HAMMERSPACE

A Detailed Analysis of Using Hammerspace Tier 0 for Checkpointing

V. 1.0 - November 2024

Introduction

In high-performance computing (HPC) and AI applications, checkpointing is the process of periodically saving the state of an application to persistent storage, allowing recovery from failures without restarting computations from the beginning.

However, checkpointing can introduce significant overhead, particularly when writing large amounts of data over the network to shared storage, and introduces these challenges.

Challenges Associated with Checkpointing to External Networked Storage

- **GPU Idle Time:** During checkpointing, GPUs often remain idle until all data is written to shared storage, leading to inefficient utilization of expensive compute resources.
- **Shared Storage Bottlenecks:** Simultaneous writes from multiple nodes to a shared storage system can overwhelm the network and storage bandwidth, increasing checkpointing time.
- **Risk of Data Loss:** Relying solely on local storage without redundancy can risk data loss if a node fails before the checkpoint is safely stored elsewhere.

This whitepaper defines Tier 0, explains how to use it in a checkpointing workflow, and quantifies the benefits of using Hammerspace Tier 0 storage for checkpointing.

Benefits of Using Hammerspace Tier 0 for Checkpointing

- **Dramatic Reduction in Checkpoint Times:** From minutes to seconds.
- **Significant GPU Time Savings:** Equivalent to adding hundreds of GPUs.
- **Substantial Financial Benefits:** Avoiding tens of millions in additional hardware costs.

Using Hammerspace Tier 0 for Checkpointing

Hammerspace Tier 0 is a new tier of ultra-fast, shared storage that uses the local NVMe storage in GPU servers and turns it into a tier of shared storage by making those storage volumes part of the Hammerspace Parallel Global File System.

Data in Tier 0 can then be protected using Hammerspace protection policies – for example mirroring data across GPU server nodes or ensuring three copies of data exist at all times on different tiers – and can be orchestrated to move seamlessly between tiers of storage, or between sites and clouds.

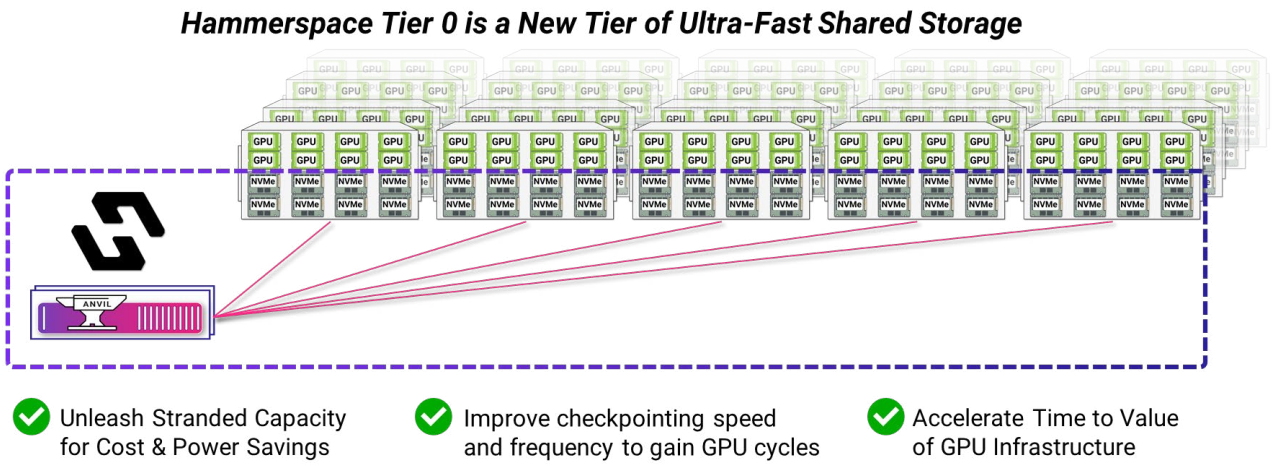

Figure 1 – Hammerspace Tier 0 is a new tier of ultra-fast shared storage that leverages GPU server local NVMe storage as part of a Parallel Global File System.

