Reimagining Storage Architecture: Confronting Bottlenecks to Scaling

Alan Bumgarner, Director Strategic Planning
Jonathan Hinkle, Sr Director of Storage Pathfinding

MSST 2026
The International Conference on Massive Storage Systems and Technology

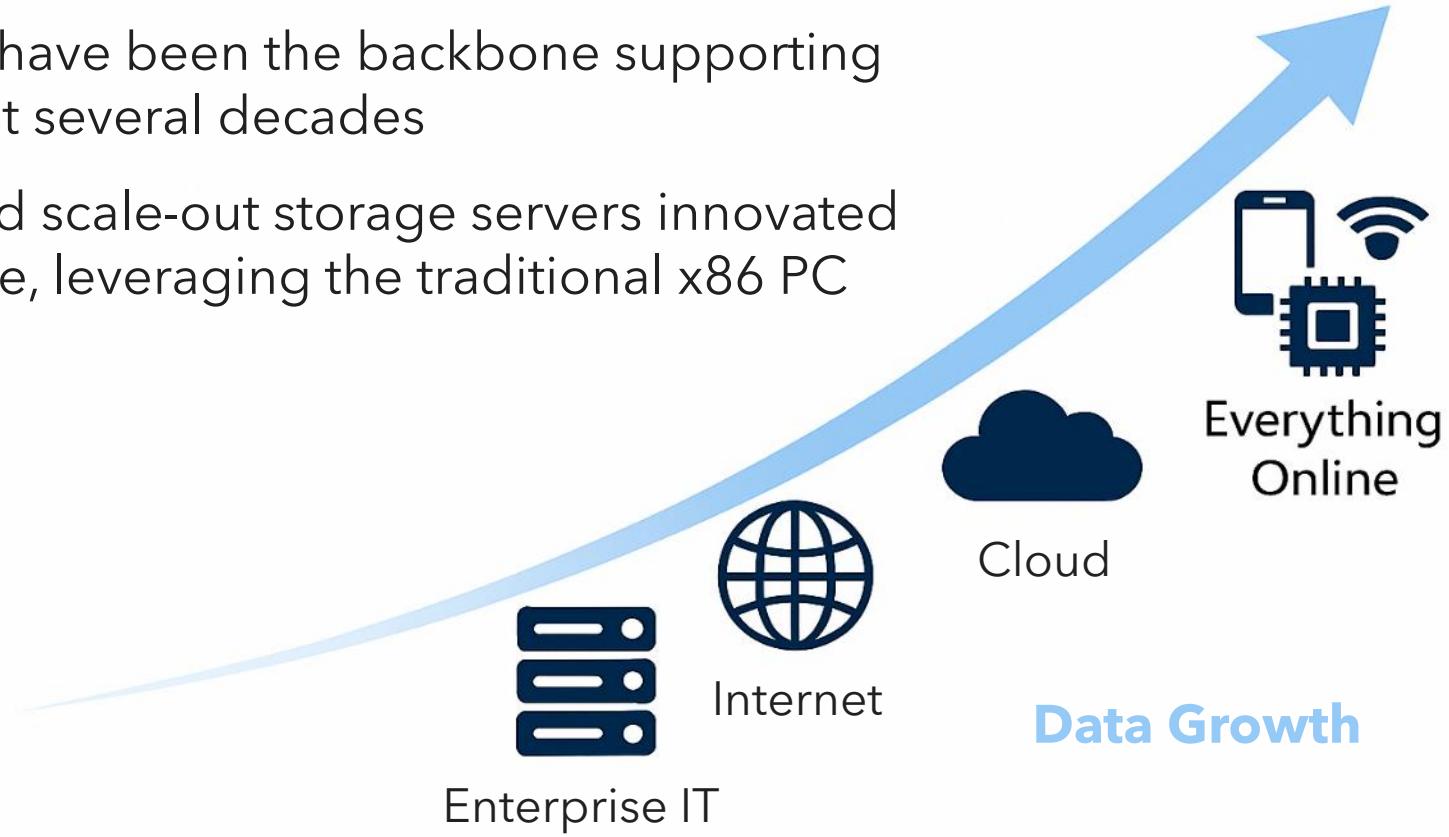
June 2, 2026

The Familiar Progression

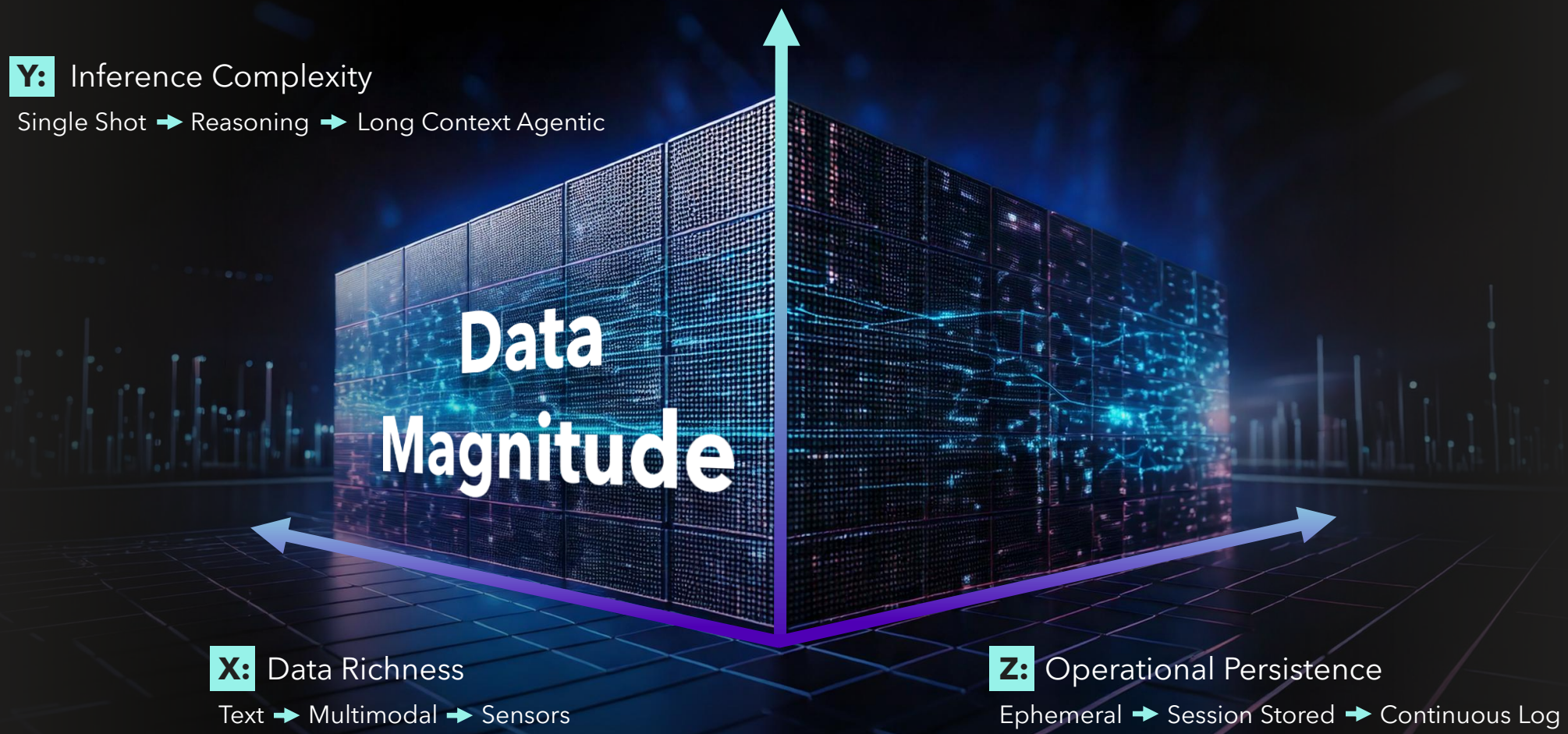
Foundation of the Digital Era: Systems and storage supporting significant data growth

- Modern storage architectures have been the backbone supporting waves of data growth over past several decades
- Dedicated storage systems and scale-out storage servers innovated and rose to meet this challenge, leveraging the traditional x86 PC architecture and supply chain

Ruben de Rijcke, Wikimedia Commons, CC BY-SA 3.0



AI data growth is not linear, it's volumetric.



*AI generated image

AI data growth is not linear – **it's volumetric**

Per-inference-cycle working set *compounds multiplicatively across three axes.*

X **Data Richness**

Text → Multimodal → Sensor streams

Y **Inference Complexity**

