overhead and try this combination:

```
$ srun -n1 nv-nsight-cu-cli --set default \
 --section SpeedOfLight_RooflineChart -o output ./app
collect section files included in default set and section file SpeedOfLight_RooflineChart
this Roofline chart is device memory only
```

- For **hierarchical** Roofline (device memory, L2 and L1),

```
$ srun -n1 nv-nsight-cu-cli --set default \
 --section SpeedOfLight_HierarchicalDoubleRooflineChart -o output ./app
collect metrics for double, single, half precisions, or for tensor core
SpeedOfLight_HierarchicalSingleRooflineChart
SpeedOfLight_HierarchicalHalfRooflineChart
SpeedOfLight_HierarchicalTensorRooflineChart
```

# Instructions on Cori GPU



- To **specify** kernels and metrics for data collection,  

```
$ srun -n1 nv-nsight-cu-cli -k kernelname -s 5 \
 --metrics sm__cycles_elapsed.avg --csv -o output.csv ./app
```

# skip the first 5 invocations of *kernelname* and collect metric for number of cycles elapsed  
# metrics can be a comma-separated list
- To **query** available metrics,  

```
$ srun -n1 nv-nsight-cu-cli --query-metrics --query-metrics-mode all
```

# available modes are base (default), suffix, and all
- Example metrics for hierarchical Roofline that can be integrated to your workflow:
- <https://gitlab.com/NERSC/roofline-on-nvidia-gpus/>



# NERSC

**Thank You**



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

