WHITE PAPER

HAMMERSPACE

Hammerspace Technology White Paper

An Overview of the Key Components and Performance Metrics of **Hammerspace Software**

Table of Contents

Table of Contents

Executive Summary

This White Paper is designed as a technical overview of Hammerspace software, to provide you with a summary of the key components of the architecture and its primary capabilities.

The flexible scale-out architecture of the system is an essential attribute that enables Hammerspace to deliver high-performance user/application access and I/O across heterogeneous storage silos, but also across distributed environments that include one or more on-premises data centers, clouds, and cloud regions.

As such, this paper also includes a section on performance testing where we illustrate the linear scalability of the system, showing that it is constrained only by the capabilities of the underlying infrastructure. These results show how Hammerspace can scale to accommodate virtually any workflow across multi-vendor and otherwise incompatible storage environments. In this way, as the performance of the backing storage or network is increased, the throughput and IOPS of Hammerspace for data orchestration and file access can scale up and out linearly as well, virtually without limitation.

This is important to facilitate high performance use cases, and when needed to bridge multiple storage systems in parallel to achieve even extreme levels of I/O, IOPS, and support for large distributed user communities.

1. Overview–Hammerspace Technology

What is Hammerspace?

Hammerspace is a software-defined data orchestration and storage solution that provides unified file access via a high-performance Parallel Global File System that can span different storage types from any vendor, as well as across geographic locations, public and private clouds and cloud regions.

Designed to make data a global resource across distributed or otherwise incompatible storage platforms, Hammerspace with its Parallel Global File System presents a cross-platform global namespace where users and applications can have direct multi-protocol access to all files, regardless of which storage type or location they are in today, or move to in the future.

With the Hammerspace Parallel Global File System, the metadata layer is common across all users, everywhere. In other words, all users access the same file metadata regardless of where the files are actually stored, without the need to manage file copies between silos or locations. If a local instance of a file is needed for processing in one part of the world, orchestration of a file instance is a background operation that is transparent to the users or applications. The metadata layer of the file system is still the same across all silos and locations. This means users everywhere are accessing the same file metadata, not forked file copies.

Why This Matters?

The key benefit of Hammerspace technology is that it enables organizations to decouple file access by users or applications from changes to storage infrastructure, or from data movement between different storage types and locations. Doing so enables always-on global file access to users and applications via standard protocols with persistent mount points, regardless of which storage types the data moves to, or what location they are accessing the data from.

Objective-based policies may be established within Hammerspace for moving data at a file-granular level completely in the background for tiering, platform migration, data protection, workflow provisioning and much more. Users and applications retain the same view to the global file system regardless of changes at the infrastructure layer or data placement actions over time.

In addition, this also means that critical data services may be implemented globally, for data protection and other use cases, without the complexity or fragmentation often associated with limited point solutions, gateways, or other vendor-locked techniques.

Scalability and Flexibility

Of critical importance, Hammerspace is a software-defined architecture designed to enable maximum flexibility in deployment options and scalability. In this way, Hammerspace may be initially configured for small workloads, but later expanded dynamically as needed to accommodate changing performance requirements.

This includes deployments that must scale-up to accommodate high-IOPS application needs, or scale-out to accelerate large volume throughput and/or large numbers of concurrent users. In fact, this deployment flexibility enables environments to be designed around any combination of these three variables (IOPS, throughput, user load) to within the limits of the underlying storage hardware and networking infrastructure.

As use cases evolve over time, Hammerspace can also dynamically expand or contract on any of these performance axes, by adding/removing storage or compute resources and/or increasing network bandwidth. All of this can occur transparently in the background without interruption to user/application access to data.

The ability to non-disruptively reduce the size of the Hammerspace clusters is of particular importance for burst-to-cloud use cases. In such scenarios, this enables rapid provisioning and decommissioning of resources for specific application runs. HPC-type workflows for EDA (Electronic Design Automation), genomics research, seismic processing and other compute intensive operations benefit from this capability.

This enables a high-performance compute environment to be rapidly provisioned in the Cloud to accommodate intense I/O or IOPS requirements for a limited run, and then just as rapidly be decommissioned to reduce cloud compute and networking costs. This also enables such workflows to keep their other application licensing costs and at a minimum.

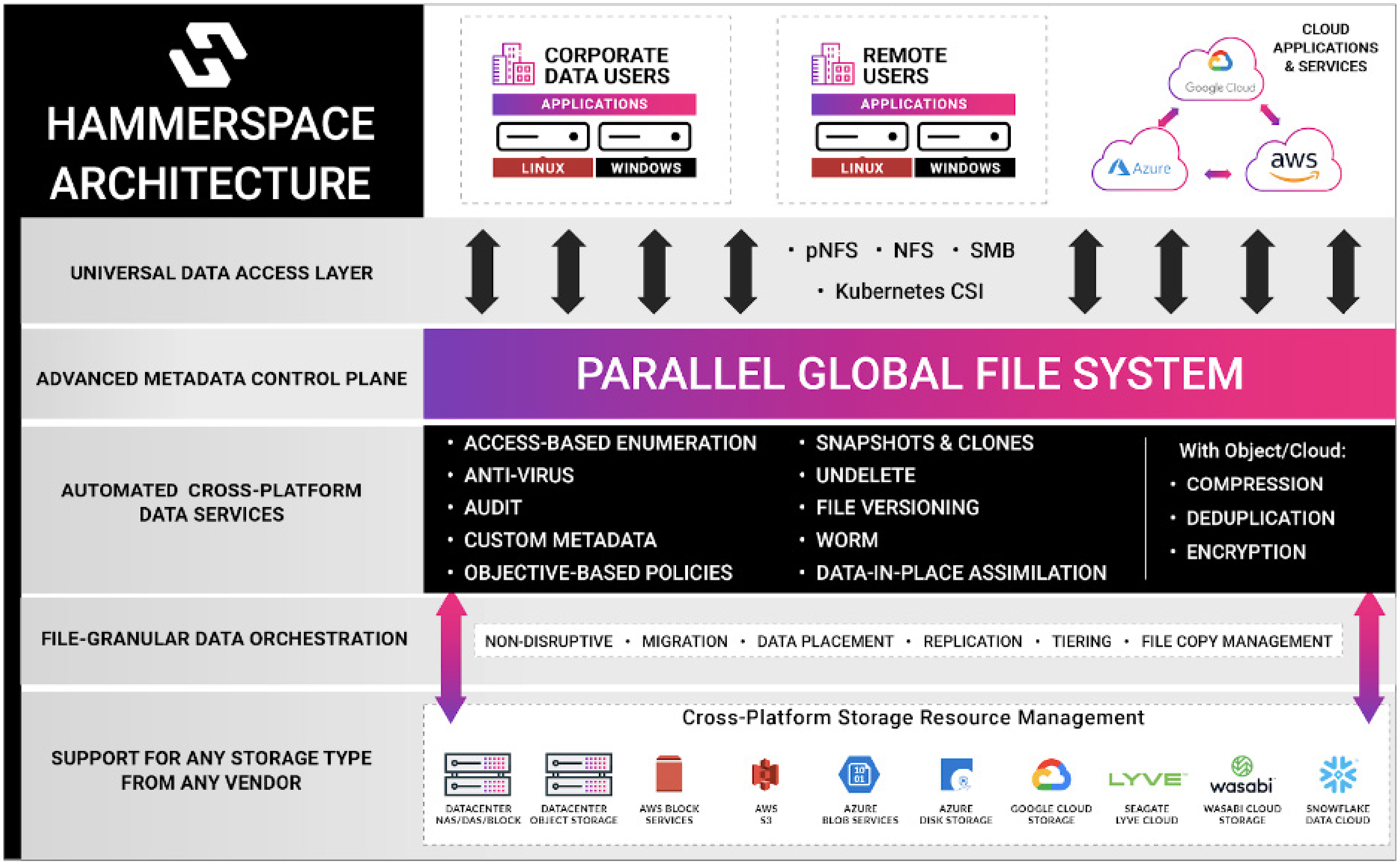
What Follows Below

In the sections below, we will outline the key components of Hammerspace technology. We'll introduce the architecture of the Hammerspace software stack, show the basic building blocks, for deploying Hammerspace, and highlight some performance examples based upon different configurations and use cases.

2. Hammerspace Architecture

Hammerspace is deployed as a fully-integrated software solution built upon open standards, which includes all components within a single installer that are necessary for deployment on bare metal servers, VMs or in the Cloud-based compute environments. No external software dependencies are required, including the LinuxOS.

Hammerspace software is functionally organized in the following five capability layers. Although all are seamlessly integrated in the Hammerspace software stack, each layer organizes capabilities logically to more easily understand key software functionality.



For deployment choices, and a technical overview of the software components, see the Building Blocks section below. In this section you'll find a description of each of the logical layers of the Hammerspace architecture, and the role these capabilities contribute to the whole solution.

2.1 Universal Data Access Layer

The Universal Data Access Layer is the on-ramp to Hammerspace, presenting multi-protocol file access for users and applications. All users/apps see the same file system view, based upon their permissions, regardless of which protocol they are using. Authenticated SMB users see the same folder/file structure that authenticated NFS users do across all underlying storage, including NAS, object, and cloud storage.

Data is presented to users via file shares, with industry-standard protocols as listed below. From a user's perspective, it is a hierarchy of directories and files just like any NAS or file server. But unlike conventional storage platforms, the file system metadata is elevated above the infrastructure layer, and does not need to be kept in the same disk silo as the contents of the files.

In this way, Hammerspace provides front-side access by users and applications to all data, across all back-end storage types from any vendor, in any location, including the Cloud. This capability enables global access by users and applications, regardless of which storage platform the files may be on at the moment, or move to in the future.

Hammerspace also includes a comprehensive open API for direct application integration, so third-party applications, machine learning processes, or other use cases can directly interact with data globally across all the underlying data silos.

As noted below in the topics on the Parallel Global File System, and Hammerspace Building Blocks, this elevation of the file system metadata above the infrastructure layer is a crucial component of the Hammerspace scale-out architecture. Because the file system is not trapped at the infrastructure layer, it means that performance may be increased linearly when needed to bridge multiple storage systems in parallel, without disruption to user access, to achieve even extreme levels of I/O, IOPS, and support for large distributed user communities.

This flexibility is critical to understanding the ability of Hammerspace to scale up and out to achieve required performance targets, within the limitations of the available storage hardware and networking bandwidth. We'll talk more about that topic below in the Performance test examples.

