

The Future of AI is Hardware-Agnostic Thanks to Portable Models and Frameworks

Written by:

Peter Rutten

Research Vice President, AI, HPC and Quantum Computing, Core Infrastructure

Madhumitha Sathish

Research Manager, AI and HPC, Core Infrastructure

Published:

July 2026

Table of contents

IDC Opinion 3

Situation Overview..... 3

 How is standardization achieved? 3

 High-level frameworks are not limited to a particular parallel processing software 4

 Why this matters..... 4

 Not all AI demands an AI factory 5

 The interoperability extends to CPUs 5

Considering AMD..... 6

 The AMD interoperability ecosystem 6

 AMD ROCm 6

 Other tools..... 7

 Core components of the stack 7

 Key benefits 7

Challenges and opportunities 8

Conclusion 8

About the IDC analysts 9

IDC Opinion

AI inferencing is a significantly different workload from AI training. Some organizations may have trained their own AI models, but most will have licensed a commercial model or obtained an open-source model. In either case, there is the perception that the AI inferencing environment should look like the original AI training environment in terms of type of infrastructure and the brand of parallel processing software, which, for AI training, is typically a proprietary stack. This notion needs to be corrected.

Today's high-level AI frameworks are not tied to specific hardware or parallel processing software. They enable models to run across diverse platforms, including AMD, Intel, NVIDIA, Google TPUs, and emerging architectures, without major code changes. Interoperability across high-level frameworks, such as PyTorch, abstracts parallel processing software and allows seamless migration between AMD ROCm, Intel OpenVINO, and NVIDIA CUDA. Furthermore, standards for interactions allow easy switching between models and simplify integrating AI models into other applications.

In a similar vein, there is a perception that AI inferencing requires large, costly AI factories. AI factories are useful for rack-scale, real-time training and inferencing, but most organizations just need rightsized, flexible, and distributed AI-ready platforms that combine accelerated computing with existing CPU-intensive applications via microservices and container orchestration.

This white paper will discuss the high levels of portability that are enabling AI inferencing across platforms without concern for the original training platform. The paper will also provide an overview of the work that AMD has done to ease this portability, enabling businesses to avoid vendor lock-in from proprietary approaches.

Situation Overview

How is standardization achieved?

After a period of experimentation with different architecture types, most AI language models today have been standardized on the transformer architecture. This neural network architecture has revolutionized AI by processing data sequences such as sentences in parallel rather than word-by-word, determining the importance of each word relative to other words, pulling in context, and identifying dependencies.

Transformers are at the core of modern generative AI. Organizations have incorporated them into many higher-level frameworks for generating text (e.g., in alphabetical order: Anthropic's Claude, Google's Gemini, Meta's Llama, and OpenAI's GPT), images (e.g., Stable Diffusion and Midjourney), or multimodal outputs (e.g., DALL-E). It's also possible to integrate transformers into frameworks for specific domains of expertise, such as bioinformatics or robotics. Other models that are gaining popularity include state space models (SSMs) and linear-time deep

learning models such as Mamba — both provide greater speed and efficiency. Hugging Face is a popular model hub for downloading, fine-tuning, and deploying tens of thousands of such pretrained transformer models (e.g., BERT and RoBERTa).

High-level frameworks are not limited to a particular parallel processing software

For organizations planning to run an AI model in production, it is crucial to understand that these higher-level frameworks are not tied to any specific architecture (be it AMD, Intel, or NVIDIA) or its related parallel processing software stack (ROCm, or CUDA). Many models can be deployed across the GPUs and software of these vendors, as well as on platforms such as Google TPUs, although this may involve additional steps such as compilation (e.g., via XLA) or the use of framework-specific integrations depending on the target environment.

Support is also emerging for architectures such as Cerebras, SambaNova, and Groq, typically through vendor-provided software stacks, and may involve some degree of adaptation or optimization based on the model and deployment requirements.

This is because most of the processing functionality is built into the frameworks. A data scientist will only need to code to the software when they are performing specific custom work on a transformer model or developing a model that is not based on a transformer but on another AI model type. For the bulk of AI projects, this is uncommon.

