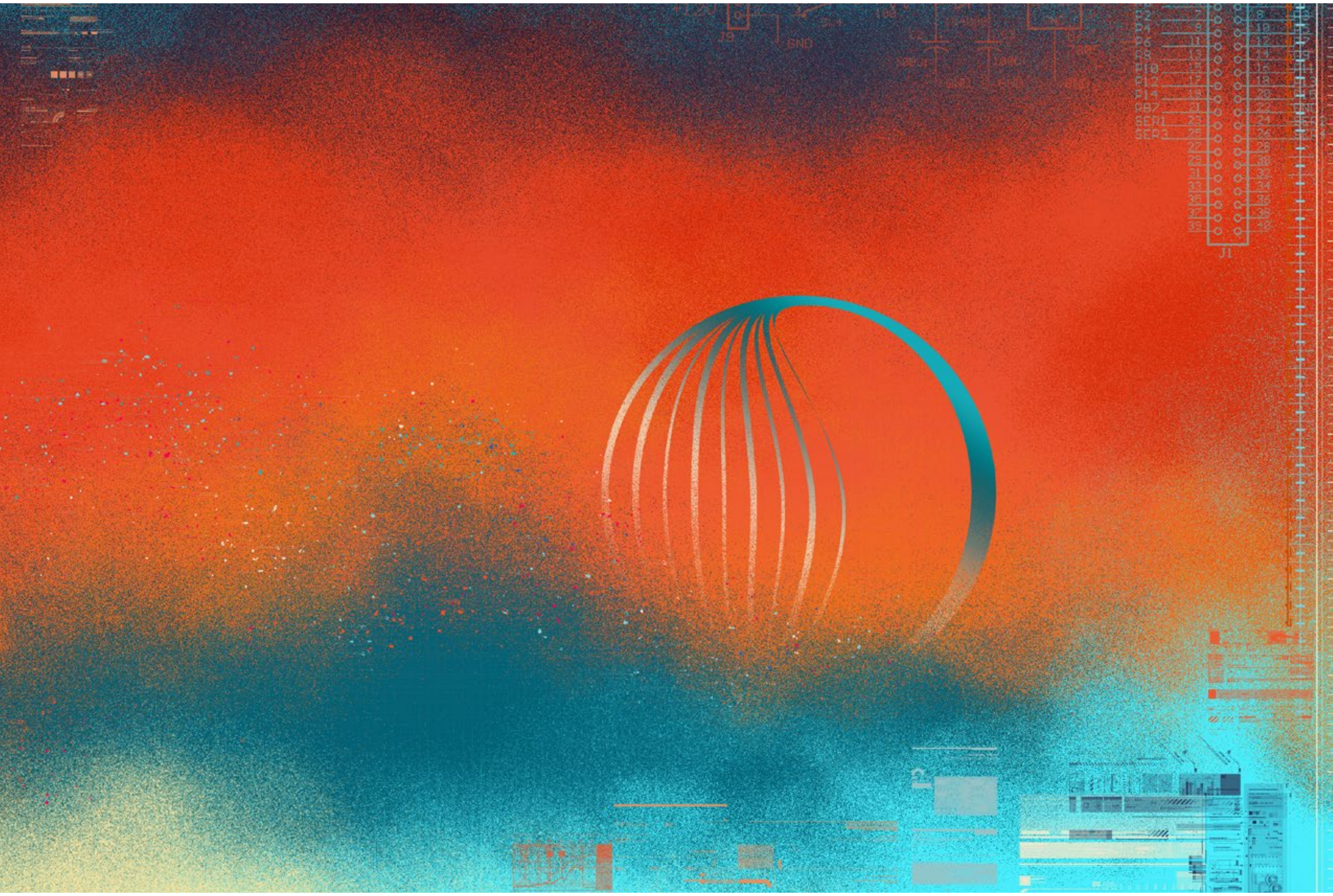
signal65



Improving AI Inference with AMD EPYC Host CPUs

AUTHOR

Mitch Lewis
Performance Analyst | Signal65

FEBRUARY 2026

IN PARTNERSHIP WITH

AMD

Executive Summary

AI workloads typically leverage GPU-based computation, and as a result, the performance considerations of AI are often heavily focused on GPUs. While GPUs are often the primary driver of AI performance, other infrastructure components can become a bottleneck. In particular, host CPUs provide a critical role for AI inference, even when utilizing GPUs for the bulk of the workload. In this context, the host CPU refers to the CPU inside the GPU server responsible for request handling, scheduling, and data movement between the application layer and GPUs.

This report explores the important role of the host CPU and evaluates the impact they have on AI performance. To isolate the impact of the host CPU, Signal65 conducted hands-on AI inference testing of two nearly identical systems. Both systems were configured with the same GPUs and technical specifications, with the key differentiator being the CPUs. One server was configured with AMD EPYC CPUs and the other with Intel Xeon CPUs.

Throughout this testing, Signal65 found AMD EPYC CPUs to provide notable performance improvements during AI inferencing. Key findings include:

- Up to 14.64% greater request throughput
- Up to 14.38% greater output throughput
- Up to 46.54% faster time to first token
- Up to 11.46% lower latency

Key Highlights

AMD EPYC host nodes achieved higher throughput, faster time to first token, and lower inter-token latency than Intel Xeon.

Testing demonstrated consistent performance advantages across 7 distinct AI models.

AMD EPYC High Frequency Processors are purpose built for AI workloads and present a practical approach to maximize the performance and cost-efficiency of large AI datacenters.

The Importance of Host CPUs for AI Inference

Large scale AI inference, as well as many other HPC workloads, is primarily reliant on GPU computation. As a result, the performance of AI workloads is often centered around GPU capability. As the primary driver of performance, this focus on GPU performance is well warranted, however, just as with all applications, truly maximizing performance requires a more holistic view of the infrastructure. While organizations are focused on deploying the highest performance GPUs, without careful consideration other components may become a bottleneck.

