Did You Win the GPU Cloud Lottery? Benchmarking from TFLOPS to Tokens/$

Platon Slynko
Silicon Data
New York, New York, USA
platon@silicondata.com

Jay Lu
Silicon Data
jlu@silicondata.com

Jason Cornick
Silicon Data
jcornick@silicondata.com

Laksh Sharma
Silicon Data
laksh@silicondata.com

Shou-Kai Cheng
Silicon Data
kcheng@silicondata.com

Benjamin Cornick
Silicon Data
ben@silicondata.com

Arthur O. Dias dos Santos
Silicon Data
arthur@silicondata.com

Caique Sobral
Silicon Data
caique@silicondata.com

Carmen Li
Silicon Data
cli@silicondata.com

Daoxuan Xu
William & Mary
Williamsburg, Virginia, USA
dxu05@wm.edu

Xinxin Mei
Silicon Data
Jefferson Lab
Newport News, Virginia, USA
xinxin@silicondata.com

Yifan Sun
William & Mary
Williamsburg, Virginia, USA
ysun25@wm.edu

Abstract

Cloud GPUs are typically considered to deliver the same performance for the same GPU model. However, such assumptions may not hold, as cloud providers may have different configurations and GPUs may exhibit slight manufacturing differences (silicon lottery). In this work, we present a large-scale measurement study of GPU performance variability across 11 cloud providers, covering over 3,500 physical GPUs and 6,800 benchmark runs. Our hierarchical analysis shows that substantial variability arises at both the intra-provider and inter-provider levels. Silicon lottery explains the majority of observed performance variation. Additionally, cloud providers introduce persistent and systematic second-order differences by shaping the performance distribution.

Keywords

GPU, Performance Benchmarking, Cloud Computing

1 Introduction

The rapid rise of large language models (LLMs) and agentic AI workflows has driven an unprecedented expansion in GPU cluster scale [1–5]. State-of-the-art training and inference deployments increasingly rely on hundreds of thousands of GPUs [6]. At the same time, the infrastructure and capital requirements of modern GPU platforms have grown substantially, driven by the demands of power delivery, advanced cooling, high-bandwidth networking, and specialized operational expertise. For example, a single NVL72 rack represents a multi-million U.S. dollar investment, with rack-level power draw exceeding 100 kWatts [7]. As a result, local GPU cluster deployment is no longer the default option for many users.

In parallel, cloud-based GPU availability has expanded rapidly. Beyond traditional hyperscalers such as AWS and Google Cloud, a growing ecosystem of GPU and AI-focused neocloud platforms now offers flexible, on-demand access to modern AI accelerators, with deployment and pricing models tailored to both training and inference workloads. These providers enable users to rent GPUs by the minute or in GPU-count blocks, substantially lowering the barrier to entry for large-scale AI experimentation. Consequently, an increasing fraction of AI workloads, particularly inference and exploratory training, are now executed on geographically distributed cloud GPU infrastructure rather than on local data centers.

Prior work has shown that GPU performance is not perfectly uniform even in tightly controlled HPC environments. Sinha *et al.* [8] demonstrated that GPUs of the same model deployed within the same cluster can exhibit substantial performance variability that cannot be explained by data center–level power and cooling conditions alone. Similar

GPGPU '26, Pittsburgh, USA
2026. ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

manufacturing-induced variability has also been observed in CPU-based HPC systems, where heterogeneity was largely attributed to power throttling and clock-boosting mechanisms. The variability arising from intrinsic differences in silicon characteristics is commonly referred to as the *silicon lottery* and has been extensively studied in the context of performance-aware and power-aware task scheduling for large-scale computing systems [9–12].

Compared to traditional HPC clusters, cloud platforms exhibit significantly greater heterogeneity, including variation in GPU submodules, geographic deployment, power and cooling policies, virtualization layers, and software configurations. As a result, GPU cloud environments are more likely to expose and amplify performance variability.

At the same time, recent studies of cloud workloads and LLM execution have shown that peak compute capability, typically expressed in floating-point operations per second (FLOPS), is an insufficient predictor of realized performance. Instead, memory behavior, interconnect performance, and system-level constraints frequently decide end-to-end execution efficiency [13–16]. This observation has motivated the development of alternative performance metrics tailored to LLM workloads. For example, Javier *et al.* showed that tokens per second does not reliably reflect end-to-end execution speed [17], while Saad *et al.* proposed *Intelligence per Watt* as an efficiency metric that explicitly accounts for growing power constraints in modern AI systems [18].

Motivated by these trends, we design a measurement-driven toolset, *SiliconMark*, to systematically characterize GPU performance variability in contemporary commercial cloud environments. Our framework benchmarks randomly select GPU instances across multiple cloud providers, capturing device, node, and application-level performance metrics alongside runtime power and thermal behavior, while also tracking GPU rental and inference pricing over time. This approach enables us to quantify the magnitude and structure of GPU performance variability in the cloud and, consequently, to expose the performance–cost trade-offs faced by users deploying AI workloads.

