Data Mobility Group

Robin Harris
rharris@datamobilitygroup.com
May 30, 2007

Research Perspective

Parallel NFS:
Finally, NFS Optimized for Clusters

Scale-out high-performance I/O presages a new era in high-performance systems and applications. High-performance computing often requires data sets far larger than system memories or a single interconnect can accommodate. Compute clusters can process huge amounts of data, but storage I/O often becomes the bottleneck. A new IETF (Internet Engineering Task Force) standard called Parallel NFS (pNFS), with wide industry backing, is the first standards-based solution for high-performance I/O.

NAS or network-attached storage is the generic name for file services across a network. The first open NAS protocol, NFS (Network File System), originally developed by Sun, is still the workhorse storage protocol for large-scale and clustered computing. Intended for file sharing across networks of technical workstations, its purview now includes databases, supercomputing, webserving, and workgroups.

File server sales continue to grow more quickly than sales of block serving storage arrays for many reasons, including ease of management, shared access to dynamic data, higher storage utilization - better ROI - and the lower cost and greater flexibility of Ethernet infrastructure. Often preferred for cluster computing, whose sheer scale rules out more costly I/O infrastructures, Ethernet's performance has been a continuing bottleneck in the search for higher performance. That is about to change.

This paper discusses a major new technology refresh for NFS that will be released as the NFS v4.1 standard. This is the first performance improvement to NFS in many years and is a real reason to upgrade when NFS v4.1 becomes available later this year or early in 2008. The key new technology is parallel

NFS (pNFS). The pNFS standard enables high-speed parallel data transfers without requiring any changes to applications or operating systems. It can eliminate the storage bottlenecks in the current generation of NAS products and enable applications deployed on server clusters to scale as never before.

Parallelism in Servers and Storage

Compute nodes were the first to go parallel as witnessed by the accelerating deployment of Linux clusters. Substantial gains for high-performance commercial applications encouraged investment in larger clusters—until performance leveled off. The most common bottleneck: storage bandwidth.

High-performance applications with relatively modest I/O needs found some relief through technologies such as TCP offload engines, costly high-bandwidth NFS servers, or NFS v3-based cluster storage. Each of these technologies offloads or parallelizes a piece of the problem, but none is a completely parallel solution. This is why the NFS community is creating a standard for pNFS. The future of storage, like the future of servers, is parallel.

The original pNFS problem statement was written by Dr. Garth Gibson, a professor at Carnegie Mellon University and founder and chief technology officer of Panasas, a company specializing in parallel storage. (Dr. Gibson was also an author of the original paper on RAID architecture.) The pNFS concepts and architecture reflect the experience which Dr. Gibson and Panasas developed building parallel file servers for supercomputer and commercial high performance computing (HPC) applications. Panasas has deep expertise in parallel file systems and expects to be one of the first vendors to support pNFS. Network Appliance, EMC, IBM, and Sun are also likely to implement the new standard, as they are key members of the IETF pNFS committee.

10-gigabit Ethernet: Train Wreck or Savior?

With the advent of 10-gigabit Ethernet (gigE), one might expect that NFS performance will no longer be an issue. The network is 10 times faster, after all. Yet the problem isn't just the network, it is the NFS protocol.

NFS, developed in the era of 10-Mbit Ethernet, evolved to support 100-Mbit Ethernet and then gigabit Ethernet (gigE). And it scaled well from 10 Mbit to 100 Mbit, but not so well from 100 Mbit to 1000 Mbit. Scaling problems appeared with gigabit speeds. Users found that NFS performance with gigE over 100 Mbit grew only 2x to 4x, not 10x. IP processing overhead was one culprit. Handling multiple I/O requests while transmitting the file was another.

Many companies sought to solve the NFS-over-gigE problem. Some developed high-performance NAS servers with an optimized hardware pipeline to push a full gigabit of data. TCP/IP offload engines (TOEs) moved IP processing from the CPU to the host interface, but cost several times as much as standard gigabit host bus adapters (HBA). Clustered data servers of various architectures also grew in popularity.

