INTERNET-DRAFT
Expires: January 2005

Garth Gibson
Panasas Inc. & CMU
Peter Corbett
Network Appliance, Inc.

Document: draft-gibson-pnfs-problem-statement-01.txt July 2004

pNFS Problem Statement

Status of this Memo

Copyright Notice

Abstract

This draft considers the problem of limited bandwidth to NFS servers. The bandwidth limitation exists because an NFS server has limited network, CPU, memory and disk I/O resources. Yet, access to any one file system through the NFSv4 protocol requires that a single server be accessed. While NFSv4 allows file system migration, it does not provide a mechanism that supports multiple servers simultaneously exporting a single writable file system.

This problem has become aggravated in recent years with the advent of very cheap and easily expanded clusters of application servers that are also NFS clients. The aggregate bandwidth demands of such clustered clients, typically working on a shared data set preferentially stored in a single file system, can increase much more quickly than the bandwidth of any server. The proposed solution is to provide for the parallelization of file services, by enhancing NFSv4 in a minor version.

Table of Contents

1. Introduction

The storage I/O bandwidth requirements of clients are rapidly outstripping the ability of network file servers to supply them. Increasingly, this problem is being encountered in installations running the NFS protocol. The problem can be solved by increasing the server bandwidth. This draft suggests that an effort be mounted to enable NFS file service to scale with its clusters of clients. The proposed approach is to increase the aggregate bandwidth possible to a single file system by parallelizing the file service, resulting in multiple network connections to multiple server endpoints participating in the transfer of requested data. This should be

achievable within the framework of NFS, possibly in a minor version of the NFSv4 protocol.

In many application areas, single system servers are rapidly being replaced by clusters of inexpensive commodity computers. As clustering technology has improved, the barriers to running application codes on very large clusters have been lowered. Examples of application areas that are seeing the rapid adoption of scalable client clusters are data intensive applications such as genomics, seismic processing, data mining, content and video distribution, and high performance computing. The aggregate storage I/O requirements of a cluster can scale proportionally to the number of computers in the cluster. It is not unusual for clusters today to make bandwidth demands that far outstrip the capabilities of traditional file servers. A natural solution to this problem is to enable file service to scale as well, by increasing the number of server nodes that are able to service a single file system to a cluster of clients.

Scalable bandwidth can be claimed by simply adding multiple independent servers to the network. Unfortunately, this leaves to file system users the task of spreading data across these independent servers. Because the data processed by a given data-intensive application is usually logically associated, users routinely co-locate this data in a single file system, directory or even a single file. The NFSv4 protocol currently requires that all the data in a single file system be accessible through a single exported network endpoint, constraining access to be through a single NFS server.

A better way of increasing the bandwidth to a single file system is to enable access to be provided through multiple endpoints in a coordinated or coherent fashion. Separation of control and data flows provides a straightforward framework to accomplish this, by allowing transfers of data to proceed in parallel from many clients to many data storage endpoints. Control and file management operations, inherently more difficult to parallelize, can remain the province of a single NFS server, inheriting the simple management of today's NFS file service, while offloading data transfer operations allows bandwidth scalability. Data transfer may be done using NFS or other protocols, such as iSCSI.

While NFS is a widely used network file system protocol, most of the world's data resides in data stores that are not accessible through NFS. Much of this data is stored in Storage Area Networks, accessible by SCSI's Fibre Channel Protocol (FCP), or increasingly, by iSCSI. Storage Area Networks routinely provide much higher data bandwidths than do NFS file servers. Unfortunately, the simple array of blocks interface into Storage Area Networks does not lend itself

to controlling multiple clients that are simultaneously reading and

writing the blocks of the same or different files, a workload usually referred to as data sharing. NFS file service, with its hierarchical namespace of separately controlled files, offers simpler and more cost-effective management. One might conclude that users must chose between high bandwidth and data sharing. Not only is this conclusion false, but it should also be possible to allow data stored in SAN devices, FCP or iSCSI, to be accessed under the control of an NFS server. Such an approach protects the industry's large investment in NFS, since the bandwidth bottleneck no longer needs to drive users to adopt a proprietary alternative solution, and leverages SAN storage infrastructures, all within a common architectural framework.

2. Bandwidth Scaling in Clusters

When applied to data-intensive applications, clusters can generate unprecedented demand for storage bandwidth. At present, each node in the cluster is likely to be a dual processor, with each processor running at multiple GHz, with gigabytes of DRAM. Depending on the specific application, each node is capable of sustaining a demand of 10s to 100s of MB/s of data from storage. In addition, the number of nodes in a cluster is commonly in the 100s, with many instances of 1000s to 10,000s of nodes. The result is that storage systems may be called upon to provide an aggregate bandwidth of GB/s ranging upwards toward TB/s.

