Hammerspace Assimilation

SOLUTION BRIEF

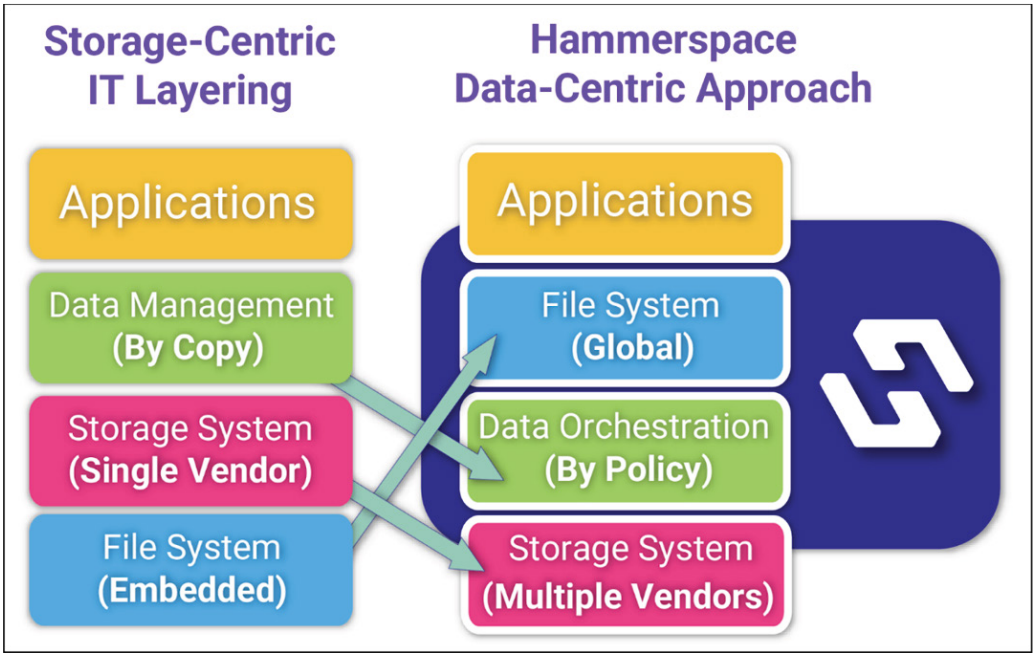
Introduction

Most organizations already have more storage than they realize. What they lack is a practical way to unify data stored across different storage types, apply consistent policies, and do so without disrupting users or applications. File systems are tightly bound to individual storage platforms, creating operational silos that are difficult to migrate, modernize, or retire. As a result, routine infrastructure changes often turn into complex, high-risk data migration projects.

This challenge has become more visible as organizations adopt AI and analytics workflows that require fast, shared access to data distributed across sites, tiers, and environments. Copying data into new repositories to support these workloads is expensive, slow, and reinforces the siloed architectures teams are trying to escape.

Hammerspace addresses this problem by elevating the file system out of the storage layer, creating a cross-platform global file system that spans storage systems, tiers, sites and clouds. Silos are eliminated, data services are centralized, and data orchestration becomes non-disruptive.

In this way, Hammerspace enables an organization to transition from managing individual storage systems to managing data as a shared resource.



Assimilation is the process used to bring existing data into a Hammerspace global file system. It allows Hammerspace to unify access to data across incompatible storage silos without physically relocating it. This enables data placement, protection, and other data orchestration tasks to occur non-disruptively over time.

This Solution Brief is focused on the what and why of assimilation. Other documentation is available that provides more granular details of the how.

Metadata vs. Data

File systems contain two types of information: metadata and data. Metadata is what you see when browsing a file system, including filenames, permissions, timestamps and other attributes. Metadata also records where the underlying data resides on storage. Data, by contrast, is the actual file content viewed through applications, such as database records, spreadsheets, web content, etc. With very few exceptions, the volume of metadata in a filesystem is much smaller than the volume of data.