2.1.1 - Multi-Protocol Support

- NFS v.3 & 4.2 file protocols.
    - Support for NFS 4 ACLs
- SMB v.1, 2. 3. 3.1 & later

2.1.2 - Authentication and User Mapping

Hammerspace provides unified permissions and user mapping between Windows and Linux environments, with full support for RFC2307 and RFC2307bis user mapping.

- Users are authenticated using standard enterprise sign-on with Active Directory.

2.1.3 - Access-based Enumeration

Hammerspace supports Access-based Enumeration for both SMB and NFS protocols.

- This allows for files and folders to be hidden when users do not have list or read permissions to those objects.
- ABE can be implemented at the share/directory level, or at a file-granular level using Hammerspace Objectives.

2.2 Parallel Global File System

As noted above, unlike conventional storage platforms that embed the file system within the infrastructure layer, Hammerspace elevates the file system above the storage layer so it can span otherwise incompatible storage silos across one or more locations including the cloud.

In traditional storage architectures where the file system is embedded in the storage platform, if files need to be moved to another storage type or location, a copy of both the file metadata and the file essence is required to be sent. That action now creates a second, forked copy of the file that must be later reconciled.

Because the Hammerspace Parallel Global File System is independent of the storage layer, the need to wrangle such forked file copies is no longer necessary. With Hammerspace, all users in all locations have read/write access to all data everywhere. Not to file copies, but to the same files via this unified global metadata control plane.

The actual file instance, or essence, may exist in one or more locations based upon Objectives-based policies, user workflows, and so on. But unlike with the forked copies needed for traditional storage architectures, all users accessing the Hammerspace Parallel Global File System are sharing the same file metadata just as they would if they were all in a local office on a single network share.

2.2.1 - Custom Metadata

Additionally, Hammerspace enables files and directories to be tagged with user-defined custom metadata, creating a richer set of descriptive information about what the file is, which department or cost center it belongs to, or other variables as needed. In this way, all metadata variables including custom metadata may be used to trigger data placement actions at a file-granular level based upon Objective-based policies.

Hammerspace's innovation is to enable these capabilities seamlessly and at high performance across one or more storage platforms from any vendor, and across multiple locations, including public and private cloud at any scale. With Hammerspace's Parallel Global File System spanning incompatible storage types and locations, this means customers can select storage from any vendor or any performance/price band, to mix and match as needed to meet their specific budget and performance needs.

In addition, this decoupling of the file system layer from the storage layer enables independent scaling of I/O and IOPs at the data layer. Extremely high performance NVMe storage can now co-exist with lower cost and lower performing tiers including cloud in a global data environment. Data orchestration between tiers and/or locations is controlled transparently as a background operation based upon workflows or Objective-based policies.

In the performance tests below we illustrate the linear scalability of the system, showing that it is limited only by the capabilities of the underlying infrastructure. The result is, as the performance of the backing storage or network is increased the I/O throughput and IOPs Hammerspace can deliver also scales up and out linearly, virtually without limitation.

In Summary

Hammerspace provides a POSIX-compliant high-performance Parallel Global File System that enables global shared NAS access and file-granular control of data across silos that may include storage from any vendor on-premises, multi-site, multi-cloud, and multiple cloud regions.

2.3 Cross-Platform Data Services

Since the Parallel Global File System spans all underlying storage types across one or more locations, this enables Hammerspace to automate key data services across them all in a way that is non-disruptive to users and applications. Such services may be automatically applied at a file-granular level using Objective-based policies, triggered by one or more metadata variables, including custom user-created metadata tags.

This global control reduces the complexity for IT teams who manage siloed environments, and can reduce or eliminate the need for many point solutions that are typically required to overcome the existing barriers between incompatible storage vendor silos.

Below is a summary list of data services that may be applied across vendor storage types and locations.

2.3.1 - Data-in-place Metadata Assimilation

- This refers to the rapid harvest of file system metadata from existing storage systems in customer environments, which is done without the need to migrate the data to another storage location.
- Even very large environments can be rapidly assimilated, so users can begin browsing contents via the global file system within minutes, even as additional file metadata is being harvested in the background.
- This also enables customers to extend the life of existing storage investments, effectively extending them to include additional storage types, and the cloud rather than creating new silos of additional storage, or replacing them prematurely.

2.3.2 - Custom metadata tags with inheritance

- Custom metadata may be applied manually or automatically via script or other methods, and as such can be included as variables to trigger policy-based file actions.
- In addition, third-party applications can trigger workflows based upon these metadata variables with direct integration via the Hammerspace open API.
- When combined with file system metadata, such as the age of the file, or file size/type, these custom metadata tags enable fine granularity in setting of policy objectives for how data of different types, business value, or use cases are to be managed across shared environments.

2.3.3 - Audit of file system operations

- Ensuring that sensitive data is not stored inappropriately.

2.3.4 - Snapshots

Share-level snapshots across multiple storage types, which may be scheduled or immediate. Snapshots are consistent across all sites, and may be stored anywhere, including in the cloud.

- The key here is that Hammerspace eliminates the need to allocate high-cost storage space on primary arrays for local snapshots. Instead, Snapshots from any or all storage types may be routed to Cloud, or other lower cost storage locations.

Snapshots may be recovered in the following ways:

- Entire Snapshot: Roll back the share to that snapshot.

- Place the entire snapshot into a sub-directory of the same share that can be browsed like any other sub-directory.
- Roll back a single subdirectory from within the snapshot.

2.3.5 - File and Directory Clones

Where snapshots are at the share-level, clones are at the file and directory level. Snapshots are read-only, where clones are writable and can be moved. With snapshots, nothing is copied until it changes, whereas clones are immediate.

2.3.6 - Undelete

Undelete may be enabled for files as an additional layer of protection, with time-stamped versions kept for a specific time range in the snapshot.

2.3.7 - Versioning

Along with Undelete, versioning adds another level of protection to mitigate against and recover from unwanted changes or attacks that may occur in the interval between snapshots.

2.3.8 - File Mobility for Backup and DR

The reach of the Parallel Global File System enables the creation of Objectives to ensure that one or more instances of files are placed on different target storage types or locations based upon business requirements for those data.

- This can be set with Objective-based policies for data durability, for example, that may only apply to certain data types, or data associated with custom metadata for a specific project, department, or whatever user-created variable is required.

Since the Hammerspace Parallel Global File System spans all instances and locations in a multi-site configuration, centralized backup across them all plus DR are effectively combined in one.

- In other words, in the event of a site failure, DR is enabled without the need for explicit failover. The file system is the same across all sites, and users are able to see the secondary or tertiary instances of files at other locations seamlessly, without needing an emergency procedure to repoint applications or shares to a DR site.
- When the downed site is brought back online, Hammerspace automatically reconciles changes that occurred during the outage, and ensures all sites are in sync again without user intervention or interruption.

2.3.9 - WORM, Immutability, Compliance

Hammerspace can block I/O to specific tiers and lock instances of files for data protection and to comply with immutability requirements.

- When combined with Versioning, Undelete, Snapshots and Clones, these provide storage administrators with multiple lines of defense to help mitigate and rapidly recover from Ransomware or other attacks.

2.3.10 - Anti-Virus

Scanning files for anti-virus is supported for on-access and background scanning. ICAP protocol support on the anti-virus servers is required for anti-virus scanning functionality. Hammerspace will scan files on access and prevent file opens if a virus is detected.

2.3.11 - Object/Cloud archiving capabilities

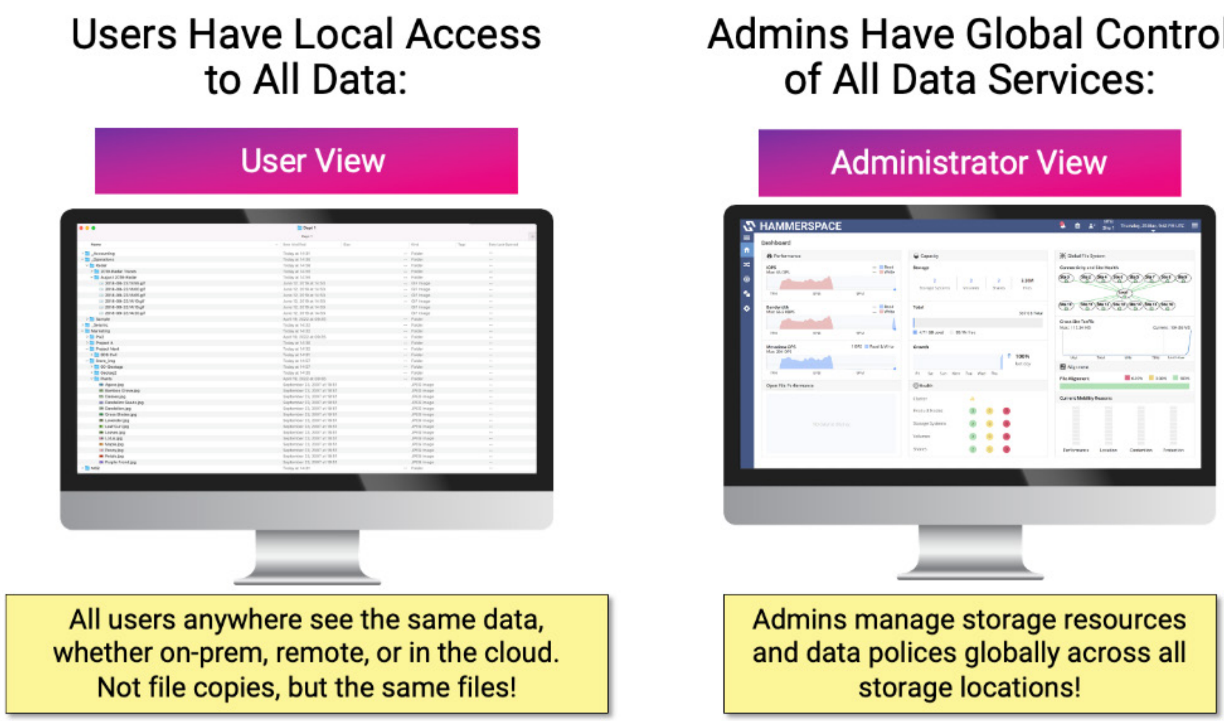
Additional services are supported in Object and Cloud storage, such that when data is moved to cloud/object storage it is globally deduplicated and losslessly compressed.

- Compression
- Dedupe
- Encryption

When global snapshots are stored in cloud/object storage this helps reduce the storage needed to protect that data.

