GPFS/Local Disk Access Performance Model

Ioan Raicu

25 September 2007

The following experiments attempt to summarize the GPFS performance on the ANL/UC TG cluster. These graphs represent 160 different experiments, covering 19.8M files transferring 3.68TB of data and consuming 162.8 CPU hours (these numbers do not include repeated or failed experiments, as the numbers would probably double if those were included); the majority of this time was spent measuring the GPFS performance, but a small percentage was also spent measuring the local disk performance of a single node. The dataset I used was composed of 201K files making up 1.25TB of data. In the hopes to eliminate as much variability (or bias) as possible from the results, I wrote a simple program that took in some parameters, such as the input list of files, output directory, length of time to run experiment (while never repeating any files for the corresponding experiment); the program then randomized the input files and ran the workload of reading or reading+writing the corresponding files in 32KB chunks (larger buffers than 32KB didn't offer any improvement in read/write performance). Experiments were ordered in such a manner that the same files would only be repeated after many other accesses, making the probability of those files being in cache small. With the exception of a few datapoints (which I could redo), I feel confident that these results are representative of the ANL/UC cluster performance GPFS.

Each figure has a label in parentheses, which is the name of the figure from the attached XLS file. I try to say the essential results in each graph, and to describe what I was measuring. Most graphs (unless otherwise noted) represent the GPFS read or read+write performance for 1 to 64 (1, 2, 4, 8, 16, 32, 64) concurrent nodes accessing files ranging from 1 byte to 1GB in size (1B, 1KB, 10KB, 100KB, 1MB, 10MB, 100MB, 1GB).

Figure 1 looks mostly OK with the exception of the 1KB 1 Node – Local value, which has a high standard deviation. I had repeated the experiment 3 times, and it happened very time. I'll investigate further later if I can find the cause of it.

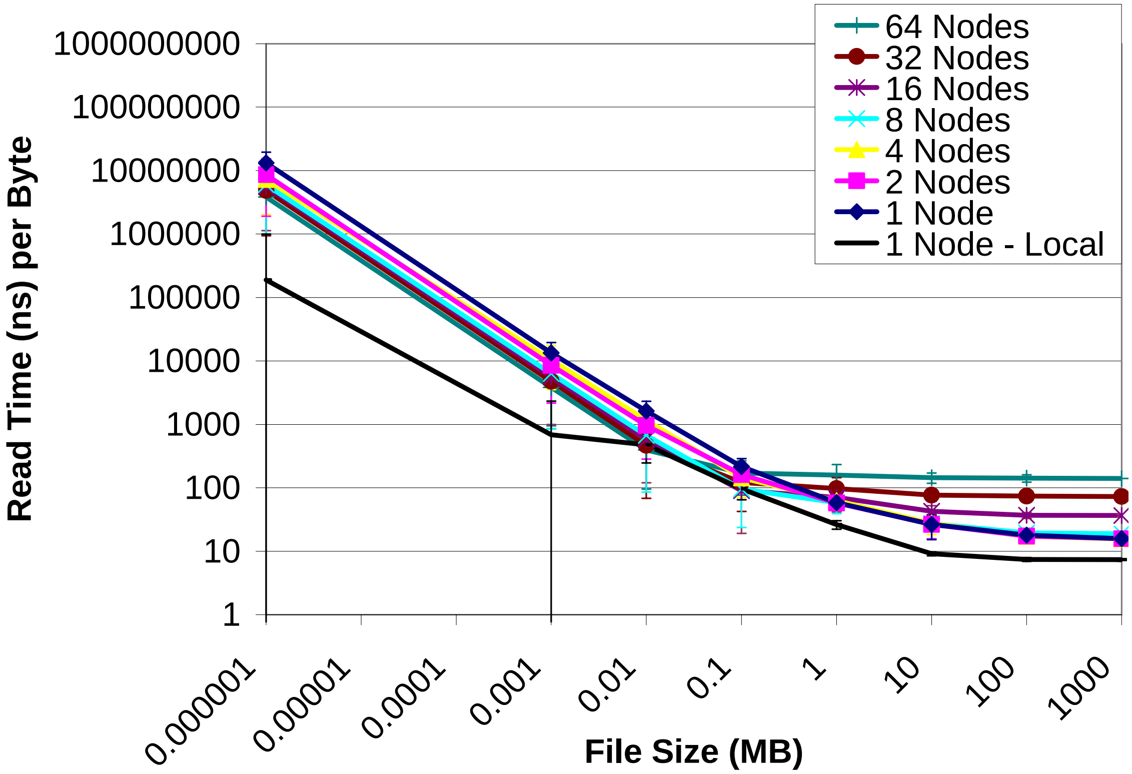

Figure 1 (gpfs model 1-64 nodes sum r tim): Read performance expressed in time (in nanoseconds) per byte; both axes are logarithmic; 1-64 nodes for GPFS, and 1 node for local disk access; 1B – 1GB files

