Comparative Performance Analysis of Workload on Enterprise GPUs with Consumer Platforms Accelerated by CUDA Graphs

Leandro L. Retzlaff[1], Calebe C. Pereira[1], Helena P. Veltri[1], Murilo Faganello[1], William T. Klein[1], and Alexandre S. Roque[1]

[1]Universidade Regional Integrada do Alto Uruguai e das Missões – Santo Ângelo – RS
Curso de Ciência da Computação
98802-470 – Santo Ângelo – RS – Brazil

{leandrorfilho,helenapveltri,murilofaganello}@aluno.santoangelo.uri.br

{calebepereira,williamtklein}@aluno.santoangelo.uri.br

roque@santoangelo.uri.br

Abstract. *This work investigates the feasibility of reproducing benchmarks originally run on datacenter GPUs such as the NVIDIA A100 and RTX 8000 using consumer-grade graphics cards, focusing on the NVIDIA GeForce RTX 3050 and GTX 1060 with CUDA Graphs support. Seven NAS Parallel Benchmarks (BT, LU, SP, EP, IS, MG, and CG) are evaluated across problem classes W, A, B, and C. Results show that the RTX 3050 delivers stable performance, typically 6×–12× slower than the A100, while VRAM limitations severely constrain the GTX 1060 for larger-scale problems. Although enterprise GPUs remain essential for massive, memory-bound workloads, modern consumer hardware combined with CUDA Graphs enables economical reproduction of moderate scientific experiments, supporting the democratization of high-performance computing research.*

1. Introduction

The growing demand for artificial intelligence (AI) computing power has propelled the development of high-performance graphics processing units (GPUs) and raised new challenges regarding architecture and energy efficiency [Navaux and da Silva Serpa, 2021, Špet'ko et al., 2021]. Datacenter architectures offer optimized capabilities for mixed-precision operations and intensive workflows—notably through High Bandwidth Memory 2e (HBM2e) in the NVIDIA A100, reaching over 1.5 TB/s—while GeForce line GPUs operate with Graphics Double Data Rate 6 (GDDR6), which creates significant bottlenecks in data-intensive applications.

However, the high cost and restricted availability of server-level GPUs limit access for researchers, students, and institutions in developing countries [Hong and Kim, 2025]. Understanding how hardware limitations can be mitigated by software optimizations is essential for the democratization of advanced computing [Shah, 2024]. This is especially relevant for climate and meteorological applications, where local infrastructure capable of processing large environmental datasets enables early warning systems and adaptation plans without dependence on centralized datacenters.

This study evaluates seven NASA Advanced Supercomputing (NAS) Parallel Benchmarks—Block Tridiagonal (BT), Lower-Upper (LU), Scalar Pentadiagonal (SP), Embarrassingly Parallel (EP), Integer Sort (IS), Multi-Grid (MG), and Conjugate Gradient (CG)—across four problem classes on three consumer GPUs—two NVIDIA GeForce RTX 3050 units and one NVIDIA GeForce GTX 1060—using execution time as the primary metric and taking results reported by [Diakun and Czarnul, 2025] on enterprise GPUs as the reference baseline. Cross-validation between elite and commercial hardware ensures the transparency and reproducibility of scientific results.[1]

2. Background

Contemporary literature highlights that optimizing pre-existing computing infrastructures, even when based on low-cost hardware, can robustly meet growing demands for data processing [Ghimire et al., 2025]. Replicating benchmarks on consumer-market devices validates algorithm portability and establishes an economically viable experimental environment [Diakun and Czarnul, 2025, Cortes et al., 2025]. This movement addresses the “compute gap", where concentration of processing power in large corporations threatens the scientific sovereignty of smaller institutions [Gowda et al., 2023].

Nonetheless, consumer GPUs impose architectural challenges: limited video random access memory (VRAM) and the absence of high-speed interconnects directly impact scalability [Zangana et al., 2024]. Research has therefore converged on techniques such as task parallelization, weight quantization, and dynamic memory management [Wu et al., 2024, Mencattini et al., 2025]. Validating commercial hardware for scientific use also contributes to sustainability by reducing dependence on large datacenters with high water and carbon footprints.

