



Awide Data Processor

Same GPUs. Four times the AI capacity.

A shared, NVMe-backed memory tier that removes the GPU memory wall - more concurrent users, fewer servers, proven in benchmarks.



AI ACCELERATION



INFRASTRUCTURE



TECHNOLOGY

AI serving is limited by memory, not compute



The memory wall

HBM tops out at 80-192 GB - too little for long-context KV cache. GPUs sit full on memory while compute is idle.



Recompute storms

When context is evicted, the GPU re-processes the whole history. Latency jumps from milliseconds to seconds.



Expensive scaling

2× larger models demand 4-8× more infrastructure - adding GPU servers just to hold memory.

An NVMe KV-cache tier that rivals RAM



Transparent tier below HBM

Slots in beneath GPU memory - no changes to software or GPU drivers.

GPU · HBM

scarce, fast - kept free for compute



NVMe rivals RAM offload

Others offload KV cache to system RAM; ADP matches that on 3D NAND - far cheaper and far larger in capacity.

Awide Data Processor

NVMe KV-cache tier - rivals RAM offload



Context fetched, not recomputed

KV context is stored in ADP and served on demand - no recompute, no latency spikes.

Bulk storage

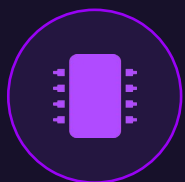
cold capacity

The building blocks of ADP



AcceKV

Hardware key-value engine for instant access to context elements - built for RAG and agentic workloads.



Custom ASIC

Awide-designed processor runs KV management, dedup and compression in hardware, off the host CPU.



ADP-RAIDplus

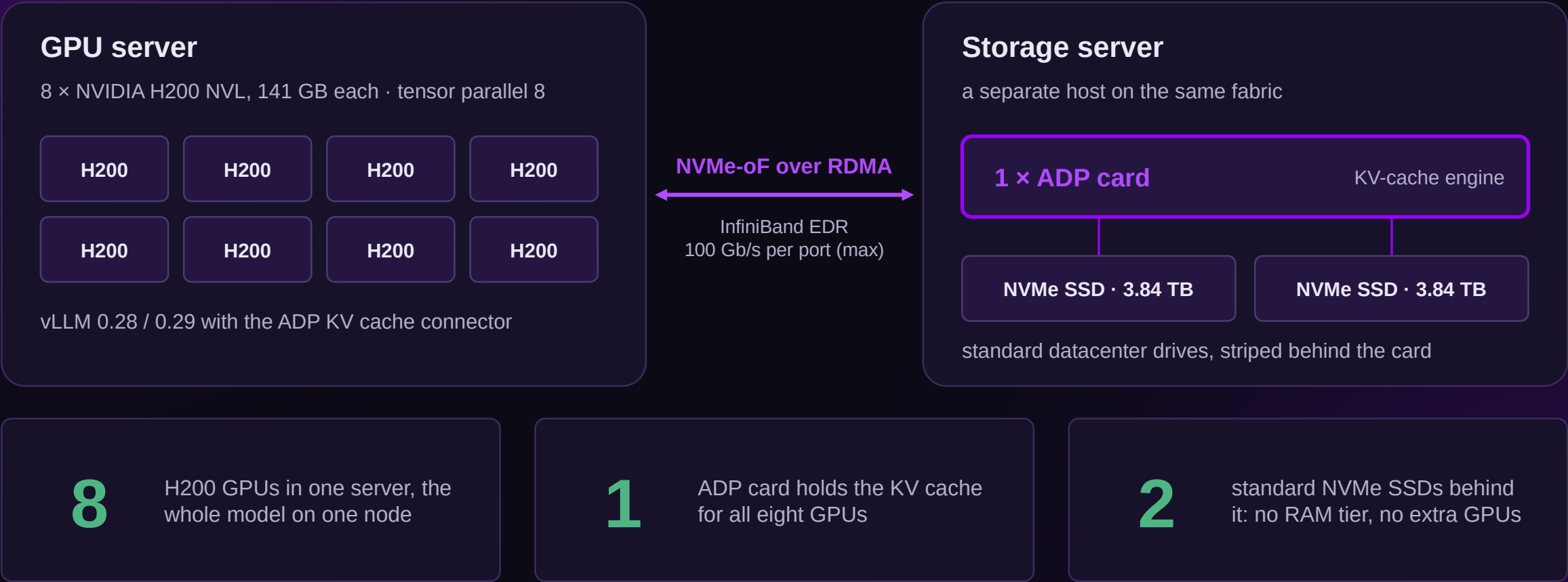
AI-optimized redundancy that keeps data available and fast through multiple drive failures.



