ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 1, January 2013

Experimental Evaluation of pNFS Protocol using Network Direct Attached Hard Disk Drives for Mass Storage System

Han-gyoo Kim[1], Kee-cheol Lee[2]

Faculty of Computer Engineering Department, Hongik University, Seoul, Korea[1, 2]

ABSTRACT: **Experiments were performed to investigate the storage I/O performance of pNFS file system protocol using network direct attached hard disk drives as its storage subsystem. Comparison experiments were done to NFS4.0 and NFS4.1 with pNFS in order to further investigate the scalability of the two network file systems. It is found that NFS4.1 preserves the scalability as storage capacity is increased while NFS4.0 loses its scalability rapidly as its storage capacity is increased. It is also found that NFS4.1 shows, on average, over 4 times higher I/O speed than NFS4.0 when both systems uses the same set of network direct attached hard disk drives in our experiments.**

Keywords:**mass storage system, parallel NFS, cloud storage, network file system, NFS4.1**

I. INTRODUCTION

We are entering the era of "Big Data". Cloud computing[10] and multimedia data are probably the two major contributors for the phenomenon. IDT research reported in 2011 that the whole world is now facing unprecedentedly huge amount of data of 1.8 Zettabytes (or 1.8 Billion Terabytes) to be stored and its growth rate is faster than doubling every two years only to hit 7.9 Zettabytes in 2015. More and more applications like cloud computing and multimedia SNS demand insatiable amount of storage space, and it is not uncommon to see a storage server that has over 100 Terabytes or even 1,000 Terabytes of hard disk drives in a single enclosure.

As a consequence of the demands for ever-increasing storage space, many new file systems for huge capacity storage service have emerged from academia and industries. PVFS2[1], Lustre[2], and GDFS[3] are some of the examples of high performance parallel file systems designed to cope with the needs for high performance huge capacity storage servers.

While such parallel file systems exhibit their merits including performance, expandability, and scalability, they, on the other hand, suffer from their own short comings that they are expensive to implement, are not based on standard technologies hindering them from being accepted wide spread, and few of them has been verified for its long term reliability in real field applications.

NFS[n], as the only network file system standard for the Internet, has long served for more than a quarter of a century its purposes as reliable data storage server for various types of applications. Despites its undistinguished performance and less stringent data consistency policy, NFS has edges over many other distributed file systems thanks to its simplicity, economy, reliability, and openness due to standardization.

However, the NFS server's intrinsic architectural bottleneck of trafficking data to and from its clients has posed serious questions whether NFS can continue to satisfy the storage requirements in today's Big Data ecosystems.

As is well known, NFS server suffers from the lack of scalability as its storage space increases. Adding more hard disk drives to an NFS server does not alone lend itself a scalable storage server. Despite of the increased total number of hard disk drives, the total aggregated I/O bandwidth of the NFS server does not increase at all.

In fact, the total aggregated I/O bandwidth of an NFS server actually diminishes sharply due to various system overheads as the number of hard disk drives attached to its internal bus of the NFS server increases. Any bus based architecture cannot have scalability even though additional resources can be attached to the bus because the total I/O bandwidth of the system will be bounded by the limit of the bandwidth of its bus system no matter how many additional devices can be attached to the bus. So the traditional bus based NFS server does not scale in terms of storage space and its due I/O performance.

Recognizing the NFS server bottleneck issues, NFS communities have developed pNFS protocol[4] that takes advantages of parallel accesses to the pool of storage subsystems over the network and announced NFS4.0[n] and NFS4.1[5],[6],[7],[8] which can be considered as

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 1, January 2013

evolution of conventional NFS utilizing pNFS protocol in them. A number of research papers have been published that investigated the performance of NFSv4 (4.0 and 4.1) since its birth, it is yet to be verified whether NFS with pNFS can be effectively applied to build scalable and high speed storage servers, especially using economic off-the-shelf networked direct attached hard disk drives.

