Redbooks Paper

Raymond L Paden

Native GPFS Benchmarks in an Integrated p690/AIX and x335/Linux Environment

Abstract

A series of benchmark tests were completed during October 4 - 6, 2004, at the IBM® Benchmarking Center in Poughkeepsie, NY. The primary goal of these tests was to evaluate GPFS performance for HPC applications in a mixed p690/AIX® and x335/Linux® environment. A secondary goal was to assess how well GPFS scales by adding additional storage controllers.

The author

Raymond L. Paden, Ph.D.
HPC Technical Architect, Deep Computing

Technical contributors to this benchmark

Andrey Ermolinskiy
Bill Hartner
Tom Henebery
Lerone LaTouche
Gautam Shah

Executive overview

The goals of these benchmark tests were to evaluate GPFS performance for HPC applications in a mixed p690/AIX and x335/Linux environment, and to assess how well GPFS scales by adding additional storage controllers. The only internode communication mechanism was Gigabit Ethernet (GbE); there were no HPS, Myrinet or Quadrics switches, for example.

DS4500 (FAStT900)/DS4000 (EXP700) Fibre Channel (FC) storage was used in this testing.

Note: The IBM model numbers for the FAStT900 and EXP 700 have been changed to DS4500 and DS4000, respectively. The new model numbers are used in this report.

The DS4500s were directly attached to the p690, which in turn acted as a Network Storage Device (NSD) server for the x335 nodes via GbE. EtherChannel was used to aggregate multiple GbE adapters under a single IP address on the p690, since GPFS can only work over a single IP address per node; EtherChannel effectively increased disk I/O bandwidth (BW) to/from the p690. A system was thus created in which GPFS was running in a mixed AIX/Linux environment.

The GPFS benchmark tests that were conducted in this study included the following:

- I/O tests done on the p690 with directly attached disk
- I/O tests done on the x335 cluster accessing the disk attached to the p690
- I/O tests executed simultaneously on the p690 and x335 cluster with disk attached to the p690

When interpreting the test results, keep in mind that there was just one GPFS file system and it was mounted natively on the p690 and the x335 cluster at the same time (it was not exported to the x335 nodes via NFS).

Overall, the results of the tests were acceptable, generally meeting or exceeding expectations. However, one test did not meet expectations, but its results were “good enough” for many applications.

System configuration

The benchmark configuration was an integrated cluster with one AIX/p690 node, 32 Linux/x335 nodes and four DS4500 disk controllers. Table 1 contains a detailed list describing the major cluster components and their configuration (also refer to Figure 1 on page 3). The actual number of DS4500s used in any test was determined by which disks were selected to build the file system.

Table 1 Major benchmark system components

Quantity	Description	Specification
1	Compute client and NSD server	p690 (32-way POWER4™ @ 1.9 GHz, 128 GB RAM, no LPARs[a]) ports 16 FC HBAs, 8 GbE ports AIX 5.2, GPFS 2.2
32	Compute client	X335 (2-way Xeon @ 3.06 GHz, 4 GB RAM) 1 GbE port Linux 2.4.21-9.ELpok, RHEL 3.0, GPFS 2.2, PVM 3.4
1	GbE switch	Cisco 6509
1	SAN switch	Brocade 2109 F32
4	Disk controller	DS4500 (with 4 host side FC connections)
16	Disk “drawers”	DS4000 (14 disks @ 73 GB and 15 Krpm)

a. With only one p690, it should have had at least two LPARs in order to provide primary and secondary NSD servers for GPFS. However, due to scheduling constraints it was not possible to configure it this way.

