Roofline Models

Lecture 13

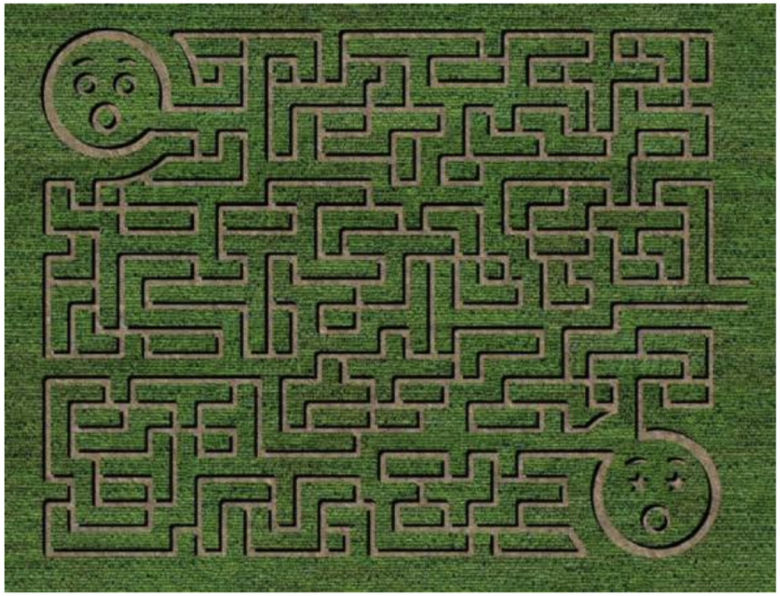
Sunita Chandrasekaran
Associate Professor, University of Delaware
PDC Summer School
Aug 2023

Q: When to use GPUs?

A: Determine if and what are the compute intensive portions of the code/program

Performance Models

NERSC



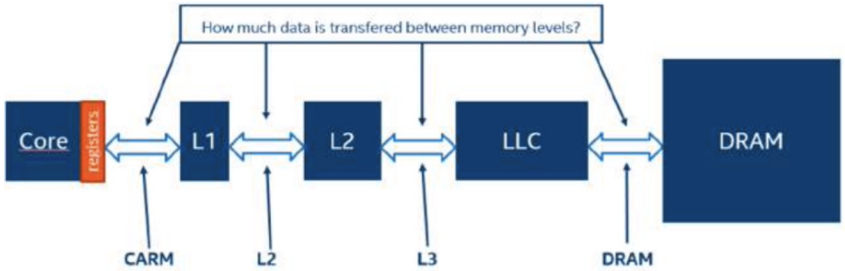
The Maze of Performance Optimization

The Map !!!

Performance Models

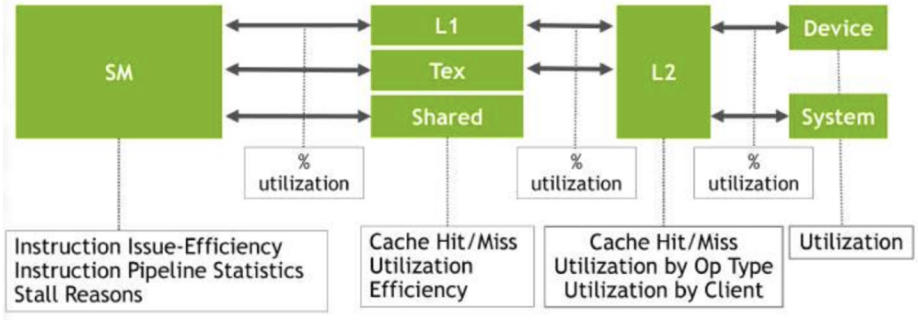
NERSC

Modern architectures are complicated!