Indeed, this high level of interoperability is also true for AI training. Here too, developers rarely need to touch the software. They will be working in frameworks such as PyTorch or TensorFlow or Jax, which are hardware agnostic.

Even if a developer needs to tweak certain architecture structures, they still do not need to touch the underlying kernels. They will be writing in PyTorch, which calls the kernels, not in CUDA, ROCm or OpenVINO. This interoperability is a direct result of industry collaboration to integrate the required functionality into PyTorch.

Why this matters

Cross-architecture interoperability is important to grasp because organizations are often concerned that an AI model will not port to an architecture different from the one it was developed on. If, for example, a developer has trained a model on PyTorch with proprietary software underneath, and they wish to move that model to ROCm or OpenVINO, all they need to do is call the ROCm or OpenVINO library. The libraries support most higher-level frameworks, and many of the most popular ones have been optimized for performance.

As for integration with other applications, this is straightforward because most models have the required hardware optimizations, and applications can be added via standards-based APIs provided in the stack. An alternative is the Model Context Protocol, an open-source framework

that standardizes how AI agents interact with external tools and data sources, such as APIs, databases, and file systems.

In the less common case where a developer has created custom, proprietary kernels that they need to port to ROCm or other parallel processing software, tools exist to convert the kernels. Porting such custom kernels to ROCm is feasible by using ROCm's HIPIFY tools to convert the code to the Heterogeneous-Computing Interface for Portability (HIP), followed by compilation with hipcc. For Open Visual Inference and Neural Network Optimization (OpenVINO), developers can port custom kernels by implementing them as Open Computing Language (OpenCL) kernels and configuring them through OpenVINO's extension mechanisms.

Not all AI demands an AI factory

The current buzz about AI factories is affecting the inferencing conversation. An AI factory is generally defined as dedicated rack-scale infrastructure with very large, dense GPU capacity, high-speed networking, and performance-intensive storage used primarily for AI training.

An AI inferencing factory is similar but designed to deploy and manage AI models in production at scale. It is even more focused on advanced networking to keep latency as low as possible, which is important for real-time applications. AI inferencing factories can handle millions of simultaneous requests for standalone services, such as chatbots, recommendation engines, generative AI services, and agentic AI.

However, most organizations do not need such large-scale inferencing and prefer not to have a siloed infrastructure stack for AI models. They start small and, over time, begin integrating AI into their other general-purpose workloads, for example, email generation, document drafting, code generation, customer service chatbots, language translation, content creation, and predictive maintenance. The practice of rightsizing AI for specific needs in the inferencing stage is common.

Therefore, organizations must integrate AI into their general-purpose environments, a move away from centralized, batch-processing models toward more distributed, real-time AI serving. They achieve this with what many vendors call "AI-ready" platforms that combine accelerated computing — GPUs or field programmable gate arrays (FPGAs) — with existing CPU-intensive applications via microservices, API layers, and container orchestration. An example of this is wrapping AI models in containers and deploying them via microservices.

The interoperability extends to CPUs

Inferencing is not exclusively a GPU task. The role of CPUs is vital for such functions as preprocessing, retrieval, and filtering. However, CPUs can also run the small to midsized AI models, because AI models' interoperability across parallel processing software extends to CPUs when using PyTorch or CPU-optimized model inference libraries. For applications that do not require the lowest latency, CPUs can be a cost-effective way to inference on an AI model.

Unpublished IDC research has shown that organizations put the maximum model size that can run on CPUs at around 80,000 parameters.

Another approach for running AI models on CPUs is converting them to Open Neural Network Exchange (ONNX), which is a standard, open-source format for AI models that enables interoperability between different AI frameworks, such as PyTorch, TensorFlow, and scikit-learn. Developers can train models in frameworks such as PyTorch or TensorFlow and export them to ONNX, enabling deployment across a variety of hardware platforms via compatible ONNX runtimes. For example, many AI models can be deployed on IBM Z systems using ONNX without requiring framework-specific rewrites, although some adaptation or runtime optimization may be needed to achieve optimal performance.

Considering AMD

The AMD interoperability ecosystem

AMD provides a robust hardware and software ecosystem for running AI models – specifically Instinct datacenter GPUs, Radeon client GPUs, EPYC CPUs, and Ryzen client CPUs with the ROCm software stack. The approach is to bridge the gap between proprietary parallel processing software and the AMD open HIP standard.

