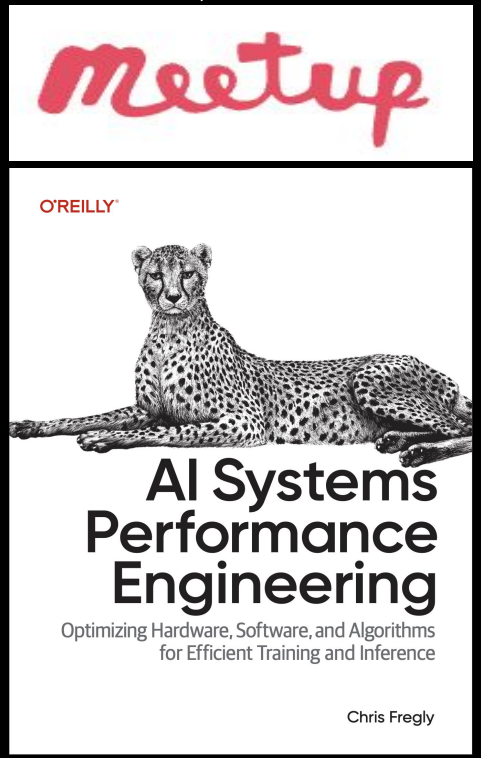
AI Performance Engineering: Apr 21, 2025

Talk #0: Introductions and Meetup Updates
by Chris Fregly and Antje Barth

Talk #1: AI Model Performance Optimizations for Beginners (and More Than Beginners)
by Chaim Rand (https://chaimrand.medium.com/)

Talk #2: NVIDIA GTC 2025 Highlights for AI Systems Performance
by Chris Fregly (https://linkedin.com/in/cfregly/)



Meetup: https://meetup.com/ai-performance-engineering/
YouTube: https://youtube.com/@AIPerformanceEngineering
Book: https://amazon.com/Systems-Performance-Engineering-Optimizing-Algorithms/dp/B0F47689K8/
GitHub: https://github.com/cfregly

AI Systems Performance Engineering O'Reilly Book

Chapter 1: Introduction and AI System Overview

- 1.1 The AI Systems Performance Engineer
- 1.2 Towards 100-Trillion-Parameter Models
- 1.3 NVIDIA's "AI Supercomputer in a Rack"
- 1.4 Mechanical Sympathy: Hardware-Software Co-Design
- 1.5 Measuring "Goodput" Useful Throughput
- 1.6 Book Roadmap and Methodology
- 1.7 Key Takeaways
- 1.8 Conclusion

Chapter 2: AI System Hardware Overview

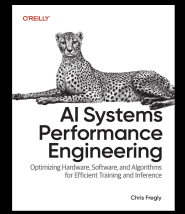
- 2.1 The CPU and GPU "Superchip"
- 2.2 Ultra-Scale Networking Treating Many GPUs as One
- 2.3 Compute Density and Power Requirements
- 2.4 Liquid Cooling vs. Air Cooling
- 2.5 Performance Monitoring and Utilization in Practice
- 2.6 Sharing and Scheduling
- 2.7 ROI of Upgrading Your Hardware
- 2.8 A Glimpse into the Future: NVIDIA's Roadmap
- 2.9 Key Takeaways
- 2.10 Conclusion

Chapter 3: OS, Docker, and Kubernetes Tuning for GPUs

- 3.1 Operating System
- 3.2 GPU Driver and Software Stack
- 3.3 Configuring the CPUs and OS for GPU Environments
- 3.4 GPU Driver and Runtime Settings for Performance
- 3.5 Container Runtime Optimizations for GPUs
- 3.6 Kubernetes for Topology-Aware Container Orchestration and Networking
- 3.7 Key Takeaways
- 3.8 Conclusion

Chapter 4: Distributed Communication and I/O Optimizations

- 4.1. Overlapping Communication and Computation
- 4.2 NVIDIA Magnum IO Optimization Stack
- 4.3 High Speed, Low Overhead Data Transfers with RDMA
- 4.4 NCCL for Distributed, Multi-GPU Communication
- 4.5 NIXL for Accelerated Data Transfer and Inference
- 4.6 Storage I/O Optimizations
- 4.7 Tuning the Data Pipeline
- 4.8 Key Takeaways
- 4.9 Conclusion



