

# Experiences with the Parallel Virtual File System (PVFS) in Linux Clusters

Kent Milfeld, Avijit Purkayastha, Chona Guiang  
Texas Advanced Computing Center  
The University of Texas  
Austin, Texas USA

## Abstract

The Parallel Virtual File System (PVFS) [1] is a shared file system for Linux clusters. PVFS distributes I/O services on multiple nodes within a cluster and allows applications parallel access to files. Also, the abstraction of I/O services as a “virtual” file system provides a high flexibility in the location of the I/O services within the cluster. For small, dedicated clusters, this flexibility provides an opportunity to tailor I/O services in unique ways that are not found in the “dedicated node” approach of larger systems. In this paper we measure the I/O performance for several PVFS configurations on systems with single-processor and dual-processor nodes, and show the importance of a high-speed interconnect and the feasibility of sharing I/O and compute services on dual-node systems.

## 1 Introduction

One of the key elements in the evolution of clusters to HPC status is the networking technology used to interconnect nodes within the cluster. It is the backbone for “sharing” data in a distributed memory environment. This backbone, when combined with local disks, also provides the framework for developing parallel I/O paradigms and implementing parallel I/O systems—an HPC environment for “sharing” disks in a parallel file system.

In PVFS, parallel I/O requests are initiated in a user application as clients (C), and the parallel I/O service occurs on I/O servers (IOS). The nodes where these processes occur are usually called compute nodes and I/O server nodes, respectively. Beside providing parallel access to a file, the PVFS API includes a partitioning mechanism to handle parallel, simple-strided access[2] with a single call. The file system is virtual in the sense that multiple “independent” I/O servers form a global file space, and application clients can run without kernel support (that is, in user space) in its native form (pvfs); although, a UNIX I/O interface is supported through a loadable kernel module. Also, PVFS is built *on top of* the local file system of each I/O server. Hence, there is a rich parameter space for obtaining optimal performance through local file systems. The authors of PVFS (Ceteci, Ross & Ligon)[3] have championed this file system, and have performed many different benchmarks to illustrate its capabilities[4].

In this paper we review PVFS configurations that are important to smaller clusters (10's of nodes), and at the same time point out some concerns of the larger clusters (100's of nodes). While some aspects of large and small systems are usually the same, the differences are worth noting, and will help give a complete view for configuring and using PVFS.

In the **PVFS Introduction** section the principal components of PVFS are reviewed and the configuration flexibility is explored. (The latter is fully understood when viewing the configuration possibilities from the perspective of a system administrator *and* an application developer.) The next section, **I/O Server Configurations**, presents the hardware and PVFS configurations used in our performance tests for: dedicated I/O servers, shared-node I/O servers and workloads. This section is followed by the **Test Results** section, and includes a discussion on the performance tests. The final section, **Conclusion**, summarizes the results and lists hardware components and application needs that should be evaluated when configuring and optimizing PVFS I/O performance.

## 2 PVFS Introduction

While much of the emphasis in cluster computing has focused on optimizing processor performance (e.g., compilers, hardware monitors, numerical libraries, optimization tools) and increasing message passing efficiency (e.g., MPICH-GM, MPI-Pro, LAM, M-VIA), much less effort has been directed towards harnessing the parallel capacity of multiple processors and their disks to obtain “HPC I/O”. This is partly due to the fact that the MPI-I/O aspects of the standard were not formulated until version 2 of MPI. It is also due to the cost and other uncertainties of dedicating processors or nodes to I/O services. For workloads that include only a few I/O intensive applications, dedicated nodes for parallel I/O may be an underused resource, to say the least. In addition, dedicated I/O nodes introduce heterogeneity into the system configuration, and therefore require additional resources, planning, management, and user education. But even so, the configuration flexibility and ease of a PVFS installation, without introducing hardware heterogeneity, is a boon to HPC cluster users and administrators that want to build parallel systems on their clusters.

The main attributes of PVFS are:

- Parallel I/O through a global file-name space
- I/O servers can be created from a pool of compute nodes
- Metadata manager (mgr): metadata is separated from data
- Independent I/O server, client and manager location
- Data transfer through TCP/IP
- Multiple user interfaces:  
(MPI-I/O, UNIX/POSIX I/O, native pvfs)  
(PVFS has been integrated into the ADIO layer of ROMIO)
- free and downloadable

The main components of PVFS are the metadata server (MS), the I/O servers (IOS), and the user application, or client (C). The metadata server performs the bookkeeping (file opening, closing, configuration information, etc.) through an mgr daemon that uses an unnoticeable fraction of resources (memory, cpu time, and network bandwidth). Each I/O server is created by installing an iod daemon on a node, and creating a directory on the local file system for data storage. Applications that have been built with the pvfs library, run on the compute nodes and become PVFS clients of the daemons when performing I/O on the PVFS file system (either through native PVFS, UNIX-I/O, or MPI-IO calls).