2.4 - File-Granular Data Orchestration

Hammerspace enables non-disruptive file-granular data orchestration to automate and control the placement of file instances across any storage resource or location, transparently and non-disruptive to applications and user access.



Users and applications see their data at the same file share locations and in the folder hierarchy that they are accustomed to. But behind the scenes administrators can automate placement of those file instances to different locations or storage types when needed without user interruption.

2.4.1 - Objective-based policies for data placement & orchestration

Unlike the simple policies used by HSMs or point solutions that are typically one-dimensional commands with limited options, Hammerspace enables administrators to create comprehensive service-level objectives that may be finely-tuned regular expressions based upon business logic.

Objectives can be set to accommodate multiple use cases, and may apply to all or intelligently selected subsets of the data across various storage types. They may be applied conditionally to determine at a granular level how files and directories are to be managed.

In addition, multiple Objectives may be applied to shares, directories, and files, which the system will monitor for alignment. This enables global control of a wide range of business requirements across all digital assets and storage resources in multi-siloed environments.

For example, Objectives may be set for durability of certain data datasets, or data sovereignty, or DR requirements, or other requirements that may be triggered by any combination of metadata variables.

instances across any storage resource or location, transparently and non-disruptive to applications and user access.

- These triggers may simply be file system metadata such as the age of the file, or access time, file type, or location.
- But the variables may also include custom metadata, such as tags linking the data to a cost center or project, or the actual loaded cost of the storage platform housing the data.

2.4.1.1 - Example: Data Profiler

Data Profiler is a capability within Hammerspace to use when creating a tiering Objective, enabling the user to determine if savings can be achieved by moving data to different tiers or even into the cloud.

The Data Profiler gives the user a 'Before' and 'After' view of costs to help determine the business value of tiering before actually pulling the trigger to do so.

The following data points can be seen in the data profiler:

- Total cost of the analyzed storage environment
- Cost savings vs. previous or reference configuration
- Cost per storage tier
- Amount of capacity per tier
- Number of files per tier

The Data Profiler can run on data managed by Hammerspace and it does not affect the data or client data access. Data can also be imported from other NAS storage systems by leveraging Read-Only volume assimilation to build a virtual share.

2.4.1.2 - Example: Custom Metadata Triggers

Unlike the simple policies used by HSMs or point solutions that are typically one-dimensional commands with limited options, Hammerspace enables administrators to create comprehensive service-level objectives to manage data and storage resources that may befinely-tuned regular expressions based upon business logic.

For example:

- An Objective can be set so that if a given file or directory is tagged with a “Project ID”, or “Dept. Name”, or other custom variable, the system would automatically ensure that DR instances are pushed to the cloud, or a specified durability for those data is established and ensured, etc.
- All files in that directory with those metadata variables would be automatically aligned to the conditions of the Objective, and the Administrator can monitor this alignment across all storage types and instances.
- Such Objectives may trigger data movement, or replication, or any number of background actions, but as always without interruption to users.

Objectives can be applied intelligently to data, and are dynamically inherited at the directory level. This is critical to ensuring that Administrators no longer have to rely on manual processes, or on users to remember to do manual actions.

Hammerspace monitors storage usage continuously against the Objectives that have been defined for the environment.

- Hammerspace allows administrators and data owners to see the data alignment of tiers and objectives to help them understand whether the objectives set for data can be met with the current resources and configuration.
- Administrators are able to determine in real time the implementation of objectives, and whether files are in alignment with the actions defined by the objectives across the entire system.

And as with all file movement within Hammerspace, Objectives may be applied transparently as background operations. Even files that are open and being actively worked on may be moved in the background to otherwise incompatible storage without interruption to users or applications.

2.4.1.3 - Creating Objectives:

Hammerspace Objectives may be implemented programmatically in multiple ways including:

- via Hammerspace GUI
- via Hammerspace CLI
- via the Hammerspace REST API
- via Hammerscript, a query language based upon Excel and Visual Basic.

2.4.1.4 - Example Objectives:

- Tiering - locally and/or to Cloud
- Data Profiler
- File copy management
- Migration
- Replication
- Workflow automation
- Leveraging custom metadata
- Storage resource monitoring
- Data Durability
- Availability
- Data protection
- Virus Scan
- Archive
- etc.

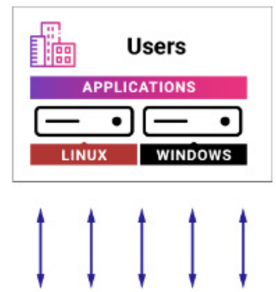
2.5 - Multi-Vendor Storage Support

Hammerspace is a software-defined solution that is designed to support any storage type, including NVMe, SSD, hard disks, and which can include block, file, object and cloud storage platforms from virtually any vendor. No special integration is required for storage that uses standard protocols.

Hammerspace supports all the major Cloud vendors, including AWS, Azure, GCP, Seagate Lyve, Wasabi, Snowflake, and more.

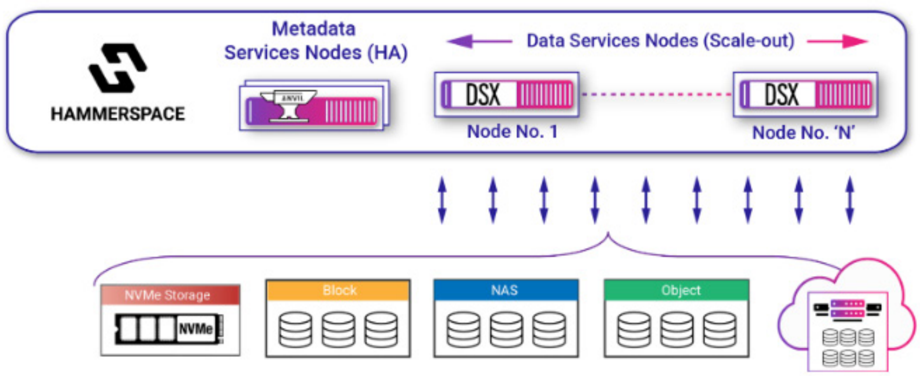
3. Hammerspace Building Blocks

3.1 Hammerspace Node Types:



Hammerspace software is deployed in a scale-out architecture in a cluster for each site that comprises two node types that work together as a single system;

- **Anvil metadata services nodes,** which house the metadata control plane, and drive the intelligence of the system;
  - No file I/O passes through the Anvil nodes.
  - Anvils are typically deployed as an HA pair in production.



- **DSX, or data services nodes. (Data Service eXtensions)**
  - DSX nodes handle all I/O operations, replication, data movement, etc., and are designed to scale out when needed to over 60 nodes in a cluster to accommodate any level of performance requirements.
  - In this way, Hammerspace clusters can grow or contract as needed, to take advantage of and parallelize performance of the underlying storage resources and networking.
    - The ability for Hammerspace to parallelize front-side and store-side I/O across the network enhances this cross-platform capability.
    - When additional performance is needed, higher-speed networking and storage may be brought online, and additional DSX nodes provisioned at any time, non-disruptively.
  - In the performance section below, the direct results of this ability to scale-out and scale-up linearly are demonstrated.

3.1.1 - Hardware Agnostic

As a software-defined platform, Hammerspace is hardware agnostic and may be deployed on bare-metal servers, VMs, and in Cloud instances. It is loaded from a single installer that handles both node types. The installer includes all software components, including LinuxOS, with no dependencies on external software or third-party products.

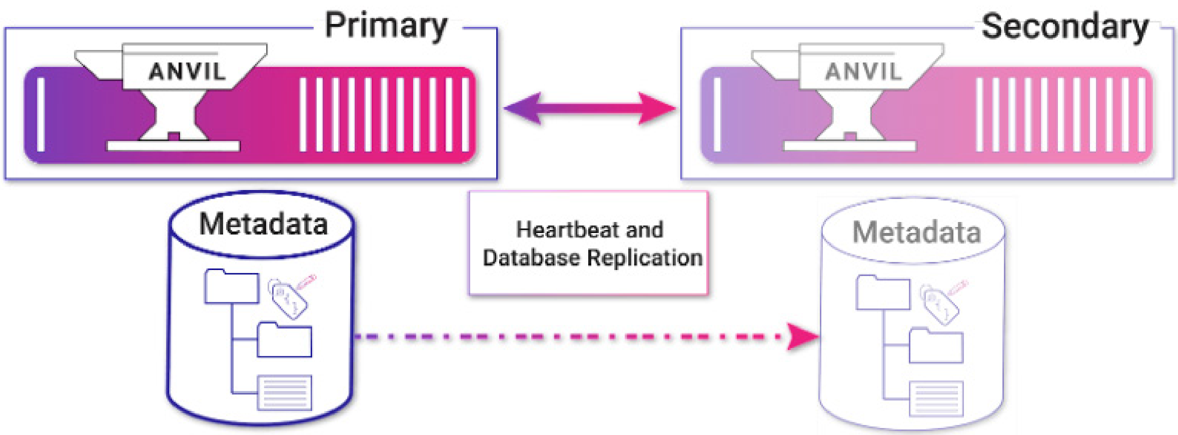
There is no one-size-fits-all specification for the server requirements for Anvil or DSX nodes, which means the system can be tuned to the specific load requirements of the customer's use cases. This enables system designs to be dialed in to minimize unnecessary infrastructure expenses. At any time the system can expand to higher-performing infrastructure without disrupting user or application access. Some customers have begun with initial cloud-based implementation to rapidly provision a remote office, for example, which they later convert to on-premises servers. User access was seamless, and never impacted by this change in infrastructure.

- **Key points: Single installer for the entire Hammerspace system, including both Anvil and DSX node types.**
  - ISO for interactive deployments
  - OVA for automated deployments
  - Ansible recipes for automated cloud deployments
  - CSI driver for Kubernetes and other containers, to enable persistent storage for both block and file use cases.

3.2 - Anvil - Metadata Nodes

As noted above, a key attribute of the Hammerspace system is the ability to elevate the file system above the storage infrastructure, effectively creating the metadata control plane that can span all the underlying storage hardware. I/O pathways do not go through the Anvil nodes.

The primary purpose of the Anvil node is to manage all of the metadata within the system and control all of the I/O and other actions that are performed by the DSX nodes, noted below. Although an Anvil node may be deployed as a stand-alone for lab use, in production they will typically be deployed as an HA pair. The secondary Anvil constantly monitors the primary, and replicates metadata automatically.



3.3 - DSX - Data Services Nodes

The DSX Nodes are the workhorses of the Hammerspace platform, providing scale-out engines that connect data and storage to users and applications, performing all I/O operations, replication, data movement, and so on.