The key AMD tools and technologies for running AI models trained on proprietary software are:

AMD ROCm

- AMD ROCm software is the open, high-performance AI and HPC software stack designed to power AMD Instinct GPUs across research labs, enterprises, sovereign AI programs, and hyperscale datacenters.
- Built upstream-first and aligned to open standards, ROCm delivers performance at no cost and preserves software investments when porting to AMD Instinct GPUs.
- Teams can move quickly using familiar frameworks and engines, including PyTorch, TensorFlow, JAX, ONNX Runtime, vLLM v1, and SGLang.
- HIP is a C++ runtime API and kernel language that allows developers to write portable code that can run on both AMD and other GPUs.
- With HIPify, ROCm simplifies migration by enabling developers to convert CUDA code to HIP, preserving existing workflows while unlocking portability across AMD GPU environments.
- The hipcc (HIP C++ compiler driver) and amdclang++ (The AMD Clang-based C++ compiler) tools are compilers that take HIP code and compile it to run on AMD hardware.

Other tools

- The SCALE toolkit is a compiler toolkit that enables running proprietary binaries directly on AMD GPUs without source code modification.
- Triton language from OpenAI lets researchers write high-level Python code that compiles to optimized kernels on AMD GPUs, reducing dependency on proprietary implementations.

The AMD enterprise AI reference stack is a comprehensive, open-source full-stack software solution that helps enterprises move AI projects from experimentation to large-scale production in minutes. Built on the ROCm 7 software stack and optimized for AMD Instinct GPUs, the stack focuses on Kubernetes-native orchestration to streamline deployment, optimize compute utilization, and provide vendor independence.

Core components of the stack

The stack is modular, allowing organizations to adopt specific parts or the entire stack.

- **AMD Inference Microservices (AIMs):** These are prebuilt, high-performance containers that include AI models, engines such as vLLM or SGLang, and configurations optimized for AMD hardware. They support open-weight models (e.g., Llama and Mistral) and offer OpenAI-compatible APIs for rapid integration.
- **AMD AI Workbench:** This is a user-friendly, low-code interface that enables data scientists and developers to manage the end-to-end life cycle, including browsing the AIMs catalog, fine-tuning models, and managing development workspaces, such as JupyterLab and VS Code.
- **AMD Resource Manager:** This is a control plane for managing compute clusters and provides real-time monitoring, intelligent job scheduling, quota management, and role-based access control to ensure efficient GPU usage across teams.
- **AMD Solution Blueprints:** These are pre-validated, ready-to-use reference architectures for common enterprise use cases (e.g., RAG, agentic AI, and chatbots) that reduce engineering effort and risk.

Key benefits

- **Rapid deployment:** The stack enables moving from bare metal to production-ready AI services in minutes, according to AMD.
- **Optimized performance:** Hardware-aware tuning for AMD Instinct GPUs ensures maximum throughput and precision (FP8/BF16).
- **Open architecture:** It is based on open-source frameworks, preventing vendor lock-in and improving flexibility.
- **AI-native orchestration:** The stack features intelligent scheduling to minimize GPU idleness and support multitenant environments.

- **Enterprise security:** It includes Secure Boot, encrypted GPU links (Infinity Fabric security features that utilize AES-128 encryption to secure model traffic during distributed training and inference across GPUs) and integrated identity management.

The stack, which includes tools AMD developed following its acquisition of Silo AI, complements ROCm, providing a higher-level abstraction for IT and enterprise application teams.

Challenges and opportunities

AI infrastructure interoperability is a critical cornerstone of the AMD AI strategy. It challenges the proprietary approach of its largest competitors by lowering the threshold for organizations to add or switch to AMD hardware. AMD promotes open standards that deliver an equivalent, albeit more flexible AI approach for businesses without vendor lock-in and with the added benefit that they can scale their AI workloads across various hybrid and multi-cloud environments.

That said, for many organizations the proprietary solutions in the market are perceived as the "easy button" or as part of the "must-have" strategy. This perception will take time to challenge and help businesses overcome. The shift to inferencing should give AMD some strong momentum as preconceptions about the ideal platform for AI inferencing are much less embedded than they were for AI training.

