pNFS/PVFS2 over InfiniBand: Early Experiences

LEI CHAI, XIANGYONG OUYANG, RANJIT NORONHA, AND D. K. PANDA

Technical Report
OSU-CISRC-10/07-TR69

pNFS/PVFS2 over InfiniBand: Early Experiences *

Lei Chai Xiangyong Ouyang Ranjit Noronha Dhabaleswar K. Panda

Department of Computer Science and Engineering
The Ohio State University
{*chail, ouyangx, noronha, panda*}*@cse.ohio-state.edu*

1. Introduction

Commodity clusters are a popular cost effective platform for both High Performance Computing (HPC) and datacenters. Large scale clusters running long running scientific applications use and generate terabytes and in some cases petabytes of data. Satellite imagery, oceanography and a variety of other fields generate petabytes of data, which is must be stored and accessed in a efficient manner.

Network File System (NFS) is currently being used as a ubiquitous standard in most clusters. It has several advantages and one of the most important is that it is an open standard and any vendors can pick up and have their own implementations that are interoperable with others. While served sufficiently well in the past, NFS has revealed performance problem as the size of clusters scales. The main reason is that NFS is a single server model which becomes a bottleneck in a large scale cluster. In this situation, researchers have proposed parallel file systems that decouple the data and metadata paths and distribute data to multiple storage servers and allows clients to access storage servers in parallel, such as PVFS2 [4], Lustre [3], etc. These parallel file systems have shown very good performance. However, the lack of a standard protocol makes interoperability an issue. And the clients often need to be reinstalled when deploying a new type of back-end file system on the server.

pNFS has been proposed to bridge these gaps. It is an extension to NFSv4 that allows clients to access multiple storage servers directly and in a parallel manner thus eliminating the single server bottleneck. Since it is still NFS, it facilitates interoperability. pNFS currently supports three types of data layouts - blocks, files, and objects. The pNFS clients can essentially access different types of back-end file systems in a transparent manner. And when a new type of file system emerges, the clients will just need to install the layout driver for the new file system. Previous work has focused on pNFS performance using a PVFS2 layout driver [6, 8, 7]. where PVFS2 used TCP over Gigabit Ethernet as the underlying transport. The limited bandwidth of Gigabit Ethernet in concert with the state and copying overhead of multiple TCP connections imposed natural bounds on the stripping width and scalability of large horizontally scaled parallel file systems. With InfiniBand being one of the most popular high performance networks for clusters and pNFS proposed as the next generation parallel file system solution for HPC, it is important to have a comprehensive study on pNFS performance over InfiniBand. In this paper we evaluate pNFS with a PVFS2 layout driver; where PVFS2 uses InfiniBand as the underlying transport fabric. To the best of our knowledge this is the first such study in the literature. Specifically, we want to answer these questions:

- Can the performance of contemporary single server NFSv4 deployments be improved by using a parallel file system like PVFS2 as the underlying file system and what are the limits to this approach?
- What are the advantages of InfiniBand over Gigabit ethernet in a parallel file system environment?
- Is there overhead introduced by the pNFS PVFS2 layout driver compared to a native PVFS2 installation and what are the relative merits of one approach over the other?
- Is there a fundamental limit exist in the PVFS2 layout driver approach which can be exposed by existing workloads?

Our experiments show that when pNFS is used through PVFS2 over InfiniBand Read throughput increased by up to 170% compared with a similar setup of pNFS that used Gigabit Ethernet. It adds very little overhead and achieves

*This research is supported in part by DOE's Grants#DE-FC02-06ER25749 and #DE-FC02-06ER25755; NSF's Grants #CNS-0403342 and #CPA-0702675; grants from Intel, Mellanox, Cisco systems, Linux Networx and Sun Microsystems; and equipment donations from Intel, Mellanox, AMD, Apple, Appro, Dell, Microway, PathScale, IBM, SilverStorm and Sun Microsystems.

almost the same throughput as native PVFS2. Compared with the traditional NFS, pNFS through PVFS2 provides significantly higher throughput and shows better scalability. From our experience, we believe that pNFS on InfiniBand clusters is promising.

