TECHNICAL VALIDATION OF AI-OPTIMIZED HARDWARE AND SOFTWARE ECOSYSTEM

Abstract

This report presents an empirical validation of an AMD MI300X-based high-performance computing (HPC) system, designed for machine learning and AI workloads. The study assesses operational independence from cloud-based resources, and evaluates thermal performance under inference workloads. Validation procedures included throughput testing using Stable Diffusion v1-4, a text-to-image generative model, and Whisper, an automatic speech recognition (ASR) model. Key focus areas were network traffic analysis, GPU utilization, and thermal diagnostics to determine on-device processing integrity and system reliability under sustained load. Results from three experiments confirm the system's ability to execute compute-intensive models on bare-metal infrastructure, maintain stable thermal profiles across active and inactive GPU groups, and demonstrate efficient model throughput scaling, all without external dependencies. The findings validate the system's suitability for diverse AI workloads requiring local execution, model flexibility, and thermal efficiency.

Introduction

This study aims to validate the operational independence and reliability of a high-performance computing (HPC) system purpose-built for AIML and AI inference workloads. The system under evaluation incorporates AMD MI300X GPUs and is designed to run entirely on bare-metal infrastructure without offloading compute tasks to external cloud services.

Validation efforts focused on two dimensions: (1) verifying that inference workloads were executed locally by monitoring network traffic for external activity, and (2) assessing thermal behavior under load to ensure consistent and expected temperature profiles across GPU cards. Two models representative of real-world AI applications were selected: Stable Diffusion v1-4, a generative model that converts text prompts into images, and Whisper, a model for transcribing speech to text. These workloads provided meaningful inference operations to test system performance under varying load and batch-size conditions.

Date Published: May 2025

Methodology

The validation methodology consisted of the following five phases:

1. **System Configuration and Environment Setup:**
   We evaluated the setup process and compatibility of the AMD-based HPC system with standard machine learning frameworks. This included adapting open-source repositories to run on AMD hardware using ROCm libraries, in place of CUDA, ensuring support for PyTorch-based workloads without vendor lock-in.
2. **Model Selection and Integration:**
   Two models were selected based on their widespread usage and workload diversity: Stable Diffusion v1-4 for generative text-to-image inference and Whisper for automatic speech recognition. Both models were sourced via Hugging Face to maintain reproducibility and accessibility.
3. **Profiling and Instrumentation:**
   Custom profiling code was developed to measure performance characteristics, including inference throughput and thermal behavior. Metrics were collected via on-device sensors and supplemented with external thermal-imaging instrumentation. This phase also included monitoring network traffic at the system level to ensure that model execution—such as image generation via Stable Diffusion—was performed locally on the AMD GPUs without transmitting data to external compute resources. Traffic logs were analyzed to confirm the absence of outbound responses that would indicate remote inference.
4. **Thermal and Network Profiling:**
   Thermal diagnostics were performed using a TOPDON TC004 thermal imaging device mounted on a tripod for stability. Network activity was concurrently monitored to verify that inference occurred locally on the device with no external API calls or cloud-based computation.
5. **Throughput Testing:**
   Stable Diffusion was used to evaluate system performance under varying batch sizes and process counts. Whisper was evaluated with smaller-scale tests to measure average inference time per audio sample and to confirm functional compatibility with the ROCm stack.

Date Published: May 2025

Experimental Setup

Experiment One: Bare-Metal Validation

The hardware configuration consisted of 16 AMD MI300X accelerator cards, physically arranged in groups of four to facilitate workload isolation and thermal comparison. Stable Diffusion v1-4, a text-to-image generative AI model, was used as the primary workload. The model was configured to generate 512×512 pixel images from textual prompts, where each image is composed of 512 horizontal by 512 vertical pixels—representing a moderately high-resolution output commonly used in image generation tasks.

Inference tests were conducted using the following batch configurations: 1×512, 2×256, 4×128, 8×64, 16×32, 32×16, 64×8, and 128×4. In this context, a batch refers to the number of inference requests processed simultaneously, and the format *N*×*M* indicates that *N* processes each handle a batch of *M* images. For example, 2×256 refers to two processes each generating 256 images per batch.

To evaluate thermal behavior, a TOPDON TC004 thermal imaging device was used to record surface temperatures. The camera was mounted on a tripod and aligned with the GPU array for consistent framing across runs. Testing was performed by activating a defined set of GPUs—first cards [12–15], then cards [0–3]—allowing comparative analysis of thermal output per quadrant. Cool-down phases were introduced between runs to track system recovery. Idle GPU temperatures ranged from 90–100°F, while actively loaded GPUs reached 115–130°F, consistent with AMD's expected operational range under sustained AI inference workloads.

The server’s physical layout is as follows: cards [0–3] are located in the top left quadrant; [4–7] in the top right; [8–11] in the bottom left; and [12–15] in the bottom right. AMD’s XGMI interconnect links GPUs [0–7] and [8–15] into two high-bandwidth domains, while inter-domain communication occurs over PCIe.

Image 1: Physical Layout of AMD MI300X GPU Cards and Junctions in 4x4 Configuration

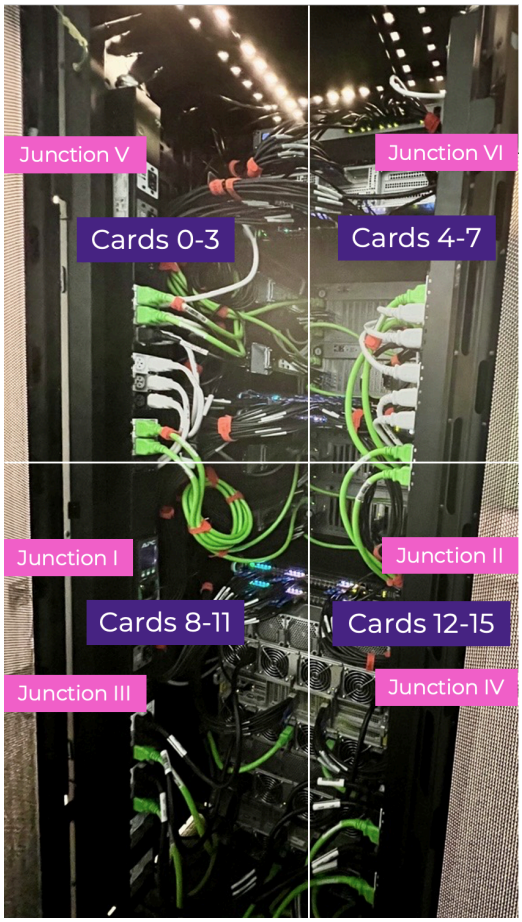

Image 1 shows the server rack contains 16 MI300X cards arranged into four logical quadrants: [0–3], [4–7], [8–11], and [12–15]. Physical junctions (I–VI) are labeled to indicate cable aggregation and airflow channels associated with each GPU cluster. This configuration facilitates isolated testing for thermal profiling and workload distribution.