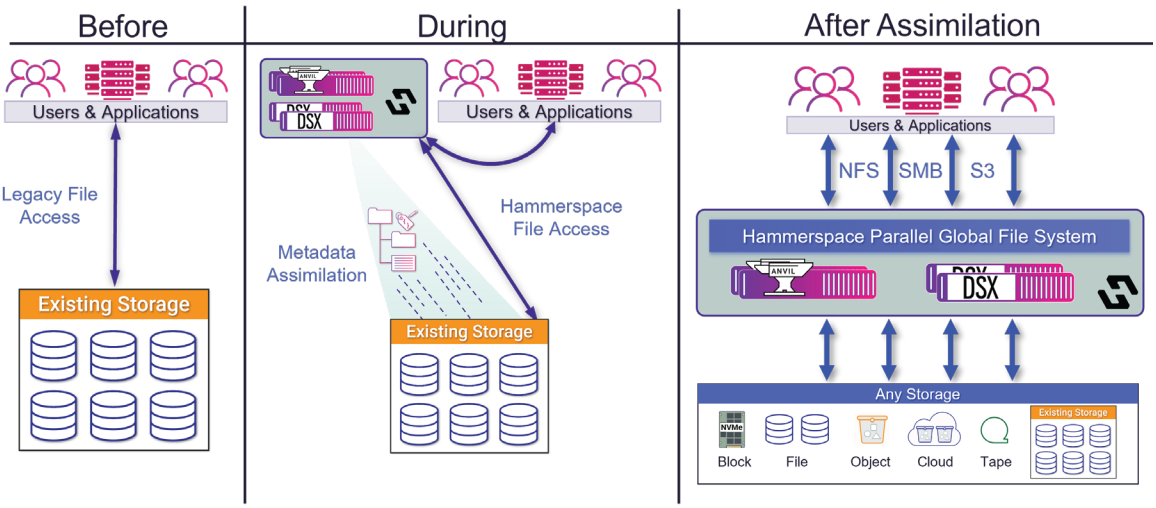
Assimilation involves ingesting the file and directory metadata from existing NAS file systems into the Hammerspace metadata database. Users and applications are then redirected to mount shares presented by Hammerspace rather than the original NAS shares.

Assimilation vs. Copy

Unlike traditional storage migrations, which involve a bulk copy of both metadata and data from one platform to another, Hammerspace assimilation is strictly a metadata operation. As a result, data access is restored almost immediately, even while assimilation continues in the background. This means that instead of taking weeks or months, petabytes can be assimilated in hours or days. Data remains in place, at least initially. More on that later.

Assimilation proceeds quickly because metadata is relatively small. Data access is restored almost immediately, even before the assimilation process is completed. As soon as the root directory metadata is ingested, typically in seconds or minutes, full access to all the source data is available through Hammerspace shares. Remaining metadata is assimilated in the background. If a client accesses a file whose metadata has not yet been assimilated, that metadata is assimilated on demand and the access will succeed.

This combination of background and on-demand metadata assimilation creates a user experience that customers have described as “magic.” As soon as assimilation starts, the system can be used as though it has completed. Rather than fencing out users, shutting down applications, or managing a complex copy-and-sync operation on live data over weeks or months, assimilation requires only enough downtime to remap users and applications from legacy NAS shares to the new Hammerspace shares.