NVMe-oF fabric

Standard NVMe over Fabrics pools memory across servers into one shared high-speed tier.

Eight H200s, one ADP card, two NVMe drives



Coding agents on 8×H200, with ADP and without

≥8×

more agents at median ≥20 tok/s
(50 ms/token): 32 vs 4

5.4×

higher total output throughput at
32 agents (190 vs 35 tok/s)

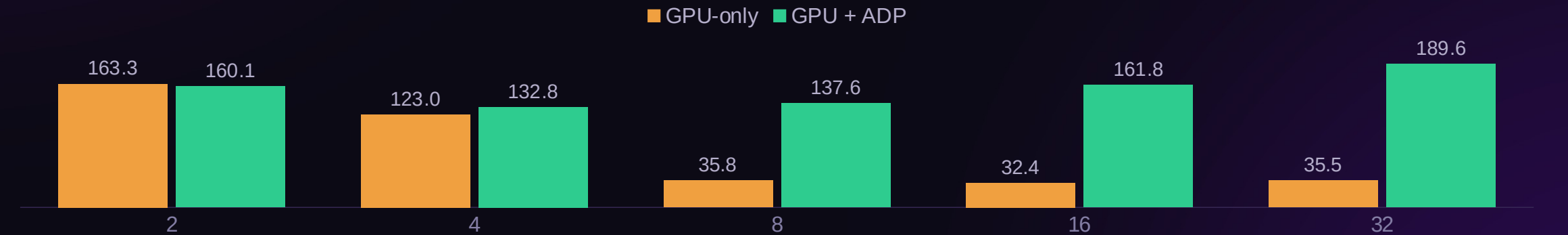
7×

faster generation per agent at 32
agents (25.4 vs 3.6 tok/s)

99.4%

of a returning prompt served from
cache (GPU-only: 0%)

Total output tokens/s across all agents, by concurrent agents · higher is better



Total KV-cache size: approximately 480 GB across all agents · 1 ADP card with 2 × 3.84 TB NVMe SSDs

GLM-5.3 FP8 · 8× H200 · vLLM 0.28 · 5-73 sessions · output tokens / full workload time, including prefill and waits

Prefill becomes a read

22×

faster first token on a 196K-token prompt (2.12 s vs 47.3 s)

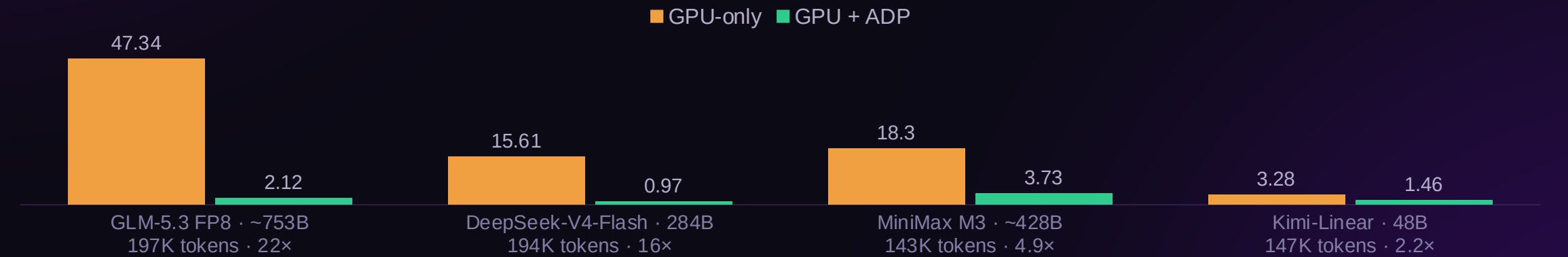
93K tok/s

restored from ADP, against ~4.2K tok/s of GPU prefill

0

mismatches in 1.2 M restored pages, checked byte for byte

Time to first token after a vLLM restart, seconds · GPU-only recomputes, ADP restores · lower is better



