February 2009

Evaluation of GPFS Connectivity Over High-Performance Networks

J. Srinivasan, S. Canon and M. Andrews, NERSC, LBNL
{jsrinivasan,scanon,mnandrews}@lbl.gov

Abstract

We present the results of an evaluation of new features of the latest release of IBM's GPFS filesystem (v3.2). We investigate different ways of connecting to a high-performance GPFS filesystem from a remote cluster using Infiniband (IB) and 10 Gigabit Ethernet. We also examine the performance of the GPFS filesystem with both serial and parallel I/O. Finally, we also present our recommendations for effective ways of utilizing high-bandwidth networks for high-performance I/O to parallel filesystems.

I. Introduction

Access to high-bandwidth and high-performance I/O has typically been done using parallel filesystems such as Lustre[1] or GPFS[2], and using a high-performance interconnect. The newest release of IBM's GPFS filesystem provides ways of using non-proprietary and commodity interconnects such as Infiniband much more efficiently than before. With the availability of multiple ways of connecting to GPFS filesystems at high-performance, we decided to evaluate these different connectivity methods and compare their performance with the goal of implementing the chosen configuration as the method of choice for I/O to a small cluster to be used for running tightly coupled parallel jobs with a non-trivial I/O bandwidth requirement.

We first present an outline of cluster configuration and the I/O requirements of the cluster. Then, we present the configuration of the filesystem we will be accessing, followed by possible ways of accessing the filesystem. The three methods we considered were direct fibre-channel (F/C) connectivity to the filesystem, connecting to the filesystem server nodes using 10Gbps Ethernet over a high-performance switch and finally, a "Gateway Model" which could use elements of both direct F/C and 10Gbps connectivity methods.

II. Cluster Configuration and I/O requirements

The "PLANCK" cluster, which is part of the PDSF [3] system at NERSC is a small cluster composed of 32 quad-core, dual-socket AMD compute nodes with 1 interactive node. Each node has 32GB of RAM and a dual-port DDR2 Infiniband card. The nodes are connected using a simple Infiniband network with 3 24-port switches. The cluster will mainly be used for analysis of Cosmic Microwave Background data from the PLANCK satellite[4]. The jobs run on the cluster will be parallel jobs using upto 256 cores each and will require a low-latency interconnect (provided by the IB network). They will also require high-bandwidth access to a large, parallel filesystem that has to be both accessible to all the nodes in the cluster itself, and to other systems at NERSC (such as the Cray, the IBM P5 system, etc). Additionally, the nodes in the cluster are connected with a GigE network to the rest of the PDSF cluster, which will provide basic home directories, and other storage with less demanding I/O requirements. Fig. 1 shows the schematic of the cluster configuration.

All nodes in the cluster run Scientific Linux (SL v5), which is a variant of RedHat Enterprise Linux v5 using the latest kernel that comes with that distribution (at the time of these tests was 2.6.18-92.1.13).

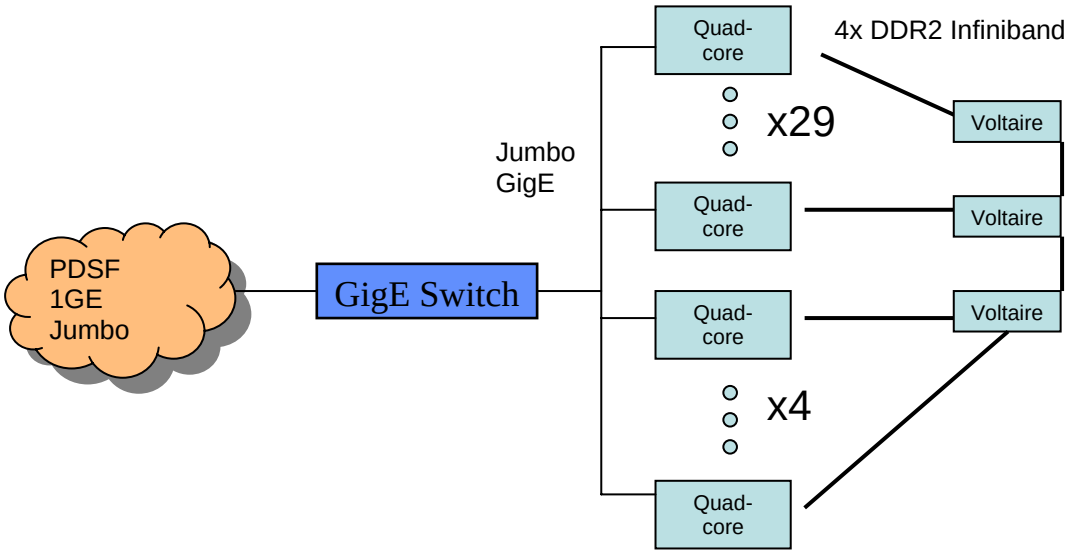

Fig 1. Schematic of the PLANCK cluster

Since the jobs that will be run on the cluster will perform highly coordinated access to the storage, a good parallel filesystem is required. Additionally, since the nodes will be, in general, performing both I/O and computation (although not necessarily simultaneously), we would like to ensure that I/O traffic and inter-process communication traffic are separate to minimize the impact that one has on the other. Finally, since the PLANCK cluster jobs are just one element of the larger workflow scheme for the simulation and analysis of the project, the same filesystem that is available on the PLANCK cluster should be available on the other systems at NERSC as well, where other steps in the workflow occur.