In this paper, we performed a series of experiments to investigate the I/O performance and scalability of NFSv4.1 in order to find out the feasibility of pNFS protocols for mass storage system using network direct attached hard disk drives.

II. NFS 4.1 AND PNFS

Originally developed by Sun Microsystems in 1984, NFS, the distributed file system protocol standard of the Internet, has evolved through version upgrades, NFSv2[n], NFSv3, and NFSv4. The single most distinct feature of NFSv4 from the previous versions is to incorporate SAN FC architecture into it so that storage subsystems can be accessed over the network instead of being attached to the internal system bus of an NFS server.

NFSv4.0's SAN-like architecture resolves the issues of storage capacity expansion by allowing storage devices be available to the server over the network, thus making NFS servers appealing to many newly emerging applications that require huge storage space including email repository and video archiving. However, NFSv4.0 still suffers from limited I/O bandwidthproblem[15] because it is practically impossible for single NFS server to provide I/O bandwidth high enough to satisfy the aggregate storage I/O bandwidth requirement of a cluster of clients. NFSv4.0 server can provide huge pool of storage devices to a cluster of clients, but it does not scale, that is, NFSv4.0 server does not provide storage aggregate I/O bandwidth *proportional* to the number of storage devices deployed in the server system.

IETF later came up with NFSv4.1 in order further to exploit NFS clients' parallel and direct accesses to the cluster of storage devices. Latest RFC on NFSv4.1 was announced in 2010.

As depicted in Fig. 1, parallel NFS (pNFS) is a part of the NFS v4.1 standard that allows NFSv4.1 clients to accessdirectly and in parallel storage devices that are attached to the network as in SAN. The pNFS architecture eliminates the scalability and performance issues associated with NFS servers of lower versions up to NFSv4.0. This is achieved by the separation of data and metadata, and moving the metadata server out of the data path. Fig. 1 shows the structural diagram how pNFS works in NFSv4.1.

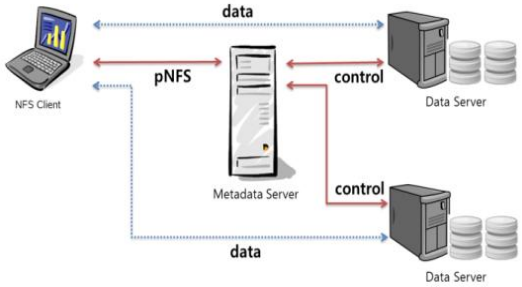


Fig.1 NFSv4.1 System Architecture

Data are stored in data servers in stripes thus distributed fashion depending on the configuration of the data servers. NFSv4.1 server called as MDS (metadata server) stores and controls metadata of the files stored in data servers. NFS clients communicate with MDS according to pNFS protocol such that individual client requests MDS to send *layout* of the file to/from which the client wants to write or read. Each client then maps the exact block location on the data servers from the layout and performs read/write operations directly and in parallel to the storage servers bypassing the MDS for the data access.

Notice that NFSv4.0 and v4.1 can have three categories of storage servers, i.e., block devices, object filers, and file servers providing different levels of data access services to the clients. A number of researches on the performance experiments of NFSv4.1 have been reported, but they were mostly on NFSv4.1 performance using file servers or object filers as its storage devices. Few has been reported how NFSv4.1 performs when it uses block devices as its storage devices although NFSv4.1 servers with off-the-shelf block devices such as inexpensive SATA hard disk drives would have the best performance/price ratio compared to other types of NFSv4.1 servers using expensive object filers and file servers as their storage devices.

It is expected that NFSv4.1 delivers high performance for various applications and allows massive scalable storage without diminished performance. Among a number of researches, Hildebrand and Honeyman applied pNFS to PVFS distributed file system and performed comparison studies on NFVv4 servers[n]. Two years later in 2007, Hildebrand et al. applied pNFS to GPFS file system to investigate the performance and scalability[n]. NetApp, a company headquartered in California, USA, implemented their pNFS server and release the source code to the public in 2008. Most of the researches have been done on the pNFS servers that use file servers as their storage devices.

