Secure Unified Storage essentially integrates On Demand PFS and On Demand Archive to provide an intelligent, scalable, and high-performance parallel file system integrated with archiving system tailored for HPC and AI workloads. Both solutions can also be deployed independently or together in an hybrid environment and sharing the management plane.

This Document provides detailed information about components and system architecture.

Secure Unified Storage plaform

White Paper

Authored by:
Rakesh Sabharwal
Founder & CEO
On Demand Systems Pte Ltd

ON DEMAND PFS

+

ON DEMAND ARCHIVE

Unified Platform

COPYRIGHT STATEMENT:

TABLE OF CONTENTS

EXECUTIVE SUMMARY

In the realm of high-performance computing (HPC) and artificial intelligence (AI), designing a storage system that strikes the right balance for efficient and optimal performance poses significant challenges. These environments, often consisting of hundreds of compute/worker nodes, handle vast amounts of data to solve intricate and complex problems. They need to support both large data sets for big data analytics and numerous small files for machine learning (ML) and AI applications without experiencing performance issues. Organizations are seeking affordable, high-performance, and highly available solutions that are also easy to manage while adapting to the growing demands of diverse I/O-intensive workloads.

The challenge is that HPC/AI storage infrastructure requirements can change quickly, requiring companies to scale up and scale out their resources. Composable Disaggregated Infrastructure (CDI) represents the modern architectural approach to data centre infrastructure, disaggregating compute, storage, and network resources into shared pools that can be composed for on-demand allocation. Composable disaggregated infrastructure (CDI) for HPC/AI storage is the key to solving this optimization problem. It enables valuable resources to be deployed through software at just the right levels and within a minimal amount of time, even inside an HPC or AI cluster.

Data-intensive workloads in High-Performance Computing (HPC) and Artificial Intelligence (AI) are growing exponentially. With petabyte-scale datasets becoming the norm, organizations need a well-designed storage strategy that balances performance, scalability, and cost efficiency.

Traditional storage architectures, whether all-flash or disk-based systems struggle with scalability, access latency, and cost overheads. This is where On Demand Archive comes in: an intelligent, high-performance archiving solution designed to optimize HPC and AI data workflows and storage.

In association with On Demand PFS and other third-party partner solutions, the solution can support advanced features of Intelligent Data Tiering and Policy based archiving and retrieval.

PROBLEM STATEMENT

One of the primary challenges HPC/AI Infrastructure teams face today is managing data flow. With vast amounts of data generated every day, simply storing everything on expensive, high-performance hardware is no longer practical.

All data is not equal due to factors such as frequency of access, security needs, and cost considerations; therefore, data storage architectures need to provide different storage tiers to address these varying requirements. Storage tiers differ depending on disk drive types, RAID configurations or even completely different storage sub-systems, which offer different intellectual property profiles and cost impact.

By systematically categorising your data, you can enhance retrieval speeds, improve your system's efficiency, and reduce costs significantly. Implementing data tiering can also mitigate risks associated with data management. Think of it this way, if your systems crash, you want to ensure that your most important data is prioritised and recoverable first. Data tiering allows you to create a structured backup strategy that protects your vital information while relegating less critical data to lower-cost solutions.

This document provides insights into various options available and their pros and cons.

As organisations continue to generate more data, the importance of intelligent data tiering will only grow. The ability to maintain comprehensive data access while optimising costs will become a critical competitive advantage.

ON DEMAND SYSTEMS STORAGE SOLUTIONS HIGHLIGHTS

The following sections of this paper provide an overview of On Demand PFS and On Demand Archive solutions.

ON DEMAND PFS

On Demand PFS consists of a tightly coupled and pre-integrated stack using Rocky Linux, BeeGFS File System, XiRAID and Prometheus/Grafana monitoring tools. The system is built using Industry Standard Hardware, HPE Server/Storage, Nvidia Mellanox InfiniBand Switches and Adapters.

The solution combines:

- HPE's Standard Proliant Servers with low-latency locally attached NVMe SSDs.
- Nvidia Mellanox IB or Ethernet network fabric.
- xiRAID is a high-performance software RAID developed specifically for NVMe storage devices to utilise up to 97% of hardware performance capabilities.
- BeeGFS is a high-performance parallel file system designed for performance-oriented environments like HPC, AI, and deep learning workloads. BeeGFS includes a distributed metadata architecture for scalability and flexibility reasons. Its most important aspect is data throughput.

By combining BeeGFS and xiRAID with HPE Standard Server and Nvidia Mellanox Fabric, organisations can benefit from the parallel file systems:

- High Performance
- High Availability
- Scalability
- Robustness
- Easy to deploy and integrate with existing infrastructure
- Easy data management
- Optimised for highly concurrent access
- BeeGFS - software with enterprise features

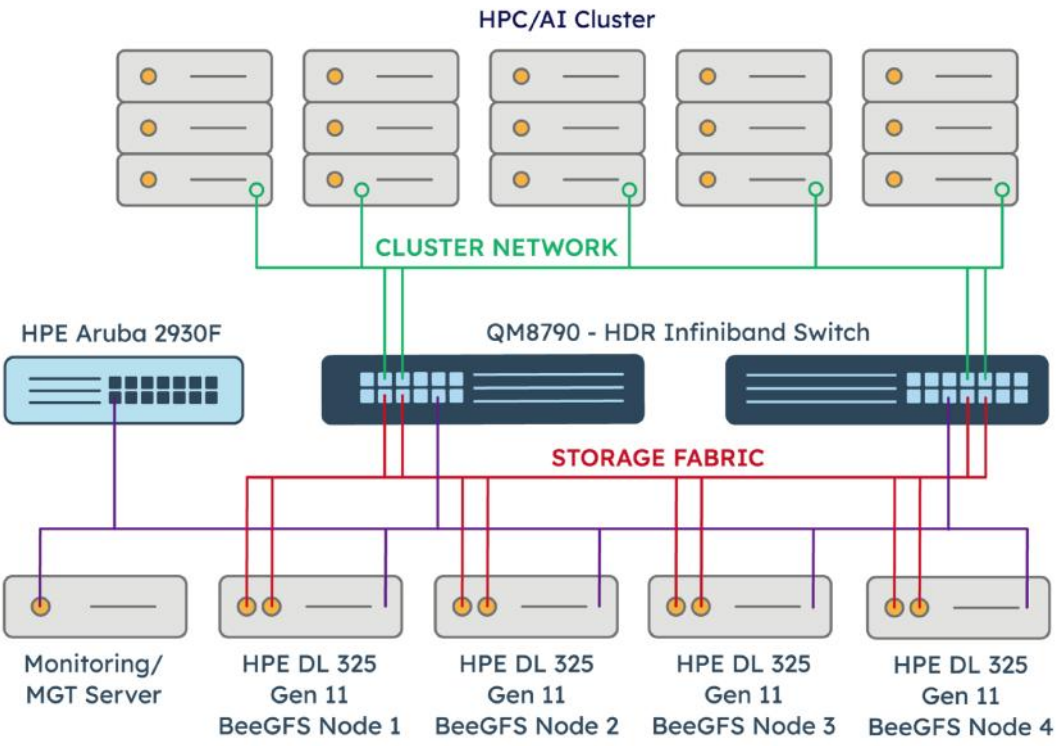


Figure 1: On Demand PFS - System Architecture (4-Nodes with IB Fabric)

KEY BENEFITS:

- Achieve HPC Performance: Unmatched Performance, Scalability, Robustness & Ease of use! Performance that is well balanced from small to large files. On Demand PFS is built on highly efficient and scalable multithreaded core components with native RDMA support.
- Eliminate Bottlenecks: By distributing file contents and metadata across multiple storage and metadata servers, On Demand PFS makes it possible to avoid architectural bottlenecks. The system can scale to any workload requirement in terms of throughput and I/O requirements.
- Seamless Concurrent Access: On Demand PFS lets you avoid the usual performance problems and was built to deliver optimal performance when the I/O load is high.
- Simplify Management: On Demand PFS eliminates the complexity of managing open-source parallel file systems. With patchless kernel module and user-space server daemons, it lets you easily scale and manage systems using graphical tools and monitoring.