Cluster-based NFS v3 file servers such as Isilon, Polyserve, and NetApp's OnTAP GX sought to drive performance over gigE through multiple gigE links. Although this does increase the total NAS bandwidth to multiple servers, any one file is still limited to the speed of a single gigE port—and that's assuming that the NAS server's compute capability is itself not creating a bottleneck.

In short, gigabit Ethernet has already pushed the current NFS protocol to its practical limits. Now comes 10-gigabit Ethernet—how on earth can today's NFS scale with that? Short answer: It can't and it won't.

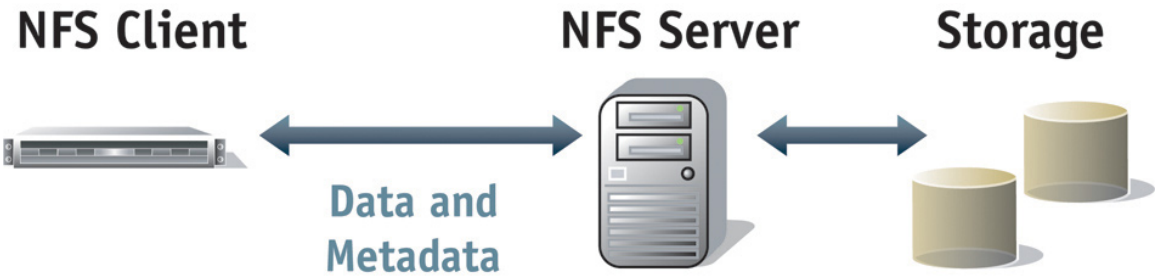
But computing technology has faced similar performance bottlenecks before. Conceptually, the choices are simple: Increase the speed of a single stream or move to parallel streams. We've already maxed out existing single-stream NFS technologies just to support full gigabit bandwidth. So what's left to deal with 10 gigabits? Parallel NFS.

Current NFS Infrastructure

Part of the popularity of NFS is its ease of use. Mount the file system and it accesses files as if they were local rather than on the network. The NFS client sees directories and files. When the client requests a file the NFS server sends it over the network to the client.

Today's NFS server stores not only the file but also the file metadata, including the file's length, type (.doc, .pdf, etc.), and creation date and which blocks on the storage system contain the file.

Storing the file and its metadata together has one major advantage: ease of implementation. The same storage that stores the file stores the metadata. One network connection supplies the file and its metadata. Changes to the file and/or its metadata are easily coordinated within a single NFS storage controller. For applications that are not storage-bandwidth-limited, current NFS implementations are fine.



Today's NFS Architecture

Yet the simplicity of standard NFS comes at a price. It is difficult to break a file into pieces for parallel delivery from multiple storage systems to multiple clients over the network. For large files, or for large numbers of clients, the network becomes a bottleneck.

That's where pNFS comes in.

Parallel NFS, the New Standard

A new version of the NFS standard has almost completed the IETF standards process with important new technology: parallel NFS. pNFS is in the v4.1 draft of the NFS standard and should reach final draft status in the second half of 2007. Final draft status is the final step before ratification and is typically the version vendors start implementing.

No Changes to Applications or Operating Systems

Designed to integrate seamlessly with existing infrastructures, pNFS requires no changes to existing applications or operating systems. NFS v4.1 clients simply replace existing NFS clients. If you have a v4.1 pNFS-enabled storage system, the new clients will start using it. If not, they'll operate in standard NFS client serial mode.

Backwards-compatible

More importantly, pNFS is backwards-compatible with existing NFS implementations. If an NFS v3 client wants to access the data from a pNFS-enabled cluster, it mounts the pNFS cluster just as it would an ordinary NFS filer. The metadata server gathers the data from the data servers and presents it to the NFS client just as any NFS filer would. The backward compatibility and interoperability make migrating to NFS V4.1 easy and painless.

Where Does pNFS Add Value?

