

# Performance of a Parallel File System

## With Infiniband Interconnects

Yeo Eng Hee

(SVU/Academic Computing, Computer Centre)

With the newest addition of a Linux cluster with the latest high-bandwidth, low-latency Infiniband interconnect (see <http://www.topspin.com/index.html>), we were able to explore a scalable storage solution that will enable the nodes in the cluster to have a large, scalable, shared filesystem to complement the existing NFS-based filesystem in SVU. The objectives were:

1. To achieve faster file access (as compared to NFS)
2. To support applications that need to read or write large files
3. To provide a scalable, easy-to-manage file system that can grow with the cluster size

A few solutions are currently available in the market that can meet the above objectives, including, IBM's GPFS (General Parallel File System), RedHat's Cluster File System (CFS), and the open sourced PVFS (Parallel Virtual File System). We decided on IBM's GPFS solution because of its stability and ease of management. In addition, it is included in the Scholar's Pack, and NUS departments currently have access to the suite of software under this scheme for free.

In this article, we present the results of the I/O tests done on the GPFS filesystem, and compare the results with similar I/O operations on both the local hard disk and NFS. Table 1 below shows the comparison of the total amount of time it took to transfer files (via the unix 'dd' command) of various sizes from the local hard disk to GPFS, local and NFS filesystems.

Filesize (GB)	GPFS (s)	Local (s)	NFS (s)
4	161.46	261.48	670.42
8	266.11	371.11	1138.82
10	457.83	928.83	1659.14
20	551.32	1830.89	2592.80

Table 1: File I/O write tests with different file sizes

Clearly, the benchmark tests above shows that the GPFS file writing operations took a significantly shorter time to complete than even the local hard disk operations. Figure 1 is the graph of the results in Table 1.

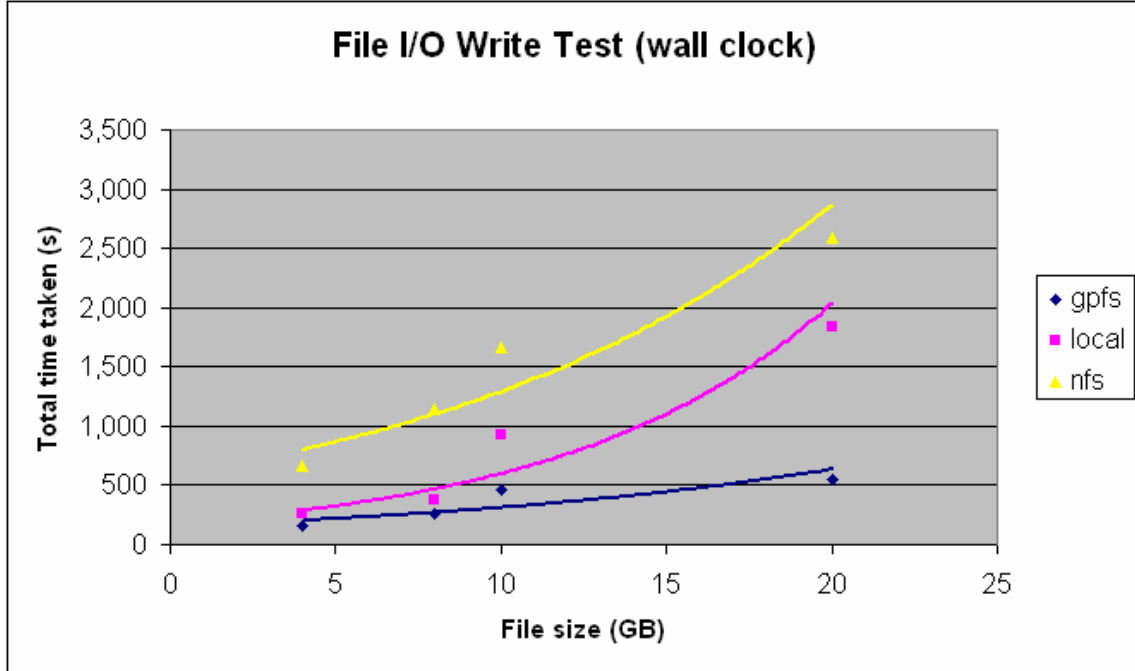


Figure 1: Comparison of wall clock time for file write I/O between GPFS, local and NFS file systems

Table 2 shows the breakdown of the time spent in each I/O writing operation, taking note that for such operations, the system is actually waiting for the I/O operation to complete.

filesize (GB)	Data	GPFS	Local	NFS
4	Average of real (s)	161.46	261.48	670.42
	Average of user (s)	5.48	4.92	4.13
	Average of sys (s)	80.83	60.66	43.19
	Average of wait (s)	75.15	195.90	623.10
8	Average of real (s)	266.11	371.11	1138.82
	Average of user (s)	9.90	7.68	5.96
	Average of sys (s)	152.03	99.31	72.63
	Average of wait (s)	104.18	264.12	1060.23
10	Average of real (s)	457.83	928.83	1659.14
	Average of user (s)	14.09	12.58	10.44
	Average of sys (s)	209.13	303.19	117.53
	Average of wait (s)	234.61	613.05	1531.17
20	Average of real (s)	551.32	1830.89	2592.80
	Average of user (s)	24.12	19.09	14.84
	Average of sys (s)	362.54	571.78	176.82
	Average of wait (s)	164.66	1240.02	2401.14

Table 2: Breakdown of total time spent

The following observations and conclusions can be derived from the results:

- GPFS performs 10 times faster than NFS! - the GPFS filesystem running on the faster Infiniband interconnect has a bandwidth of 10Gbps (giga-bits per second) to each system, compared to the 1Gbps Ethernet connection used by the NFS filesystem. In addition, the 1Gbps Ethernet connection has to be shared among all the cluster nodes, whereas the Infiniband interconnect provides dedicated 10Gbps connection to each system.
- GPFS also runs faster than local harddisk IO. One possible reason is that when writing to remote disk via GPFS, the system does not have the local harddisk overheads. Furthermore, the GPFS disks are made up of multiple hard disks, hence the data stream can be split to write to separate disks simultaneously, speeding up writing time.

For more information or comments on the above article, please write to Yeo Eng Hee ([cceyeoeh@nus.edu.sg](mailto:cceyeoeh@nus.edu.sg))