Take a Break: A Deep Dive into Checkpointing for Large-Scale AI Training

April 29, 2026

John Mazzie
MTS, Systems Performance Engineer - Micron

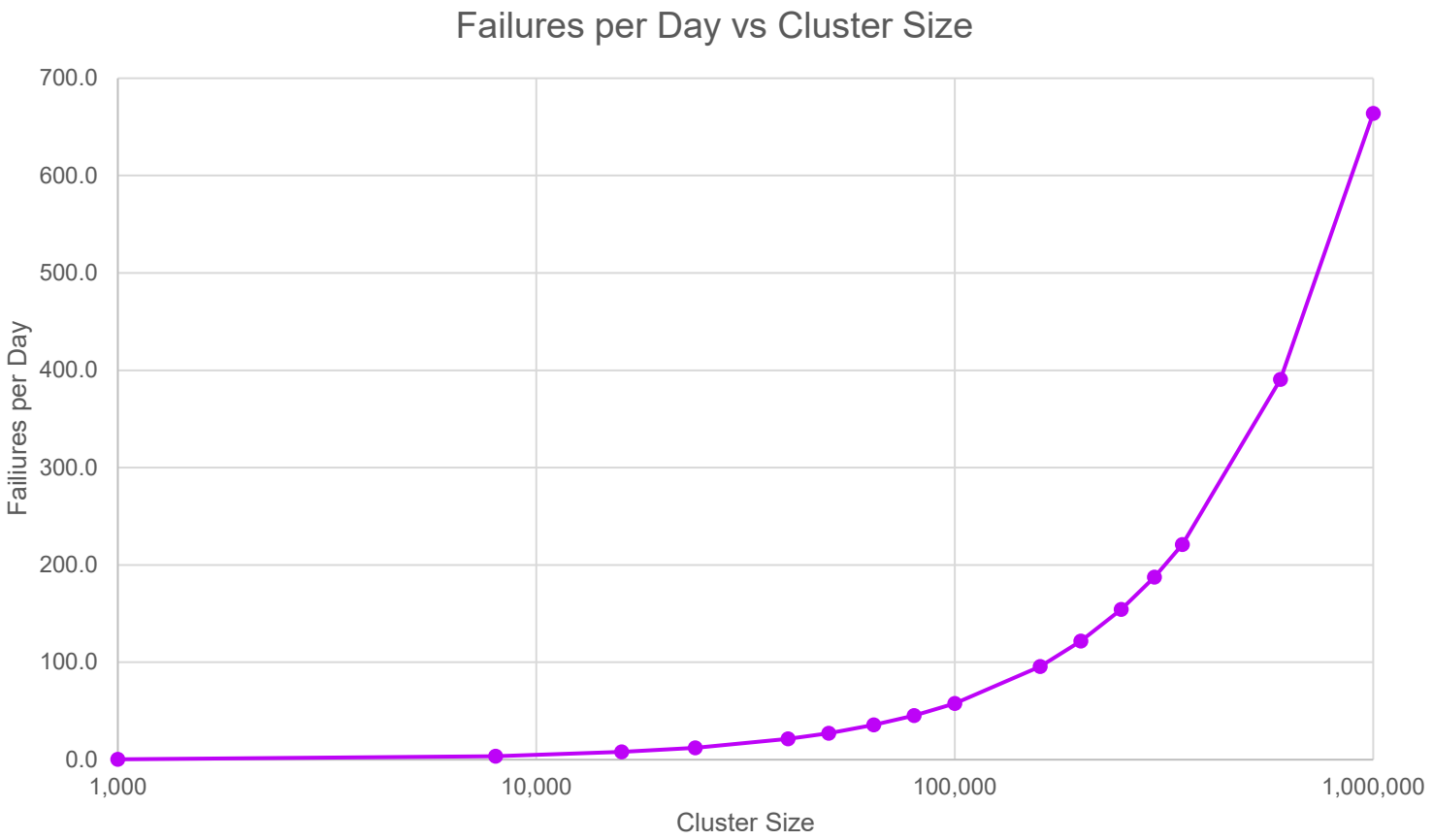
micron Intelligence Accelerated™

SDC | StorageAI™
A SNIA Event
April 29, 2026 • Denver, Colorado

Why Checkpointing

- Training runs now span days to months
- Clusters scale to tens or hundreds of thousands of GPUs
- Failures are inevitable, not exceptional
- Checkpointing determines whether training makes forward progress