Intel Haswell CPU[1]

NVIDIA Volta GPU[2]



1. https://software.intel.com/en-us/articles/integrated-roofline-model-with-intel-advisor
2. http://on-demand.gputechconf.com/gtc/2016/presentation/s6659-avinash-baliga-perfworks.pdf

Performance Models

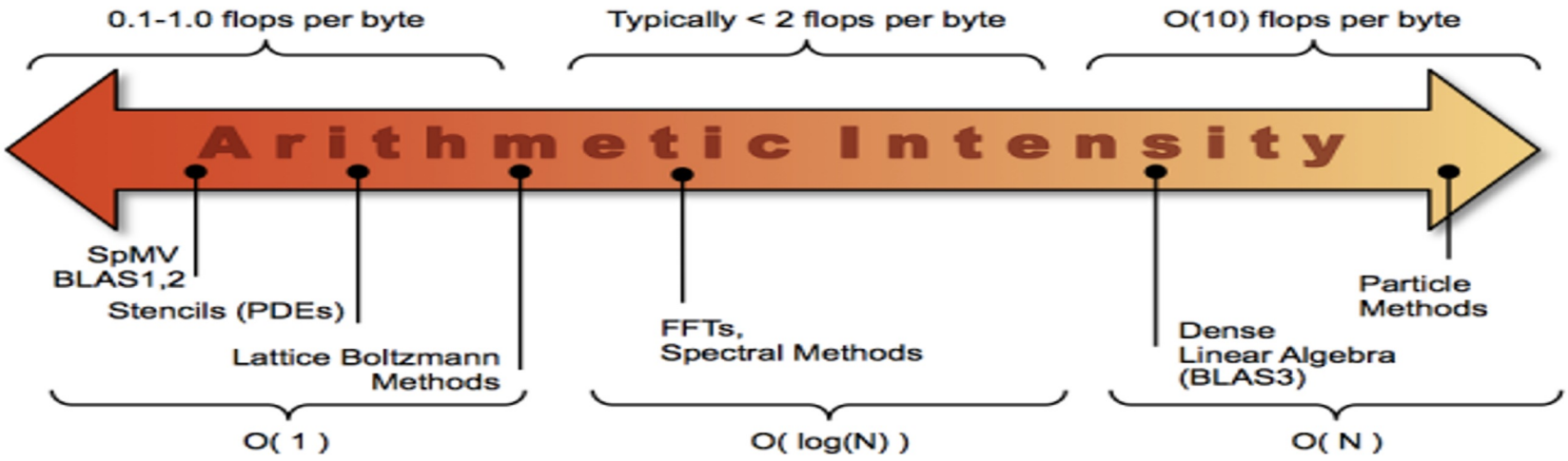
- **Many components contribute to the kernel run time**
- **An interplay of application characteristics and machine characteristics**

#FP operations	FLOP/s
Cache data movement	Cache GB/s
DRAM data movement	DRAM GB/s
PCIe data movement	PCIe bandwidth
MPI Message Size	Network Bandwidth
MPI Send:Wait ratio	Network Gap
#MPI Wait's	Network Latency
IO	File systems

Roofline Model

Roofline Model

- Core parameter of Roofline model is “arithmetic intensity”
  - ratio of floating point (math) operations to total data movement (bytes)
  - Fetch data from memory less often (share/reuse data across fragments
  - Request data less often (instead, do more math)



Why should we care about Roofline Models

- Determine when we're done optimizing code
  - Assess performance relative to machine capabilities
  - Track progress towards optimality
  - Motivate need for algorithmic changes
- Identify performance bottlenecks & motivate software optimizations
- Understand performance differences between Architectures, Programming Models, implementations, etc...
  - Why do some Architectures/Implementations move more data than others?
  - Why do some compilers outperform others?
- Predict performance on future machines / architectures o Set realistic performance expectations
  - Drive for Architecture-Computer Science-Applied Math Co-Design

