zett.ai: Redefining AI Infrastructure with Fully Disaggregated, Fully Heterogeneous Architecture

Bay Area, USA

Abstract

The scaling laws of modern Artificial Intelligence have collided with a fundamental hardware barrier: the "Memory Wall." While compute capabilities (FLOPs) have grown exponentially, memory capacity and bandwidth in High Bandwidth Memory (HBM) have failed to keep pace. Consequently, AI infrastructure is currently defined by massive overprovisioning, where expensive GPU clusters are deployed solely to fit model parameters into VRAM, leaving compute logic dangerously underutilized.

We introduce **ZettEngine**, a vertically integrated AI infrastructure platform powered by the **ZettFabric** architecture. By leveraging CXL (Compute Express Link) and a proprietary FPGA-based low-latency bridge, ZettEngine creates a unified, tiered memory pool that scales to 100 TB. This architecture decouples memory from compute, enabling heterogeneous accelerators (NVIDIA H100, RTX 5090, and future ASICs) to access a shared fabric with sub-400ns latency. We demonstrate that ZettEngine achieves up to 10x lower inference costs compared to NVL72 clusters, 95% compute utilization, and offers a seamless software experience through ZettOS, requiring no code changes for standard PyTorch and CUDA workloads.

1 Introduction

The deployment of Large Language Models (LLMs) such as DeepSeek-V3.2, Kimi 2, Qwen 3, and proprietary frontier models has fundamentally shifted the bottleneck of AI infrastructure. Inference workloads are no longer strictly compute-bound; they are capacity-bound. A single instance of a full-precision frontier model can require over 1.5 TB of VRAM.

To serve such a model using traditional architecture, an organization must deploy a cluster of roughly 20 NVIDIA H100 GPUs, not because the FLOPs are required, but simply to aggregate enough HBM to hold the weights. This results in a "Utilization Gap," where expensive tensor cores sit idle while waiting for memory, leading to exorbitant Total Cost of Ownership (TCO) and excessive energy consumption.

Current solutions are insufficient:

1. **NVLink Scaling:** Effective but prohibitively expensive and rigidly locked to a single vendor.
2. **CPU Offloading:** Traditional DRAM offloading via PCIe is too slow for interactive inference (high latency).

zett.ai presents a third path: **Global Fabric Attached Memory (GFAM).** By treating memory as a pooled, fabric-attached resource rather than a device-local constraint, we democratize access to large-model inference.

2 The zett.ai Architecture

zett.ai is the underlying interconnect architecture that powers the ZettEngine appliance. It moves away from the monolithic server model toward a fully disaggregated rack-scale design.

2.1 The Unified Memory Fabric

At the core of zett.ai is the ability to pool memory across the PCIe/CXL bus. Unlike standard NUMA (Non-Uniform Memory Access) configurations, zett.ai utilizes a custom **ZettFabric** to expose expanded memory (up to 100 TB) directly to the GPU's memory addressing space.

This tiered memory architecture allows the "hot" working set (KV-cache, active layers) to reside in HBM, while the bulk of model weights and "cold" context reside in the CXL pool. The ZettFabric ensures that data movement between tiers occurs with deterministic low latency.

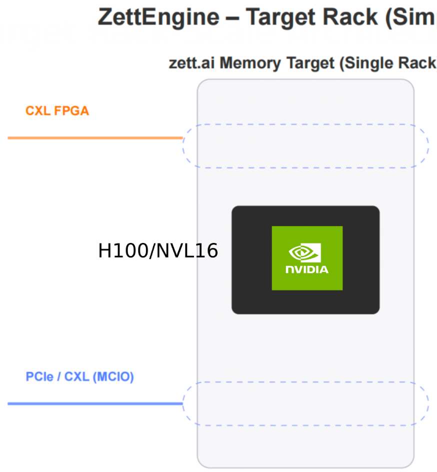
2.2 Heterogeneity by Design

A key tenant of zett.ai is hardware agnosticism. While optimized for NVIDIA architectures (H100, RTX 5090), the protocol layer is designed to support:

- **High-End Data Center GPUs:** NVIDIA H100/H200/Blackwell/Rubin.

- **Prosumer Hardware:** NVIDIA RTX 5090, enabling cost-effective inference for small-to-medium enterprises.
- **Future Silicon:** Native support for RISC-V accelerators and custom ASICs via CXL 3.0 switching.