AI Systems Performance Engineering O'Reilly Book

Chapter 5: Optimizing Memory Access and Data Movement

- Uncoalesced Global Memory Access
- Lack of Data Reuse and Tiling
- Shared Memory Bank Conflicts
- Misaligned and Scalar Access vs. Vectorized Loads
- Ignoring Read-Only Data Caches
- Using Tensor Memory Accelerator (TMA)
- CPU-GPU Transfer Bottlenecks
- Key Takeaways
- Conclusion

Chapter 6: Tuning Compute Strategy and Parallelism

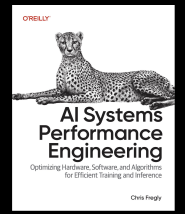
- Using Tensor Cores and Transformer Engines
- Instruction-Level Parallelism (Loop Unrolling)
- Warp Specialization and Producer-Consumer Parallelism
- Warp Branch Divergence
- Eliminating Redundant Computation and Memory Access
- Imbalanced Work Distribution and Persistent Kernels
- CUDA Graphs for Launch Overhead Reduction
- Asynchronous Data Transfers and Memory Prefetching
- Key Takeaways

Chapter 7: Improve Kernel Scheduling, Execution, Orchestration

- Excessive Global Synchronization
- Sequential Host-Driven Execution vs. Streams Overlap
- High Kernel Launch Overhead and Kernel Fusion
- Improve Kernel Batching (CUDA Graphs)
- Increasing GPU Utilization with Multi-Stream Kernels
- Reduce Kernel Overhead with Persistent Kernels
- Reduce Device-Launch Latency (Dynamic Parallelism)
- Balance Work Distribution and Dynamic Load Balancing
- Reduce Throttling by Insufficient CPU Submission

Chapter 8: PyTorch and Triton Optimizations

- Profiling and Bottleneck Identification
- Compiler-Level Optimizations with torch.compile
- Mixed-Precision and Tensor-Core Techniques
- Attention Variants and High-Level PyTorch Attention APIs
- Fusion and Custom Kernels
- Advanced GPU Kernel Techniques
- Memory Management and Data Pipelines
- Distributed Training and MLPerf
- PyTorch XLA Fundamentals and Workloads



AI Systems Performance Engineering O'Reilly Book

Chapter 9: Distributed Training at Ultra-Scale

- Fundamental Challenges and the Need for Distributed Systems
- Core Parallelism Strategies
- Advanced Hardware and Infrastructure
- Overlapping Communication with Computation
- Memory Optimization and Mixed Precision Techniques
- Advanced Scheduling and Kernel-Level Optimizations
- In-Depth Communication and Network Optimizations
- Convergence, Stability, and Distributed Optimization Nuances
- Energy Efficiency, Economic Considerations, and Resource Management
- Enhanced Observability, Instrumentation, and Self-Tuning
- Emerging Trends and Future Directions
- Advanced Operational Considerations
- Key Takeaways
- Conclusion

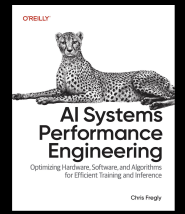
Chapter 10: High-Performance Inference Optimizations

- Core System Architectures
- Distributed Inference Strategies for Large Models
- Low-Level GPU Kernel and Execution Optimizations
- Memory Management and KV Cache Strategies
- Advanced Decoding Techniques
- Batching and Scheduling Strategies
- Precision, Quantization, and Sparsity Techniques
- Profiling and Roofline Analysis
- Application-Level Best Practices
- Key Takeaways
- Conclusion

Chapter 11: AI System Performance Optimization Case Studies

Chapter 12: Future Trends in Ultra-Scale AI Systems Performance Engineering

Chapter 13: 250+ Tips for AI Systems Performance Engineers



NVIDIA GTC 2025 Keynote - March 2025 - Jensen Huang (CEO)

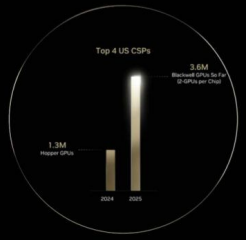
NVIDIA is at a $1 trillion computing inflection point.
Reasoning and Agents - AI computing demand is accelerating rapidly

NVIDIA Blackwell (GPU) and Grace (CPU) + Blackwell (GPU) Systems are in Full Production.
Delivering 40x the performance of Hopper.