- Availability: On Demand PFS supports both InfiniBand and Ethernet connections, serving RDMA (InfiniBand), RoCE and TCP/IP simultaneously. It automatically switches to a redundant path if any connection fails.

ON DEMAND ARCHIVE

On Demand Archive is an appliance-based preconfigured Archival solution, built on Ceph storage, aimed at mid-sized HPC/AI installations looking to protect their research data. It can be easily scaled horizontally by adding more nodes, thus increasing performance and capacity.

On Demand Archive consists of a tightly coupled and integrated stack using Rocky Linux, Ceph and monitoring tools. The system is built using industry-standard hardware, HPE server/storage, Nvidia Mellanox Ethernet switches (or equivalent) and adapters. The System comes with 2x Nvidia Mellanox Ethernet Switches (100GbE, 200GbE and/or 400GbE options available).

Hardware:

- HPE Alletra Storage Server
- Nvidia Mellanox Ethernet switches
- Nvidia Mellanox Ethernet Adapters
- HPE Aruba Ethernet Switch

Software Stack:

- Rocky Linux
- Ceph Storage (Option for supported Ceph Distribution)
- Prometheus/Grafana

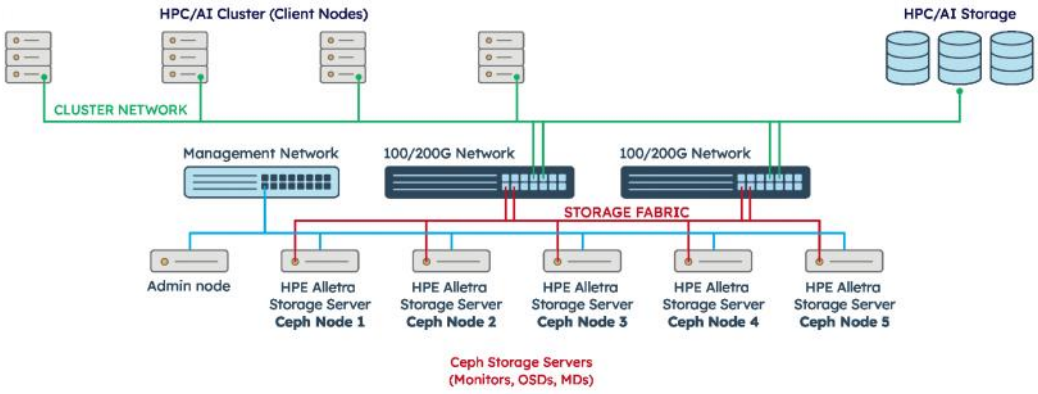


Figure 2: On Demand Archive - System Architecture (4-Nodes with Ethernet Fabric)

KEY BENEFITS:

- True cost-effectiveness: The open-source architecture eliminates license fees, while the ability to run on commodity hardware reduces hardware costs. Coupled with its scalability and efficiency, On Demand Archive maximises resource utilisation, driving down operational expenses and delivering long-term value.

- Supreme flexibility: On Demand Archive offers complete freedom with no vendor lock-in or restrictions, ensuring users are not forced to replace storage after adoption. Hardware updates or replacements can be carried out seamlessly, with zero downtime, ensuring continuous operation and adaptability.

- Rapid scalability: Easily scale your storage by adding or removing nodes as needed. Ceph automatically adjusts data distribution, ensuring optimal performance and seamless expansion without manual intervention.

- Outstanding reliability: On Demand Archive ensures data integrity, with guarantees that all data is stored correctly on the underlying media. Features like scrubbing are implemented to prevent bit rot, ensuring long-term reliability and data protection.

ON DEMAND OBJECTSTOR

On Demand ObjectStor is an appliance-based, preconfigured Object Storage Solution, built on MinIO Enterprise Object Store, aimed at small to mid-sized HPC/AI installations, which provides high performance, especially aimed at AI workloads working on a Kubernetes platform. It can be easily scaled horizontally by adding more nodes, thus increasing performance and capacity.

On Demand ObjectStor consists of a tightly coupled and integrated stack using Rocky Linux, MinIO and monitoring tools. The system is built using industry-standard hardware, HPE server/storage, and Nvidia Mellanox Ethernet Switches and Adapters.

Hardware:

- HPE ProLiant Servers
- Nvidia Mellanox Ethernet switches
- Nvidia Mellanox Ethernet Adapters
- HPE Aruba Ethernet Switch

Software Stack:

- Rocky Linux
- MinIO Enterprise
- Prometheus/Grafana

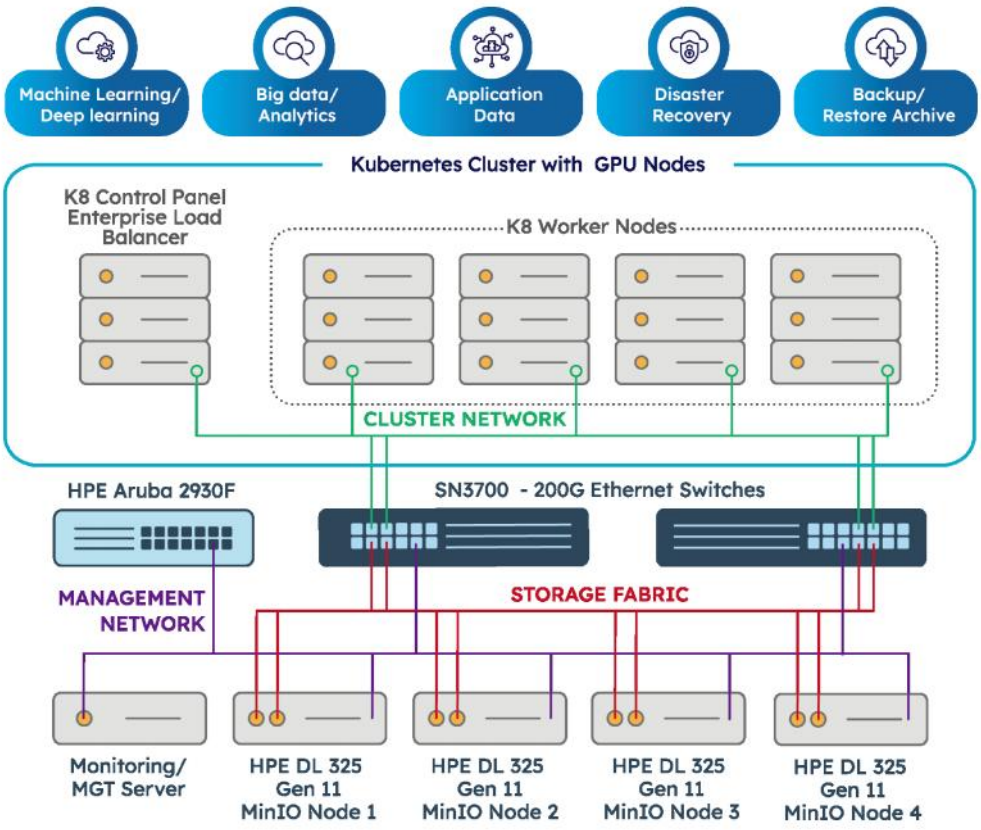


Figure 3: On Demand ObjectStor - System Architecture (4-Nodes with Ethernet Fabric)