The last requirement demands that we access a filesystem that is "global" across the NERSC center and we describe the configuration of this filesystem below.

III. NERSC Global Filesystem

The NERSC Global Filesystem[4] (NGF) is a filesystem that is accessible across all computing platforms at NERSC. This allows users to easily share code and data files across multiple platforms without explicitly copying files. The filesystem itself is a GPFS filesystem and uses the MultiCluster capabilities of GPFS to enable access from multiple systems. Fig. 2 shows the schematic of the NGF. Systems at NERSC typically connect to NGF over one of two ways: either using Ethernet by connecting to the NERSC 10Gbps network (although individual hosts do not need to be at 10Gbps) or by directly attaching to the F/C fabric that NGF hosts.

The NGF currently consists of approximately 250TB of storage (a combination of DataDirectNetworks storage, IBM FastT and Sun storage). While the theoretical peak bandwidth out of NGF is currently approximately 8GB/sec, we can probably only

achieve about 6GB/sec, and it should be noted that this bandwidth will be shared by the multiple systems that will access NGF.

For some of the evaluation done for this paper, we also used another filesystem that was setup similarly to NGF, except that it had a lower bandwidth of approximately 2GB/sec.

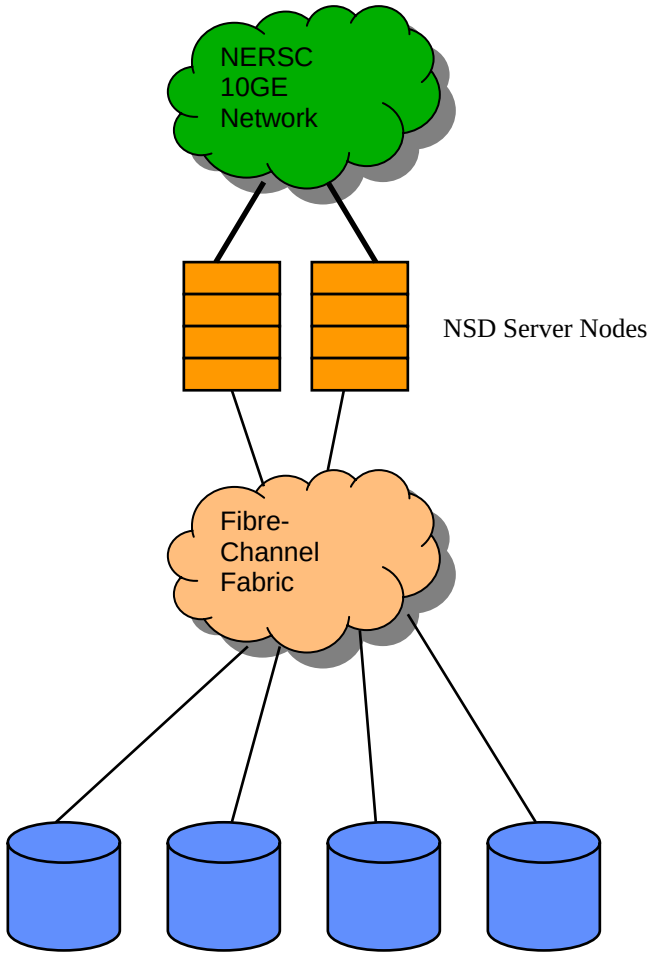

Fig. 2 Schematic of the Nersc Global Filesystem Configuration

There are two main ways to connect to NGF (or, more generally, any GPFS filesystem) and we list them below:

a. All GPFS filesystem instances (also called “clusters” in GPFS terminology) are accessible to other remote GPFS clusters over the Ethernet – although it is recommended that either 1Gbps or 10Gbps connectivity be used.
b. We can directly access the disks that comprise the filesystem – in this case, the storage is F/C attached storage, so we can directly connect to the F/C fabric that connects the storage. This is called SAN-mode connectivity in GPFS terminology.

Additionally, both methods of connectivity can co-exist between GPFS clusters, and GPFS can be configured to either use one method or the other or always use only one of the methods available. Typically, SAN-mode provides better performance access to a GPFS filesystem, although the performance itself will depend on the details of how the connections are made (how much of the bandwidth is shared, the F/C configuration,

access from other GPFS clusters, etc.). We first present an outline of how we utilized 10Gbps connectivity to NGF.

IV. 10Gbps Connectivity to NGF

Fig. 3 shows the 10Gbps configuration for the cluster. We utilized 10GE Ethernet (PCI-e(4x)) cards in the nodes, connected to a Woven Systems EFX-1000 switch, which was then connected to the NERSC 10Gbps network with 4 10Gbps connections. While the maximum possible bandwidth out of the cluster is 40Gbps, due to the network configuration on NGF and the NERSC 10Gbps network, we can only expect around 20Gbps all the way into NGF.