The performance of a single NFS server has been improving, but it is not able to keep pace with cluster demand. Directly connected storage devices behind an NFS server have given way to disk arrays and networked disk arrays, making it now possible for an NFS server to directly access 100s to 1000s of disk drives whose aggregate capacity reaches upwards to PBs and whose raw bandwidths range upwards to 10s of GB/s.

An NFS server is interposed between the scalable storage subsystem and the scalable client cluster. Multiple NIC endpoints help network bandwidth keep up with DRAM bandwidth. However, the rate of improvement of NFS server performance is not faster than the rate of improvement in each client node. As long as an NFS file system is associated with a single client-side network endpoint, the aggregate capabilities of a single NFS server to move data between storage networks and client networks will not be able to keep pace with the aggregate demand of clustered clients and large disk subsystems.

3. Clustered Applications

Large datasets and high bandwidth processing of large datasets are increasingly common in a wide variety of applications. As most

computer users can affirm, the size of everyday presentations, pictures and programs seems to grow continuously, and in fact average file size does grow with time [Ousterhout85, Baker91]. Simple copying, viewing, archiving and sharing of even this baseline use of growing files in day-to-day business and personal computing drives up the bandwidth demand on servers.

Some applications, however, make much larger demands on file and file system capacity and bandwidth. Databases of DNA sequences, used in bioinformatics search, range up to tens of GBs and are often in use by all cluster users are the same time [NIH03]. These huge files may experience bursts of many concurrent clients loading the whole file independently.

Bioinformatics is an example of extensive search in science application. Extensive search is much broader than science. Wall Street has taken to collecting long-term transaction record histories. Looking for patterns of unbilled transactions, fraud or predictable market trends is a growing financial opportunity [Agarwal95, Senator95].

Security and authentication are driving a need for image search, such as face recognition [Flickner95]. Databasing the faces of approved or suspected individuals and searching through many camera feeds involves huge data and bandwidths. Traditional database indexing in these high dimension data structures often fails to avoid full database scans of these huge files [Berchtold97].

With huge storage repositories and fast computers, huge sensor capture is increasingly used in many applications. Consumer digital photography fits this model, with photo touch-up and slide show generation tools driving bandwidth, although much more demanding applications are not unusual.

Medical test imagery is being captured at very high resolution and tools are being developed for automatic preliminary diagnosis, for example [Afework98]. In the science world, even larger datasets are captured from satellites, telescopes, and atom-smashers, for example [Greiman97]. Preliminary processing of a sky survey suggests that thousand node clusters may sustain GB/s storage bandwidths [Gray03]. Seismic trace data, often measured in helicopter loads, commands large clusters for days to months [Knott03].

At the high end of science application, accurate physical simulation, its visualization and fault-tolerance checkpointing, has been estimated to need 10 GB/s bandwidth and 100 TB of capacity for every thousand nodes in a cluster [SGPFS01].

Most of these applications make heavy use of shared data across many clients, users and applications, have limited budgets available to fund aggressive computational goals, and have technical or scientific users with strong preferences for file systems and no patience for tuning storage. NFS file service, appropriately scaled up in capacity and bandwidth, is highly desired.

In addition to these search, sensor and science applications, traditional database applications are increasingly employing NFS servers. These applications often have hotspot tables, leading to high bandwidth storage demands. Yet SAN-based solutions are sometimes harder to manage than NFS based solutions, especially in databases with a large number of tables. NFS servers with scalable bandwidth would accelerate the adoption of NFS for database applications.

These examples suggest that there is no shortage of applications frustrated by the limitations of a single network endpoint on a single NFS server exporting a single file system or single huge file.

4. Existing File Systems for Clusters

The server bottleneck has induced various vendors to develop proprietary alternatives to NFS.

Known variously as asymmetric, out-of-band, clustered or SAN file systems, these proprietary alternatives exploit the scalability of storage networks by attaching all nodes in the client cluster to the storage network. Then, by reorganizing client and server code functionality to separate data traffic from control traffic, client nodes are able to access storage devices directly rather than requesting all data from the same single network endpoint in the file server that handles control traffic.

Most proprietary alternative solutions have been tailored to storage area networks based on the fixed-sized block SCSI storage device command set and its Fibrechannel SCSI transport. Examples in this class include EMC's High Road (www.emc.com); IBM's TotalStorage SAN FS, SANergy and GPFS (www.ibm.com); Sistina/Redhat's GFS (www.readhat.com); SGI's CXFS (www.sgi.com); Veritas' SANPoint Direct and CFS (www.veritas.com); and Sun's QFS (www.sun.com). The Fibrechannel SCSI transport used in these systems may soon be replaceable by a TCP/IP SCSI transport, iSCSI, enabling these proprietary alternatives to operate on the same equipment and IETF protocols commonly used by NFS servers.