KEY BENEFITS:

- Fast, High Performance: On Demand ObjectStor, powered by MinIO, simplifies storage with a single-layer, object-only design, eliminating metadata databases and reducing latency. With read/write speeds of 183GB/s-171GB/s on standard hardware, it supports AI/ML workloads like Spark, TensorFlow, Presto, Hadoop HDFS, and H2O.

- Kubernetes Native storage: The Kubernetes-native and high-performance object storage platform of On Demand ObjectStor with MinIO is designed to cater to the hybrid cloud demands. It can deliver stable functionality for your applications.

- Kubernetes Multi-tenancy: Deploy and manage multiple isolated tenants within the same Kubernetes cluster. Tenants are fully isolated, protected from disruption by others, and scale independently. MinIO secures each tenant separately and encrypts data saved on drives and transmitted across the network.

- Software-defined: On Demand ObjectStor supports multiple use cases for wide-ranging environments. The software-defined suite of MinIO runs in public and private clouds seamlessly at the edge and establishes itself as a front-runner in hybrid cloud object storage.

HPC/AI STORAGE NEEDS

HPC workloads involve parallel computing clusters running large-scale simulations, modeling, and data analytics. **These workloads demand:**

✓ High-throughput I/O to process terabytes per second.

✓ Low-latency access for real-time computations.

✓ Scalable capacity to store datasets generated over time.

AI Storage Challenges

AI pipelines involve:

✓ Training datasets requiring rapid access for iterative model updates which equates to “throughput”.

✓ Inference workloads that need quick retrieval of archived models which equates to “Latency”

✓ Hybrid or Unified storage workflows, where AI dynamically moves data between hot and cold tiers.

Traditional NAS, SAN, Object storage and backup solutions cannot efficiently handle the performance-cost tradeoff, leading to excessive CapEx and OpEx.

Apart from the above challenges, other aspects such as Capacity optimization, Data resilience, Cost optimization, and Data Security are important as well.

HOW UNIFIED STORAGE SOLVES THESE CHALLENGES

On Demand PFS and On Demand Archive provide an intelligent, scalable, and high-performance parallel file system integrated with archiving system tailored for HPC and AI workloads. Both solutions can also be deployed independently or together in a hybrid environment and sharing the management plane.

INTELLIGENT TIERING & POLICY-BASED ARCHIVING

- Automated data movement between high-performance storage (NVMe, SSD) and cost-efficient archive layers (HDD, tape, object storage).
- Policy-driven archival—data is classified based on usage frequency, enabling seamless tiering.

- Hierarchical Storage Management (HSM) integration for tape, object, and cloud archival.

HIGH-PERFORMANCE METADATA INDEXING

- Uses distributed metadata management to accelerate file lookup.
- Metadata caching with SSD/NVMe layers to enable sub-millisecond search times.
- Eliminates cold storage delays through predictive prefetching algorithms.

PARALLEL DATA ACCESS FOR HPC & AI

- Supports high-speed parallel file systems.
- Optimized for RDMA & InfiniBand fabrics, reducing network latency.
- Provides S3-compatible APIs for seamless integration with AI data lakes.

SCALABILITY & COST OPTIMIZATION

- Linear scalability across petabyte to exabyte-scale deployments.
- Object-based deduplication & compression reduce storage footprint.
- Pay-as-you-go model for cloud-based archival, reducing TCO.

Note: These capabilities are included in the software stack using BeeGFS and Ceph and other third-party solutions which are supported as part of the overall solutions offering.

Major use cases for this solution:

Optimisation: a small flash-based storage for scratch and user data as tier 1, capacity storage as tier 2 and object storage as tier 3.

Resilience: adding a highly protected and available data platform using ODA in addition to high performance ODPFS.

Cost optimisation: using common services and infrastructure

ON DEMAND PFS + ON DEMAND ARCHIVE

Unified Platform

SYSTEM ARCHITECTURE

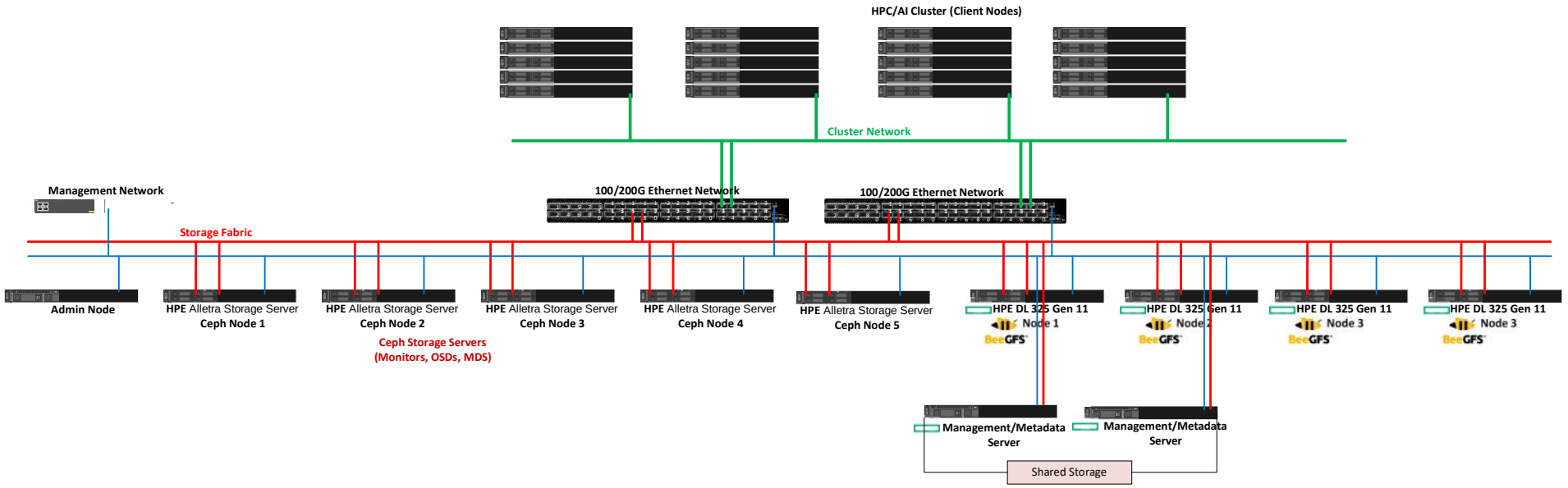


Figure 4: Unified Storage - System Architecture

USE CASES

SCENARIO 1 – ON DEMAND PFS AS SCRATCH AND ON DEMAND ARCHIVE AS ARCHIVAL

In this case, the customer wants to use the On Demand PFS for scratch and active data and use On Demand Archive for archival requirements.