Before looking at how pNFS works, let's look at where it adds value:

Large Files

Large files and cluster computing are common in geophysical modeling, finite element analysis, computational chemistry, high-energy physics, financial risk analysis, 3D animation and rendering, video editing, and other data-intensive workloads. If data-transfer times are a significant part of a workflow, pNFS can help.

As the name implies, pNFS moves data over parallel paths to the compute server. While most servers and workstations support two or more Ethernet ports, in most cases the most common client will be a server cluster, where multiple servers are clustered together to work on a single application in parallel. Parallel NFS is cluster I/O for cluster computing.

When a large file is requested, multiple pNFS storage systems respond with parts of the file in parallel, increasing the aggregate bandwidth from a single gigE link to many gigE links. The compute cluster's pNFS client, working in parallel, assembles the pieces into a complete file.

Many Concurrent Users

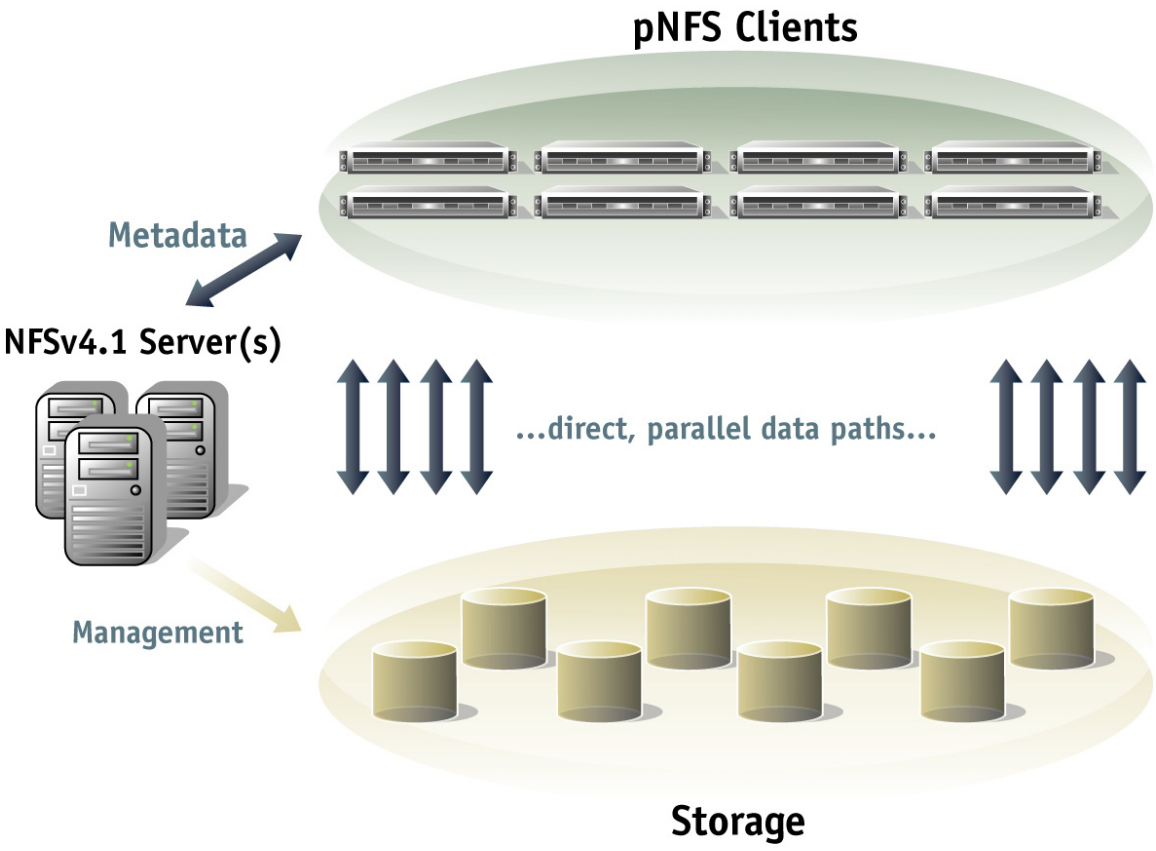
When many small files are requested, a single NFS server can be overwhelmed by contention for the disks or by processor overload. In a pNFS system, multiple sets of disks respond, spreading the workload across multiple storage servers. Multiple storage servers sharing the work means that scaling the number of clients is as easy as scaling the storage.