Experiment Two: Stable Diffusion Throughput

This experiment focused on quantifying the throughput and latency performance of the system under realistic AIML workloads, without additional thermal imaging instrumentation. The Stable Diffusion v1-4 model was again used to generate 512×512 pixel images from text prompts. The same batch configurations were tested: 1×512, 2×256, 4×128, 8×64, 16×32, 32×16, 64×8, and 128×4.

Unlike Experiment One, this test relied solely on system-level telemetry to assess performance. Network traffic and onboard sensors were monitored to

Date Published: May 2025

capture inference duration, throughput (images per second), and GPU utilization across varying configurations.

Experiment Three: Whisper Model Evaluation

This experiment evaluated the system's compatibility with automatic speech recognition (ASR) workloads using the Whisper model. Whisper was selected due to its widespread use and computational demand in transcribing audio to text, providing a contrasting workload to the image generation tasks used in earlier experiments.

Audio clips were used as input data, and Whisper was executed to generate corresponding text transcriptions. Inference latency per sample was recorded to assess runtime performance under AMD ROCm environments.

Model Compilation

To enhance performance on AMD hardware, the Whisper model was compiled using PyTorch's `torch.compile` function with the 'inductor' backend and 'max-autotune' optimization mode. This compilation step aimed to maximize inference efficiency by leveraging kernel fusion and runtime tuning during execution.

Results

Experiment One: Bare-Metal Validation

This experiment evaluated thermal behavior and system independence under controlled GPU activation. Results confirmed that inference workloads were executed locally and that thermal characteristics were consistent with vendor-reported expectations for sustained AIML workloads, confirming nominal thermal operation.

Network Traffic Analysis

Network monitoring during Stable Diffusion inference showed negligible transmission activity. Both sent and received data remained within kilobyte-scale, validating that all inference operations were conducted on the local hardware.

Figure 1: Network Activity During 8-Card Inference Run

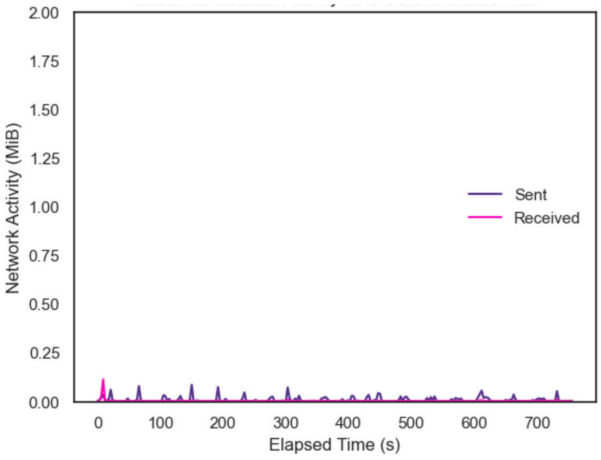

Figure 1 displays network traffic over time, showing minimal inbound and outbound data during inference. This confirms that all computation occurred locally without cloud-based execution.

Thermal Imaging and Sensor Observations

Thermal readings from the tripod-mounted TOPDON TC004 device aligned with expected operational ranges. Idle GPUs remained within the 88–100°F range, while active cards consistently reached 115–130°F without exceeding thermal thresholds.

