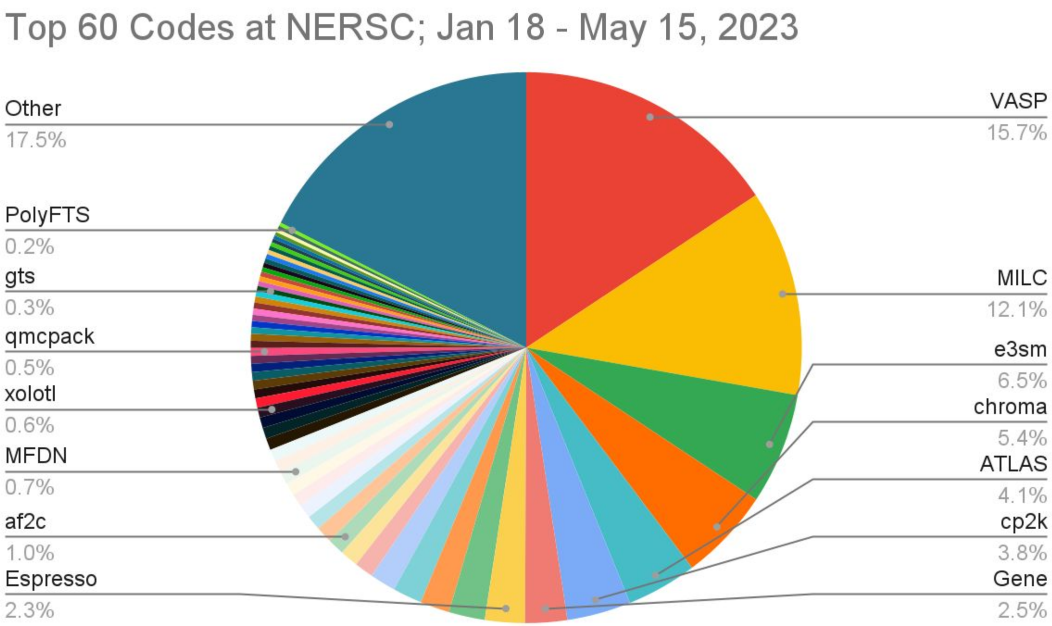
System-Wide Roofline Profiling - A Case Study on NERSC's Perlmutter Supercomputer

Brian Austin, Dhruva Kulkarni, Brandon Cook, Samuel Williams, Nicholas Wright
Lawrence Berkeley National Laboratory

PMBS24, held in conjunction with SC24
Atland GA
November 18, 2024

SC24
Atlanta, GA | hpc creates.

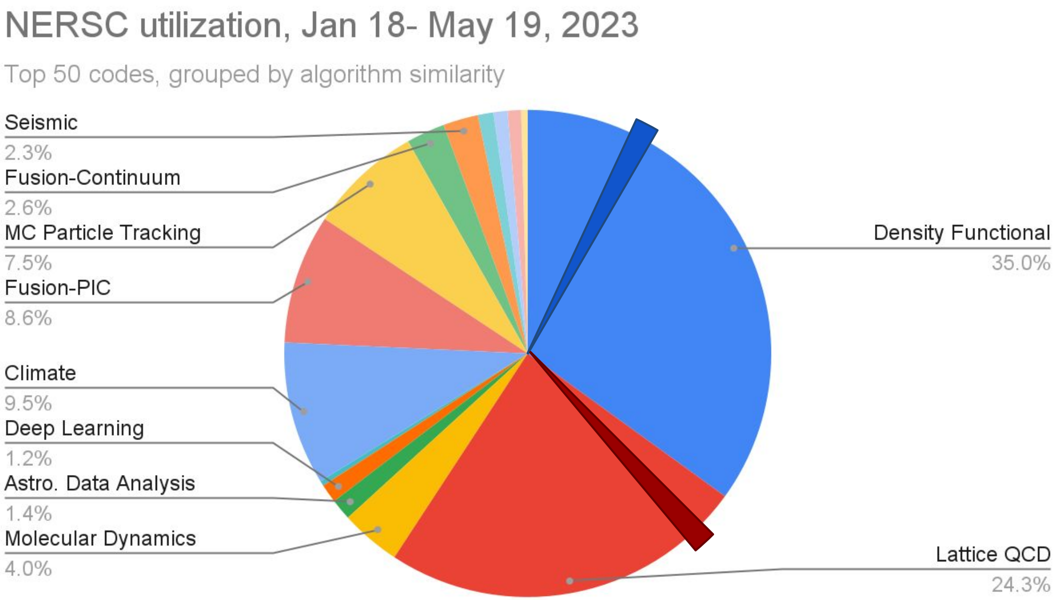
How do we describe HPC workloads to help system architects make better design choices?