DSX nodes are controlled by the Anvil metadata nodes, and operate together in concert as a single parallelized scale-out system, with linearly scalable performance. The DSX nodes are stateless, and may be configured with fixed and floating IP addresses with automatic fail-over.

Front-side NFS and SMB access go through the DSX nodes via a single mount point that spans across the entire cluster. Traffic is load-balanced across them all, and cross-protocol file locking is provided within the cluster.

DSX Nodes connect to backing storage platforms of multiple types via standard file or object protocols. In addition, DSX nodes can be configured with direct-attached block storage of any type. This capability enables flexibility for customers who can add additional raw storage capacity directly to the system, to complement existing commercial NAS or Object storage infrastructure, or for specialized high-performance use cases. These may include:

- Internal SSD, NVMe, HDD, RAID, JBOD
- VMDK, VHD and other virtual disks for VMs
- Optional striping and mirroring for local volumes

Such storage can be hot-plugged into or out of the DSX nodes, and become a seamless part of the storage fabric.

3.3.1 - Live Data Mobility

Data movement between storage types is fully automated, and transparent to user access. This includes balancing between DSX nodes themselves, or other background data placement actions between DSX and NAS systems, or NAS-to-NAS.

File-to-object data mobility is also ensured over HTTPS using standard protocols, including S3, GCP, Azure Blob, Seagate Lyve, etc. For use cases moving files to object/cloud storage, Hammerspace will automatically apply lossless compression and deduplication, both to optimize network traffic, but also to conserve storage space downstream. Data sent to object and/or cloud storage may also be encrypted, with customer-managed KMS.

Data movement tasks include fully-automated actions based upon Objective-based policies, as noted above, workflows, or user actions. DSX nodes can run thousands of concurrent jobs. And as with other performance characteristics, increasing the number of nodes or the performance of the nodes themselves will linearly expand the capabilities of the system to accommodate increased performance requirements over time as needed.

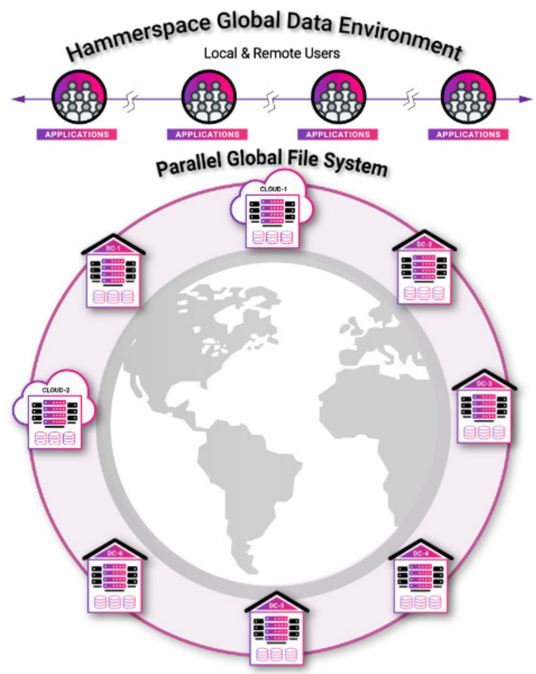
3.3.2 - Scalability

DSX Nodes may scale out at any time to dynamically increase performance to even extreme levels, for IOPS, throughput, and to accommodate any number of users. Refer to the performance metrics in the section below to see empirical results of this scale-out capability.

Currently Hammerspace supports a maximum of 64 DSX nodes per cluster, although there is no hard limit to that number.

3.4 - Global Data Environment - Bridging Multiple Sites

Hammerspace's Parallel Global File System not only bridges on-premises data silos and extends them to the Cloud, but it can also expand globally across multiple terrestrial or cloud-based sites, or any combination of both, to provide a Global Data Environment that can provide seamless file access across them all to users anywhere.



Current customer deployments include a high-performance production environment that spans nine data centers plus the cloud in a single online Parallel Global File System.

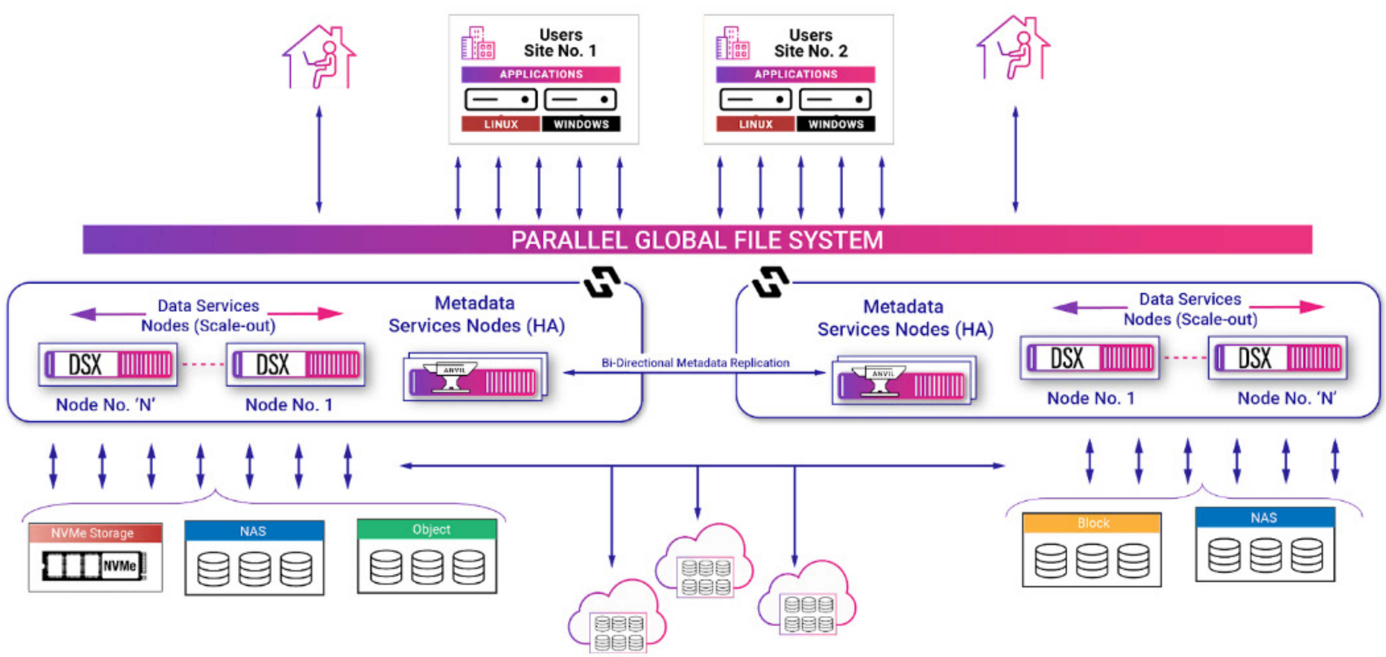
A distributed system can begin with two or more Hammerspace clusters, which may export one or more shares. Although Hammerspace is currently qualified to support up to eight sites per share, there is no hard limit to the number.

Each site can include a different mix of storage resources, and different hardware configurations for the Hammerspace nodes. A minimum of one Anvil and one DSX node are needed per site, although an HA pair of Anvil metadata nodes are recommended for production environments, plus at least two DSX nodes for resilience.

As noted above in the discussion about the Parallel Global file System, all users and applications at all sites have a consistent view to the same file system. All metadata is replicated across all sites, and users may access files globally, regardless of where they are located, or where the file instances are located.

Read/write access is supported for all data across all sites, with permissions/ACLs respected across them all globally.

Unless required for local processing, file volumes and file instances do not need to move to local storage for users to browse the file system. A file instance may move locally based upon a user action, such as opening a file for modification. This is automatic, and immediate.



Additionally, an Objective may be created that will automatically and transparently stage selected subsets of data volumes to local storage for high-performance workloads. Of critical importance here is that only the subset of files actually needed for the job need to move to local instances. This is significantly different from other solutions that require actual file copies and entire volumes to be migrated to remote sites for local access.

As within a single datacenter, all users are seeing the same file system metadata across all sites, and are not having to deal with file copies. Changes that are made by remote users update the metadata across all sites automatically. And any other instances of the files are automatically conformed to those changes when accessed.

3.4.1 - Consistency

While Hammerspace supports cross-protocol file locking within a local cluster in a single data center, by design Hammerspace does not enforce global file locking across multiple sites may go off line for operational or other reasons. This is because trying to ensure live file consistency with multi-site file locking would create significant performance and other operational limitations.

Instead, Hammerspace relies on near-real-time file system metadata consistency across sites, with metadata synchronization happening every few seconds between Hammerspace clusters.

For Hammerspace customers operating with globally distributed workflows this capability has overcome significant real-world problems, and enabled them to dramatically expand their operations beyond the limitations of previous manual copy-based workflows or solutions.

User A creates a file in Site A, which is in a share that is visible to users in all remote locations. Perhaps User B at Site B opens that file to work on it, which will trigger a backend instantiation of that file to local storage at Site B.

If User B edits that file and saves it again, a second version of the file will be saved at Site B, but the global file system metadata is updated with that change instantly, and remains consistent across both sites. In other words, both User A and User B will see the same updated version of file metadata.

If User A opens the updated file, the Hammerspace references that global file metadata and serves the second version of that file, automatically updating the local instance at Site A.

All of this is completely transparent to users, operating automatically in the background.

This capability enables tremendous flexibility for follow-the-sun workflows, where Hammerspace customers can enable collaborative file operations for globally distributed resources. The near-real-time consistency of the global file system metadata, backstopped with eventual consistency of the file instantiations that is reinforced with file versioning, enables a high-performance global workflow that is resilient to network interruptions, or sites going offline.

In the case of a site going offline, if they are operational but disconnected from one or all of the other sites, Hammerspace at that site will keep track of all changes that occur locally. If any instantiations of the files users were accessing are available at a DR site, they will be able to continue working as before without interruption.

Later, when their local Hammerspace cluster rejoins the global system, the Global File System will automatically synchronize, to update all sites to current status.

4. Hammerspace Performance Testing

4.1 - Client-side Performance Examples

The goal of these tests is to show the I/O performance of Hammerspace in order to demonstrate scalability and throughput based upon different scale-out scenarios. These tests were run in AWS, with differing compute resources, networking configurations, and storage performance levels.