8× H200 NVL, TP=8 · 1 ADP card + 2 NVMe SSDs over NVMe-oF/RDMA · vLLM 0.28 (GLM, MiniMax) / 0.29 (DeepSeek, Kimi) · restored after a vLLM restart, from ADP only

B = billion total parameters. Prefill cost also depends on active parameters, attention architecture and prompt length.

The same speed without 400 GiB of RAM

DRAM TIER		ADP	
LMCache with 400 GiB of DRAM		ADP connector, no RAM tier	
Host RAM for the KV cache	412 GiB	Host RAM for the KV cache	10 GiB
Restore after a vLLM restart	2.91 s	Restore after a vLLM restart	2.26 s
Generation per agent, 32 agents	27.2 tok/s	Generation per agent, 32 agents	27.0 tok/s
Wall time, 32 agents	915 s	Wall time, 32 agents	661 s
Answers frozen > 20 s, of 380	27	Answers frozen > 20 s, of 380	0

Total KV-cache size: approximately 480 GB · LMCache adds a 400 GiB DRAM L1 tier

GLM-5.3 FP8 · 8× H200 · vLLM 0.28 · LMCache 0.5.5 MP · warm restore: RAM 1.68 s, ADP 1.86 s (10% faster)

One 8×H200 server, with ADP and without

4×

concurrent clients at ≤50 ms/token
64 with ADP vs 16 GPU-only

12.27

sustainable req/s vs 3.47 GPU-only

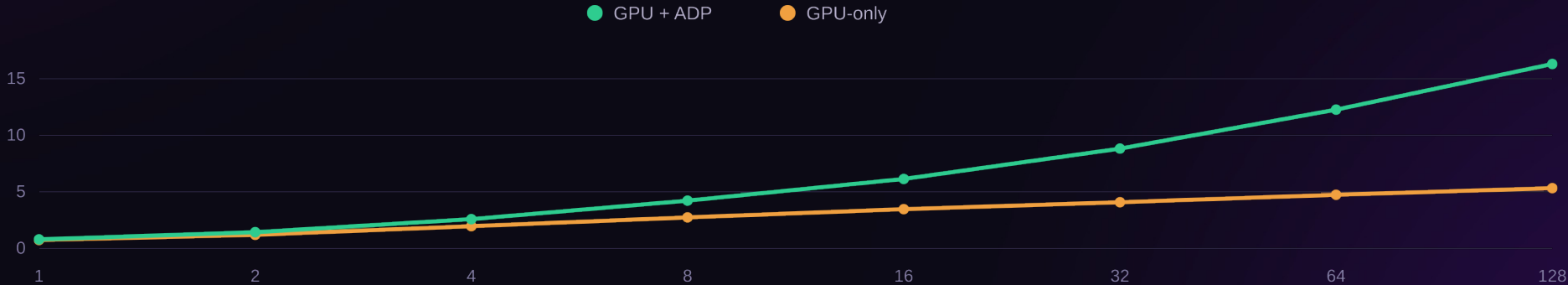
~82 ms

time-to-first-token vs ~246 ms

+206%

throughput at peak (128 clients)

Throughput by concurrent clients · completed requests per second



DeepSeek-V3-0324 · 8 × H200 141 GB · 3,200 conversations · total KV-cache pool: 966.86 GiB
GPU-only memory estimate: 13 H200s for the entire ~967 GiB KV pool plus model weights