<u>It is not a simple task</u>

- HPC workloads can be extremely diverse. For example, NERSC hosts:
  - 850 projects
  - 800 distinct codes
  - 11,000 users → 11,000 uses
- Each use-case may have different performance sensitivities.

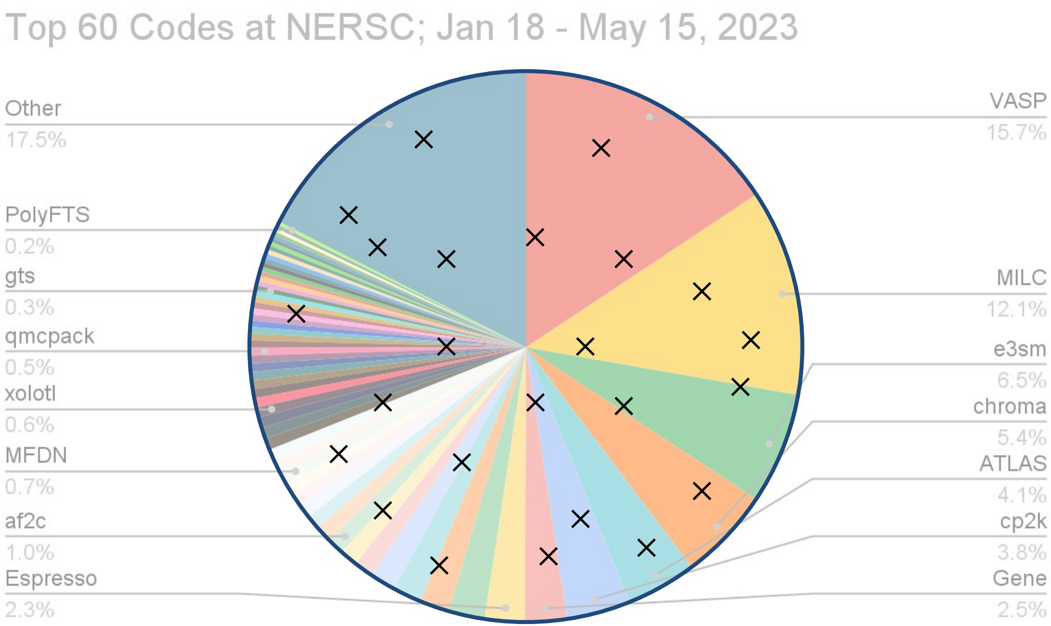
How do we describe HPC workloads to help system architects make better design choices?



Current best practice

- Carefully selection of "representative" jobs.
  - Two benchmarks can represent 60% of the workload !?
  - No information about the remaining workload
- Deep analysis of selected jobs
  - Analysis is resource intensive
  - Not easily scaled to other jobs.

How do we describe HPC workloads to help system architects make better design choices?



<u>This work: System-wide sampling</u>

- Limited performance counter selection
- Full coverage of entire workload
- Eliminates selection bias and extrapolation error
- Low sampling rates
- No insight into individual codes
- Limited to simple performance models

Outline

- **System and Monitoring Infrastructure**

What is the prevailing floating-point precision used at NERSC?

- **Roofline Performance Model**

Is the performance of NERSC's workload typically bound by FLOPs or bandwidth?

- **NERSC-10 Benchmarks**

How well does this benchmark suite reflect the FLOP/Byte ratio of the workload?

Perlmutter - an HPE Cray EX System

- 1,536 GPU accelerated nodes
  - 1x AMD Milan CPU
  - 4x NVIDIA A100 GPUs with 40 GB HBM
- 3,072 CPU nodes
  - 2x AMD Milan CPUs
- Slingshot 11 interconnect
- 35 PB all Flash Lustre file system
- Later added 346 GPU nodes with 80 GB HBM

