

[UPFRONT]

[UPFRONT]

Parallel NFS (pNFS) Bridges to a Mature Standard

Thanks to the emergence of low-cost Linux clusters, high-performance computing (HPC) is no longer the domain solely of an elite group of public-sector-funded laboratories. In fact, HPC now can be found addressing challenges as diverse as simulating the behavior of the entire earth to the simulation needs of an individual product designer.

But, as clusters have become more prevalent, new challenges have emerged. The first challenge is to define a storage and I/O architecture that is not only capable of handling the vast amount of data created and consumed by these powerful compute engines, but that also is capable of keeping those engines fully utilized and fed with data. Without data, the largest and fastest supercomputers become nothing more than expensive space heaters.

The second challenge revolves around making the data generated by clusters easily available to other systems and users outside the cluster itself. Copying or moving data to other systems is clearly an option but involves inherent overhead cost and complexity. Ideally, any node on the network should be able to access and process data where it resides on the cluster.

Initially, clusters used the ubiquitous NFS standard, which has the advantages of being well understood, almost universally supported by many vendors and of providing easy access to data for systems and users outside the cluster. However, NFS moves all data and metadata through a single network endpoint (server) that quickly creates a bottleneck when trying to cater to the I/O needs of a cluster. The result is that neither bandwidth nor storage capacity scales—a new solution is required.

Parallel Filesystems and Parallel NFS

Parallel filesystems, which enable parallel access directly from server nodes to storage devices, have proven to be the leading solution to this scalability challenge. Although parallel filesystems are relatively new, the technology clearly will become an essential component of every medium-to-large-scale cluster during the next few years. Several parallel filesystem solutions are available today from vendors such as Panasas (ActiveScale PanFS), IBM (GPFS), EMC (HighRoad) and Cluster File Systems (Lustre).

Government, academic and Fortune 500 customers from all over the globe have embraced parallel filesystem solutions; however, these solutions require that customers lock in to a particular vendor for the software and sometimes the hardware. Wouldn't it be nice to have a filesystem that has the same performance as these vendor-specific solutions but that is also a true open standard? Then, you could reap the performance benefits of parallel access to your data while enjoying the flexibility and freedom of choice that come from deploying a universally accepted standard filesystem.

This introductory article discusses Parallel NFS (pNFS), which is being developed to meet these needs. pNFS is a major revamp to the NFS standard and has gained nearly universal support from the NFS community.

Parallel NFS Origins

When people first hear about pNFS, sometimes their initial reaction is that it is an attempt to shoehorn a parallel capability into the existing NFS standard. In reality, it is the next step in the evolution of NFS with the understanding that organizations need more performance while keeping it a multivendor standard. The NFSv4.1 draft standard contains a draft specification for pNFS that is being developed and demonstrated now.

Panasas is the author of the original pNFS proposal. Since this original proposal was written, a number of other vendors, notably EMC, IBM, Network Appliance and Sun, have joined to help define and extend pNFS. Other vendors are contributing as well, so pNFS is gaining broad momentum among vendors.

Because pNFS is an evolution of the NFS standard, it will allow organizations that are comfortable with NFS to achieve parallel performance with a minimum of changes. Plus, because it will become part of the NFS standard, it can be used to mount the cluster filesystem on the desktop easily.

Architecture of pNFS

NFSv4.0 improved the security model from NFSv3.0, which is the most widely deployed version today, and it folds in file locking that was previously implemented under a different protocol. NFSv4.0 has an extensible architecture to allow easier evolution of the standard. For example, the proposed NFSv4.1 standard evolves NFS to include a high-speed parallel filesystem. The basic architecture of pNFS is shown in Figure 1.

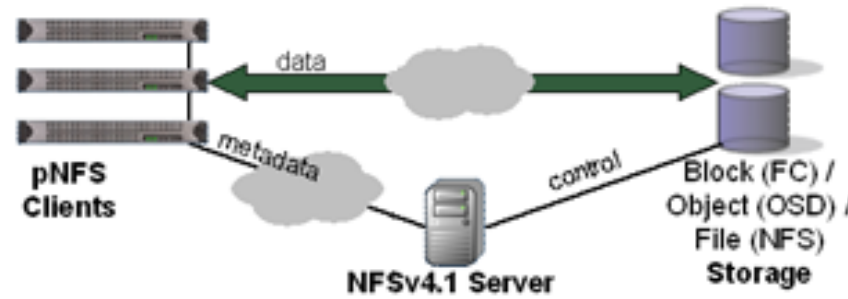


Figure 1. pNFS Architecture

The pNFS clients mount the filesystem. When they access a file on the filesystem, they make a request to the NFSv4.1 metadata server that passes a layout back to the client. A layout is an abstraction that describes where a file is located on the storage devices. Once the client has the layout, it accesses the data directly on the storage device(s), removing the metadata server from the actual data access process. When the client is done, it sends the layout back to the metadata server in the event that any changes were made to the file.