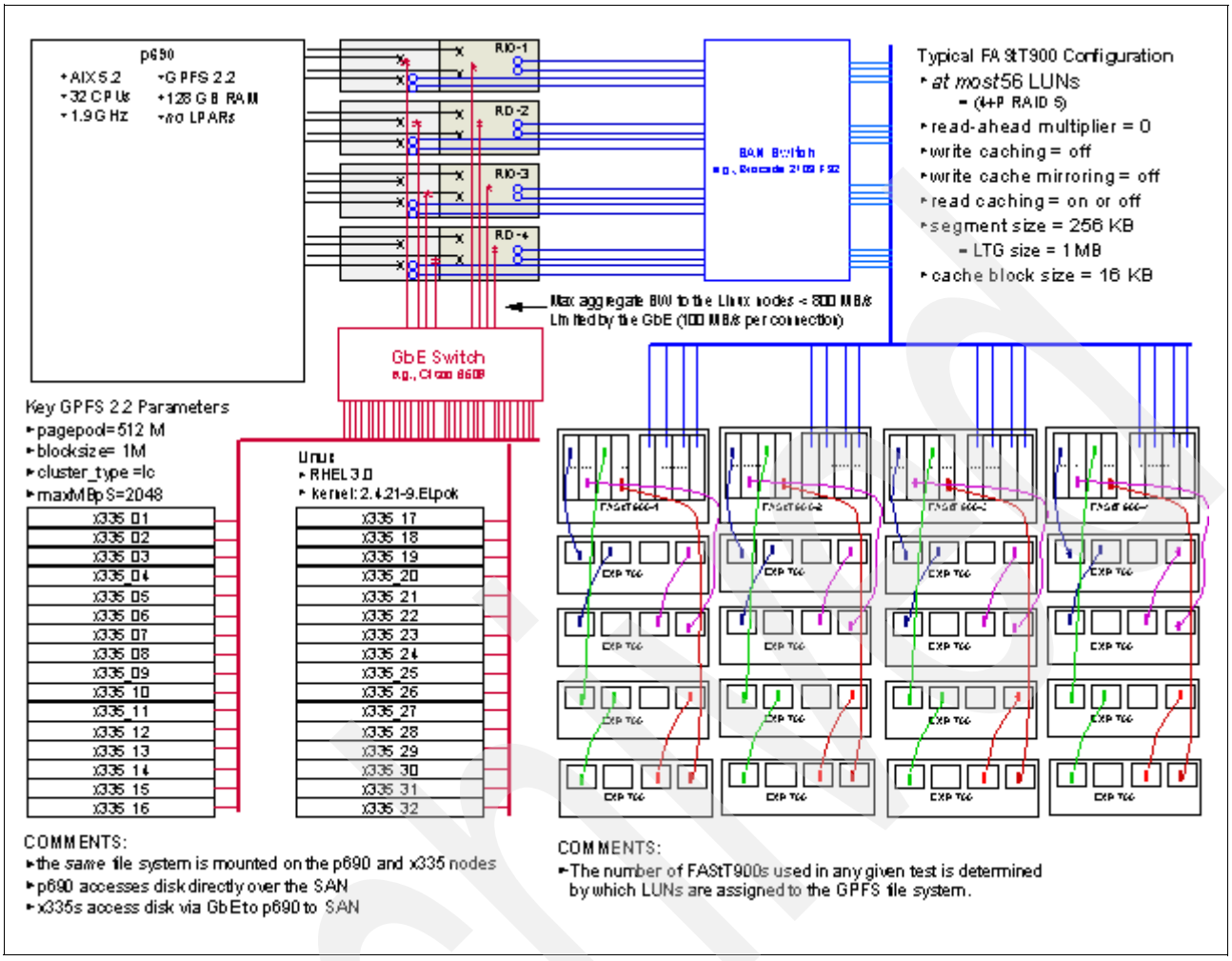

Figure 1 Benchmark cluster configuration.

Disk and file system configuration

The storage subsystem consisted of the following major components:

- One NSD server node (p690)
  - 16 FC HBAs (four dedicated to each DS4500
- One FC switch (SAN switch)
- Four disk controllers (DS4500)
- 16 “drawers” (DS4000), 4 per controller, with 14 disks/drawer

The DS4500s were directly attached to the p690 via the SAN switch. The LUNs (hdisks) presented to the p690 by the DS4500s became NSDs in a one-to-one[1] manner, and the p690 was the server for them (that is, the p690 was the NSD server).

Applications running on the p690 accessed the file system locally, which accessed the disks locally via the GPFS daemon using the local NSDs. Moreover, applications running on the x335 nodes accessed the file system locally, which accessed the disks remotely (the disks

[1] For example, using default naming conventions, hdisk42 *could* map to gpfs42nsd; however, there is no guarantee that the integer in the name is identical. Both the hdisk and the NSD would be found in /dev.

that are attached to the NSD server) via a local GPFS daemon using local NSDs; however, the local NSDs used the NSD software layer below the GPFS daemon to route storage I/O packets via the GbE network to the NSD server where they find their way to the disk.

The result of this configuration was to mount one global file system (the same global file system!) simultaneously on the p690 running AIX and the x335s running Linux; GPFS was running native semantics everywhere[2].

In a production environment, ideally there should be at least two NSD servers (a primary and secondary NSD server for each LUN) in order to guarantee GPFS reliability. However, since the resources required for this were not available for use in this benchmark, it was conducted using only a single p690 as the NSD server.

The disks attached to each DS4500 were configured as 4+P RAID5 arrays (each RAID5 array was a LUN). Unless stated otherwise, the key DS4500 parameters[3] were set as follows.

- read-ahead multiplier = 0
- write mirroring = off
- write caching = off
- read caching = off
- segment size = 256 KB
- cache block size = 16 KB

These settings yield optimum *safe* performance.

Important: If the safety requirement is relaxed, write performance can be improved significantly by enabling write caching. The problem associated with enabling write caching is that the DS4500 maintains a volatile disk cache distributed between two distinct RAID controllers. When a write operation is completed (including flushing buffers), control is returned to the application when the data is written to this cache; it need not yet be on disk. If one of these RAID controllers fails, then the data in that cache will be lost; since this can include metadata, the entire file system can be corrupted.

Given the potentially large size of GPFS file systems (for example, 100s of terabytes or more) and the fact that files are striped over all of the disks, the magnitude of the risk is multiplied. Enabling write mirroring can compensate for this risk, but write mirroring compromises the performance gains associated with write caching.

The key GPFS parameter settings impacting performance include the following.

- pagepool = 512 M
- blocksize = 1M
- cluster_type = lc
- maxMBpS = 2048

Since LUNs are associated with a given DS4500, the number of DS4500s used in a given test is controlled by selecting which LUNs (that is, the NSDs) should be included in the file system via the **`mmcrfs`** command.

[2] NFS is not being used!

