# NetApp and Red Hat Collaborate on pNFS

**Pranoop Erasani**
Technical Director, NFS

**Justin Parisi**
Technical Marketing Engineer

Access to shared data is critical to the performance of a variety of scientific, engineering, business, and financial applications. NFS—the most widely used standard for shared data access—can become a bottleneck for large-scale compute clusters, which can overwhelm file servers that are the single point of access for all files in a shared file system.

Most of the solutions available to provide high performance and shared data access have been more or less proprietary and have failed to gain the kind of heterogeneous system support and widespread adoption that standard protocols such as NFS have achieved.

Explore

**Learn More About pNFS**

Want to learn more about pNFS and how to deploy it to accelerate your IT environment? Check out the following links:

- pNFS and Future NFSv4.2 Features
- Using pNFS to Scale Out RHEL & KVM
- pNFS and NFSv4.2; a Filesystem for Grid, Virtualization and Database
- pNFS Solution Brief

The parallel NFS (pNFS) standard—a subfeature of the NFS version 4.1 protocol specification (RFC 5661)—addresses the single-server bottleneck and has great promise to become a standardized solution for parallel data access. In this article we'll explain how pNFS works, talk about efforts that NetApp and Red Hat are making to move pNFS forward, and describe how pNFS is implemented in clustered NetApp® Data ONTAP®.

## What is pNFS?

The pNFS protocol gives clients direct access to files striped across two or more data servers. By accessing multiple data servers in parallel, clients achieve significant I/O acceleration. The pNFS protocol delivers graceful performance scaling on both a per-client and per-file basis, without sacrificing backward compatibility with the standard NFS protocol; clients without the pNFS extension are still able to access data.

### pNFS Architecture and Core Protocols

The pNFS architecture consists of three main components:

- The **metadata server** handles all nondata traffic. It is responsible for maintaining metadata that describes where and how each file is stored.
- **Data servers** store file data and respond directly to client READ and WRITE requests. File data can be striped across a number of data servers.
- One or more **clients** are able to access data servers directly based on information in the metadata received from the metadata server.

Three types of protocols are used between the clients, metadata server, and data servers:

- A **control protocol** is used to synchronize the metadata server and data servers. This is not defined by the pNFS specification and varies from vendor to vendor.
- **pNFS protocol** is used between clients and the metadata server. This is essentially NFSv4 with a few pNFS-specific extensions. It is used to retrieve and manipulate layouts, which contain the metadata that describes the location and storage access protocol required to access files stored on multiple data servers.
- A set of **storage access protocols** is used by clients to directly access data servers. The pNFS standard currently defines three categories of storage protocols: file-based (RFC5661), block-based (RFC5663), and object-based (RFC5664). Clustered Data ONTAP currently supports the file-based storage protocol and uses NFSv4.1 to access the data servers.

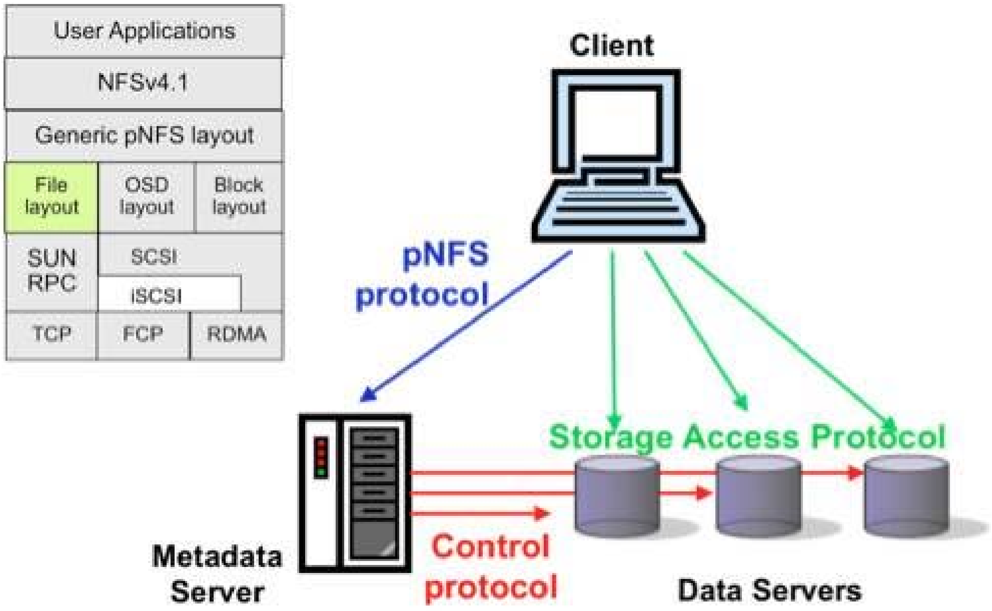


**Figure 1)** Elements of pNFS. Clients request layout from metadata server (pNFS protocol), and then access data servers directly (storage access protocol).