One key component for AI workloads is the host CPU. While GPUs typically dominate model execution, CPU host nodes play a vital role in controlling how efficiently work reaches the GPU. Core responsibilities of CPUs during AI inference include:

- Managing and routing inference requests
- Request batching and queue management
- Resource scheduling
- Data movement and serialization
- Returning inference results

Although GPUs are performing the bulk of the inference workload, these CPU-based tasks are additionally critical to maximizing AI inference performance. Choosing a less capable CPU can directly impact throughput, latency, and GPU efficiency, creating a potential bottleneck for AI workloads. Avoiding this bottleneck is crucial for organizations with significant GPU investments, making host CPU selection an additionally critical consideration.

For CPUs hosting GPUs for AI and HPC workloads, certain characteristics become increasingly important. In particular, high performance per core is key to supporting GPU workloads. Core frequency and IPC (instructions per cycle) become primary concerns, enabling faster execution of processes, reducing latency, and increasing GPU utilization. High memory bandwidth is also a crucial characteristic for enabling larger batch sizes and quickly retrieving embeddings and cached data. When investing in significant AI infrastructure, organizations should carefully consider such characteristics for their host node CPUs to maximize GPU efficiency and datacenter performance.

AI Inference with 5th Gen AMD EPYC High Frequency Processors

As part of a broad AI portfolio, AMD has developed several 5th Generation EPYC Processors that are purpose built for hosting GPU-accelerated workloads. These processors offer high performance per core, high frequency, and high memory bandwidth to avoid the CPU bottleneck that emerges with GPU-based AI and HPC workloads.

Key characteristics of AMD EPYC High Frequency Processors include:

- Up to 64 cores
- Up to 5 GHz frequency
- 12 channels DDR5 memory
- Up to 160 lanes PCIe Gen5
- Up to 245 MB cache

The combination of high frequency, large cache capacity, and high memory bandwidth make 5th Gen AMD EPYC processors uniquely well suited to handling the latency-sensitive scheduling and batching required of host CPUs during AI inference.

Testing Overview

To evaluate the performance impact of host CPUs on AI inference, Signal65 conducted a series of performance tests using two distinct host CPUs. All tests were run with both an AMD EPYC based node and an Intel Xeon based node. To isolate the impact of the CPU, both nodes were configured with the same GPUs. All other server specifications were kept identical, or as close as possible.

	AMD EPYC Node	Intel Xeon Node
CPUs	2x AMD EPYC 9575F	2x Intel Xeon 6960P
Cores / Threads	64 / 128	72 / 144
Max Frequency	5 Ghz	3.9 Ghz
GPUs	8x NVIDIA B200	8x NVIDIA B200
OS	Ubuntu 24.04	Ubuntu 24.04

Figure 1: *Host CPU and GPU Configuration Summary*

To create a comprehensive understanding of CPU-driven performance impact, testing was completed with various models, including different sizes, architectures, and model families. Testing included both single instance tests, with a single model deployed on the system, and a multi-instance test, with 8 individual instances of a model deployed on the system. To further create a broad comparison, testing utilized two distinct benchmarks – NVIDIA Genai-Perf and vLLM Bench. An overview of the models and test configurations can be seen in Figure 2.

Benchmark	Model	Instances
NVIDIA Genai-Perf		
	GPT-OSS-120B	Single Instance
	Llama-3.3-70B-Instruct	Single Instance
	Qwen2.5-Coder-Instruct	Single Instance
	Llama-3.1-8B-Instruct	Multi Instance (8x)
vLLM Bench		
	Llama-4-Scout-17B-16E-FP4	Single Instance
	DeepSeek-R1-FP4	Single Instance
	Qwen2.5-VL-72B-Instruct	Single Instance

Figure 2: *Test Overview*

Performance Results

The results of this testing found notable performance differences between the two CPU platforms tested, demonstrating the impact that host CPUs can have on AI inference. In general, the AMD EPYC system achieved consistent advantages across several key AI performance metrics.

Throughput

Throughput was measured for both the input of requests, as well as the output in tokens per second. For both throughput metrics, AMD EPYC consistently achieved higher throughput across all models and configurations tested.

The most notable throughput advantage was found when running GPT-OSS-120B. For this model, the AMD EPYC configuration achieved between 12.91% and 14.64% higher request throughput and between 12.27% and 14.38% higher output throughput, depending on the input/output shape.