3. Methodology

The methodology of this study is based on the experimental framework proposed by [Diakun and Czarnul, 2025], who investigated the performance impact of Compute Unified Device Architecture (CUDA) Graphs on selected parallel applications. Although their results indicate that CUDA Graphs may not improve performance across all NAS Parallel Benchmarks, this limitation was also considered relevant to the present analysis. Since the objective of this work is to reproduce and evaluate this methodology on more accessible consumer-grade GPUs, benchmarks exhibiting both favorable and unfavorable behaviors were included. This allows for verifying whether the performance trends reported in datacenter-oriented environments are also observed on consumer platforms, as well as quantifying not only potential gains but also overheads and architectural limitations. The experimental workflow, illustrated in Figure 1, was designed to ensure a controlled environment and reproducible results across different GPU architectures.

3.1. Algorithm Selection and Implementation

To assess the performance of various GPUs in various scenarios, the NAS Parallel Benchmarks (NPB) suite was selected because it serves as the benchmark framework adopted by [Diakun and Czarnul, 2025], whose results on enterprise GPUs (NVIDIA A100 and RTX 8000) constitute the reference baseline for the comparisons performed in this study.

[1]Source code and data: `https://doi.org/10.5281/zenodo.20300643`

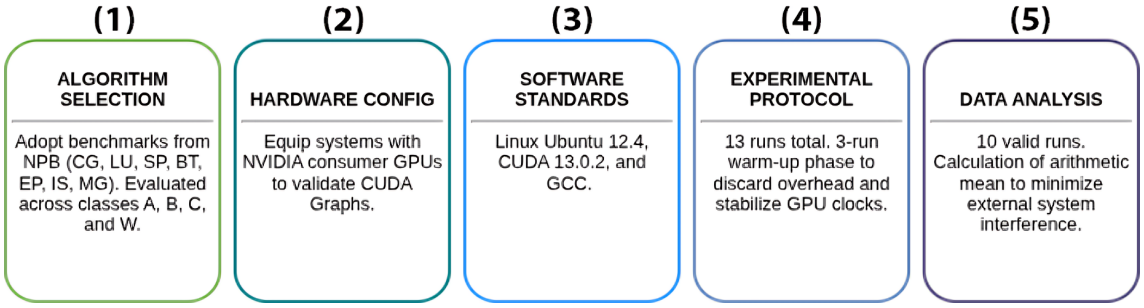

Figure 1 – Methodology workflow stages.

The selection covers a wide range of computational patterns. Iterative solvers and computational fluid dynamics (CFD)-oriented kernels—CG, LU, SP, and BT—evaluate mathematically intensive operations and synchronization dependencies typical of CFD. The EP and IS benchmarks test the system's ability to sustain massive parallelism without inter-process communication and to stress memory bandwidth, respectively. The MG algorithm addresses partial differential equations across multiple resolutions, combining complex computation with irregular communication patterns.

In this study, NPB classes W, A, B, and C were selected to evaluate each GPU under progressively more demanding workloads, from initial validation (W) through intermediate sizes (A and B) to the most demanding configuration (C) that exposes scalability and memory limits. Classes D and E were excluded because their memory requirements exceed the VRAM capacity of the consumer-grade GPUs evaluated.

3.2. Hardware and Software Configuration

Standardizing the software environment is critical when the primary objective is to evaluate hardware performance. This ensures that the results remain consistent, comparable, and exclusively attributable to the hardware characteristics under analysis.

The experiments were conducted on systems equipped with NVIDIA consumer-grade GPUs. The selected models, the NVIDIA GeForce RTX 3050 and the NVIDIA GeForce GTX 1060, represent widely available hardware in the consumer market. Replicating the study on these GPUs allows for the validation of CUDA Graphs' applicability in more accessible environments compared to the enterprise-grade baseline (NVIDIA A100 and RTX 8000).

