Learn about your Novus system

Novus

NetApp
November 03, 2026

This PDF was generated from https://docs.netapp.com/us-en/novus/get-started/learn-novus.html on November 03, 2026. Always check docs.netapp.com for the latest.

Table of Contents

Learn about your Novus system . . . 1
- Learn about Novus storage systems . . . 1
  - Overview . . . 1
  - System design principles . . . 1
  - Network File System (NFS) implementation . . . 2
  - Additional technology and concepts . . . 3
  - Related information . . . 3
- Novus storage system architecture . . . 4
  - ONTAP data server . . . 4
  - Novus Data Director . . . 4
  - Novus deployment . . . 5
- Novus storage system environment and processing . . . 5
  - Novus Data Director and ONTAP data server internal communication . . . 5
  - High-availability and resiliency for the NDD cluster . . . 6

Learn about your Novus system

Learn about Novus storage systems

NetApp Novus is a powerful storage system built for the demands of AI at a very large scale. It extends NetApp's experience in enterprise storage to the extremely high-performance infrastructure needed for training and inference. Understanding its administrative structure and configuration options is crucial for effective management and optimal performance.

Overview

NetApp Novus is a scale-out file storage system solution designed for the throughput and scale needed with modern AI workloads. It joins many independent ONTAP storage clusters into a single namespace that clients see as one file system. Novus is built on parallel NFS (pNFS), which separates metadata from the file data. A dedicated metadata service manages the namespace while the storage clusters serve file data directly to clients in parallel. This removes the throughput ceiling of a single storage cluster which no longer limits how fast the system can deliver data.

A Novus deployment can grow simply by adding storage clusters. Each new cluster adds its full bandwidth and capacity to the namespace, so total throughput increases as the system grows without client remounts or disruptive rebalancing. The metadata service scales separately from the data servers, so you can expand namespace size and metadata performance to fit the workload. Novus is designed to run each storage node at full speed and to treat component failures as routine and non-disruptive at large scale. The primary goal is to keep data flowing to compute even as the system grows to very large sizes.

Clients use a standard NFS client, making Novus simple to adapt to large GPU and compute clusters. And Novus is multi-tenant, so that one deployment can host several isolated customers or workloads, each with its own separate namespaces. It also builds on ONTAP's proven data management and protection. For customers, this means storage that can keep the GPUs busy, lower infrastructure cost by getting more performance from each node, and support very large datasets in a single namespace.

Novus can help organizations that run AI and other data-intensive workloads at scale. This typically includes neoclouds, GPU-as-a-service providers, sovereign clouds, hyperscalers, as well as large enterprises and research organizations running AI training and inference. Novus can benefit any customer using high-performance computing or media processing that needs multiple terabytes per second of throughput and want to scale storage alongside their compute.

System design principles

Novus has been designed and built based on several core principles. These underlying architectural choices drive its performance, scalability, and resilience. This helps customers that need high-performance storage that keeps the GPU busy and reduce infrastructure expenses.

Exascale-ready storage

Novus is designed to be an exascale ready software defined storage foundation stack. It combines horizontally scaled pNFS metadata servers with separate and disaggregated data servers.

Throughput beyond traditional ONTAP cluster limits

Traditionally, a single ONTAP cluster reaches a fixed throughput ceiling as set by its controller count and interconnect bandwidth. AI training and inference need throughput which a single ONTAP cluster cannot

reach.

Scale by adding clusters

Novus joins many independent ONTAP clusters, each as small as one HA pair, into a single namespace. Each new cluster contributes its full bandwidth and so the total throughput grows linearly. Clients don't need to remount and the fleet doesn't need rebalancing.

Metadata that scales separately

The metadata service layer and the sharded metadata store can each grow from one to more than 1,000 instances. Metadata capacity and request rate don't depend on how many data nodes are used.

Line-rate data nodes

Each data node should run at full NIC speed for reads and full NVMe speed for writes. Any unused per-node throughput appears as just extra hardware and cost.

Resiliency as the main design constraint

At the scale of 1,000 or more storage nodes, failures are not uncommon. Resiliency is a primary design constraint and not an operational afterthought.

Network File System (NFS) implementation

Novus is a software-defined implementation of the Network File System (NFS) protocol. It relies on NFSv4—specifically the Parallel NFS (pNFS) extension—to split the metadata and data paths. Clients can use standard NFS clients to access the Novus Data Director (NDD) using NFSv4, while accessing the data directly at the ONTAP data servers using NFSv3.