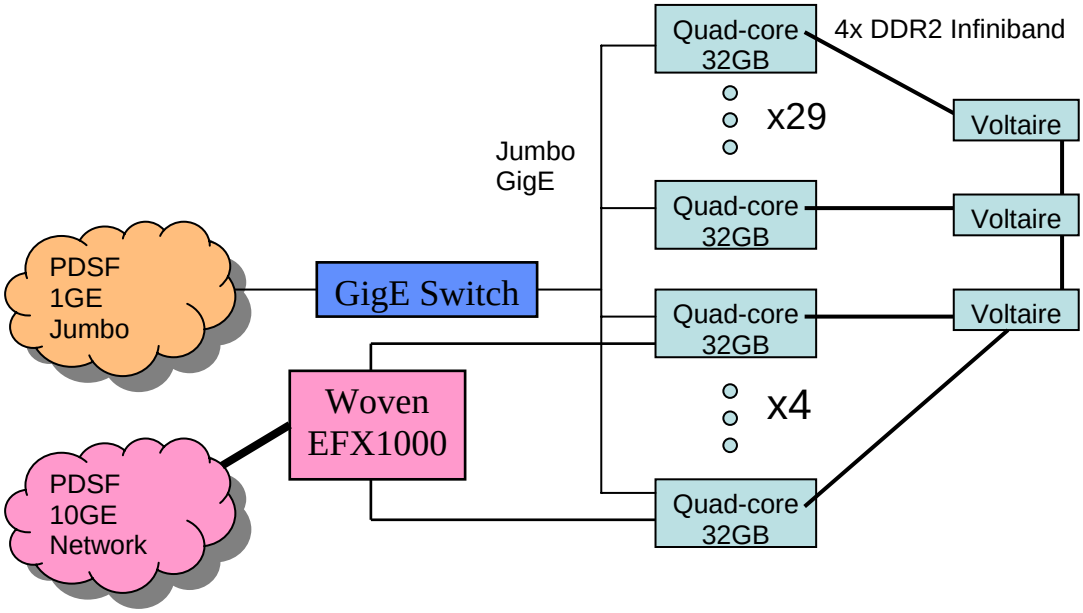

Fig 3. 10Gbps configuration of the PLANCK cluster

The Woven EFX-1000 is a low latency 144 port 10Gbps switch, so we could conceivably connect all the nodes in the cluster to the switch and then have them all access NGF over the 10Gbps network. For the purposes of this test however, we only connected 4 nodes to the switch, since this is sufficient for us to match the bandwidth of the path to NGF and will also allow us to test the "Gateway mode", which is described later.

On the four nodes directly connected to the Woven switch, we used either Chelsio and NetXen 10Gbps NICs. The remaining 29 nodes talked to the 10Gbps network using the Infiniband network (IP over IB, or IPoIB). Thus, the NGF filesystem was mounted on four nodes using direct 10Gbps connectivity, and on the remainder using IPoIB.

V. 10Gbps connectivity tests

Our first tests were to determine if we could infact get line-rate performance between nodes on the Woven switch. Since we had a combination of NetXen and Chelsio cards, we also tested different combinations of senders and receivers. We used IPerf (version 2.0.2) for the tests. Fig. 4 shows the results of Iperf runs both between nodes on

the same switch and between nodes on the Woven switch and an Iperf server 2 hops away on the NERSC network. Due to a quirk in the version of NetXen cards we were testing, the "Jumbo Frame" MTU of the NetXen cards was 8K instead of 9K. As we can see, we get to within 3-4% of line-rate with the Chelsio cards and to within 10-12% of line-rate with the NetXen cards. For transfers to 2 hops away, we get between 5.5 and 7 Gbps for the Iperf tests. All of these are excellent results and validate the 10Gbps network as a viable transport mechanism for GPFS traffic.

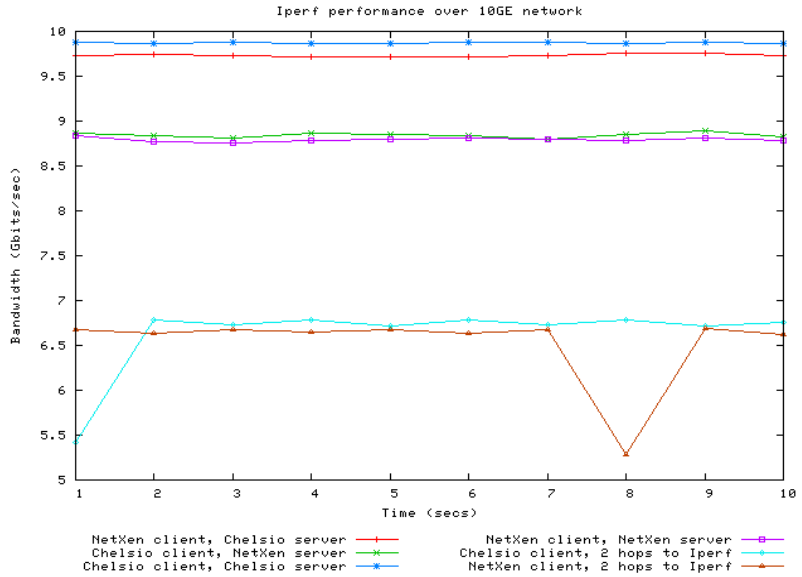

Fig. 4. Iperf tests on nodes on the Woven switch and to a node 2 hops away on the NERSC network using Chelsio and NetXen cards as servers/clients.

Additionally, we also tested the bandwidth over the IB network (using IPoIB). In this case, we ran the iperf client on a node which communicated with the nodes that had the 10GE cards over the IB network using IPoIB. Fig. 5 shows the results of these tests.

