Roofline Model
Using Nsight-Compute

Jonathan Madsen
NERSC
Lawrence Berkeley National Lab
JRMadsen@lbl.gov
GPU Speed-of-Light

High-level overview of the utilization for compute and memory resources of the GPU. For each unit, the Speed Of Light (SOL) reports the achieved percentage of utilization with respect to the theoretical maximum. High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

<table>
<thead>
<tr>
<th>SOL SM [%]</th>
<th>Duration [msecond]</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.74</td>
<td>53.19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.06</td>
<td>64,232,068</td>
<td>1.86</td>
<td>64,068,674.54</td>
<td>0.28</td>
<td>1.21</td>
<td>0.75</td>
<td>850.53</td>
</tr>
</tbody>
</table>

**GPU Utilization**

**SOL SM Breakdown**

<table>
<thead>
<tr>
<th>SOL SM: Pipe Fp64 Cycles Active [%]</th>
<th>99.74</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOL SM: Pipe Shared Cycles Active [%]</td>
<td>99.74</td>
</tr>
<tr>
<td>SOL SM: Issue Active [%]</td>
<td>26.02</td>
</tr>
<tr>
<td>SOL SM: Issued Executed [%]</td>
<td>26.02</td>
</tr>
</tbody>
</table>

**SOL Memory Breakdown**

| SOL L1: Data Pipe Lsu Wavefronts [%] | 1.06 |
| SOL L1: Lsuin Requests [%] | 1.04 |
| SOL DRAM: Cycles Active [%] | 0.75 |
| SOL DRAM: Busy Factors [%] | 0.46 |
Roofline Chart

Floating Point Operations Roofline

- **Bottleneck**: The kernel is utilizing greater than 80.0% of the available compute or memory performance of the device. To further improve performance, work will likely need to be shifted from the most utilized to another unit. Start by analyzing workloads in the Compute Workload Analysis section.

- **Roofline Analysis**: The ratio of peak float (fp32) to double (fp64) performance on this device is 2:1. The kernel achieved 0% of this device’s fp32 peak performance and 50% of its fp64 peak performance. If Compute Workload Analysis determines that this kernel is fp64 bound, consider using 32-bit precision floating point operations to improve its performance.
Roofline Chart: Peak Values

Floating Point Operations Roofline

Double Precision Roofline

- Peak Work: 6,177,023,355,826.80
- Peak Traffic: 653,205,752,680.93
- Arithmetic Intensity [FLOP/byte]: 9.46
- Performance [FLOP/s]: 6,177,023,355,826.80
Roofline Chart: Achieved Values
Roofline Chart: Regions
Hierarchical Roofline: L1 Peak Values

High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

Floating Point Operations Roofline

<table>
<thead>
<tr>
<th>L1 Roofline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak Work:</td>
</tr>
<tr>
<td>6,168,617,760,617.75</td>
</tr>
<tr>
<td>Peak Traffic:</td>
</tr>
<tr>
<td>49,348,942,084,942.08</td>
</tr>
<tr>
<td>Arithmetic Intensity [FLOP/byte]:</td>
</tr>
<tr>
<td>0.13</td>
</tr>
<tr>
<td>Performance [FLOP/s]:</td>
</tr>
<tr>
<td>6,168,617,760,617.75</td>
</tr>
</tbody>
</table>
Hierarchical Roofline: L2 Peak Values

![Graph showing theHierarchical Roofline: L2 Peak Values

Graph showing the GPU Speed Of Light Hierarchical Roofline Chart (Double Precision).

High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

Floating Point Operations Roofline

**L2 Roofline**
- Peak Work: 6,168,617,760,617.76
- Peak Traffic: 1,755,471,594,043.02
- Arithmetic Intensity [FLOP/byte]: 3.51
- Performance [FLOP/s]: 6,168,617,760,617.76
Hierarchical Roofline: Achieved Values

GPU Speed Of Light Hierarchical Roofline Chart (Double Precision)

High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

Floating Point Operations Roofline

- **L1 Achieved Value**
  - Arithmetic Intensity [FLOP/byte]: 0.03
  - Performance [FLOP/s]: 35,300,606,729.18

- **L2 Achieved Value**
  - Arithmetic Intensity [FLOP/byte]: 0.03
  - Performance [FLOP/s]: 35,300,606,728.18

- **DRAM Achieved Value**
  - Arithmetic Intensity [FLOP/byte]: 0.06
  - Performance [FLOP/s]: 35,300,606,729.18
Roofline Analysis Example
BerkeleyGW