Parallel Network File System (pNFS)

Parallel NFS (pNFS) is a feature introduced with NFSv4.1 that separates file metadata management from file-data access. The NFS server handles metadata operations and provides clients with information about where file data is stored. Clients can then directly access the data at the dedicated storage devices and data servers, rather than sending all reads and writes through the single server.

The NFS server supplies a layout which describes how each file's data is arranged and how the client can access it. Depending on the layout and storage configuration, a client can communicate with one or more data servers in parallel. This increases data-transfer performance and reduce the data-handling load on the metadata server.

pNFS supports several layout types, including file, block, object, and flexible file. Each type describes a different way of mapping file data to underlying storage and a corresponding way for clients to access it. Novus uses the flexible file layout.

The pNFS flexible file layout is a more recent addition to the layout options. It provides a flexible file-based layout that can describe data stored across multiple data servers and can also support mirrored copies. The client uses the layout information to select an appropriate data-server path for access. This allows deployments to adapt data placement and access behavior to the storage configuration and workload requirements.

Major components

There are two major Novus components which are built on the pNFS standard and NetApp's enterprise storage expertise.

ONTAP data server

The ONTAP data server implements the Novus data plane. It stores file data and serves it to NFS clients. A Novus namespace can spread its file data across several ONTAP clusters, and one ONTAP cluster can serve many namespaces.

Novus Data Director (NDD)

The Novus Data Director provides the Novus metadata plane. It manages the namespace metadata, handles client requests for metadata operations, and provides clients with layouts that describe where file data is stored across the ONTAP clusters. Clients interact with the NDD using NFSv4.2 with pNFS, while the actual file data is accessed directly from ONTAP.

Review the Novus architecture for more details about these components and how they work together.

Additional technology and concepts

There are several additional concepts relevant to Novus and the underlying technology.

Namespace

A namespace is a logical structure that organizes and identifies resources, allowing them to be referenced consistently within a defined scope. With NFSv4, the namespace is the hierarchy of files and directories presented to clients; it provides a unified view of exported file systems, which clients can navigate using paths. Each NDD namespace has its own metadata server instance.

Federation

Federation is a technique for connecting independent systems so they can work together while remaining separately managed; it can be applied to identity, application services, networks, databases, and other resources. When used with data providers, federation enables users or applications to access data from multiple sources through a unified interface, while the providers retain control of their data—supporting interoperability and reducing the need to centralize or duplicate information.

Sharding

Sharding is a technique for splitting a database's data across multiple servers, with each server storing and serving a portion of the data. It improves capacity and performance by distributing storage and the query load.

Border Gateway Protocol

Border Gateway Protocol (BGP) is the routing protocol that exchanges reachability information between autonomous systems—the networks that make up the internet—and selects paths using routing policies and attributes. Novus relies on BGP to support its multi-tenant architecture.

Scalable distributed metadata store

A scalable distributed metadata store is the system managing metadata across multiple nodes, ensuring high availability, fault tolerance, and efficient access. In Novus, the Novus Data Director (NDD) implements a scalable distributed metadata store (or just *metadata store*) to handle namespace metadata and provide clients with layouts for accessing file data.

Related information

- RFC 8435: Parallel NFS (pNFS) Flexible File Layout
- RFC 8178: Rules for NFSv4 Extensions and Minor Versions
- RFC 8881: Network File System (NFS) Version 4 Minor Version 1 Protocol

- RFC 7862: Network File System (NFS) Version 4 Minor Version 2 Protocol

Novus storage system architecture

The Novus architecture consists of instances of the Novus Data Director (NDD) and ONTAP data servers. The ONTAP data servers support the data path while the NDD manages metadata and client access through Parallel NFS (pNFS).

ONTAP data server

The data servers are ONTAP clusters specially built for Novus. They implement the Novus data plane and stores file data and serves it to the NFS clients. A Novus namespace can spread its file data across several ONTAP clusters; one cluster can serve many different namespaces. This flexibility enables capacity and performance to grow by adding data servers as needed.

Clients access the ONTAP data servers using a standard NFS client with no need for custom NetApp software. After a client gets a file layout from the Novus Data Director, it can read and write file data directly on the ONTAP data servers over NFSv3 and optionally over RDMA. Bulk data never passes through the Novus Data Director so data traffic travels in parallel across the data servers.