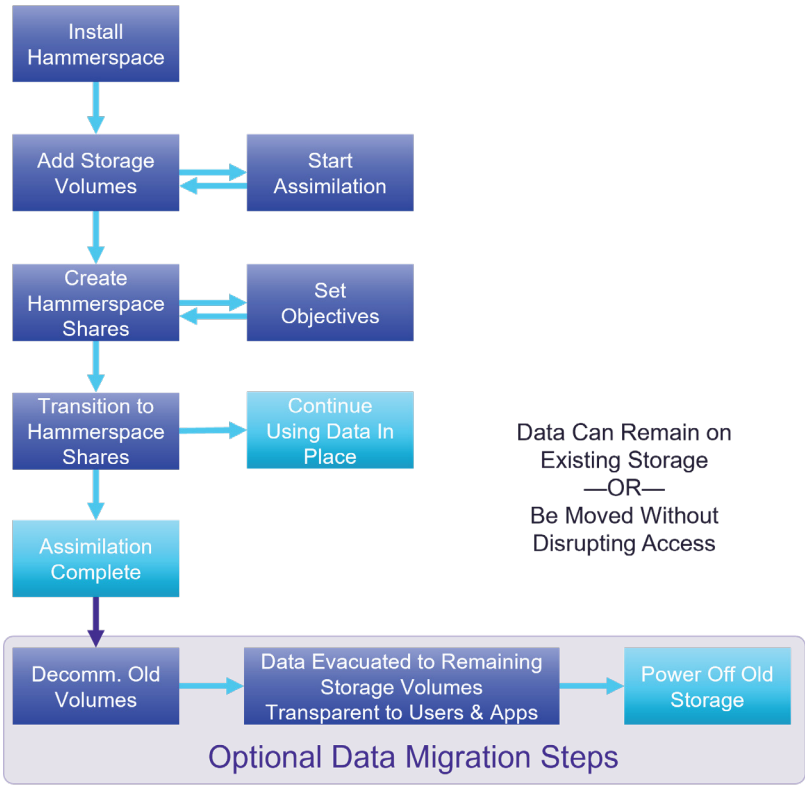


By bringing file system metadata from existing silos into the Hammerspace global file system, assimilation decouples the file system presentation from the underlying data storage location. As a result, when data migration is required, it is done transparently in the background, even for data that is in use. Copying data still takes time, but users and applications are not interrupted while it happens.

Assimilation Process

Assimilation is part of the broader process of bringing existing storage under Hammerspace management. Existing NAS shares are first added as Hammerspace storage volumes, at which point metadata assimilation can be enabled. Hammerspace shares are then created and presented to clients as access points into the global file system, enabling consistent access via multiple protocols regardless of the capabilities of the underlying storage.

Objective-based policies applied to these shares determine data placement, protection, security, and other behavior, independent of where the data physically resides.



Once assimilation is started, clients can be immediately redirected from the legacy NAS exports to the new Hammerspace shares, with continued access to their data as before. The data stays in place; only the mount point changes. No client software is required, and no other changes are needed to user systems.

When it is time to sunset the existing storage systems, whether immediately or years in the future, the associated storage volumes are decommissioned in Hammerspace as a background operation. This process migrates all data from the source NAS to other storage systems under Hammerspace management without interrupting users or applications, even for data that is actively in use.

For large volumes of data, the physical migration can take a long time, limited by storage and network performance. In that respect, it is no different than any other data migration. What is different with Hammerspace is that all the data movement occurs transparently. Because Hammerspace presents a global file system spanning both the old and new storage, mount points remain consistent even as data moves between systems. Users and applications continue to see files and objects where they expect them, with uninterrupted access throughout the process.

Once the data has been migrated, the original storage system can be retired.

Types of Assimilation

There are two types of assimilation, read-only and read-write. The appropriate choice depends on whether the source storage is being retired, or will continue to be used.

Read-only assimilation is used when cutover from existing to new storage will be immediate. This is common when the source system is full, failing, or out of support. After a read-only assimilation, all new and modified data is written to other storage within the Hammerspace environment. The source storage and its existing data remain unchanged and are treated as read-only until the data is fully migrated, and the system is retired.

Read-write assimilation is used when existing storage will remain in service and continue to store new data. After a read-write assimilation, new and modified data is written into a Hammerspace-managed data structure on the source storage, while existing data remains in place and unmodified. Any available free space on the system becomes part of the global capacity pool managed by Hammerspace.