[3] The settings for these parameters are atypical to general GPFS guidelines, but effective nonetheless. See REDP-3909-00 for details on standard GPFS recommendations for the DS4500.

Cluster network configuration

The only network available for node-to-node communication in this cluster was a GbE network. Each x335 node has a single GbE port configured, which was used in these benchmarks for both message passing and GPFS NSD communication to the p690. The p690 has eight GbE ports aggregated into a single IP address as an EtherChannel. In this benchmark, it was used primarily for GPFS NSD communication to the x335 nodes.

On the p690, the GbE adapters are set to use jumbo frames with an MTU = 9000. For the EtherChannel (the GbE interfaces,) the following parameters are set as shown:

- rfc1323 = 1
- tcp_nodelay = 1
- tcp_sendspace = 1310720
- tcp_recvspace=1310720

The EtherChannel is configured to use standard routing; ad hoc tests indicated that round robin routing is less effective.

On the x335 nodes, the GbE adapters were set to use jumbo frames with an MTU = 9000. Also associated with the GbE configuration, the following line was contained in the /etc/modules.conf file:

Example 1 GbE configuration settings in /etc/modules.conf

```
options bcm5700 mtu=1500,9000 adaptive_coalesce=0,0 rx_coalesce_ticks=1,1
rx_max_coalesce_frames=1,1 tx_coalesce_ticks=1,1 tx_max_coalesce_frames=1,1
```

The following stanzas were contained in the /etc/sysctl.conf file:

Example 2 GbE configuration settings in /etc/sysctl.conf

```
# Improve TCP Performance

net.core.rmem_max = 262144
net.core.rmem_default = 262144
net.core.wmem_max = 262144
net.core.wmem_default = 262144
net.ipv4.tcp_window_scaling = 1

# Increase Shared Memory

kernel.shmmax = 134217728
kernel.shmall = 134217728
```

Results and analysis

Benchmark application description

Benchmark code description

The benchmark codes used in this study are custom, synthetic codes designed and programmed by the lead author to evaluate common storage I/O paradigms used in seismic processing as well as most other HPC market segments. These codes have been used many times to evaluate I/O designs for seismic processing and other HPC customers. They

correlate very closely with other benchmark codes like xdd[4] or iozone[5] when parameterized similarly. They run on either AIX or Linux systems. They use either PVM or MPI for parallelism when it is needed. The name of this benchmark is ibm.v3g (I/O Benchmark, version v3g); refer to “Additional material” on page 17 for information about downloading this benchmark code.

Benchmark parallelism

There are two classes of nodes in the configuration used for this study. The first is a p690; having 32 processors, it is well suited for SMP parallelism. The second is an x335; having two processors, it is more inclined toward distributed or cluster parallelism [see Pfister98]. It is commonplace for customers to schedule only one task per node when using an x335.

Given the different architectures of these node types, multiple tasks were scheduled per node on the p690, while only one task was scheduled per node on the x335s. The benchmark code used in this study uses message passing for communication between tasks on both classes of nodes. However, message passing was used only to distribute parameters at the beginning of a job and to collect statistics at the end of the job, and therefore was an inconsequential aspect of these jobs. In this study, no jobs were submitted which spanned both node classes, though tests were executed in which two jobs were submitted at the same time, one on the p690 and one on the x335 nodes.

Metrics used to assess results

Performance was measured as bandwidth in units of MB/s for these tests. To measure bandwidth, one must therefore measure data volume and time.

Regarding data volume, unless stated otherwise, each task in a job accessed 4 GB data, with the total data being accessed for the job equal to the number of tasks times the data per task; each task in a job accessed the same file at the same time (that is, a multi-task job did parallel I/O). For example, a four-task job accessed 16 GB of data.

Time was assessed using wall clock time starting at a beginning and terminating at an end. Thus no distinction was made between user, system, and real time. Since the test jobs were parallel, time was collected on a per task basis, and no task started its timer until all tasks were spawned, their parameters initialized, and their file opened; because the timer starts immediately following a barrier, all tasks start at approximately the same time.

The task’s timer was stopped after the last record had been accessed and the file has been closed. Because of the stochastic nature for the various queuing systems associated with parallel tasks, under normal circumstances tasks can and do terminate with some variance[6], though on the whole they remain uniformly active.

Three different measures of bandwidth were used to assess performance:

Natural aggregate Total data divided by the time of the longest task

Harmonic mean Total data divided by the sum of the task times

Harmonic aggregate The harmonic mean multiplied by the number of tasks

The harmonic aggregate has the effect of “smoothing out” the variance across the tasks of a job and gives a more typical rate as viewed from a system perspective (it is generally consistent with performance monitors like iostat measuring I/O rates at the LUN level). It cancels the effects of outliers. The natural aggregate is the I/O rate as viewed from a job

[4] http://www.ncsa.uiuc.edu/~aloftus/Scripts/Filesystem_Testing/xdd60.pdf
[5] http://www.iozone.org/
[6] The variances were not that extreme in this study.

perspective; from a system perspective, this rate is often lower than the actual system rate since it is heavily biased by outliers.

Access patterns

An access pattern is determined by the size of the records and the order in which records are read or written in a file. Within this benchmark, two record sizes and several patterns were used.