The data servers don’t take part in managing the namespace. They store each file’s data as backing files in ONTAP volumes. An ONTAP data server doesn’t maintain the Novus namespace metadata, such as the directory tree or how files are striped across data servers.

The AFF A90 for NetApp Novus is a version of the AFF A90 built for Novus running ONTAP. Its characteristics include:

- Additional dual NVRAM cards
- High-bandwidth network adapters
- Configured as a standard ONTAP HA pair
- Protects data with RAID-DP or RAID-TEC.

Standard ONTAP clusters can’t be used as Novus data servers; only platforms specifically built for Novus are supported.

Novus Data Director

The Novus Data Director (NDD) is an implementation of the metadata server (MDS) role in the pNFS architecture and provides the Novus metadata plane. An *MDS host* is one physical server running Linux plus the MDS protocol services and metadata stores.

The NDD presents each namespace to clients as a single file system. It owns the namespace metadata: files and directories, file attributes, the map of where each file’s data is stored, and the protocol state needed to coordinate client access. Clients mount a namespace through its virtual IP address and talk to the NDD using NFSv4.2 with parallel NFS (pNFS). The NDD handles all metadata operations such as lookups, file creates, and attribute changes. When a client opens a file, the NDD gives it a pNFS Flexible File layout, which tells the client which data servers hold the file data. The client can then do its reads and writes directly with the ONTAP data servers; the NDD is in the metadata path but not the data path.

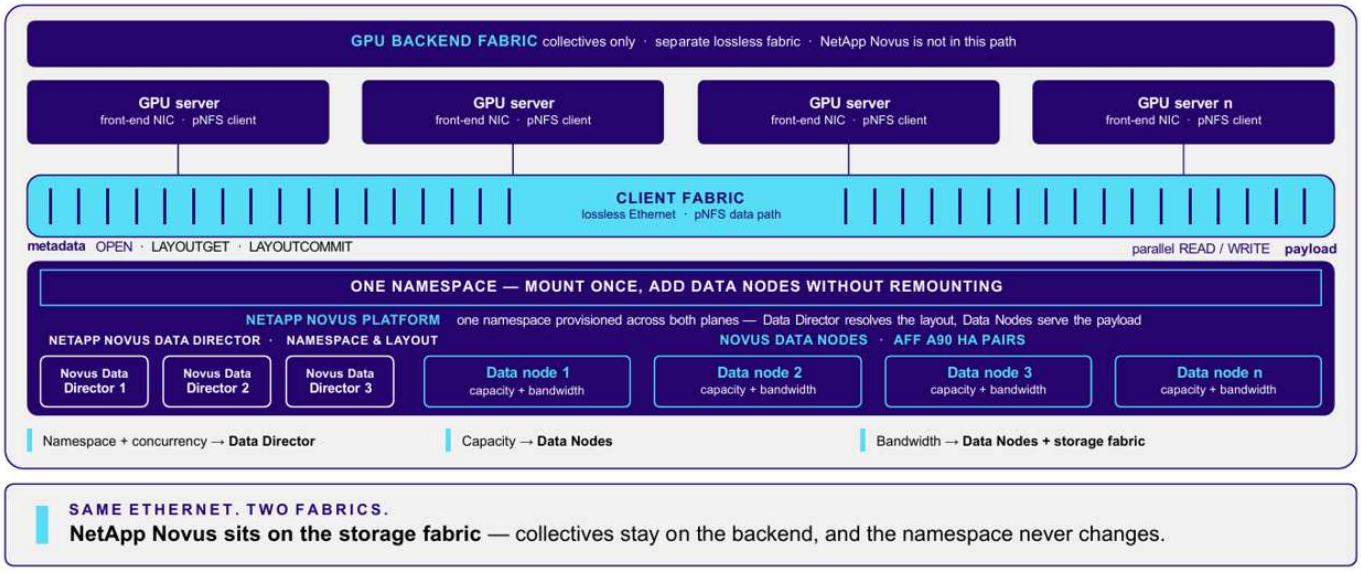
The NDD is implemented as software that runs on a cluster of at least three dedicated servers. It combines the metadata server processes with a replicated, distributed database that stores the namespace metadata on encrypted local storage. Each namespace runs its own metadata server processes on several of these

servers. All healthy servers announce the namespace's virtual IP address, which spreads the client load and lets clients fail over without remounting. Because the authoritative metadata lives in the replicated database, a server can fail and restart without losing the namespace state.