Architecture of pNFS

In principle, pNFS employs the parallelism used in RAID arrays. Storage arrays spread data across multiple disk drives to improve array performance. For example, a request for a large number of blocks is answered by several disk drives, speeding the data in parallel to the array controller. With pNFS, the data is spread across multiple pNFS storage servers instead of multiple disk drives.

Separating Control and Data

How do the storage systems know what data to send? The key is that pNFS-enabled storage separates data and metadata. The storage systems have the data. The metadata is stored on a pNFS metadata server that acts as a control node. The metadata server keeps track of where the data is stored and provides the intelligence of a filer head. This is what the architecture looks like:



Parallel NFS architecture

The pNFS metadata server is the single point of contact for the compute cluster, presenting a single namespace to users and ensuring coordination between storage systems in order to keep data available and protected. The control node is not in the data path and has minimal interaction with both the storage systems and the compute cluster nodes. This ensures that the pNFS metadata server does not become a bottleneck.

A Simple Example

Let's say a compute cluster needs to access a 1-terabyte file named foo.bar. It sends a request to the pNFS server for foo.bar. The pNFS server looks up the metadata, commonly stored in fast DRAM for performance, and sees that foo.bar is stored across 10 storage systems in 100-GB chunks.

The pNFS server sends "layout" of the foo.bar data—including data server addresses, data location, and striping information—to the pNFS client. Once the layout is received, the pNFS clients can interact directly with the data servers—in parallel—without involving the metadata server. Sending the layout gives the client permission to access the data servers directly and acts as a file-locking mechanism if read/write permission is needed.

pNFS Supports Blocks, Files, and Objects

Parallel NFS may be implemented using block-based iSCSI or Fibre Channel, gigE file storage, or something new—object-based storage. In all cases, pNFS is transparent to the client. The client asks for the file it wants and pNFS serves it up. However, the differences between the three storage-side architectures have implications for storage system performance and scalability.

Block. Storage devices are assumed to have limited intelligence and will read/write whatever is requested. In block storage the metadata server has to manage each individual block. With 1 billion blocks on a single 500 GB drive, the overhead quickly mounts up.

Studies have found that some 90% of metadata CPU cycles are consumed managing blocks. pNFS block support protects existing investments in Fibre Channel and iSCSI. The support also makes it easy to test pNFS.

File. Data is striped as files across multiple file servers. This requires that each of the file servers still act as both metadata and data storage for the files. This architecture performs well under many workloads. However, as the system and the workload scale up, or if security is important, this protocol may bottleneck sooner than the object-based protocol.

Object. Files are broken into smaller chunks called objects which are identified by unique numbers rather than traditional path names. This simplifies and speeds accessing the objects. In addition, objects have extensible metadata, which enables low overhead security techniques. For example, object servers will not accept commands or data unless they've been authorized by the metadata server to guard against accidental or malicious changes. This allows the storage system to scale in performance and capacity in the same manner as compute clusters.

You may never use all three, but it is good to know that the options are there if needed.

Conclusion

NFS v4.1 implements a major architectural enhancement to the already robust NFS protocol.

Separating the control and metadata from the data enables considerable bandwidth and concurrency gains through parallelism. The user and application interfaces are unchanged while the performance is dramatically improved. The new standard is interoperable and backward-compatible with earlier versions of the NFS standard.

Parallel NFS brings parallel storage to the mainstream and overcomes the performance and scalability limits of today's clustered NAS products. Parallel NFS is a huge advance for improving the performance of data-intensive applications, of the business processes that use them, and of the decision-makers who depend on those processes.

Storage buyers should put NFS 4.1 support on their shopping list to help ensure that the industry adds it quickly. Every buyer with high-performance requirements will stand to benefit.