As mentioned previously, as a software-defined solution Hammerspace performance can scale linearly such that if increased IOPS or throughput are required, additional nodes or upgraded hardware may be added dynamically.

For these tests, we take advantage of modern hardware in the AWS environment, which includes:

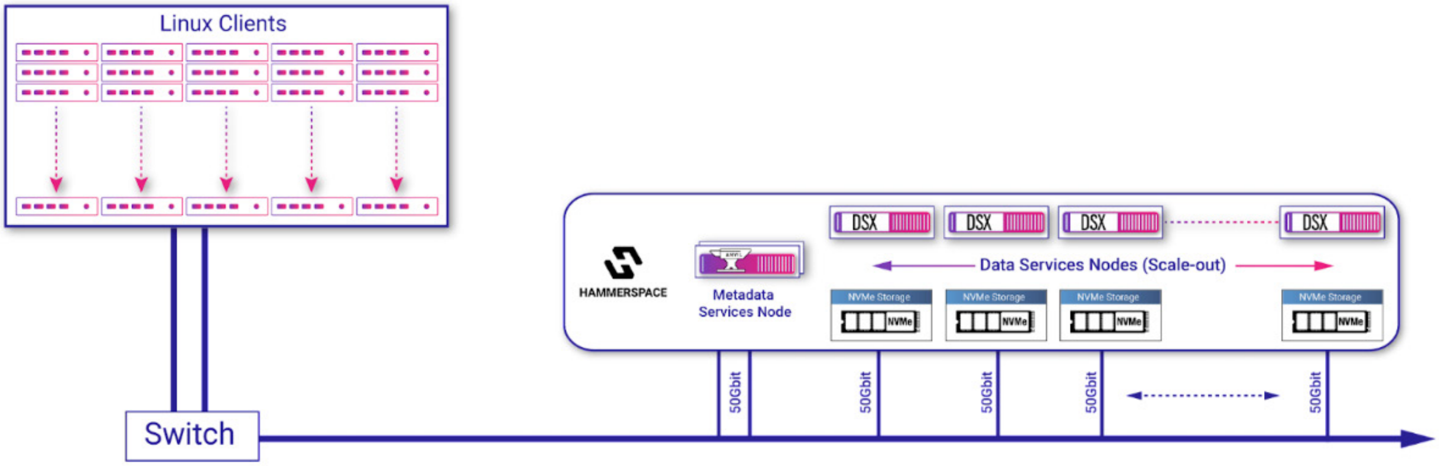
- I4i instance types
- M6i/M6id instance types
- NVMe ephemeral storage

4.1.1 - Test Workload characteristics

- Simulate M&E Render workload
- Predominantly read (90%) and write of results (10%)
- 32k IOsize -> drives high throughput needs and high IOPS needs
- 4k IOsize -> to demonstrate extreme IOPS
- Workload size exceeded available memory to ensure this matches as close as possible to real-world workflows with frequent disk access rather than reading from memory.

4.1.2 Software & Hardware

- Hammerspace 4.6.6 (GA)
- Flexible IO tester (FIO) 3.16
- CentOS Stream 9 – Clients
- AWS Instance types:
  - Anvil Metadata Nodes:
    - m6i.8xlarge (32 vCPU, 128 GB mem, 12.5 Gbit/s Network)
  - DSX Data Services Nodes: Count
    - M6id.metal – 50 Gbit/s throughput, 4 NVMe
    - M6id.16xlarge – 25 Gbit/s throughput, 2 NVMe
    - i4i.metal – with AWS Nitro SSD
  - Clients:
    - C5.large (2 vCPU, 4 GB mem, Up to 10 Gbit)
    - Counts include 16, 32, 48, and 192 clients
- Notes:
  - NVMe drive(s) using m6id instance types appears to be limited to approx. 1.1 GB/s (write)



4.2 - Summary of Results:

Hammerspace scales efficiently from small to large:

- The system successfully takes full advantage of available network bandwidth for throughput-dependent workloads.
- It successfully takes full advantage of the available performance of the backend storage for IOPS-dependent workloads.

Testing proves that scalability is linear, and fully utilizes the available networking and disk performance:

- With 16 DSX nodes, the configuration hit 1.17 Tbits/s with 32kb file sizes.
    - In testing for Raw IOPS with this configuration, the same test using small 4k files achieved 6.17m IOPS
- CPU utilization for the DSX nodes was around 50% for 32kb tests. For smaller 4k files, CPU utilization increased, as would be expected
    - But even with increased CPU utilization based upon the small file test, the load was within normal operating parameters of the DSX server configuration that was used.
- With a mix of six DSX nodes (different instance types) performance saturated each of the DSX nodes, achieving up to 250 Gbit/s throughput, which was the maximum throughput possible for the backend network and storage in this configuration.
    - This test proves that not all nodes needs to be the same type to take advantage of the scale-out nature of NFS 4.2
- Tests were designed to exceed available memory by at least 4x, forcing disk access to ensure the test match was as close as possible to real-world workflows.
- Hammerspace can scale up to 60 nodes, so if additional storage and networking were added, the performance would scale linearly to extreme levels.

Test No. 1 - Extreme IO & IOPS - Terabit I/O Architecture

Two test scenarios were run, to demonstrate:

- Test 1a = Raw throughput based upon 32k files
- Test 1b = Maximum IOPs with 4k small files

- **AWS Instance types**
  - Anvil: m6i.8xlarge (32 vCPU, 128 GB mem, 12.5 Gbit/s Network)
  - Count = Single instance

- **DSX:** i4i.metal (128 vCPU, 1,024 GB mem, 75 Gbit Network, 8x 3750 AWS Nitro SSD (40 Gbit EBS bandwidth on each instance)
  - Count = 16

- **Clients:** c5.large (2 vCPU, 4 GB mem, Up to 10 Gbit)
  - CentOS 9 Stream
  - Count = 192

- **Network:**
  - All testing was done in the same VPC and Availability Zone.
  - All NFS exports were mounted using NFS 4.2 with default mount options.

- **Test Parameters:**
  - Hammerspace 4.6.6 (GA)
  - Hammerspace prometheus exporters were manually installed to collect statistics
  - Flexible IO tester (FIO) 3.16
    - 90/10 R/W mix was used to simulate a Render-style workload.
    - IO pattern was RANDOM
    - 32 KB IO size was picked to be application friendly and considered a good mix to exercise both bandwidth and IOPS
  - Benchmark runtime was 3 minutes to avoid network throttling by AWS.
    - Longer spot testing was done and performance was sustained until network throttling kicked in due to the client size picked.

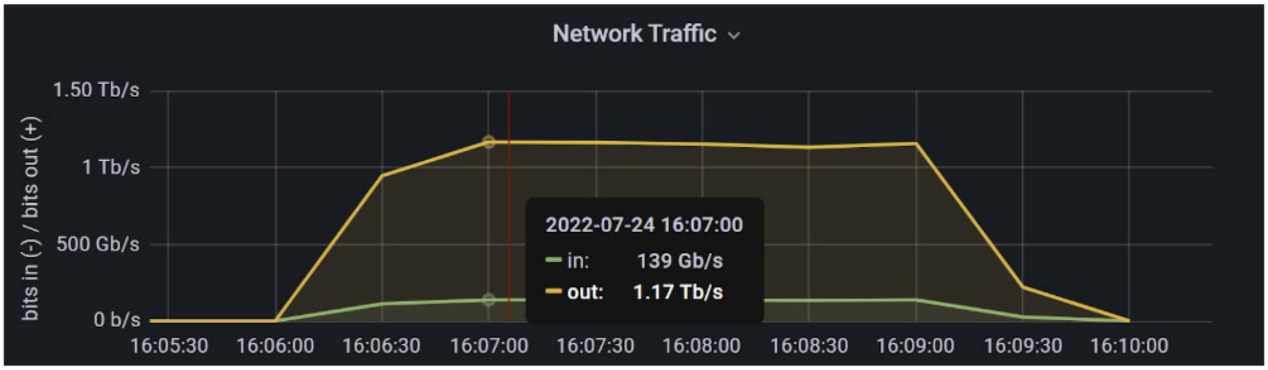
```
FIO Config
- [global]
- numjobs=16
- iodepth=1
- group_reporting
- direct=1
- name=file
- size=10g
- directory=/data/
- ioengine=libaio

[run1]
- rw=randrw
- bs=32k
- rwmixread=90
- runtime=180
- time_based
```

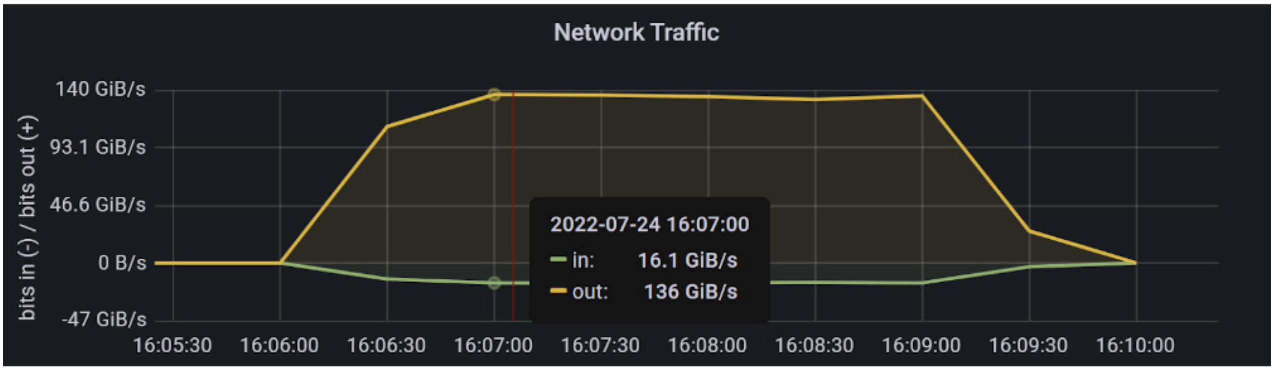
Test 1a - Maximum throughput (network view from Grafana) for 32k 90/10 R/W RANDOM

- **Reported by FIO**
  - Read: IOPS = 4,196k, BW=128Gi (138G)(22.5TiB/180004msec)
  - Write: IOPS = 467k, BW=14.3Gi (15.3G)(2565GiB/180004msec); 0 zone resets