Our paper makes three main contributions.

- First, we quantify *silicon lottery* effects at scale, demonstrating that device-level heterogeneity is a dominant source of GPU performance variation.
- Second, we show that cloud providers can amplify intrinsic performance differences driven by provider-specific characteristics.
- Third, we demonstrate that GPU performance variability, in conjunction with rental price dynamics, leads to substantial differences in inference cost efficiency, challenging the assumption that newer or higher-end GPUs necessarily deliver better tokens/$.

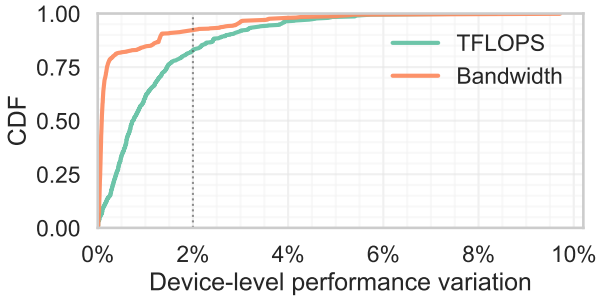


Figure 1: Cumulative distribution function (CDF) of the normalized performance range (746 GPU devices).

Together, our results reveal an implicit *GPU cloud lottery*: renting a GPU in the cloud selects a performance–cost distribution rather than a uniform computational resource, undermining the common assumption of GPU equivalence in cloud deployments.

2 Methodology

We run our benchmarking suite on 11 GPU cloud platforms, spanning both hyperscalers and neocloud providers.

We anonymize all cloud providers (denoted as *Cld*-x) throughout this work for several reasons. First, most cloud providers' Terms of Service explicitly prohibit public benchmarking or comparative performance claims without prior written approval. Second, as our author team includes industry practitioners, publishing named performance comparisons could expose authors and affiliated institutions to legal liability, particularly given that observed performance differences may partially stem from factors outside provider control, such as NVIDIA's silicon binning practices during manufacturing. Finally, and most importantly, our scientific contribution concerns the existence, structure, and magnitude of GPU performance variability as a systemic phenomenon in cloud computing, rather than producing a consumer ranking of specific vendors.

2.1 GPU Micro-Benchmarking

Our single-GPU micro-benchmark suite, *QuickMark*, reports a set of metrics designed to characterize device-level GPU performance. In this study, we focus on two:

(1) FP16 matrix multiplication (MatrixMul) throughput, reported in TFLOPS, using a 16,384 × 16,384 workload;
(2) GPU device-to-device (D2D) memory copy bandwidth, reported in GB/s, measured via 1 GB data transfers.

While these metrics do not capture all aspects of GPU performance, they are lightweight and representative indicators of dominant compute and on-device memory behavior in modern AI workloads. FP16 MatrixMul throughput captures

Table 1: Definition of Hierarchical GPU Performance Variation

Level	Grouping	Baseline	Insight
Execution-level	Same physical GPU device	Per-device mean performance	Benchmarking effectiveness and intrinsic device behavior
Intra-provider	Same GPU model within a single provider	Per-model mean within provider	Device-level variability under a common provider environment
Inter-provider	Same GPU model across different providers	Per-model mean across providers	Systematic performance differences attributable to provider-level system factors

the compute capability of the GPU's matrix or tensor accelerators, while D2D memcpy is a proxy for GPU on-device memory subsystem performance, which directly impacts collective communication and tensor data movement in modern AI workloads. Host-side factors, including CPU performance, PCIe interconnects, and host memory, are intentionally deemphasized, as they exert indirect effects.

Our objective is not to achieve optimal performance, but to enable fast, fair, and consistent comparisons across cloud platforms. We evaluate these metrics under each platform's default software stack, including the native operating system, CUDA toolkit, and GPU drivers. All benchmarks are implemented in PyTorch and executed via automated agents.

To compare GPU performance across cloud providers, we analyze performance variation at three hierarchical levels: execution-level, intra-provider, and inter-provider. Execution-level variation captures performance differences across repeated executions on the same physical GPU device. Together, these levels isolate variability ranging from intrinsic device behavior to provider-wide system effects. At each level, we apply a normalization baseline appropriate to the scope of comparison. Table 1 summarizes the definition, normalization baseline, and interpretation associated with each level of variation.

On most cloud platforms, the rent-on-demand pricing model does not guarantee access to the same physical GPU device across scheduled runs. To mitigate this limitation, our benchmark suite records the UUID of each NVIDIA GPU, enabling performance tracking across repeated executions. In total, our dataset contains measurements from 3,534 distinct GPU instances and over 6,800 benchmark samples collected over a four-month period.