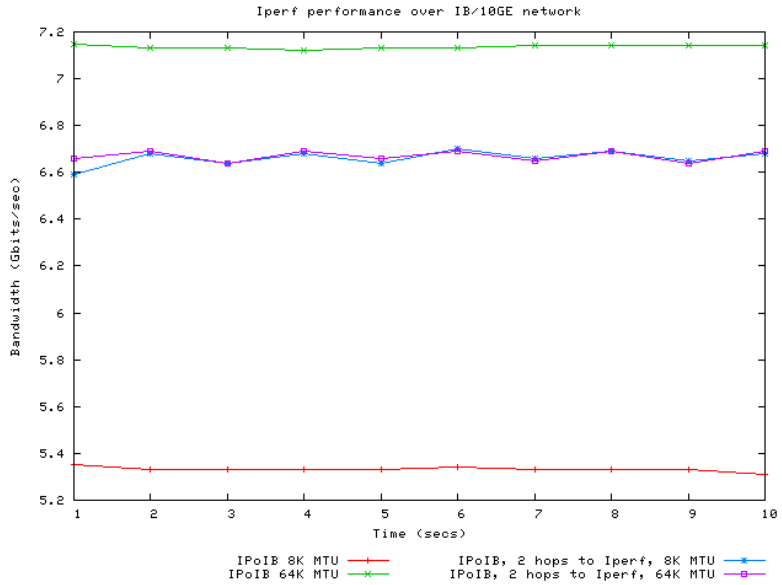

Fig. 5. Iperf tests between nodes on the IB network

As Fig. 5 shows, we can expect reasonable performance using IPoIB as well, which validates using the mixed IB/10Gbps configuration to carry GPFS traffic from NGF to the cluster.

We had to make a number of changes to the kernel parameters (the sysctl parameters) on the Linux kernel and Table 1 shows these parameters and their values. Additonally, since we were using 2 different networks (the 10Gbps network and the IPoIB network) we had to set specific routes to access the NGF system from the nodes on the IPoIB network.

As mentioned earlier, since the NetXen cards we used only supported a maximum MTU of 8K, we had to set different MTUs for the nodes that accessed NGF through the nodes with the NetXen cards and for nodes which access NGF through nodes with the Chelsio cards (which supported the full 9K MTU for Jumbo Frames).

Parameter Name	Value
Net.core.rmem_max	16777216
Net.core.wmem_max	16777216
Net.ipv4.tcp_rmem	4096 87830 8388608
Net.ipv4.tcp_wmem	4096 65536 8388608
Net.ipv4.tcp_mem	16777216 16777216 16777216

Table 1. Kernel (sysctl) parameters for tuning the 10Gbps network on SL5

VI. SAN-mode connectivity to NGF

SAN-mode connectivity to NGF is achieved by directly connecting the PLANCK nodes to the F/C fabric in NGF. This allows us to access the disks comprising the GPFS filesystem directly, while control traffic to the GPFS servers still goes over the Ethernet network. This connectivity mode is transparent, in that GPFS will discover the local path to the storage automatically, and use it if available. If the local path to storage is not available, GPFS will transparently fail-over to utilizing the network path to the storage.

For the SAN-mode testing, we used Qlogic QLA2400 series F/C cards. Additionally, due to the NGF configuration, we needed to use multipathing software on the Linux nodes in order to access the IBM FastT storage.

GPFS Configuration Parameters	Value
maxMBpS	2000
socketRcvBufferSize	131072
socketSndBufferSize	65536
nsdThreadsPerDisk	5
nsdMinWorkerThreads	16
nsdbufspace	70
pagepool	512m

Table 2. GPFS configuration parameters that need to be adjusted for optimal SAN-mode access performance

Fig. 6 shows the direct F/C configuration to the Planck cluster. For the purposes of this test, we connected first one node to the F/C fabric and later tested with two nodes (in order to test some failover capabilities). There were a number of changes we had to

make to the GPFS configuration in order to achieve good I/O performance to the filesystem. Table 2 lists the parameters and the values we had to set for the GPFS configuration.

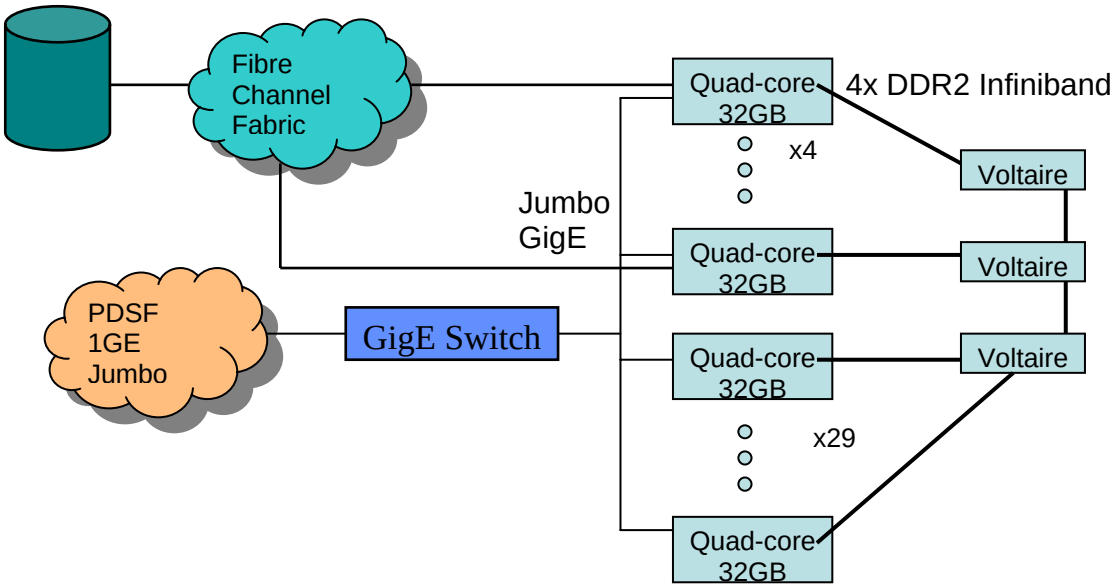

Fig. 6. Schematic of the configuration for the direct F/C connection from the PLANCK cluster to NGF (SAN-mode)

VII. SAN-mode connectivity tests

Our tests of SAN-mode connectivity were made to ensure we were capable of getting the maximum possible bandwidth out of the F/C cards in a node connected to the NGF system (or the test system).