- Massively parallel package for GW calculations
- Sits on top of DFT codes

- Computational Motifs
  - FFTs
  - Dense linear algebra
  - Large reductions
GPP Pseudocode

do i in 1, nbands:  ! n' ~= 2763
  do j in 1, ngpown:  ! G' ~= 6633
    do k in 1, ncouls:  ! G  ~= 26529
      do h in 1, nw:      ! E  ~= 3
        compute()         ! Mixed data types:
          ! complex double, double, int
          ! Various memory access patterns
          ! Complex number divisions

        reduction() ! Complex numbers
                      ! Billions of iterations
Initial GPU Port – SOL and Roofline
Initial GPU Port – Memory Analysis

Memory Workload Analysis Chart

- Global: 4.60 G Inst, 4.60 G Req, 524.28 K Req
- Local: 0.00 Inst, 0.00 Req
- Texture: 0.00 Inst, 0.00 Req
- Surface: 0.00 Inst, 0.00 Req
- Load Global: 0.00 Inst, 0.00 Req
- Stored Shared: 14.94 M Inst, 8.65 M Req, 6.29 M Req
- L1/TEX Cache: Hit Rate: 53.56%, 8.55 MB, 561.17 GB
- L2 Cache: Hit Rate: 43.17%, 446.84 GB, 364.45 GB
- Shared Memory: 0.00 B, 0.00 B, 0.00 B
- Device Memory: 0.00 B, 0.00 B, 4.50 MB
- System Memory: 0.00 B, 0.00 B
- Peer Memory: 0.00 B, 0.00 B, 0.00 B
- L2 Compression Ratio: 0.00
Optimization Step #1: Loop Reordering

< !$ACC LOOP GANG VECTOR reduction(+:...) collapse(3)
< do i = 1, nbands ! 0(1000)
< do j = 1, ngpown ! 0(1000)
< do k = 1, ncouls ! 0(10000)
---
>
> !$ACC LOOP GANG VECTOR reduction(+:...) collapse(2)
> do j = 1, ngpown ! 0(1000)
> do k = 1, ncouls ! 0(10000)
> !$ACC LOOP SEQ
> do i = 1, nbands ! 0(1000)

Runtime: 1.17 sec → 1.10 sec

Speed-up: ~6%
Optimization Step #1: SOL

Bottom value (green) represents baseline
Optimization Step #1: Memory Analysis

Memory Workload Analysis Chart
Optimization Step #1: Roofline

Floating Point Operations Roofline

- GPU Speed Of Light Hierarchical Roofline Chart (Double Precision)

Performance [FLOP/s]

Arithmetic Intensity [FLOP/byte]
Optimization Step #2: Data Reuse

< complex(DPC) :: ssx_array_2, ssx_array_3, sch_array_2, sch_array_3

---

> complex(DPC) :: ssx_array, sch_array

- More changes to accommodate data restructuring
- Split kernel into two iterations

Runtime: 1.10 sec → 0.98 sec

Speed-up: ~11%

Total Speed-up: ~17%
Optimization Step #2: SOL

<table>
<thead>
<tr>
<th>GPU Speed Of Light</th>
<th>SOL SM [%]</th>
<th>SOL Memory [%]</th>
<th>SOL L1/TEX Cache [%]</th>
<th>SOL L2 Cache [%]</th>
<th>SOL DRAM [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>80.15 (+9.28%)</td>
<td>24.56 (+14.48%)</td>
<td>11.66 (+3.95%)</td>
<td>22.60 (-0.26%)</td>
<td>24.56 (+14.48%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>487.39 (-58.46%)</td>
<td>534,036,779 (-58.42%)</td>
<td>533,728,027.36 (-58.41%)</td>
<td>1.10 (+0.09%)</td>
<td>1.22 (+0.09%)</td>
</tr>
</tbody>
</table>

GPU Utilization

- SM [%]
- Memory [%]

Speed Of Light [%]

- 0.0
- 10.0
- 20.0
- 30.0
- 40.0
- 50.0
- 60.0
- 70.0
- 80.0
- 90.0
- 100.0
Optimization Step #2: Memory Analysis
Optimization Step #2: Roofline

![Roofline Chart](image-url)
Optimization Step #3: Arithmetic Optimization

\[
\begin{align*}
\text{< delw} &= \text{wtilde} / \text{wdiff} \\
\text{<} \\
\text{< if} (\text{abs}(ssx) .gt. ssxcutoff .and. ...) \text{ ssx=0.0d0 }
\end{align*}
\]

\[
\begin{align*}
\text{> wdiffr} &= \text{wdiff} * \text{CONJG(wdiff)} \text{ ! reciprocal math} \\
\text{> rden} &= 1.0d0 / \text{wdiffr} \\
\text{> delw} &= \text{wtilde} * \text{CONJG(wdiff)} * \text{rden} \\
\text{> rden} &= \text{ssx} * \text{CONJG(ssx)} \text{ ! replace abs with squares} \\
\text{> ssxcutoff} &= \text{sexcut}**2 * ... \\
\text{> if} (\text{rden} .gt. \text{ssxcutoff} .and. ...) \text{ ssx=0.0d0 }
\end{align*}
\]