The virtual environment was standardized to guarantee reproducibility. All runs used Linux Ubuntu 22.04 Long Term Support (LTS), CUDA Toolkit 13.0.2, and the GNU Compiler Collection (GCC). CUDA kernels from the NPB suite were compiled to utilize CUDA Graphs, including proper initialization and capture of computation graphs.

3.3. Experimental Protocol

To ensure statistical reliability and adhering to a **Simple Design** strategy, where factors are isolated to analyze their individual impact on performance, an execution protocol was established for each combination of algorithm, problem class, and hardware configuration. To mitigate the stochastic nature of execution times on modern operating systems and GPU hardware, a total of 13 consecutive runs were performed for each unique configuration. The first three executions were discarded as a warm-up phase to mitigate initialization overheads, allow GPU clock frequencies to stabilize, and ensure instruction

caches are populated. The remaining ten executions were recorded as valid data points; the primary metric collected was wall-clock time, measured with high precision using the benchmark's internal timers. The final performance metric for each configuration is reported as the arithmetic mean of the ten valid runs, minimizing the influence of outliers caused by external system interference.

4. Experiments and Results

Table 1 summarizes the execution times (in seconds) for all benchmarks and problem classes on each GPU, compared to the enterprise reference. Figures 2 and 3 illustrate representative compute-intensive and memory-bound behavior, respectively.

Table 1. Execution times (seconds) for NPB benchmarks by GPU and problem class.

	GPU	W	A	B	C
BT	3050-A	0.28	3.70	13.7	54.1
	3050-I	0.27	3.93	17.2	58.6
	1060	0.42	5.70	18.9	—
	A100	0.10	0.64	1.89	7.60
	8000	0.15	1.02	4.10	17.3
EP	3050-A	0.15	0.48	1.93	7.71
	3050-I	0.15	0.49	1.94	7.77
	1060	0.16	0.41	1.63	6.19
	A100	0.05	0.05	0.07	0.11
	8000	0.14	0.16	0.48	1.93
LU	3050-A	0.79	3.34	12.7	44.1
	3050-I	0.79	3.30	12.6	43.5
	1060	0.79	3.35	13.3	52.1
	A100	0.33	0.66	1.37	3.34
	8000	0.52	1.63	4.14	17.4
SP	3050-A	0.45	1.56	7.44	28.0
	3050-I	0.48	1.60	7.67	28.8
	1060	0.41	1.74	7.57	31.1
	A100	0.09	0.32	0.76	2.51
	8000	0.19	0.52	2.15	10.7
IS	3050-A	0.00	0.02	0.21	1.44
	3050-I	0.00	0.02	0.20	1.29
	1060	0.00	0.02	0.48	2.51
	A100	0.00	0.00	0.00	0.03
	8000	0.00	0.00	0.04	0.69
MG	3050-A	0.01	0.08	0.38	—
	3050-I	0.01	0.09	0.43	—
	1060	0.01	0.10	0.47	—
	A100	0.00	0.01	0.04	—
	8000	0.01	0.03	0.10	—
CG	3050-A	0.04	0.07	2.30	7.26
	3050-I	0.04	0.08	2.81	7.88
	1060	0.05	0.10	3.29	13.1
	A100	0.02	0.03	0.44	1.03
	8000	0.02	0.04	0.78	1.84

3050-A/I: RTX 3050 (AMD/Intel). —: not completed (insufficient VRAM).