We do have an option in On Demand PFS to use BeeGFS Storage Pools. Storage pools allow the cluster administrator to group storage targets and mirror buddy groups together in specific classes. For example, there can be one pool consisting of fast, but small flash drives, and another pool for bulk or capacity storage, using big but slower spinning disks. Pools can have descriptive names, making it easy to remember which pool to use without looking up the storage targets in the pool. The SSD pool could be named "fast or performance" and the other "bulk or capacity".

While all-flash systems usually are still expensive for systems that require large capacity, a certain amount of flash drives are typically affordable in addition to the spinning disks for high capacity. The goal of a high-performance storage system should then be to take optimal advantage of the flash drives to provide optimal access speed for the projects on which the users are currently working.

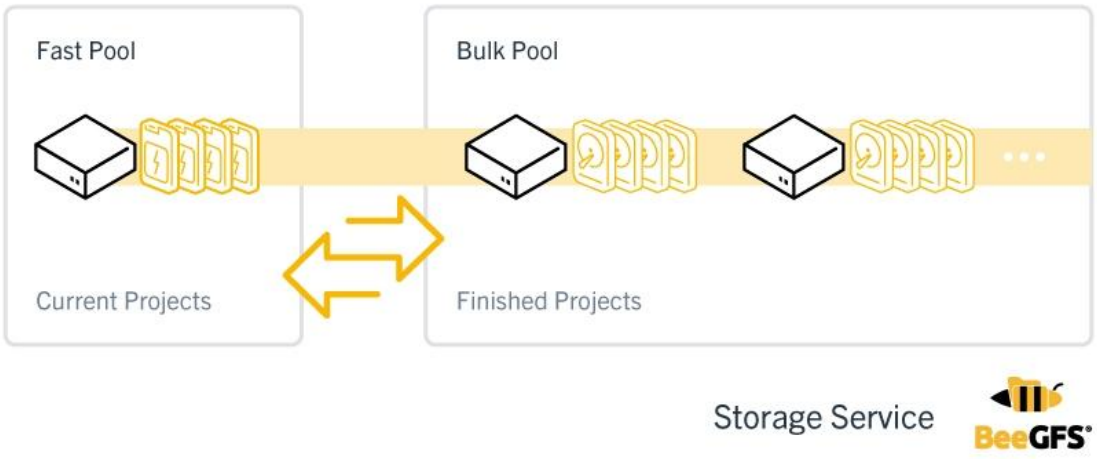


Figure 5: BeeGFS Storage Pools

FLASH-BASED	HDD-BASED
BeeGFS storage targets can be combined into a "Fast Pool" and tied to a directory used for active/ higher-performance I/O work.	BeeGFS storage targets can be combined (or left in the default storage pool) and used in either a defined directory or as the default for the rest of the BeeGFS storage pool for archiving or work that doesn't need to be done on the "Fast Pool".

To enable users to get the full, all-flash performance for the projects on which they are currently working, the BeeGFS storage pools feature makes the flash drives explicitly available to the users. This way, users can request from BeeGFS to move the current project to the flash drives and thus all access to the project files will be served directly and exclusively from the flash drives without any access to the spinning disks until the user decides to move the project back to the spinning disks. The placement of the data is fully transparent to applications. Data stays inside the same directory when it is moved to a different pool and files can be accessed directly without any implicit movement, no matter which pool the data is currently assigned to. To prevent users from putting all their data on the flash pool, different quota levels can be defined for each pool, based on which a sysadmin could also implement a time-limited reservation mechanism for the flash pool.

NOTE: On Demand PFS does support multi-tiered architecture where we could configure e.g. flash storage as tier 1 and spindle HDDs as tier 2 thus offering as solution to store active and archival data within the same system, but this would mean that all data is within the same system which is not a best practice.

Considering this, we may want to explore a system where we have the archival or capacity tier but not within the same storage platform but well integrated to move the data between the tiers as required.

On Demand PFS with BeeGFS v8.x has a feature which could enable moving data based simple policies to remote storage e.g. object storage thus creating a tier 2/3 based storage which can be at a different site or location.

The Remote and Sync services can be deployed on the same physical server as other core file system services, or on dedicated servers, depending on the available hardware and requirements of a particular environment. This allows remote targets to be added to existing BeeGFS deployments where the servers running core services may not have been sized with the remote targets feature in mind. This also makes it possible to avoid exposing the servers running core BeeGFS services directly to the internet (in case the remote target is cloud service provider) and treat the Remote/Sync servers as “gateway” nodes.

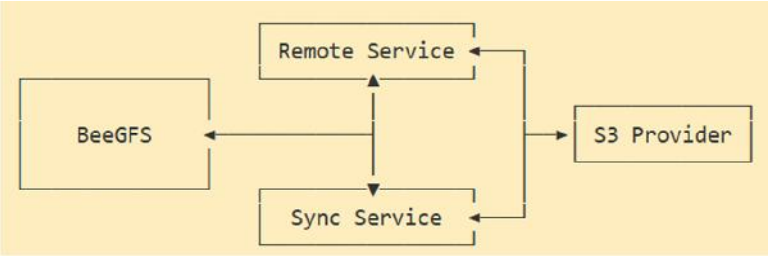


Figure 6: BeeGFS Remote Target Service

Further, we could also create a tier 4 of object storage in cloud where the data can be deep frozen for longer durations.

We propose to use On Demand Archive (using object storage) as the remote target with high availability and data resilience.

SCENARIO 2 – ON DEMAND PFS FOR ACTIVE DATA AND MOVING DATA TO MULTIPLE STORAGE TARGETS

In this case, we could use other third-party tools to fulfil this requirement.

RESILIO

Resilio Active Everywhere Platform

Resilio is a highly reliable and efficient alternative for global file sharing, caching, replication, and synchronization. Combined with your chosen on-premise workflows or hybrid cloud storage services, it is a superior alternative to conventional solutions.

The Resilio Approach

Resilio Active Everywhere offers a compelling solution for enterprise file sharing and synchronization. Key advantages include 10x faster access to your storage solutions through peer-to-peer technology, superior scalability, and lower total cost of ownership with no CapEx requirements.

The Resilio approach enhances remote work productivity, reduces cloud egress costs, and provides greater flexibility across edge, on-premises workflows and cloud-native environments. Its Zero Gravity Transport™ protocol optimizes data transfer across various network types, while the platform's data storage-agnostic approach and automation readiness offer significant operational benefits.

For businesses seeking efficient, scalable, and cost-effective data management, Resilio presents a robust and future-proof option.