The placement of data on the ONTAP clusters is controlled by the NDD. It keeps a registry of data servers, ensures that each data server is available using lightweight NFS probes, and tracks the free capacity on each. When a file is created, the NDD chooses the data servers, creates the backing files over NFS, and records their locations. If a data server becomes unavailable, the NDD stops placing new data there and recalls client layouts that point to it.

Novus deployment

The following figure illustrates a typical Novus deployment at a high level. Notable aspects include a single namespace with multiple Novus Data Directors and ONTAP data server nodes. This architecture provides high availability and low latency in support of the backend GPU servers.



Novus storage system environment and processing

There are several important aspects of the Novus environment and its processing that you should understand.

Novus Data Director and ONTAP data server internal communication

The Novus Data Directors communicate with the ONTAP data servers to support file access and management. Most of this is initiated by the NDD rather than the ONTAP servers.

ONTAP data server discovery

The Novus Data Directors need to discover what data servers are available. ONTAP never initiates communication with the Novus Data Director. An operator or a tool registers each data server through the `mds.conf` configuration file, which is used at startup time; data servers can also be manually added using the command `mds-admin ds add`. Each data server is represented as a row in the distributed metadata store including its host, export, ports, state and capacity.

ONTAP data server availability

There are several processes used to determine the availability of each ONTAP data server.

Heartbeat

The MDS sends an NFSv3 NULL RPC call over TCP to each online data server. By default it does this every 5 seconds using a heartbeat message.

Going offline

After a set number of failures in a row, the data server is marked as offline. The MDS stops placing new files on it and informs clients to give back layouts that point to it.

Coming back

After enough successful probes in succession, it returns to online. A cool down period stops it from flipping back and forth.

Client errors don’t count

When a client reports an error it’s only logged. ONTAP data server health is decided by the probes alone.

ONTAP failover is invisible

If an ONTAP HA pair does a takeover, the MDS only notices through probe timeouts.

ONTAP data server data content

The NDD doesn’t ask ONTAP what it stores. It already knows, because it creates every backing file:

- When a file is created, the MDS picks the data servers, creates the backing files on them, and saves the result in the stripe map in the distributed metadata store. The stripe map records which data server and NFS file handle hold each stripe and mirror.
- Backing files on ONTAP have a specific name format in the NDD.
- Capacity is the only operational value the NDD reads from ONTAP using statvfs() on each mounted export.
- File size isn’t verified against ONTAP. It’s determined from the client’s layout commit.

High-availability and resiliency for the NDD cluster

The metadata plane anchored by the Novus Data Directors ensures high availability and resilience in the system.

Cluster design

Novus uses a multi-host cluster to ensure high availability and fault tolerance for the NDD services.

Three hosts with three failure domains

The minimum NDD cluster is three physical hosts or virtual machines (VMs) deployed on separate servers. Each host runs one MDS process and one metadata store node. The cluster is designed to survive the loss of one host.

All replicas serve

Every MDS replica handles clients at the same time with none designated in standby mode. All healthy replicas announce the same VIP through BGP. BFD (fast link-failure detection) is required because BGP’s own timers alone are not a completely acceptable failover mechanism. A replica announces the VIP only when its NFS service, gateway and metadata store connection are all ready for client handling.

Metadata replication

The distributed metadata store keeps three synchronous copies. An arbitrator prevents a split-brain scenario by choosing to go unavailable rather than letting two sides both write.

Restarts are safe

MDS processes keep no state of their own; the distributed metadata store holds the authoritative metadata.

Limitations and cautions

You should also be aware of the following limitations and cautions:

- Subtree ownership doesn’t fail over automatically. The namespace can be split into subtrees, each owned by one MDS. If an owner fails, its subtree hangs until the owner comes back, even though the VIP has already moved.
- One host failure affects every tenant on that host because each tenant has one replica per host.
- Clients must mount the VIP and not a host IP. Otherwise the client can’t fail over.
- Hard mounts may block during recovery. A blocked mount doesn’t mean data was lost.
- Always perform maintenance one replica at a time. Take the replica out of the VIP and drain it first.

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.