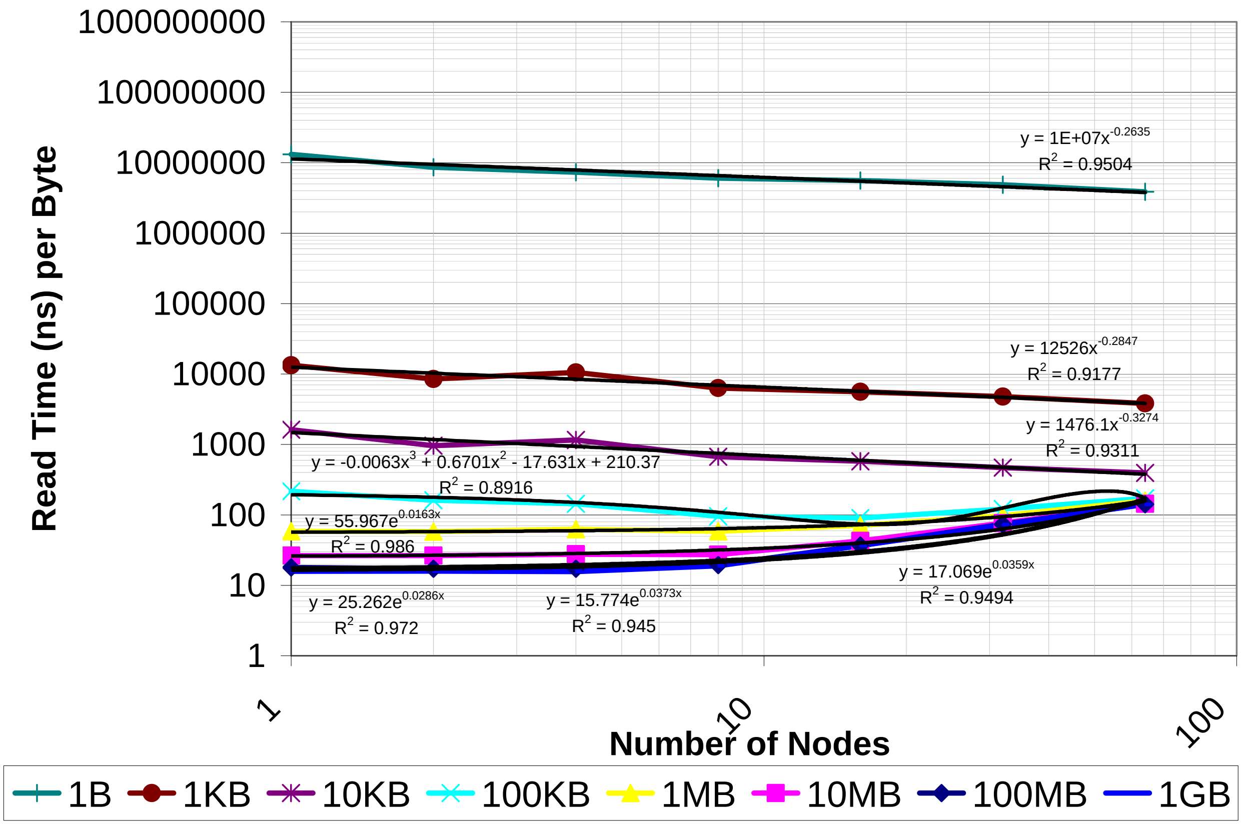

Figure 2 (gpfs model 1-64 nodes sum r t2): : Same as figure 1, but plotting the number of nodes on the x-axis instead of the file size; also shows the function approximating the corresponding data, including the R^2 value which can evaluate how well the approximation fits the data (value of 1 means perfect fit)