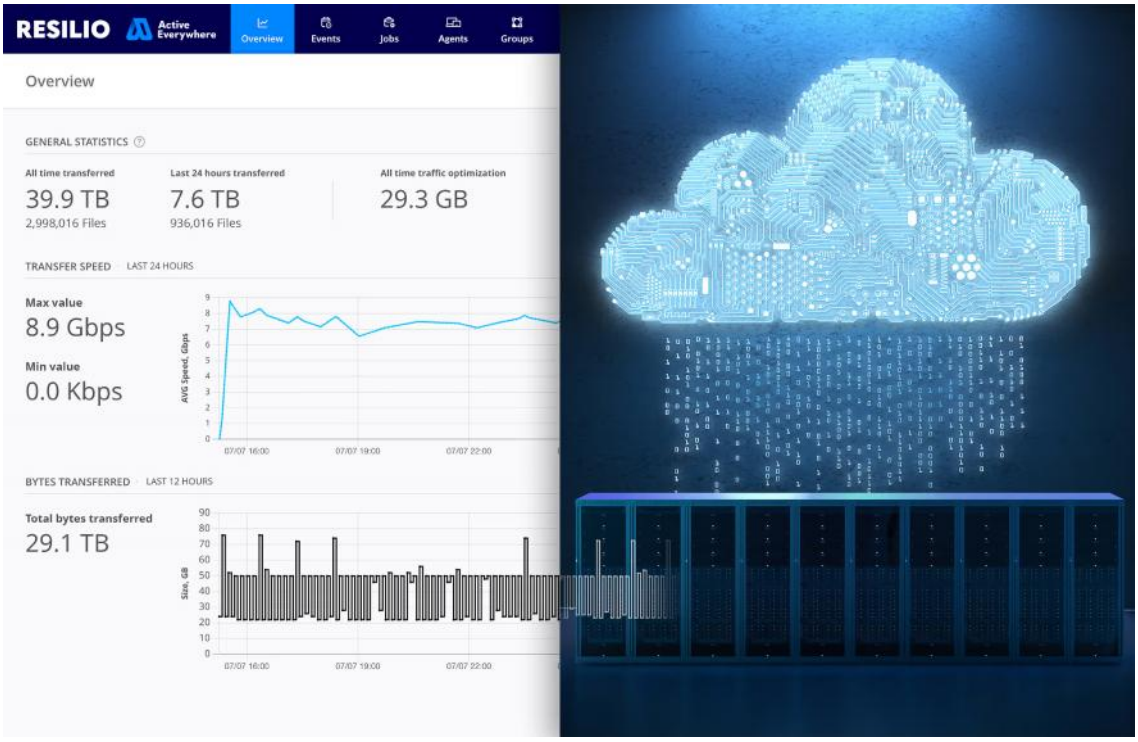


Figure 7: Resilio Dashboard

The Architecture Makes the Difference

In traditional hub-and-spoke architectures, adding more endpoints linearly increases synchronization time, creating a bottleneck as your environment grows.

The Resilio advantage is our peer-to-peer, distributed architecture for file synchronization—a revolutionary approach.

- Every endpoint collaborates: all endpoints in your environment work together to sync files.
- Organic scalability: adding more endpoints increases sync speed and improves data availability.
- More demand equals more supply: as your network grows, so does its synchronization capacity.

Not only is Resilio's architecture faster by design, but it also dramatically reduces egress costs. In a distributed, p2p architecture, most data comes from other devices on your network. Unlike traditional hub-and-spoke-based solutions, which incur egress costs at each endpoint for replication.

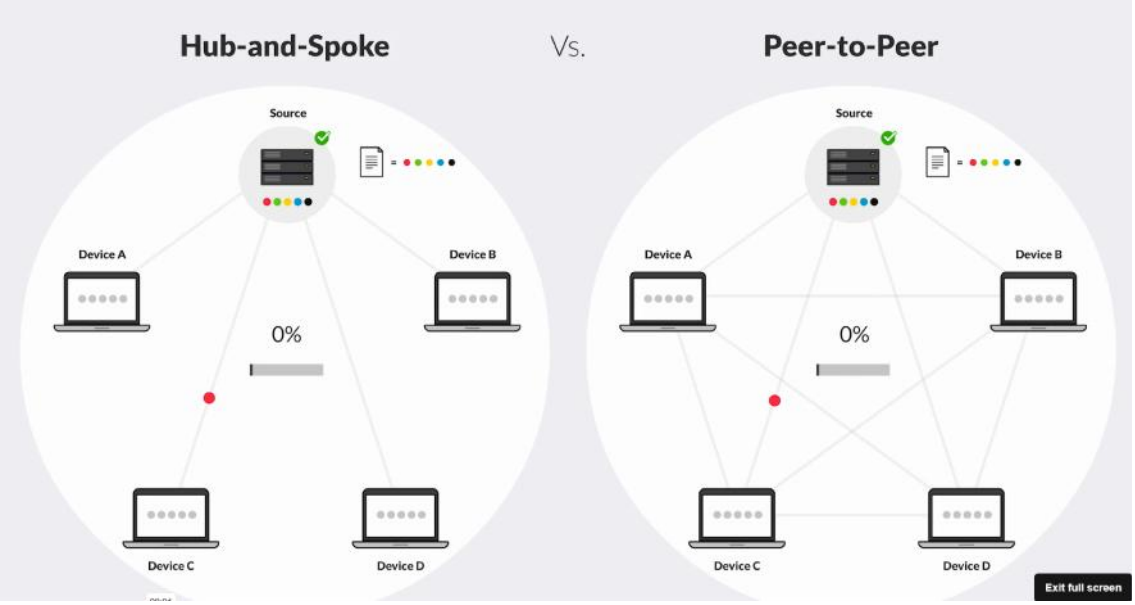


Figure 8: Resilio Hub Spoke Data Synchronising

Configuring the Storage tiering job

Before creating a job, edit Default Job Profile or create a new one with the following parameters: *Lazy Indexing: Yes.* Otherwise file (objects) will be retrieved from AWS Glacier archive twice. *fs_enable_meta: false* (advisable for AWS Glacier storage, custom parameter). Disabling metadata synchronization is not compulsory. If enabled and metadata on a retrieved object changes, the object will be moved to AWS Glacier archive again.

Click Jobs -> Create a new job -> Storage Tiering and Archival job.

Select the created Job profile.

Only one source Agent and one destination Agent are supported. Data can be transferred between cloud and non-cloud storages.
Resilio AE 4.2.0 implements native support for AWS Glacier storage tiers.

High availability groups are supported as source and destination.

Parameters for data transfer.

On tab STORAGE TIERING configure the parameters for files to be transferred. These parameters apply to the files/objects in the source directory.
Modification / access time. Select the files depending on their modification or access time.
Profile parameters "Max/min file modification time (sec)" from Job profile are not applied to this job.
Access time configuration is disabled in case source Agent is pointed to a cloud storage, cloud storages don't support access time on objects.
Access time must be enabled on a Windows Server (it's disabled by default, see **here** for more details)

Exact files list. Supports regular expressions delimited by \n, locations relatively the job path configured for the source. Case sensitive. Field cannot be empty. To transfer all files in the source folder, add .* as the list.

Please click outside of the editbox after editing the rule for the rule to be initialized by MC and *Next* button to activate to continue editing the job..

Indexing optimization can be enabled by using ^ - at the beginning of the line, in all patterns. Agent performs partial matching to exclude unsuitable folders while indexing. For instance, with pattern ^folder1/.*\\.txt and data structure

folder1/

file1.txt

file2.txt

folder2/

file3.txt

file4.txt

directory ‘folder2’ wont be even visited by the Agent, since there are no possible paths inside it that could match any given pattern.

Storage class. Select storage class, if the AWS storage is selected as destination. Files from source will be placed in the chosen storage tier.