The rest of the paper is organized as the following: Section 2 discusses the background of pNFS, PVFS2, and InfiniBand. The experiment architecture is illustrated in Section 3. Section 4 presents our preliminary experimental results and the design of the additional experiments that we will do for the final version of this paper. We conclude and and point our future directions in Section 5.

2. Background

In this section we discuss the background of pNFS, PVFS2, and InfiniBand.

Parallel NFS (pNFS) is an extension to NFSv4 that separates metadata and data paths, and allows clients to access storage devices directly and in parallel. pNFS has been proposed to eliminate the single server bottleneck associated with the current NFS servers and is being standardized by IETF [9]. pNFS data operation protocol supports three storage layouts - blocks, files, and objects. There are pNFS projects being developed on both Linux and Solaris [10] operating systems. In this paper we focus on the Linux implementation by the CITI group at University of Michigan [5].

Parallel Virtual File System version 2 (PVFS2) [4] is a high performance parallel file system designed for clusters. PVFS2 provides multiple interfaces, including PVFS2 specialized interface, VFS interface through a Linux kernel module, and MPI-IO via ROMIO. PVFS2 supports multiple networks as the transport, such as Ethernet, Myrinet, and InfiniBand.

InfiniBand [1] is a high performance and RDMA-capable network for interconnecting both processing nodes and I/O nodes. It is widely deployed in clusters to achieve low latency and high throughput for communication among nodes. The InfiniBand standard supports single data rate (SDR), double data rate (DDR), and quad data rate (QDR), which provides bandwidth equal to 10Gbps, 20Gbps, and 40Gbps, respectively.

3. Experimental Setup

Figure 1 shows the three configurations used - pNFS with a PVFS2 layout driver (pNFS/PVFS2), PVFS2 with a VFS mount (PVFS2) and finally; an NFSv4 server using a PVFS2 file system as the backend (NFSv4/PVFS2). In these configurations, InfiniBand (IB) or Gigabit Ethernet (GigE) is used as the network transport for PVFS2. For simplicity we only show one client in Figure 1.

4. Experimental Setup

Each node has dual 2.66GHz Intel Xeon processor and 2GB main memory, and is equipped with a Mellanox MT23108 InfiniBand HCA (SDR) on a 133 MHz PCI-X bus. The nodes run Linux kernel 2.6.17 with pNFS support. All nodes use OpenFabrics stack OFED 1.2.

The PVFS2 setup consists of 1 node as the metadata server and 4 nodes as IO servers. The metadata server exports this PVFS2 file system through pNFS and NFSv4 respectively. 4 nodes are used as clients.

We run IOzone [2] with clustering mode in this cluster to measure the aggregated write/read throughput. All IOzone test threads are evenly distributed across the client nodes. We let PVFS2 stripe size be 2MB, IOzone record size be 512KB, and IOzone file size be 64MB.

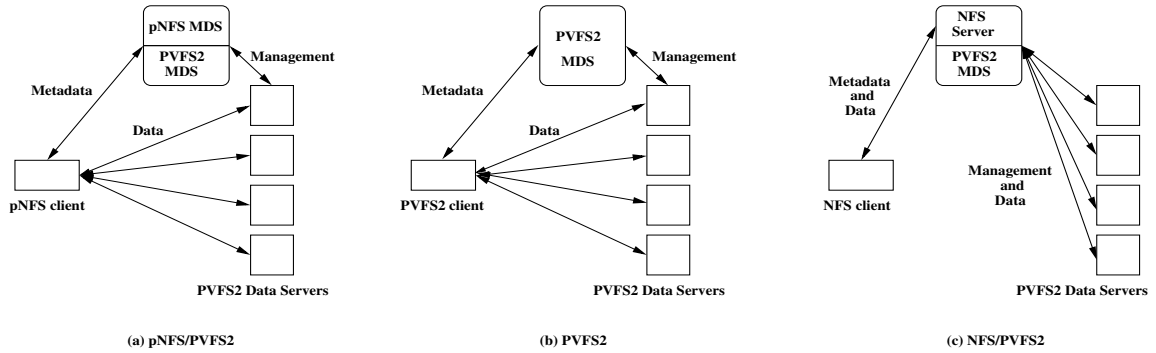
4.1 Impact of InfiniBand on pNFS/PVFS2 performance