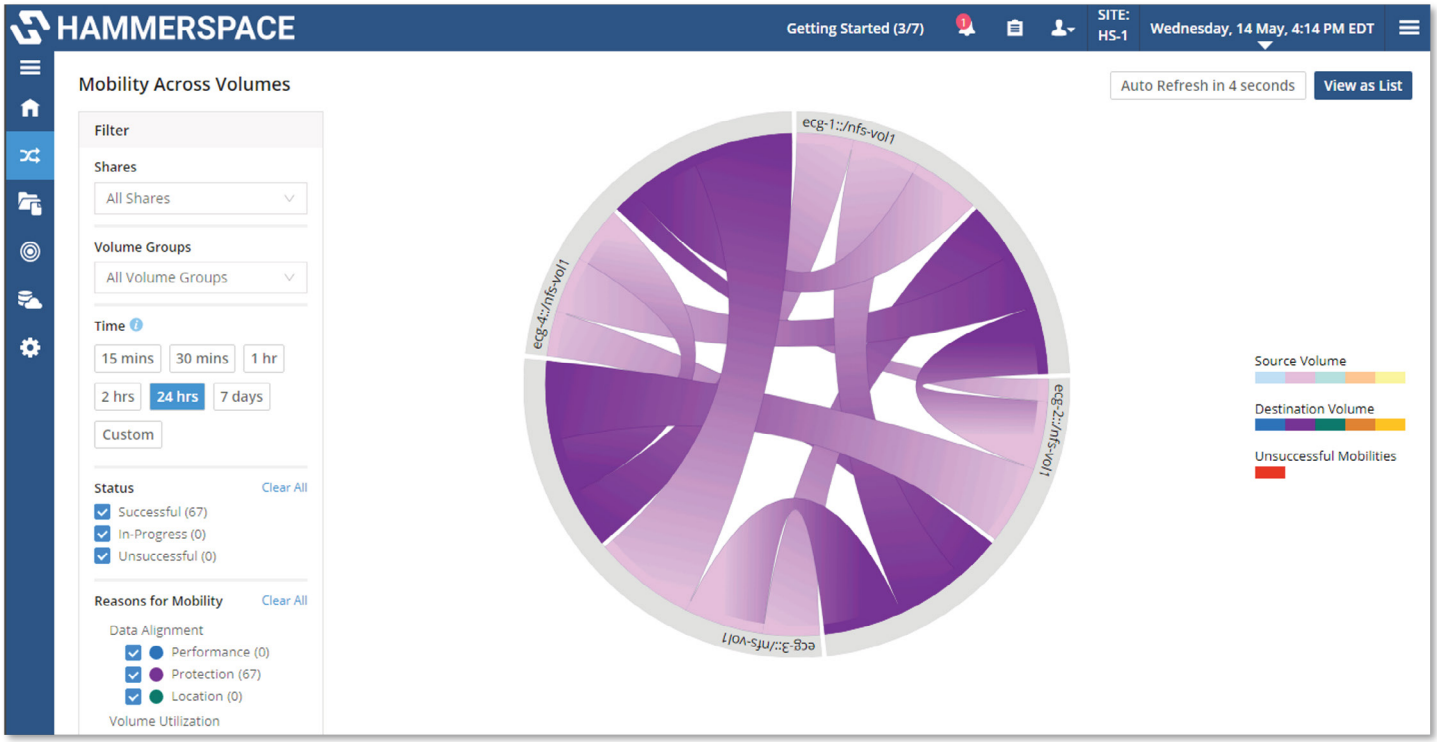
Assimilation can be performed against a live filesystem or from a snapshot, although snapshot-based assimilation is typically only used for testing or proof-of-concept work.

Data Orchestration

Data migration is one example of a broader capability that Hammerspace refers to as data orchestration: the intelligent movement of data across available storage platforms. While Hammerspace supports a wide range of data orchestration scenarios, simple data migration as described above is often the first practical expression of this capability that customers encounter.

The policy-based objectives used to define data orchestration can serve many purposes, including tiering older data off to cloud or object storage, creating multiple file instances for resilience, staging datasets from lower- to higher-performance storage for processing, and more. Regardless of how or why data moves, these operations remain transparent to users and applications.

Administrators may easily monitor data mobility using the Hammerspace GUI, Grafana dashboards, or custom tooling built on the REST API.



Conclusion

Since the dawn of computing, file systems have been bound to individual storage systems, forcing organizations to manage multiple independent silos for different applications, use cases, or scale requirements. These silos introduce unnecessary operational overhead, user confusion, governance challenges, and cost. The problem is not new, but AI and analytics workloads are bringing it into sharper focus by demanding broad, high-performance access to data across the organization.

The answer is not to deploy yet another storage system for AI and copy data into it, exacerbating the silo problem. The answer is to address the underlying architectural limitation that created the silo problem in the first place.

This is what Hammerspace does. Assimilation is the first step in liberating data from isolated silos into the Hammerspace high-performance, global shared file system.