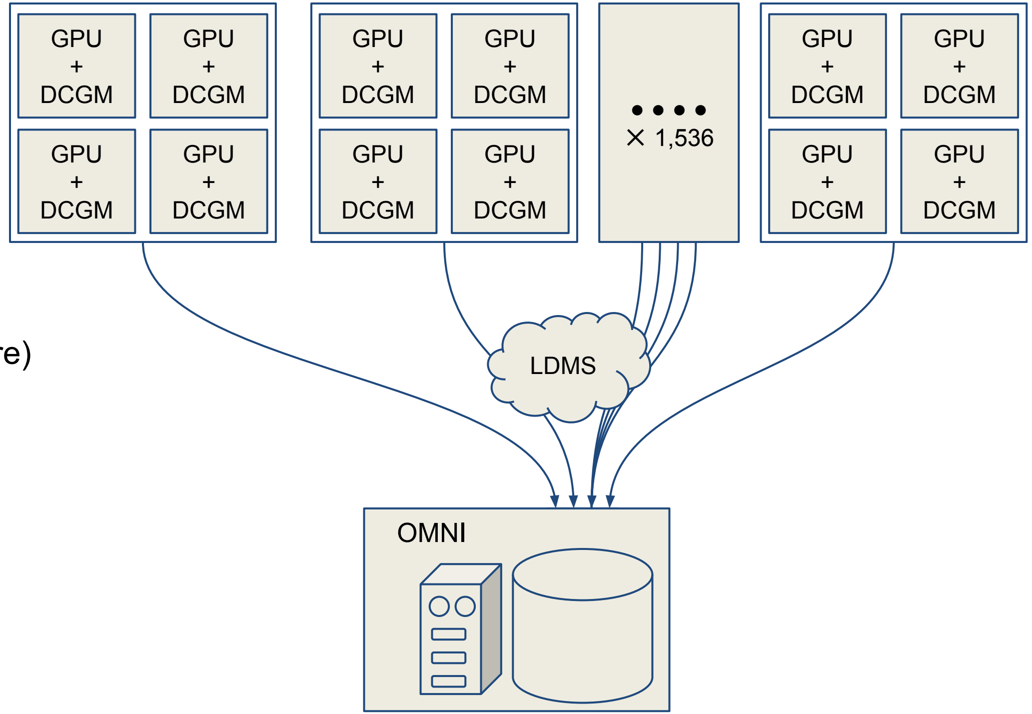
Data Acquisition

Collection Pipeline

- On-node metrics sampled using NVIDIA DCGM (Data Center GPU Manager)
- Cross-system aggregation using LDMS (Lightweight Distributed Metrics System)
- Stored in NERSC's OMNI (Operations Monitoring and Notification Infrastructure)

Volume

- All 1,536 40GB A100 GPU nodes
- Sampled at 1-second intervals
- Entire month of July, 2024
- ≈ 16 Billion samples, each corresponds to one GPU-second

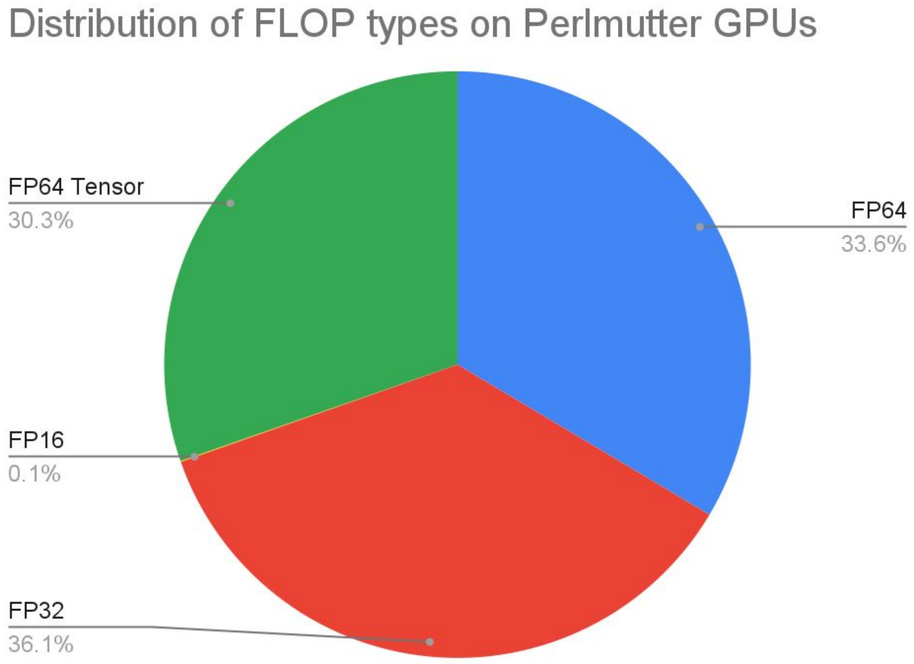


GPU Metrics Collected

Feature	A100 Peak Performance	DCGM Metric	Description
FP16	78 TF/s	`fp16_active`	The fraction of cycles the FP64/32/16/Tensor pipes were active. $FLOPS_{FP64} = \texttt{fp64\_active} \times Peak_{FP64}$
FP32	19.5 TF/s	`fp32_active`	
FP64	9.7 TF/s	`fp64_active`	
FP64 Tensor	19.5 TF/s	`tensor_active`	
HBM	1.555 TB/s	`dram_active`	The fraction of cycles where data was sent to or received from device memory. $Bytes_{HBM} = \texttt{dram\_active} \times Peak_{HBM}$

Each metric value represents an average over a time interval (i.e. our 1 second sampling period) and is not an instantaneous value.