- Large record: 1 MB or 1024 KB
- Small record: 16 KB
- Sequential: the file is partitioned into N subsets (N = number of tasks), and each task accesses its records contiguously
- Strided: skip 100 records, access the next record and repeat the pattern, wrapping around the file till all records are accessed once

This benchmark code is capable of testing other access patterns including random or semi-random; hints can be supplied to improve the performance of these patterns. Given time constraints, random and semi-random patterns were not tested.[7]

Access patterns have a significant impact upon performance. Generally speaking for GPFS, large record patterns, especially sequential, produce the best results. However, due to caching algorithms in GPFS, a small record sequential access pattern can often perform nearly as well. These patterns are frequently used in evaluating file system performance. Unfortunately, many applications cannot adopt these optimal access patterns and thus other access patterns were tested.

At the other extreme in terms of performance is small record irregular patterns (that is, random or semi-random). These are patterns for which no optimizations exist to improve their performance and which also violate working set rules rendering caching algorithms ineffective. In many cases, however, the application can be written in such a manner as to predetermine when records will be accessed. If this can be done, this information can be passed to the file system and GPFS can pre-fetch the records asynchronously significantly improving performance.

Strided access patterns lie in between the extremes of small record random and large record sequential. GPFS has optimizations which can recognize this pattern and pre-fetch records, for example, to improve performance without the explicit use of hints. This access pattern commonly occurs in seismic applications, for example.

Results and analysis for each test set

In this benchmark there were three basic sets of tests done; the item distinguishing each of these test sets was the number of DS4500s used. Recall that there are 42 physical disks associated with each DS4500. Since they were configured as 4+P RAID5 arrays, there were eight LUNs per DS4500. The number of DS4500s used in a test was controlled by creating a file system only using LUNs from the DS4500s to be used in the test.

Due to scheduling constraints (limited access time on the benchmark system), some of the planned tests intended could not be conducted. Also, a few other ad hoc tests were performed.

[7] See REDP-3909-00 for tests using small record random access patterns. Contact the lead author for other unpublished technical reports describing other various random pattern results.

One DS4500

The following tests were conducted with the GPFS file system accessing a single DS4500; the file system was comprised of eight LUNs. The DS4500 and GPFS parameters were set as previously described.

Two tests (Figure 2 on page 8) compared a large record (1 MB) sequential access pattern on the p690 and on x335 nodes. The performance profiles are quite different. On the p690, the bandwidth remains nearly the same for 1 to 16 tasks; approximately 330 MB/s for writes and around 400 MB/s for reads (which exceeds the standard sizing recommendation for a single DS4500 under GPFS[8]).

By contrast, the bandwidth on the x335 nodes scaled, reaching its peak at four nodes. The reason for this scaling is that access to the NSD server (the p690) is gated by the single GbE adapter on each x335; its peak sustained performance in isolation was about 110 MB/s. The peak aggregate read performance is then gated by a single DS4500 at 390 MB/s.

What is not clear is why the write performance levels off at around 260 MB/s, since the same DS4500 yields 330 MB/s on the p690. This observation is explored in greater detail in “Two DS4500s with write caching disabled” on page 9. Note that the size of the files significantly exceeds the size of the pagepool (GPFS file cache); therefore caching effects can be ignored in this analysis.

Test #1	**p690**	Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
1	1	325.05	388.97					4
1	2	336.30	397.73	325.74	388.19	162.87	194.10	8
1	4	330.86	402.15	319.38	387.25	79.84	96.81	16
1	8	328.93	417.80	316.73	385.73	39.59	48.22	32
1	16	328.04	392.83	316.22	390.09	19.76	24.38	64

Test #2	**x335**	Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
1	1	109.27	110.89					4
2	1	180.27	219.73	181.18	219.81	90.59	109.91	8
4	1	256.28	388.48	257.37	389.79	64.34	97.45	16
8	1	260.75	390.53	260.93	392.65	32.62	49.08	32
16	1	266.58	390.52	267.51	392.00	16.72	24.50	64

Figure 2 Large record sequential access pattern tests; a p690 vs. x335 nodes with one DS4500. The record size = 1MB. Write caching is disabled. Compare these results to Figure 4 on page 11 and Figure 8 on page 14.

Another test (see Figure 3 on page 9) was conducted in which two jobs were submitted. The access pattern was large record (that is, 1 MB) sequential. The first job is on the p690 and the second job is on the x335 nodes; write and read jobs are submitted separately. Each job was first executed individually to establish a baseline for comparison, then again at the same time. The file sizes were normalized so that the execution time for each job was roughly the same.

[8] Standard conservative sizing recommendation for GPFS on a DS4500 is 300 MB/s per DS4500, assuming it has four FC connections to at least two NSD servers and the access pattern is large record sequential.

The objective of this test was to determine if the jobs interfered with each other and reduced overall performance.

The aggregate harmonic was used measure this. Because the jobs did not finish at exactly the same time, this measure is slightly pessimistic (that is, 369 MB/s and 342 MB/s for the combined p690/x335 read and write jobs, respectively). However, by observing iostat, the measured rates were observed at 400 MB/s when both the p690 and x335 jobs were running at the same time. Therefore, aggregate performance for both jobs is gated by the single DS4500 used. The aggregate cluster thus delivered the maximum sustained aggregate performance possible by this storage system, without starving either job.