- division → reciprocal math & abs → squares

Runtime: 0.98 sec → 0.53 sec

Speed-up: ~45%

Total Speed-up: ~54%
Optimization Step #3: SOL

<table>
<thead>
<tr>
<th>SOL SM (%)</th>
<th>87.35 (19.10%)</th>
<th>Duration [msecond]</th>
<th>265.85 (-77.34%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOL Memory (%)</td>
<td>43.78 (104.08%)</td>
<td>Elapsed Cycles [cycle]</td>
<td>2911369399 (-77.33%)</td>
</tr>
<tr>
<td>SOL L1/TEX Cache (%)</td>
<td>21.38 (90.70%)</td>
<td>SM Active Cycles [cycle]</td>
<td>290958996.19 (-77.33%)</td>
</tr>
<tr>
<td>SOL L2 Cache (%)</td>
<td>39.96 (76.34%)</td>
<td>SM Frequency [cycle/nsecond]</td>
<td>1.10 (+0.03%)</td>
</tr>
<tr>
<td>SOL DRAM (%)</td>
<td>43.78 (104.08%)</td>
<td>DRAM Frequency [cycle/nsecond]</td>
<td>1.22 (+0.03%)</td>
</tr>
</tbody>
</table>

**GPU Utilization**

- SM [%]: 87.35
- Memory [%]: 43.78

**Recommendations**

- SOL Speed of Light [%]: 87.35
- Duration [msecond]: 265.85
- Elapsed Cycles [cycle]: 2911369399
- SM Active Cycles [cycle]: 290958996.19
- SM Frequency [cycle/nsecond]: 1.10
- DRAM Frequency [cycle/nsecond]: 1.22
Optimization Step #3: Memory Analysis
Optimization Step #3: Roofline
Deoptimization Step #4: Fixed vector length

```
< !$ACC PARALLEL PRESENT(I_eps_array, aqsntemp)

vector_length(512)

> !$ACC PARALLEL PRESENT(I_eps_array, aqsntemp)

- Set the vector length to a non-optimal value
```

Runtime: 0.53 sec → 0.58 sec
Speed-up: -10%
### Step #4: SOL of v4 vs. v3

Bottom value (orange) represents optimization step #3.
Step #4: SOL of v4 vs. v2

Bottom value (pink) represents optimization step #2
Step #4: Memory Analysis of v4 vs. v3
Step #4: Memory Analysis of v4 vs. v2
# Recap of Step 4 vs. Steps 2 and 3

<table>
<thead>
<tr>
<th></th>
<th>Runtime (msec)</th>
<th>SM SOL %</th>
<th>Memory SOL %</th>
<th>L1 Cache Hit %</th>
<th>L2 Cache Hit %</th>
<th>Memory Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step #4</td>
<td>292</td>
<td>79.5</td>
<td>19.6</td>
<td>54.2</td>
<td>72.7</td>
<td>186 GB A*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>71 GB B*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>37 GB C*</td>
</tr>
<tr>
<td>Step #2</td>
<td>487</td>
<td>+0.8%</td>
<td>+25.54%</td>
<td>-6.8</td>
<td>-52.9</td>
<td>+8.2 %</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+163 %</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+366 %</td>
</tr>
<tr>
<td>Step #3</td>
<td>265</td>
<td>+9.8%</td>
<td>+123%</td>
<td>-6.2</td>
<td>-51.8</td>
<td>+7.6 %</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+158 %</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+352 %</td>
</tr>
</tbody>
</table>

A* = L1 to L2 traffic, B* = L2 partition traffic, C* = Device to L2 traffic
Step #4: Roofline of v4 vs. v3
Step #4: Roofline of v4 vs. v2
References


Acknowledgement

- This research used resources at the National Energy Research Scientific Computing Center (NERSC), which is supported by the U.S. Department of Energy Office of Science under contract DE-AC02-05CH11231.

- This research used resources at the Oak Ridge Leadership Computing Facility (OLCF) through the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program, which is supported by the U.S. Department of Energy Office of Science under Contract No. DE-AC05-00OR22725.

- This work was supported by the Center for Computational Study of Excited-State Phenomena in Energy Materials (C2SEPEM), funded by the U.S. Department of Energy Office of Science under Contract No. DEAC02-05CH11231.