Image 2: Thermal Imaging During Inference on Designated GPU Groups

Date Published: May 2025

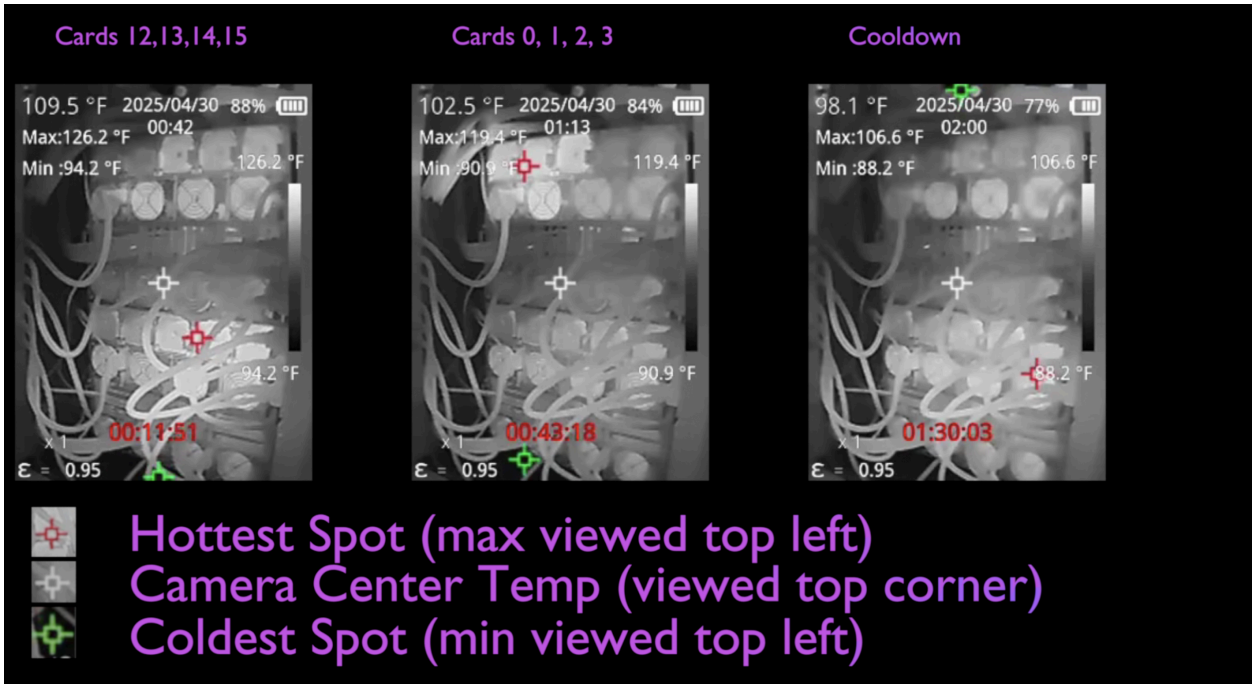

Image 2 depicts thermal infrared imagery comparing GPU surface temperatures across three states: inference on cards [12–15], inference on cards [0–3], and post-inference cooldown. Hottest (red), coldest (green), and central (white) temperature readings are marked. Observed thermal profiles validate expected heat distribution and recovery behavior consistent with GPU activation patterns.

Date Published: May 2025

Figure 2: GPU Utilization During 8-Card Inference Run

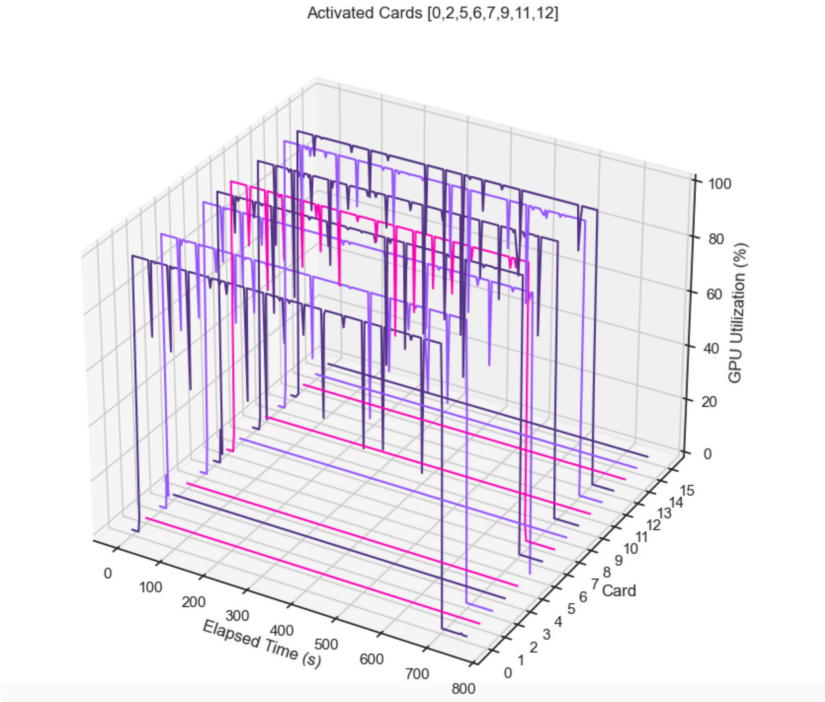

Figure 2 shows 3D plot illustrating GPU utilization across all 16 cards. Highlights selective activation of 8 cards while the remaining 8 remain idle, confirming proper workload distribution.