Roofline Performance Model

NERSC

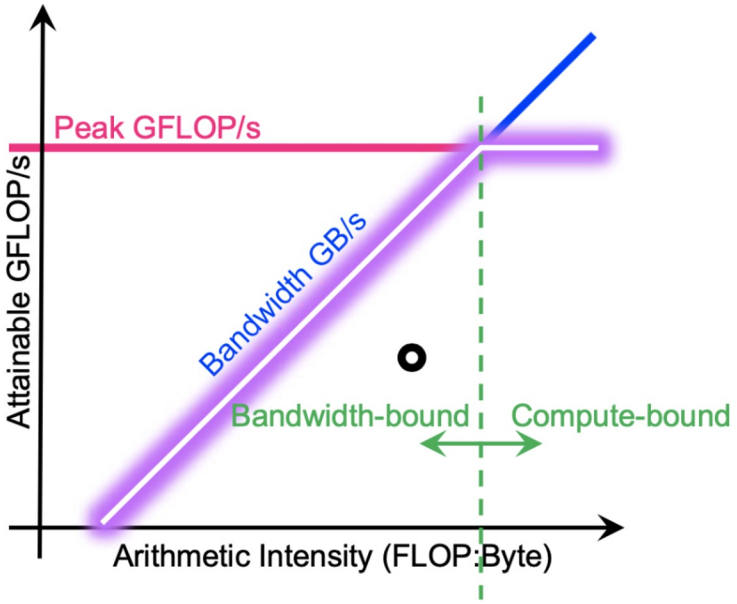
- Thus we obtain the model as

$$\text{GFLOP/s} = \min \begin{cases} \text{Peak GFLOP/s} \\ \text{AI} * \text{Peak GB/s} \end{cases}$$

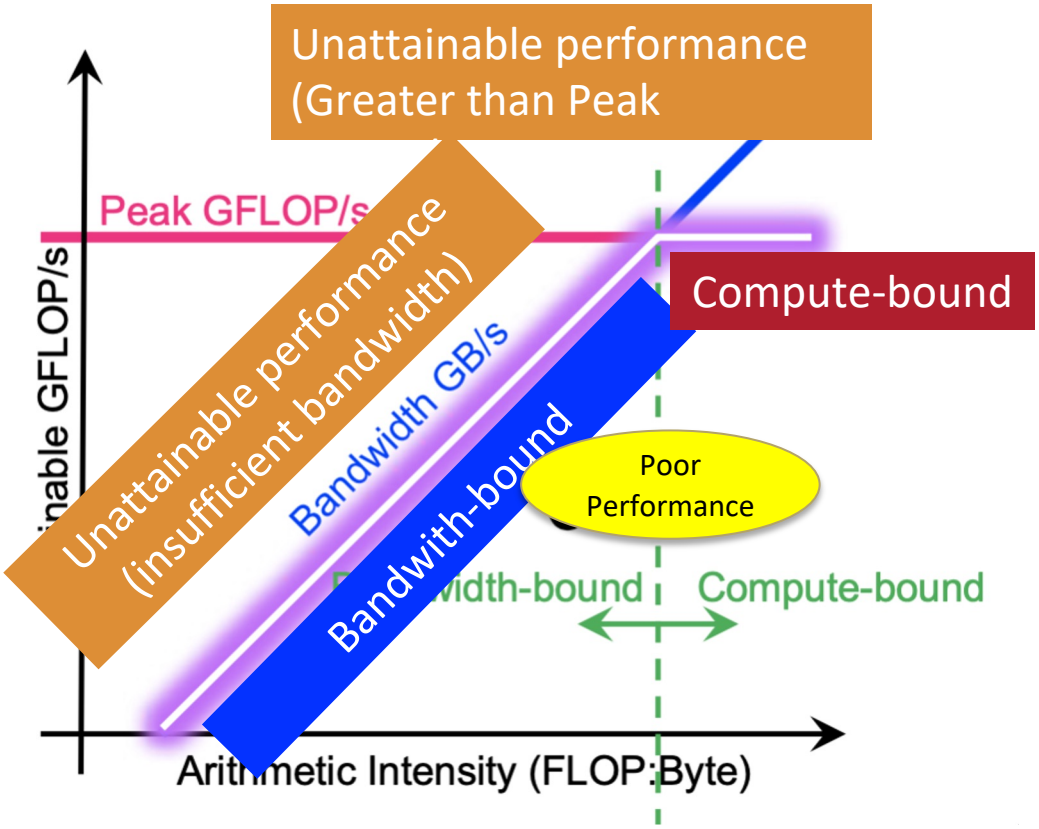
where Arithmetic Intensity (AI) is

FLOPs / Bytes

- Machine Balance (FLOPs/Byte) =
  8.9 (V100, DP, HBM) or 5.1 (KNL, DP, HBM)