Clusters with a large number of processors usually dedicate nodes specifically to be only I/O servers. The metadata mgr daemon is often installed on a “frontend” (as is also done for small clusters). The rest of the nodes, where the applications

execute, are the clients (C) whenever I/O calls access the PVFS file system. (The use of PVFS does not preclude the access to any other file systems because PVFS is a virtual file system.) In this “dedicated node” configuration, the system administrator removes the I/O server nodes from the compute pool, and computations and I/O services occur on separate nodes. Diagram 1 illustrates this configuration.

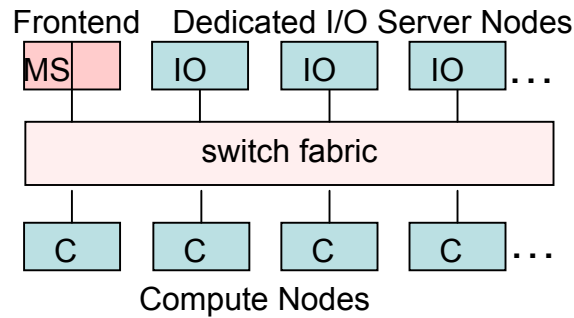


Diagram 1: Large cluster system: I/O servers (IO) run on dedicated nodes, application clients (C) run on compute nodes, and a meta-data server (MS) runs on the frontend.

On small departmental systems of less than 32 processors or “workclusters” (dedicated to a single application and/or user), the relative cost of dedicating nodes for I/O is high, and must often come at the expense of having fewer compute nodes. Determining the right number for the ratio of I/O servers to clients is a matter of knowing the IO/computation requirements of the major I/O application(s) on the system and what PVFS can deliver. We perform some basic performance measurement for various ratios below.

On many of the newer workcluster systems the nodes are frequently 2-way SMPs, because even a computation boost of only 15 to 20% from a second processor on a node can be economically justified. In these systems, where jobs are often run in dedicated mode (all processors devoted to a single user), it may be reasonable to use the “second” processor for I/O services as illustrated in Diagram 2. In this configuration the I/O servers are installed on nodes where application clients execute. In essence, the parallel I/O service is performed “on node” during the I/O phase of an execution. We describe this arrangement as a “shared node” configuration. In the Results section we perform measurements to compare the relative performance of shared and dedicated I/O nodes.

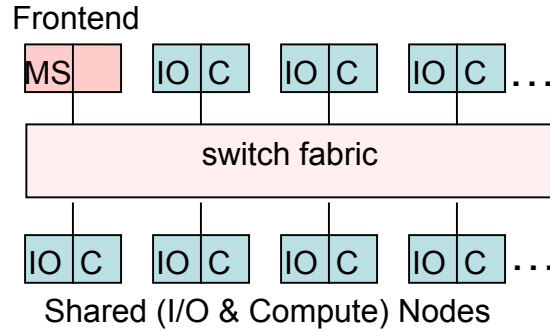


Diagram 2: Shared-node PVFS configuration on a dual-processor cluster system: both I/O servers (IO) and application clients (C ) share the same node, and a meta-data server (MS) runs on the frontend.

On a multi-user cluster, several users can simultaneously request I/O services from PVFS. Under these conditions, the state of the system is often far from the conditions under which benchmarks are performed. As our third PVFS test, we simulate a heavy multi-user I/O “workload”. Workload tests should not be overlooked when assessing the parallel I/O performance on clusters that run multiple jobs simultaneously.

### 3 I/O Server Configurations

Two different Pentium III clusters were used to evaluate the PVFS file system. System I consists of single-processor nodes and a 100Base-T network to connect the I/O servers and clients for PVFS I/O data transfers. (It has a Myrinet 2000 switch, but it is used exclusively for message passing.) System II consists of dual-processor nodes and a Myrinet 2000 switch for both message passing and the PVFS file system. System I and System II configurations are listed below:

#### System I

16 nodes	single-processor Pentium III 1 GHz ServerWorks motherboard
2.5GB	Memory per node
18GB	ATA33 IDE disks
16 ports	Myrinet-2000 Switch (for MPI ONLY)
16 ports	10/100 Base-T Cisco 2400 switch (for PVFS)

#### System II

32 nodes	2-way Pentium III 1 GHz (Coppermine) IBM x330 ServerWorks motherboard
2GB	Memory per node
18GB	10K-RPM Ultra 160 SCSI disks
32 ports	Myrinet-2000 Switch (for MPI & PVFS)

In the dedicated node tests we evaluate the performance of PVFS over a range of I/O server nodes on System I. The ratio of I/O servers to client nodes was varied, keeping fixed the total number of nodes, 16.