Date Published: May 2025

Figure 3: GPU Memory Utilization During 8-Card Inference Run

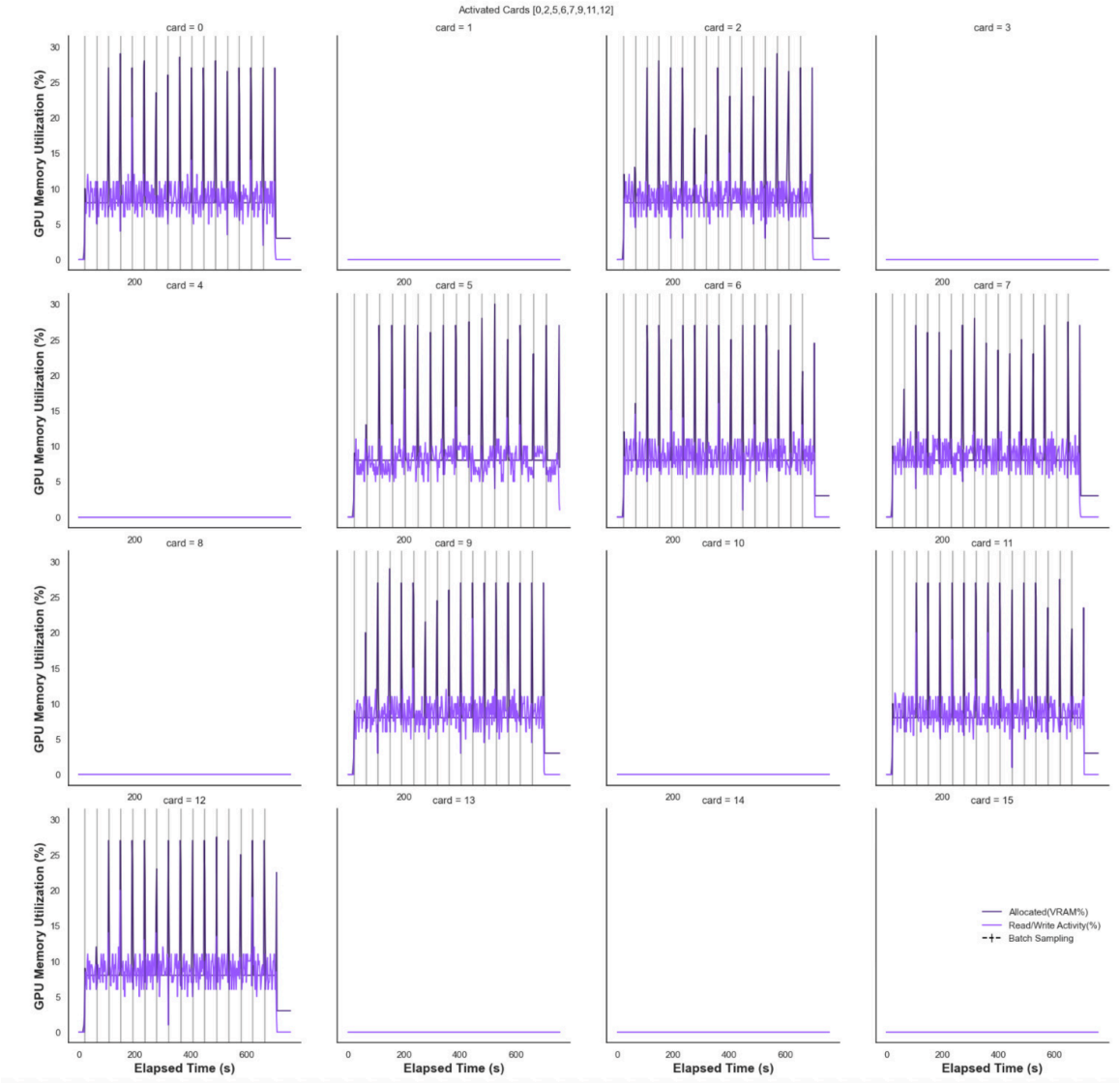

Figure 3 shows GPU memory utilization shows sustained usage patterns that correspond with inference sampling intervals. The vertical lines represent the time which sampling occurred for each batch.

Date Published: May 2025

Figure 4: Temperature Profiles by Card (Memory and Junction Sensors) During 8 Card Inference Run

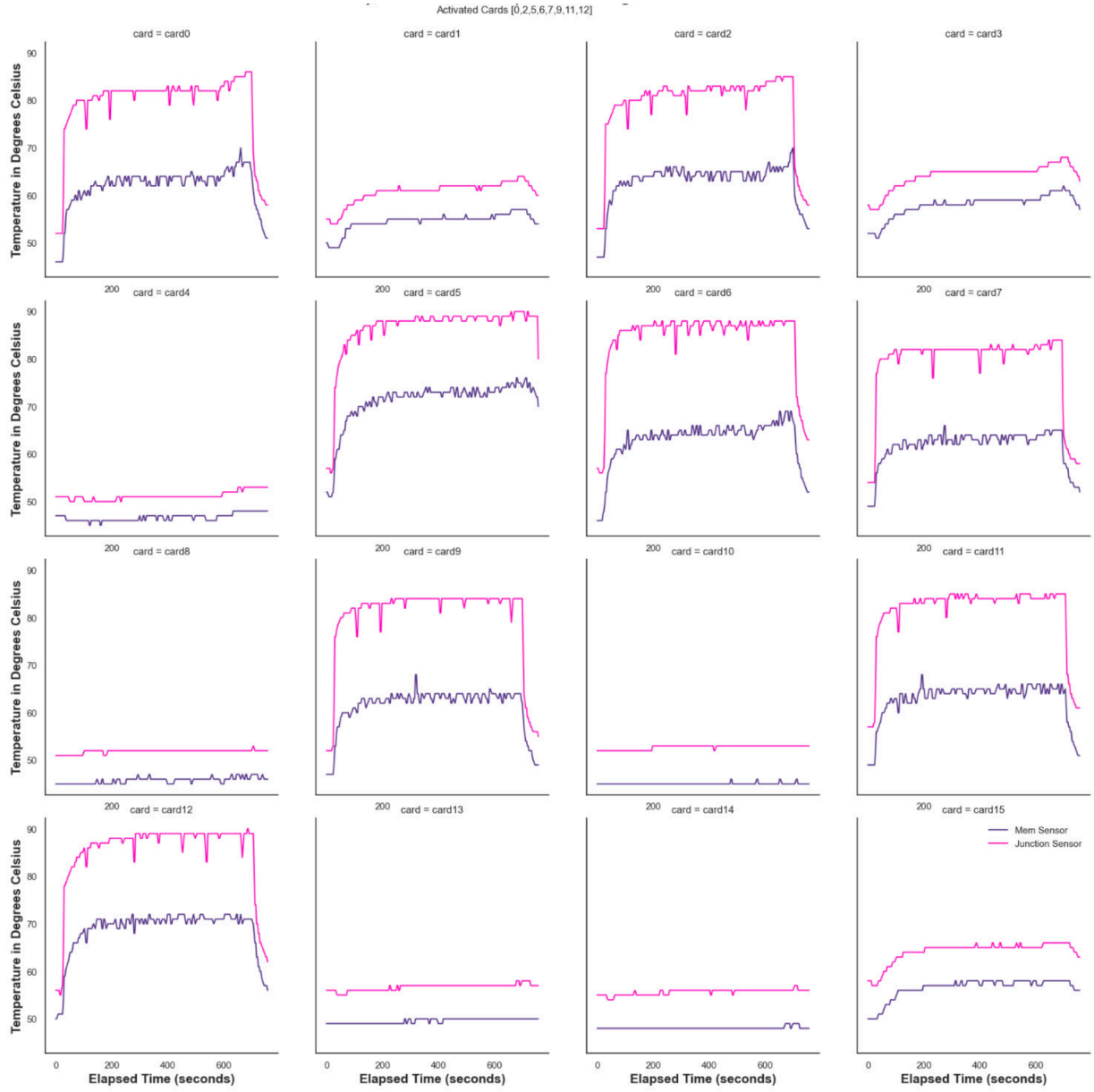

Figure 4 shows thermal trends per card during the 8-card inference run. Memory is in pink, and junction is in purple. Active cards exhibit higher junction and memory temperatures than idle cards, demonstrating effective thermal containment.