Roofline Model



What is Arithmetic Intensity?

- Measure of data locality (data reuse)
- Ratio of **Total Flops** performed to **Total Bytes** moved
- For the DRAM Roofline…
  - Total Bytes to/from DRAM
  - Includes all cache and prefetcher effects
  - Can be very different from total loads/stores (bytes requested)
  - Equal to ratio of sustained GFLOP/s to sustained GB/s (time cancels)

What is bandwidth?

- According to Little's Law
  - effective application bandwidth is directly proportional to the number of outstanding memory requests and inversely proportional to memory access latency

$$effective\ bandwidth \propto \frac{outstanding\ memory\ requests}{memory\ access\ latency}$$

- What is outstanding memory requests?
  - Properties of the application (such as the portion of memory accesses in the overall instruction mix and data and control dependencies) and the CPU (such as core count, out-of-order issue, speculative execution, branch prediction, and prefetching)

- Do you think 3D-stacked DRAM will help in this situation?

How do you calculate bandwidth?

- Bandwidth = Memory Frequency x $\frac{Bus\ width}{8}$) x operations/cycle

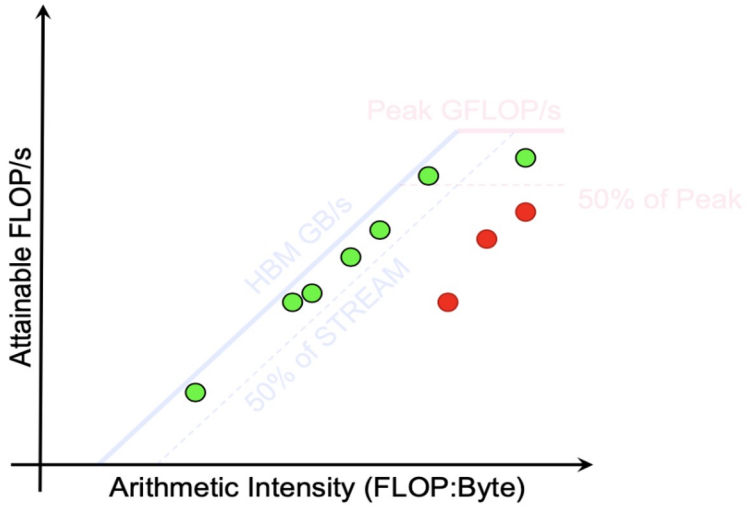
(Divide by 8 to change from BIT to BYTE)

- Frequency = 800Mhz
- Bus width = 128 bits
- No. of operations per clock cycle = 2 or 4 or ... (add/multiple)

$800 * 10^6 \times \frac{128}{8} \times 2 = 800 * 10^6 \times 16 \text{ bytes} \times 2 = 25600 \text{ MB/s}$