However, it is yet to be verified through the real field experimental studies what degree of scalability and how high the aggregate I/O bandwidth particular types of NFSv4.1 servers with block device layout, instead of with file server layout, will have.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 1, January 2013

III. OUR EXPERIMENT ENVIRONMENT

In our experiments of which system configuration is similar to the system diagram shown in Fig. 1, we used network direct attached hard disk drives (NDAS drives) from Ximeta, Inc.[n] for storage devices in NFSv4.1 server. NDAS drive is conceptually similar to iSCSI drive except it uses efficient proprietary communication protocol based on Ethernet data link protocol instead of TCP in iSCSI. Besides, NDAS drive is implemented using cost effective SATA hard disk drive instead of expensive SCSI drive. This high performance/price ratio was the major factor why we chose NDAS drives as our devices instead of iSCSI or AoE devices for our experiments.Our MDS and client hosts were installed on Intel i3 based machines. Table 1 summarizes installation details of our experimental environment.

TABLE I
EXPERIMENT ENVIRONMENT

Component	Specification
MDS	CPU: Intel i3-2100 (3.1 GHz) Memory: 4GB OS: Linux 3.2.0
NFS 4.1 Clients	CPU: Intel i3-2100 (3.1 GHz) Memory: 4GB OS: Linux 3.2.0
Networked HDD Array	NDAS Network Direct Attached Disk Array SATA2, 7200rpm, 32MB buffer

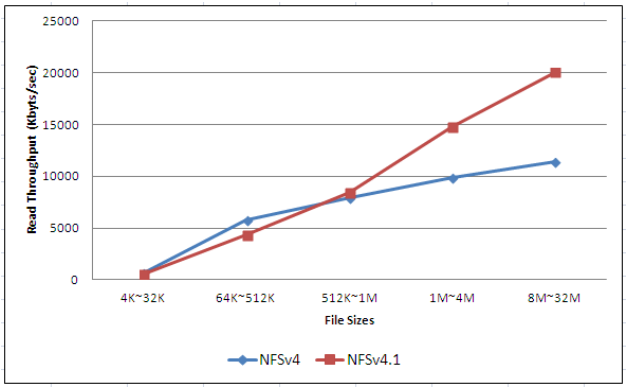
NFS4.1 clients obtain layouts from the NFS4.1 MDS(metadata server) before they can directly access the disk blocks on the array of networked hard disks that are attached to the Gigabit Ethernet. Individual networked hard disks have their own Ethernet links connecting the hard disk drive to the Ethernet port thus providing 1 Gbps of data link to and from each hard disk drive.

IV. PERFORMANCE EVALUATION

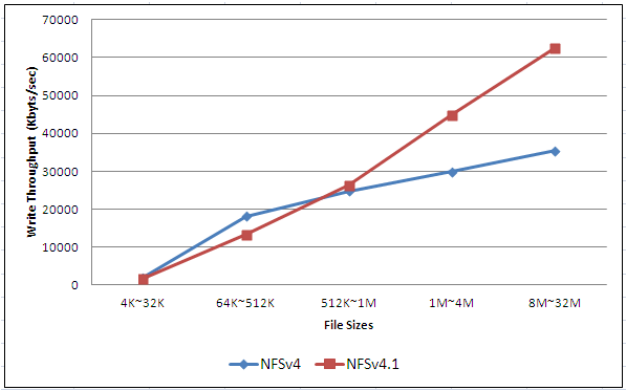
We performed I/O speed testing of the storage system of our experimental setup in order to find out if the storage system scales.We used two widely known benchmarks, Postmark[12] and IOzone[13] to measure the performance of NFSv4.0 and NFSv4.1. The variations of I/O bandwidth due to the various sizes of the data files were observed using Postmark, and IOzone was used to find out the effects of number of processes that access the files concurrently.