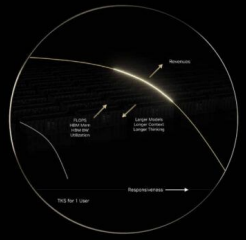
NVIDIA Will Release New Hardware Annually.
Upcoming NVIDIA Vera-Rubin (next gen Grace-Blackwell) in 2026+.

Photonics-based Networking and AI-Optimized Storage are Next Innovation Segments
Advanced networking and storage solutions will improve AI scalability, efficiency and energy consumption

Physical AI for Industrial and Robotics is $50 Trillion Opportunity.
NVIDIA Isaac and Cosmos for AI-powered robotics and automation for manufacturing, logistics, healthcare, etc.

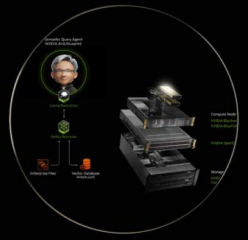


BLACKWELL IN FULL PRODUCTION



BLACKWELL NVL72 AND DYNAMO 40X FOR INFERENCE


VERA RUBIN NVIDIA PHOTONICS AI INFRASTRUCTURE


AI INFRASTRUCTURE FOR $500B ENTERPRISE IT


AI INFRASTRUCTURE FOR ROBOTS

Hardware Updates

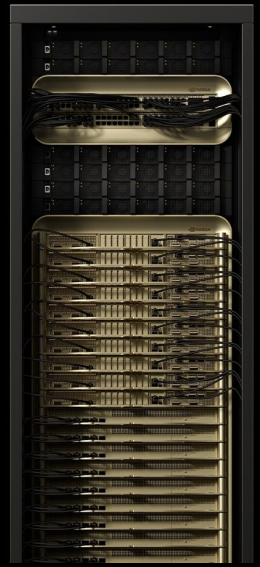
Blackwell GPUs are in Full Production. Blackwell GPUs are shipping at hyperscale, delivering up to 40× the AI throughput of Hopper for training and inference workloads.

Blackwell Ultra GPU (H2 2025). Double Transformer-Engine attention acceleration and deliver 1.5× more AI compute FLOPS with FP4/FP6 support.

GB300 NVL72 Rack-Scale System (H2 2025). 36 Grace CPUs + 72 Blackwell Ultra GPUs with 1.8 TB/s NVLink-C2C boasting up to 30× faster LLM inference.

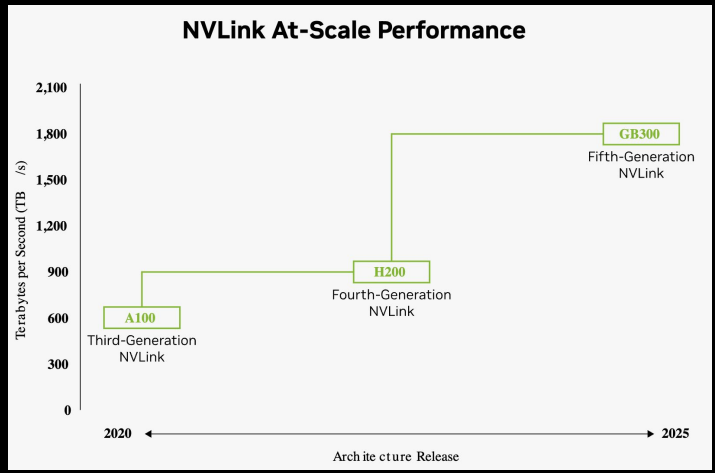
DGX Spark & DGX Station. DGX “Spark” and DGX Station personal desktop AI supercomputers with Grace-Blackwell and Blackwell Ultra for local prototyping

Vera-Rubin and Rubin Ultra SuperChip Previews. Vera (CPU) - Rubin (GPU) SuperHip (2026) and Rubin Ultra (2027) - annual release cadence



Networking Hardware Updates

NVLink-C2C 1.8 TB/s Interconnect. The fifth-gen NVLink-C2C interconnect (1.8 TB/s per link) enables unified CPU-GPU and GPU-GPU memory domains with sub-µs latency.