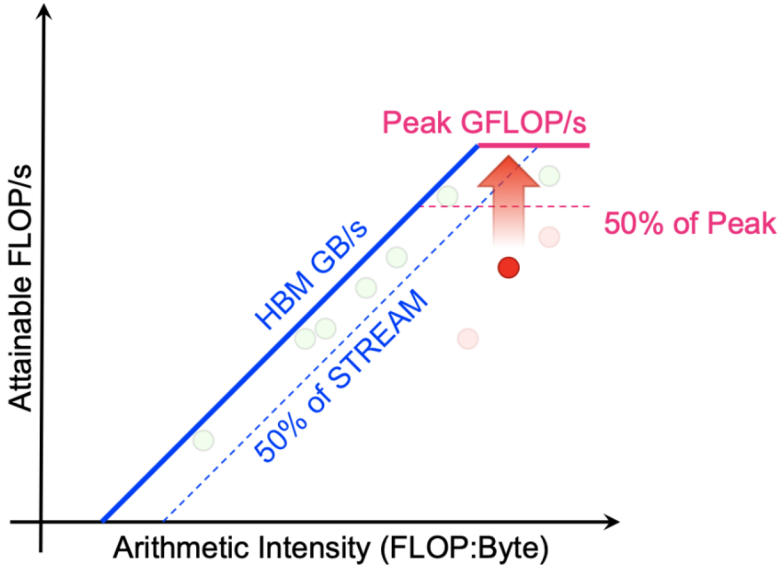
2 components that makes a roofline model

- Machine Model
  - Lines defined by peak GB/s and GF/s (**Benchmarking**)
  - Unique to each architecture
  - Common to all apps on that architecture
- Application Characteristics
  - Dots defined by application GFLOP's and GB's (**Application Instrumentation**)
  - Unique to each application
  - Unique to each architecture



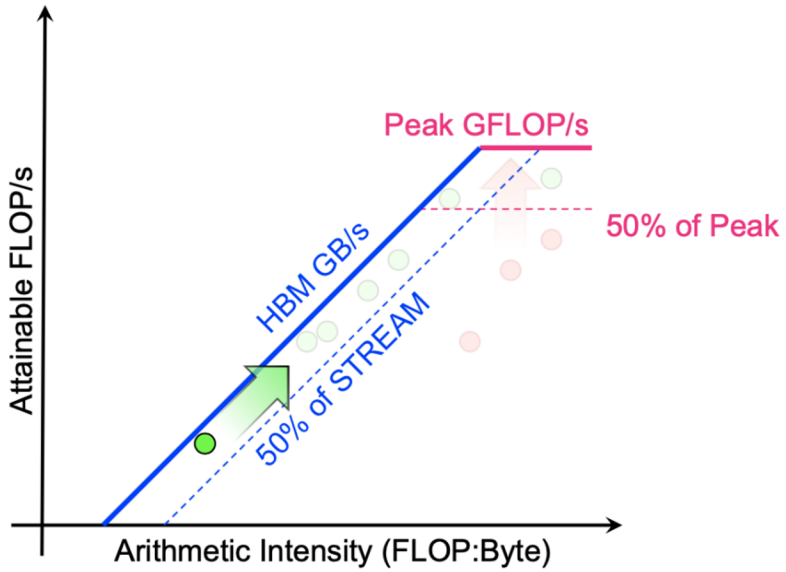
General Performance Optimization Strategy

- Get to the Roofline



General Performance Optimization Strategy

- Get to the Roofline
- Increase Arithmetic Intensity when bandwidth-limited
  - Reducing data movement increases AI



Performance Below the Roofline?