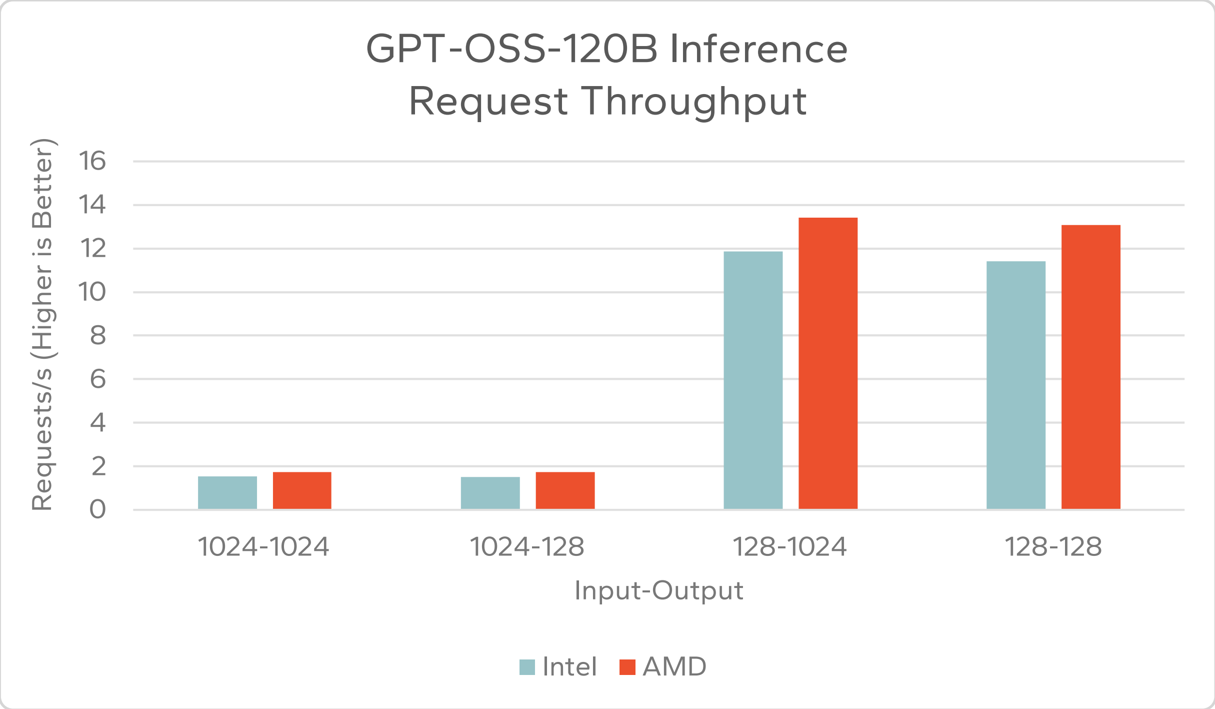


Figure 3: *GPT-OSS-120B Request Throughput*

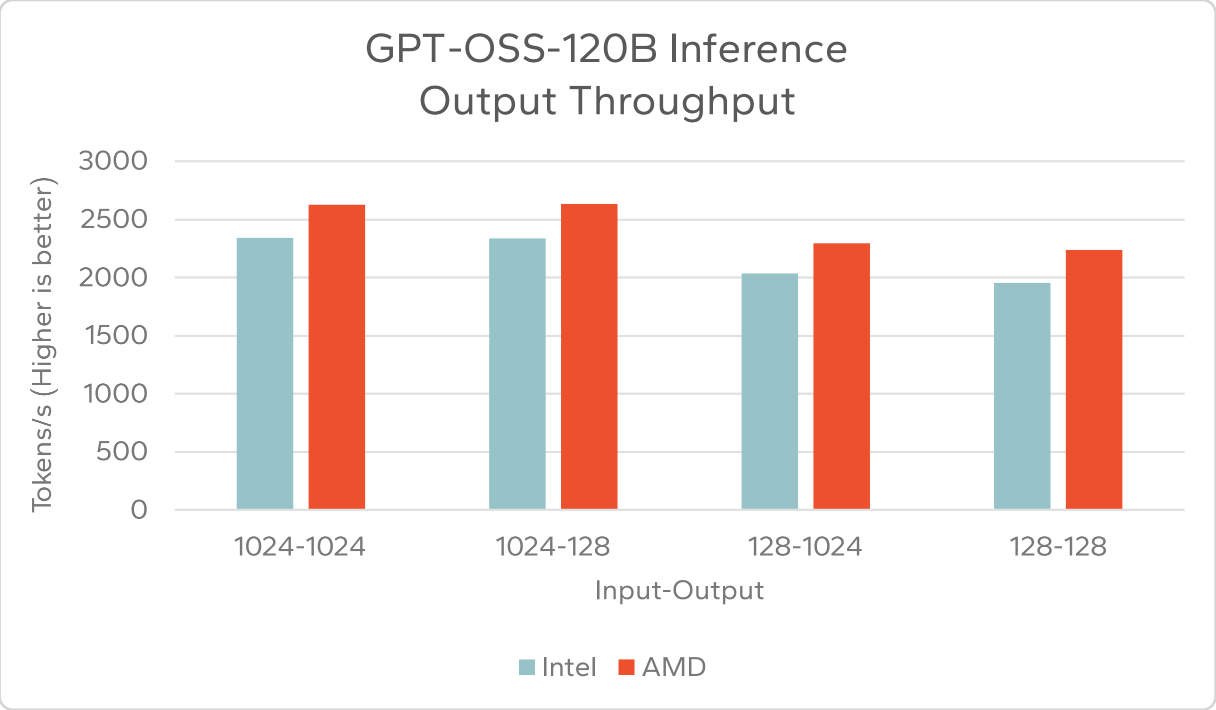


Figure 4: *GPT-OSS-120B Output Throughput*

AMD EPYC additionally achieved a notable throughput advantage in the multi-instance testing of Llama-3.1-8B-Instruct. For this model, AMD achieved both request and output throughputs between 9.5% and 11.2% higher than Intel Xeon.