Single-shot → Reasoning → Long-context

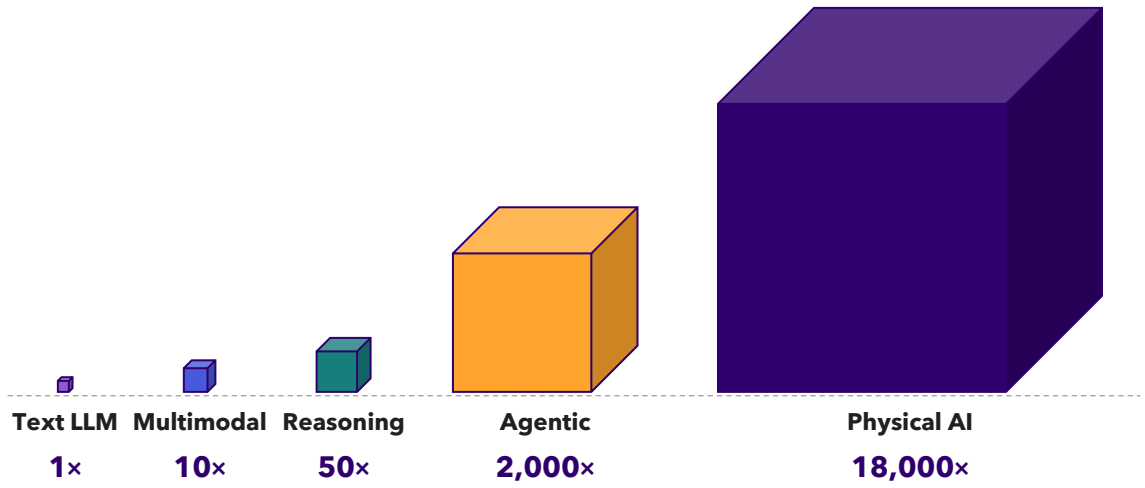
Z **Operational Persistence**

Ephemeral → Session-stored → Continuous log

Row by row – per-inference-cycle multipliers

CATEGORY	X	×	Y	×	Z	=	×/CYCLE	~ SIZE
Text LLM	1×	×	1×	×	1×	=	1×	~GBs
Multimodal	10×	×	1×	×	1×	=	10×	~10s GB
Reasoning	10×	×	5×	×	1×	=	50×	~100s GB
Agentic	10×	×	20×	×	10×	=	2,000×	~TBs
Physical AI	30×	×	20×	×	30×	=	18,000×	~10s TB

Multipliers as volume – cube ∝per-inference-cycle working-set



Sources: *X: Llama 2→4 corpus · image/video input sizes. Y: VAST 61 GB/128K session · Samsung 3 GB baseline. Z: Dell 20–30× agentic · VAST 48 PB/10K users. Byte estimates use ~2 GB per-cycle working set baseline (KV cache + activations + I/O) for a single-shot text query; multipliers applied linearly.*

Per-inference-cycle working-set magnitude. *Aggregate deployment footprint = per-cycle × cycle volume × active users.*

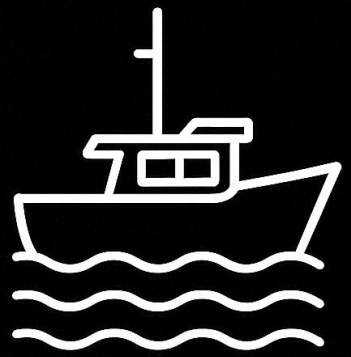
Agentic workloads (2,000×) are forcing the rebuild today. Physical AI (18,000×) compounds it another order of magnitude.