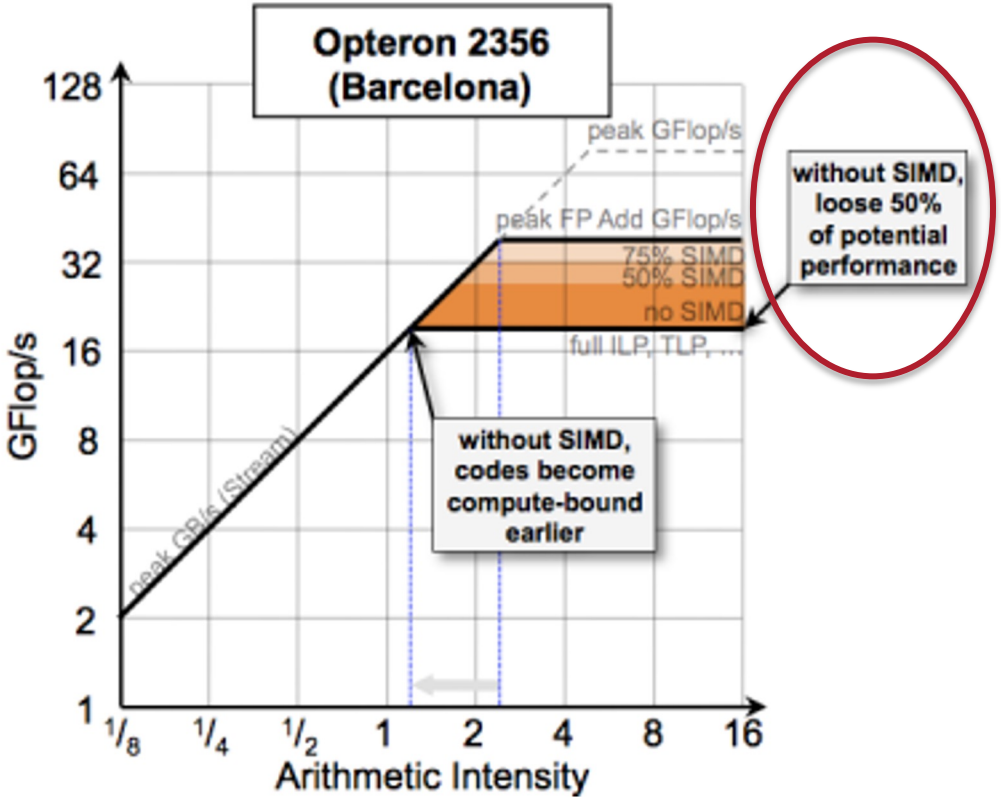
- Insufficient cache bandwidth and data locality
- Instruction Mix
  - Lack of FMA
  - Mixed Precision effects
  - Lack of Tensor Core operations
- "Lack of Parallelism"
  - Thread Divergence (idle threads)
  - Insufficient Occupancy (idle warp sched)
  - Insufficient #Thread Blocks (idle SMs)
- Integer-heavy Codes
  - Non-FP instructions impair FP performance
  - No FP instructions… AI=0

Data Level Parallelism for arithmetic intense operations

```
for(i=…){
 sum0+=b[i  ];
 sum1+=b[i+1];
 sum2+=b[i+2];
 sum3+=b[i+3];
}
```

```
for(i=…){
 sum0=_mm_add_sd(sum0,…b[i  ]…);
 sum1=_mm_add_sd(sum1,…b[i+1]…);
 sum2=_mm_add_sd(sum2,…b[i+2]…);
 sum3=_mm_add_sd(sum3,…b[i+3]…);
}
```

```
for(i=…){
 sum01=_mm_add_pd(sum01,…b[i  ]…);
 sum23=_mm_add_pd(sum23,…b[i+2]…);
 sum45=_mm_add_pd(sum45,…b[i+4]…);
 sum67=_mm_add_pd(sum67,…b[i+6]…);
}
```

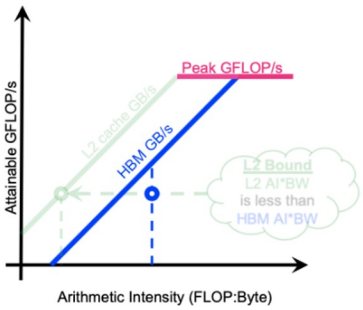


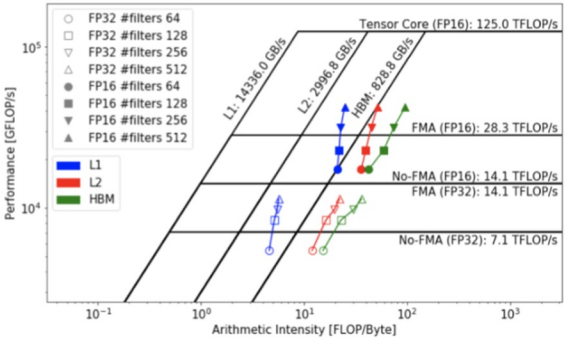
https://crd.lbl.gov/departments/computer-science/PAR/research/roofline/

Performance Below the Roofline?