For the 736 GPU devices with at least three measurement samples, we normalize FP16 TFLOPS and D2D bandwidth by the corresponding per-device mean. We quantify execution-level performance variation using *normalized range*, defined as $(max - min)/mean$, and plot its cumulative distribution function (CDF) in Figure 1.

Building on this foundation, we quantify the *silicon lottery* effect across individual GPU devices and examine whether cloud providers amplify this variability.

2.2 LLM Inference Benchmarking

We next examine how microbenchmark-level performance variability propagates to end-to-end LLM inference throughput. We evaluate a representative LLM hosted on Hugging Face [19], NVIDIA's Llama-3.1-Nemotron-Nano-4B-v1.1 [20], a compact model well suited for single-GPU inference. The model is served using the vLLM inference engine [21] and is evaluated as released, without tuning or tensor parallelism. All experiments use the bfloat16 data type. The extracted model weights occupy approximately 8.4 GiB of GPU memory, with an additional 58 GiB of GPU memory allocated for the KV cache.

We employ two synthetic workloads: (1) *Chat*, with both input sequence length (ISL) and output sequence length (OSL) set to 128; and (2) *Summarization*, with ISL/OSL set to 4096/512. For each workload, we evaluate four concurrency levels: 1, 25, 50, and 100, to capture performance from lightly loaded to highly concurrent scenarios. Here, concurrency denotes the number of simultaneous inference requests issued to the vLLM server, rather than batch size or degrees of model parallelism.

3 Results

3.1 GPU Performance Distribution

Table 2 reports the achieved FP16 throughput and memory bandwidth for 11 GPU model groups. For each model, we present the observed minimum and maximum values, along with the corresponding per-model normalized performance range. Only GPU models with at least three measurement samples are included; each group contains between 47 and 2,096 measurements. The table also lists the CUDA compute capability (CC) associated with each GPU model.

It is well established that, within a given GPU family (e.g., A100), distinct product variants (e.g., SXM versus PCIe) can exhibit substantially different performance characteristics. Consistent with this expectation, our measurements show

Table 2: Measured FP16 TFLOPS and Memory Bandwidth by GPU Model

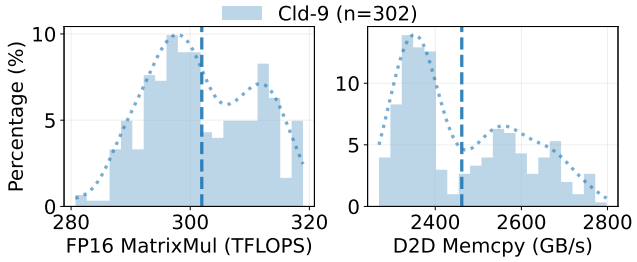
			FP16 MatrixMul (TFLOPS)			D2D Memcpy (GB/s)		
Model	CUDA CC	# GPUs	Min	Max	Range (%)	Min	Max	Range (%)
NVIDIA Tesla T4	7.5	939	15.72	19.36	20.68	232.36	244.96	5.19
NVIDIA A100 PCIe 40GB	8.0	47	174.56	207.98	17.17	1,383.05	1,385.62	0.19
NVIDIA A100 PCIe 80GB	8.0	254	213.80	235.85	9.70	1,521.01	1,727.28	12.64
NVIDIA A100 SXM4 40GB	8.0	94	250.21	269.28	7.32	1,379.17	1,381.60	0.18
NVIDIA A100 SXM4 80GB	8.0	2096	242.06	291.21	19.12	1,458.62	1,758.82	17.19
NVIDIA A10G	8.6	940	67.68	69.35	2.42	481.85	488.61	1.39
NVIDIA L4	8.9	1170	48.74	56.91	15.43	231.49	238.87	3.10
NVIDIA H100 NVL	9.0	302	280.73	318.93	12.65	2,269.38	2,797.87	21.47
NVIDIA H100 PCIe	9.0	259	296.24	424.39	34.49	1,655.08	1,837.31	10.08
NVIDIA H100 SXM	9.0	452	648.52	719.80	10.40	2,543.86	3,026.97	16.67
NVIDIA H200 SXM	9.0	253	634.66	681.70	7.12	2,939.47	4,212.67	37.82

that the maximum FP16 throughput reaches 720 TFLOPS on H100 SXM (HBM3), but only 319 TFLOPS on H100 NVL. Beyond this model-level variation, individual GPUs of the same model also exhibit significant performance differences, with the observed normalized performance range reaching up to 37.8%. Notably, the largest performance variation is observed not in older or lower-end GPUs but in flagship data center GPUs.