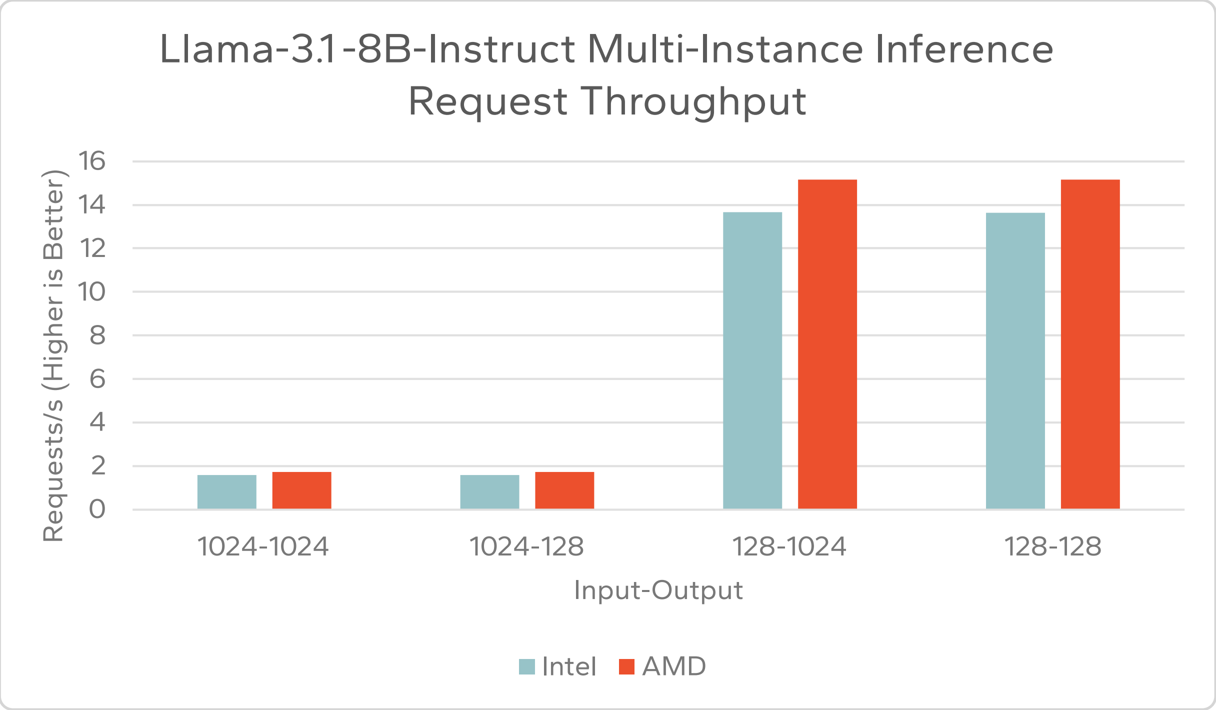


Figure 5: *Llama-3.1-8B-Instruct Request Throughput*

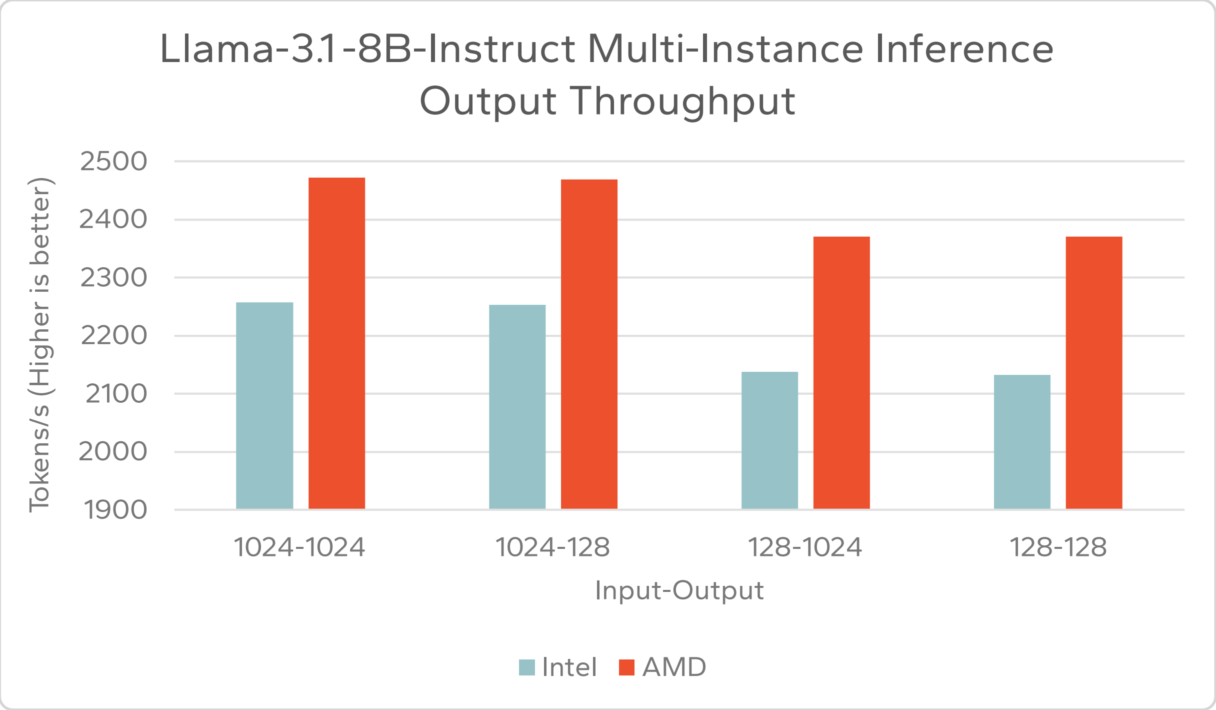


Figure 6: *Llama-3.1-8B-Instruct Output Throughput*

The lowest overall throughput advantages were found when testing Qwen2.5-Coder-Instruct, in which AMD EPYC outperformed Intel Xeon by approximately 3% for both throughput metrics. Although not as notable as the advantages found across other models tested, a 3% increase in throughput is still quite notable when considering the overall scale and cost of large AI datacenters.

Across all models tested, AMD EPYC consistently delivered higher request and output throughput. The observed performance gap between the two systems underscores the influence of host CPUs on overall throughput, with AMD demonstrating a clear and repeatable advantage.

Time to First Token

Time to first token is a metric that can be particularly sensitive to host node CPU performance due to the impact of request handling, batching, and scheduling overhead. Similar to the throughput results, AMD EPYC consistently achieved time to first token advantages over Intel Xeon for most models tested.

Just as with throughput, AMD EPYC achieved a significant time to first token advantage when running GPT-OSS-120B, ranging between 10.28% and 36.31% faster, depending on the input / output shape.