Hierarchical Roofline Model

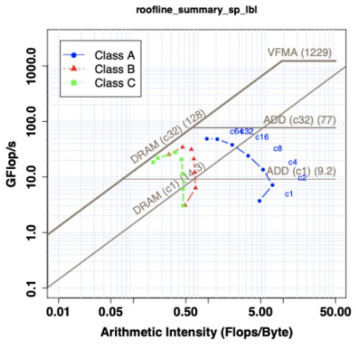
Charlene Yang, Thorsten Kurth, Samuel Williams, "Hierarchical Roofline analysis for GPUs: Accelerating performance optimization for the NERSC-9 Perlmutter system", Concurrency and Computation: Practice and Experience (CCPE), August 2019.





Additional FP Ceilings

Charlene Yang, Thorsten Kurth, Samuel Williams, "Hierarchical Roofline analysis for GPUs: Accelerating performance optimization for the NERSC-9 Perlmutter system", Concurrency and Computation: Practice and Experience (CCPE), August 2019.

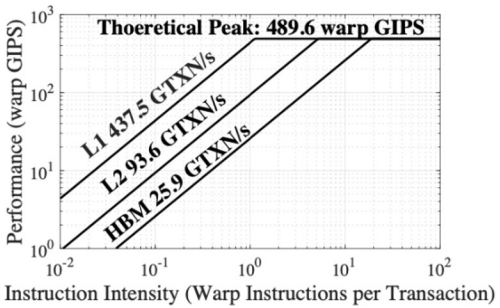


Roofline Scaling Trajectories

Khaled Ibrahim, Samuel Williams, Leonid Oliker, "Performance Analysis of GPU Programming Models using the Roofline Scaling Trajectories", International Symposium on Benchmarking, Measuring and Optimizing (Bench), BEST PAPER AWARD, November 2019.

Instruction Roofline Model

Nan Ding, Samuel Williams, "An Instruction Roofline Model for GPUs", Performance Modeling, Benchmarking, and Simulation (PMBS), BEST PAPER AWARD, November 2019.

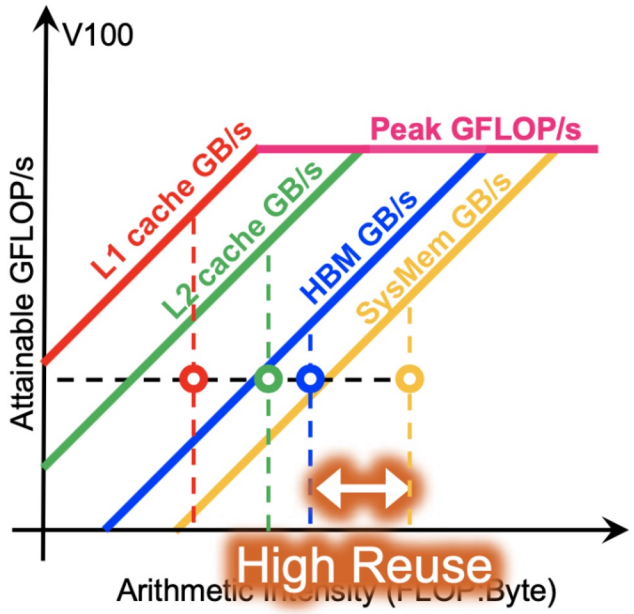


Hierarchical Roofline

NERSC

- **Superposition of multiple Rooflines**
  - **Incorporate full memory hierarchy**
  - **Arithmetic Intensity =**

    FLOPs / Bytes$_{\text{L1/L2/HBM/SysMem}}$



- **Each kernel will have multiple AI's but one observed GFLOP/s performance**

- **Hierarchical Roofline tells you about cache locality**