One-Prompt Inference Context

128k tokens

↓

~**61GB** KV cache[1]

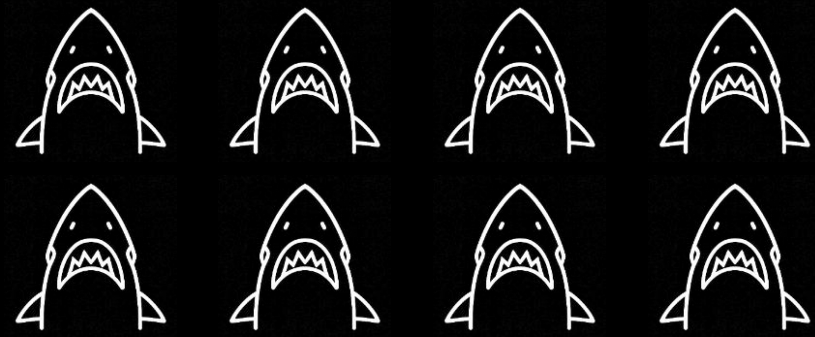


NVIDIA Vera Rubin NVL72

20.7TB HBM4

+

54TB LPDDR5X



Inference Context at Scale

100k users * 64k tokens * 15 conversations

↓

~**45PB** KV cache[1]

1. VAST Data math based on Llama 3.1-405B with "standard parameters" – see https://www.vastdata.com/blog/nvidia-dynamo-vast-scalable-optimized-inference

Storage is a day-zero design consideration

One thread, six observations, one consequence.

1. **User prompt ≠ compiled prompt**
   8 tokens become ~42K once the orchestrator amplifies the prompt.

2. **Storage gates $T_{assembly} + T_{prefill}$**
   Two of the five TTFT components live on storage, not compute.

3. **RAG is a low-QD random-read storm**
   Many concurrent jobs at QD≈1-4 — not peak IOPS at QD=128.

4. **Cache reuse turns prefill from O(N²) into O(N)**
   Stable prefix is reusable across turns; volatile suffix is not.

5. **GB per session → TB per fleet**
   Agentic fan-out under concurrency turns 13 GB into 1.68 TB of working state.

6. **Tier to the workload, not the datasheet**
   RAG and KV cache have opposite IO shapes; spec accordingly.

Storage is a day-zero design consideration

— alongside compute, memory, and network.
Plan the working set first; size the GPUs around it.

Read the paper

Anatomy of a Prompt

Storage in long-context inference and RAG

What is **Unique about Today's Storage** and AI?