Hammerspace Tier 0 leverages the Hyperscale NAS architecture – a standards-based parallel file system architecture – and takes advantage of an update to the Linux kernel that bypasses the NFS client and server, along with the networking stack and network adapter hardware that connects the NFS client and server (depicted in Figure 2 below).

Effectively this creates a 'shortcut' or more direct data path between the GPUs and the local NVMe storage, which reduces latency, increases bandwidth and as demonstrated in this analysis speeds up checkpointing by a factor of 10x-100x.

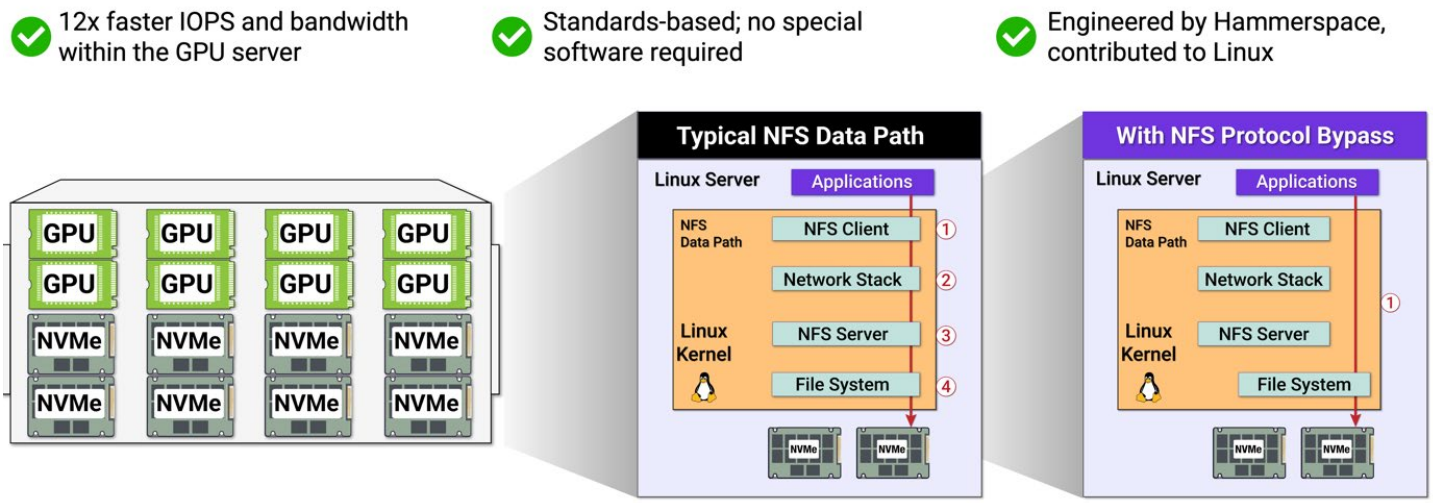

Figure 2 – Hammerspace Tier 0 takes advantage of a Linux kernel update that creates a more direct data path between GPUs and local storage to reduce latency.

Example Checkpointing Workflow with Hammerspace Tier 0

Hammerspace Tier 0 delivers data directly to GPUs at local NVMe access speeds, is an ideal solution for extreme high performance use cases like AI/HPC applications and delivers several advantages for checkpointing.

An example checkpointing workflow with Hammerspace Tier 0 looks like this:

Step 1 – Checkpoint Initiation: The application triggers a checkpoint, at which point each node creates a file on the Hammerspace global shared storage system. Based on Service Level Objectives set up in Hammerspace, the storage node selected to back each file is the NVMe that is local to each node, thus making it possible to bypass the NFS protocol and networking stack as described above.

Step 2 – Local Write Completion: Once the write is complete, the GPU resumes computation without delay.