Pre-seeded AWS Glacier storage

Pre-seeded objects in the AWS storage already have a storage class assigned. On the next job run, such objects will be moved to the new storage class per job’s settings only if the file on source is different from that in the bucket (by size, timestamp). Otherwise, the storage class will be preserved as is.

Restoring. Select restoring tier to retrieve data from an AWS Glacier archive.
Expedited retrieval is not available in Glacier Deep Archive class. Such objects will not be retrieved and the Agent will give an error.

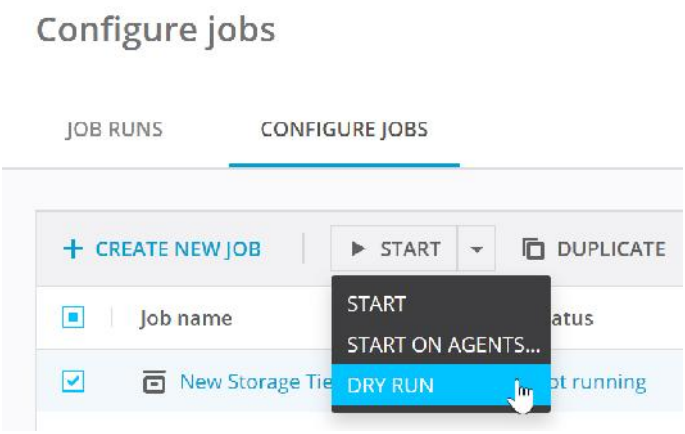
Retrieval classes are also supported for Oracle cloud storage.

By default objects are restored for 3 days. Configurable with custom parameter cloudfs.s3.retrieval_days in Job profile.

File retention and deletion settings. Option to clear files from source Agent once they’re transferred to the destination.

Dry run for the Archival job

Before starting the job itself, it’s possible to launch a dry run. It’s launched only on the source Agent, it scans the files that match the configured files list and will be transferred in the job.



Please note, the dry run only checks the list of files that match the rule! It does not calculate file hashes and does not verify files' availability for the transfer, does not perform any action with the files. Besides, the files may change after dry run completes but before the job run is started. Files may also change during the job run.

Dry run can be started only manually. Dry run cannot be started if the job itself does not have a destination agent.

The job run itself and the dry run for this job cannot work simultaneously. The dry run must be stopped in order to launch the job run itself.
Dry run cannot be paused, but it can be stopped.

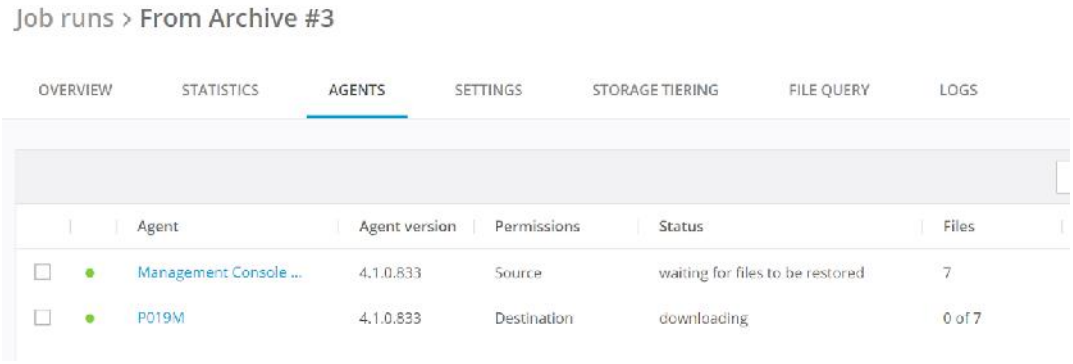
Resulting list of files is available from FILES tab.

The following MC functionality is not supported for the dry run:
- **Mail** and **webhook** notifications
- **Ignoring some errors** or **aborting the dry run on an error**
- MC API
- Starting by the job scheduler
- **Job triggers**
- Launching the dry run on specific Agents and **restarting the dry run on Agents**
- **File query**
- Extended statistics information
- Some profile parameters regarding hashing, aborting the job on timeout, executing scripts.

Monitoring the storage tiering job

During retrieval from the Glacier archive, source Agent reports status 'waiting for files to be restored'. Until an object is retrieved, there's no active data transfer and the job run's ETA shows "unknown" state.

Destination Agent reports status “downloading” with the corresponding files count.



Deleting files from source after transfer in distribution jobs

Available in version: 4.1
Supported for: Distribution and Consolidation jobs. Local, network and cloud storages.

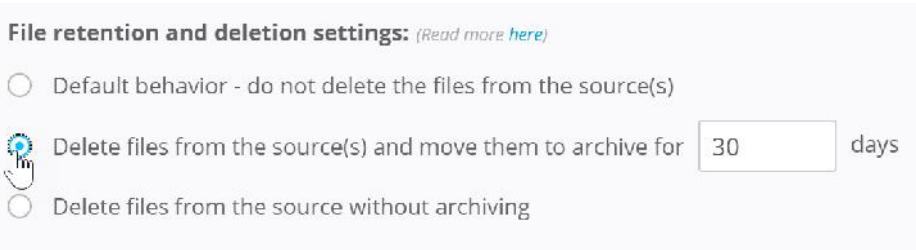
When files are cleared from source.
What files are cleared and not cleared from source
Where the files are deleted from source and how they are stored there
Peculiarities and considerations

Option to enable clearing the files from storage on source is available from SETTINGS tab in Distribution and Consolidation jobs.
If enabled, the source Agent(s) will clear the local files (move to Archive or skip the Archive) after the file transfer is complete and all **job triggers** are executed.

Avoid dataloss

If the option "Delete files to archive" is disabled, files are deleted from the system skipping system bin. Files can be restored only from the destination Agents or a third-party backup.

Below outlined are specifics of the



functionality.

When files are cleared from source

Clearing the files from source is the final action done by the Agents in the job.
Source Agent will clear the files only after it verifies with **all** the destination Agents that they have a copy and after all job triggers are executed. An offline destination Agent blocks this operation, keeping the job in progress until aborted or timed out.

This also means that files are not cleared from source if the job is aborted before it successfully finishes.

Two new statuses can be reported by the source Agent: "resolving peers availability before cleanup" and "cleaning up source folder".

What files are cleared and not cleared from source

All files that were transferred to destination Agents, including the pre-seeded files (that were not transferred to destination because there is already a copy of the file).

These files can be seen in Job run -> Agents -> <source_agent> -> Files events, by status "Archived".

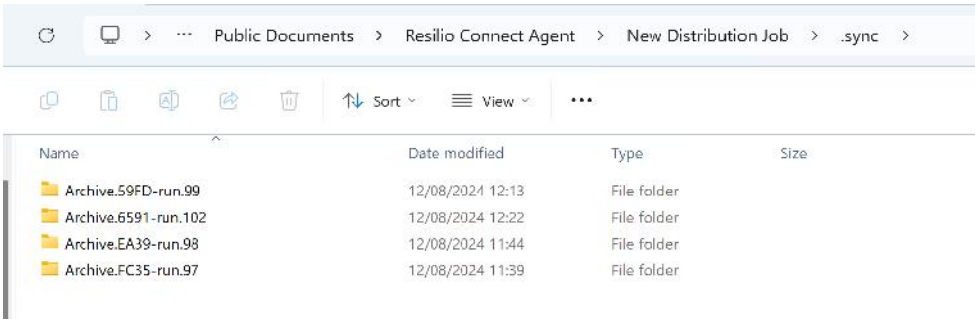
Files are not cleared from source in the following cases:
- not confirmed to be synced to destination, e.g. skipped because of an error or are in IgnoreList;
- not all destinations confirmed they received the files, e.g. a destination Agent is offline.
- files were removed from the destination Agent(s) in a trigger.
- this is a large dataset and/or slow storage, and job run was aborted during the clearing process, manually, by error or parameter "Wait for job run to complete before starting new one" in job scheduler.