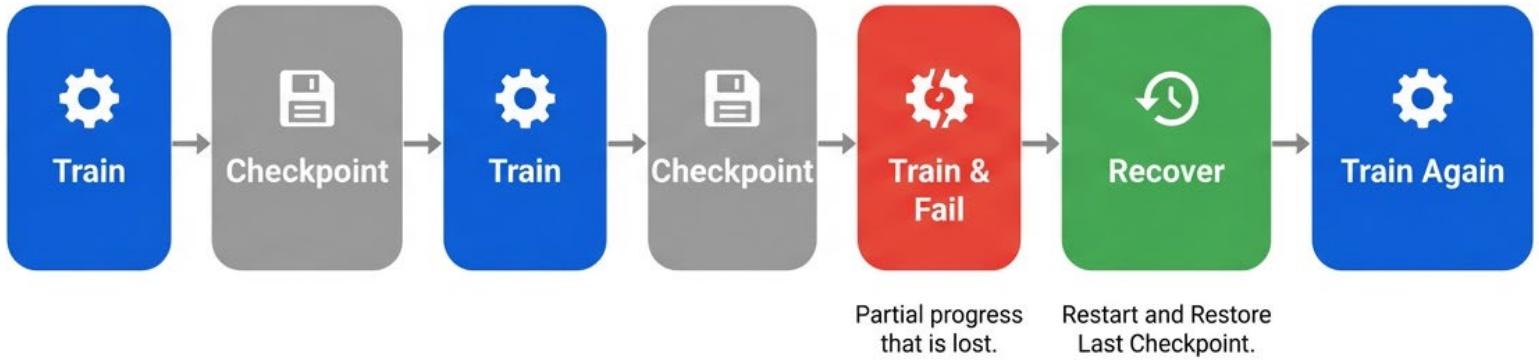
The Hidden Cost of Failure



- AI training is synchronous
- One failed node stalls the entire job
- Failure rate rises with cluster size
- Without checkpoints: lost work = lost GPU hours

Checkpointing: The Safety Net

- Periodically save training state
- Enables:
  - Fault recovery
  - Pause / resume
  - Fine-tuning and reuse
- Previously “background I/O”… now a bottleneck