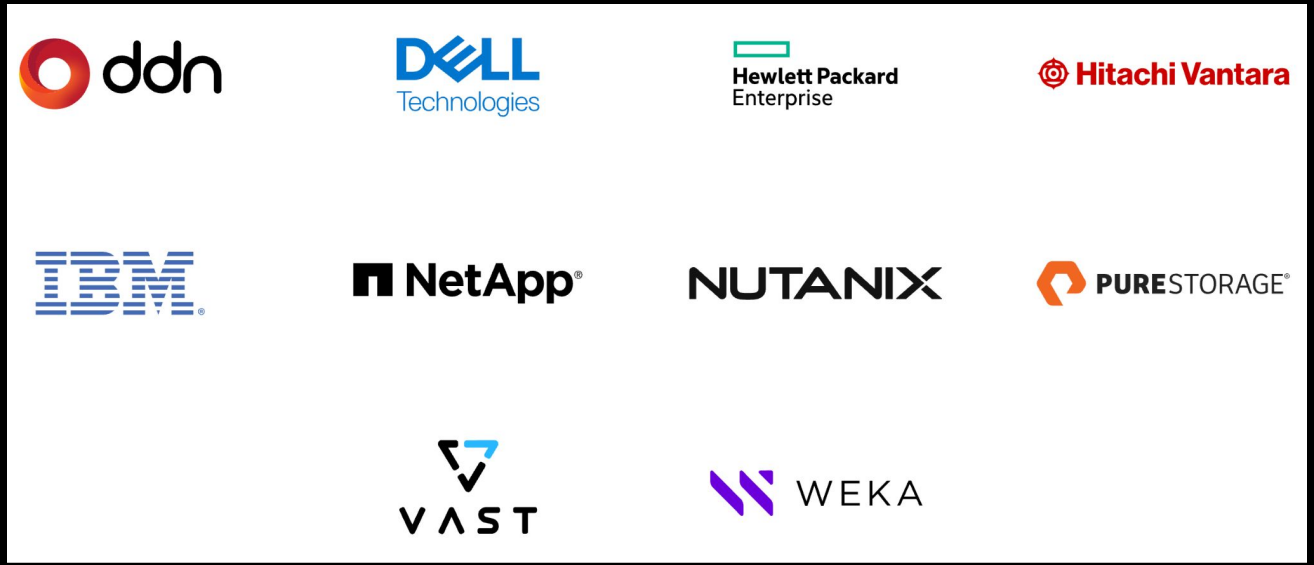
Co-Packaged Optics (CPO)

Silicon photonics switches embedding optics on-chip, 144 ports, 800 Gb/s per port (liquid cooled)

Data Infrastructure and Storage Updates

[Reference Design] NVIDIA AI Data Platform.
Integrates BlueField DPUs, Blackwell GPUs, networking, GPUDirect Storage, and Dynamo (!!) for accelerated inference.

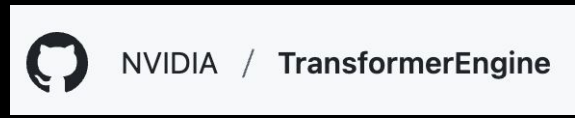
Designed for enterprise storage providers.



CUDA + Python + PyTorch

CUDA Toolkit 12.8.1 Highlights. Updated CUDA Toolkit 12.8.1 with lazy module/kernel loading, updated dynamic parallelism APIs, enhanced CUDA Graph capture/reuse, reduced CPU–GPU sync overhead

Transformer Engine with FlashAttention Primitives. Hits up to 1.2 PFLOPs FP8 attention computations for Blackwell GPUs GitHub.