A. Benchmarking with Postmark

Fig. 2 shows that NFSv4.1 exhibits better I/O bandwidth than NFSv4.0 when accessing for reading or writing the files of which sizes are larger than 1MB measured using Postmark benchmark[12]. These results can be explained that parallelism in data paths in accessing the data starts to show the positive effects when the total size of files grows bigger because the overheads incurred in transferring layout messages from server to clients cancel out the benefits from data parallelism if the size of the data is small. Such performance degeneration in NFS4.1 in small sized data accesses has been reported by some of the previous researches [14].



(a) Read Performance



(b) Write Performance

Fig. 2Postmark I/O Performance of NFSv4.1

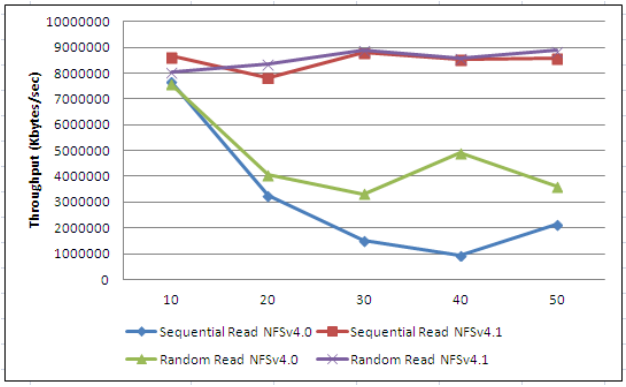
B. Benchmarking with IOzone

Unlike the experiments we performed using Postmark benchmark suites, we, in IOzone benchmarking[13], fixed the size of the files to be 30MB in the benchmarking, and investigated the scalability of the storage systems when the total number of concurrent I/O processes increases from 10 to 50. As can be seen in Fig. 3(a), NFS v4.1 exhibits strong scalability such that with 50 of concurrent I/O processes there is virtually no per process I/O bandwidth. On the other hand with NFS v4.0, there is sharp degrade in I/O bandwidth per process as the total number of concurrent processes increases from 10 to 50, to be exact, 28% drop in sequential read and 48% drop in

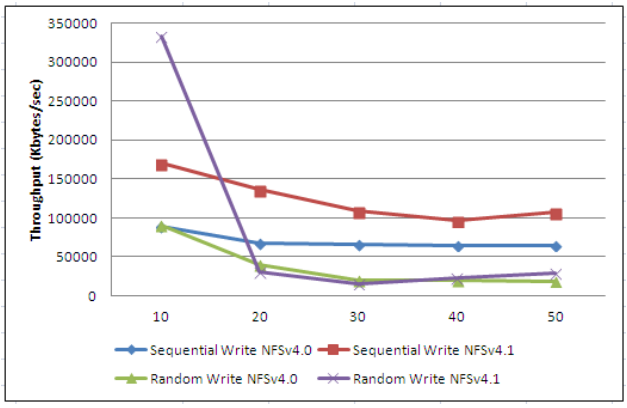
ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 1, January 2013

random read. However, NFS v4.1 shows not too much improvement over NFS v4.0 under random write workloads, which requires further research why random writes is less susceptible to parallelism in data paths.



(a) Read Performance



(b) Write Performance

Fig. 3IOzone Performance of NFSv4.1

V. CONCLUSION

In this experimental study to evaluate the overall I/O performances of pNFS[8], we constructed a large scale disk storage system in order to compare the I/O characteristics of NFS v4.0 and NFS v4.1.

The results from extensive experiments under various types of different I/O workloads show that NFS v4.1 has better overall I/O bandwidth over the span of various sized file read/write accesses except with very small sized files. This result is expected due to the parallelism in data paths intrinsic in the disk storage subsystem of NFS v4.1 while NFS v4.0 has no such parallel data paths between the processes and disk arrays over the network.