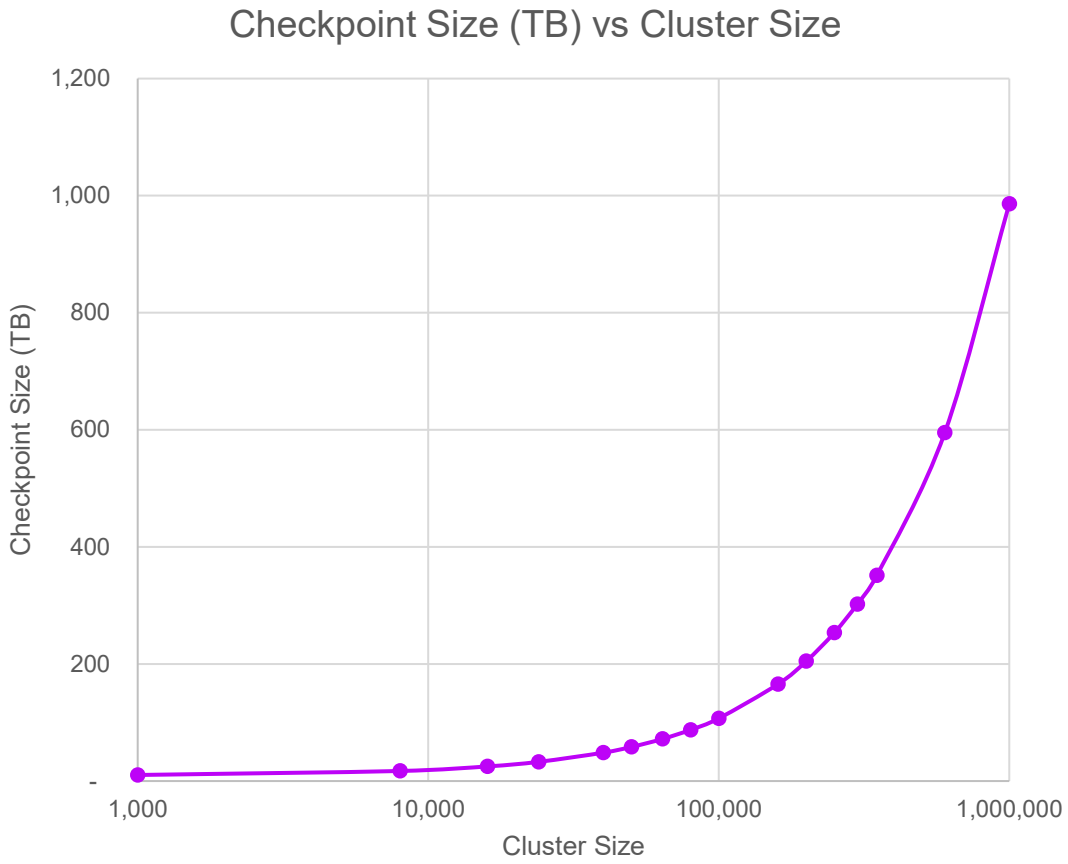


What is in a checkpoint

- Model parameters
- Optimizer state (often dominant)
- RNG state and metadata
- Size grows with:
  - Parameter count
  - Precision
  - Parallelism strategy

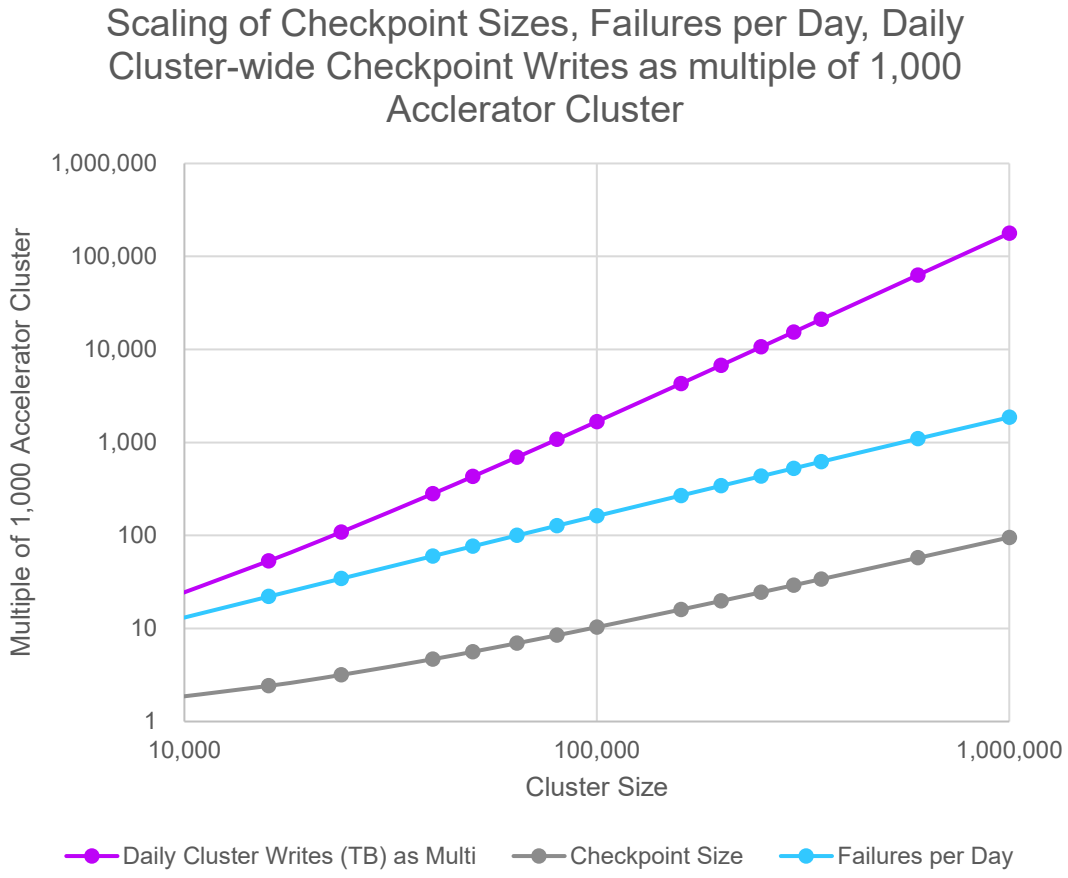
Modern Checkpoint Are Huge

- 10s of GB → TBs per checkpoint
- LLMs: optimizer state scales aggressively
- Checkpoint size does not shrink with more GPUs
- Frequency increases as failure rate increases