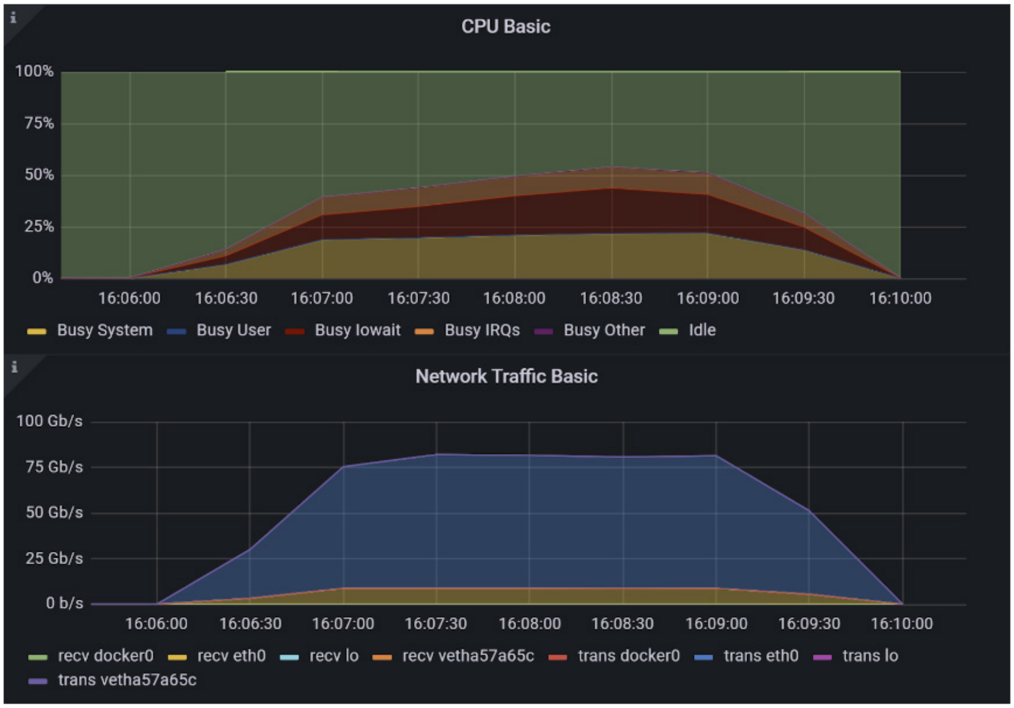
- **Network throughput in Grafana**
    - 1.17 Tbit/s , 136 GiB/s
- **Read/Write network throughput**



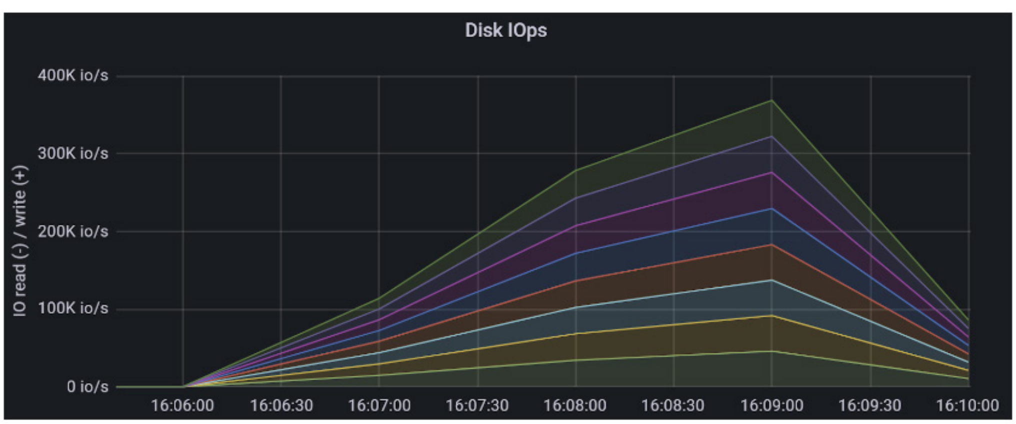
- **Read/Write network throughput using GiB units:**



- **Individual DSX node stats:**
  - 16 DSX Nodes were used. This graph shows utilization of one node, but all nodes showed very similar stats.
  - Nearly 50% idle CPU during larger IO (32kb) test.



  - Each DSX node had a lot of memory, but the dataset exceeded the available memory in order to demonstrate a more real-world workflow with significant disk I/O.

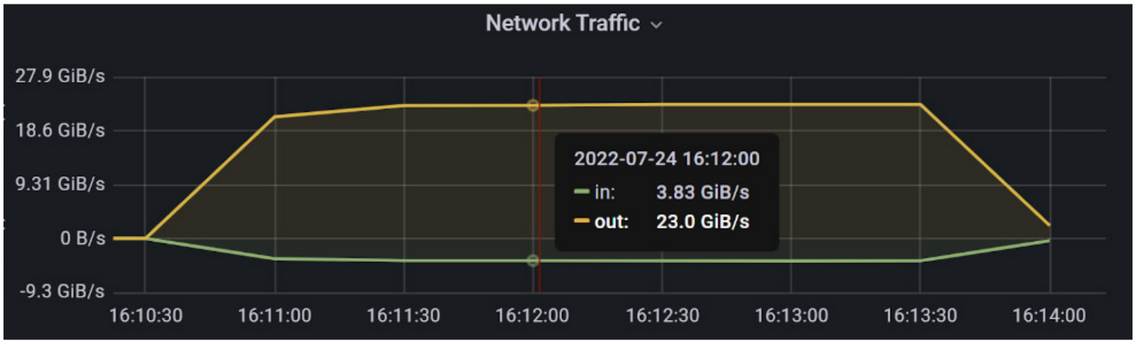


Test 1b - Maximum IOPS when testing 4k IOPS 90/10 R/W

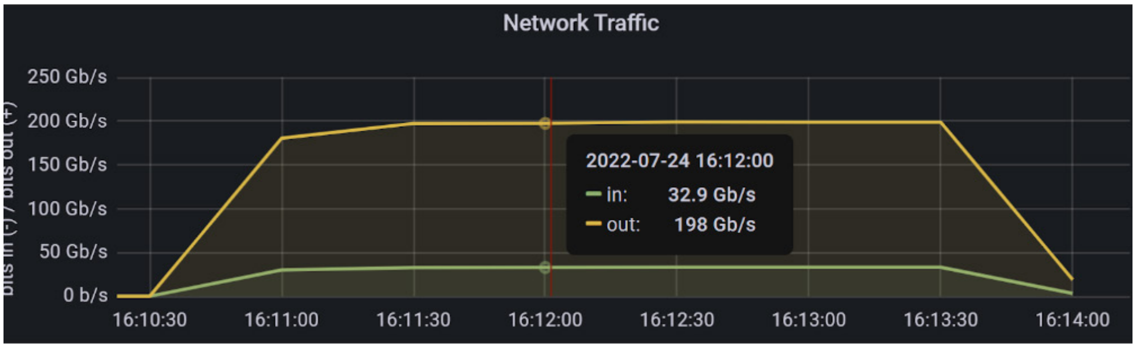
- **Reported by FIO**
    - Read: IOPS = 5,556k, BW=21.2Gi (22.8G)(3815GiB/180003msec)
    - Write: IOPS = IOPS=619k, BW=2418Mi (2536M)(425GiB/180003msec); 0 zone resets
- **Test Results: 6.17 M IOPS in total.**

Network throughput in Grafana

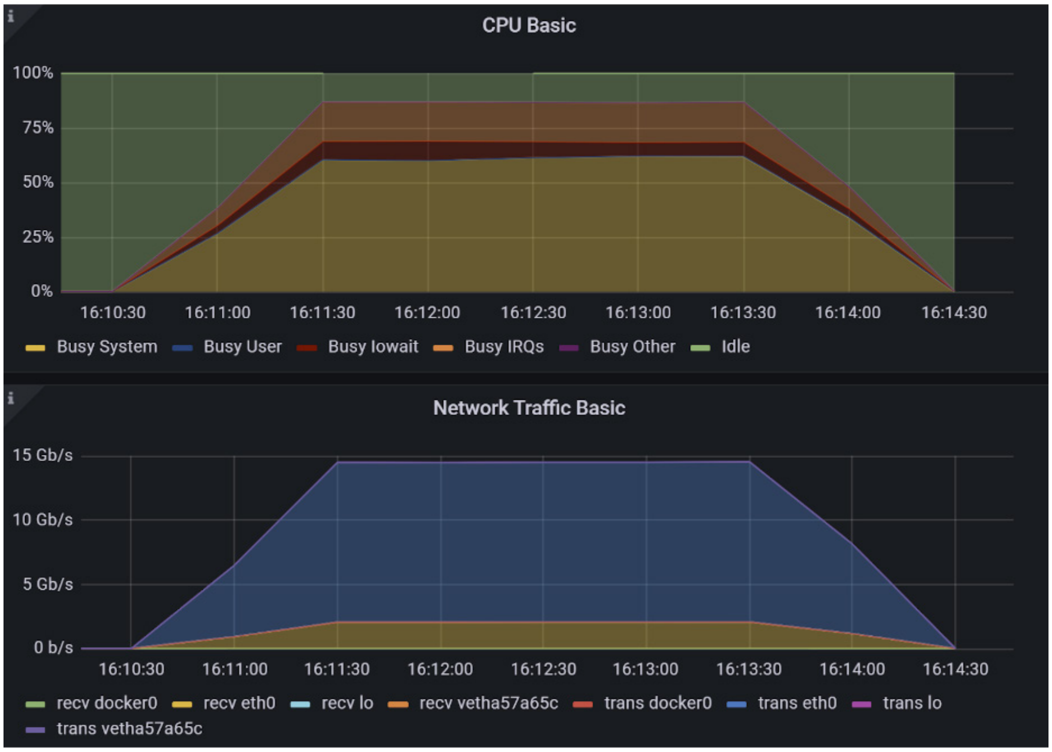
- **Read/Write network throughput:**



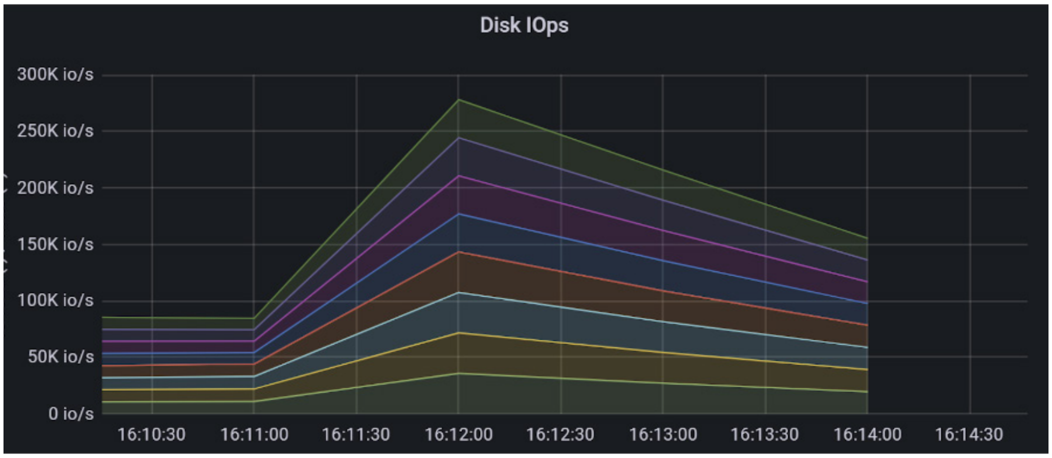
- **Read/Write network throughput using GiB units:**



CPU and network usage for one of the DSX nodes during the 4k IOSIZE test.