Memory / storage hierarchy – A view

G1 GPU HBM

G2 System DRAM

G3 In-System Performance NVMe

G3.5 In-Rack Scaled NVMe Capacity

G4 Network Attached NVMe and some HDD

How Solidigm and NVIDIA are Reinventing Storage for the AI Factory Era

Anatomy of Storage in an AI Data Center

Hot / Direct-Attached (DAS) / G3 Tier

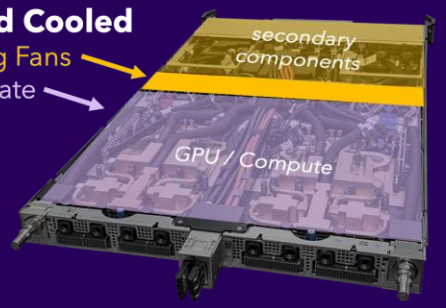
e.g. NVIDIA GB300 NVL72 System

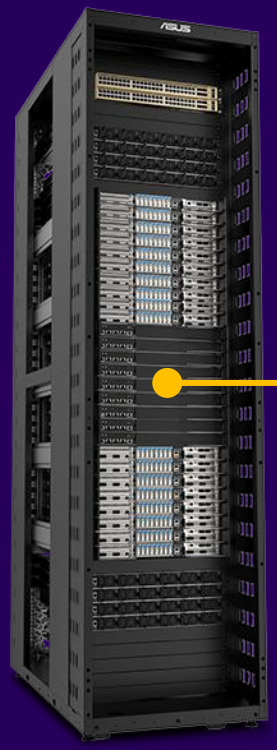
100% Fanless

Cold Plate
Liquid Cooling

Hybrid Cooled

Cooling Fans
Cold Plate





18x compute trays
8x E1.S SSD / tray
144x E1.S / rack

~1.1PB / rack

Warm / Network-Attached (NAS) / G4 Tier

18x 2U storage servers
24x U.2/2.5" / server
432x SSDs / rack

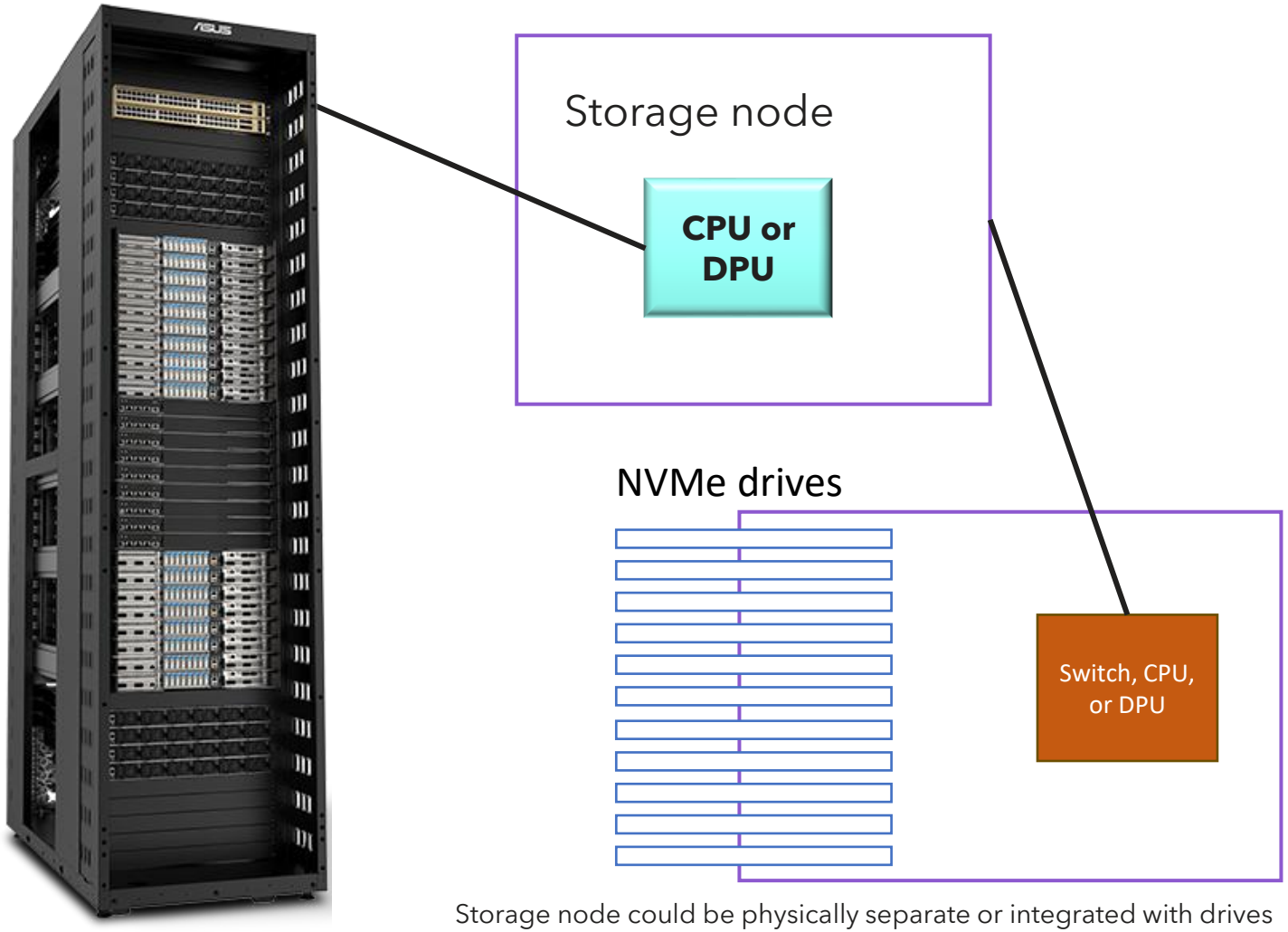
122.88TB / eSSD

~53PB / rack

Liquid Cooling Rapidly Growing Mandatory

Driving 122TB for Improved Efficiency