File sizes, between 10KB and 1GB, were written and read by the clients. Here, we use KB, MB, and GB to represent 1000, 1000000, and 1000000000 bytes,

respectively. I/O performance, or “bandwidths” were derived from 3 trials (as were all other tests) from the PVFS read/write times. We don’t call this measure “disk I/O” because we don’t include the time to flush the kernel I/O buffers to disk on the I/O servers. However, the IO bandwidth that we measure is a reasonable metric for PVFS performance, under the auspice that this kind of I/O can be found in many applications. Only when very large I/O files are created, beyond 2 GB per I/O server, will the I/O bandwidth approach the disk I/O speed.

Only certain I/O-server/client ratios were used, {1/15, 2/14, 4/12, 8/8, 12/4, 14/2, 15/1}, along with an appropriate file striping, in order to insure an even distribution of work across the clients and I/O servers, and to create the access patterns (maps) shown in Diagram 3 and 4. (These patterns are not required in the normal use of PVFS.)

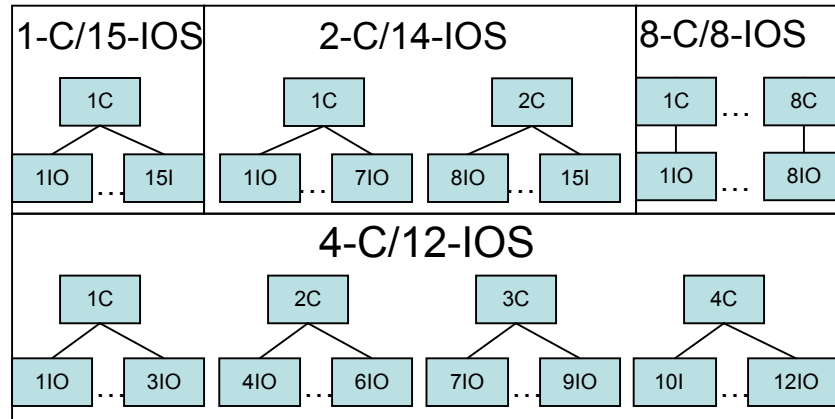


Diagram 3: I/O-server/client maps. I/O server file-stripes and client (compute node) read/write sizes were adjusted to map I/O requests to specific I/O servers (many to one).

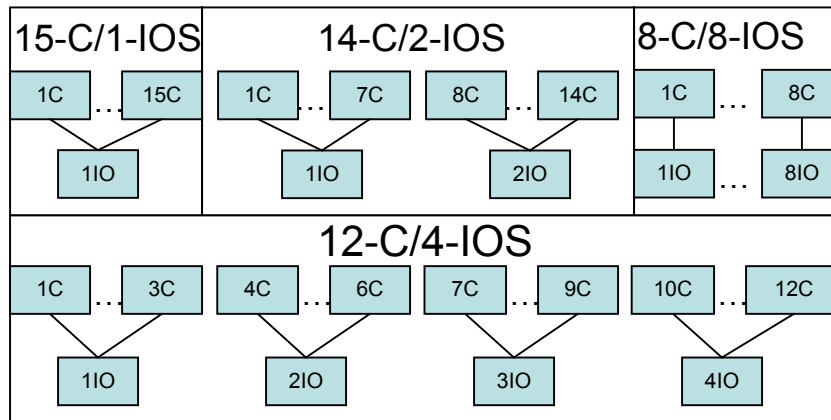


Diagram 4: Compute node / I/O server node maps. I/O server file stripes and client (compute node) read/write sizes were adjusted to map I/O request to specific I/O servers (one to many).

For the 1, 2, 4 and 8 I/O server cases the file stripes were 1/1, 1/2, 1/4 and 1/8 of the file size, respectively. Likewise the 12, 14 and 15 I/O server cases use file stripes that are 1/12, 1/14 and 1/15 of the file size, respectively. For the cases

where I/O servers outnumber the clients, the reads and writes are serial across the PVFS file stripe group. To maintain this symmetry when the clients outnumber the I/O servers, I/O operations within the groups were serialized by a simple MPI "token passing" mechanism; that is, only one member of a group was allowed to write to an I/O servers at a time. (We didn't want to measure contention in this experiment. That is measured in the workload test.)

In the shared-node tests, the same configurations (ratios of I/O servers to clients) of the dedicated tests were used on System II. On this system, PVFS is configured over the Myrinet switch.

Also, in the shared-node tests, the same experiments in the dedicated-node tests were first rerun on the System II, using only one processor on a dedicated node for either the I/O server or the client. Next, the tests were run using the shared mode illustrated in Diagram 2. I/O servers and compute nodes were paired by matching "IO" and "C" numbers of the corresponding maps in Diagram 3 and Diagram 4. That is, 1C and 1IO were paired on the first node; 2C and 2IO were paired on the second node, etc.