What is the prevailing floating-point precision used at NERSC?



- Double precision (FP64) FLOPS are twice as common as single precision (FP32) FLOPS.
- Half of the FP64 FLOPS run on tensor cores.
  - All tensor activity attributed to FP64
  - Tensor cores support TF32, but not FP32
  - No corresponding non-tensor FP16
- Half precision (FP16) is rarely used

Outline

- **System and Monitoring Infrastructure**

What is the prevailing floating-point precision used at NERSC?

- **Roofline Performance Model**

Is the performance of NERSC’s workload typically bound by FLOPs or bandwidth?

- **NERSC-10 Benchmarks**

How well does this benchmark suite reflect the FLOP/Byte ratio of the workload?

Roofline Performance Model

Hypothesis

- Kernel performance is limited by either:
  a) the rate of executing operations, e.g. FLOP/s (“compute-bound”), or
  b) the rate of transferring operands to the cores (“memory-bound”)

Processor Characteristics

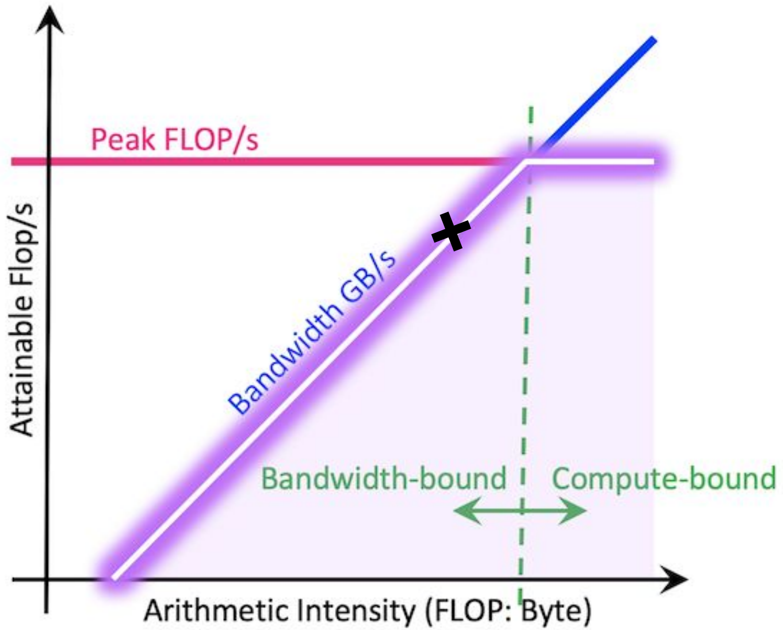
- Peak floating point performance
- Peak memory bandwidth
- → Attainable Performance Ceiling

Kernel Characteristics

- $\text{Arithmetic Intensity} = \dfrac{\text{Number of FLOPs executed}}{\text{Number of bytes transferred to/from memory}}$