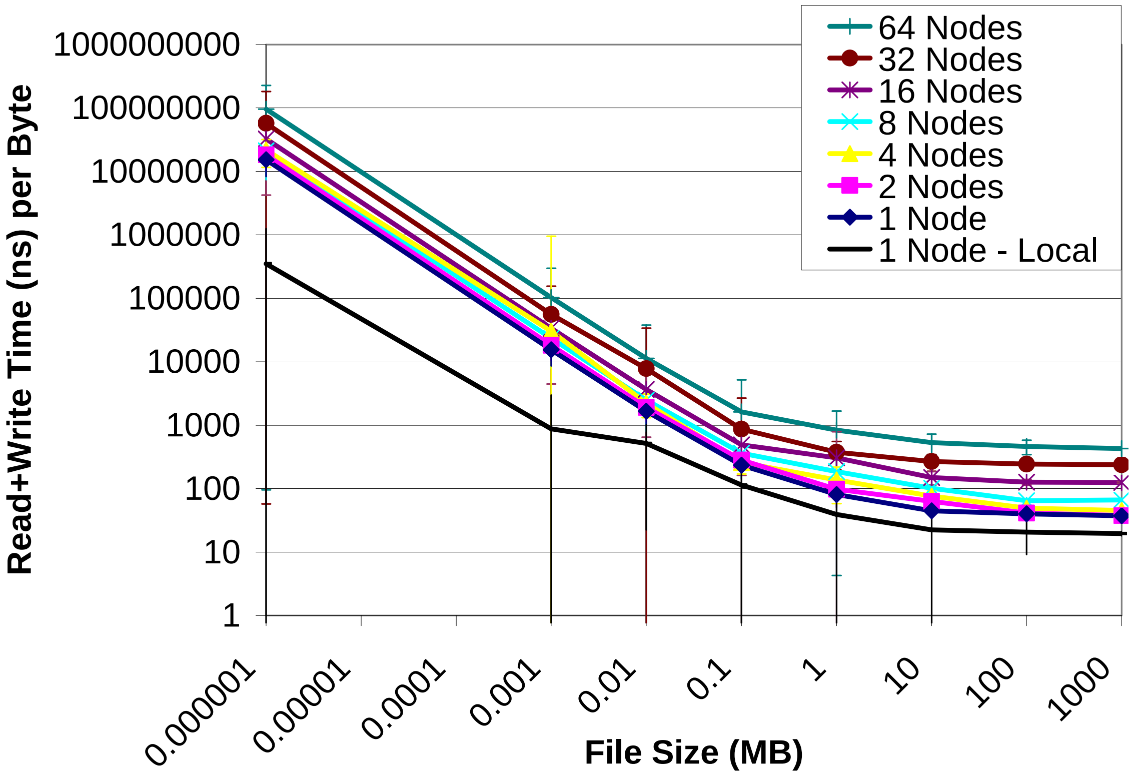

Figure 3 (gpfs model 1-64 nodes sum r+w t): Read+write performance expressed in time (in nanoseconds) per byte; both axes are logarithmic; 1-64 nodes for GPFS, and 1 node for local disk access; 1B – 1GB files