While fixed-sized block SCSI storage devices are used in most file systems with separated data and control paths, this is not the only

alternative available today. SCSI's newly emerging command set, the Object Storage Device (OSD) command set, transmits variable length storage objects over SCSI transports [T10-03]. Panasas' ActiveScale storage cluster employs a proto-OSD command set over iSCSI on its separated data path (www.panasas.com). IBM's research is also demonstrating a variant of their TotalStorage SAN FS employing proto-OSD commands [Azagury02].

Even more distinctive is Zforce's File Switch technology (www.zforce.com). Zforce virtualizes a CIFS file server spreading the contents of a file share over many backend CIFS storage servers and places their control path functionality inside a network switch in order to have some of the properties of both separated and non-separated data and control paths. However, striping files over multiple file-based storage servers is not a new concept. Berkeley's Zebra file system, the successor to the log-based file system developed for RAID storage, had a separated data and control path with file protocols to both [Hartman95].

5. Eliminating the Bottleneck

The restriction of a single network endpoint results from the way NFS associates file servers and file systems. Essentially, each client machine "mounts" each exported file system; these mount operations bind a network endpoint to all files in the exported file system, instructing the client to address that network endpoint with all requests associated with all files in that file system. Mechanisms intended for primarily for failover have been established for giving clients a list of network endpoints associated with a given file system.

Multiple NFS servers can be used instead of a single NFS server, and many cluster administrators, programmers and end-users have experimented with this alternative. The principle compromise involved in exploiting multiple NFS servers is that a single file or single file system is decomposed into multiple files or file systems, respectively. For instance, a single file can be decomposed into many files, each located in a part of the namespace that is exported by a different NFS server; or the files of a single directory can be linked to files in directories located in file systems exported by different NFS servers. Because this decomposition is done without NFS server support, the work of decomposing and recomposing and the implications of the decomposition on capacity and load balancing, backup consistency, error recovery, and namespace management all fall to the customer. Moreover, the additional statefulness of NFSv4 makes correct semantics for files decomposed over multiple services without NFS support much more complex. Such extra work and extra problems are

usually referred to as storage management costs, and are blamed for causing a high total cost of ownership for storage.

Preserving the relative ease of use of NFS storage systems requires solutions to the bandwidth bottleneck that do not decompose files and directories in the file subtree namespace.
A solution to this problem should continue to use the existing single network endpoint for control traffic, including namespace manipulations. Decompositions of individual files and file systems over multiple network endpoints can be provided via the separated data paths, without separating the control and metadata paths.

6. Separated control and data access techniques

Separating storage data flow from file system control flow effectively moves the bottleneck away from the single endpoint of an NFS server and distributes it across the bisectional bandwidth of the storage network between the cluster nodes and storage devices. Since switch bandwidths of upwards of terabits per second are available today, this bottleneck is at least two orders of magnitude better than that of an NFS server network endpoint.

In an architecture that separates the storage data path from the NFS control path there are choices of protocol for the data path. One straightforward answer is to extend the NFS protocol so it can accommodate can be used on both control and separated data paths. Another straightforward answer is to capture the existing market's dominant separated data path, fixed-sized block SCSI storage. A third alternative is the emerging object storage SCSI command set, OSD, which is appearing in new products with separate data and control paths.

A solution that accommodates all of these approaches provides the broadest applicability for NFS. Specifically, NFS extensions should make minimal assumptions about the storage data server access protocol. The clients in such an extended NFS system should be compatible with the current NFSv4 protocol, and should be compatible with earlier versions of NFS as well. A solution should be capable of providing both asymmetric data access, with the data path connected via NFS or other protocols and transports, and symmetric parallel access to servers that run NFS on each server node. Specifically, it is desirable to enable NFS to manage asymmetric access to storage attached via iSCSI and Fibre Channel/SCSI storage area networks.

As previously discussed, the root cause of the NFS server bottleneck is the binding between one network endpoint and all the files in a

file system. NFS extensions can allow the association of additional

network endpoints with specific files. These associations could be represented as layout maps [Gibson98]. NFS clients could be extended to have the ability to retrieve and use these layout maps.

NFSv4 provides an excellent foundation for this. We may be able to extend the current notion of file delegations to include the ability to retrieve and utilize a file layout map. A number of ideas have been proposed for storing, accessing, and acting upon layout information stored by NFS servers to allow separate access to file data over separate data paths. Data access can be supported over multiple protocols, including NFSv4, iSCSI, and OSD.