In the first experiment, we compare the performance of pNFS/PVFS2 on InfiniBand with TCP/IP. Figure 2(a) and figure 2(b) shows that pNFS achieves a better throughput InfiniBand compared with that on TCP/IP. InfiniBand is about 10% better than TCP/IP in terms of write in this experimental configuration. In terms of read performance, InfiniBand outperforms TCP/IP by about 1.7 times at 16 client threads.

4.2 Comparison of pNFS, PVFS2 and NFSv4

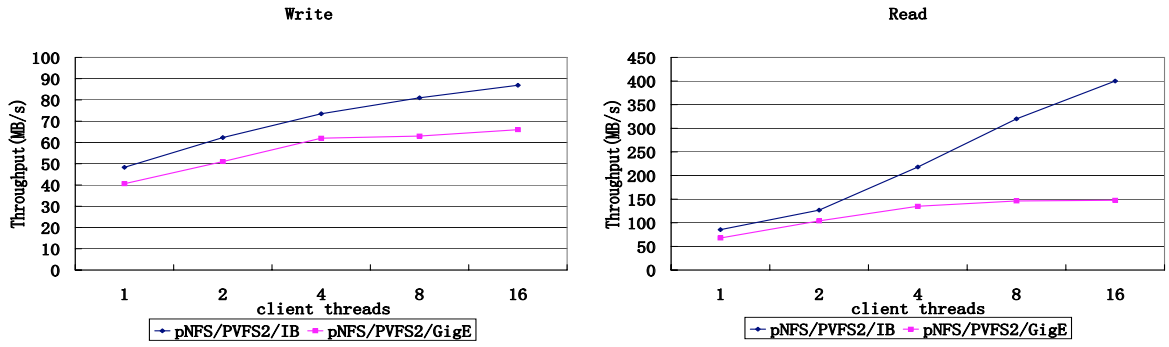
In this test we measured performance of PVFS2, pNFS/PVFS2 and NFSv4/PVFS2 running on InfiniBand transport. We used a disk based and memory based back-end file systems at the storage nodes. Figure 3(a) shows the write performance results, and figure 3(b) gives read performance results. These results show that pNFS closely matches the performance of PVFS2, and scales up with the back-end PVFS2. We also see that pNFS achieves much better performance than NFSv4. In NFSv4, all data I/O has to concentrate at the single metadata server before data are sent to client threads, which makes the metadata server a bottleneck in the system. On the contrary, pNFS enables a client to directly fetch data from data servers, thus effectively eliminating the single server bottleneck in NFSv4. pNFS may potentially scale up well with the back-end PVFS2 file system.

This test also shows that ramfs-based storage largely outperforms disk-based storage in write performance test. In disk-based storage the back-end PVFS2 writes data directly to disk, so disk speed becomes the highest constraint in write throughput. On the other hand, ramfs-based storage stores data in memory instead of writing them to disk. Its



(a) pNFS/PVFS2 (b) PVFS2 (c) NFS/PVFS2

Figure 1. The three configurations used in the experiments



(a) Write Performance (b) Read Performance

Figure 2. Comparative performance of pnfs/PVFS2 on InfiniBand and TCP/IP

performance is only subject to network performance, network protocol stack overhead etc. instead of disk speed. When it comes to read, disk-based storage achieves a similar performance to ramfs-based storage. After write test, data is temporarily buffered in memory in disk-based storage, and immediate following read test will get data directly from memory of the storage nodes. So the situation is basically the same in disk-based storage as in ramfs-based storage in this test scenario. Again we notice that pNFS achieves a scalable performance that is very close to the backend PVFS2.