The workload tests were also performed on System II. In these tests, four dedicated nodes were used as I/O servers, and groups of four clients (one per node) executed I/O requests to a file, each group writing to a different file. The number of groups simultaneously reading or writing to a file ranged from one to four. The file stripes were adjusted so that each client performed I/O to a single I/O node, and 10, 100 and 1000 MB file sizes were used.

## 4 Test Results

### 4.1 Dedicated I/O Servers

Figures 1 and 2 show the PVFS I/O performance on System I for 1, 2, 4, 8, 12, 14, and 15 I/O server nodes, keeping the total node count (I/O + compute) fixed at 16. Data for file sizes from 1KB to 1GB are drawn as curves over the different configurations to identify the effect of the file size.

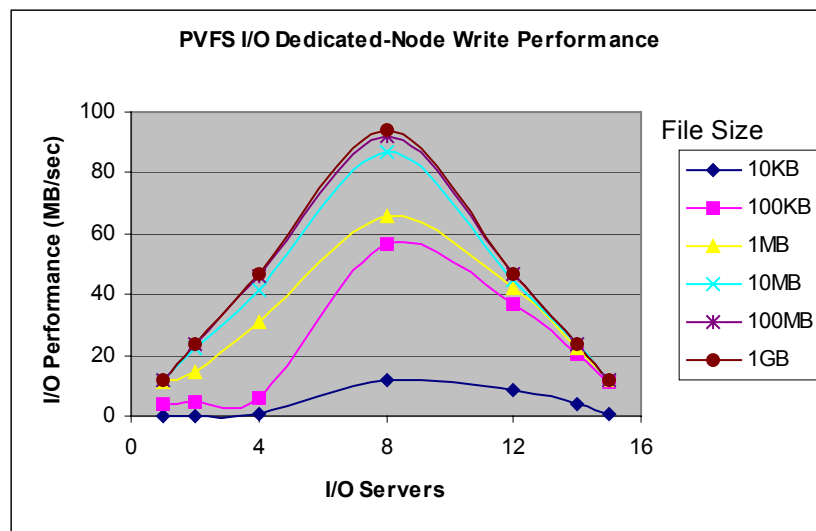


Figure 1: Write performance for dedicated I/O server nodes on System I.

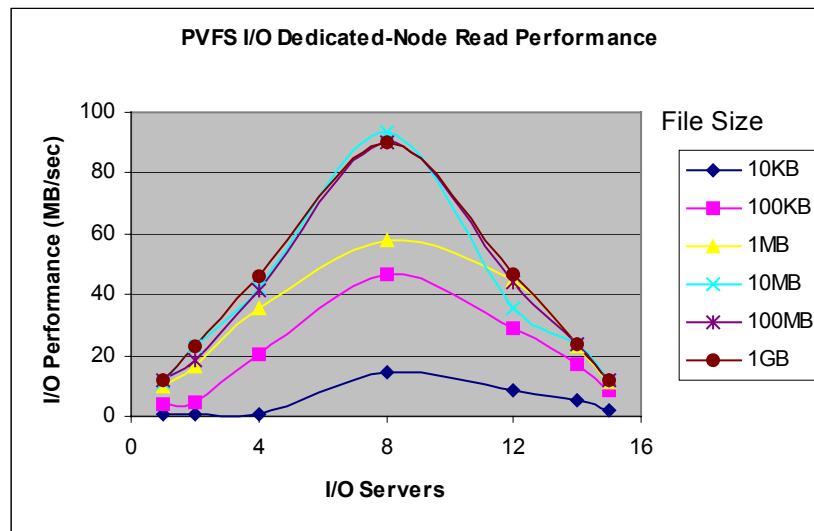


Figure 2: Read performance for dedicated I/O server nodes on System I.

Both the read and write performances show a similar behavior, with the write performance up to 10% higher than the read performance. The file size has a significant impact on the performance, (markedly so for the 8-IOS configuration). Throughout the range of file sizes, 10KB to 1MB, the performance increases for all configurations, and in general, remains the same for the 10MB, 100MB and 1GB file sizes.

Since the highest I/O performance is obtained for 8 I/O servers and 8 clients, splitting the number of nodes evenly between I/O and computation provides the highest reading and writing I/O throughput.

Figures 3 and 4 show linear and logarithmic plots for I/O write performance to a local file system on Systems I and II. Because of the large memory (2.5GB in System I and 2.0GB in System II), the kernel buffering affects the I/O performance over a wide range of file sizes and peaks at 85 MB/sec. A “near” asymptotic value is only reached at 8GB, providing a disk I/O performance of about 6 MB/sec. Because the performance of the PVFS file system is never larger than 90 MB/sec, it is apparent that the 100Base-T network is the limiting factor for the I/O performance. For the 8-IOS/8-client case, 8 paths are used through the switch, providing 100MB/sec of bisectional bandwidth, while the peak large-file performance is only 10% under this value.