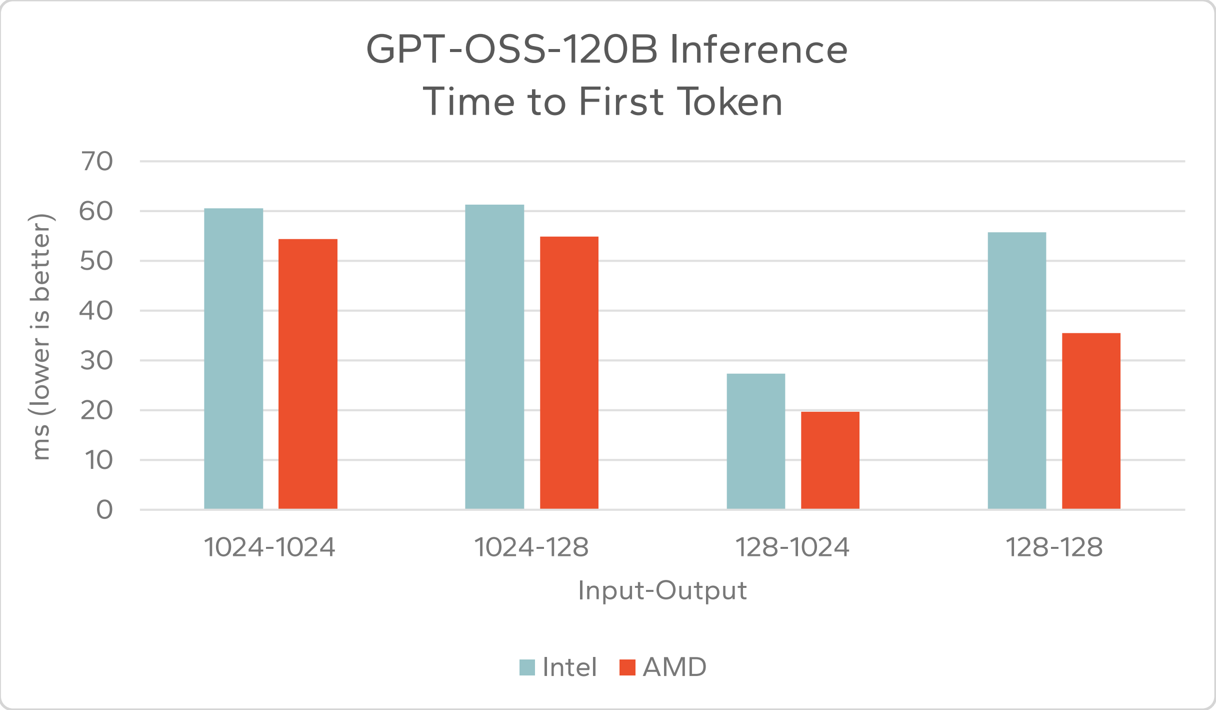


***Figure 7:** GPT-OSS-120B Time to First Token*

Other notable advantages were found when running DeepSeek-R1-FP4, where AMD EPYC achieved 46.54% faster mean time to first token, and Llama-4-Scout-17B-16E-FP4, where AMD EPYC achieved 19.56% faster mean time to first token.

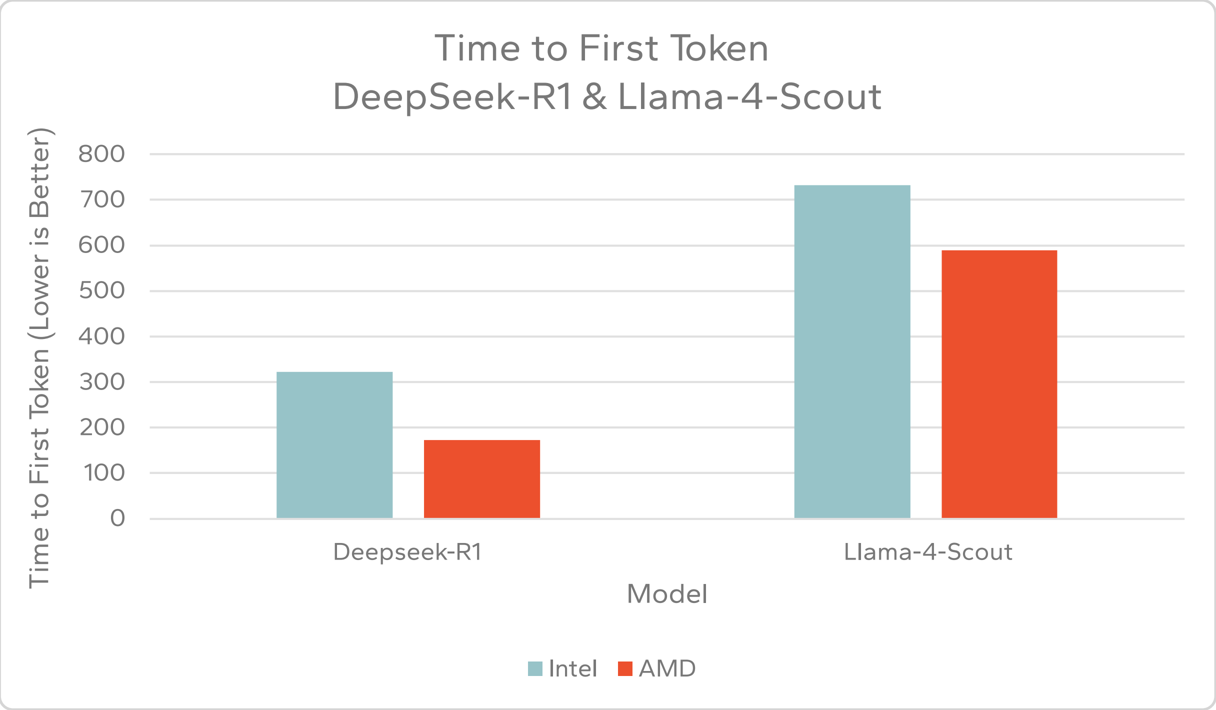


***Figure 8:** DeepSeek-R1-FP4 & Llama-4-Scout-17B-16E-FP4 Time to First Token*

AMD EPYC was found to achieve a faster time to first token for every model tested, across nearly all input / output configurations. In two configurations, Llama-3.3-70B-Instruct with a 128 / 1024 shape and Llama-3.1-8B-Instruct (multi-instance) with 1024 / 128 shape, Intel Xeon achieved a slightly faster time to first token. Both of these results were within 1% of AMD.

Across all models, AMD EPYC achieved a time to first token advantage of 3% or more for at least one test configuration, again showcasing consistent performance advantages.

Inter-Token Latency

Test results also found AMD EPYC achieving lower inter-token latency across the broad range of models tested. As with other metrics, AMD achieved notable advantages when running GPT-OSS-120B, ranging between 10.83% and 11.46% lower latency.