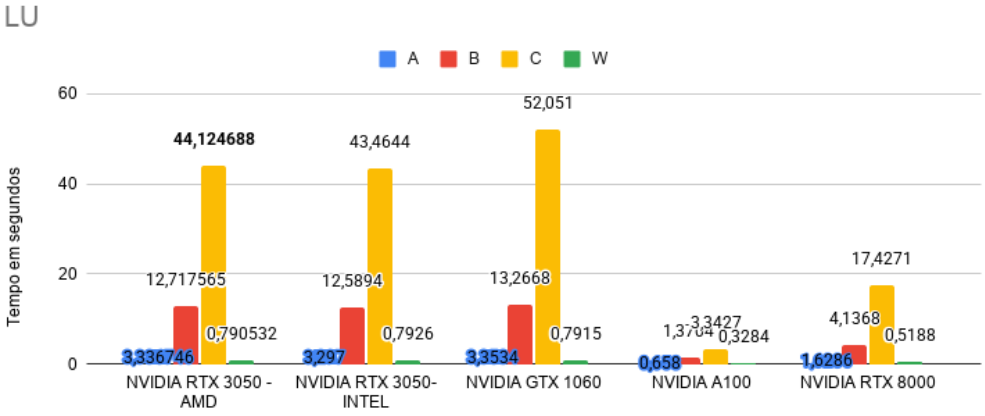

Figure 2 – LU benchmark execution times by GPU and problem class (compute-intensive example).

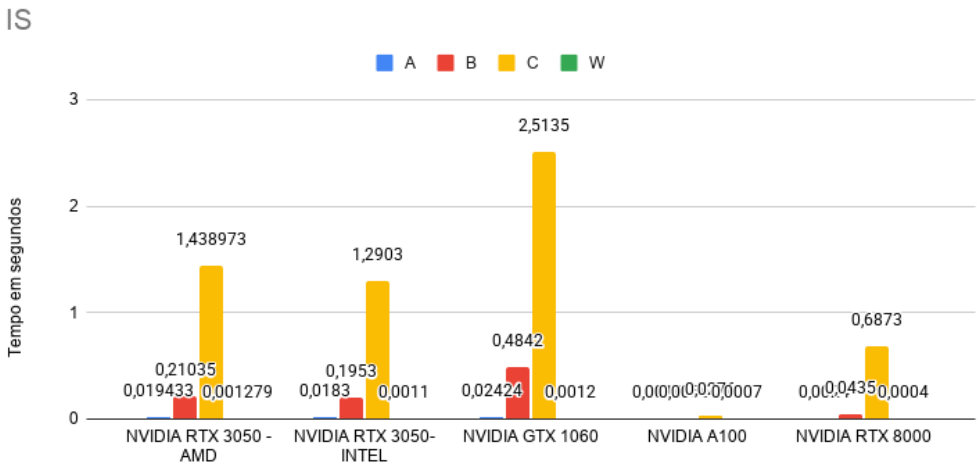

Figure 3 – IS benchmark execution times by GPU and problem class (memory-bound example).

4.1. Execution Analysis

For compute-intensive and memory-bound benchmarks (BT, LU, SP, IS, EP, and CG), the RTX 3050 delivers stable, predictable performance across CFD-oriented workloads. In **LU** (Class C), it reached 40 s versus 3.34 s on the A100 (13.2× slowdown), yet outperformed the GTX 1060 (52.05 s), highlighting Ampere-over-Turing architectural gains. In **BT**, the RTX 3050 reached 50 s while the A100 completed the same task in 7 s. In **SP**, execution times were 27.99 s (RTX 3050) versus 2.51 s (A100), an 11× ratio. Neither the RTX 3050 nor the GTX 1060 could complete the MG Class C benchmark due to memory constraints. By contrast, the **IS** benchmark reveals the sharpest disparity: the RTX 3050 required 1.44 s versus only 0.0276 s on the A100—a gap of approximately 52×—consistent with the enormous bandwidth advantage of HBM2e over GDDR6. The RTX 8000 (0.69 s) also remained far ahead of the consumer cards, confirming that memory-throughput-intensive applications continue to depend on enterprise hardware. The **EP** benchmark shows that consumer GPUs sustain embarrassingly parallel workloads without communication overhead, with the RTX 3050 maintaining stable execution across classes W through C relative to the enterprise baseline. The **CG** results follow the same trend observed for BT and LU: the RTX 3050 remains between 6× and 13× slower than the A100 while consistently outperforming the GTX 1060, confirming that iterative sparse solvers remain viable on accessible hardware when CUDA Graphs reduces kernel launch overhead.