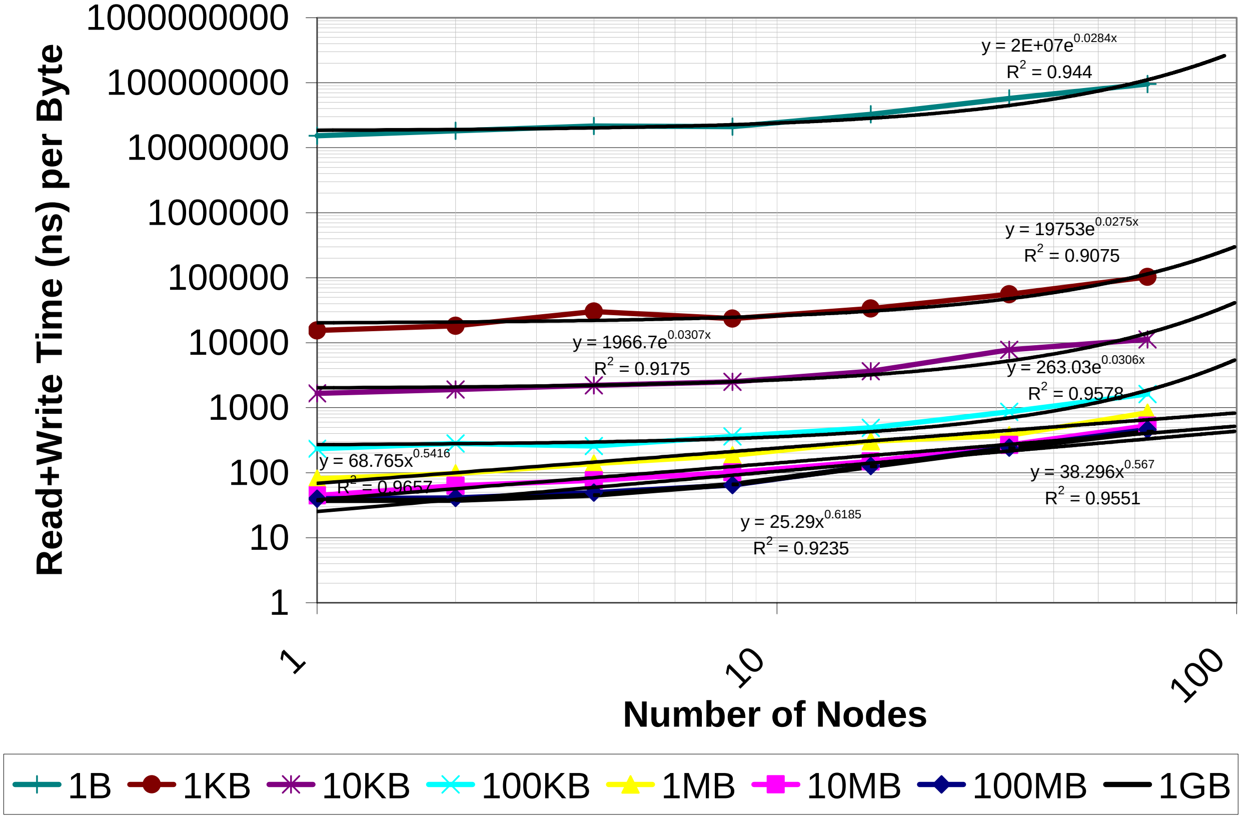

Figure 4 (gpfs model 1-64 node sum r+w t2): Same as figure 3, but plotting the number of nodes on the x-axis instead of the file size; also shows the function approximating the corresponding data, including the R^2 value which can evaluate how well the approximation fits the data (value of 1 means perfect fit)

Notice that the GPFS read performance (Figure 5) tops out at around 3400 Mb/s for large files, and it can achieve almost 3Gb/s throughput with files as small as 100KB if there are enough nodes concurrently accessing GPFS. Its worth noting that the performance increase beyond 8 nodes is only apparent for small files; for large files, the difference is small. This is most likely due to the fact that there are 8 I/O servers serving GPFS (I believe the I/O servers are identical in configuration as the compute nodes), and 8 nodes are saturating the 8 I/O servers.