One Significant Challenge to Serve this Growth

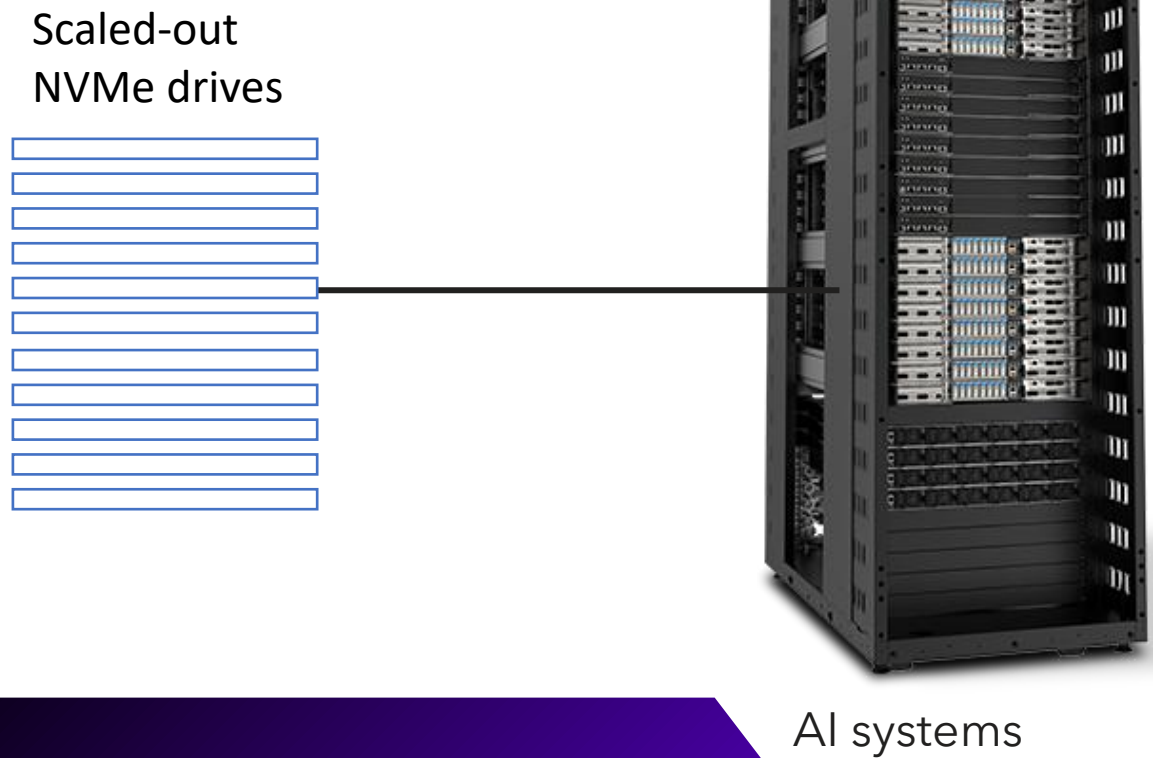


Storage node could be physically separate or integrated with drives

- Node-level scaling can be significant challenge to volumetric data growth in new G3.5 tier
- AI systems demanding PBs of data
- Need scaling of storage capacity outside of AI system node
- Bottleneck can soon become storage node CPU or DPU instead of direct access to drives
- Result is insufficient performance or capacity, higher power and higher cost

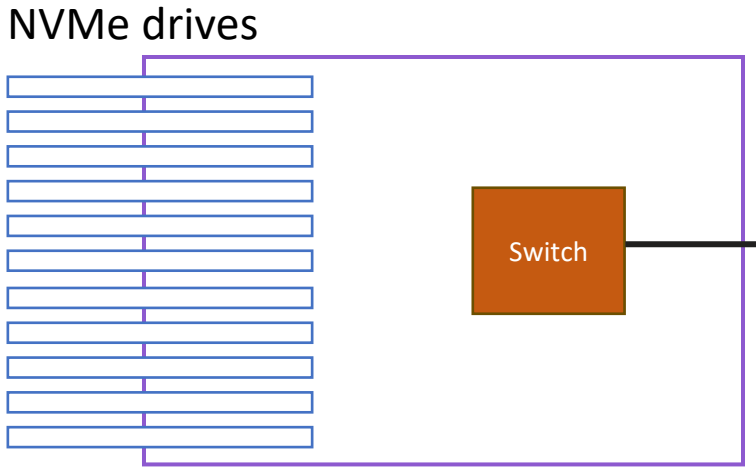
How Could We Confront The Challenge?

Ideal scenario: **performance** of direct-attached local SSD (G3) + **capacity** of cost-effective expansion (G3.5)



How Could We Confront The Challenge?

- Maybe something closer to that is possible?
- What's needed to enable?