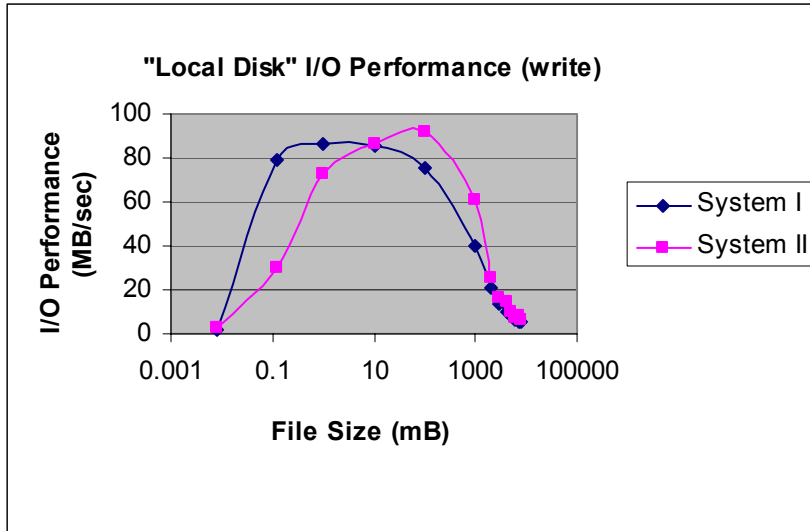


Figure 3: I/O performance to local disk on System I and System II (logarithmic scale)

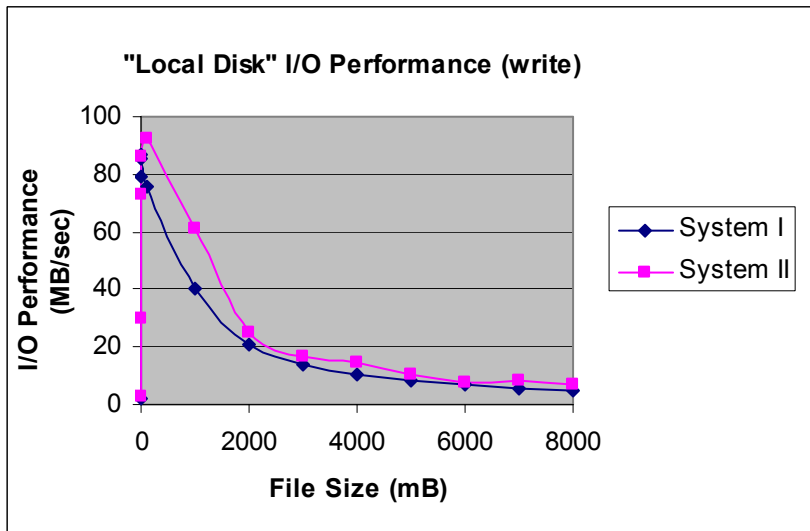


Figure 4: I/O performance to local disk on System I and System II (linear scale)

## 4.2 Shared I/O Servers

Figures 5 and 6 show the I/O performance on System II for 1, 2, 4, 8, 12, 14, and 15 I/O servers, keeping the total count (I/O servers + clients) fixed at 16. Only a single processor per node was used; and each node was dedicated as an I/O server or a client, as in the tests for System I (Figures 1 and 2). Figure 6 and 7 show the results for node sharing. Data for file sizes from 1KB to 1GB are drawn as curves over the different configurations to identify the effect of the file size.

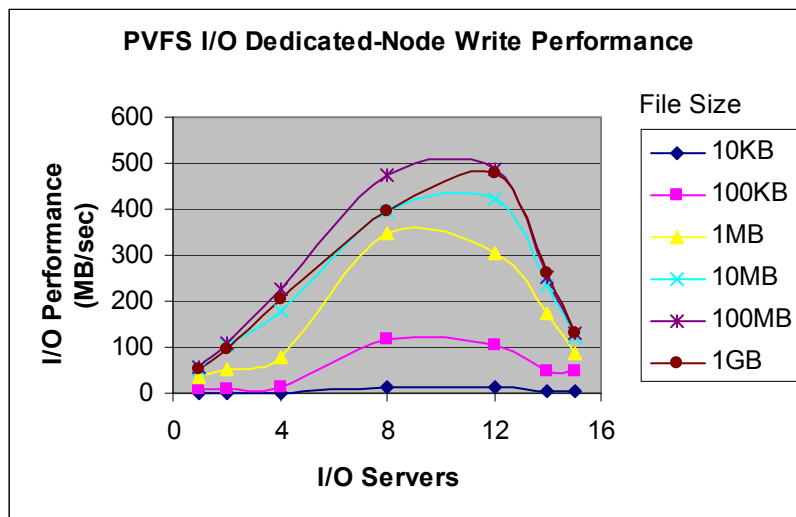


Figure 5: Write performance for dedicated I/O server nodes on System II.

