NFS Extensions for Parallel Storage

Panasas Position

Garth Gibson, Benny Halevy, Brent Welch, 4 December, 2003

Summary

We believe that there is significant user and vendor benefit to be gained by standardizing NFSv4 extensions for exporting and controlling storage layout information that would enable a client of the extended NFSv4 server to directly access storage (out-of-band) without assistance from the server. We believe that such a mechanism could be general enough to support multiple storage access protocols, specifically SCSI SBC, SCSI OSD and non-parallel NFS, without significantly changing the functional model of an NFSv4 server. We liken such metadata to extended (or named) attributes that are semantically valid for out-of-band access only when the holding client also holds a delegation on the associated file.

Value Proposition for Out-of-Band Access: Striping all the way to the clients

Today's NFS moves all data for a given file, the files of a given directory or the files of most closely linked subtree of directories through a single IP address at a given time. Out-of-band, or parallel, NFS would enable NFS clients to access storage through multiple network addresses.

Multiple network ports, and the implied storage capacity and access hardware (e.g. hard drive heads and spindles) provides performance and manageability advantages for NFS clients.

Principle among these value propositions are:

- **Scalable bandwidth**: Concurrently transferring data from one file, or, on aggregate, concurrently transferring data from multiple closely linked files, scales file and file system bandwidth with the number of network ports employed.
- **Scalable capacity**: Distributing the data of one file, or the files of a closely linked set of files, scales the size of the distributed file or files with the storage containers accessed by a scaling number of network ports.
- **Load balancing**: Distributing the data portions of a file, or of a closely linked set of files, over multiple network links at a sufficiently fine grain of distribution is a simple scheme for probabilistically balancing the load on the network ports, links and associated storage hardware.
- **Capacity balancing**: Distributing the data portions of a file, or a closely linked set of files, at a sufficiently fine grain of distribution, over storage hardware associated with multiple network links is a simple scheme for exploiting the capacity of multiple storage containers such that free space is available to any and all files.

Depending on the implementation, multiple network addresses for one file or closely linked set of files can induce other values.

Scalable bandwidth and load balancing will **lower access latency** when load levels approach saturation for storage behind a single network port. Moreover, bypassing a store-and-forward node that aggregates many storage subsystems behind one network port saves access latency.

Enabling clients to directly access multiple storage subsystems, instead of accessing storage through an aggregating proxy, can also **improve cost-effectiveness**. As the load and bandwidth scale, the proxy cost need not scale because the multiple network ports of multiple storage subsystems can be directly accessed by clients.

With advantages like these, it is not surprising that many vendors offer non-NFS filesystems, including proprietary client software, exposing multiple network ports into storage. Examples include IBM's TotalStorage, EMC's High Road, SGI's CXFS, Sun's QFS, and Panasas' ActiveScale Storage Cluster, to name a few.

Equipping NFS with support for multiple network ports into storage offers another potential benefit. It offers a path for vendors to **standardize out-of-band client interfaces** and produce reference implementations, leading to multi-vendor interoperability and NFS-class market acceptance.

Panasas Inc., www.panasas.com, 6250 Kaiser Drive, Fremont, CA 94555. Authors email addresses are ggibson@panasas.com, bhalevy@panasas.com, bwelch@panasas.com.

Out-of-Band Access and NFSv4: pNFS

Basic Approach: Delegate file layout metadata

NFSv4 introduces recallable delegations allowing clients holding a delegation to locally make many decisions normally made by the server. A client holding a delegation can delay propagating state changes (e.g. written data) and does not have to revalidate cached data on every open.

We believe that NFSv4 delegations could be used to delegate file layout metadata to clients. Provided that such a client has direct network access to storage and a way to obtain a map of the file's locations on storage, then such a client can directly access the file's data on storage.

Clients that access storage without store-and-forward support from an NFS server put less load on the NFS server, enabling much more aggregate data traffic to be supported by the same NFS server. If the data of a single file or directory is striped over storage devices, then a single client can access multiple storage devices in parallel and move data at a bandwidth not supportable by a single NFS server. Striping is also a simple strategy to achieve approximately balanced distribution of data, which balances load and capacity consumption.

Support multiple kinds of network storage

Some proprietary file systems with multiple storage ports employ fixed block (SBC) SCSI command set over FC SANs. Others employ the emerging object storage (OSD) SCSI command set over iSCSI SANs. Still others spread directories, and maybe soon files, over component files on multiple NFS file servers.

We do not recommend picking a winner from these multiple approaches. All have advantages and disadvantages. Instead we recommend that all three storage command sets, SBC (on FCP/FC or iSCSI/TCP/IP/GE), OSD (on iSCSI/TCP/IP/GE) and NFS (on ONCRPC/TCP/IP/GE), be accommodated, and perhaps others in the future. A set of NFS extensions enabling out-of-band data access over all leading storage interfaces should have more longevity and market appeal.

Figure 1 illustrates our proposal. The extensions to NFSv4 we seek allow the NFSv4 server to be a metadata server for out-of-band storage access from clients. Like the VFS layer in UNIX, this exported metadata is logically a vnode, and it includes opaque storage format specific data, logically the underlying filesystem's inode data.

We propose that the NFSv4 protocol be extended so that a client can ask for permission to access the contents of a server's remote vnode (rvnode) provided that the associated remote inode (rinode) is of a type that the client supports (SBC, OSD, or NFS). Permission to access rvnode/rinode structures would be granted in the form of a delegation of the file, and enable use of a few new operations to transfer the contents of these rvnode/rinode structures.