Step 3 – Asynchronous Replication: Hammerspace detects the new checkpoint file and begins replicating it to designated storage tiers based on policies.

Step 4 – Data Availability: The checkpoint is now safely stored in multiple locations, ensuring it can be used for recovery if needed.

Advantages of Using Hammerspace Tier 0

This workflow delivers several advantages over external storage:

Advantage # 1: Write Checkpoints to Local NVMe Storage, Bypassing Network Stack

- **Local Speed Advantage:** Compute nodes write checkpoint data directly to their own high-speed local NVMe storage, eliminating network latency and congestion.
- **Non-Redundant Writes:** Initial writes are non-redundant and local, allowing for rapid data dumping from memory to disk without the overhead of network transfers or data replication.

Advantage # 2: Automate Data Protection, Affinity, and Movement with Data Orchestration

- **Policy-Based Objectives:** Using Hammerspace's data orchestration, administrators set policies – called Objectives that specify how checkpoint data should be handled.
- **Affinity Policies:** Ensure that checkpoint files are written locally on the node's NVMe storage.
- **Replication Policies:** Define when and how checkpoint data is replicated to other storage tiers for redundancy.

Advantage # 3: Asynchronously Replicate and Migrate Checkpoint Files

- **Data Movement:** After the checkpoint file is closed, Hammerspace's data movers asynchronously replicate the data to other nodes or storage tiers.
- **Background Operation:** Replication occurs in the background, allowing GPUs to resume computation immediately after the local write completes.
- **Data Safety:** The checkpoint data is safely stored elsewhere for recovery purposes without impacting the compute workload.

Advantage # 4: Local Storage Becomes Part of a Global File System

- **Unified Namespace:** The checkpoint file resides within the global shared file system, making it accessible and consistent across the cluster.
- **Logical Consistency:** Physically, the data moves from local NVMe to other storage, but logically, it remains the same file within the shared system.

The Power of Hammerspace Data Orchestration in a Global Namespace

Even as Hammerspace is mirroring data for protection purposes or moving data for tiering purposes – it is the same file in the same file system namespace at all times. It is only instantiations of the file that move from local storage to other locations (other GPU servers or external shared storage nodes). The file is logically always in the same place in the namespace, even though the contents of the file are physically moved out elsewhere.

Quantifying the Benefits of Using Hammerspace Tier 0 for Checkpointing

This analysis quantifies the benefits of using Tier 0 storage—specifically, leveraging local NVMe storage within compute nodes—for checkpointing, compared to traditional methods that rely on external networked storage systems, even those connected via high-speed 800Gb Ethernet (800GbE).

We will use the NVIDIA A100 GPU for our calculations, consider a 1,000-node cluster with 8,000 GPUs, 100 Petabytes of storage capacity, and 1 TB/sec of aggregate throughput.

This analysis then estimates failure rates, calculates optimize checkpoint frequency, and calculates the annualized total GPU hours gained, and the associated opportunity cost.

Estimating Checkpoint Data Size

System Configuration:

- **Compute Node:** NVIDIA DGX A100 or HGX system
- **GPUs per Node:** 8 NVIDIA A100 GPUs
- **GPU Memory per GPU:** 80 GB (also available in 40 GB variants)
- **Total GPU Memory per Node:** 8 GPUs × 80 GB = 640 GB
- **CPU Memory:** Assume 256 GB per node (can vary)
- **Total Memory to Checkpoint:** GPU Memory + Relevant CPU Memory

Checkpoint Data Size Estimate = 600 GB

Estimating Checkpoint Time When Writing to External Storage

Considerations when Writing Checkpoints to External Storage

- **Network Overheads:** Protocol overheads, congestion, and latency reduce effective bandwidth.
- **Shared Infrastructure:** Multiple nodes checkpointing simultaneously can saturate the network and storage array.
- **Effective Bandwidth per Node:** Often significantly less than theoretical maximum. Let's conservatively estimate 1 GB/s per node.