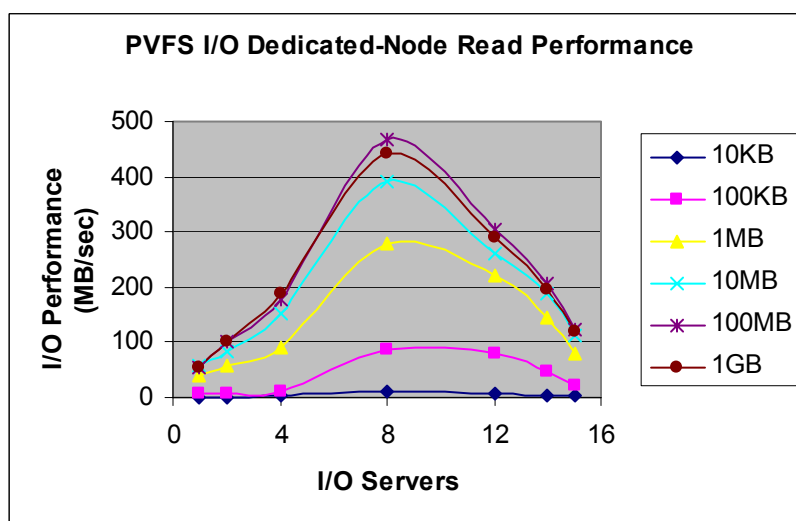


Figure 6: Read performance for dedicated I/O server nodes on System II.

For the dedicated-node configurations (Figures 5 and 6), both the read and write performances show a similar behavior, with peak write performance about 10-25% higher than the read performance, except for the 1GB case. Throughout the range of file sizes the performance increases up to 100MB for all configurations, and in general, remains the same for the 1GB case, except for the 8-IO/8-client. Unlike the tests for System I, the I/O write performance is skewed more to performing better for the 12-IO/4-client case, and less so for the reads.

The highest I/O write performance is obtained for the 8-IO and 12-IO cases (using 8 clients and 4 clients, respectively) and in general the 1GB file curves are the same as those for the 100MB, except for the extraneous 8-IO case. For small files, the 8-IO case gives better performance than the 12-IO configurations; but, for large files the trend is reversed. Also, the 1GB 8-IO I/O write case seems to be abnormally low. The read I/O performance is highest for the 8-IO cases, just as found for System I. So, in general, splitting the number of nodes evenly between IO and computation provides a high overall read and write IO throughput.

The most noticeable difference between the “dedicated” I/O servers on System I and System II is the relative scale of the performance. Since PVFS uses a Myrinet switch on System II, each I/O server has a path to each client that is 20 times faster (2000Mb/sec compared to 100Mb/sec). This is most noticeable in the peak PVFS performances for the two systems. System II I/O peak rates are over 5 times larger than those for System I. Since the System II local disk (buffered) performance for 100MB and 1GB files is only 40 to 90 MB/sec (Figures 3 and 4) it is reasonable to assume that the System II peak performance values for large-file I/O are limited by the local disk I/O performance at the node, and not the Myrinet switch. (We obtained measured MPI-GM point-to-point bandwidths over 200MB/sec, but have not measured Myrinet bandwidths over TCP/IP at this time.)

Figures 7 and 8 show the I/O performance on System II for 1, 2, 4, 8, 12, 14, and 15 I/O servers, keeping the total count (I/O servers + clients) fixed at 16. In these tests the I/O servers were arranged to occupy one processors on a sequence of nodes; and clients were paired to occupy the other processors, when possible. For the cases of 12, 14 and 15 I/O servers, there were nodes exclusively used by I/O servers.

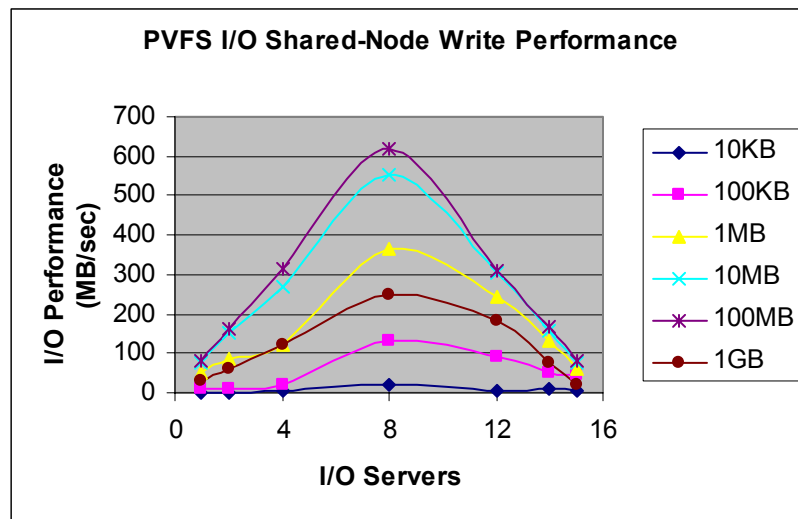


Figure 7: Write performance for shared-node I/O servers on System II.