One H100, long dialogs

+68%

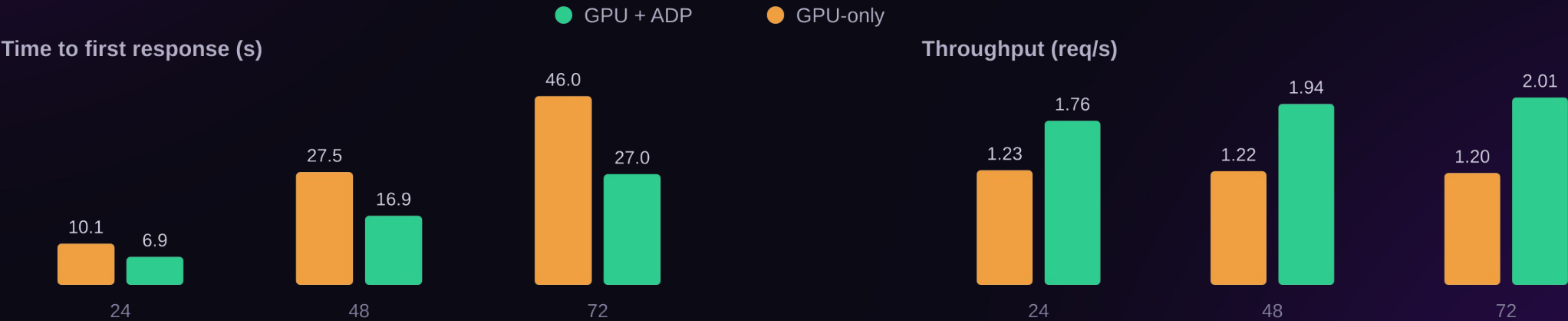
more req/s at 72 concurrent dialogs

-41%

faster first response at 72 dialogs

2.01

req/s vs 1.20 GPU-only at 72 dialogs



Single H100 80GB · Qwen2.5-32B · 6-turn dialogs · X-axis = concurrent dialogs (GPU-only vs GPU + ADP) · turn 6 at 72: 40.0 s vs 68.4 s.

Note: load was capped at 72 concurrent dialogs - ADP's lead kept widening, so there is clear headroom beyond this point.

Each result maps to the mechanism



TTFT → prefill avoided

Recomputing a 196K-token prompt takes 47 s of prefill; ADP reads it back from NVMe in 2.1 s, so responses stay fast.



TPOT → memory pressure relieved

Holding KV cache off-VRAM keeps more sequences resident - 4-8× more users under the 50 ms SLA.



Throughput → freed capacity

Freed VRAM keeps the GPU compute-utilized, lifting sustained throughput at every concurrency level.



The offload really happened

196,480 of 196,531 prompt tokens served back from ADP after a vLLM restart; 1.2 M pages checked byte for byte, 0 mismatches.

Faster across the AI stack - no new GPUs

8×

faster LLM inference on long-context workloads

4×

faster VectorDB queries for RAG & search

0

additional GPUs to reach these gains



Acceleration is workload-dependent

Gains scale with context length and reuse: long, multi-turn, context-heavy traffic benefits most; short stateless requests see little change.

The same workload on half the hardware

GPU-ONLY		GPU + ADP	
Scale by brute force		Consolidate with ADP	
GPUs	16 × H200	GPUs	8 × H200
Rack space	14U	Rack space	7U
Power	20-24 kW	Power	10-12 kW
Infrastructure	\$1,000,000	Infrastructure	\$500,000

Lower cost for the same result

-\$500K

server-equivalent CapEx avoided
per workload

≈50%

less power and cooling under load

2×

higher workload density per rack

3-4×

lower 3-year total cost of
ownership



Turn GPU expansion into a memory-tier upgrade

Fewer GPU servers for the same inference workload means lower capital, energy and rack costs - and less exposure to GPU supply constraints.

SUSTAINABILITY

Efficient Computing for a Net-Zero Future

The Awide Data Processor moves the KV cache out of GPU memory, so LLM inference serves the same users on half the servers. Less electricity consumed, less hardware manufactured, fewer CO2 emissions per token generated.

Energy efficiency is not a side effect of our technology. It is the point: acceleration and sustainability are the same engineering problem, solved once.