Time to write a 600 GB checkpoint to external storage:
600 GB / 1 GB/sec = 600 seconds

Estimating Checkpoint Time When Writing to Tier 0 Local NVMe Storage

Local NVMe Storage Specifications:

- **NVMe Devices per Node:** 8 NVMe drives
- **NVMe Interface:** PCIe Gen5
- **Bandwidth per NVMe Device:** Approximately 14 GB/s (real-world write performance)
- **Total Aggregate Bandwidth:** 112 GB/s per node
- **Effective Bandwidth Assuming 90% Efficiency:** 100.8 GB/s per node

Time to write a 600 GB checkpoint to Tier 0 storage:
600 GB / 100.8 GB/sec = 5.95 seconds (round to 6 seconds)

Calculating the Time and GPU Hours Saved with Tier 0 Storage

Based on the assumptions above and estimating GPU server failure rates, we can calculate optimal checkpointing intervals, then calculate total overhead per hour associated with checkpointing and recomputation.

The comparison between using external networked storage and using Tier 0 storage is shown in the summary table below.

Parameter	External Storage	Tier 0 Storage
Checkpoint Data Size per Node	600 GB	600 GB
Effective Bandwidth per Node	1 GB/s	100.8 GB/s
Time to Write Checkpoint	600 seconds	6 seconds
GPU Server Failure Rate per Hour	0.0417 failures/hr	0.0417 failures/hr
Optimal Checkpoint Interval	54 minutes	17 minutes
Checkpoints per Hour	1.12	3.54
Time Spent Checkpointing per Hour	212.13 sec/hr	21.21 sec/hr
Expected Recomputation Time per Hour	212.13 sec/hr	21.29 sec/hr
Total Overhead per Hour per Node	424.26 sec/hr	42.5 sec/hr
Total Overhead per Day per Node	170 min/day	17.0 min/day
Total Overhead per Year per Node	1032.4 hours/year	104.17 hours/year
Total Annual Savings per Node	N/A	928.2 hours/year
Total Annual Savings for 1,000 node Cluster	N/A	928,196 hours/year
Equivalent Additional GPU Servers (total annual savings / 8,760 hrs/yr)	N/A	~106 servers
Equivalent Additional GPUs (8x per server)	N/A	~848 GPUs

Calculating the Economic Benefit of Using Tier 0

The result of the analysis above shows that in a 1,000 node GPU cluster with 8,000 GPUs, using Tier 0 instead of external storage adds the equivalent of another 106 GPU nodes, or another 848 GPUs.

This is equivalent to adding **10.6% more GPU computational capacity** per year without any additional investment!

To calculate the equivalent opportunity cost in dollar terms:

- **GPUs Gained:** 848
- **Cost Per GPU:** $20K to $40K
- **GPU Opportunity Cost Per Year:** $17M - $34M

So using Tier 0 for checkpointing frees up anywhere from **$17 Million to $34 Million in GPU Opportunity Cost annually.**

In addition to the opportunity cost gains above, Tier 0 reduces the need for external storage capacity. This delivers a significant cost savings – for the capacity point of 100 Petabytes used in this analysis, the savings is on the order of $40 Million to $50 Million dollars. You can read more about the ROI of Tier 0 in this Executive Brief.

Conclusion

The analysis demonstrates that external shared storage systems cannot match the efficiency and performance of local NVMe storage for checkpointing in large-scale GPU clusters.

By unlocking that local NVMe storage and making it available as a new tier of shared storage as part of a Parallel Global File System, Hammerspace Tier 0 provides:

- **Dramatic Reduction in Checkpoint Times:** From minutes to seconds.
- **Significant GPU Time Savings:** Equivalent to adding hundreds of GPUs.
- **Substantial Financial Benefits:** Avoiding tens of millions in additional hardware costs.

Hammerspace Tier 0 is essential for maximizing the performance and cost-effectiveness of modern HPC and AI workloads.