Test #5			Natural Aggregate		Harmonic Aggregate		Harmonic aggregate over x335 and p690		
	Number of Nodes	Number of tasks per node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	Write	Read	File Size GB
x335 only	4	1	253.85	389.03	254.11	390.08	N/A	N/A	16
p690 only	1	4	313.95	381.33	324.05	402.66	N/A	N/A	16
x335 rates with p690	4	1	167.23	267.50	167.23	268.30			28**
p690 rates with x335	1	4	196.76	199.97	195.32	200.35	369.01*	341.64*	28**

Note * job times are not identical; iostat showed aggregate rate ~= 400 MB/s
Note ** Size of combined files; file sizes adjusted so job times were approximately equal.

Figure 3 Jobs were executed separately on each of the architectures in order to establish baseline performance; their results are reported on the rows labeled "x335 only" and "p690 only". The jobs were then repeated at the same time. The row labeled "x335 rates with p690" shows the x335 rates while the p690 job is running at the same time; the row labeled "p690 rates with x335" shows the opposite. The access pattern is sequential with record size = 1 MB. Write caching is disabled. Compare these results to Figure 6 on page 12 and Figure 9 on page 15.

Refer to "Appendix B - Summary of benchmark tests" on page 19 for the results of additional single DS4500 tests.

Two DS4500s

The following tests were conducted with the GPFS file system accessing two DS4500s controllers; the file system was comprised of 16 LUNs. The DS4500 and GPFS parameters were set as previously described with the exception of write cache; some of the experiments were repeated with and without write caching enabled.

Two DS4500s with write caching disabled

Two tests (Figure 4 on page 11) compared a large record (1 MB) sequential access pattern on the p690 and on x335 nodes while using two DS4500s with write caching *disabled*. As with the single DS4500 case, the performance profiles were quite different.

On the p690, the bandwidth remained nearly the same for 1 to 16 tasks; approximately 640 MB/s for writes and around 740 MB/s for reads (which exceeds the standard expectation for two DS4500s under GPFS[9]). By contrast, the bandwidth on the x335 nodes scaled, reaching its peak at eight nodes for the read case and four nodes for the write case. As explained

[9] Standard conservative sizing recommendation for GPFS on two DS4500s is 600 MB/s subject to standard assumptions (see previous footnote).

previously, the reason for this scaling is in part due to the fact that access to the NSD server (the p690) is gated by the single GbE adapter on each x335 (each with an isolated peak sustained performance of about 110 MB/s).

The sustained peak aggregate performance on the x335 nodes is gated as well, but the reasons for the gating *appear* to be different for the reads and writes. Concerning reads, the peak sustained performance was 624 MB/s reached at eight nodes, while the expected performance on two DS4500s is roughly 700 MB/s. Note that the storage I/O access into the NSD server is over an EtherChannel with eight ports; if one accepts perfect scaling based on the single x335 node test, that would yield an EtherChannel peak performance at < 900 MB/s (iperf write tests are consistent with this; see “iperf test - x335 GPFS write tests did not meet expectations” on page 16). If one looks ahead at the similar four DS4500 case (see “Four DS4500s” on page 13), its peak sustained performance was 692 MB/s reached again at eight nodes, while the expected performance of four DS4500s is 1400 MB/s.

It would appear, then, that the gating factoring is the EtherChannel combined with reasonable bandwidth loss attributed to normal GPFS parallel overhead. Another “rule of thumb” used by the lead author is that when sizing a GPFS cluster using a GbE network, one should roughly amortize GPFS traffic at 80 MB/s[10] per GbE port; with eight ports in the EtherChannel, that works out to 640 MB/s. In short, the read tests met expectations.

The write test analysis was less optimistic. In that case, the peak sustained aggregate performance was approximately 270 MB/s and was reached at four nodes. It stayed flat up to 32 nodes. Similar behavior is seen in the four DS4500 case (see “Four DS4500s” on page 13). Using the iperf network test tool (see “iperf test - x335 GPFS write tests did not meet expectations” on page 16), an aggregate rate ~= 900 MB/s was measured writing over eight GbE ports (one per node).

Thus, it seems unlikely that the network was the gating factor. While the cause of this performance shortfall requires further investigation, this rate may be good enough for many applications.

[10] The lead author has been challenged by some technical people for using 80 MB/s per GbE port for GPFS traffic amortization; by contrast they have argued for figures closer to 100 MB/s. Experience compels the lead author to use the more pessimistic number. See REDP-3909-00 and other unpublished reports by the lead author (contact him for details) for confirmation of the 80 MB/s figure.

Test #6	p690	Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	Write	read	sizeof(file) GB
1	1	640.10	765.28					4
1	2	641.07	760.75	648.02	761.71	324.01	380.85	8
1	4	631.84	739.31	646.62	758.13	161.65	189.53	16
1	8	649.14	724.88	670.83	755.63	83.85	94.45	32
1	16	646.97	721.26	671.29	788.48	41.96	49.28	64

Test #7	x335	Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	Write	read	sizeof(file) GB
1	1	109.27	110.89					4
2	1	198.92	218.65	198.92	219.51	99.46	109.76	8
4	1	269.91	435.37	269.95	437.68	67.49	109.42	16
8	1	282.44	624.19	283.44	626.81	35.43	78.35	32
16	1	253.23	595.50	281.78	598.18	17.61	37.39	64
32	1	269.77	577.53	269.83	581.23	8.43	18.16	128

Figure 4 Large record sequential access pattern tests; a p690 vs. x335 nodes with two DS4500s. Record size = 1 MB. Write caching was disabled. Compare these results to Figure 2 on page 8 and Figure 8 on page 14.

