Analyzing the Impact of Architectural Design Decisions on Performance Across Generations of NVIDIA GPUs

1st Tavin Mistry
Electrical and Computer Engineering
Queen's University
Kingston, ON, Canada
tavin.mistry@queensu.ca

2nd Tobias Salamon
Electrical and Computer Engineering
Queen's University
Kingston, ON, Canada
21tes9@queensu.ca

3rd Matthew Tindale
Electrical and Computer Engineering
Queen's University
Kingston, ON, Canada
22gbq4@queensu.ca

4th Ian Karlin
Electrical and Computer Engineering
Queen's University
Kingston, ON, Canada
Ian.Karlin@queensu.ca

I. Introduction

The high-performance computing (HPC) industry is moving beyond the era of uniform generational performance gains [1]. As more of the GPU silicon area is allocated to specialized hardware for operations such as matrix multiplication and reduced-precision arithmetic, workload characteristics increasingly determine whether architectural changes deliver meaningful performance and energy efficiency gains [2].

In this work we show that applications capable of exploiting specialized matrix hardware achieve significant gains in performance and energy efficiency, whereas workloads reliant on scalar/vector compute or irregular memory access are constrained by slower scaling elements, such as High Bandwidth Memory bandwidth and Streaming Multiprocessor counts.

We quantify these architectural impacts on performance and energy efficiency across three NVIDIA GPU generations. Workloads spanning scientific computing, AI training, AI inference, and targeted microbenchmarks reveal which hardware advances yield tangible speedups and efficiency gains. Application limiters and hardware counters explain which applications benefit least and why.

II. Test Environment

Experiments compared three generations of NVIDIA datacentre GPUs: a 32 GB V100 PCIe, 40 GB A100 SXM4, and 80 GB H100 SXM5, on systems operated by the Centre for Advanced Computing and Digital Research Alliance of Canada. Each experiment used one GPU to isolate device performance and eliminate inter-GPU communication. We collected application performance, execution time, power, energy consumption, and GPU hardware counters.

The application suite spans scientific computing, AI training, AI inference, and targeted hardware benchmarks. HPL measures dense FP64 performance, HPCG exercises memory-intensive sparse linear algebra, and STREAM measures sustainable HBM bandwidth. Scientific workloads include LAMMPS (molecular dynamics) and MILC (lattice-QCD). AI workloads include mixed-precision ResNet50 training and Megatron-LM Llama1B training, plus PyTorch Llama7B inference. Together they expose different combinations of Tensor Core throughput, FP64 vector performance, memory bandwidth and latency, and synchronization overhead.

III. Analysis

Performance improvements from V100 to H100 range from 2.95x to 8.32x, demonstrating that generational gains depend strongly on workload characteristics. Across these generations, rather than primarily increasing SM counts, NVIDIA devoted more of its transistor budget to increasingly capable Tensor Cores and their supporting data paths, resulting in significantly larger performance gains for matrix heavy workloads. This specialization also shapes energy consumption. As discussed in [3], the energy efficiency of computation is improving faster than data movement. These gains arise from both process scaling (Volta is N12 and Hopper is N5) and Tensor Core specialization: newer process technologies reduce the energy per operation, while larger Tensor Cores increase data reuse and reduce costly data movement [4].

HPL and maximum batch size Llama1B exhibit the largest performance gains because their dense, high-arithmetic-intensity matrix operations effectively utilize Tensor Cores. HPL achieves an 8.32x speedup, approaching the 9.57x increase in maximum FP64 throughput, while its performance per joule improves by 2.30x. Similarly, maximum-MBS Llama1B achieves a 2.49x improvement in energy efficiency and a 7.52x speedup, closely tracking the 7.91x increase in dense FP16 Tensor Core throughput. Llama7B inference