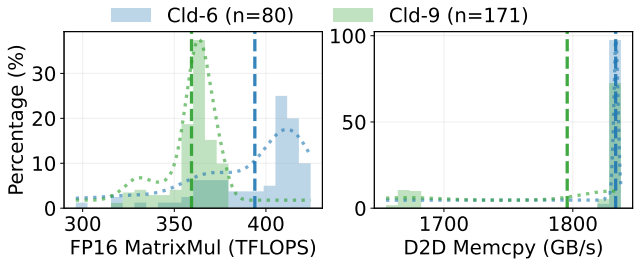
Figure 2 and Figure 3 illustrate the performance distributions of H100 and H200 family GPUs across cloud providers. Corresponding distribution for the A100 family GPUs is provided in Appendix Figures 8. Each histogram uses 20 bins, with the y-axis reporting the percentage of samples relative to the total number of measurements for each provider. The dashed curve denotes the kernel density estimate (KDE), the vertical line marks the mean performance, and the number of samples is indicated as n in the legend.

For H100 SXM GPUs, we observe a clear inter-provider difference: Cld-5 consistently exhibits lower FP16 TFLOPS than other providers, while its D2D bandwidth remains comparable. A similar pattern is observed for H100 PCIe GPUs, where Cld-9 has lower FP16 throughput than Cld-6.

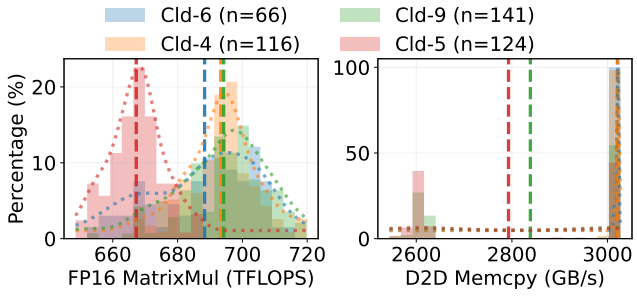
Within each provider, FP16 throughput distributions are tightly clustered around the mean, with only a small number of outliers, indicating relatively consistent compute performance. In contrast, D2D bandwidth exhibits a markedly different pattern. Measurements frequently cluster around two distinct values, indicating a bimodal distribution. This behavior is consistently observed across the A100, H100, and H200 GPU families. While identifying the root causes of this behavior is an important direction for future work, in this paper, we focus on quantifying the extent and structure of such variability rather than attributing it to underlying mechanisms.



(a) NVIDIA H100 NVL



(b) NVIDIA H100 PCIe



(c) NVIDIA H100 SXM

Figure 2: GPU core and memory performance distributions of NVIDIA H100 GPUs across cloud providers.

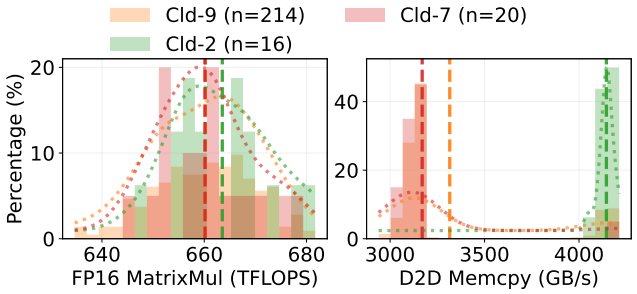


Figure 3: GPU core and memory performance distribution of NVIDIA H200 SXM across cloud providers.

3.2 GPU Performance Variation Hierarchy

We next narrow our analysis to GPU models offered by multiple cloud providers, including five models: A100 80GB PCIe, A100 80GB SXM4, H100 PCIe, H100 SXM, and H200 SXM, spanning eight providers. Figure 4 presents box plots of their normalized performance ranges, with normalization performed relative to the cross-provider mean. The box distributions capture both intra-provider and inter-provider variability, and reveal that the *silicon lottery* effects transit into persistent provider-level variability.

For FP16 throughput, the H100 PCIe model exhibits the largest performance variation among the evaluated GPUs. In particular, Cld-9 outperforms Cld-6 by approximately 9% on average, in addition to the substantial intra-provider TFLOPS variability at both providers. Another notable case is the A100 SXM4 deployed by Cld-9. While the remaining four providers show intra-provider variation within 10%, the intra-provider variation of Cld-9 reaches 19%, indicating pronounced heterogeneity within a single provider.

Memory bandwidth exhibits a larger variation. For the H200 SXM model, the average D2D bandwidth reaches 4,145 GB/s on Cld-2, compared to 3,170 GB/s on Cld-7, corresponding to a 28% difference. The observed bimodal bandwidth distributions contribute significantly to intra-provider variability. For example, on Cld-7 and Cld-9, H200 D2D performance consistently groups around either 3 TB/s or 4 TB/s.