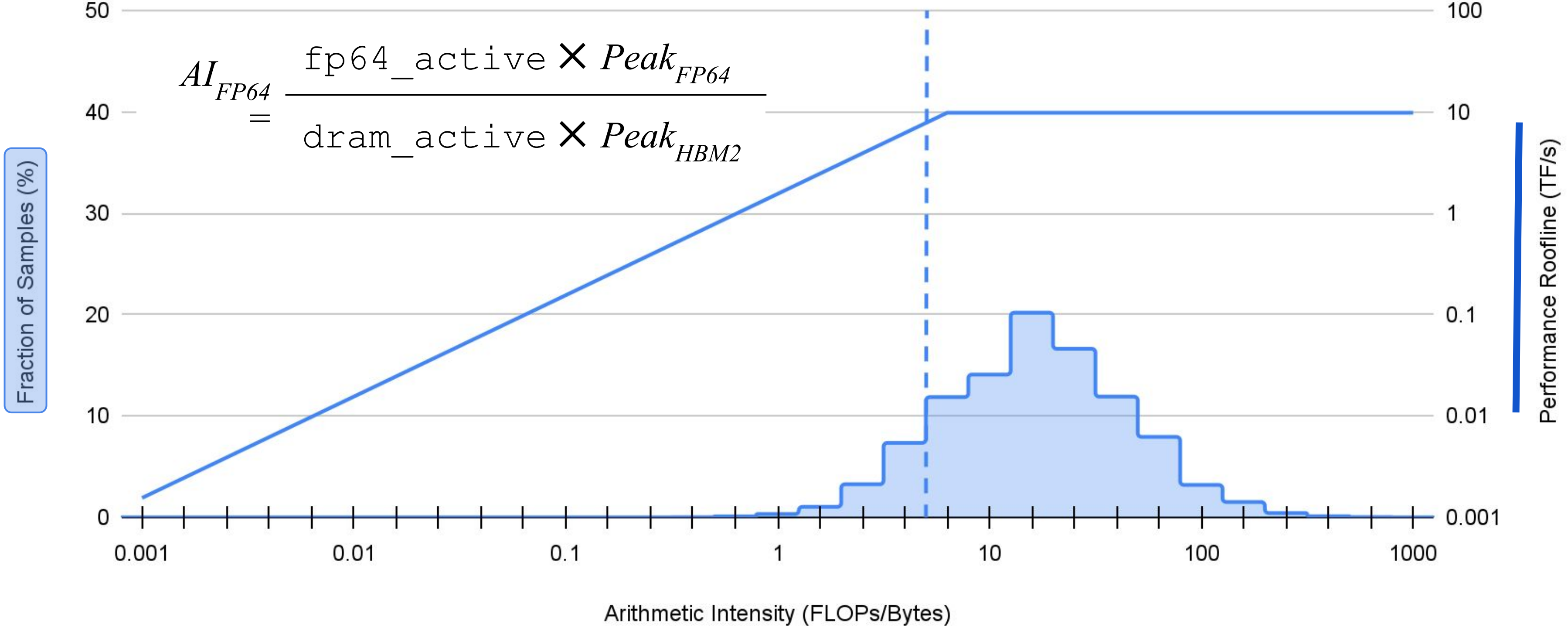
Kernel Performance

- $\text{Performance} = \min \begin{cases} \text{Arithmetic Intensity} \times \text{Peak Bandwidth}, \\ \text{Peak FLOP/s} \end{cases}$



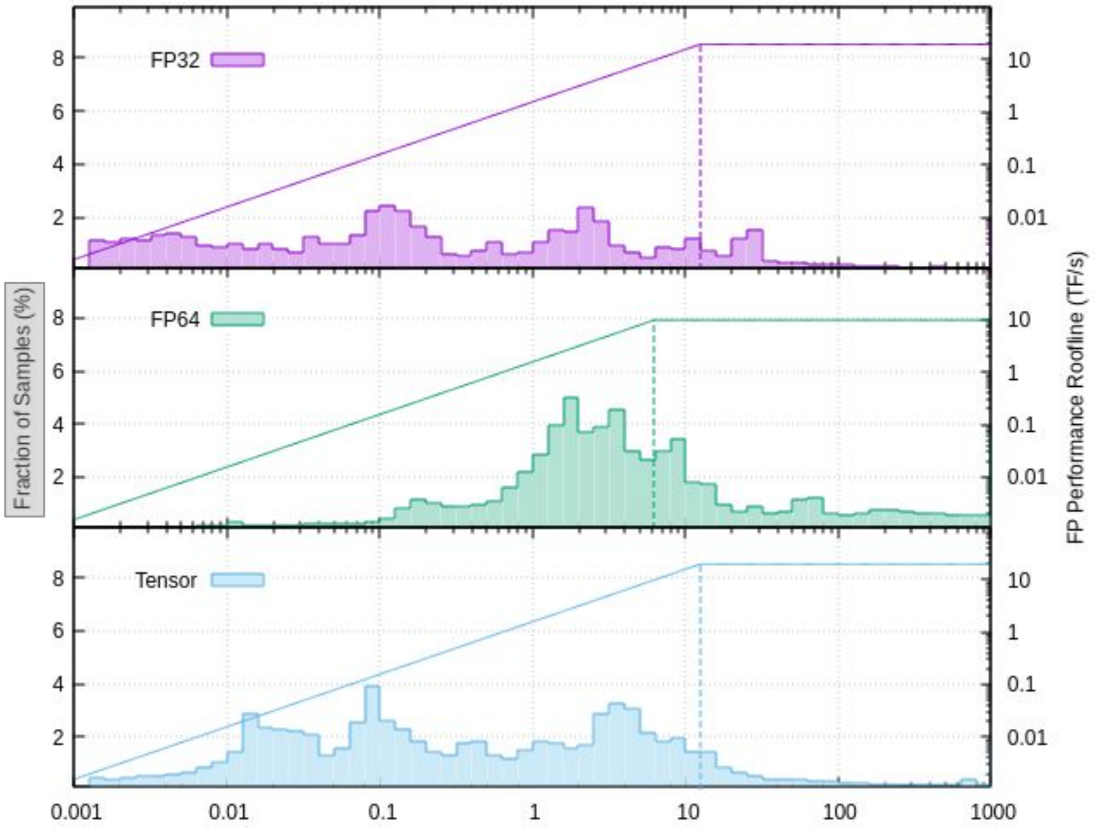
Arithmetic intensity distributions are easily computed from DCGM metrics

