Information Roofline: A Model for the Era of Datatype Optimization and Emulation

Max Hawkins
mhawkins60@gatech.edu

Rich Vuduc
richie@cc.gatech.edu

***Abstract*—By modeling performance in physical bits and undifferentiated floating-point operations, the classical roofline model implicitly assumes every bit is needed and every operation performs comparable work. These assumptions held when IEEE-754 64-bit precision dominated the computational landscape, but the rise of AI has brought mixed-precision workloads, block-scaled formats, and emulation schemes that break these assumptions and complicate analysis. We propose an information-theoretic extension of the roofline model to help developers reason about performance and hardware utilization in this new era. By quantifying the throughput of both communication and computation channels with bits of mutual information, we aim to expose underutilized datatypes and more fairly characterize the performance of emulated operations.**

I. Introduction

As datatypes and emulation become first-class dimensions of hardware/software co-design, performance models must measure what those datatypes and operations actually accomplish. To do so, we build on recent information-theoretic interpretations of computation [1] and present the information roofline: an information-centric roofline model that measures performance as information across a communication or computation channel.

An information-centric analysis is similar to the idea that although a gallon jug (*physical* bit) can hold up to a gallon of water (*informational* bit), the jug could contain far less than a gallon or even no water at all. To maximize throughput, we suggest that modeling tools measure the information content rather than the container's capacity. We may be far from the informational limits of our computing hardware, and making these gaps visible is the first step toward closing them.

A simple demonstration of this idea is a memory-bound kernel that communicates and computes on unsigned, 8-bit integers (UInt8) stored in FP64 containers. If properly optimized under a model that measures communication throughput in physical bits, the peak performance will sit just under the memory bandwidth bound, indicating no further memory optimizations to perform. This analysis is correct for physical throughput but offers no insight into the information those bits carry. In fact, most of the bits in the datatype are unneeded and can be removed without downstream effects. Under our proposed performance model, such a kernel would achieve at best 12.5% information bandwidth utilization. This example highlights that an information-centric analysis considers the information utilization of the datatypes used, potentially signaling an opportunity to leverage modern innovations in datatypes and emulation schemes.

Further, the information roofline applies this notion of information throughput not just to memory movement (communication channels) but also to computation. Although memory channels are often reported in physical bits (rather than "elements"), compute channels still rely on the generic unit of (floating-point) "operations." Our model analyzes computing performance in atomic units of informational bits and captures the same notions of information utilization for arbitrary operations, not just the identity operation that communication performs. This measure captures the inefficiency of adding two UInt8 variables in an FP64 pipeline. A physical bit that carries no information signals an opportunity for optimization regardless of whether it's transmitted through a memory bus or through a compute pipeline.

II. Information Roofline Model

Our model extends the classic roofline [2] by incorporating Shannon's information theory. Rather than treating a variable as a single value, we treat it as a random variable X with a distribution observed across kernel calls or GPU threads. Its Shannon entropy, $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$, quantifies the uncertainty of the variable. A constant has zero entropy, and a variable with uniform probability over all 2^N patterns of an N-bit datatype has $H(X) = N$ bits (the maximally "utilized" case bit-width scaling assumes). Most application data falls between these two extremes, and the gap between $H(X)$ and the datatype width is the underutilization we aim to expose.

We quantify the work of an operation with output Y and inputs X using mutual information: $I(X;Y) = H(Y) - H(Y \mid X) = H(X) - H(X \mid Y)$. The conditional entropy terms penalize hardware noise and capture the information lost when many-to-one operations map distinct inputs to the same output. Maximizing mutual information over input distributions yields channel capacity, $C = \max_{p(x)} I(X;Y)$, and multiplying by the hardware channel's operating frequency, f_{op}, converts per-use information to an information rate $\dot{I} = I(X;Y) \cdot f_{\text{op}}$.

The y-axis is information rate $\dot{I}$ (bits/s) for both communication and computation, and the x-axis becomes information intensity (Ω): the unitless ratio of bits computed to bits communicated. The resulting information roofline retains the familiar shape, and a single roofline is made across all

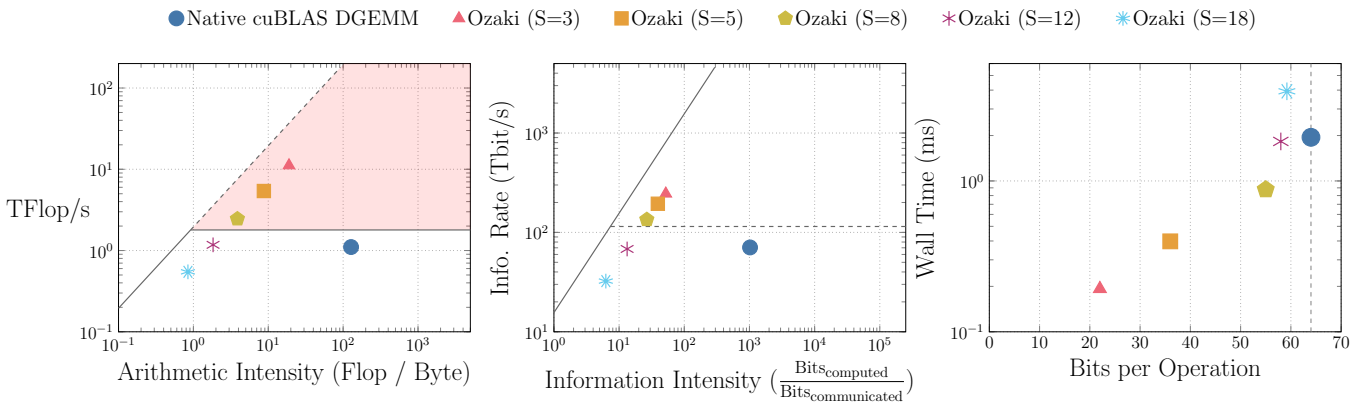

Fig. 1. The operation roofline (left) reports all Ozaki configurations as equal FP64 operations, collapsing precision/range differences and showcasing previously 'unobtainable' FP64 performance shaded in red. The information roofline (center) captures these differences and results in vertically condensed emulation points. The accuracy/range/throughput spectrum (right) makes these tradeoffs explicit and captures the non-linear relationship between the number of Ozaki splits (S) and the resulting bits per operation due to mapping integers to float encodings. All data is collected from an NVIDIA RTX PRO 6000 Blackwell with $L = M = K = 1024$ GEMM.