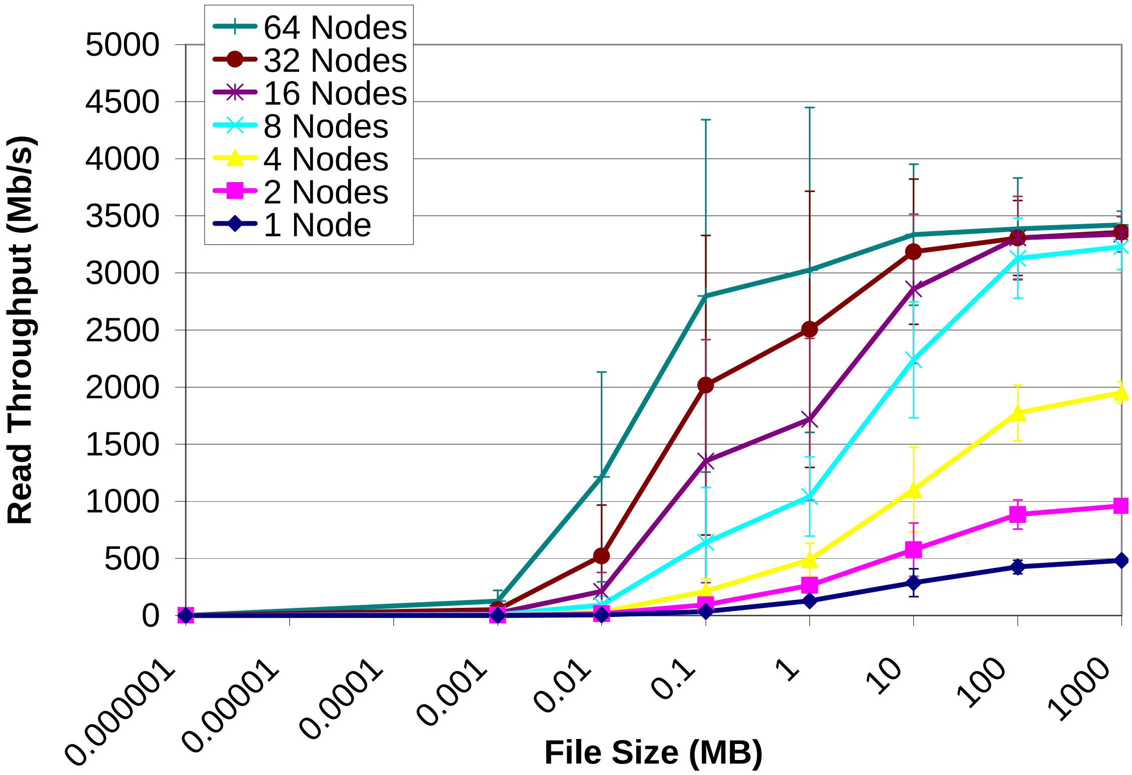

Figure 5 (gpfs model 1-64 nodes sum r): Read performance for GPFS expressed in Mb/s; only the x-axis is logarithmic; 1-64 nodes for GPFS; 1B – 1GB files

The read+write performance is not nearly asgood as the read performance, as it tops out just over 1Gb/s. Just as in the read experiment above, there seems to be little gain from having more than 8 nodes concurrently accessing GPFS (with the exception of small files).

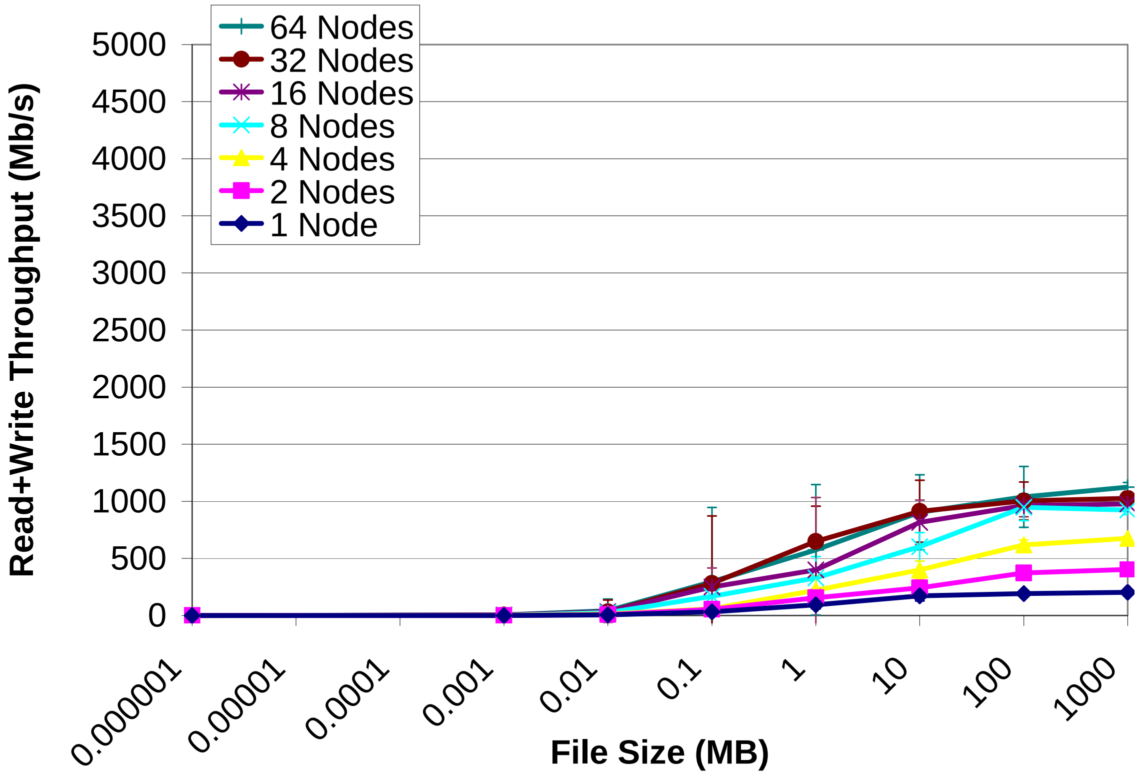

Figure 6 (gpfs model 1-64 nodes sum r+w): Read+write performance for GPFS expressed in Mb/s; only the x-axis is logarithmic; 1-64 nodes for GPFS; 1B – 1GB files

Figure 7 shows the performance of a single node, when running the benchmarks on GPFS and the local disk for both read and read+write. Its worth noting that local disk performance is in general double that of GPFS, but it seems to have more variability.