Abstractly, these rvnode/rinode structures might be thought of as named attributes in the NFSv4 protocol. Out-of-band clients seek a delegation, then fetch rvnode/rinode named attributes of the delegated file that define the storage type, data layout and network addresses. While the delegation is held, the client is assured that the contents of the rvnode/rinode are valid and can be used for direct storage access –

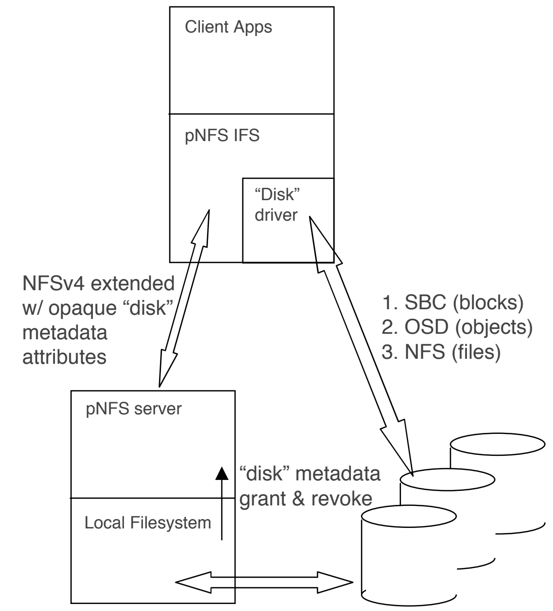

Figure 1. Out-of-band management of multiple flavors of network storage

reading and writing of file data. We are not necessarily recommending that rvnode/rinode information be implemented as named attributes and manipulated with attribute read and write; a new set of commands may be more efficient and a cleaner way to extend NFSv4.

A discovery and negotiation process will be needed for clients to declare their capability to handle out-of-band access in terms of code and storage network interfaces, discover the network storage format and address range, and test reachability from the client to all potentially addressable storage devices

Suggested NFS Extension Principles

Complimentary to RDMA activities

Remote Direct Memory Access support for NFS is a transport optimization for high efficiency and lower latency on a given connection between client and server. Making each connection more efficient is orthogonal and complimentary to endowing clients with the ability to address multiple storage subsystems out-of-band. Fortunately there are concurrent efforts for storage transports such as iSCSI over RDMA (iSER) as well as NFS over RDMA.

Start from NFSv4 and make minimal changes

While various vendors have non-NFS file systems with extensive use of multiple network ports, many of these implementations are dramatically different from NFS, in part because NFS compliance was not feasible, and, therefore, not a design goal.

As we apply support for multiple network ports to NFS storage, starting fresh from NFS and applying only necessary changes is the best way to approach standardization. Many people, vendors and users, understand NFS in detail, so the less we change the faster those changes can be understood, accepted, and widely supported.

NFSv4 server can do anything out-of-band can do

In a system whose clients are equipped to request and exploit out-of-band access to storage, it could easily become the case that the standard required each client to use out-of-band access all the time. We think that ensuring that any and all out-of-band work could have been done by the NFSv4 metadata server provides a more desirable approach.

Ensuring that the NFSv4 server can proxy for any work that can be done out-of-band provides

- inclusiveness for legacy clients,
- flexibility for client implementations to roadmap full exploitation of all capabilities,
- a simple way to allocate on-disk storage, using the NFSv4 server,
- a simple way to avoid complex client cache consistency algorithms — all clients will give up their delegations and rely on the server during concurrent write sharing semantics,
- a simple error handling strategy for out-of-band clients — repeat the work suffering an out-of-band error as inband work by the NFSv4 server.

Accordingly, we recommend that all out-of-band operations that a client can issue should be idempotent, so that repeated execution yields the same result.

Make out-of-band metadata consistent & opaque

As discussed in the previous section, it is commonplace for vnode data structures to contain other data structures uninterpreted at the vnode layer. What matters is that the vnode associate the file with code able to access it and the private data that this code needs to maintain.

Similarly we are proposing that NFSv4 extensions grant access to out-of-band metadata mapping information, and manage the consistency of this information, but, at least in the NFSv4 extension protocol, do not define it. While this mapping information is abstractly similar to on-disk inode data structures, we do not intend to suggest it is identical.

We recommend separate RFCs defining the format and meaning of each type of metadata map since the contents of these maps depend extensively on the underlying storage protocol, SBC, OSD or NFS. This emphasizes the role of management, not data path, for the NFSv4 protocol extension, and allows the timelines for each type of storage format to evolve at their own pace.

These out-of-band metadata defining RFCs would serve the role of “verbs,” to use a RDMA terminology. By defining the meaning of the storage specific metadata, we will implicitly sketch the functional capabilities of the storage specific driver code in pNFS client software. That is, we can avoid getting into the client software API game, while providing plenty of functional guidance. Similarly, these RFCs provide functional guidance to pNFS server implementations, since it is the storage layer of the server that creates and manipulates on-disk file representation. There is no need for RFCs for the storage interface commands and protocols because these are pre-existing ANSI (SCSI) and IETF (NFS) protocol.

Suggested NFS Extension Areas

Discovery

The parallel NFS client should be able to discover that a file or a hierarchy of files under a directory are accessible with out-of-band access.

Extensibility

The pNFS client should be able and to discover and perhaps negotiate the specific parallel access metadata transfer protocol version and metadata format (storage fabric and protocol type and version).

Security

Where applicable, the server should communicate with storage elements to allow or disallow specific client access to stored data. For example, pNFS clients using OSD storage should be able to exploit OSD's opaque per-object capability, to be provided along with out-of-band metadata maps, to enforce access restrictions in the storage components.

Layout

An operation to get a map describing the layout and location of a contiguous logical byte range in the file.

Allocation

An operation to allocate storage to the file.

Panasas Inc., www.panasas.com, 6250 Kaiser Drive, Fremont, CA 94555. Authors email addresses are ggibson@panasas.com, bhalevy@panasas.com, bwelch@panasas.com.