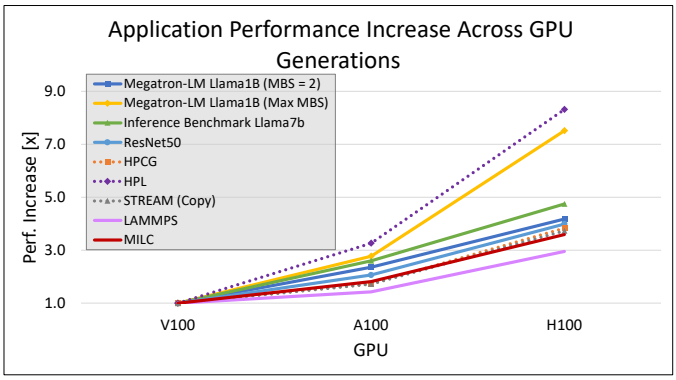


Fig. 1. Plot depicting application performance increase across GPU generations

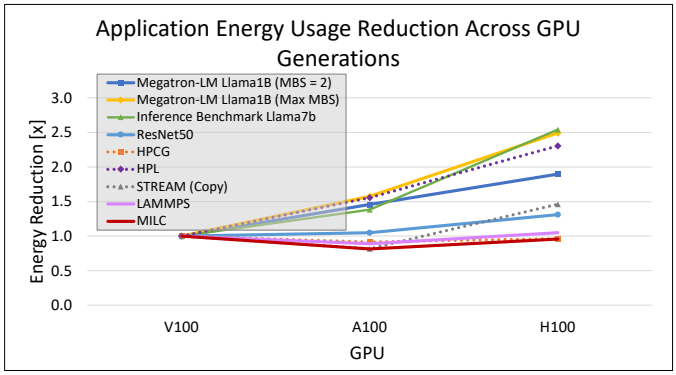


Fig. 2. Plot depicting application energy savings across GPU generations

performance increased by 6.75x with a 2.54x energy efficiency gain because its compute-intensive prefill phase benefits from increased FP16 matrix throughput, but its decoding phase has low arithmetic intensity and is HBM bandwidth limited.

Memory-bound workloads (Stream, HPCG and MILC) performance is within 5% of H100's 3.73x increase in HBM bandwidth. STREAM's regular, contiguous data transfers convert this bandwidth gain into a 1.46x improvement in energy efficiency, whereas HPCG and MILC consume more energy (0.96x from V100 to H100) due to less regular data access patterns. ResNet50 lies slightly above this bandwidth-dominated group, achieving a 3.99x speedup and a 1.31x energy efficiency gain, indicating a mixed bottleneck. Tensor Cores accelerate convolutions, but memory-intensive operations prevent it from approaching FP16 compute scaling. Finally, LAMMPS achieves the lowest speedup and performance per joule improvement at 2.95x and 1.05x because its neighbor-based calculations are FP64 vector limited and cannot use Tensor Cores.

IV. Conclusions and Future Work

From Volta to Hopper, application performance scaled differently based on hardware bottlenecks. HBM-bound applications scaled proportionally with transistor gain; however, energy efficiency flattened due to trailing data-movement efficiency. In contrast, Tensor Core-bound workloads achieved

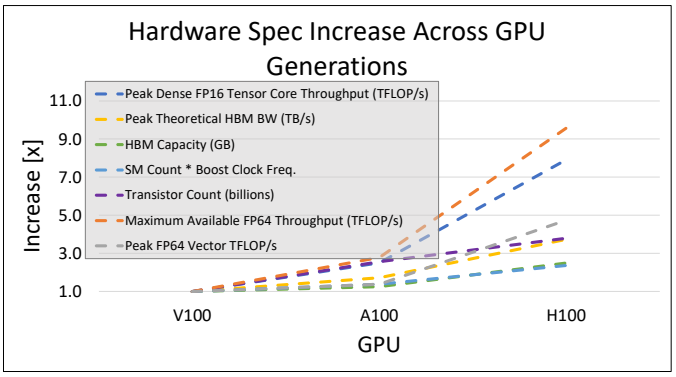


Fig. 3. Plot depicting hardware specification increases across GPU generations

speedups nearly double the transistor count increase alongside improvements in energy efficiency. Partially vector compute-bound applications (e.g., LAMMPS) exhibited the weakest scaling; their performance was constrained by slow SM count growth, with MILC being slowed by HBM bandwidth, ALU and FMA instruction throughput as well.

Future work will expand this evaluation to Pascal and Blackwell architectures and additional applications. Published Blackwell specifications suggest that this specialization trend will continue through expanded low-precision Tensor Core capabilities [5]. We will refine our energy and architectural analyses, by implementing frequency capping to identify optimal perf/watt points and compare performance scaling directly with detailed die-area allocations.

Acknowledgments

The research was undertaken thanks in part to funding from the Connected Minds Program, supported by Canada First Research Excellence Fund, Grant #CFREF-2022-00010. This research was enabled in part by support provided by Queen's University's Centre of Advanced Computing using the Frontenac Cluster, the Digital Research Alliance of Canada National Host Site Machines including, the Fir, Nibi and Rorqual systems.

References

[1] N. C. Thompson and S. Spanuth, "The decline of computers as a general purpose technology," *Commun. ACM*, vol. 64, p. 64–72, Feb. 2021.

[2] J. Domke, E. Vatai, A. Drozd, P. ChenT, Y. Oyama, L. Zhang, S. Salaria, D. Mukunoki, A. Podobas, M. WahibT, and S. Matsuoka, "Matrix engines for high performance computing: A paragon of performance or grasping at straws?," in *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 1056–1065, 2021.

[3] J. Shalf, "The future of computing beyond Moore's Law," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 378, no. 2166, p. 20190061, 2020.

[4] J. Tu, I. Karlin, J. Camier, V. Dobrev, T. Kolev, S. Henneking, and O. Ghattas, "Accelerating high-order finite element simulations at extreme scale with fp64 tensor cores," in *ISC High Performance 2026 Research Paper Proceedings (41st International Conference)*, pp. 1–12, 2026.

[5] NVIDIA Corporation, *NVIDIA Blackwell*, Oct. 2025. Datasheet, document no. 4204213. Accessed: Aug. 5, 2026.