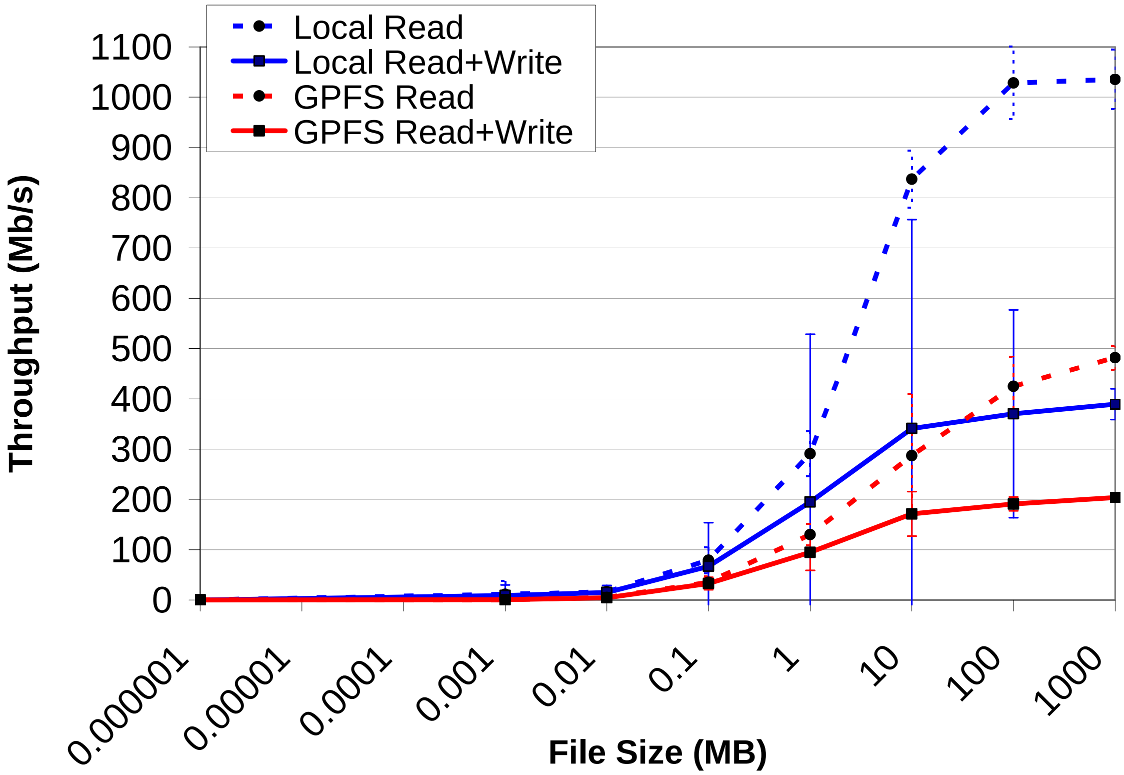

Figure 7 (local vs gpfs 1 node): Read and read+write performance expressed in Mb/s comparing local disk and GPFS performance for 1node; only the x-axis is logarithmic; 1 node for GPFS, and 1 node for local disk access; 1B – 1GB files

Figure 8 shows the same data as Figure 7, but as time (nanoseconds) per byte.