Where the files are deleted from source and how they are stored there

By default, files are deleted to Archive, unless the option is unchecked manually by the Admin. Archive is located inside the job folder on the source in the hidden .sync directory. It is created even if the source folder was empty.
Agent performs a *move* operation.

This is not the Archive which is created in Synchronization, File Cache and Hybrid jobs, and on destination Agents in Distribution and Consolidation jobs, which is designed to store file version and deleted files. It is not managed by the "Use archive" and "Max archive file age" parameters in Agent and Job profiles.

Archive name includes the shareID (internal ID calculated by Agent and not visible to user or admin) and runID (this is the ID of the job run as seen in Job Runs table, column ID on the MC).



Files are stored in the Archive per job run during the number of days configured in the job.
Files will be cleared from the archive skipping the system bin after that time.

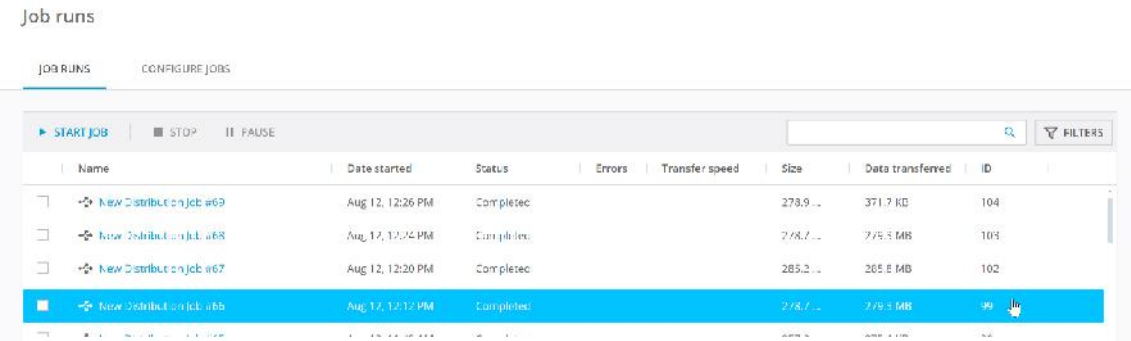
Storing the files in the Archive is not dependent on the existence of the job itself on the Management Console. Even if the job is already deleted, the Archive will remain on the storage and files will be

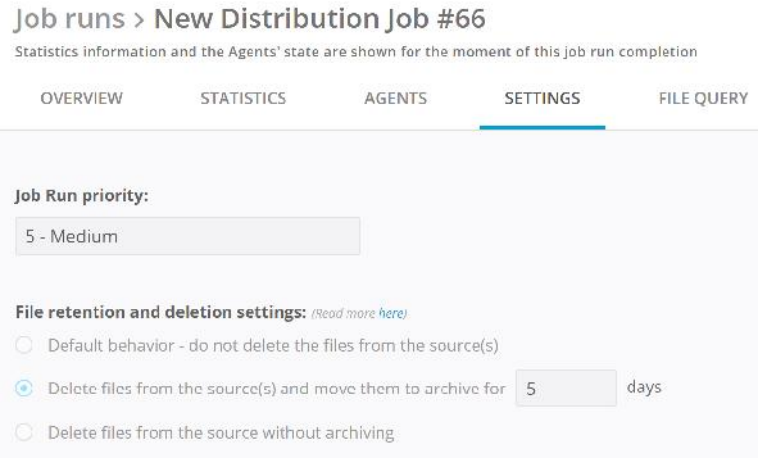
deleted from in accordance with the setting in the job run.

How to check when the files will be deleted from the Archive

If the job or job run is already deleted from the MC, there's no way to do that. Rely on the Agent's mechanism to delete the files from Archive when the time comes. If this is an emergency question, please contact support.

For still existing job run:
- in the storage on source agent, spot the Archive folder of interest and its run.ID. In the image above, let it be Archive with run.99.
- in the Management Console, open job runs and in column ID find the corresponding job run. Open it and check tab settings.
- the files will be deleted in 5 days.





PANZURA

4.3.1 Panzura Symphony

Panzura Symphony optimizes storage usage and reduces associated costs. Dynamic Workload Placement capabilities allow teams to precisely orchestrate data placement, archive, and migration to transform their data operations framework. The platform delivers a comprehensive approach to data management and movement, enabling efficient data workflows, and allowing for the seamless transfer and management of data across different environments.

- Precise orchestration of data placement, archive, and migration
- Enhance efficiency and ensure data management is done swiftly and without disruption
- Support for multiple storage tiers and enhanced control over data assets
- Visibility and seamless integration between file system deployments and diverse storage landscapes
- Significant cost savings and reductions to overall storage footprint
- Streamline and accelerate data migrations to avoid being locked into specific cloud providers or hardware vendors

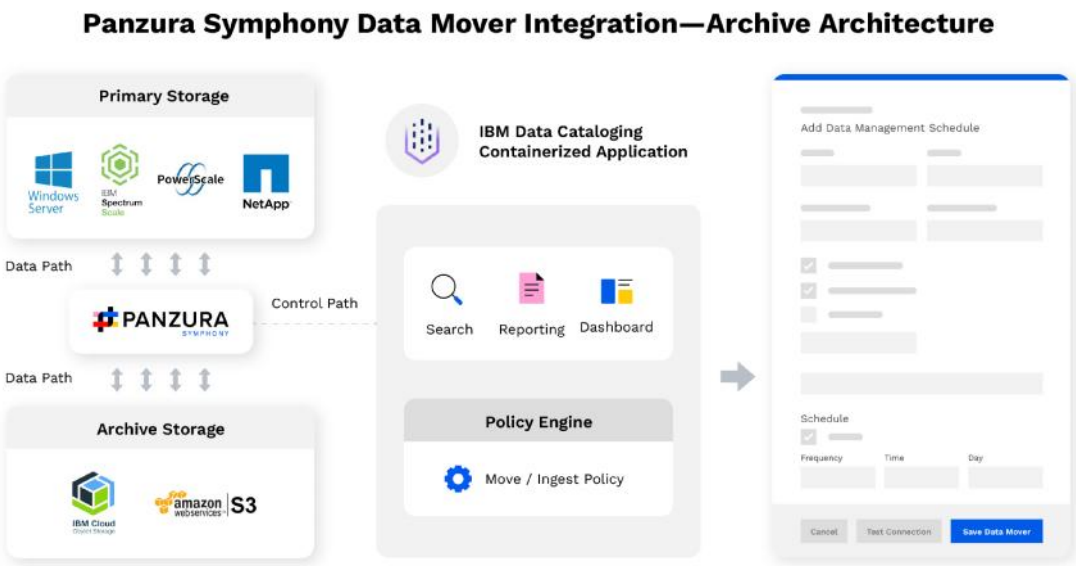


Figure 9: Panzura Symphony

Extensive Native Integrations and Support

Symphony offers extensive integrations and support for file system deployments and protocols, and on-premises, private, public, and hybrid cloud object storage.