CUDA Python: New Python-first CUDA Libraries (PyTorch, etc.)

cuTile Library: Python DSL for working with matrix tiles and Advanced Tensor-Core pipelining and optimization techniques

- cuda.core: Pythonic access to CUDA Runtime and other core functionalities
- cuda.bindings: Low-level Python bindings to CUDA C APIs
- cuda.cooperative: A Python package providing CCCL's reusable block-wide and warp-wide *device* primitives for use within Numba CUDA kernels
- cuda.parallel: A Python package for easy access to CCCL's highly efficient and customizable parallel algorithms, like `sort`, `scan`, `reduce`, `transform`, etc, that are callable on the *host*
- numba.cuda: Numba's target for CUDA GPU programming by directly compiling a restricted subset of Python code into CUDA kernels and device functions following the CUDA execution model.

```
TILE_SIZE = wp.constant(256)
TILE_THREADS = 64

@wp.kernel
def compute(a: wp.array2d(dtype=float), b: wp.array2d(dtype=float)):

    # obtain our block index
    i = wp.tid()

    # load a row from global memory
    t = wp.tile_load(a[i], TILE_SIZE)

    # cooperatively compute the sum of the tile elements; s is a single element tile
    s = wp.tile_sum(t)

    # store s in global memory
    wp.tile_store(b[i], s)

N = 10

a_np = np.arange(N).reshape(-1, 1) * np.ones((1, 256), dtype=float)
a = wp.array(a_np, dtype=float)
b = wp.zeros((N,1), dtype=float)

wp.launch_tiled(compute, dim=[a.shape[0]], inputs=[a, b], block_dim=TILE_THREADS)
```

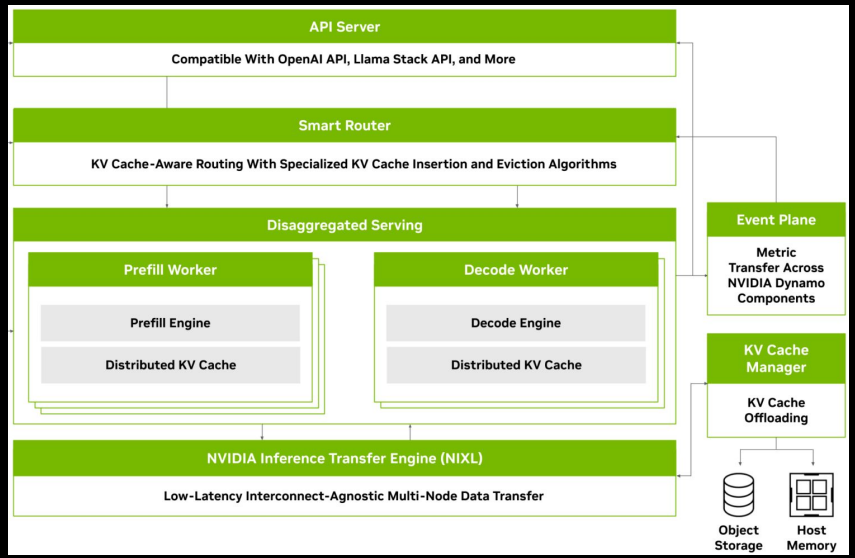
Model Inference Updates - NVIDIA Dynamo!!

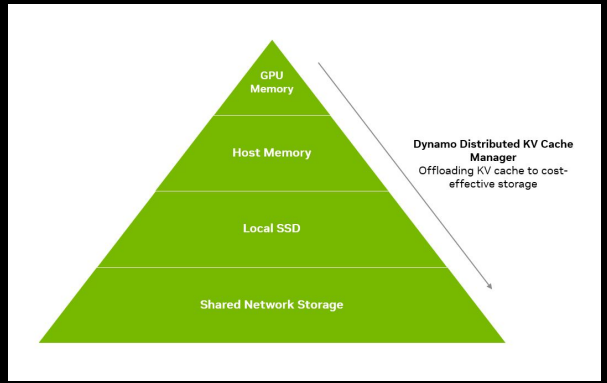
NVIDIA Dynamo Model Inference Server. Successor to Triton Inference Server, orchestrates disaggregated prefill/decode, smart routing, dynamic GPU scheduling, KV-cache offloading

Considered the new "Inference OS"