7. Security Considerations

Bandwidth scaling solutions that employ separation of control and data paths will introduce new security concerns. For example, the data access methods will require authentication and access control mechanisms that are consistent with the primary mechanisms on the NFSv4 control paths. Object storage employs revocable cryptographic restrictions on each object, which can be created and revoked in the control path. With iSCSI access methods, iSCSI security capabilities are available, but do not contain NFS access control. Fibre Channel based SCSI access methods have less sophisticated security than iSCSI. These access methods typically use private networks to provide security.

Any proposed solution must be analyzed for security threats and any such threats must be addressed. The IETF and the NFS working group have significant expertise in this area.

8. Informative References

[Afework98] A. Afework, M. Beynon, F. Bustamonte, A. Demarzo, R. Ferriera, R. Miller, M. Silberman, J. Saltz, A. Sussman, H. Tang, "Digital dynamic telepathology - the virtual microscope," Proc. of the AMIA'98 Fall Symposium 1998.

[Agarwal95] Agrawal, R. and Srikant, R. "Fast Algorithms for Mining Association Rules" VLDB, September 1995.

[Azagury02] Azagury, A., Dreizin, V., Factor, M., Henis, E., Naor, D., Rinetzky, N., Satran, J., Tavory, A., Yerushalmi, L, "Towards an Object Store," IBM Storage Systems Technology Workshop, November 2002.

[Baker91] Baker, M.G., Hartman, J.H., Kupfer, M.D., Shirriff, K.W. and Ousterhout, J.K. "Measurements of a Distributed File System" SOSP, October 1991.

[Berchtold97] Berchtold, S., Boehm, C., Keim, D.A. and Kriegel, H. "A Cost Model For Nearest Neighbor Search in High-Dimensional Data Space" ACM PODS, May 1997.

[Fayyad98] Fayyad, U. "Taming the Giants and the Monsters: Mining Large Databases for Nuggets of Knowledge" Database Programming and Design, March 1998.

[Flickner95] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. and Yanker, P. "Query by Image and Video Content: the QBIC System" IEEE Computer, September 1995.

[Gibson98] Gibson, G. A., et. al., "A Cost-Effective, High-Bandwidth Storage Architecture," International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 1998.

[Gray03] Jim Gray, "Distributed Computing Economics," Technical Report MSR-TR-2003-24, March 2003.

[Greiman97] Greiman, W., W. E. Johnston, C. McParland, D. Olson, B. Tierney, C. Tull, "High-Speed Distributed Data Handling for HENP," Computing in High Energy Physics, April, 1997. Berlin, Germany.

[Hartman95] John H. Hartman and John K. Ousterhout, "The Zebra Striped Network File System," ACM Transactions on Computer Systems 13, 3, August 1995.

[Knott03] Knott, T., "Computing colossus," BP Frontiers magazine, Issue 6, April 2003, http://www.bp.com/frontiers.

[NIH03] "Easy Large-Scale Bioinformatics on the NIH Biowulf Supercluster," http://biowulf.nih.gov/easy.html, 2003.

[Ousterhout85] Ousterhout, J.K., DaCosta, H., Harrison, D., Kunze, J.A., Kupfer, M. and Thompson, J.G. "A Trace Drive Analysis of the UNIX 4.2 BSD FIle System" SOSP, December 1985.

[Senator95] Senator, T.E., Goldberg, H.G., Wooten, J., Cottini, M.A., Khan, A.F.U., Klinger, C.D., Llamas, W.M., Marrone, M.P. and Wong, R.W.H. "The Financial Crimes Enforcement Network AI System (FAIS): Identifying potential money laundering from reports of large cash transactions" AIMagazine 16 (4), Winter 1995.

[SGPFS01] SGS File System RFP, DOE NNCA and DOD NSA, April 25, 2001.

[T10-03] Draft OSD Standard, T10 Committee, Storage Networking Industry Association(SNIA), ftp://www.t10.org/ftp/t10/drafts/osd/osd-r08.pdf

9. Acknowledgments

David Black, Gary Grider, Benny Halevy, Dean Hildebrand, Dave Noveck, Julian Satran, Tom Talpey, and Brent Welch contributed to the development of this problem statement.

10. Author's Addresses

Garth Gibson
Panasas Inc, and Carnegie Mellon University
1501 Reedsdale Street
Pittsburgh, PA 15233 USA
Phone: +1 412 323 3500
Email: ggibson@panasas.com

Peter Corbett
Network Appliance Inc.
375 Totten Pond Road
Waltham, MA 02451 USA
Phone: +1 781 768 5343
Email: peter@pcorbett.net

11. Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.