This approach may seem familiar, because both Panasas (ActiveScale PanFS) and Cluster File System (Lustre) use the same basic asymmetric metadata access approach with their respective filesystems. It is attractive because it gets the metadata server out of the middle of the data transaction to improve performance. It also allows for either direct or parallel data access, resulting in flexibility and performance.

Currently, three types of storage devices will be supported as part of pNFS: block storage (usually associated with SANs, such as EMC and IBM), object storage devices (such as Panasas and Lustre) and file storage (usually associated with NFS file servers, such as NetApp). The layout that is passed back to the client is used to access the storage devices. The client needs a layout driver so that it can communicate with any of these three storage devices or possibly a combination of the devices at any one time. These storage devices can be products such as an EMC SAN, a Panasas ActiveScale Storage Cluster, an IBM GPFS system, NetApp filers or any other storage systems that use block storage, object storage or file storage. As part of the overall architecture, it is intended to have standard, open-source drivers (layout drivers) for block storage, object storage and file storage back ends. There will be other back ends as well. For example, PVFS2 was used in the first pNFS prototype as the back-end storage.

How the data is actually transmitted between the storage devices and the clients is defined elsewhere. It will be possible for the data to be communicated using RDMA (Remote Direct Memory Access) protocols for better performance. For example, the InfiniBand SDP protocol could be used to transmit the data. The data can be transmitted using SCSI

Block Command (SBC) over Fibre Channel or SCSI Object-based Storage Device (OSD) over iSCSI or using Network File System (NFS).

The "control" protocol shown in Figure 1 between the metadata server and the storage is also defined elsewhere. For example, it could be an OSD over iSCSI.

The fact that the control protocol and the data transfer protocols are defined elsewhere gives great flexibility to the vendors. It allows them to add their value to pNFS to improve performance, improve manageability, improve fault tolerance or add any feature they want to address as long as they follow the NFSv4.1 standard.

Avoiding Vendor Lock-in

A natural question people ask is "how does the proposed pNFS standard avoid vendor lock-in?" One of the primary aspects of pNFS is that it has a common filesystem client regardless of the underlying storage architecture. The only thing needed for a specific vendor's storage system is a layout driver. This is very similar to how other hardware is used in Linux—you use a driver to allow the kernel to access the hardware.

Parallel NFS also works well for the vendors because it allows their storage to work with a variety of operating systems without porting their whole proprietary filesystem stack. Because NFSv4.1 will be a standard, the basic client would be available on a variety of operating systems as long as the OS had the client. The only piece the vendor would have to provide is the driver. Writing drivers is generally an easier process than porting and supporting a complete filesystem stack to various operating systems.

If you have a current parallel filesystem from one of the storage vendors, what does pNFS do for you that the vendor does not? Initially, pNFS is likely to perform more slowly than a proprietary filesystem, but the performance will increase as experience is gained and the standard pNFS client matures. More important, pNFS allows you to mount the filesystem on your desktop with the same performance that the cluster enjoys. Plus, if you want to expand your storage system, you can buy from any vendor that provides a driver for NFSv4.1. This allows your existing clients to access new storage systems just as your computers today access NFS servers from different vendors, using the filesystem client software that comes with your UNIX or Linux operating system.

Parting Comments

Parallel NFS is well on its way to becoming a standard. It's currently in the prototyping stage, and interoperability testing is being performed by various participants. It is hoped that sometime in 2007 it will be adopted as the new NFS standard and will be available in a number of operating systems.

If you want to experiment with pNFS now, the Center for Information Technology Integration (CITI) has some Linux 2.6 kernel patches that use PVFS2 for storage (www.citi.umich.edu/projects/asci/pnfs/linux).

—LARRY JONES