Due to time limitations, it was not possible to fully test small record strided and small record random access patterns thoroughly. However, a few small record strided (16 KB) tests were included for completeness. Figure 5 summarizes one of these tests for a p690. The stride for this test was 1.56 MB and the block size was 1 MB. A task accessed 16 KB, skipped 1.56 MB, and accessed the next record and so on. As such, only subblocks were accessed (a subblock = (1/32) * block size). Compared to tests on SP systems[11] and Linux clusters [see Paden04a], these results were more irregular.

		Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
1	1	25.44	35.79					2
1	2	18.14	54.53	18.16	55.00	9.08	27.50	4
1	4	17.34	23.22	17.44	23.87	4.36	5.97	8
1	8	18.88	26.80	19.18	28.01	2.40	3.50	16
1	16	20.48	34.95	20.71	36.06	1.29	2.25	32

Figure 5 Small record strided access pattern tests on a p690. Record size = 16 KB. Write caching is disabled.

As with the single DS4500 case, a test (see Figure 6 on page 12) was conducted in which two jobs were submitted individually and again at the same time. The access pattern was large record (1 MB) sequential. The first job was submitted on the p690 and the second was

[11] These results are described in unpublished technical reports; contact the lead author for details.

submitted on the x335 nodes. The file sizes were normalized so that the execution time for each job was nearly the same. The objective of this test was to determine if the jobs interfered with each other and reduced overall performance.

The aggregate harmonic was used measure this. The write rate was 707 MB/s and the read rate was 802 MB/s for the combined p690/x335 jobs. These rates were slightly greater than the rates for jobs executed only on the p690 alone for a read or write job, which essentially was the gating rate of the two DS4500s. The aggregate cluster thus delivered the maximum sustained aggregate performance possible by this system, without starving either job.

Test #10			Natural Aggregate		Harmonic Aggregate		Harmonic aggregate over x335 and p690		
	Number of Nodes	Number of tasks per node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	File Size GB
x335 only	4	1	267.33	435.72	268.29	437.23	N/A	N/A	16
p690 only	1	4	632.45	771.00	650.98	788.25	N/A	N/A	16
x335 rates with p690	4	1	233.03	386.27	233.03	389.61			24**
p690 rates with x335	1	4	470.22	407.79	477.30	412.81	707.48*	801.66*	32**

note * Job times are nearly identical; therefore, iostat measured rate is close to harmonic aggregate rate.

note ** Size of combined files from each job; file sizes adjusted so job times were approximately equal. Combined files for write = 24 GB, combined files for the read = 32 GB.

Figure 6 Jobs were executed separately on each of the architectures in order to establish baseline performance; their results are reported on the rows labeled "x335 only" and "p690 only". The jobs were then repeated at the same time. The row labeled "x335 rates with p690" shows the x335 rates, while the p690 job is running at the same time; the row labeled "p690 rates with x335i"shows the opposite. The access pattern is sequential with record size = 1 MB. Write caching is disabled. Compare these results to Figure 3 on page 9 and Figure 9 on page 15.

Two DS4500s with write caching enabled

As explained in "Disk and file system configuration" on page 3, it is not recommended that write caching be enabled on a DS4500 when using GPFS. However, since some customers are willing to accept the risk associated with it enabled, limited tests using two DS4500s were conducted with write caching enabled. The structure of these tests was similar to the former dual DS4500 tests. Only the results for the large record sequential write tests are documented in this report[12].

As can be readily seen, comparing Test #11 in Figure 7 on page 13 with Test #6 in Figure 4 on page 11, the write results on the p690 improved significantly (from roughly 645 MB/s up to 770 MB/s, an improvement of approximately 20%). It should also be noted the write rates are similar to the corresponding read rates.

However, comparing Test #12 in Figure 7 on page 13 with Test #7 in Figure 4 on page 11, the results for the x335 did not change appreciably. This is a test artifact, attributable to the fact that the write performance was gated by the EtherChannel below the peak rates possible by the two DS4500s. By comparison, in other tests using a DS4500 directly connected to an x335 with write caching enabled, the write rates and read rates were equivalent [see

[12] Refer to "Appendix B - Summary of benchmark tests" on page 19 for complete test results.

Paden04b]. Thus it is fair to conclude that had the write rates on the x335 nodes *not* been gated, a performance improvement similar to the p690 case would have been observed.

Test #11	**P690**	Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	Write	read	sizeof(file) GB
1	1	768.21						4
1	2	774.97		775.80		387.90		8
1	4	767.94		784.99		196.25		16
1	8	776.23		810.23		101.28		32
1	16	721.80		755.05		47.19		64

Test #12	**X335**	Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	Write	read	sizeof(file) GB
1	1	110.19						4
2	1	189.97		189.97		94.99		8
4	1	271.39		272.78		68.19		16
8	1	270.35		301.89		37.74		32
16	1	286.21		287.15		17.95		64
32	1	280.56		282.29		8.82		128

Figure 7 Large record sequential access pattern tests with write caching enabled; a p690 vs. x335 nodes with two DS4500s. Record size = 1 MB. Compare these results to Figure 4 on page 11.