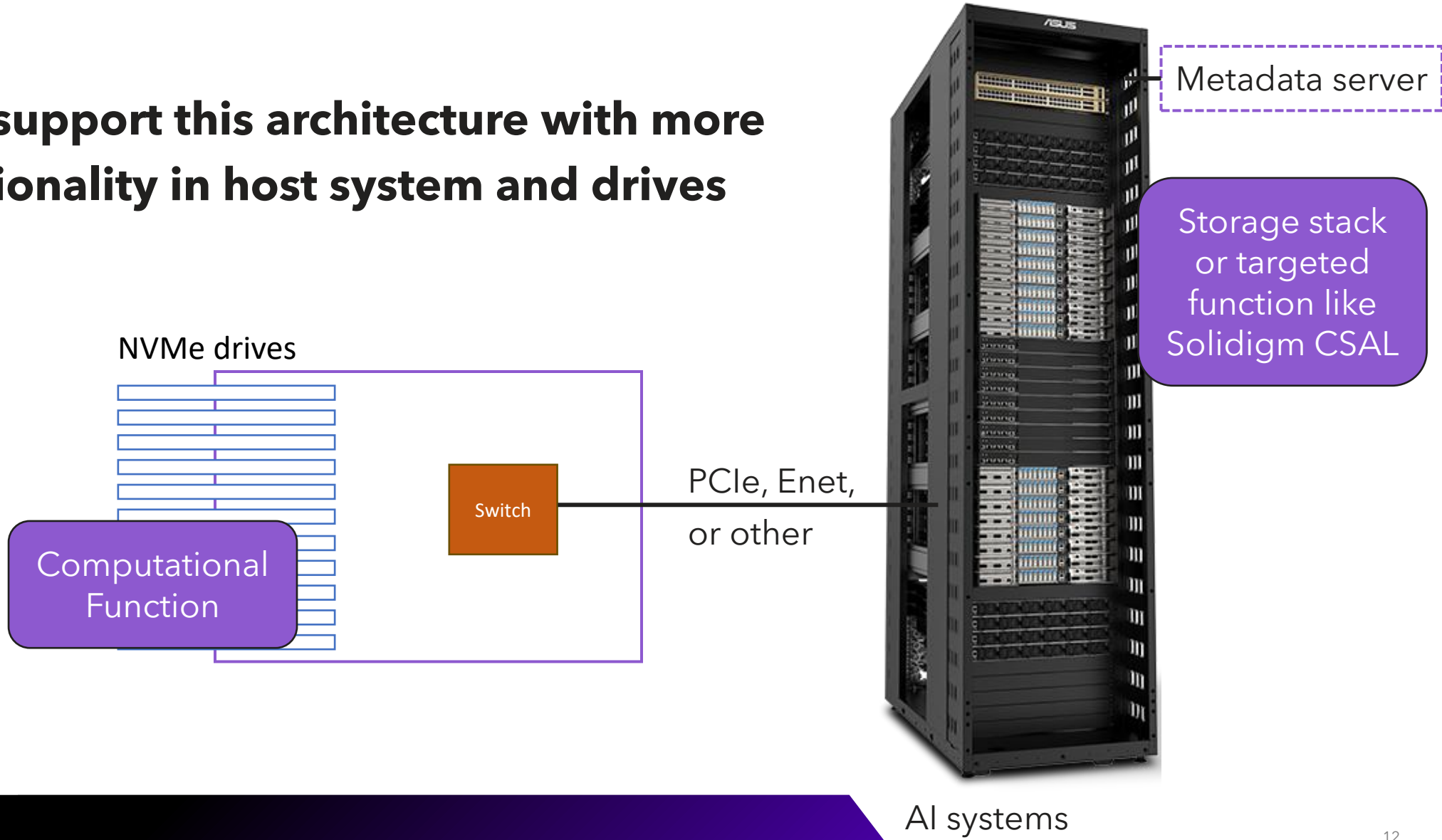


PCIe, Enet,
or other

AI systems

How Could We Confront The Challenge?

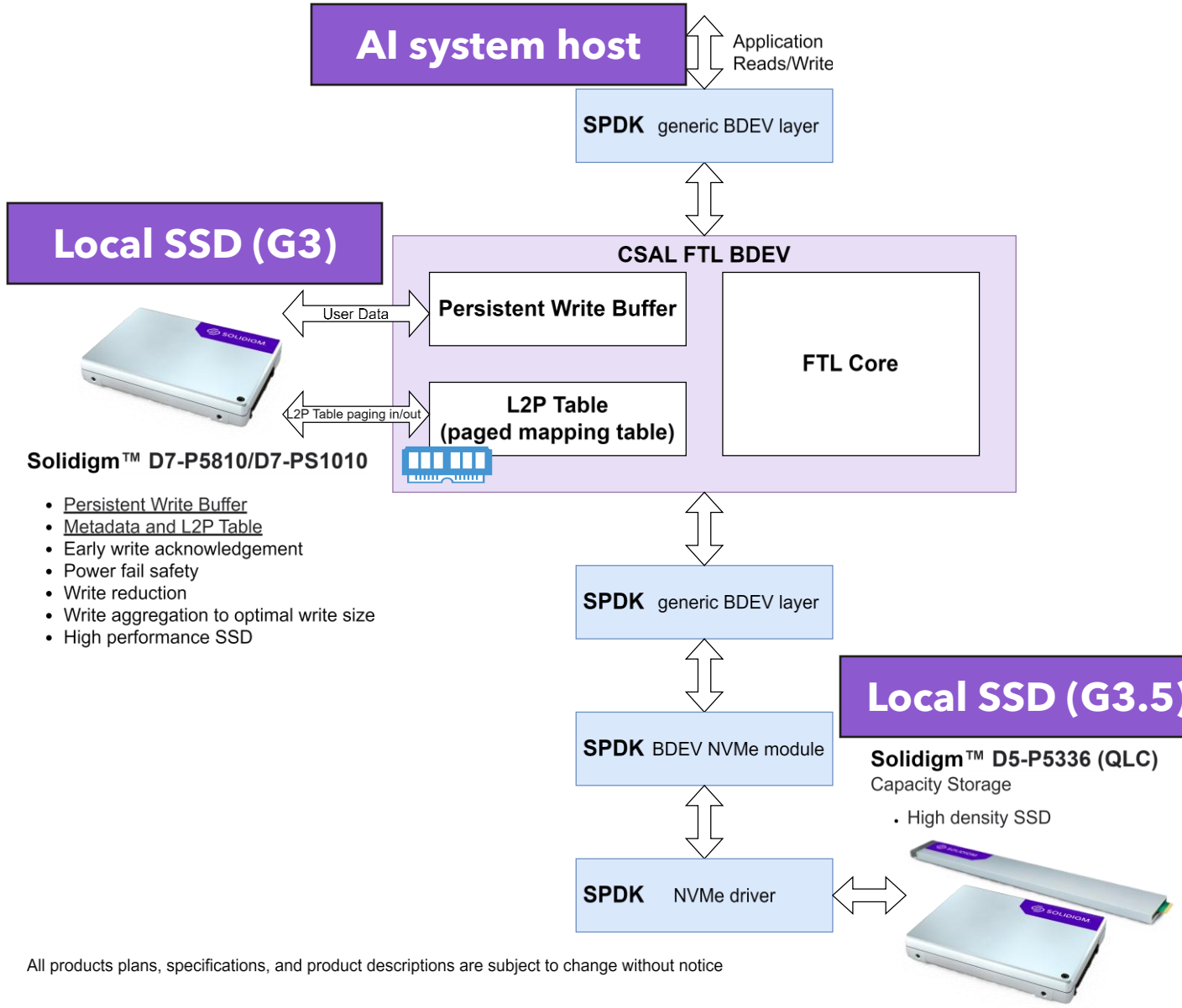
Could better support this architecture with more storage functionality in host system and drives



Write shaping

- CSAL provides transparent block device to the upper application
- Ultra-fast cache and write shaping tier to improve performance and endurance to scale QLC value
- Consistent performance in multi-tenant environment
- Flexible scaling of NAND performance and capacity to the user/workload needs