Backends: vLLM, TensorRT-LLM, SGLang

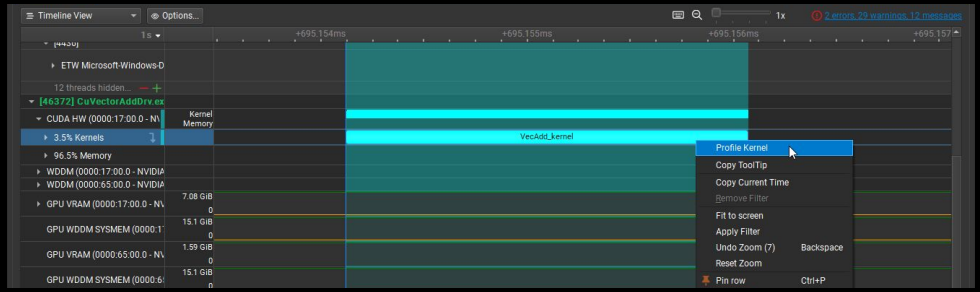
Dynamo + 671B DeepSeek-R1 World Record. Achieved 253 tokens/s per user (>30,000 tokens/s aggregate) on a single 8-GPU Blackwell DGX node using optimized fusion kernels.



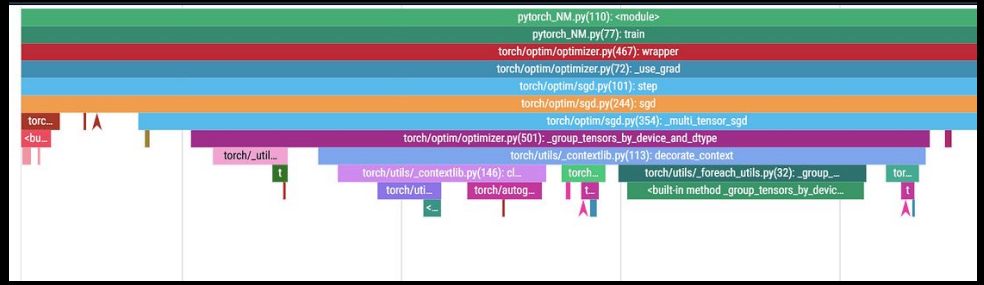


Performance Tooling Updates: Python/PyTorch Call-Stack!!

Nsight Compute. Chart mode for CUDA Graph visualization, zoomable memory timelines, and enhanced kernel profiling.



Nsight Systems. Python call-stack views!!

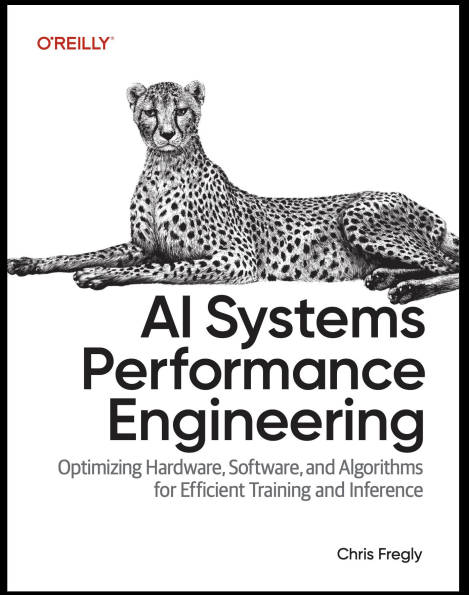


Thanks! AI Performance Engineering: Apr 21, 2025

Talk #0: Introductions and Meetup Updates
by Chris Fregly

Talk #1: AI Performance Optimizations for Beginners (and More Than Beginners!)
by Chaim Rand (https://chaimrand.medium.com/)

Talk #2: NVIDIA GTC 2025 Recap (Performance Focus)
by Chris Fregly (https://linkedin.com/in/cfregly/)



Meetup: meetup.com/ai-performance-engineering/
YouTube: youtube.com/@AIPerformanceEngineering
Book: amazon.com/Systems-Performance-Engineering-Optimizing-Algorithms/dp/B0F47689K8/
GitHub: github.com/cfregly