4.3 Additional Results for the camera ready

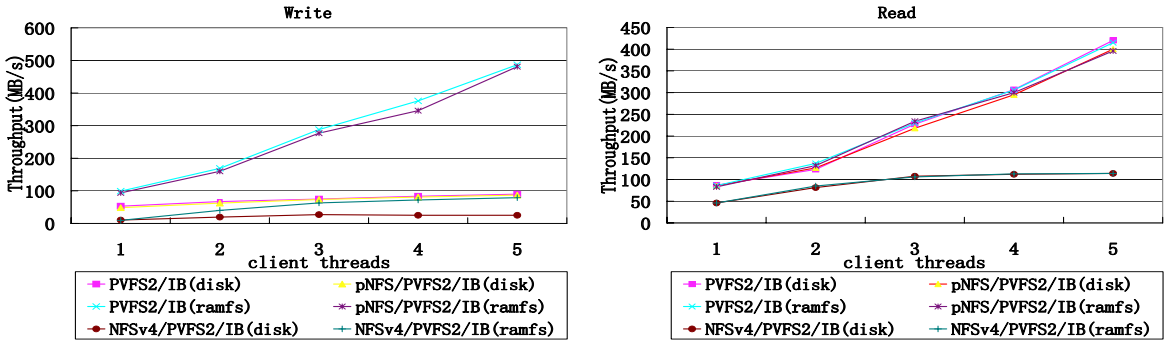
For the camera ready version, we hope to include experiments with a larger number of nodes. We also plan to look at the impact of RAID based disk on performance.

We will also evaluate different workloads for these configurations. The purpose is to see the performance and scalability of pNFS for real workloads.

5. Conclusions and Future Work

In this paper we did an extensive performance evaluation of pNFS/PVFS2 on an InfiniBand cluster. From our experimental results, we see that pNFS/PVFS2 can take advantage of InfiniBand well. pNFS/PVFS2 over InfiniBand can achieve a peak Write throughput of 500 MB/s and a peak Read throughput of 445 MB/s. We also see that with InfiniBand. It adds very little overhead and achieves almost the same throughput as native PVFS2. Compared with the traditional NFS, pNFS/PVFS2 provides significantly higher throughput and shows better scalability. From our experience, we believe that pNFS on InfiniBand clusters is promising.

As part of future work, we would like to explore the impact on performance and scalability of pNFS with a file layout driver, using NFS/RDMA.



(a) Write Performance

(b) Read Performance

Figure 3. pNFS PVFS2 and NFSv4 Comparison

References

[1] InfiniBand Trade Association. http://www.infinibandta.com.

[2] Iozone Filesystem Benchmark. http://www.iozone.org .

[3] Lustre a network clustering FS. http://wiki.lustre.org/index.php?title=Main_Page .

[4] Parallel Virtual File System. http://www.pvfs.org .

[5] CITI. Linux pNFS Kernel Development. http://www.citi.umich.edu/projects/asci/pnfs/linux .

[6] Dean Hildebrand and Peter Honeyman. Exporting storage systems in a scalable manner with pnfs. In *Proceedings of the 22nd IEEE - 13th NASA Goddard (MSST2005) Conference on Mass Storage Systems and Technologies*, April 2005.

[7] Dean Hildebrand and Peter Honeyman. Direct-pnfs: scalable, transparent, and versatile access to parallel file systems. In *HPDC*, pages 199–208, 2007.

[8] Dean Hildebrand, Lee Ward, and Peter Honeyman. Large files, small writes, and pnfs. In *ICS*, pages 116–124, 2006.

[9] IETF. NFS V4.1 Specification. http://tools.ietf.org/wg/nfsv4/draft-ietf-nfsv4-minorversion1 .

[10] OpenSolaris. OpenSolaris Project: NFS version 4.1 pNFS. http://opensolaris.org/os/project/nfsv41 .