Further researches, however, are required in order to investigate I/O behaviours in more detail under various types of I/O workloads, especially under the realistic workloads including real world video streaming where read sharing of huge size video files are common among tens of concurrent processes and heavily crowded Cloud storage systems where there is a strong mixture of small and medium sized file I/O's by thousands of independent processes sharing a limited number of file systems at the same time.

Acknowledgment

This work was supported by Hongik University 2012 Research Fund.

REFERENCES

[1] Parallel Virtual File System -Version2: http://www.pvfs.org..
[2] Lustre a network clustering FS: http://wiki.lustre.org.
[3] F. Schmuck and R. Haskin, "GPFS: A shared-Disk File System for Large Computing Clusters," Proceedings of the 1st USENIX Conference on File and Storage Technologies, USENIX, pp. 234-238, 2002.
[4] Parallel Network File System: http://www.pnfs.com
[5] S. Shepler, Ed., M. Eisler, Ed. and D. Noveck, Ed., "Network File System (NFS) Version 4 Minor Version 1 Protocol," RFC5661, IETF, 2010.
[6] S. Shepler, Ed., M. Eisler, Ed. and D. Noveck, Ed., "Network File System (NFS) Version 4 Minor Version 1 External Data Representation (XDR) Description," RFC5662, IETF, 2010.
[7] B. D. Black, S. Fridella and J. Glasgow, "Parallel NFS (pNFS) Block/Volume Layout," RFC5663, IETF, 2010.
[8] B. Halevy, B. Welch and J. Zelenka, "Object based Parallel NFS (pNFS) Operations," RFC5664, IETF, 2010.
[9] D. Hildebrand and P. Honeyman, "Exporting storage systems in a scalable manner with pNFS," Proceedings of 22nd IEEE/13th NASA Goddard Conference On Mass Storage Systems and Technologies, IEEE, pp. 402-410, 2005.
[10] D. Hildebrand, P. Honeyman and W. A. Adamson, "pNFS and Linux: WorkingTowardsa Heterogeneous Future," Proceedings of 8th LCI International Conference on High-Performance Clustered Computing, pp. 101-108, 2007.
[11] D. Muntz, M. Sager, and R. Labiaga," spNFS: A Simple pNFS Server, " 2008. http://www.con-nectathon.org/talks08/dmuntz-spnfs-cthon08.pdf
[12] J. Katcher, "PostMark: A New File System Benchmark," Technical Report TR3022, Network Appliance, 1997.
[13] IOzoneFilesystem Benchmark: http://www.iozone.org
[14] D. Hildebrand, L. Ward, and P. Honeyman, "Large files, small writes, and pNFS," Proceedings of the 20th Annual International Conference on Supercomputing, ACM, pp. 116–124, 2006.
[15] G. Gibson and P. Corbett, "pNFS Problem Statement," Internet-Draft, July 2004, http://www.pdl.cmu.edu/pNFS/archive/gibson-pnfs-problem-statement.html

Biography

Han-gyoo Kim Prof. H. Kim was born in Seoul, Korea in 1959. He received B.S. degree in mechanical engineering from SeoulNationalUniversity in 1981, and his Ph. D in computer science from University of California at Berkeley in 1994. Since August 1994, he has been on the faculty of computer engineering department, HongikUniversity, Seoul, Korea. His areas of research include networked storage systems, scalable information retrieval systems, and high speed large scale big data systems.

.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 1, January 2013

Kee-cheol Lee Prof. K. Lee was born in Seoul, Korea on Feb. 21, 1955. He received a BS degree in electronic engineering from Seoul national University in 1977, a MS degree in computer science from Korea Advanced Institute of Science in 1979, and a Ph.D degree in electrical and computer engineering from University of Wisconsin-Madison in 1987. Since March 1989, he has been on the faculty of computer engineering department, HongikUniversity, Seoul, Korea, and currently he is a professor. His academic and research interests cover the fields of artificial intelligence, machine learning, and information retrieval.