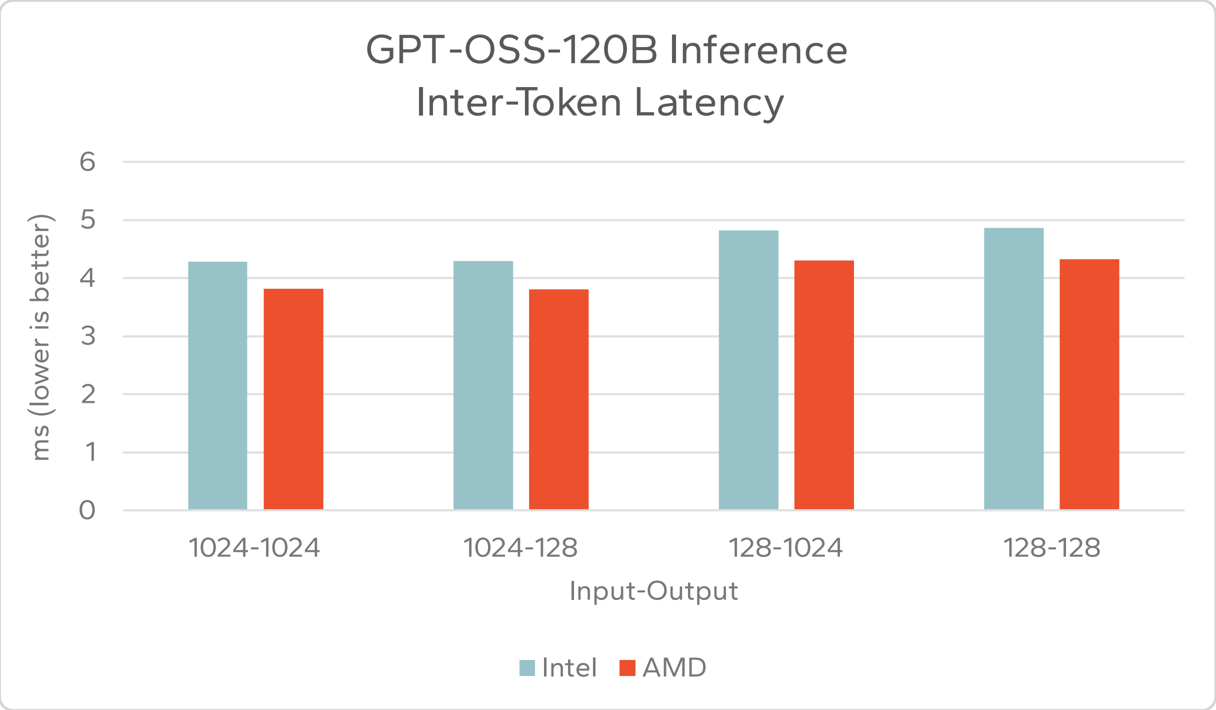


Figure 9: *GPT-OSS-120B Inter-Token Latency*

Other key advantages were found when running Llama-3.1-8B-Instruct and Qwen2.5vl-72B-Instruct. For Llama-3.1-8B-Instruct, which was run in a multi-instance configuration, AMD EPYC achieved a notable latency advantage, between 8.84% and 9.94% lower than Intel Xeon. For Qwen2.5vl-72B-Instruct, AMD EPYC achieved 6.8% lower mean inter-token latency.

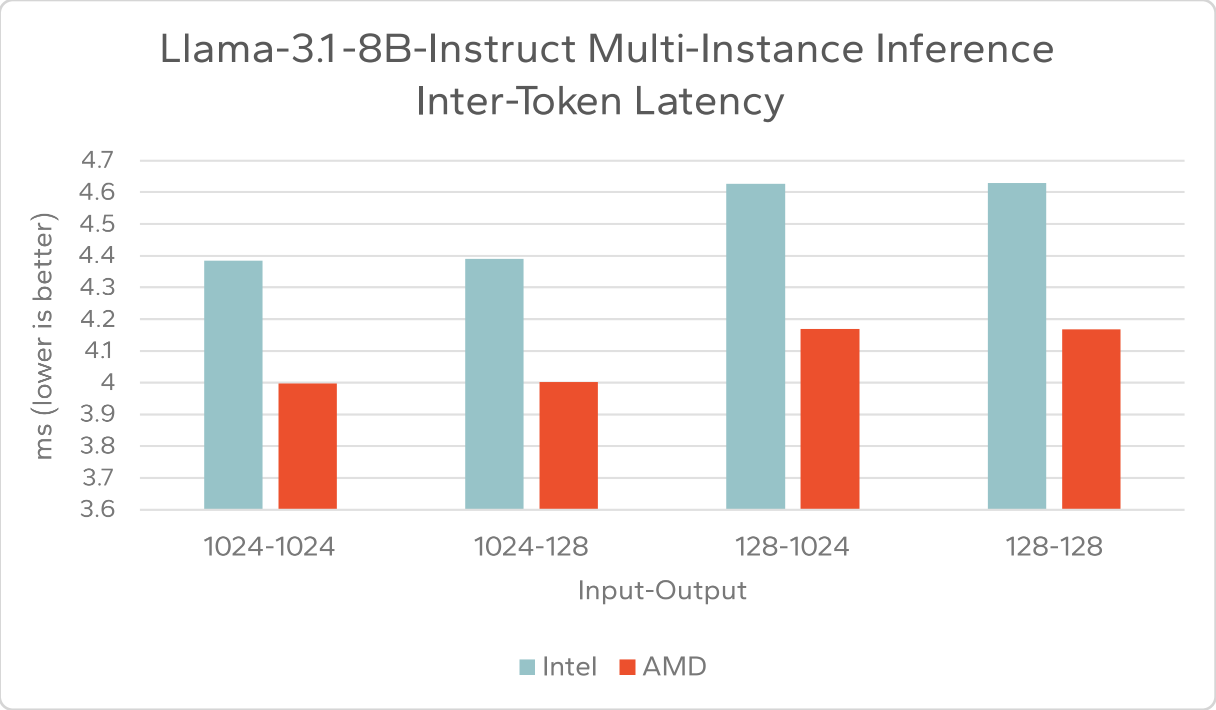


Figure 10: *Llama-3.1-8B-Instruct Inter-Token Latency*

In total, AMD EPYC achieved lower inter-token latency across all models and configurations tested, further demonstrating the consistent performance advantages it can provide during AI inference.