Table 3 summarizes the performance ranges across the three hierarchical levels. For both FP16 MatrixMul throughput and D2D bandwidth, execution-level variation remains below 9%, whereas provider-level normalized ranges reach up to 38%. These results provide strong empirical evidence of *silicon lottery* effects in cloud GPU deployments. In Table 3, inter-provider normalized ranges are not necessarily larger than intra-provider ones, due to normalization against different baselines. When the intra-provider range exceeds the inter-provider range for a given GPU model, performance variability is dominated by heterogeneity among devices within a small subset of providers. In contrast, larger inter-provider ranges indicate that provider-specific characteristics and device populations amplify *silicon lottery* effects.

3.3 Sources of GPU Performance Variation

To identify the sources of performance variation observed in Figure 4, we apply linear regression to decompose the observed variance into contributing factors. Specifically, we model FP16 performance as a function of the cloud provider (denoted by *cid*), the individual GPU device (denoted by *gid*), and the test month, which serves as a proxy for temporal system effects (such as OS upgrades) within a provider.

We fit separate regression models for each GPU product, covering three H100 models and four A100 models. Across all variants, the month feature consistently contributes less than 1% of the total variance, indicating negligible temporal effects over the measurement period. In contrast, *gid* accounts for 32–73% of the variance, depending on the GPU model, confirming that *silicon lottery* effects dominate the observed performance variability. While *gid* presents more variance than *cid*, *cid* captures a persistent, second-order system effect. This effect reflects long-lived differences in device population, configuration policies, and deployment practices rather than short-term system noise. Overall, the regression models exhibit high explanatory power, with six of the seven models achieving R^2 values between 0.97 and 0.99. This indicates that our three factors capture nearly all observed performance variation.

Although device identity explains the majority of the observed variance, they are not independent of cloud providers in practice. Each provider maintains a largely fixed inventory of physical GPUs, making the performance distribution a stable property of the provider over time. To examine the *silicon lottery* effects at the inter-provider level, we refit the regression models using only *cid* as the predictor. The resulting models reveal clear and persistent performance differences among providers for all GPU variants. While the relative ranking of providers varies by GPU model, stronger performance on one model does not guarantee the same advantage on other models. These differences remain stable over time, suggesting that inter-provider variability arises from long-lived infrastructure characteristics rather than transient system effects, and should therefore be considered when selecting a cloud provider.

3.4 AI Inference Performance

For the Llama-3.1-4B inference throughput, we collect 61 samples on H100 SXM GPUs, spanning 18 individual devices on Cld-4 and Cld-6, and 23 samples on H200 SXM GPUs, collected from 9 devices on Cld-4. For this subgroup of GPU devices, both provider-level heterogeneity and run-to-run

Table 3: Maximum Normalized Performance Ranges across Hierarchy Levels

	FP16 MatrixMul, Max Range (%)			D2D Memcpy, Max Range (%)		
GPU Model	Execution-Level	Intra-Provider	Inter-Provider	Execution-Level	Intra-Provider	Inter-Provider
A100 PCIe	3.86	9.73	9.70	1.85	12.70	12.64
A100 SXM4	2.56	18.70	19.12	8.65	17.51	17.19
H100 PCIe	8.37	32.53	34.49	1.18	10.07	10.08
H100 HBM3	5.73	9.58	10.40	2.56	16.95	16.67
H200 SXM	2.78	6.97	7.12	5.34	38.14	37.82

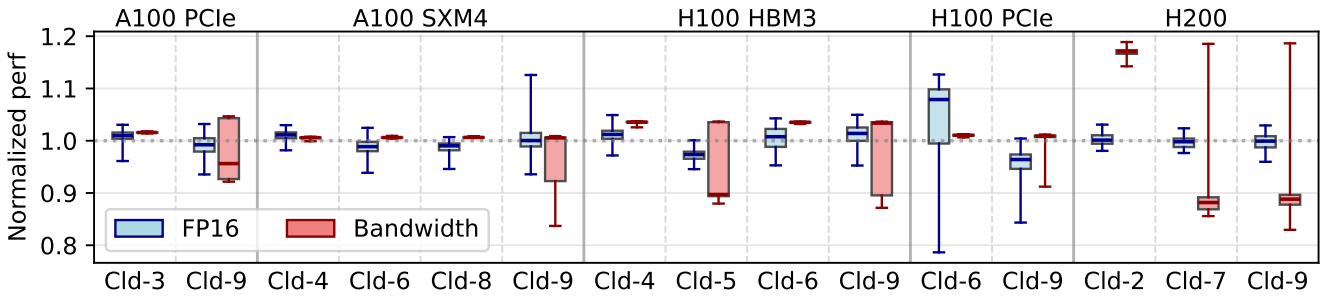


Figure 4: Performance variation of five target GPU models, normalized to the cross-provider mean. Each box represents the distribution of the intra-provider performance.