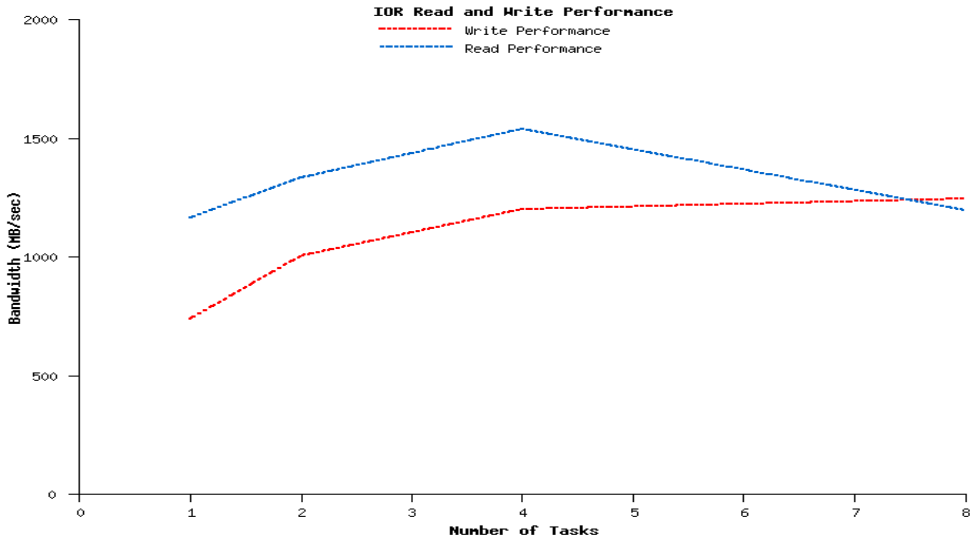

Figure 7. Read and Write I/O performance from a single node with direct F/C connectivity to a GPFS filesystem

Fig. 7 shows the results of I/O (both read and write) from a node with F/C cards connected to a GPFS filesystem in SAN-mode. The node had 4 single port F/C cards each capable of 4Gbps. The filesystem we connected to was capable of about 2GB/sec, and we are seeing I/O rates of 700 MB/sec to 1.5 GB/sec, which validates the use of direct F/C connectivity as a viable mechanism for getting good I/O performance to the filesystem.

VIII. Gateway Model of Connectivity to NGF

The "Gateway (G/W) Model" is essentially a subset of the 10Gbps or SAN-mode connectivity models, and in fact, in our tests, we already implicitly use the model for our connectivity, since we only connected a sub-set of nodes using either the 10Gbps or Direct F/C methods. The remaining nodes then communicate to the gateway nodes over an existing high-performance network – in our case, this is IB. Thus, a number of options become available to us:

a. We can use the same IB network for both inter-process communication and for I/O
b. We can use a second IB network by utilizing the second port on the dual-port IB cards.
c. The IB communication can be done using IPoIB or by using "native" IB methods (for instance, RDMA over IB).

The advantages of the gateway model are immediately apparent. The I/O connectivity is simpler, since we have fewer nodes to connect, and it is easily scalable as the I/O requirements grow. We should note, however, that the gateway model is an option only if there is a sufficient difference between the backend bandwidth of the filesystem being accessed and the bandwidth out of the cluster that is performing the I/O.

Fig. 8 shows the schematic of the gateway model we use (it shows both the 10Gbps and Direct F/C connectivity modes, although we would typically only use one or the other at one time).