$$AI_{FP64} = \frac{\texttt{fp64\_active} \times Peak_{FP64}}{\texttt{dram\_active} \times Peak_{HBM2}}$$



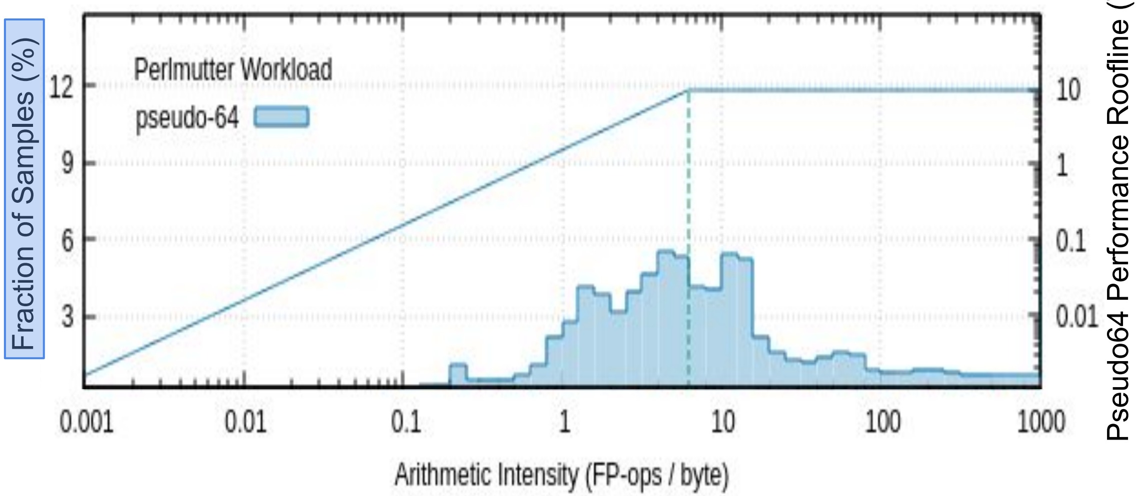
Full-system Arithmetic Intensity Distributions

- FP32:
  - Almost always memory-bound
    Median = 0.06 FLOPs/ byte
- FP64:
  - Long tail of high intensity
    Median = 3.2 FLOPs/byte
- FP64 Tensor:
  - Median = 0.2 (why?!)
- All precisions:
  - The majority of samples have AI values substantially below the machine balance.



Is the performance of NERSC's workload typically bound by FLOPs or bandwidth?

- Introducing a "pseudo64" FLOP type
  - Needed to compute a compute a single arithmetic intensity from multiple FLOP types
  - $FLOPS_{Pseudo64} = 1 \times FLOPS_{FP64} + ½ \times FLOPS_{FP32} + ¼ \times FLOPS_{FP16} + 1 \times FLOPS_{Tensor64}$
- Median pseudo64 Arithmetic Intensity: 7.5 FLOPS/Byte
- On Perlmutter's A100 GPUs,
46% of cycles memory-bound
54% are compute-bound.



Outline

- **System and Monitoring Infrastructure**

  What is the prevailing floating-point precision used at NERSC?

- **Roofline Performance Model**

  Is the performance of NERSC’s workload typically bound by FLOPs or bandwidth?