Scaling Breaks the Old Assumptions

- Compute scales well with parallelism
- Checkpointing often does not
- Result:
  - GPU idle time
  - Longer time-to-convergence
  - Worse training goodput



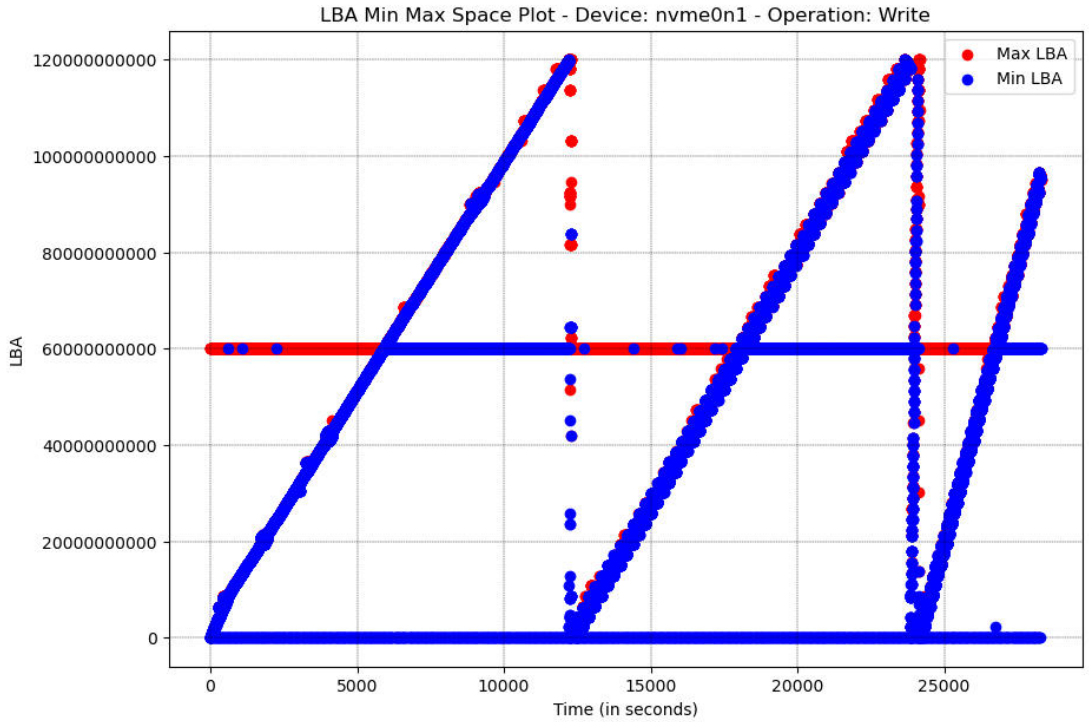
Checkpointing as a Performance Metric

- Goodput = useful training time / wall time
- ETTR = effective training time ratio
- You can't checkpoint:
  - Too slowly
  - Too infrequently

- Historical Analysis
  - Failure rates increase to the power of cluster size
  - Meta's 16k GPU cluster for Llama3 training averaged 7.8 failures per day
  - Modeling to 100,000 GPUs:
    - 50 – 60 failures per day
    - MTTF = 15 – 30 minutes
    - 2,000 – 2,500 checkpoints per day targeting 97.5% Goodput (30 - 40 second target checkpoint interval)

Checkpointing Is a Storage Workload

- Primarily large sequential writes
- Extremely high concurrency
- Highly bursty behavior
- Reads dominate recovery, not steady-state

