Roofline on NVIDIA GPUs Hackathon

Charlene Yang
NERSC, July 8 2020

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

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

```bash
$ 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/)

```bash
$ srun -nl nv-nsight-cu-cli --set default -o output ./app
```

# detailed, full

```
cjyang@cori65:--> nv-nsight-cu-cli --list-sets

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Sections</th>
<th>Enabled</th>
<th>Estimated Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>LaunchStats, Occupancy, SpeedOfLight</td>
<td>yes</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>ComputeWorkloadAnalysis, InstructionStats, LaunchStats, MemoryWorkloadAnalysis nosis, Occupancy, SchedulerStats, SourceCounters, SpeedOfLight, SpeedOfLight_RooflineChart, WarpStateStats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>detailed</td>
<td>ComputeWorkloadAnalysis, InstructionStats, LaunchStats, MemoryWorkloadAnalysis nosis, Occupancy, SchedulerStats, SourceCounters, SpeedOfLight, SpeedOfLight_RooflineChart, WarpStateStats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>full</td>
<td>ComputeWorkloadAnalysis, InstructionStats, LaunchStats, MemoryWorkloadAnalysis nosis, Occupancy, SchedulerStats, SourceCounters, SpeedOfLight, SpeedOfLight_RooflineChart, WarpStateStats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>source</td>
<td>SourceCounters</td>
<td>no</td>
<td>47</td>
</tr>
</tbody>
</table>
```
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,

```bash
$ 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,

```bash
$ 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/](https://gitlab.com/NERSC/roofline-on-nvidia-gpus/)
Thank You