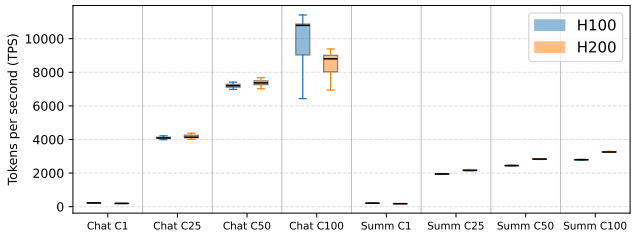


Figure 5: Achieved Llama-3.1-4B inference throughput in tokens per second (TPS), aggregated by GPU model.

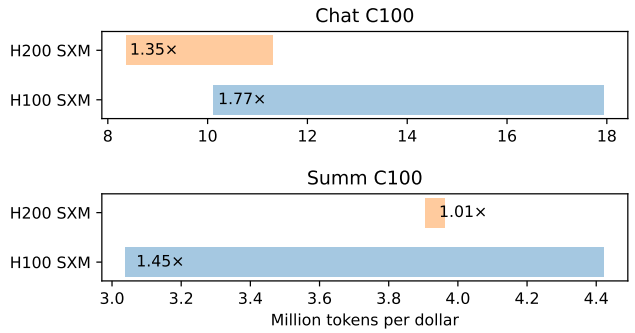


Figure 6: Variation in tokens-per-dollar in a single day for two Llama-3.1-4B workloads. Annotations denote *max*/*min* ratios within each GPU model.

variability are tightly bound. The H100 inter-provider normalized ranges of FP16 TFLOPS and D2D bandwidth are 8.3% and 0.6%, while the corresponding ranges for H200 are 6.6% and 3.3%. Execution-level performance variation remains low, with the maximum observed normalized range limited to 5.4% across six runs.

Figure 5 presents the distribution of Llama-3.1-4B inference throughput in tokens per second (TPS) across eight workload configurations. Although higher-end GPUs are commonly assumed to deliver superior TPS, our results show that this does not hold in practice. H100 outperforms H200 in three configurations, with a peak advantage of 1.19× under the Chat C100 workload. In contrast, H200 achieves higher TPS under heavier workloads and higher concurrency, benefiting from its memory capacity.

Across the two cloud platforms, Cld-6 delivers stable TPS for all workloads, with a maximum intra-provider normalized range of 6.3%. In general, TPS on Cld-4 is also stable. However, under the Chat C100 workload, the normalized range increases to 29.0% for H200 and 53.0% for H100 GPUs.

Under a fixed dollar-per-GPU-hour pricing model, performance differences would map directly to differences in tokens/$. In practice, cloud GPU pricing varies substantially across providers and over time, making tokens/$ a joint function of both throughput and cost. Using the Llama-3.1-4B

Chat C100 and Summarization C100 workloads, and considering a single-day pricing snapshot for the two corresponding cloud providers, we observe an intra-provider max-to-min tokens/$ ratio of up to 1.77, as illustrated in Figure 6. Notably, the larger ratio observed for H100 instances is driven primarily by greater pricing disparity between providers under comparable performance dispersion, rather than by increased performance variance alone.

Moreover, Figure 6 reveals a non-negligible overlap in tokens/$ between H100 and H200 instances. This overlap indicates that, in the presence of *silicon lottery* effects, different GPU models may achieve comparable cost efficiency for the same inference workloads. In particular, newer GPU generations do not necessarily yield higher tokens/$, despite offering higher peak TFLOPS or memory bandwidth.

4 Conclusion

This study quantifies the silicon lottery effect at scale across more than 3,500 GPU devices and 11 cloud providers. While micro-benchmark performance varies by less than 9% at the execution level, the difference reaches up to 38% across providers, revealing a pronounced cloud lottery effect. Using an LLM workload, we further show that an 8% micro-benchmark level performance variation can translate into up to a 1.77× difference in tokens per dollar for identical GPU models within a single day. Future work may explore how these insights can inform adaptive resource selection and scheduling strategies that explicitly account for performance variability and pricing dynamics in cloud environments.

References

[1] OpenAI. GPT-4 Technical Report. Technical report, OpenAI, 2024. URL https://arxiv.org/abs/2303.08774.

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. The LLaMA 3 Herd of Models. *arXiv preprint arXiv:2404.14219*, 2024.

[3] DeepSeek-AI. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv preprint arXiv:2405.04434*, 2024.

[4] Albert Q. Jiang et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

[5] Mistral AI. Mixtral of Experts. *arXiv preprint arXiv:2401.04088*, 2024.

[6] Min Si, Pavan Balaji, Yongzhou Chen, Ching-Hsiang Chu, Adi Gangidi, Saif Hasan, Subodh Iyengar, Dan Johnson, Bingzhe Liu, Regina Ren, et al. Collective Communication for 100k+ GPUs. *arXiv preprint arXiv:2510.20171*, 2025.