available computing pipelines $d \in D$ (e.g. FP64 vector units or FP8 matrix cores):

$$\text{Perf.} = \dot{I} \leq \min\left(\dot{I}_{\max_{\text{memory}}} \cdot \Omega,\ \max_{d \in D}\left(\dot{I}_{d,\max_{\text{compute}}}\right)\right) \quad (1)$$

A kernel's gap below this ceiling now reflects both operational underutilization (wasted clock cycles) *and* datatype-/emulation inefficiency (wasted range/precision/sign bits of the datatype). Following Shannon, the model is deliberately semantics-agnostic to generalize across applications: it considers states, not their meaning or downstream error.

III. Ozaki Emulation Case Study

Emulation schemes unite traditional high-performance computing workloads with low-precision AI hardware by mimicking high-precision computations with many smaller bit-width operations. The Ozaki-I scheme [3] is one such approach and performs emulated floating-point matrix multiplication by splitting each floating-point value into S integer slices forming a logically wider integer. When two floats are multiplied with emulation, each slice of one float is multiplied with each slice of the other (ignoring implementation-specific optimizations), so a single logical floating-point operation costs $O(S^2)$ sub-type operations. If the hardware device has sufficiently more performant sub-type throughput than the emulated type's throughput (e.g. Int8 vs FP64 performance on many modern GPUs), this can result in throughput improvements.

The Ozaki emulation implementation we tested, OzIMMU, uses 8-bit signed integer sub-types but each Int8 slice only adds 7 bits to the wide integer. Figure 1 shows native cuBLAS DGEMM and emulated variants' performance on an NVIDIA RTX PRO 6000 Blackwell plotted with the traditional roofline model (left) and our proposed information roofline model (center). While the traditional roofline counts every logical FP64 operation equally (regardless of number of sub-type slices), the information roofline accounts for these differences by considering the information each configured operation delivers (bits of information per operation). The rightmost plot shows how this metric varies with S: bits per operation rises with the number of slices. The growth is non-linear because each added slice contributes mantissa precision linearly whereas contributing dynamic range costs exponentially more bits (inherent in mapping integers to floats and results in poor emulated performance for large dynamic range data [3]). This variation in bits per operation drives the relative performance differences between the two rooflines, and the relationship could differ for emulation schemes other than OzIMMU.

IV. Novelty Versus the State of the Art

The classical roofline measures compute in operation count and communication in bytes. Only the latter changes across datatypes, so arithmetic intensity shifts even when neither the computation nor the data has changed. Grützmacher [4] and Anzt et al. [5] resolve this by *reducing* the specificity of communication (counting values instead of bytes), which holds the x-position fixed while the communicated datatype moves the ceiling. However, that approach cannot distinguish emulation configurations: counting logical FP64 operations collapses every Ozaki setting to the same quantity of computational work regardless of precision or range differences. If we were to instead count physical sub-operations, this accounting would inflate work by $O(S^2)$ and ignore the effects of integer-to-float mapping. Matsuoka's Tensor-Memory Equilibrium model [6] accounts for this overhead via a multiplicative factor but, by construction, also places each emulation configuration at the same operational intensity. Information intensity instead *increases* the specificity of computational work and captures the differences among configurations. Finally, because the conditional entropy term penalizes noise, the information roofline can also distinguish a reliable operation from a noise-corrupted one that returns the correct value only 90% of the time (which the classical roofline model treats identically).

References

[1] M. Hawkins and R. Vuduc, "Back to Bits: Extending Shannon's communication performance framework to computing," Aug. 2025. https://doi.org/10.48550/arXiv.2508.05621.

[2] S. Williams, A. Waterman, and D. Patterson, "Roofline: An Insightful Visual Performance Model for Multicore Architectures," *Commun. ACM*, vol. 52, pp. 65–76, Apr. 2009.

[3] H. Ootomo, K. Ozaki, and R. Yokota, "DGEMM on integer matrix multiplication unit," *The International Journal of High Performance Computing Applications*, vol. 38, pp. 297–313, July 2024. https://doi.org/10.1177/10943420241239588.

[4] T. Grützmacher, H. Anzt, and E. S. Quintana-Ortí, "Using Ginkgo's memory accessor for improving the accuracy of memory-bound low precision BLAS," *Software: Practice and Experience*, vol. 53, no. 1, pp. 81–98, 2023. https://doi.org/10.1002/spe.3041.

[5] H. Anzt, G. Flegar, T. Grützmacher, and E. S. Quintana-Ortí, "Toward a modular precision ecosystem for high-performance computing," *The International Journal of High Performance Computing Applications*, vol. 33, pp. 1069–1078, Nov. 2019. https://doi.org/10.1177/1094342019846547.

[6] S. Matsuoka, "FP8 is All You Need (Part 1): Debunking Hardware FP64 as the HPC Holy Grail," May 2026. arXiv:2606.06510 [cs.AR].