To access a file, a client contacts the metadata server to open the file and request the file's layout. Once the client receives the file layout, it uses that information to perform I/O directly to and from the data servers in parallel, using the appropriate storage access protocol without further involving the metadata server. pNFS clients cache the layout until they are done with the parallel I/O operations. pNFS servers have the right to revoke the layout of the file, if the server cannot promise parallel access to the servers. Further, pNFS does not modify the current mechanism available in the NFS server for metadata access.

## NetApp and Red Hat Team for pNFS

A pNFS solution needs both client and server components to function. NetApp and Red Hat have collaborated extensively with the upstream community to deliver the first standards-based end-to-end pNFS offering.

NetApp addresses the challenges of scale by combining storage clustering and pNFS. NetApp FAS and V-Series storage running clustered Data ONTAP 8.1 or later can scale from just a few terabytes of data to over 69 petabytes, all of which can be managed as a single storage entity, simplifying management of a pNFS environment and helping eliminate both planned and unplanned downtime.

With the first-to-market, fully supported pNFS client—delivered in Red Hat Enterprise Linux®— you can begin to plan and design next-generation, scalable file system solutions based on pNFS. Application workloads can take full advantage of pNFS without modification, allowing a seamless transition for existing applications.

## pNFS and Clustered Data ONTAP

NetApp implemented pNFS starting in clustered Data ONTAP 8.1. (There is no 7-Mode or Data ONTAP 7G implementation.) pNFS implemented in clustered Data ONTAP offers a number of advantages:

- **Simplified infrastructure.** The overall infrastructure for pNFS is simpler in comparison to other parallel file systems such as Lustre and GPFS, which require many dedicated servers in addition to storage.
- **Manageability.** Typically, pNFS includes multiple file servers that have to be managed separately. Clustered Data ONTAP lets you manage all the server-side pNFS components as a single system.
- **Nondisruptive operations.** A pNFS installation on a NetApp cluster benefits from nondisruptive operations for maintenance and load balancing—such as storage failover, LIF migrations, and nondisruptive volume moves—just like any other workload.
- **All nodes can act as metadata servers.** In the clustered Data ONTAP implementation, every node in the storage cluster can act as both a metadata server and a data server. This eliminates the potential bottleneck created by having a single metadata server and helps in distribution of metadata operations across the cluster.