- **NERSC-10 Benchmarks**

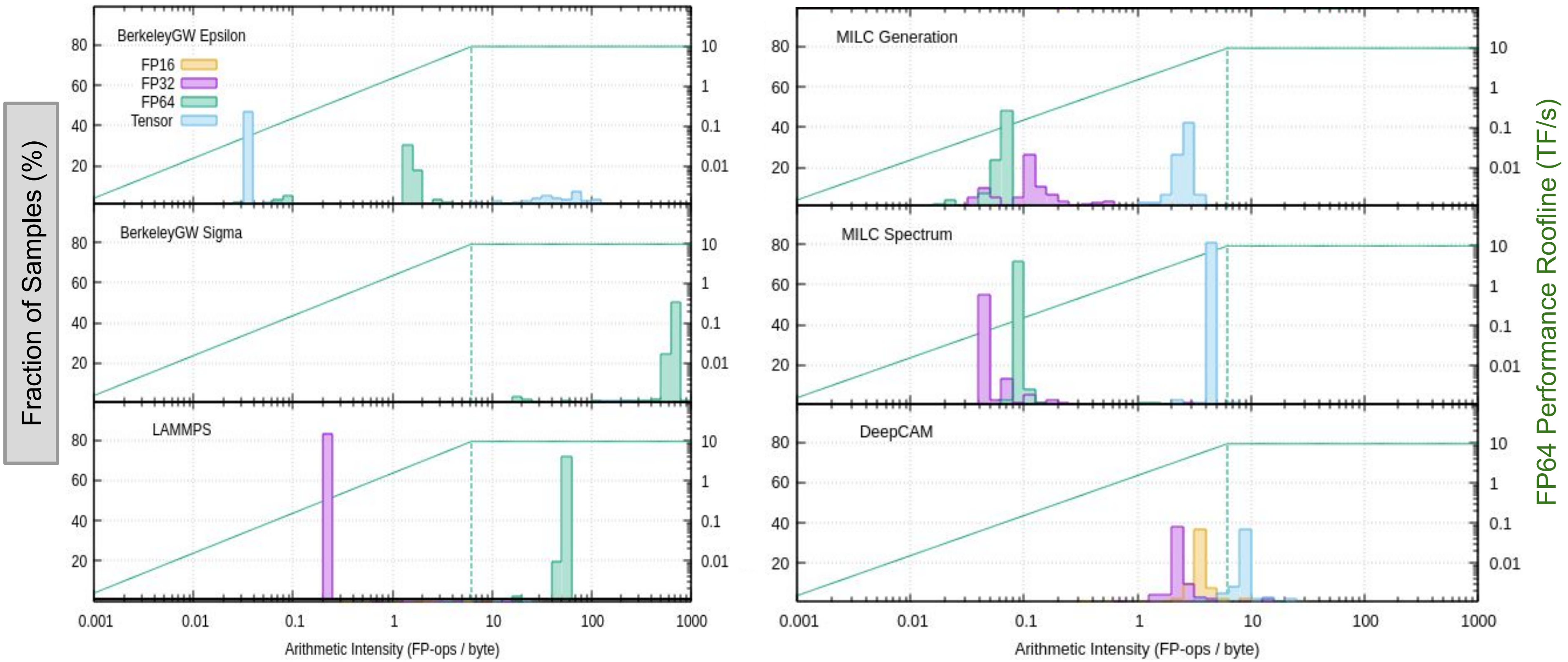
  How well does this benchmark suite reflect the FLOP/Byte ratio of the workload?

NERSC-10 Benchmark Suite

- Cross section of NERSC's workload
  - Spans many axes of computational diversity
  - Six GPU codes + two CPU codes
- Profiled configuration
  - "Small" problem size
  - Ran using 4 GPUs on 1 node