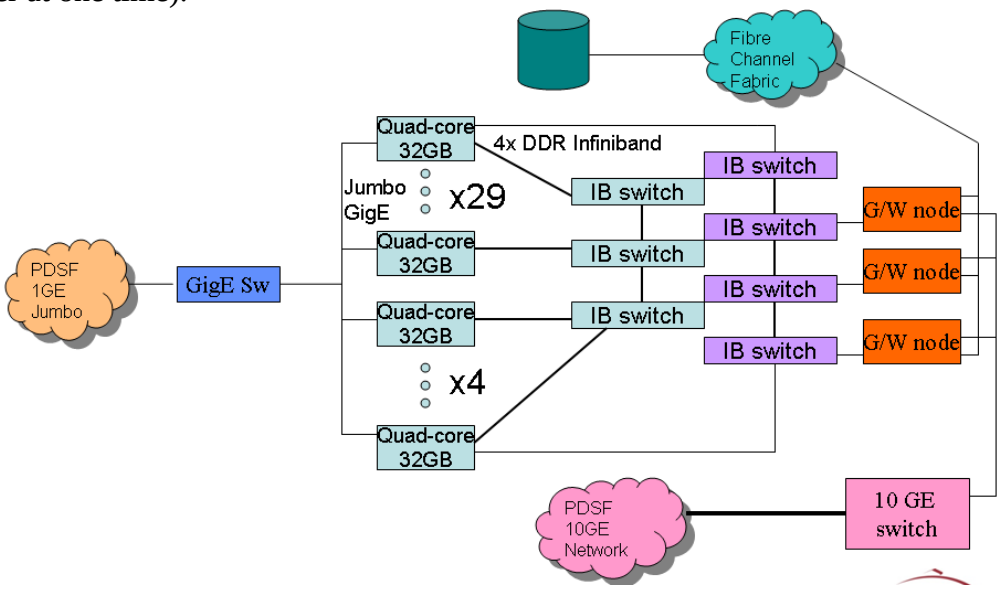

Figure 8. Schematic of the Gateway model of connectivity to a GPFS filesystem showing both 10Gbps and Direct F/C connectivity modes

VIII.A. 10Gbps connectivity using the Gateway Model

When using the 10Gbps connectivity mode in the G/W model, we put the 10Gbps NICs in a small number of nodes and use IPoIB for communication between the G/W nodes and the compute nodes. We can either use the same IB network that is used for inter-process communication or use the second port in dual-port IB cards to create a second IB network that will only carry I/O traffic.

While cabling can be an issue when using the second IB port, it may be necessary to separate I/O traffic from inter-process communication traffic. It will also be necessary to setup the second IB network to be completely distinct, i.e. use a separate subnet manager as well.

Our tests of the G/W model for the 10Gbps connectivity mode were done using IOR (v 2.10.1). The 10Gbps G/W nodes were setup similar to what was described earlier in Section IV. Fig. 9 shows the results of IOR read and write tests for 4-32 tasks over 4 G/W nodes. There is a substantial mismatch between the read and write performance, which is still under investigation. However, the tests show that the performance scales well as we increase the number of nodes that access the G/W nodes.

Additionally, we also did basic POSIX I/O write tests to the filesystem over the G/W nodes. These are shown in Figure 10, and the results show similarly impressive scaling performance to 128 tasks.

These results give us confidence that the G/W model using the 10Gbps connectivity mode performs well and is capable of scaling to the extent of the cluster configuration.

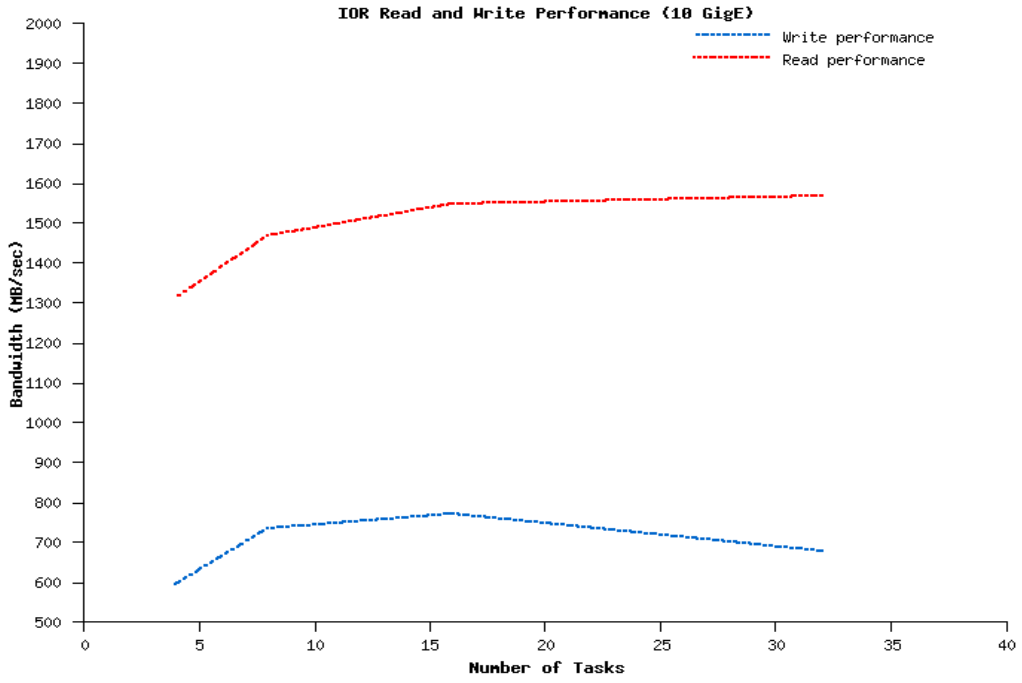

Figure 9. IOR read and write performance tests for the G/W model using 10Gbps connectivity mode