Date Published: May 2025

Figure 5: GPU Power Consumption by Card During 8 Card Inference Run

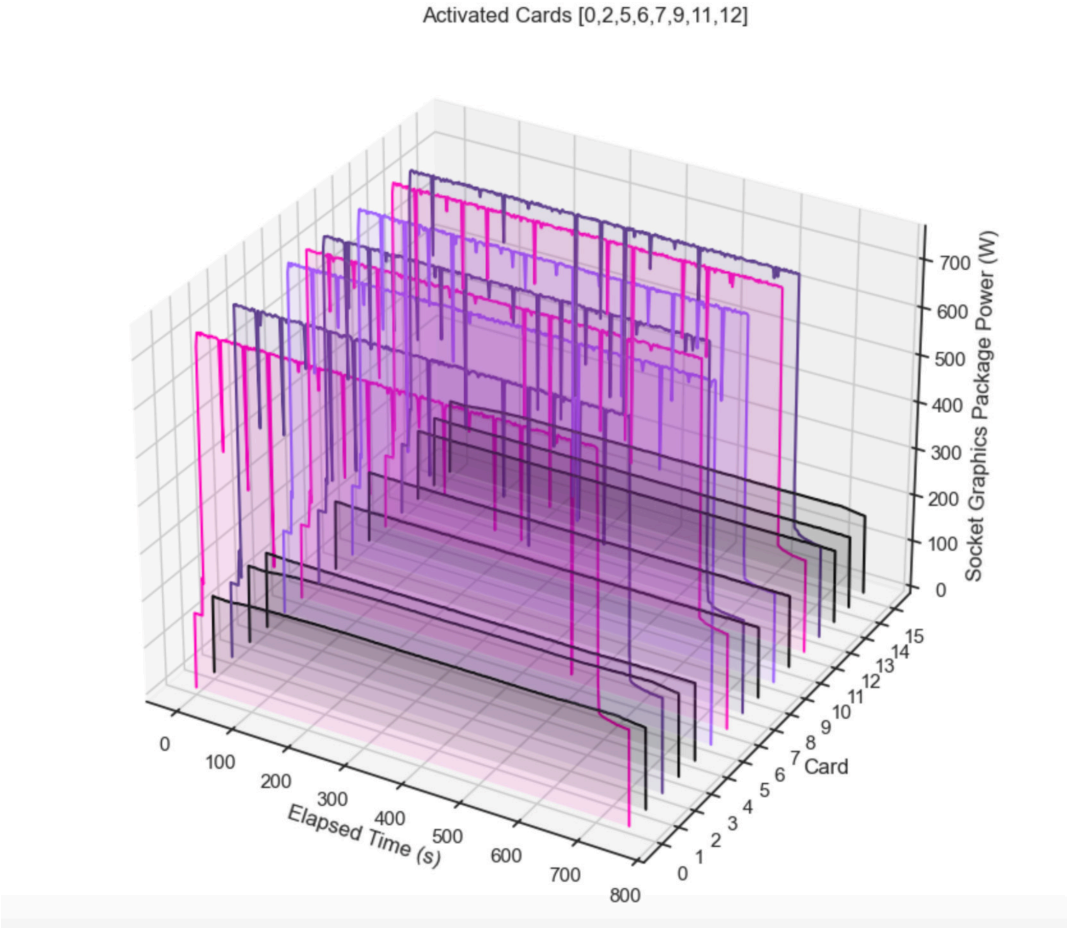

Figure 5 shows power consumption traces the activation of specific cards. Consistent draw patterns indicate sustained compute load without anomalies or power spikes.

Summary of Findings

The system exhibited predictable thermal and power behavior with no anomalous spikes. Network traffic confirmed no reliance on cloud APIs or external endpoints during execution.

Experiment Two: Stable Diffusion Throughput

This experiment evaluated inference latency across a matrix of batch sizes and process counts. Thermal imaging was excluded, relying instead on onboard sensors and profiling instrumentation.

Date Published: May 2025

Normalized Latency Per Image
Latency decreased with increasing batch size and process count, up to a point of diminishing returns. Efficiency gains plateaued after 4–8 processes, indicating potential overhead from inter-process coordination.

Appendix A: Full latency distribution tables, including mean, median, min, max, and standard deviation values.

Overall Inference Time Per Batch
Raw batch inference time scaled linearly with batch size. As expected, larger batches (e.g., 128x4) introduced greater memory and processing demands, with a corresponding increase in total inference time.

Figure 6: 3D Surface Plot of Median Batch Inference Latency

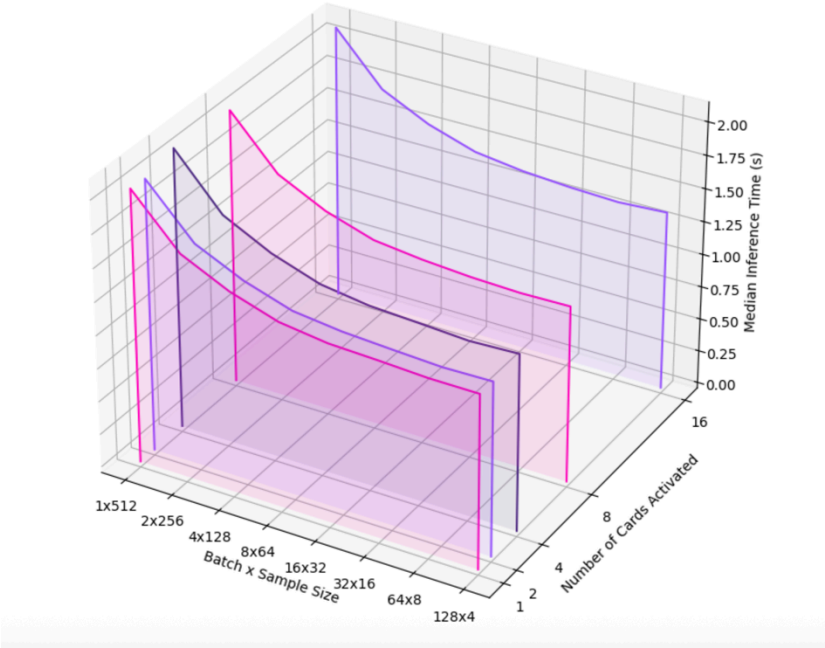

Figure 6 shows 3D surface plot of median batch inference latency, highlighting nonlinear scaling effects at higher GPU and batch counts.