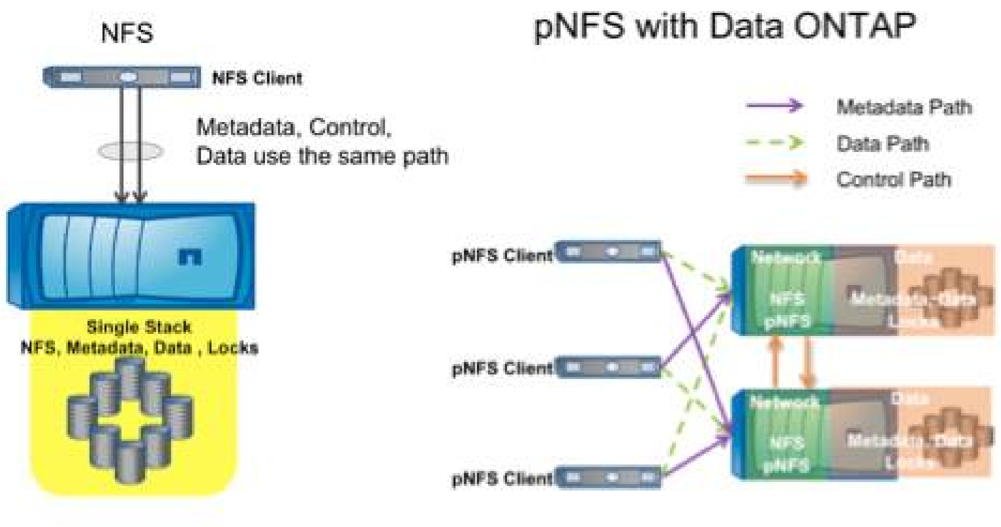


**Figure 2)** pNFS on Data ONTAP versus NFS. Every node can serve as both a metadata server and a data server.

To understand how pNFS works with clustered Data ONTAP, suppose that a client has mounted a pNFS file system from one node in a cluster. To access a file, it sends a metadata request to that node. The pNFS implementation gathers and returns information that includes location, layout of the file, and network information needed to reach the location. The client uses the information to access the data directly from the node or nodes where it resides. By providing a direct path to the volume, pNFS helps applications achieve higher throughput and lower latency.

pNFS seamlessly integrates with clustered Data ONTAP nondisruptive operations such as LIF migrate, storage failover, and Volume Move. When one of these operations occurs, the pNFS client and server automatically negotiate the new direct I/O path to the server, which helps keep throughput the same, all without any disruption to the application. This is a huge benefit for storage administrators, because they don't have to explicitly provision network paths to file systems when they do maintenance operations on the cluster. Thus pNFS with clustered Data ONTAP not only helps with performance, it also simplifies administrative workflows during maintenance operations. In provisioning and deploying large clusters, this becomes a necessity.

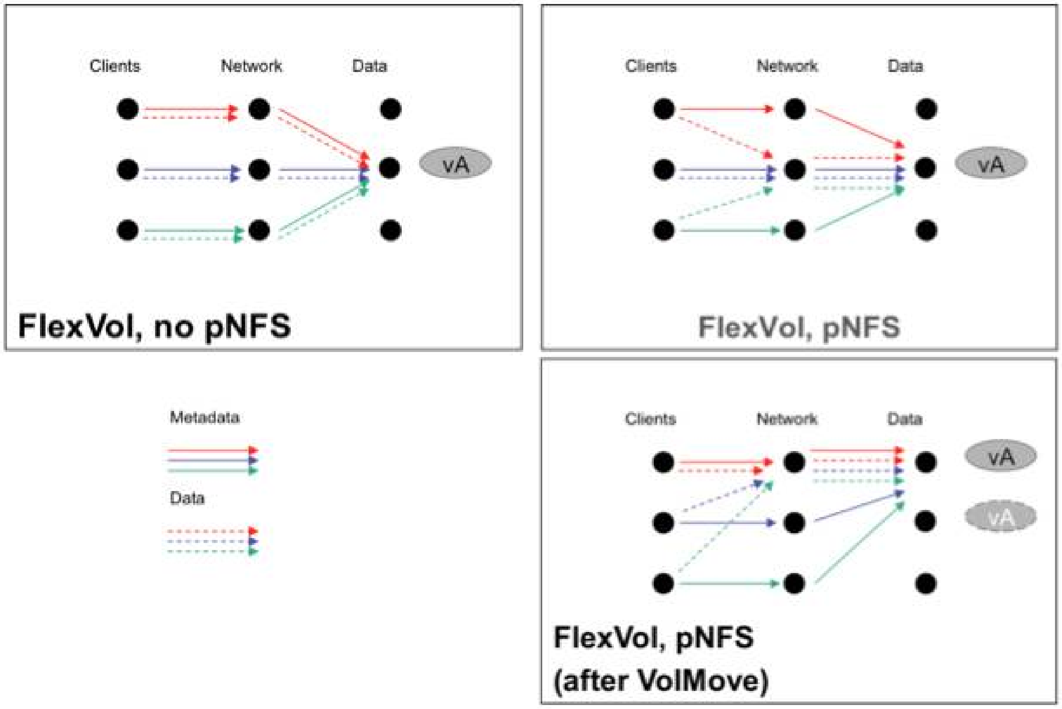


**Figure 3)** Without pNFS, both metadata and data paths are more or less static. With pNFS, metadata service is distributed across multiple nodes, while data paths are direct to the network interface of the node that is storing the file. When data moves, data paths adapt automatically to maintain optimal performance.