Regarding legacy hardware limitations, the GTX 1060 (3 GB VRAM) failed to complete BT and MG in Class C, as CUDA Graphs requires prior allocation of buffers and descriptors that exceed its memory capacity. Even in successful runs, it was consistently 2×–3× slower than the RTX 3050 (e.g., SP: 31.11 s; LU: 52.05 s), demonstrating that Pascal-generation hardware no longer keeps pace with workloads optimized for modern CUDA Graphs flows.

5. Discussion

The results confirm that modern consumer GPUs can execute moderate scientific applications effectively when combined with optimizations such as CUDA Graphs. For compute-limited CFD algorithms (BT, LU, SP), the RTX 3050 achieves slowdown factors between

6× and 13× relative to the A100, situating it in the same order of magnitude while maintaining execution stability across all tested classes. This validates the use of accessible hardware for development cycles, debugging, and medium-scale experimentation.

For memory-bound workloads (IS) or problems requiring high VRAM (MG Class C), datacenter hardware remains irreplaceable. The GTX 1060 reinforces generational limits: Pascal GPUs no longer support modern CUDA Graphs workloads reliably, while the Ampere-based RTX 3050 partially compensates for the performance gap.

6. Conclusion

This study evaluated the feasibility of reproducing NVIDIA A100 performance benchmarks on consumer hardware using the NAS Parallel Benchmarks. The NVIDIA RTX 3050 executed complex Class C workloads with stable performance, presenting slowdown factors between 11× and 13× for computation-limited algorithms and consistently surpassing the GTX 1060 across all benchmarks. CUDA Graphs reduces overheads and maximizes GPU occupancy on constrained hardware, supporting the use of consumer cards for development and medium-scale testing while reserving enterprise platforms for final large-scale runs. These findings validate accessible GPUs as economical tools for moderate high-performance computing research.

References

D. Cortes, C. Juiz, and B. Bermejo. Estudio de la eficiencia en la escalabilidad de gpus para el entrenamiento de inteligencia artificial, 2025. arXiv:2509.03263.

O. Diakun and P. Czarnul. Investigation of cuda graphs performance for selected parallel applications. In *International Conference on Computational Science*, page 130. Springer, 2025. doi: 10.1007/978-3-031-97635-3_16.

S. Ghimire et al. Cost-effective deep learning infrastructure with nvidia gpu. *Kathmandu University Journal of Science, Engineering and Technology*, 2025. doi: 10.48550/arXiv.2503.11246. arXiv:2503.11246.

S. N. Gowda et al. Watt for what: Rethinking deep learning's energy-performance relationship, 2023.

Y. Hong and D. Kim. Performance and efficiency gains of npu-based servers over gpus for ai model inference. *Systems*, 13(9):797, 2025. doi: 10.3390/systems13090797.

A. Mencattini et al. Evolutionary model merging on consumer gpus, 2025.

P. O. A. Navaux and M. da Silva Serpa. Desafios do processamento de alto desempenho. In *Seminário Integrado de Software e Hardware (SEMISH)*, pages 39–49. SBC, 2021. doi: 10.5753/semish.2021.15805.

R. K. Shah. Global gpu network (ggn): Performance benchmarking and comparative analysis against traditional gpu computing models, 2024.

M. Špet'ko, O. Vysockỳ, B. Jansík, and L. Říha. Dgx-a100 face to face dgx-2—performance, power and thermal behavior evaluation. *Energies*, 14(2):376, 2021. doi: 10.3390/en14020376.

R.-F. Wu, Y. Wang, and D. Kutscher. Affordable hpc: Leveraging small clusters for big data and graph computing, 2024.

H. M. Zangana, F. M. Mustafa, and S. Li. Large language models in cybersecurity, 2024.