[7] Ivan Goldwasser, Harry Petty, Pradyumna Desale, and Kirthi Devleker. NVIDIA GB200 NVL72 Delivers Trillion-Parameter LLM Training and Real-Time Inference. *NVIDIA Technical Blog*, 2024.

[8] Prasoon Sinha, Akhil Guliani, Rutwik Jain, Brandon Tran, Matthew D Sinclair, and Shivaram Venkataraman. Not All GPUs Are Created Equal: Characterizing Variability in Large-Scale, Accelerator-Rich Systems. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 01–15. IEEE, 2022.

[9] Bilge Acun, Phil Miller, and Laxmikant V Kale. Variation Among Processors Under Turbo Boost in HPC Systems. In *Proceedings of the 2016 International Conference on Supercomputing*, pages 1–12, 2016.

[10] Yuichi Inadomi, Tapasya Patki, Koji Inoue, Mutsumi Aoyagi, Barry Rountree, Martin Schulz, David Lowenthal, Yasutaka Wada, Keiichiro Fukazawa, Masatsugu Ueda, et al. Analyzing and Mitigating the Impact of Manufacturing Variability in Power-Constrained Supercomputing. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pages 1–12, 2015.

[11] D. Chasapis, M. Moreto, M. Schulz, B. Rountree, M. Valero, and M. Casas. Power Efficient Job Scheduling by Predicting the Impact of Processor Manufacturing Variability. In *Proceedings of the ACM International Conference on Supercomputing (ICS)*, 2019. doi: 10.1145/3330345.3330372.

[12] Xinxin Mei, Xiaowen Chu, Hai Liu, Yiu-Wing Leung, and Zongpeng Li. Energy Efficient Real-Time Task Scheduling on CPU-GPU Hybrid Clusters. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, 2017. doi: 10.1109/INFOCOM.2017.8057205.

[13] Alexander Breuer, Yifeng Cui, and Alexander Heinecke. Petaflop Seismic Simulations in the Public Cloud. In *International Conference on High Performance Computing*, pages 167–185. Springer, 2019.

[14] Michael Davies, Neal Crago, Karthikeyan Sankaralingam, and Christos Kozyrakis. Efficient LLM Inference: Bandwidth, Compute, Synchronization, and Capacity are All You Need. *arXiv preprint arXiv:2507.14397*, 2025.

[15] Pol García Recasens, Ferran Agulló, Jordi Torres, et al. Unveiling GPU Bottlenecks in Large-Batch LLM Inference. In *IEEE International Conference on Cloud Computing (CLOUD)*, 2025.

[16] Amir Gholami et al. AI and Memory Wall. *arXiv preprint arXiv:2403.14123*, 2024.

[17] Javier Conde, Miguel González, Pedro Reviriego, Zhen Gao, Shanshan Liu, and Fabrizio Lombardi. Speed and Conversational Large Language Models: Not All Is About Tokens per Second. *Computer*, 57(8):74–80, 2024. doi: 10.1109/MC.2024.3399384.

[18] Jon Saad-Falcon, Avanika Narayan, Hakki Orhun Akengin, J Griffin, Herumb Shandilya, Adrian Gamarra Lafuente, Medhya Goel, Rebecca Joseph, Shlok Natarajan, Etash Kumar Guha, et al. Intelligence per Watt: Measuring Intelligence Efficiency of Local AI. *arXiv preprint arXiv:2511.07885*, 2025.

[19] Hugging Face, Inc. Hugging Face: The AI community building the future. https://huggingface.co, 2023.

[20] NVIDA. Llama-3.1-Nemotron-Nano-4B-v1.1. https://huggingface.co/nvidia/Llama-3.1-Nemotron-Nano-4B-v1.1, May 2025. Accessed via Hugging Face.

[21] Woosuk Kwon, Zhuohan Li, Shibo Zhuang, Ying Sheng, Lin Zheng, Cody Yu, Joseph E. Gonzalez, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with vLLM. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*. ACM, 2023. doi: 10.1145/3600006.3613146.

A Supplemental Figures

A.1 Execution-Level Performance Variation

Figure 7 shows the normalized FP16 MatrixMul throughput and D2D memory bandwidth for all GPU devices with at least three measurements. For each device, performance values are normalized to the per-device mean. Data points clustering around $y = 1$ indicates stable and consistent execution-level performance on the same physical GPU device.

A.2 GPU Performance Distribution

Figure 8 presents the performance distributions of A100 families across cloud providers. These figures follow the same benchmarking methodology and visualization conventions as Figure 2 and Figure 3. They are included as supplemental evidence to demonstrate that the provider-dependent performance variability observed for H100 and H200 GPUs also applies to earlier A100 product line.