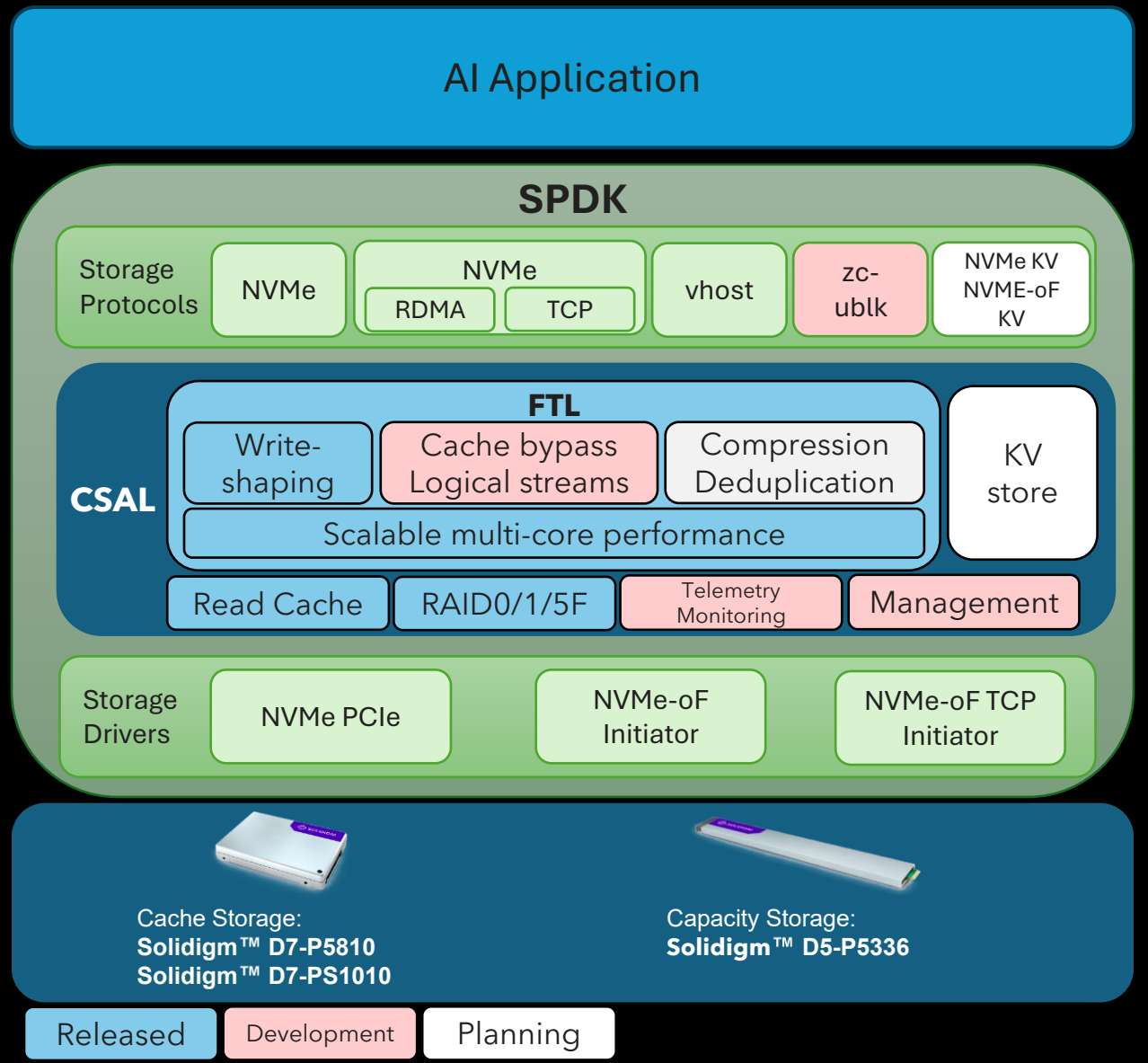
Cloud Storage Acceleration Layer (CSAL)



All products plans, specifications, and product descriptions are subject to change without notice