Date Published: May 2025

Figure 7: Inference Speed Across Batch Size and GPU Count

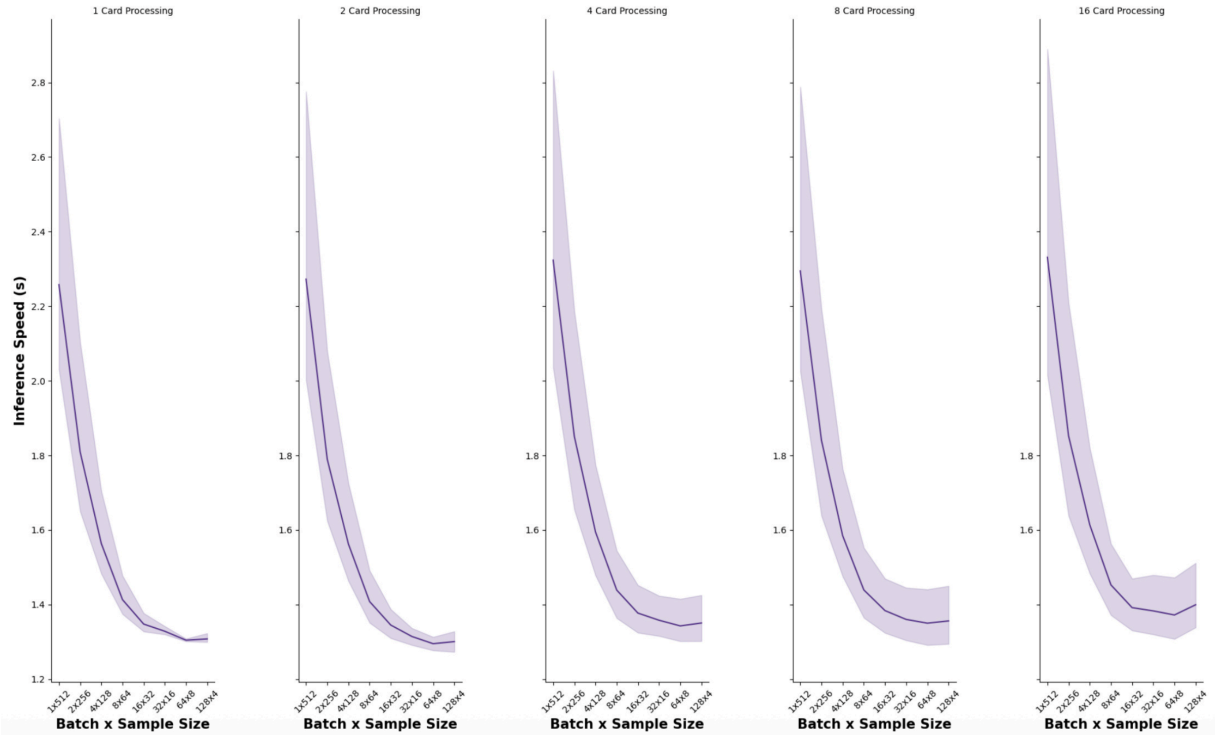

Figure 7 shows line plots showing min, median, and max inference time across multiple batch/sample sizes and GPU counts (1, 2, 4, 8, 16 cards). Demonstrates consistent improvements in latency as batch size increases.

Appendix B: Full batch latency datasets, including results with and without compilation overhead.

Summary of Findings

The system showed consistent performance scaling with increasing batch size and GPU process counts. Performance benefits diminished after a moderate level of parallelism, aligning with standard compute scaling trends.

Experiment Three: Whisper Model Evaluation

This experiment assessed the system's ability to execute an automatic speech recognition (ASR) model, Whisper, compiled using `torch.compile` for performance optimization. The goal was to verify runtime compatibility and measure token throughput under inference conditions.

Date Published: May 2025

Model Execution and Optimization
The Whisper model was compiled using PyTorch's `torch.compile` with the 'inductor' backend and 'max-autotune' setting. Compilation completed without errors, and the model executed successfully on AMD MI300X hardware, confirming compatibility with the ROCm stack.

Token Throughput Results
Inference was performed on a single GPU using audio-based input. Performance results were extracted from the inference logs and are summarized below:

- **Total Tokens Generated:** 49,664
- **Tokens per Second per GPU:** 3,507.09
- **Total Tokens per Second:** 3,507.09

Summary of Findings
The Whisper model executed reliably and achieved stable throughput during evaluation. With a processing rate of over 3,500 tokens per second, the system demonstrated functional support for ASR workloads alongside vision-based generative models.

Conclusion

This validation study demonstrates that the AMD MI300X-based HPC system performs reliably and efficiently across a range of AI inference workloads. Through a combination of thermal profiling, network activity monitoring, and performance benchmarking using real-world models, the system was confirmed to operate independently of cloud resources and to maintain consistent thermal and utilization characteristics.

In Experiment One, the absence of significant network traffic during Stable Diffusion inference confirmed that all compute operations were executed locally. Thermal imaging showed expected heat distribution and recovery behavior aligned with GPU activation patterns, further validating the system's physical and operational design.

In Experiment Two, inference latency decreased with increasing batch size and GPU parallelism up to a point, after which gains plateaued. These results align with expected compute scaling behavior and confirm the system's ability to handle large-scale generative workloads efficiently.

Date Published: May 2025

In Experiment Three, the Whisper ASR model compiled and executed successfully on the ROCm stack, achieving high token throughput and validating compatibility with non-vision model architectures.

Together, these findings confirm the system's suitability for AI-centric workloads requiring on-premises compute, thermally stable operation, and cross-model framework support.

Date Published: May 2025

Appendices

Appendix A:

Efficiency metrics demonstrate performance scalability and efficiency changes at various batch sizes and process counts.

Mean of Normalized Inference Latency per Image Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.04937738	1.67688545	1.50562806	1.38864324	1.33875865	1.32402607	1.3038516	1.30606513
2	AMD Instinct MI300X	2.0514481	1.67303323	1.50188426	1.38439498	1.334242	1.31144887	1.29422006	1.29530868
4	AMD Instinct MI300X	2.11751366	1.72171293	1.54571841	1.42257879	1.3745507	1.35176751	1.33468633	1.34251853
8	AMD Instinct MI300X	2.08910996	1.70630638	1.52815795	1.41297805	1.36694539	1.33996125	1.32829781	1.336932
16	AMD Instinct MI300X	2.10870393	1.7306104	1.55280419	1.43723724	1.38852501	1.36064278	1.34712118	1.37611377