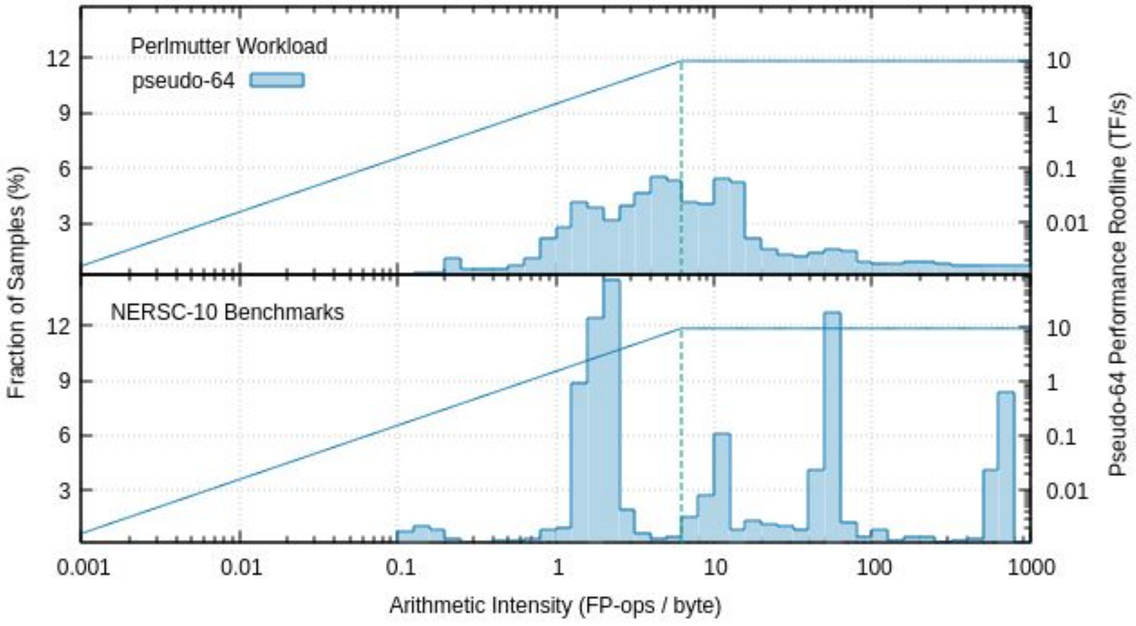
PRODUCTION WORKFLOW	Algorithm / Domain	Workflow Benchmark Tasks	Language	GPU enabled?	I/O
Lattice QCD	Lattice QCD	MILC configuration MILC analysis	C OpenMP QUDA / QPhiX (optional) MPI	Yes	MPIIO
Optical Materials	Density Functional Theory	BerkeleyGW epsilon BerkeleyGW sigma	FORTRAN OpenMP-offload or OpenACC MPI	Yes	HDF5
Materials by Design	Molecular Dynamics	LAMMPS	C++ Kokkos MPI	Yes	minimal
Climate Simulation & Analysis	Deep Learning Training	DeepCAM training	PyTorch	Yes	HDF5
CMB-S4	Cosmology	TOAST	Python front-end C++ back-end MPI4py	No	Posix FPP; FITS format
Metagenome Annotation	Genomics	HMMSearch	C OpenMP	No	Posix FPP

Diversity of NERSC-10 benchmarks is reflected by their Arithmetic Intensity Distributions



How well does the NERSC-10 benchmark suite reflect the FLOP/Byte ratio of the workload?

- Same range of arithmetic intensities
- Same 50/50 balance of memory- and compute bound samples
- Effects of using a finite suite are clearly visible

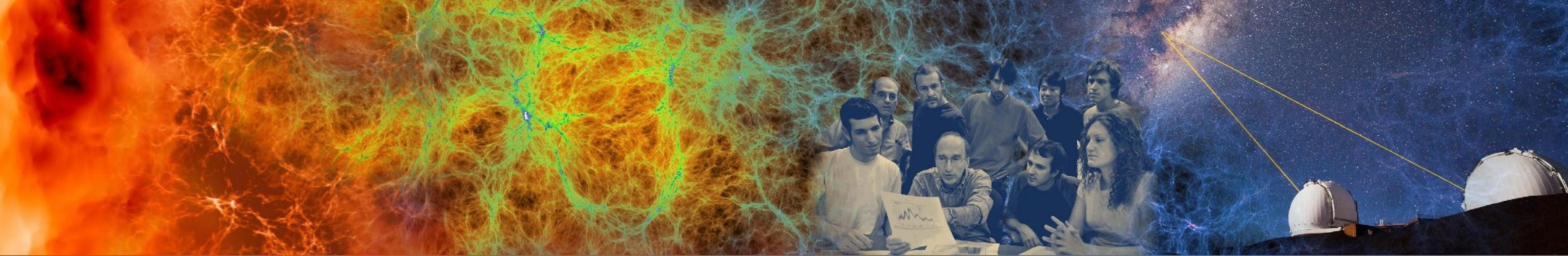


Conclusion

- First of a kind analysis using full-system sampling to understand the performance characteristics of an entire supercomputer workload, revealing:
  - Distribution of FLOP types: ⅔ FP64, ⅓ FP32, <0.1% FP16
  - Distribution of arithmetic intensities:
    Median = 7.5 pseudo-64 FLOPs/byte
  - On A100 GPUs, ½ of cycles are memory- bound,
    ½ are compute bound.
- The NERSC-10 benchmarks replicate the Perlmutter's overall balance of memory- and compute-bound samples, but the effects of using a finite suite are clearly visible in the shapes of the arithmetic intensity distribution.
- Full-system sampling and traditional performance modeling are complementary approaches to understanding architectural trade-offs.
- Results show today are preliminary.
  Many refinements, extensions & experiments will follow !

System-Wide Roofline Profiling - A Case Study on NERSC's Perlmutter Supercomputer

https://tinyurl.com/29j93uk3



Thank You

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC award ASCR-ERCAP0026875. ROR: https://ror.org/05v3mvq14