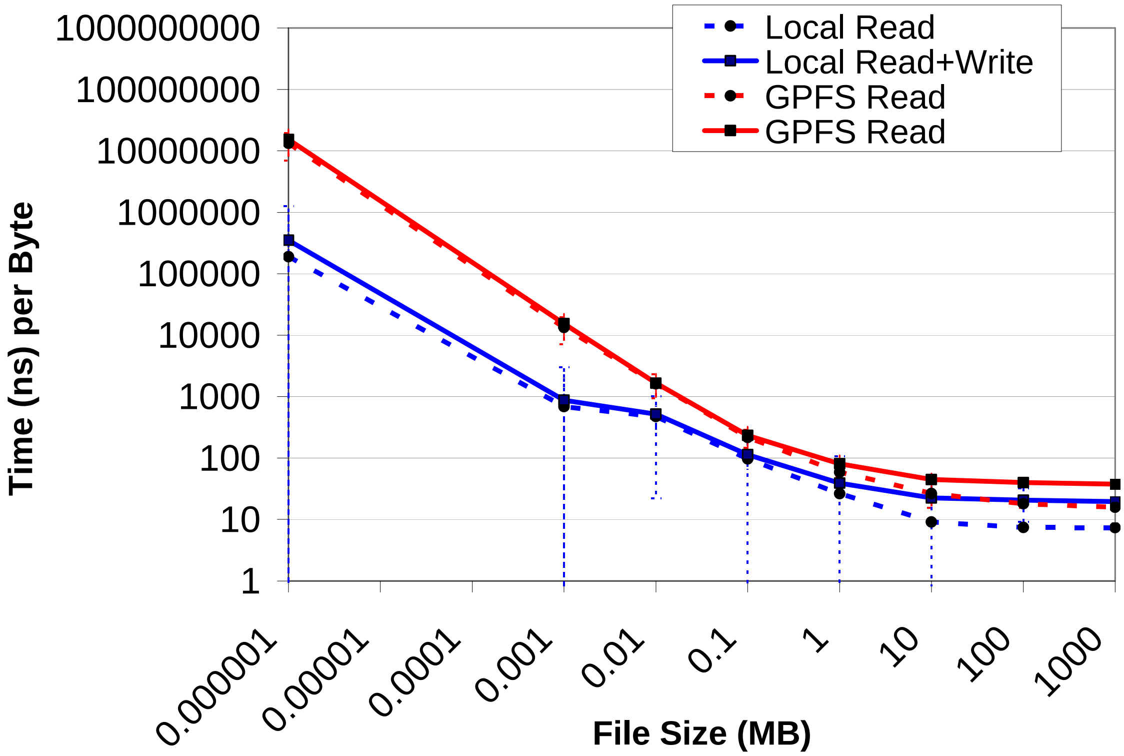

Figure 8 (local vs gpfs 1 node time_byte): Read and read+write performance expressed in time (in nanoseconds) per byte comparing local disk and GPFS performance for 1node; both axes are logarithmic; 1 node for GPFS, and 1 node for local disk access; 1B – 1GB files

These final two graphs shows the theoretical read+write and read throughput (measured in Mb/s) for local disk access. These results are theoretical, as they are simply a derivation of the 1 node performance, extrapolated to additional nodes (2, 4, 8, 16, 32, 64) linearly (assuming that local disk accesses are completely independent of each other across different nodes). Notice the read+write throughput approaches 25GB/s (up from 1Gb/s from GPFS) and the read throughput 76Gb/s (up from 3.5Gb/s for GPFS). This upper bound potential is huge if we can just harness it via Falkon!