Median of Normalized Inference Latency per Image Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.048460086	1.675569097	1.504634218	1.387609772	1.338320785	1.323200667	1.303472233	1.301002677
2	AMD Instinct MI300X	2.039970183	1.666684323	1.499126876	1.381867574	1.337768512	1.314964406	1.294937348	1.30062523
4	AMD Instinct MI300X	2.103728161	1.712470239	1.531889895	1.4084997	1.35503809	1.334471879	1.311314378	1.324358238
8	AMD Instinct MI300X	2.071142628	1.692442744	1.516141827	1.400713709	1.35786866	1.331067788	1.317942467	1.323969139
16	AMD Instinct MI300X	2.09040995	1.711248491	1.538558776	1.423536124	1.375206415	1.349699635	1.335974859	1.357222542

Standard Deviation of Normalized Inference Latency per Image Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	0.039182528	0.027147569	0.018117917	0.011638884	0.007427662	0.00480842	0.002039626	0.011180294
2	AMD Instinct MI300X	0.05022018	0.035624273	0.028245156	0.020823601	0.016884825	0.015228753	0.015720836	0.020295172
4	AMD Instinct MI300X	0.06636672	0.047402951	0.043071729	0.040119203	0.041222227	0.039485172	0.044057169	0.043496214
8	AMD Instinct MI300X	0.063374229	0.048592105	0.03947595	0.03592266	0.037792394	0.036537065	0.041532379	0.043649067
16	AMD Instinct MI300X	0.067739066	0.054245586	0.044083278	0.039921054	0.036478602	0.034250762	0.035374829	0.046480455

Minimum of Normalized Inference Latency per Image Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.01280901	1.6497543	1.48249553	1.37439937	1.32773081	1.31985466	1.30130218	1.29945726
2	AMD Instinct MI300X	1.9892524	1.6242746	1.46312807	1.35103946	1.30978927	1.29135364	1.27712383	1.27312046
4	AMD Instinct MI300X	2.01280646	1.65398145	1.47800774	1.36359664	1.32448185	1.31565482	1.30175421	1.30194866
8	AMD Instinct MI300X	2.00933665	1.63771693	1.47543046	1.36447407	1.32391899	1.30406315	1.29145796	1.29438778

Date Published: May 2025

16	AMD Instinct MI300X	1.98884452	1.63314601	1.48334736	1.37159906	1.33040549	1.31986681	1.30772843	1.32971598

Maximum of Normalized Inference Latency per Image Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.92184331	2.10286945	1.7040881	1.47727355	1.37689271	1.34169307	1.308369	1.32279791
2	AMD Instinct MI300X	3.02066629	2.07819498	1.72500152	1.49121443	1.38728978	1.3364744	1.31290286	1.32796377
4	AMD Instinct MI300X	3.0737592	2.18527845	1.77364023	1.544476	1.45191687	1.42342357	1.41504747	1.42505076
8	AMD Instinct MI300X	3.02687987	2.19131281	1.76296769	1.55244651	1.4693952	1.44504316	1.44059505	1.4497343
16	AMD Instinct MI300X	3.15538601	2.20775351	1.8199441	1.56292522	1.49577937	1.47913266	1.47236326	1.51122326

Appendix B:

Speed and latency metrics present the latency impact of varying batch sizes and process counts.

Mean Batch Inference Latency (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.04937738	3.35377089	6.02251223	11.1091459	21.4201383	42.3688341	83.4465025	167.176337
2	AMD Instinct MI300X	2.0514481	3.34606646	6.00753706	11.0751598	21.347872	41.9663639	82.8300837	165.79951
4	AMD Instinct MI300X	2.11751366	3.44342585	6.18287366	11.3806303	21.9928111	43.2565603	85.419925	171.842371
8	AMD Instinct MI300X	2.08910996	3.41261276	6.1126318	11.3038244	21.8711263	42.87876	85.0110596	171.127296
16	AMD Instinct MI300X	2.10870393	3.4612208	6.21121674	11.4978979	22.2164001	43.5405691	86.2157554	176.142563

Mean Batch Inference Latency with Overhead (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.05326045	3.35750884	6.02781091	11.1145124	21.4261503	42.3759959	83.4551713	167.183234
2	AMD Instinct MI300X	2.0562849	3.35077684	6.01421656	11.0815582	21.3550591	41.9747526	82.8401003	165.806787
4	AMD Instinct MI300X	2.1224004	3.44817848	6.18960965	11.3873803	22.0000574	43.2649855	85.4303632	171.849771
8	AMD Instinct MI300X	2.09390903	3.41714873	6.11903774	11.3104556	21.8780608	42.8868093	85.0209367	171.134947
16	AMD Instinct MI300X	2.1134949	3.46585856	6.21781897	11.5046974	22.2235046	43.5487896	86.2255751	176.149991

Median Batch Inference Latency (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.04846009	3.35113819	6.01853687	11.1008782	21.4131326	42.3424213	83.4222229	166.528343
2	AMD Instinct MI300X	2.03997018	3.33336865	5.99650751	11.0549406	21.4042962	42.078861	82.8759903	166.480029
4	AMD Instinct MI300X	2.10372816	3.42494048	6.12755958	11.2679976	21.6806094	42.7031001	83.9241202	169.517854
8	AMD Instinct MI300X	2.07114263	3.38488549	6.06456731	11.2057097	21.7258986	42.5941692	84.3483179	169.46805
16	AMD Instinct MI300X	2.09040995	3.42249698	6.15423511	11.388289	22.0033026	43.1903883	85.502391	173.724485

Median Batch Inference Latency with Overhead (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.05228547	3.35478727	6.02405715	11.1057544	21.4188836	42.3490363	83.4299891	166.532347
2	AMD Instinct MI300X	2.04456923	3.33794803	6.00257753	11.0619376	21.4111982	42.0898759	82.8887236	166.490507
4	AMD Instinct MI300X	2.10876148	3.42971152	6.13445825	11.2749947	21.6873721	42.7154409	83.9332451	169.526877
8	AMD Instinct MI300X	2.07582038	3.38916006	6.07064265	11.2120556	21.7311971	42.6043258	84.3568671	169.479389
16	AMD Instinct MI300X	2.09512208	3.42711112	6.16064313	11.3944493	22.010583	43.1999151	85.5118221	173.731471