Four DS4500s

The following tests were conducted with the GPFS file system accessing four DS4500 controllers; the file system was comprised of 32 LUNs. The DS4500 and GPFS parameters were set as documented (write caching disabled). These tests were similar to the ones previously reported; however, only the results for the large record sequential tests on the p690 and x335 nodes and the mixed p690/x335 tests are documented in this report[13].

Two tests (Figure 8 on page 14) compare a large record (1 MB) sequential access pattern on the p690 and on x335 nodes while using four DS4500s with write caching disabled. As reported for the previous similar tests, the performance profiles differed significantly between these two configurations, but in a predictable manner. On the p690, the bandwidth was roughly the same for the write jobs over 1 to 16 tasks with an access rate, in the neighborhood of 1230 MB/s. The read jobs ran with greater variance in their access rates compared to the previous benchmark tests. The natural aggregate varied from 1259 MB/s up to 1541 MB/s, while the harmonic aggregate varied much less (from 1472 MB/s up to 1544 MB/s). However, the observed variance in these tests is not unmanageable. The rates for the four DS4500 tests are roughly double the rates for the two DS4500 tests.

By contrast, the bandwidth on the x335 nodes scaled, reaching its peak at eight nodes for the read case (approximately 690 MB/s) and four nodes for the write case (approximately 230 MB/s). The reason for this behavior is attributable the same factors as in the two DS4500 case. Because the peak bandwidth was gated by the same number of ports in the EtherChannel for both the two and four DS4500 tests, the access rates for the four DS4500 tests remained roughly the same as the two DS4500 tests.

[13] Refer to “Appendix B - Summary of benchmark tests” on page 19 for complete test results.

Test #16	p690	Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	Write	read	sizeof(file) GB
1	1	1233.84	1259.48					4
1	2	1205.22	1507.61	1207.57	1515.52	603.79	757.76	8
1	4	1220.48	1541.19	1221.24	1543.55	305.31	385.89	16
1	8	1230.79	1435.37	1255.06	1478.34	156.88	184.79	32
1	16	1261.76	1388.82	1314.27	1472.10	82.14	92.01	64

Test #17	x335	Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	Write	read	sizeof(file) GB
1	1	109.27	110.89					
2	1	177.47	215.71	177.61	217.44	88.80	108.72	8
4	1	207.42	427.55	236.79	432.29	59.20	108.07	16
8	1	244.68	692.17	245.77	698.91	30.72	87.36	32
16	1	249.66	574.91	250.03	610.65	15.63	38.17	64

Figure 8 Large record sequential access pattern tests with write caching enabled; a p690 vs. x335 nodes with two DS4500s. Record size = 1 MB. Write caching is disabled. Compare these results to Figure 2 on page 8 and Figure 4 on page 11.

As with the single and dual DS4500 tests, a test (see Figure 9 on page 15) was conducted in which two jobs were submitted individually and again at the same time for four DS4500s. However, in this case, eight task jobs were used instead of four task jobs as in the previous similar tests. The access pattern was large record (1 MB) sequential.

The first job was submitted on the p690 and the second was submitted on the x335 nodes. The file sizes were normalized, so that the execution time for each job was nearly the same. As before, the objective of this test was to determine if the jobs interfered with each other and reduced overall performance. The aggregate harmonic was used measure this. The write rate was 1231 MB/s and the read rate was 1702 MB/s; these rates were greater than or equal to the rates for jobs executed only on the p690 alone for a read or write job, which essentially is the gating rate of the four DS4500s. The aggregate cluster thus delivered the maximum sustained aggregate performance possible by this system, without starving either job.

Test #20			Natural Aggregate		Harmonic Aggregate		Harmonic aggregate over x335 and p690		
	Number of Nodes	Number of tasks per node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	File Size GB
x335 only	8	1	224.45	625.55	253.99	683.67	N/A	N/A	32
p690 only	1	8	1219.9	1358.7	1253.3	1404.9	N/A	N/A	32
x335 rates with p690	8	1	223.25	590.99	226.07	646.43			40*
p690 rates with x335	1	8	1067.1	991.81	1082.0	1011.3	1231.4	1702.2	48*

note * Size of combined files from each job; file sizes adjusted so job times were approximately equal. Combined files for write = 40 GB, combined files for the read = 48 GB.

Figure 9 Jobs were executed separately on each of the architectures in order to establish baseline performance; their results are reported on the rows labeled "x335 only" and "p690 only". The jobs are then repeated at the same time. The row labeled "x335 rates with p690" shows the x335 rates while the p690 job is running at the same time; the row labeled "p690 rates with x335" shows the opposite. The access pattern is sequential with record size = 1 MB. Write caching is disabled. Compare these results to Figure 3 on page 9 and Figure 6 on page 12.

Miscellaneous observations and tests

Scaling regarding the number of DS4500s

One of the objectives of this benchmark was to assess how well GPFS scaled when increasing the number of DS4500s. Due to the EtherChannel gating factor, the x335-based tests do not make a meaningful comparison. However, the p690 tests do. Comparing tests 1, 6, and 16 from Figure 2 on page 8, Figure 4 on page 11, and Figure 8 on page 14, respectively, the scaling can be seen to be linear. This is illustrated in Figure 10 on page 16 by selecting data from these figures for 1-, 4- and 16-task write and read jobs.