CSAL today

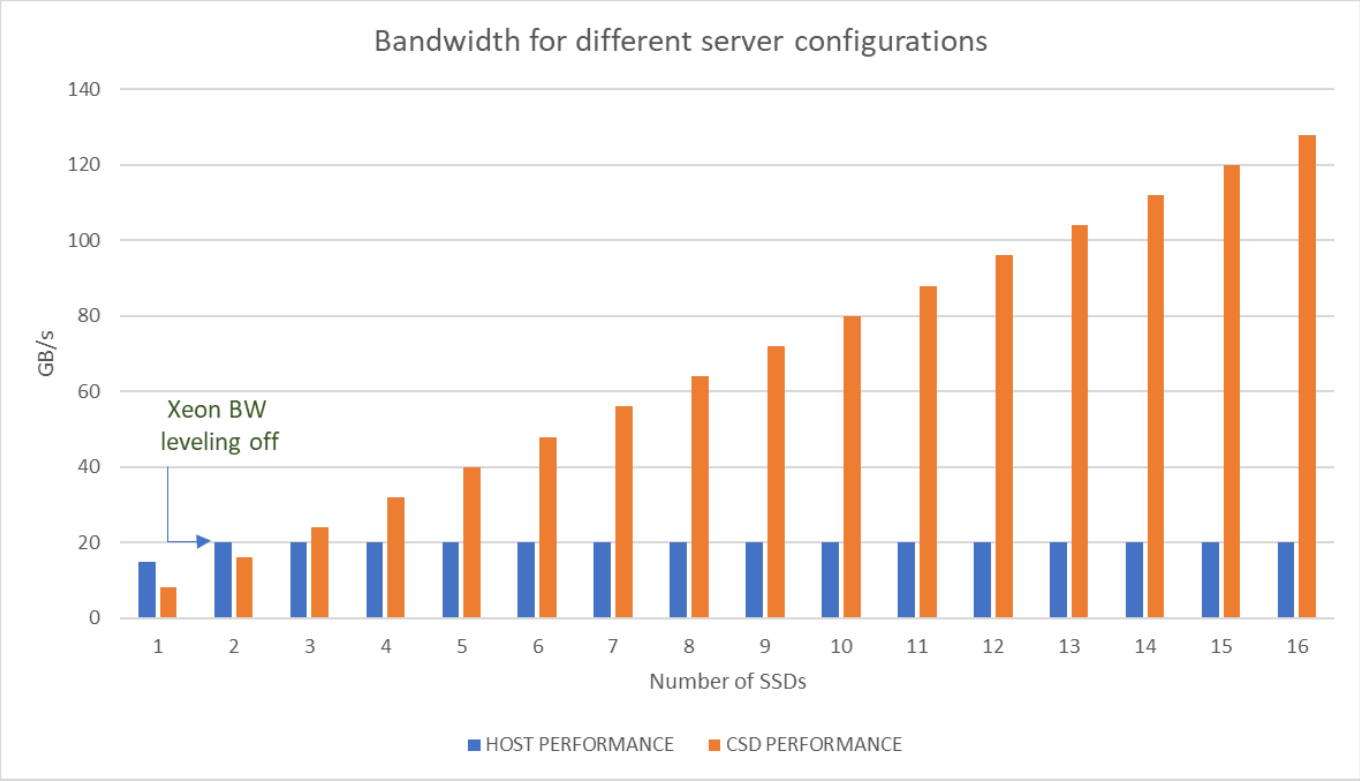
- **Write-Shaping Engine**
  Host-based FTL reshapes writes into NAND-friendly sequential patterns to improve endurance and backend throughput
- **Multi-Core Scalable Performance**
  Shared-nothing architecture distributes FTL, cache, and RAID across CPU cores for near-linear scaling and performance isolation
- **High-Performance Read Cache**
  Append-only, write-through design delivers consistently high bandwidth, low tail latency, and minimal write amplification
- **Flash-Optimized RAID (RAID5F / RAID0 / RAID1)**
  Full-stripe atomic writes eliminate RMW penalties and the RAID5 write hole, boosting efficiency, endurance, and resilience
- **Tier-Aware I/O & Cache Bypass**
  Traffic-shaping logic intelligently directs I/O across high-performance and high-capacity NVMe tiers for predictable
- **Logical Streams for Data Placement**
  Fine-grained, workload-aware data classification improves write efficiency and performance consistency in multitenant environments.
- **Telemetry, Monitoring & Management**
  Deep visibility into FTL, cache, and RAID enables precise control, diagnostics, and operational transparency.
- **Zero-Copy User-Space Data Path (SPDK-based)**
  Enterprise-grade SPDK integration delivers a fast, validated, user-space storage stack with zero-copy efficiency.



Storage Functionality in Drives

Additional storage capability integrated into devices makes data transformation tasks more efficient

- Functionality **multiplies** with number of drives, scales naturally with capacity
- Internal Drive BW >> Fabric BW, speeding up AI is all about the **bandwidth** to access data capacity



Modeling results indicate a high degree of scalability ideal for CPU offload*

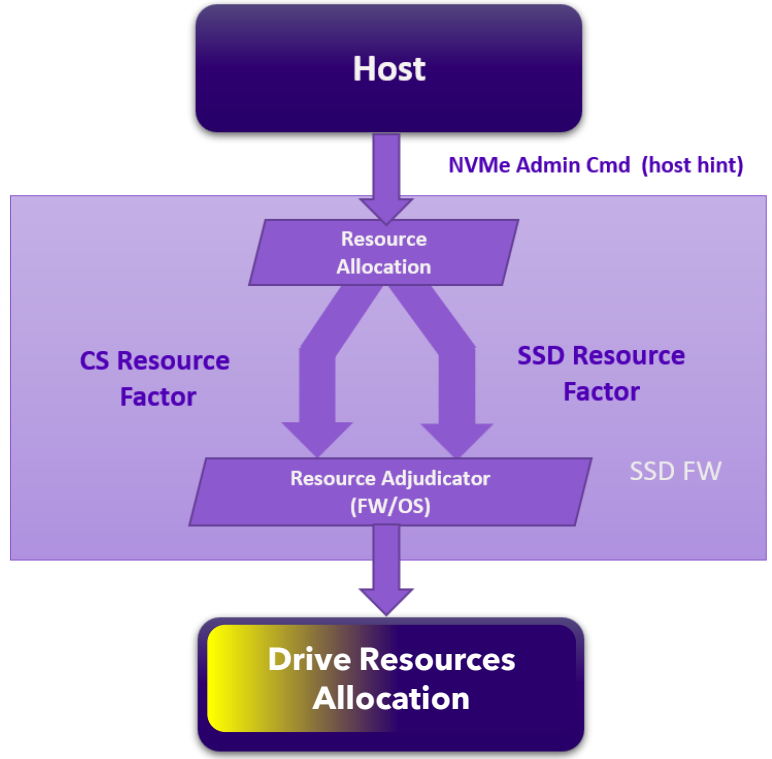
*Based on Solidigm's internal analysis. CSD theoretical performance based upon modeling.

Added Storage Functionality in Drives

Many potentially useful functions, for example:

- Autonomous Data Background Scrubbing (avoid error slowdowns)
- Dynamic Resource Allocation (optimizing drive performance for operations as best fit for needed performance or capacity)
- Offload Engine for Fabric Interface
- Complementary Acceleration of AI Context / KV Cache pre-fetching

Dynamic Resource Allocation



Moving the Industry Forward

Similar to how AI has driven fast changes in computing system architecture…

- Meeting the future needs of AI data growth is going to drive new storage architectures.

Gains when we enable storage capability that grows with AI without artificial limits…

- Higher performance: removing extra hops – bandwidth bottlenecks and latency adders
- Lower power and lower cost: scale cost-effective capacity without impacting local AI system power or significant rack-level power adders
- Simplicity / less complexity to user: increase storage resources decoupled from storage node-level technical limitations or economics