Date Published: May 2025

Standard Deviation Batch Inference Latency (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	0.03918253	0.05429514	0.07247167	0.09311107	0.11884259	0.15386945	0.13053609	1.43107768
2	AMD Instinct MI300X	0.05022018	0.07124855	0.11298063	0.16658881	0.2701572	0.4873201	1.0061335	2.59778205
4	AMD Instinct MI300X	0.06636672	0.0948059	0.17228692	0.32095363	0.65955564	1.26352549	2.81965883	5.56751539
8	AMD Instinct MI300X	0.06337423	0.09718421	0.1579038	0.28738128	0.60467831	1.16918608	2.65807225	5.58708061
16	AMD Instinct MI300X	0.06773907	0.10849117	0.17633311	0.31936843	0.58365763	1.09602439	2.26398903	5.94949825

Standard Deviation Batch Inference Latency with Overhead (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	0.03960407	0.05489274	0.07318702	0.09417166	0.1201096	0.15563419	0.1324915	1.4359491
2	AMD Instinct MI300X	0.05052636	0.07163744	0.11329914	0.16725654	0.27052923	0.48797874	1.00590845	2.60022924
4	AMD Instinct MI300X	0.06652602	0.09528355	0.17274183	0.3211265	0.65991532	1.26330906	2.819671	5.56832499
8	AMD Instinct MI300X	0.06373757	0.09761949	0.15841658	0.28784039	0.60480792	1.16924444	2.65807841	5.58745754
16	AMD Instinct MI300X	0.06806096	0.1088199	0.17670834	0.31951163	0.58387727	1.0959963	2.26365179	5.94860981

Minimum Batch Inference Latency (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.01280901	3.29950859	5.92998213	10.995195	21.243693	42.2353492	83.2833397	166.330529
2	AMD Instinct MI300X	1.9892524	3.24854921	5.85251228	10.8083157	20.9566283	41.3233164	81.7359254	162.959419
4	AMD Instinct MI300X	2.01280646	3.30796291	5.91203096	10.9087731	21.1917096	42.1009542	83.3122694	166.649429
8	AMD Instinct MI300X	2.00933665	3.27543386	5.90172184	10.9157926	21.1827039	41.7300208	82.6533093	165.681636
16	AMD Instinct MI300X	1.98884452	3.26629202	5.93338943	10.9727925	21.2864878	42.235738	83.6946195	170.203645

Minimum Batch Inference Latency with Overhead (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.01593523	3.30337435	5.93536172	11.0006177	21.2494134	42.2426788	83.2911503	166.335829
2	AMD Instinct MI300X	1.99379033	3.25292941	5.85914429	10.8145572	20.9636421	41.331643	81.7455672	162.963513
4	AMD Instinct MI300X	2.01731929	3.31241412	5.91781562	10.9167669	21.1982575	42.1089589	83.3219129	166.653276
8	AMD Instinct MI300X	2.0128021	3.28052639	5.90814021	10.9220689	21.1907683	41.7377447	82.6622635	165.6862
16	AMD Instinct MI300X	1.99244164	3.27145581	5.93924869	10.9830173	21.2930679	42.2427844	83.702913	170.208359

Date Published: May 2025

Maximum Batch Inference Latency (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.92184331	4.2057389	6.8163524	11.8181884	22.0302834	42.9341783	83.7356163	169.318132
2	AMD Instinct MI300X	3.02066629	4.15638997	6.90000607	11.9297155	22.1966366	42.7671809	84.0257831	169.979363
4	AMD Instinct MI300X	3.0737592	4.3705569	7.09456093	12.355808	23.2306699	45.5495542	90.5630381	182.406497
8	AMD Instinct MI300X	3.02687987	4.38262562	7.05187076	12.4195721	23.5103232	46.2413811	92.1980833	185.56599
16	AMD Instinct MI300X	3.15538601	4.41550703	7.2797764	12.5034018	23.9324699	47.332245	94.2312487	193.436577

Maximum Batch Inference Latency with Overhead (Seconds) Across Runs

N-processes	Accelerator	1x512	2x256	4x128	8x64	16x32	32x16	64x8	128x4
1	AMD Instinct MI300X	2.93519776	4.21912168	6.82964362	11.8322366	22.0436549	42.948205	83.7497089	169.332411
2	AMD Instinct MI300X	3.03417587	4.1724408	6.91334813	11.9439843	22.2099778	42.7838285	84.0395535	169.994472
4	AMD Instinct MI300X	3.0870205	4.38794597	7.11045614	12.3686837	23.2375394	45.5572552	90.5730299	182.422601
8	AMD Instinct MI300X	3.04215726	4.3989565	7.0677703	12.435573	23.5170206	46.248684	92.2081061	185.570252
16	AMD Instinct MI300X	3.16835645	4.42896407	7.29380542	12.5192374	23.9393882	47.340783	94.2403654	193.442049

Date Published: May 2025

Appendix C: ResNet50 Fused Kernels

As an extension to the core validation experiments, engineering teams demonstrated two real-time, end-to-end ResNet50 application executions across varied image and batch sizes. These examples highlighted kernel-level optimization strategies in a fused execution environment.

Real-Time Demonstration

Two ResNet50 inference pipelines were shown live, operating on differing batch and input dimensions.

Demonstrations validated that the system could execute complex fused kernel operations without runtime instability.

Fused Kernel Architecture Overview

The fused kernel demonstrated integration across four distinct compute classes: FPGA, CPU, GPU, and ASIC.

This multi-class orchestration forms the basis for efficient and flexible hardware abstraction within AI workloads.

Future Roadmap Discussion

A future-state vision was shared in which fused kernel workloads would be distributed across multiple servers.

The orchestration layer would optimize kernel placement dynamically for throughput and energy efficiency.

Codebase Access and Highlights

Access to the fused kernel codebase was provided for review.

Code examples illustrated how smaller sub-kernels could be individually tuned for specific architectural features and datatypes.

These sub-kernels could then be integrated into the larger fused kernel framework as reusable, composable components.

Design Philosophy

The guiding principle is to iteratively optimize atomic kernels once and reuse them as stable, callable units within broader kernel compositions.

Over time, the system would expand the library of fused-in kernels, enabling scalable inference pipelines with minimal re-optimization overhead.

Date Published: May 2025