### Best Practices

Attention to a few best practices helps deliver the best pNFS performance:

- Consult the NetApp Interoperability Matrix for the latest compatibility information with NFSv4.1 and pNFS clients (requires NetApp support site access).
- Each cluster node that supports pNFS should be configured with at least one logical interface (LIF) so that pNFS clients can access the volumes stored on that node directly.
- For metadata-intensive workloads, pNFS clients should be configured so that mounts are distributed across all the nodes in the cluster so that they can all act as metadata servers. This can be accomplished by using an external, round-robin domain name server (DNS) or through on-box DNS load balancing in clustered Data ONTAP.

For more information about deploying pNFS on NetApp storage, refer to TR-4063.

## Red Hat pNFS Client

The Red Hat pNFS client was first released in the Red Hat Enterprise Linux (RHEL) version 6.2 kernel in 2011. RHEL 6.2 and RHEL 6.3 were both labeled as "Tech Preview" versions of pNFS.

RHEL 6.4, released in February 2013, included the first general availability version of pNFS. For complete information about using Red Hat clients with NetApp storage running either NFS or pNFS, see TR-3183. (This technical report is currently being revised, and may not be immediately available when this article is published. Be sure to check back.)

## pNFS Use Cases

In addition to its obvious applicability for highly parallel science and engineering applications, the unique capabilities of pNFS make it a good fit for a variety of enterprise use cases.

**Business-critical applications**

By definition, business-critical applications require the highest service levels. Storage bandwidth and capacity must grow seamlessly with server requirements. As NetApp storage volumes are transparently migrated to more powerful controllers in the NetApp cluster, the Red Hat Enterprise Linux pNFS client automatically follows the data movement, self-adjusts, and reoptimizes the data path. The net result is near-zero downtime with no server or application reconfiguration required.

**Multi-tenant storage solutions**

Having parallel data access means that multi-tenant, heterogeneous workloads benefit directly from pNFS. The data resides on the NetApp cluster and is not tied to a specific NetApp controller. With pNFS, the Red Hat Enterprise Linux servers find the optimal data path and automatically adjust for optimum throughput.

**Mixed clients and workloads**

NFSv4.1 and pNFS can provide flexibility for mounting the file system from anywhere in the cluster namespace. Clustered applications can be mounted over pNFS, while legacy applications can still be mounted over NFSv3. File systems that are exported from storage can have clients mounted over different flavors of NFS so that they can coexist without making any significant changes to the applications that access the data. This level of flexibility reduces the overhead of frequent change management.

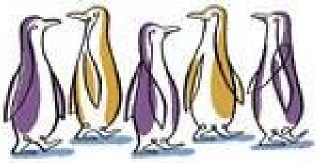
**Virtualization environments**

Hypervisors and virtual machines that use the Red Hat Enterprise Linux pNFS client are able to maintain multiple connections per session, which spreads the load across multiple network interfaces. Think of it as multipathing for NFS, without requiring a separate multipath driver or configuration.

## Conclusion

NetApp has been a major driver of both NFSv4.1 and pNFS, co-chairing the efforts of the working group. In addition, NetApp has authored and edited a significant portion of the NFSv4.1 specification. This is consistent with our commitment to tackle the problems of storage by using industry standards.

With the recent general availability of the pNFS client with the release of RHEL 6.4, you can now deploy pNFS for testing and/or production by using a combination of Red Hat clients and clustered NetApp Data ONTAP.



**By Pranoop Erasani, Technical Director, NFS and Justin Parisi, Technical Marketing Engineer**

Pranoop is the lead NFS architect in the NetApp Protocols Technology Engineering organization, where he leads NFS protocol development for Data ONTAP. He was instrumental in architecting pNFS for clustered Data ONTAP. Pranoop is a strong advocate for leveraging NFSv4.1/pNFS in clustered systems. He has participated in numerous discussions related to the pNFS IETF standard and frequently speaks at NFS interoperability events. He acts as a key technical advisor to technical marketing and product management for ongoing customer deployments and storage software solutions.

Justin is based in RTP and has spent the past 5 years in NetApp Global Support, as a technical support engineer and critical problem resolution escalation engineer. He focuses on clustered Data ONTAP and has developed troubleshooting course material as well as a number of knowledge base articles. His interests cover a wide range of areas, including CIFS, NFS, SNMP, OnCommand® System Manager, Unified Manager, SnapDrive®, and SnapManager®, as well as Microsoft® Exchange, SQL Server®, Active Directory®, LDAP, and more—making him, essentially, a Swiss Army knife of NetApp knowledge.

Quick Links

- Tech OnTap Community
- User Groups
- Archive
- PDF