File Systems and Protocols

Native integrations include Microsoft Windows Server, NetApp ONTAP, Dell EMC, SMB and NFS protocols.

Cloud and Object Storage

Support includes Alibaba OSS, Amazon S3 and compatibles, Azure Blob Storage, Backblaze, DataCore Swarm, Dell EMC ECS, Google Cloud Platform, Hitachi Vantara HCP, IBM Cloud Object Storage, NetApp StorageGRID, and Wasabi Hot Cloud Storage.

Key Advanced Features

• REST Management API: Enables rapid integration with third-party systems.

• Seamless Data Mobilization: Supports conversion and mobilization of unstructured data between file and object storage while preserving metadata.

• Event and Content Awareness: Allows data management by user or application events, file system or object attributes, direct API calls, or content inspection.

Panzura Symphony is a heterogeneous Data Management System. In addition to providing data insights from file metadata, Symphony automates and manages data mobility between file systems, object stores, and cloud storage services. Use cases include storage cost optimisation, backup optimisation, and workload placement.

Panzura supports Data Tiering, by providing stubbing of a file after the move, providing a link to the tiered data. This feature is only supported for SMB source as stubbing is not supported for NFS.

A Panzura Symphony Policy specifies an operation to perform on a set of files. Depending on the type of operation, a Policy will specify Source(s) and/or Destination(s), and possibly Rules to limit the Policy to a subset of files.

For example:

Move all files in a specific folder on the Source NFS to the Dell ECS Object Storage that have not been modified in the past year.

BEST PRACTICES FOR HPC/AI DATA STORAGE

IMPLEMENT ROBUST VERSION CONTROL FOR DATASETS

The development of machine learning models is inherently iterative, involving continuous refinement of hyperparameters, source code, and datasets to achieve optimal performance. To ensure reproducibility and facilitate efficient experimentation, it is essential to maintain comprehensive records of these modifications. Such documentation enables accurate tracking of model performance in relation to specific configurations, thereby minimizing redundant retraining efforts.

In this workflow, metadata—comprising descriptive information about both the dataset and the model—serves a critical function. While the underlying data used for training and evaluation may remain static, metadata can evolve independently. Consequently, robust versioning practices must establish clear associations between datasets and their corresponding metadata to preserve integrity and traceability throughout the development lifecycle.

Maintaining clear versioning across code, data, models, and metadata is essential for reproducibility, collaboration, and efficient model evolution. Below are concise guidelines and considerations for a robust versioning strategy.

ENSURE DATA CONSISTENCY AND INTEGRITY ACROSS PIPELINES

The initial stage in every machine learning pipeline is to collect the data that will be used to train, test, and deploy the models. This phase entails collecting, extracting, and ingesting data from various sources, including databases, APIs, web pages, sensors, and files.

To maintain data consistency and integrity in this step, each source's and destination's data schemas and formats must be defined clearly and consistently.

Data cleaning is the next phase in any ML pipeline. It entails finding and resolving issues, including outliers, duplicates, inconsistencies, noise, and errors. To assure data consistency and integrity, use EDA tools to analyse data distribution, variation, and relationships.

The final phase in any ML pipeline is data validation, which ensures that the data fulfills the models' and stakeholders' expectations and requirements. As a result, it is critical to create and apply data validation criteria for all data elements and attributes. Additionally, using data validation tools like testers, checkers, and alerts can assist in detecting and reporting mistakes.

OPTIMISE STORAGE FOR SCALABILITY AND PERFORMANCE

Scalability is essential for HPC/AI systems, so ensuring that there is always enough space/capacity to accommodate the constant influx of data that drives HPC/AI progress is a critical component of AI storage.

Scalable storage indicates that you are ready to meet and react to new requirements while developing HPC and AI applications and systems. It contributes to maintaining an optimal infrastructure for today's business ecosystem in the face of constant change.

The amount of data that HPC/AI systems process necessitates high storage performance. When HPC/AI systems have instant access to data, they run more smoothly and efficiently. Furthermore, they become less time-consuming, a significant advantage in the era of self-driving automobiles and automated stock trading, where time is crucial.

The speed of AI storage enables faster training on large, complicated datasets, making models more accurate and trustworthy. It also allows systems to scale up without slowing down.

AUTOMATE DATA LIFECYCLE MANAGEMENT AND ARCHIVING

AI data lifecycle management is the process of managing AI systems' data from beginning to end, assuring high-quality data and regulatory compliance. Proper data management is critical because poor data quality, a lack of transparency, and insufficient governance can result in biased models, incorrect forecasts, and non-compliant AI systems.

Adopting an automated data lifecycle management and archiving framework not only streamlines operations but also enforces governance at scale. Next, consider integrating drift-detection mechanisms to flag unexpected changes in data quality or usage patterns as part of your proactive data hygiene strategy.

ENFORCE STRONG SECURITY AND ACCESS CONTROLS

Security and privacy must be prioritized when storing data for HPC/AI systems especially for GenAI Data. Implementing these practices correctly is critical to creating user confidence and ensuring compliance with global standards such as GDPR and CCPA where needed.

Furthermore, prioritizing security and privacy from the outset will deter hostile actors and secure your Gen AI data from hackers, breaches, and tampering. Moreover, in today's unpredictable threat landscape, AI storage should be built on the principles of trust, transparency, dependability, resilience, and ethics.

MONITOR AND AUDIT DATA USAGE FOR COMPLIANCE

Adding data governance standards to data pipelines, like finding and masking sensitive attributes, can be useful. Because governance is used at entry, data lakes cannot leak sensitive data.

Doing so within the data fabric framework automates the process. Data fabric, with its unified view of data and support for governance activities, helps to enforce and facilitate adherence to data quality standards, security measures, and regulatory needs. Implementing strong data governance processes is critical for enterprises to make informed decisions while ensuring data assets' dependability and trustworthiness.

CONCLUSION

With On Demand PFS and On Demand Archive, organizations can automate data lifecycle management, reduce costs, and maintain high-performance access to archived datasets. This ensures AI models and HPC applications have the right data at the right time—without excessive storage overhead.

The On Demand PFS and On Demand Archive solutions are also available as an OPEX model for more flexibility.

REFERENCES

The information provided in this document is based on the references below. These can be used for further advanced and in-depth reference.

THINKPARQ DOCUMENTATION

The following BeeGFS documentation from ThinkparQ provides additional and relevant information:

- BeeGFS Documentation
- General Architecture of BeeGFS File System
- Storage Pools
- Remote Storage Targets

RESILIO DOCUMENTATION

The following Resilio documentation provides additional and relevant information:

- Storage tiering and Archival

ON DEMAND SYSTEMS PTE LTD OVERVIEW

ON DEMAND systems

ODS provides Secure High Performance Computing (HPC) and Artificial Intelligence (AI) infrastructure solutions. We provide end-to-end services from conceptualisation, design, and deployment to management. The HPC/AI solutions and services are complemented with identity and access management software and cloud-based Identity-as-a-Service that allows enterprises to securely manage identities and secure access across computer networks and cloud computing environments.

CONTACT INFORMATION:

www.odspte.com

info@odspte.com

+65 66047271