Key Takeaways

Host CPUs Impact AI Performance

By benchmarking AI inference on two nearly-identical systems, this testing isolates the host CPU, highlighting the impact that CPUs have on AI performance. The notable performance deltas found across various models tested demonstrate the importance of CPU selection in AI datacenters.

Across the broad range of models tested, AMD EPYC was found to consistently outperform Intel Xeon. Overall, the AMD EPYC system consistently achieved higher throughput, lower latency, and faster time to first token than the Intel Xeon system, even when running the same models and utilizing the same GPUs.

These consistent performance advantages highlight the importance of CPU architectures that are purpose-built for AI workloads, like 5th Generation AMD EPYC High Frequency Processors. By leveraging the memory and frequency attributes of these processors, IT organizations can remove the CPU bottleneck and enable greater GPU utilization. Although high performance GPUs remain crucial to AI performance, these results demonstrate the additional performance impact of host CPUs, and showcase how AMD EPYC CPUs can be utilized to maximize AI inference performance.

Signal65 Comment *– The performance advantages achieved by AMD in this testing highlight the importance of CPU core speed for AI host CPUs. While the two CPUs tested in this study are competitive platforms, they offer varying specifications that impact performance. The AMD EPYC 9575F tested has a total of 64 cores and operates at a maximum frequency of up to 5 GHz. Comparatively, the Intel Xeon 6960P has more available cores, with 72, but operates at a lower maximum frequency of up to 3.9 GHz. The AI inference tests included in this evaluation involve lightly threaded jobs, which do not benefit from a higher core count. Instead, higher clock frequency and stronger per-core performance become critical for AI host operations, particularly by lowering host-side latency for scheduling, dispatch, and coordination of GPU workloads. The consistent performance advantages seen in this study are likely attributed to AMD's higher maximum frequency and higher core IPC, highlighting the benefits of EPYC processors optimized for latency-sensitive AI inference workloads.*

Efficient CPUs Drive Greater Price-Performance

Beyond pure performance metrics, removing the CPU bottleneck carries notable economic implications. AI datacenters require a significant financial investment due to the high cost of GPUs, in some cases reaching the scale of hundreds of millions to billions of dollars. Given the large financial investment, it is crucial for organizations to maximize the efficiency of their GPUs.

Comparatively, CPU costs typically contribute a relatively small portion when considering the total cost of an AI datacenter. This makes strategic selection of high performance CPUs a practical way for IT organizations to increase their performance without adding significant cost.

The performance benefits delivered by AMD EPYC processors demonstrate how organizations can increase efficiency and maximize the performance per dollar from their large GPU investments. While the performance benefits achieved by AMD EPYC CPUs are notable for a single node, the economic benefits for IT organizations, cloud service providers, and AI labs become clear when considering the scale of a real AI datacenter. For example, in a 1,000 GPU cluster, a 10% increase in throughput roughly equates to the added throughput of an extra 100 GPUs.

The performance results found in this testing illustrate how AMD EPYC can significantly increase cost efficiency. While AMD EPYC achieved up to 14% higher throughput in this testing than Intel Xeon, even many of the smaller throughput advantages found – in the range of 3% to 5% - will ultimately translate to a significant overall boost in GPU processing when applied to large AI datacenters. This enables organizations to achieve greater AI processing, without the additional cost of procuring more GPUs, effectively maximizing price-performance efficiency.

Conclusion

Although AI performance is typically focused on GPUs, host CPUs also have a notable impact during AI inferencing. Careful CPU selection can assist organizations to maximize the value of their GPU infrastructure and avoid critical performance bottlenecks.

This testing demonstrated that when given identical GPU configurations, AMD EPYC processors can achieve notable performance advantages over competitive Intel Xeon processors. AMD EPYC was found to achieve consistent advantages in throughput, time to first token, and inter-token latency, across a wide range of models tested.

When applied at scale, these performance advantages can result in significant efficiency gains and substantial economic value. Although often overlooked, the impact of host CPUs presents a clear opportunity for organizations to strategically enhance their AI datacenters. By leveraging AMD EPYC high frequency processors, organizations can remove CPU bottlenecks from AI inferencing and maximize the efficiency of their AI inference applications.

Key Highlights

AMD EPYC host nodes achieved higher throughput, faster time to first token, and lower inter-token latency than Intel Xeon.

Testing demonstrated consistent performance advantages across 7 distinct AI models.

AMD EPYC High Frequency Processors are purpose built for AI workloads and present a practical approach to maximize the performance and cost-efficiency of large AI datacenters.

Appendix

NVIDIA GenAI-Perf Benchmark Request Throughput Results (Requests / s)

GPT-OSS-120B	Input/Output	AMD	Intel	Difference
	1024 / 1024	1.719693914	1.523031272	12.91%
	1024 / 128	1.721378916	1.520559287	13.21%
	128 / 1024	13.42451792	11.87065156	13.09%
	128 /128	13.07520876	11.4053927	14.64%
Llama-3.3-70B-Instruct	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	0.769697739	0.729078914	5.57%
	1024 / 128	0.769042806	0.729334773	5.44%
	128 / 1024	6.268409525	5.936869487	5.58%
	128 / 128	6.26822513	5.932330609	5.66%
Qwen2.5-Coder-Instruct	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	0.41950556	0.407485528	2.95%
	1024 / 128	0.419429744	0.407519258	2.92%
	128 / 1024	3.342115546	3.251928508	2.77%
	128 / 128	3.348842038	3.254183235	2.91%
Llama-3.1-8B-Instruct (multi)	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	1.741856819	1.590543697	9.51%
	1024 / 128	1.740887565	1.589158766	9.55%
	128 / 1024	15.15516019	13.67168551	10.85%
	128 / 128	15.15764225	13.63522057	11.17%

NVIDIA GenAI-Perf Benchmark Output Throughput Results (Tokens / s)