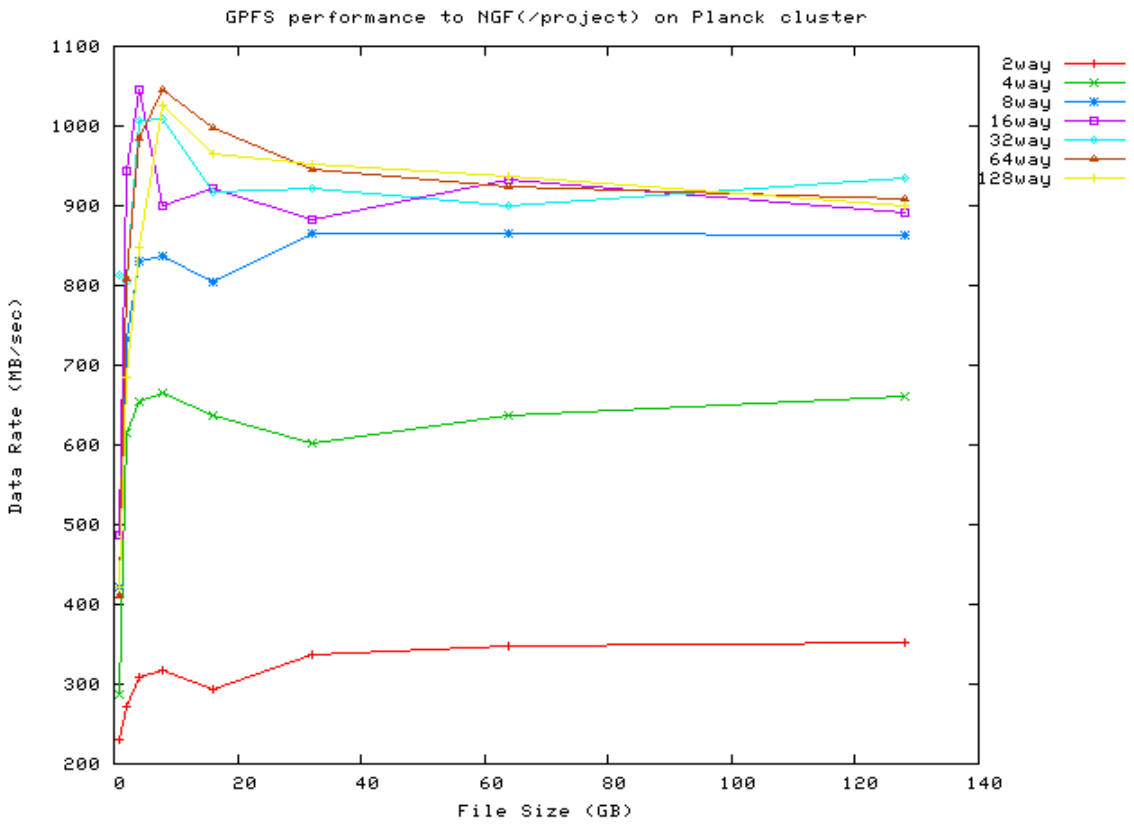

Figure 10. POSIX write performance for the G/W model using the 10Gbps connectivity mode.

VIII. B. F/C connectivity using the Gateway Model

For F/C connectivity using the G/W model, where the G/W nodes are accessing the GPFS filesystem in SAN-mode, we can communicate with the G/W node using either IPoIB or "native" methods such as RDMA over IB. The latest release of GPFS now fully supports using the latter method, which gives a substantial boost in performance compared to the former, and for our tests, we used GPFS in this mode. There are a few GPFS configuration parameters to set in order to enable this method of connectivity and they are detailed in Table 3. Of note is the "subnets" parameter, which must be set when multiple networks are available to GPFS on the same host, and especially when the default route is through an different network than the one through which access to the storage GPFS cluster is desired.

Parameter Name	Value
verbsPorts	mthca0/1 (for example)
verbsRDMA	enable
Subnets	Network to be used

Table 3. GPFS parameters to be set for using RDMA over IB

Additionally, GPFS now allows for the use of "private" NSD server nodes. These are NSD server nodes that are part of the storage-owning GPFS cluster, and are to be used in

multi-cluster GPFS deployments. Their purpose is to push the direct access point to the storage closer to the remote cluster. Thus, they may be physically located with the remote cluster, but directly attached to the storage being served. In our tests, we used our G/W nodes in this mode – i.e. as private NSDs.

There are several issues with the management of private NSDs. Since they are part of the storage cluster, they have direct access to the raw disks on the storage cluster and as such must be carefully managed. Additionally, both the storage cluster and the remote cluster must have fully privileged access to the private NSD node and security issues must be carefully worked out between the two clusters with regard to the private NSD.

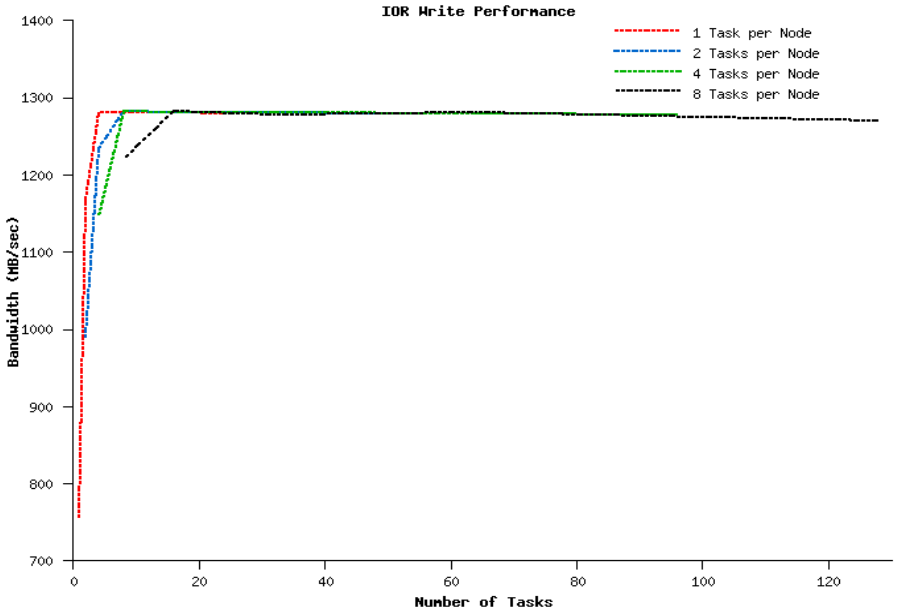

Figure 11. IOR write performance to a direct F/C attached G/W node (in private NSD mode) as a function of both tasks per node and total number of tasks