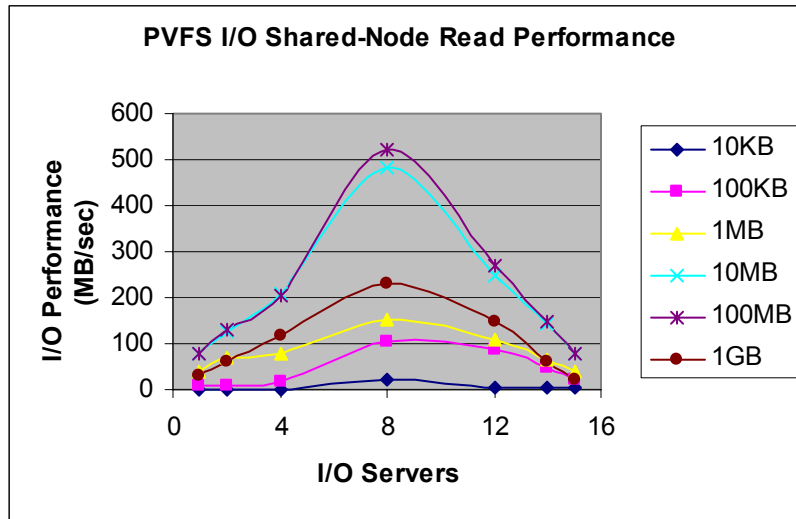


Figure 8: Read performance for shared-node I/O servers on System II.

For the shared node tests the profiles are more symmetric, and the peak performances are higher than the dedicated node results, and the peak write performances are over 100MB/sec faster. (See Figures 5 and 6.) Surprisingly, the performances shown by the read and write 1GB curves are significantly lower than the corresponding dedicated node curves. We suspect that this is due to the reduced kernel buffering introduced by memory occupation and memory contention by both the client and the server I/O.

### 4.3 Workload

Figures 9 and 10 show how simultaneous use of the PVFS file system affects the performance that a single application sees. Four I/O servers were assigned to 4 nodes, and groups of 4 clients (one to a node) were created on the other nodes. Each group accesses a different file on the PVFS files system. The "Simultaneous File Accesses" in the figures represents the number of groups simultaneously accessing their file.

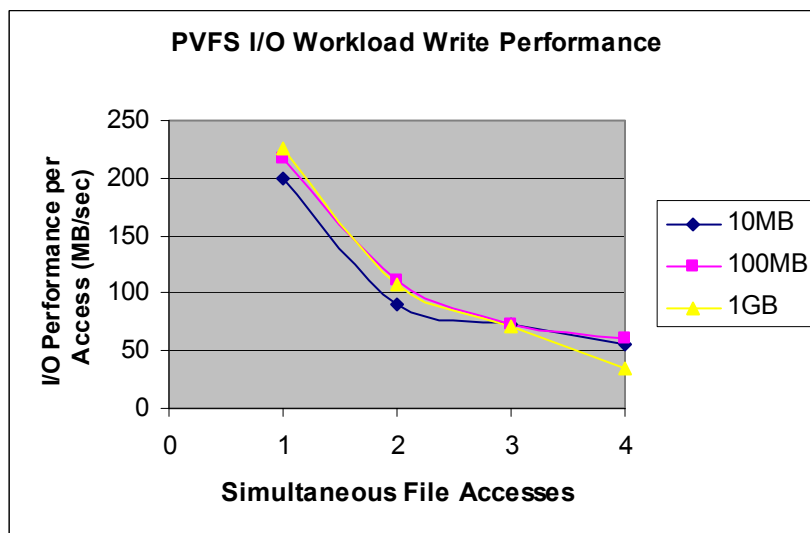


Figure 9: Write workload simulation on System II using 4 I/O servers.