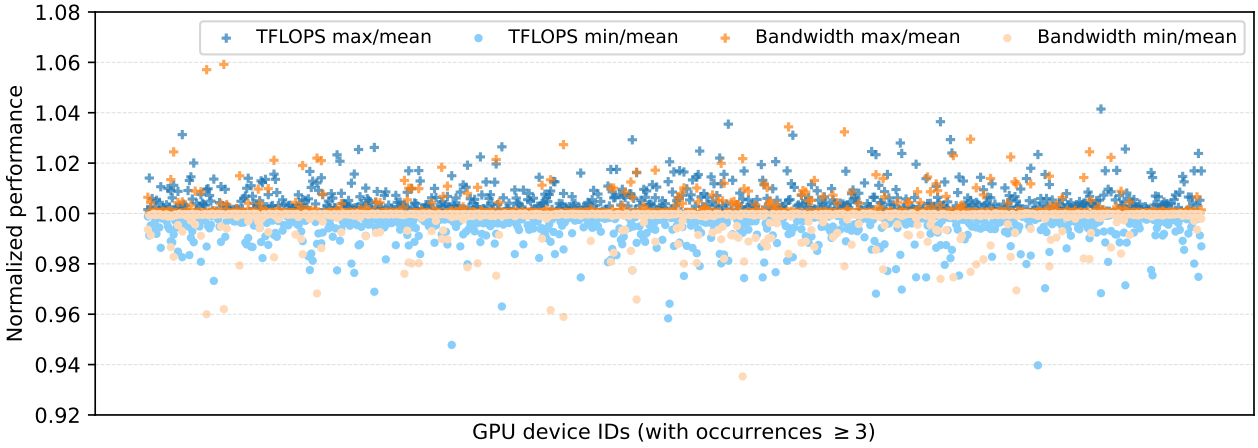
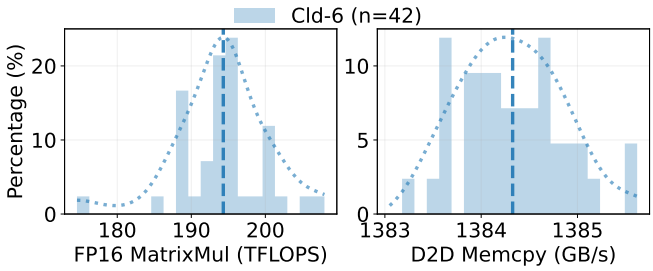
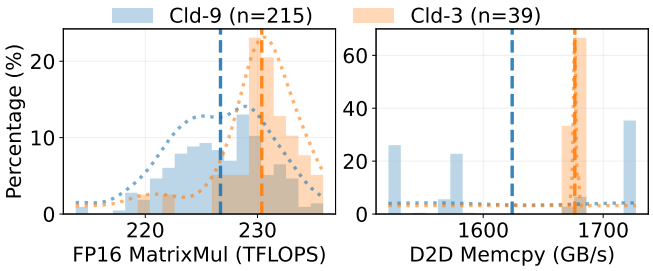


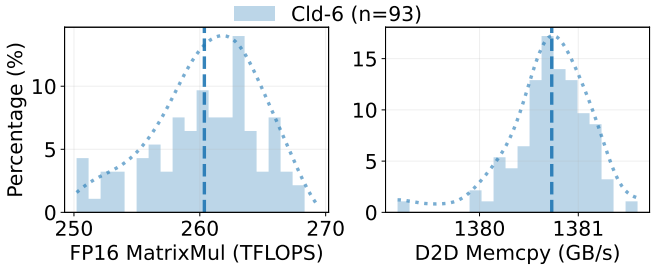
Figure 7: Normalized FP16 throughput and memory bandwidth across 746 GPU devices.



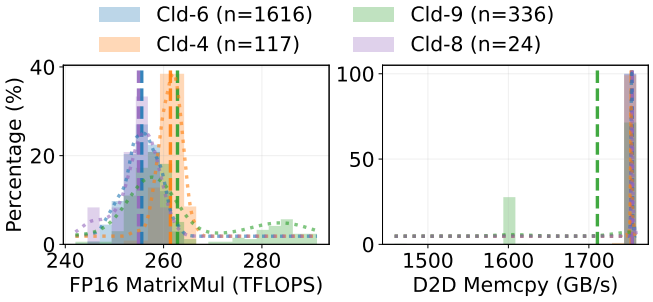
(a) NVIDIA A100 PCIe 40GB



(b) NVIDIA A100 PCIe 80GB



(c) NVIDIA A100 SXM4 40GB



(d) NVIDIA A100 SXM4 80GB

Figure 8: GPU core and memory performance distributions of NVIDIA A100 GPUs across cloud providers.