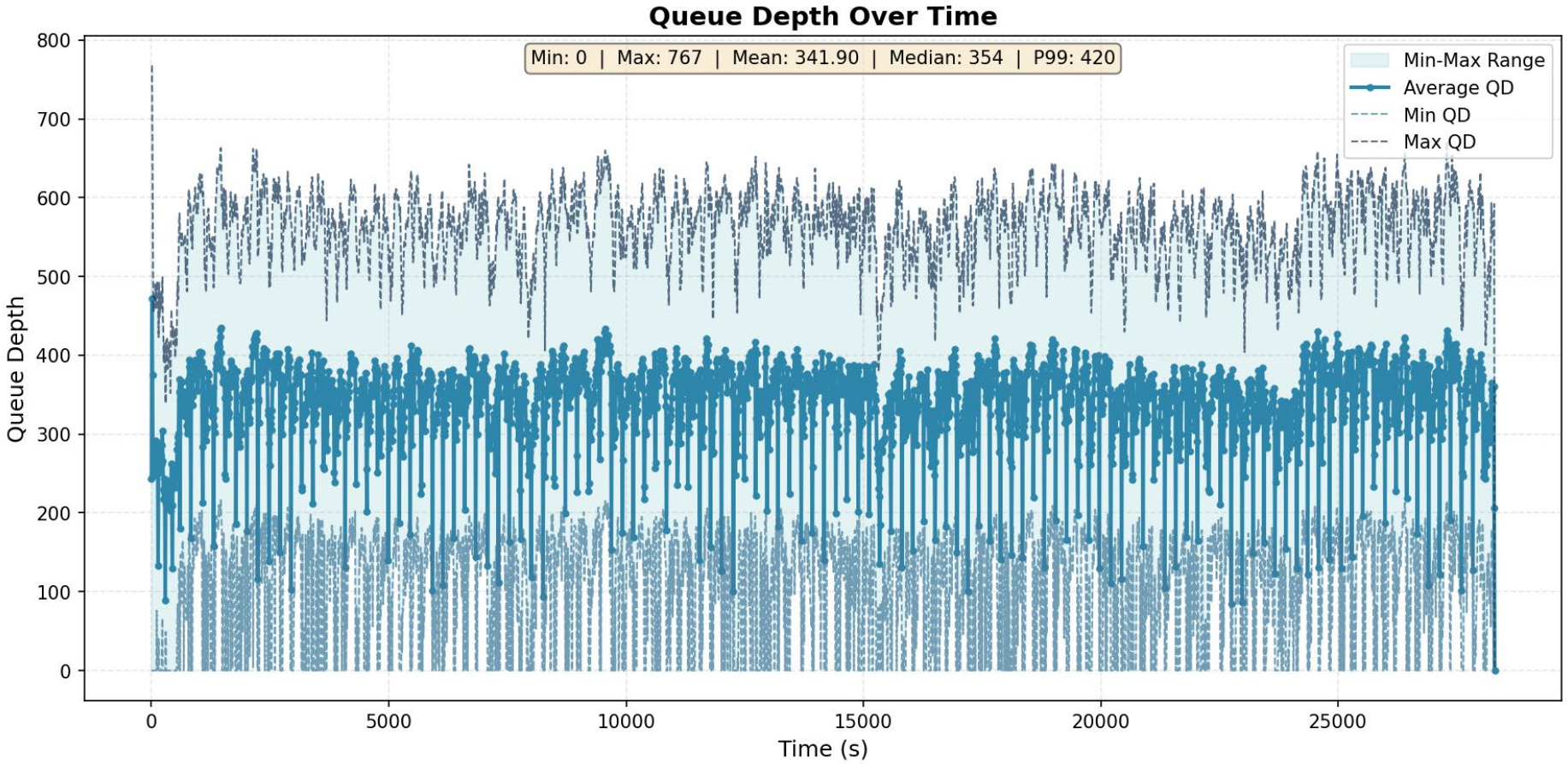


What the I/O Looks Like

- GBs to TBs written in seconds
- Queue depths in the hundreds or thousands
- fsync frequently required
- From the SSD's perspective: "worst-case workload"

# Reads	710
# Writes	340,970,988
Runtime (s)	28,295
Total Writes	39.54 TiB
Average Write Rate	1.43 GiB/s
Peak Write Rate	3.84 GiB/s
Average QD	342
Peak QD	767

QD Over Time


Queue Depth Over Time
Min: 0 | Max: 767 | Mean: 341.90 | Median: 354 | P99: 420
Min-Max Range
Average QD
Min QD
Max QD
Queue Depth
800
700
600
500
400
300
200
100
0
0
5000
10000
15000
20000
25000
Time (s)

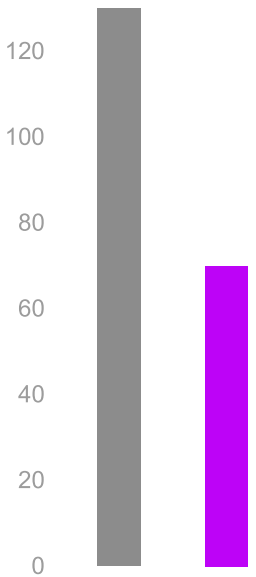
Why General-Purpose Storage Fails

- Tuned for latency or capacity—not sustained bandwidth
- Metadata pressure at scale
- Power and thermal throttling
- Storage becomes the long pole, not GPUs