Figures 11 and 12 show IOR write and read performance respectively to the direct F/C connected G/W node (in the private NSD mode). All the IOR tests were done as before, using version 2.10.1, with MPI version MVAPICH2-1.0.2. The IB stack was OFED version 1.3.1. IOR was compiled with the Pathscale compiler (version 3.1). There are several points to note in these graphs.

First the performance is good, considering that the backend bandwidth of the storage GPFS cluster is approximately 2GB/sec. Secondly, the scaling (especially in the case of writes), through a single private NSD server/gateway node with 4 single port, 4Gbps F/C cards in it is impressive. And finally, performance is uniformly good for multiple tasks per node.

Read performance, shown in Fig. 12 is less consistent, and this is an area being investigated. The reason for the drop-off in read performance at around 32 tasks for the test using 4 tasks per node is unknown and is also being investigated. However, we still get more than 1GB/sec on reads as well, through a single direct F/C attached G/W node.

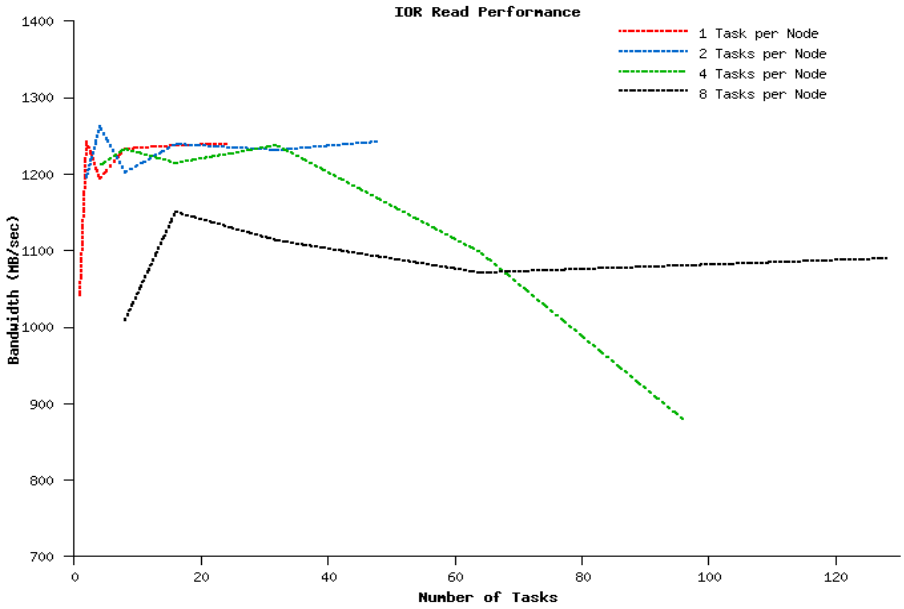

Figure 12. IOR read performance to a direct F/C attached G/W node (in private NSD mode) as a function of both tasks per node and total number of tasks

IX. Conclusions and Future Work

With the latest release of GPFS, there are a number of methods of comparable performance available to access remote GPFS filesystems. We have evaluated the setup and performance of several of these ways of access, ranging from direct access to GPFS filesystems (using F/C) to the gateway model (using 10Gbps and Infiniband).

There are several issues to consider when chosing an appropriate method. When there is a substantial mismatch between the bandwidth the remote cluster can support and the bandwidth the storage-owning GPFS cluster can support, the use of a “gateway model” to appropriately scale the bandwidth is recommended. This allows for a cost-effective way to access the GPFS filesystem and can be structured to make efficient use of existing resources on the remote cluster.

Thus, for small and medium size clusters, the gateway model provides a scalable, low-initial-investment way to provide high-peformance access, with incremental improvements in connectivity coming with small incremental costs (for example, addition of an extra gateway node).

We have also evaluated the use of a high-peformance 10Gbps network to access the GPFS filesystem and have shown this to be quite efficient. However, large-scale deployment on small and medium-scale clusters, must be carefully desgined to match the backend bandwidth available from the storage GPFS cluster.

We plan to continue testing the gateway model outlined above, especially with regard to the use of a second IB network to separate I/O and inter-process communication traffic. Additionally, we plan to test scaling and failover mechanisms for multiple private NSD/Gateway sever nodes. Performance testing using real codes that the users of the cluster will run is also underway.

X. Acknowledgements

We wish to thank the NERSC Global Filesystem team at NERSC, which is part of the Data Storage Group at NERSC for all their help. The PDSF team also provided invaluable assistance and support as did the NERSC Network Team. Large portions of this work would not have been possible without the assistance and cooperation of Woven Systems, who provided evaluation equipment and technical support for the 10Gbps testing.

This work was supported by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California.

XI. References

[1] http://www.sun.com/software/products/lustre/
[2] http://www-03.ibm.com/systems/clusters/software/gpfs/index.html
[3] http://www.nersc.gov/nusers/systems/PDSF/
[4] http://aether.lbl.gov/planck.html