Read IOPS graph:



Test 2 - Scale testing - Ranging from 16-42 Clients

The goal of this test is to demonstrate the linear scalability of the system. By adding more DSX nodes, the system will linearly increase in throughput performance.

Conclusions Demonstrated below:

- **Hammerspace can scale efficiently from small to large:**
  - Saturating the network for throughput-dependent workloads
  - And saturating available performance of the backend storage for IOPS-dependent workloads
  - Scale can be achieved using different types of nodes
- **Testing proves that with only six nodes, performance up to 250 Gbit/s can be reached.**
  - Hammerspace can scale up to 60 nodes.
  - The nodes don't need to be identical as shown in the test, two different DSX node types were used
- **By utilizing cost-effective, new, m6id instance types with NVMe, the best performance/$ can be reached as compared with other storage choices**
  - Approx. equivalent config without NVMe (m6i + IO2 Block Express storage) is over 4x more expensive per month to reach the same level performance
- **Hammerspace is the only Enterprise SDS vendor that can safely include ephemeral NVMe in the primary file system. File-granular data management enables separation of reads and writes when needed..**

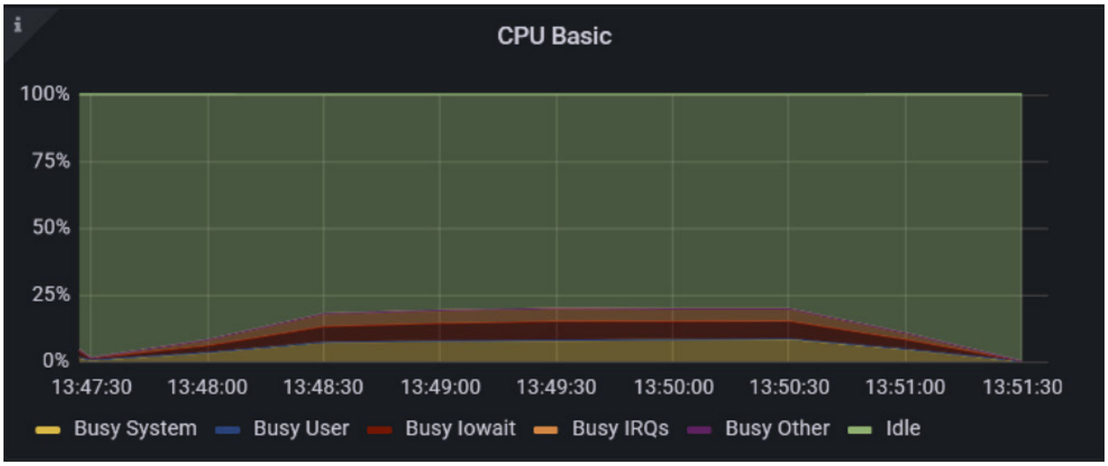
Test 2 Architecture

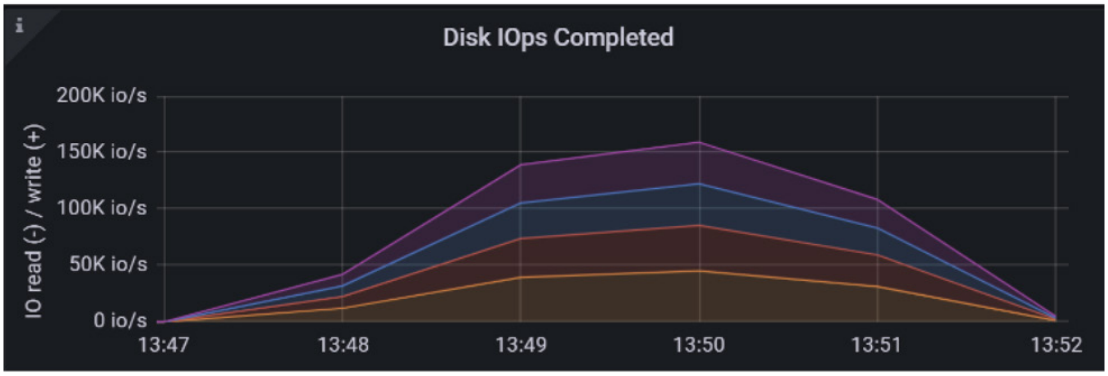
- Up to 48 clients
- CentOS Stream 9
- C5.large instances (2 CPU, 4 GB, Up to 10 Gbit/s)

Test No. 2a: 48 Clients, 6 DSX nodes (4 m6id.metal, 2 m6id.16xlarge)

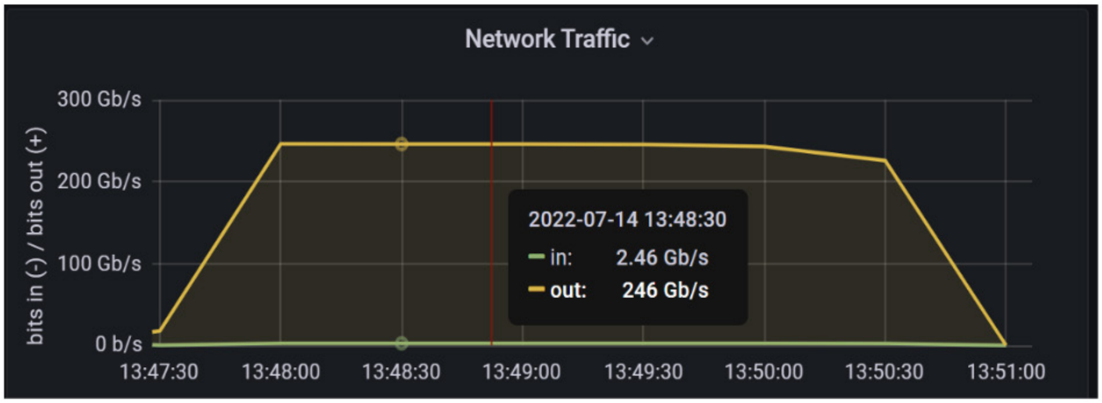
- **Summary**
  - Total Provisioned network: 250 Gbit/s
  - Utilized network: 248.5 Gbit/s
  - Client IOPS (90/10 R/W mix): 1.2 million (1.08m read, 120k write)
  - Nearly 80% IDLE CPU per DSX node

- **Per DSX Stats:**



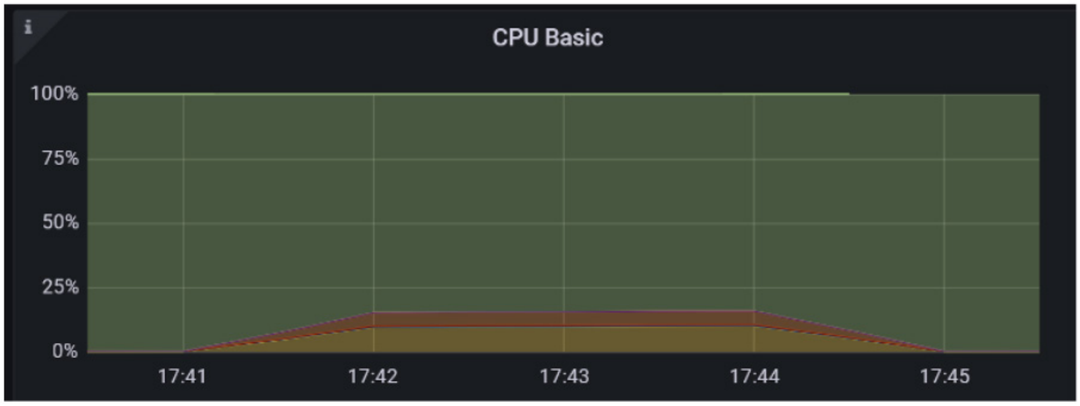


- **Total Network Throughput**

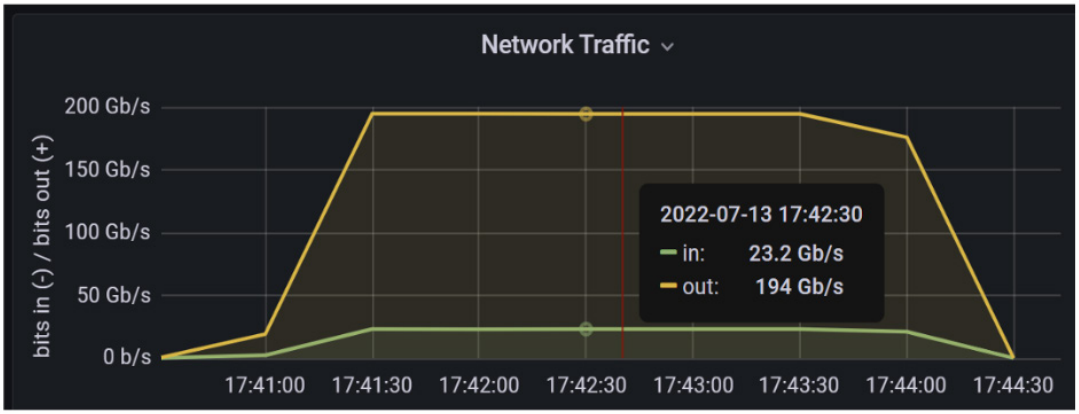


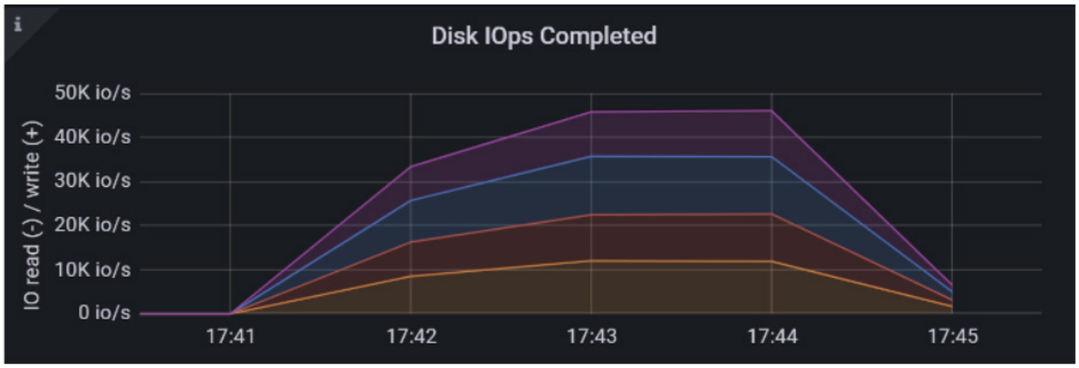
Test No. 2b: 32 Clients, 4 DSX nodes (4 m6id.metal)