Checkpoint time (s)

1.9x

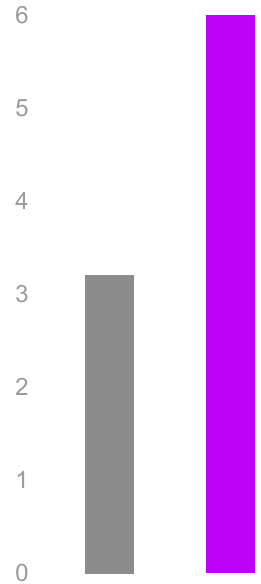
faster checkpoints



Throughput (GB/s)

1.9x

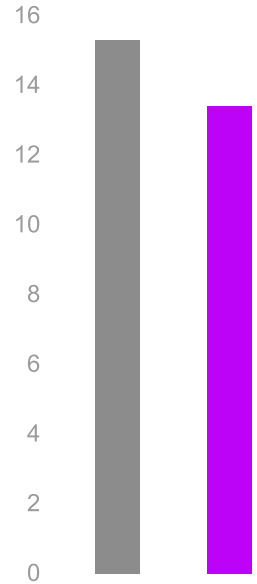
higher throughput



SSD average power (W)

12%

lower SSD power



Benchmarking Checkpointing

- MLPerf Storage v2.0 added checkpoint workload
- Focus on:
  - Save time
  - Restore time
  - Failure-driven frequency

Software Techniques to Reduce Pain

- Asynchronous checkpointing
- Overlap memcpy with training
- Sharded / distributed checkpoints
- Universal checkpoint formats for flexibility

Storage Requirements for Checkpointing

- Sustained write bandwidth
- Extreme parallelism support
- Predictable latency under load
- High energy efficiency and endurance

CPU vs GPU Involvement

- GPUs must freeze briefly
- Data often memcopy’d to CPU memory
- CPU handles persistence
- Faster storage = shorter GPU stalls

Multi-Tier Checkpointing

- Local memory or NVMe for fast saves
- Replication for resilience
- Object or shared storage for durability
- Goal: minimize mean time to recovery

Impact of Faster Storage

	PCIe Gen 4 7.68TB	PCIe Gen 5 7.68TB	PCIe Gen 5 60TB
Checkpoint Time (s)	112.3	66.2	73.9
Throughput (GB/s)	4	6.5	5.8
Energy Use (Joules)	1240	894	897

- Shorter checkpoint windows
- Higher checkpoint frequency without penalty
- Better protection against failures
- Measurable increases in goodput

Power Matters Too

- Checkpoints are write-heavy
- SSD efficiency directly affects cluster energy
- Faster ≠ higher power if designed correctly
- Storage efficiency scales with cluster size

Implications for AI System Design

- Storage can no longer be an afterthought
- Checkpointing influences:
  - Cluster sizing
  - Failure models
  - Cost per trained model
- “GPU-first” design is incomplete

Solution Comparisons

Local Storage

- Very fast checkpoint persistence (1-2 seconds)
- Low overhead
- Requires off-host copies of some checkpoints
- Host outage results in recovery to further back point in time

Flash optimized shared storage

- Fast checkpoint persistence with optional GPU Direct Storage
- No tiering required
- Simple recovery

HDD-based network storage

- Persistent checkpoints
- May require tiering
- Simple recovery
- “cost effective”

Key Takeaways

- Checkpointing is a first-class workload
- Optimizer state dominates at scale
- Storage throughput limits checkpoint frequency
- Faster checkpoints = higher training efficiency

Looking Forward

- Checkpoint-aware system design
- Better benchmarks and tooling
- Increased focus on goodput, not peak FLOPS
- Storage as an enabler of AI scaling

micron Intelligence Accelerated™

SDC | StorageAI™

A SNIA Event

References

- MLPerf Storage v2.0
  - GitHub - mlcommons/storage: MLPerf® Storage Benchmark Suite · GitHub)
- FastPersist (Microsoft DeepSpeed)
  - FastPersist: Accelerating Model Checkpointing in Deep Learning - Microsoft Research
- Argonne DLIO Benchmark
  - Deep Learning I/O Benchmark — DLIO 2.0 documentation
- Google Multi-tier checkpointing
  - Using multi-tier checkpointing for large AI training jobs | Google Cloud Blog