Interoperability allows AMD to promote their solutions without the need to replace hardware from competing vendors. The fact that AMD delivers interoperability has also been attractive to cloud services providers, neoclouds, and hyperscalers, who require a dual-vendor strategy. Finally, interoperability allows for CPUs, GPUs, FPGAs and DPUs from various manufacturers to work together in many — albeit not all — situations.

Conclusion

AI inferencing is fundamentally distinct from AI training, and the prevailing notion that inferencing environments must mirror proprietary training platforms is outdated. Modern high-level AI frameworks, such as PyTorch and TensorFlow, are hardware-agnostic and highly portable, enabling migration across diverse architectures with minimal, if any, code changes in most cases. This interoperability is driven by industry-standard APIs and open frameworks, allowing organizations to avoid vendor lock-in and integrate AI models into existing applications efficiently.

Most organizations do not require large-scale, dedicated AI factories for inferencing; instead, they benefit from flexible, distributed AI-ready environments that combine accelerated computing with CPU-intensive workloads via microservices and container orchestration. The AMD ROCm platform and AMD's enterprise AI reference stack exemplify this approach, offering open-source, modular solutions optimized for AMD hardware, rapid deployment, and robust security. AMD commitment to open standards and interoperability challenges proprietary

approaches, lowering barriers for organizations to adopt or switch hardware and scale AI workloads across hybrid and multi-cloud environments.

Ultimately, the portability and openness of AI frameworks empower businesses to deploy and scale AI inferencing without concern for original training platforms, fostering innovation and operational flexibility.

About the IDC analysts



Peter Rutten

Research Vice President, AI, HPC and Quantum Computing, Core Infrastructure

Peter Rutten is Research Vice President within IDC's enterprise infrastructure global research domain and part of core infrastructure subdomain. In his capacity as IDC's global research lead for the performance-intensive computing (PIC) practice, Peter leads a team of analysts that deliver quantitative and qualitative insights on high-performance computing (HPC), AI, big data and analytics (BDA), and quantum computing (QC) markets and vendors.

[More about Peter Rutten](#)



Madhumitha Sathish

Research Manager, AI and HPC, Core Infrastructure

Madhumitha Sathish is Research Manager within IDC's enterprise infrastructure global research domain and part of core infrastructure subdomain. As part of IDC's coverage of performance-intensive computing (PIC), Madhumitha delivers insights on artificial intelligence and high-performance computing (HPC) infrastructure.

[More about Madhumitha Sathish](#)

Message from the sponsor

AI inferencing does not require proprietary stacks or massive AI factories. Modern AI frameworks enable portability across diverse hardware and software platforms, giving organizations the flexibility to deploy AI where it delivers the most value. By embracing open ecosystems and interoperable technologies, businesses can enable vendor independence, optimize infrastructure investments, and accelerate innovation. AI solutions from AMD help organizations build scalable, flexible AI inferencing environments designed for openness, performance, and long-term choice.

Learn more in the resources below:

<https://www.amd.com/en/solutions/ai.html>

<https://www.amd.com/en/solutions/ai/ai-ready-enterprise.html>



IDC Custom Solutions produced this publication. The opinion, analysis, and research results presented herein are drawn from more detailed research and analysis that IDC independently conducted and published, unless specific vendor sponsorship is noted. IDC Custom Solutions makes IDC content available in a wide range of formats for distribution by various companies.

This IDC material is licensed for [external use](#), and in no way does the use or publication of IDC research indicate IDC's endorsement of the sponsor's or licensee's products or strategies.



IDC Research, Inc.

One Beacon Street, Suite 33100, Boston, MA 02108, USA

T +1 508 872 8200

blogs.idc.com | LinkedIn [@IDC](#) | www.idc.com

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications, and consumer technology markets. With more than 1,300 analysts worldwide, IDC offers global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries. IDC's analysis and insight help IT professionals, business executives, and the investment community make fact-based technology decisions and achieve their key business objectives.

©2026 IDC. Reproduction is forbidden unless authorized. All rights reserved. [CCPA](#)