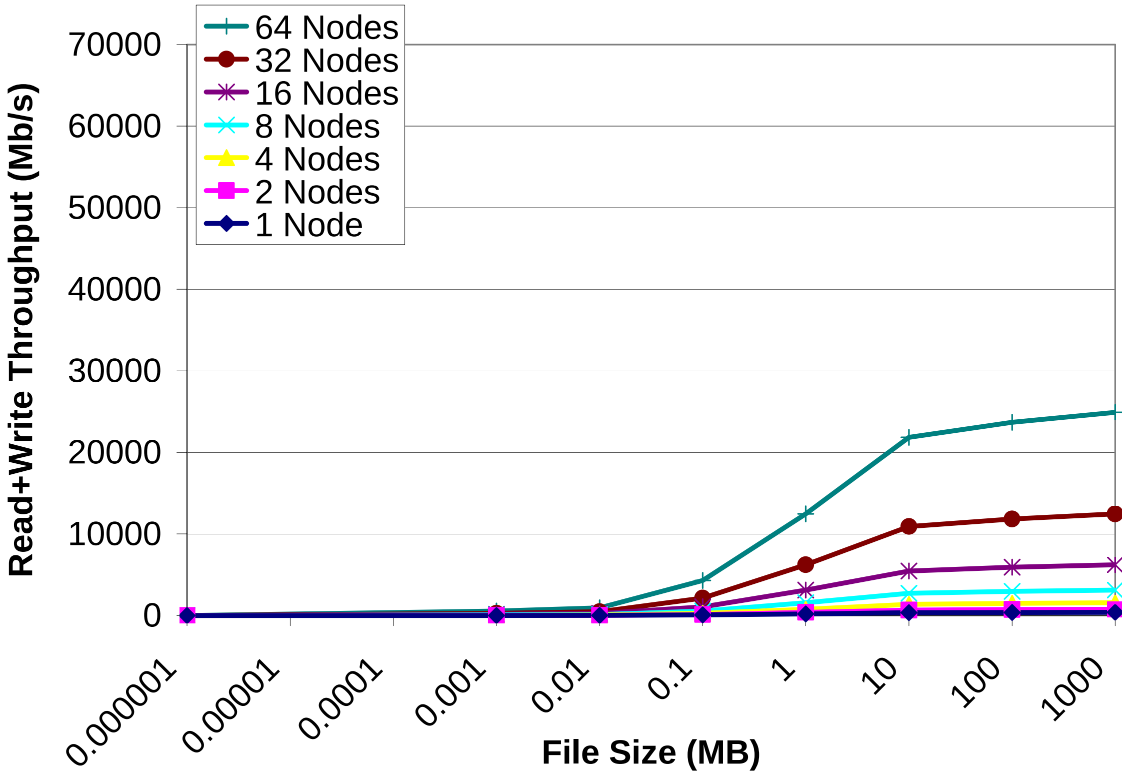

Figure 9 (local model 1-64 nodes r+w): Theoretical read+write performance of local disks expressed in Mb/s; only the x-axis is logarithmic; 1-64 nodes for local disk access; 1B – 1GB files

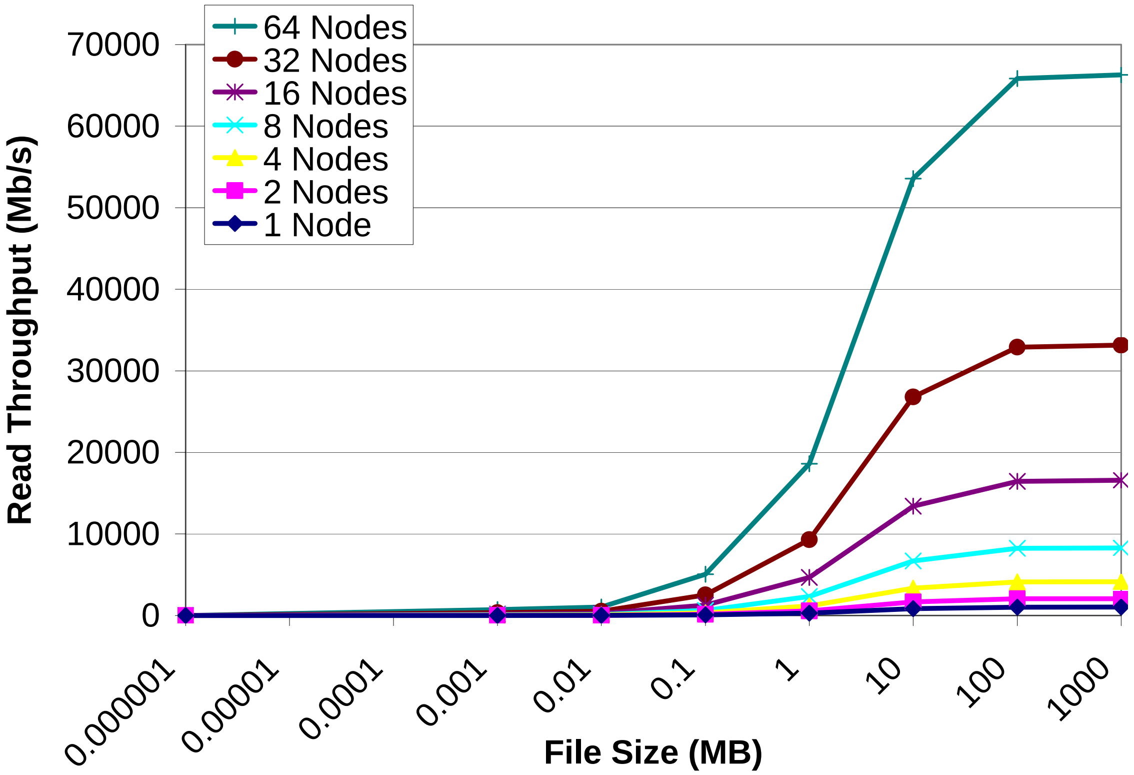

Figure 10 (local model 1-64 nodes r+w): Theoretical read performance of local disks expressed in Mb/s; only the x-axis is logarithmic; 1-64 nodes for local disk access; 1B – 1GB files