GPT-OSS-120B	Input/Output	AMD	Intel	Difference
	1024 / 1024	2627.987165	2340.733745	12.27%
	1024 / 128	2632.333197	2335.71406	12.70%
	128 / 1024	2295.340667	2034.059194	12.85%
	128 / 128	2235.317214	1954.339253	14.38%
Llama-3.3-70B-Instruct	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	964.5619572	915.833606	5.32%
	1024 / 128	966.7000392	914.4973328	5.71%
	128 / 1024	964.2063803	913.2511563	5.58%
	128 / 128	964.5932998	912.755266	5.68%
Qwen2.5-Coder-Instruct	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	582.1530505	562.2455259	3.54%
	1024 / 128	585.579213	554.7987811	5.55%
	128 / 1024	529.5611908	516.1145384	2.61%
	128 / 128	530.1311707	514.053402	3.13%
Llama-3.1-8B-Instruct (multi)	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	2472.598519	2257.036372	9.55%
	1024 / 128	2468.822058	2253.514668	9.55%
	128 / 1024	2370.349093	2137.455107	10.90%
	128 / 128	2370.412984	2132.161232	11.17%

NVIDIA GenAI-Perf Benchmark Time to First Token Results (ms)

GPT-OSS-120B	Input/Output	AMD	Intel	Difference
	1024 / 1024	54.42420447	60.66340334	-10.28%
	1024 / 128	54.98249514	61.37648826	-10.42%
	128 / 1024	19.7339165	27.40117686	-27.98%
	128 / 128	35.55424257	55.82619454	-36.31%
Llama-3.3-70B-Instruct	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	195.9628436	203.3995143	-3.66%
	1024 / 128	196.1326684	204.0568279	-3.88%
	128 / 1024	45.78534667	45.66164751	0.27%
	128 / 128	45.34340031	46.23297657	-1.92%
Qwen2.5-Coder-Instruct	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	250.015315	257.3331161	-2.84%
	1024 / 128	250.3922661	262.7677247	-4.71%
	128 / 1024	74.2306356	77.78310387	-4.57%
	128 / 128	74.34762931	78.37192966	-5.13%
Llama-3.1-8B-Instruct (multi)	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	91.97926908	92.12656211	-0.16%
	1024 / 128	93.22186112	92.45364674	0.83%
	128 / 1024	14.46864351	15.27003853	-5.25%
	128 / 128	14.73299851	16.79766674	-12.29%

NVIDIA GenAI-Perf Benchmark Inter-Token Latency Results (ms)

GPT-OSS-120B	Input/Output	AMD	Intel	Difference
	1024 / 1024	3.817032028	4.28886125	-11.00%
	1024 / 128	3.805502369	4.298297381	-11.46%
	128 / 1024	4.306387685	4.829261919	-10.83%
	128 / 128	4.33039898	4.868386271	-11.05%
Llama-3.3-70B-Instruct	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024-1024	10.28053729	10.83085898	-5.08%
	1024-128	10.26087986	10.84299114	-5.37%
	128-1024	10.18190264	10.76807022	-5.44%
	128-128	10.18347914	10.76880094	-5.44%
Qwen2.5-Coder-Instruct	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	17.87992418	18.48762015	-3.29%
	1024 / 128	17.80829702	18.60348079	-4.27%
	128 / 1024	18.8571544	19.34287477	-2.51%
	128 / 128	18.8363982	19.40042295	-2.91%
Llama-3.1-8B-Instruct (multi)	**Input / Output**	**AMD**	**Intel**	**Difference**
	1024 / 1024	3.996613765	4.384091944	-8.84%
	1024 / 128	4.001986222	4.391423467	-8.87%
	128 / 1024	4.170407873	4.626722856	-9.86%
	128 / 128	4.168394398	4.62864997	-9.94%

vLLM Benchmark Results

DeepSeek-R1-FP4

	Request Throughput (Requests / s)	Output Throughput (Tok / s)	Total Token Throughput (Tok / s)	Mean Time to First Token (ms)	Mean Inter-Token Latency (ms)
AMD	0.612523052	627.2236053	705.0140329	172.6326658	25.36191
Intel	0.59221065	606.4237057	681.6344583	322.8898443	26.08977
Difference	3.43%	3.43%	3.43%	-46.54%	-2.79%

Llama-4-Scout-17B-16E-FP4

	Request Throughput (Requests / s)	Output Throughput (Tok / s)	Total Token Throughput (Tok / s)	Mean Time to First Token (ms)	Mean Inter-Token Latency (ms)
AMD	23.49453128	24058.40003	27038.98418	589.4931572	20.70632
Intel	22.55708944	23098.45959	25960.11716	732.8613373	21.44681
Difference	4.16%	4.16%	4.16%	-19.56%	-3.45%

Qwen2.5vl-72B-Instruct

	Request Throughput (Requests / s)	Output Throughput (Tok / s)	Total Token Throughput (Tok / s)	Mean Time to First Token (ms)	Mean Inter-Token Latency (ms)
AMD	8.920464464	7438.49032	8577.522127	382.6273036	13.64299
Intel	8.308753948	6929.700205	7990.624224	414.7002487	14.63839
Difference	7.36%	7.34%	7.34%	-7.73%	-6.80%

Important Information About this Report

CONTRIBUTORS
Mitch Lewis
Performance Analyst | Signal65

PUBLISHER
Ryan Shrout
President and GM | Signal65

INQUIRIES
Contact us if you would like to discuss this report and Signal65 will respond promptly.

CITATIONS
This paper can be cited by accredited press and analysts, but must be cited in-context, displaying author's name, author's title, and "Signal65." Non-press and non-analysts must receive prior written permission by Signal65 for any citations.

DISCLOSURES
Signal65 provides research, analysis, advising, and consulting to many high-tech companies, including those mentioned in this paper. No employees at the firm hold any equity positions with any companies cited in this document.

ABOUT SIGNAL65
Signal65 is an independent research, analysis, and advisory firm, focused on digital innovation and market-disrupting technologies and trends. Every day our analysts, researchers, and advisors help business leaders from around the world anticipate tectonic shifts in their industries and leverage disruptive innovation to either gain or maintain a competitive advantage in their markets.

signal65

signal65

CONTACT INFORMATION

Signal65 I signal65.com