3 ZettEngine: Hardware Realization


ZettEngine – Target Rack (Simplified)
zett.ai Memory Target (Single Rack)
CXL FPGA
H100/NVL16
NVIDIA
PCIe / CXL (MCIO)

ZettEngine is the physical implementation of the zett.ai architecture, available in varying configurations to suit different scale requirements.

3.1 The FPGA Bridge (Current Gen)

The immediate enabler of our technology is the FPGA-based ZettFabric.

- **Spec:** PCIe/CXL 2.0 bridge utilizing x16 Gen5 links.
- **Performance:** The bridge achieves a round-trip latency of $<$ **400 ns**. This is orders of magnitude faster than NVMe swapping and approaches native DRAM access speeds.
- **SoC:** The promotion and demotion engine inside.
- **Semantics:** It extends host VRAM via CXL.mem semantics, making the pool appear as native, addressable memory to the GPU.

3.2 The RISC-V SoC (Next Gen Roadmap)

To further reduce latency and power, zett.ai is developing the **ZettBridge SoC**, slated for sampling in Q3 2026.

- **Core:** Custom Out-of-Order (OoO) RISC-V core.
- **Switching:** Integrated PCIe/CXL 3.0 switch support.
- **Function:** This SoC will replace the FPGA, offering higher bandwidth switching and smarter memory controller logic for pre-fetching and error isolation.

3.3 Configurations and Pricing

ZettEngine leverages commodity and workstation hardware enhanced by ZettFabric to deliver data-center performance at a fraction of the cost.

Table 1: ZettEngine Product Configurations

Version	Configuration	Price (USD)
ZettEngine	Threadripper PRO 9960X, SMART CXL 512GB, RTX 5090	$10,000
ZettEngine Pro	Intel Xeon 6787P, SMART CXL 512GB	$20,000
ZettEngine Pro Max	Xeon 6787P, RTX 5090, SMART CXL 512GB, ia780i FPGA	$40,000
ZettFabric	Intel VR3 FPGA Dual Port PCIe2CXL Bridge	$30,000

4 Software: Zero Code Change

Hardware innovation fails if it requires rewriting software. The **ZettOS** stack ensures seamless integration with the existing AI ecosystem.

4.1 Unified Addressing Integration

The ZettFabric driver stack creates a unified virtual address space. We provide drop-in replacements for standard allocators:

- **CUDA 13.1 Compatibility:** Applications using ‘cudaMalloc‘ transparently access the extended pool.
- **PyTorch 2.9 Integration:** The PyTorch caching allocator is aware of the memory tiers, automatically placing latency-sensitive tensors in HBM and bulk data in CXL memory.

4.2 Orchestration and Management

- **ZettCLI / SDK:** Provides telemetry, allocation policies, and per-job quota management.
- **Cluster Ready:** Fully compatible with Kubernetes (K8s) and Slurm, allowing ZettEngine nodes to be managed as standard resources in a larger cluster.

5 Key Advantages & Impact

By shifting from a monolithic to a disaggregated architecture, ZettEngine delivers quantifiable economic and operational benefits.

5.1 Cost Efficiency

ZettEngine offers up to **10× lower inference cost** versus an NVL72 cluster. By allowing cheaper compute units (like the RTX 5090 or single H100s) to access massive memory pools, users avoid the "H100 tax" solely for VRAM capacity.

5.2 Utilization and Efficiency

- **95% Utilization:** Disaggregation allows compute resources to be right-sized for the workload, eliminating idle GPUs.
- **5× Energy Efficiency:** CXL pooling reduces the number of powered-on devices required to serve a model, significantly lowering the rack-level power envelope.

5.3 Scalability

The architecture is designed for the future. With CXL 3.0 switching (via the upcoming RISC-V SoC), a single rack can address up to **100 TB** of pooled memory, supporting models orders of magnitude larger than GPT-4 or DeepSeek-V3 without sharding across thousands of GPUs.

6 Conclusion & Availability

ZettEngine is not just hardware; it is a correction to the trajectory of AI infrastructure. It solves the VRAM bottleneck today with FPGA-based bridging and paves the way for a heterogeneous, open future with RISC-V and CXL 3.0.

Availability:

- **Now:** ZettFabric FPGA (PCIe/CXL 2.0) and ZettEngine systems are shipping.
- **Future:** RISC-V SoC sampling begins Q3 2026.

For technical documentation and developer access, visit https://zettai.us/.