- **Summary**
  - Provisioned network: 200 Gbit/s
  - Utilized network: 194 Gbit/s
- **Per DSX Stats:**



- **Total Network Throughput:**

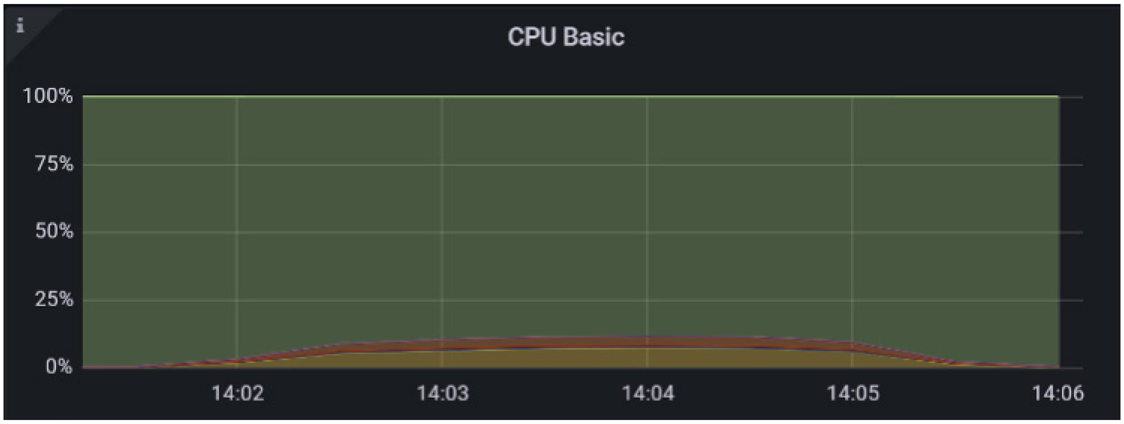


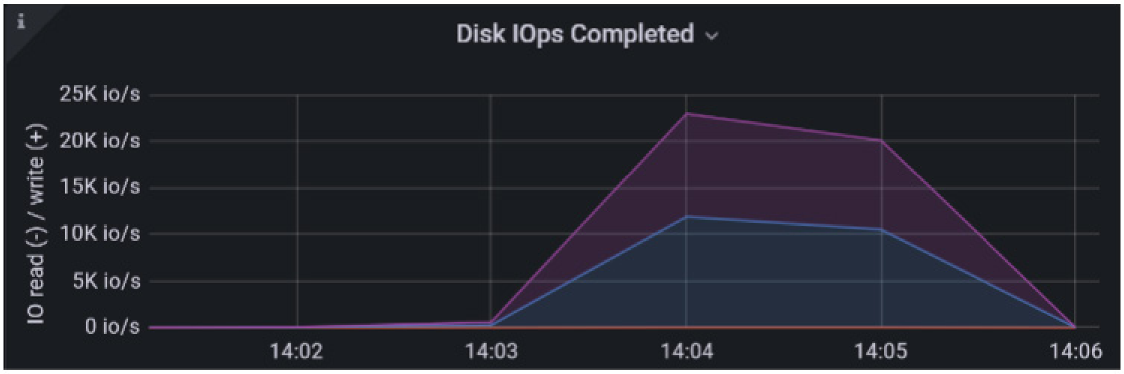


Test No. 2c: 16 Clients, 4 DSX nodes (m6id.16xlarge)

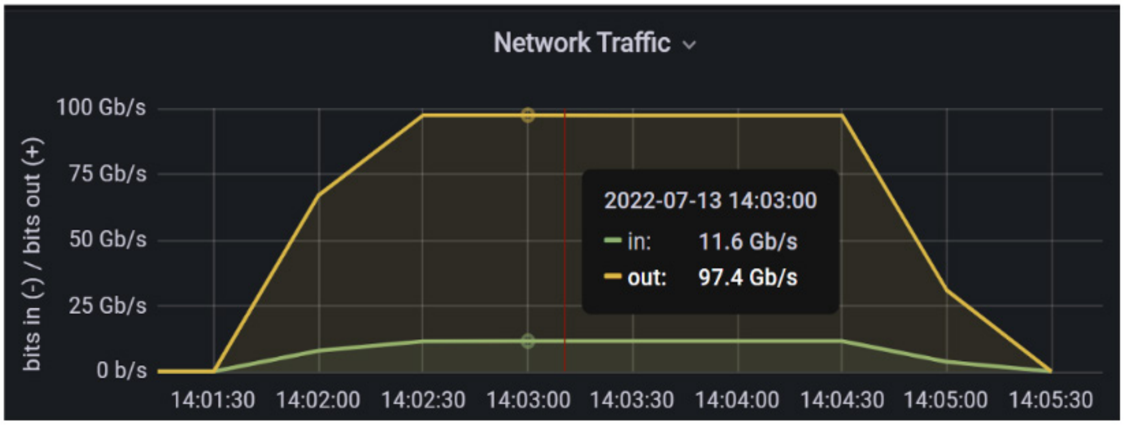
- **Summary**
  - Provisioned network: 100 Gbit/s
  - Utilized network: 97.4 Gbit/s
  - Client IOPS (90/10 R/W mix): 392k (353k read, 39k write)
  - >80% IDLE CPU per DSX node

- **Per DSX Stats:**





- **Total Network Throughput:**

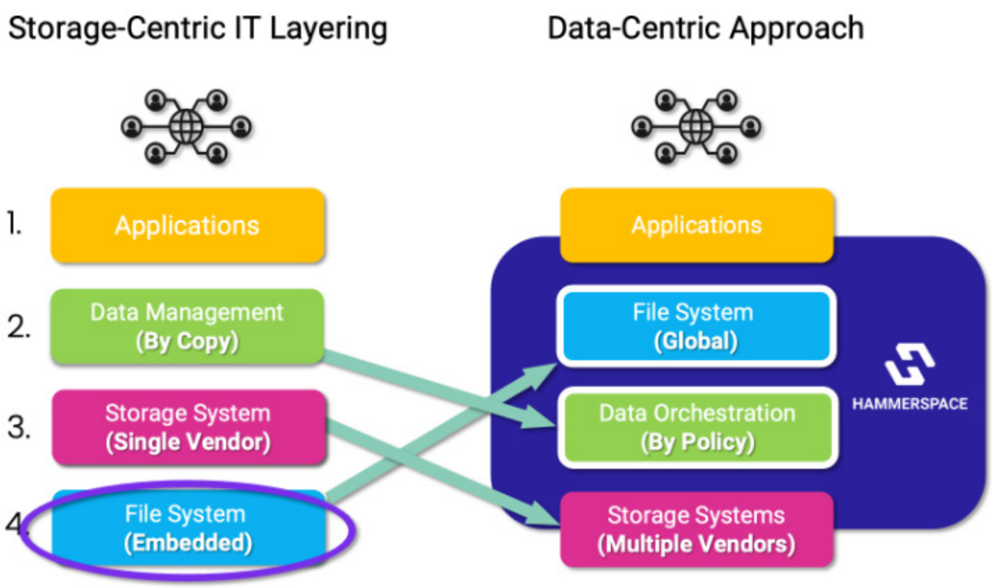


5. In Summary

As noted throughout this White Paper, Hammerspace has been designed from the ground up to solve the problems caused by fragmentation of data across silos in the data center, and increasingly across distributed systems that may span multiple data centers and the cloud.

The transformation into the Cloud era has accelerated decentralization of the enterprise across most verticals, bringing into focus the impact of file system fragmentation at the storage layer. And the problem has been additive:

- To be competitive as well as manage through the changes caused by the pandemic, companies needed to accommodate an increasingly distributed workforce. Storage and data access needed to be accessible and distributed as well, plus support performant workloads at a distance.
- And then to be agile in a dynamic environment where fixed infrastructure may be difficult to expand due to supply chain issues, the ability for companies to rapidly burst to Cloud compute and storage resources has become essential to their survival.
- At the same time, enterprises needed to bridge their existing infrastructure with these new distributed resources in a way that was cost effective, and could promote greater productivity.



Hammerspace has spent years, and done the heavy lifting needed to completely reimagine from first principles the way file systems need to work in a decentralized environment. It has done so to enable the promise of working seamlessly with the cloud and across multiple locations to be fully realized, and in so doing has addressed these fundamental issues.

In the 1990s network-attached storage solutions lifted the file system out of the individual PC operating system, eliminating the need to shuffle physical copies of the files between users in an office. With NAS systems, all users could share access to all files on their local network.

Today, Hammerspace is able to provide customers the same benefit of shared global access across multi-vendor storage silos, locations, and cloud resources. With Hammerspace, customers no longer need to shuffle file copies between vendor silos. Instead they can reduce or eliminate proliferation of point solutions and other complex workarounds to bridging the silo gaps within the data center, and across multiple sites, to support use cases that were previously impossible.

Because the Hammerspace file system is global, and all users anywhere are accessing their data as though it were on a shared local NAS, so that the need to manage data by copying it from place to place is eliminated.

To keep up with the reality of decentralization, a new global paradigm was necessary that effectively bridged the gaps between on-premises silos and Cloud. Such a solution required new technology and a revolutionary approach to lift the file system out of the infrastructure layer to enable the next wave of decentralization of business in a global economy. It is a revolution as important as when network-attached storage vendors lifted the file system out of the operating system in the 90s.

This is the Hammerspace innovation.