20-24 → 10-12 kW

power per workload

16 → 8

GPUs per workload

14U → 7U

rack space

POWER · SERVERS · GPUs



AT LEAST

-40%

less CO2 emissions for the same workload

Reference LLM inference deployment: two GPU servers drawing 20-24 kW consolidated into one with ADP at 10-12 kW for the same workload.

[See the measurements →](#)

Less GPU time for the same work

−81%

less aggregate GPU time per 1,000 output tokens: 42 vs 226 seconds across 8 GPUs

−95%

GPU time to bring back a 196K-token context: a 2.1 s read instead of 47 s of prefill

−97%

host RAM for the KV cache against a DRAM tier: 10 GiB instead of 412 GiB per node



More output per server, fewer servers for a fixed load

The same 8-GPU server delivers 190 vs 35 output tokens/s (5.35×) on this workload. This supports consolidating a fixed load onto fewer servers. GPU count was unchanged in the test. Wall power was not metered.

Measured on one 8× H200 NVL server with one ADP card and two standard NVMe SSDs behind it, over NVMe-oF/RDMA · GLM-5.3 FP8 · vLLM 0.28

Supporting the EU Net-Zero Industry Act

Through Energy-Efficient AI Infrastructure

Awide supports the objectives of the EU Net-Zero Industry Act by helping enterprises reduce the energy footprint of AI and data-intensive infrastructure.

By offloading and accelerating demanding workloads at the hardware level, ADP lowers electricity consumption, reduces cooling requirements, decreases rack-space and power-density pressure, and cuts CO2 emissions across large-scale AI and storage deployments.

[Read about the EU Net-Zero Industry Act](#) 

[Our sustainability commitment](#) →



Lower electricity use

20-24 kW down to 10-12 kW for the same workload



Reduced cooling load

Every watt not consumed never has to be cooled away



Less rack pressure

14U collapses to 7U - more performance per U



Cut CO2 emissions

At least 40% less CO2, and 8 GPUs never built

Drops into the stack you already run



GPU-agnostic

Works with NVIDIA and AMD accelerators - no vendor lock-in.



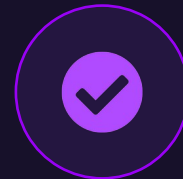
Built for vLLM

Verified on vLLM 0.26-0.29 with GLM-5.x, MiniMax M3, Kimi, DeepSeek-V4-Flash and Qwen 3.8 - zero application changes.



Shared memory pool

NVMe-oF distributes ADP memory across multiple GPU servers.



Transparent layer

Integrates at the storage layer - apps gain speed with no code changes.

ADP-RAIDplus keeps AI online



No downtime on failure

Automatic failover with no service stop - AI apps keep running through hardware faults.



Power-loss protection

Capacitors and atomic-write algorithms guarantee data integrity on sudden power loss.



Predictive monitoring

Drive health is tracked continuously, flagging failures before they happen.



AI-tuned redundancy

Distributed redundancy designed for read/write-intensive AI workloads.

Conditions and definitions for validation

Test stack

GPU	8 × NVIDIA H200 NVL · TP=8
Cache tier	1 ADP card · 2 × 3.84 TB NVMe
Fabric	NVMe-oF / RDMA · EDR 100 Gb/s max
Engine	vLLM 0.28.0 · 0.29.0
Main model	GLM-5.3 FP8
Workload	agent traces · 1-32 agents
Earlier run	vLLM 0.12 · DeepSeek-V3 · Qwen2.5

Metric definitions

TTFT time to first token - dominated by prefill; sensitive to recompute on eviction.

TPOT time per output token. Published agent comparison: median ≤50 ms (≥20 tok/s). Harness SLO: 15 tok/s.

Method same GPUs, model and workload, ADP on/off. Total throughput = all output tokens / full replay duration.

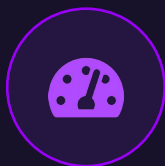
Boundary short stateless requests see little gain - at 1-4 agents both modes finish level - stated openly, not cherry-picked.

Scale models, not server farms.



Grow memory, not GPU count

Add ADP capacity instead of whole servers as models and context grow.



Peak performance per dollar

Maximum inference throughput without exponential budget growth.



Deploy on existing hardware

Transparent integration - working acceleration from day one.