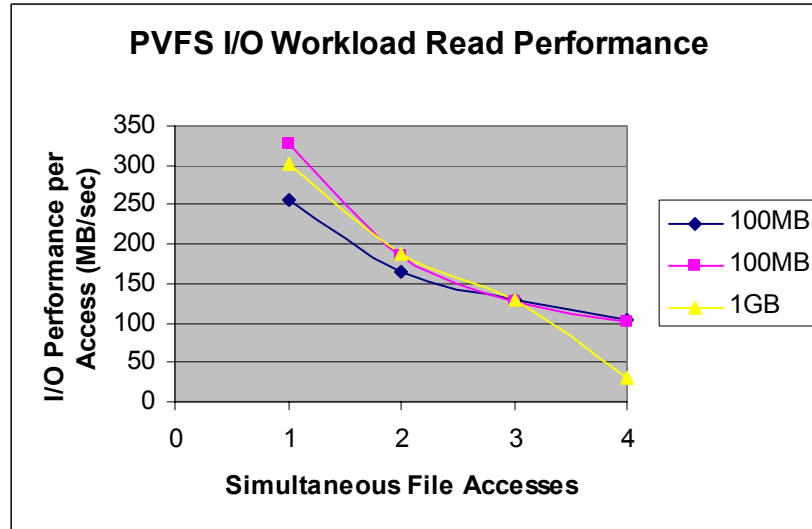


Figure 9: Write workload simulation on System II using 4 I/O servers.

For large-file accesses, 100MB and 1GB, the performance seen by an application diminishes directly with the number of other contenders for file service, as expected. For 100MB files the performances are 1, 0.5, 0.33, and 0.28 for writing and 1.0, 0.57, 0.38 and 0.30 for reading, compared to the expected ratios of 1, 0.5, 0.33, and 0.25. For 1GB files the pattern is very similar, except when 4 simultaneous accesses are occurring. In this case the performance drops to 0.15 for writing and 0.10 for reading, both well below the expected 0.25 ratio for sharing access 4 ways.

## 5 Conclusion

As one would expect, the speed of the local disks used by PVFS is a limiting factor when accessing files well above 1GB. Because PVFS is a parallel, global file system that uses the local file system on "service processors", kernel I/O buffering in large-memory systems and the speed of the interconnect to the I/O service processors affects the I/O performance in a substantial way. In the former case the effective I/O can be ten to a hundred times faster than the aggregate disk speed for file sizes around 1GB. Also, 100Base-T networks limit the buffered I/O performance, while high-speed networks, such as the Myrinet 2000, are not limiting and provide a large increase over 100Mb/sec networks.

By varying the PVFS configuration on two different systems, one using a high-speed network for PVFS and another using a low-speed network, we were able to determine the following:

- PVFS I/O servers with large memories can provide large, effective I/O transfer rates for moderate file sizes (up to a Gigabyte) through kernel buffering.
- In the PVFS file system the location of the I/O services in the cluster is highly flexible, and among other capabilities, provides the ability for 2-way SMP cluster nodes to share I/O services and computation. On a

dual-processor system, configured with I/O servers and clients (applications) on each node, I/O performance for file sizes up to 1GB were higher over a range of I/O servers. For 1GB files the performance was significantly less, and likely due to reduced memory for buffering and memory bandwidth contention.

- For a range of file sizes up to 1GB, a peak I/O performance was observed when an equal number of nodes were dedicated as servers and clients on 16 nodes in two different systems, networked with high-speed and low-speed switches. (However, in configurations where multiple clients were writing/reading to a single I/O server, the transfers were serialized. Our workload tests suggest that a greater aggregate performance will be realized in these configurations when writing/reading simultaneously over low-speed networks.)
- Clusters with 100Base-T networks can significantly bottleneck the transfer of moderate sized files to PVFS I/O servers.
- Clusters with high-speed networks provide a much higher capacity, bisectional bandwidth, for sustaining moderate sized file transfers for optimized PVFS configurations. With a Myrinet 2000 network, the network bandwidth was not a limiting factor, and peak I/O performances were increased five fold over those observed on a 100Base-T network.

Further studies of PVFS should include file sizes larger than 1GB and an in-depth analysis of the memory usage for both dedicated and shared I/O services on nodes. A larger study to investigate the economy of sharing I/O servers and real client applications on two-way, three-way or four-way processor nodes should be undertaken to determine optimal configurations for I/O and compute services in dedicated cluster systems. Also, a workload simulation over a wide range of configured I/O servers (e.g. from 4 to 32 nodes) would be helpful to system administrators in selecting appropriate PVFS configurations for peak I/O usage.

## References

- [1] <http://parlweb.parl.clemson.edu/pvfs/>
- [2] Nieuwejaar, N., Kotz, D., Purakayastha A., Ellis, C. S., & Best, M., File Access Characteristics of Parallel Scientific Workloads, *IEEE Transactions on Parallel and Distributed Systems*, 7(10):1075-1089, October 1996
- [3] Cettei, M.M., Ligon III, W.B., & Ross, R.B., Clemson University
- [4] <http://parlweb.parl.clemson.edu/pvfs/papers.html>, <http://www-unix.mcs.anl.gov/~rross/resume.htm>