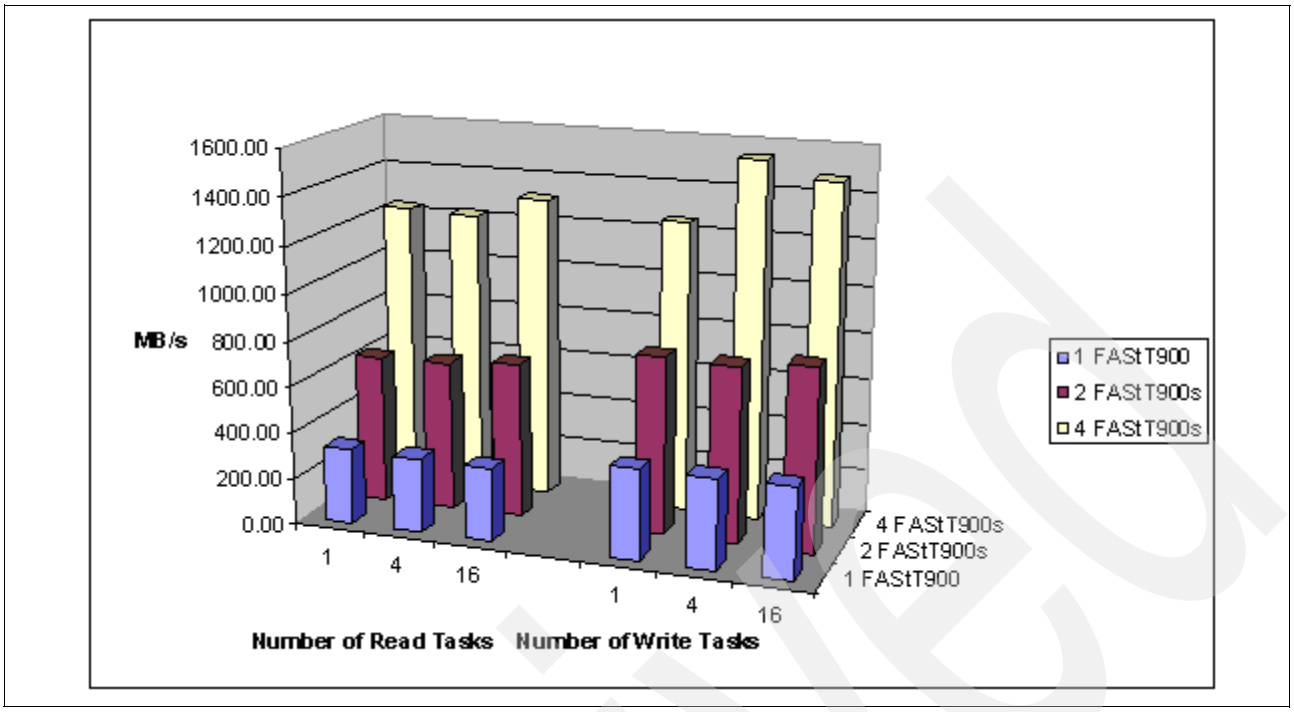

Figure 10 DS4500 scaling under GPFS using the 1-, 4-,and 16-task write and read jobs executed on the p690.

iperf test - x335 GPFS write tests did not meet expectations

The large record, sequential x335 read tests sustained peak performance rates < 692 MB/s, while the x335 write tests sustained peak performance rates < 282 MB/s. As previously explained, the read tests met expectations, but the write tests fell below expectations (even though it is good enough for many applications).

At issue is the gating factor imposed by an EtherChannel over eight ports; why is the write rate less than half the read rate over the same EtherChannel? In order to eliminate variables, the network analyzing tool iperf[14] was used to determine whether this disparity was a result of the EtherChannel/IP Stack or GPFS. The idea behind the test was to see if the iperf write rate is less than half the iperf read rate; if it was, then the conclusion would be that the disparity in read/write performance for GPFS could be attributed to the EtherChannel/IP Stack. Unfortunately, that was not the case.

This was the test procedure. Write tests using iperf were done on 1, 8 and 32 client nodes (the x335 nodes) to the p690 using the same GbE network that GPFS is using. The iperf write rate for one client is 115 MB/s (which is consistent with GPFS for 1 x335 node). However, for eight clients, the iperf write rate was approximately 900 MB/s while for GPFS, the write rate for eight clients < 282 MB/s. The read rates for iperf were comparable to the write rates. This *suggests* that the disparity in performance is attributable to GPFS. Further research is needed.

GPFS in a mixed AIX/Linux environment is stable and easy to use and maintain

While it lies outside the scope of this paper, another inference to be drawn from these tests is that using GPFS in a mixed AIX/Linux environment is not particularly difficult or complicated. From a user/programmer standpoint, no intrinsic limitations are imposed. In the test

[14] Visit http://www.noc.ucf.edu/Tools/Iperf/default.htm for details.

configuration, there was only one file system and it was globally accessible to all nodes. There was no need to partition the files in the file system between architectures, or even place them in separate directories.

From a system administration perspective, there were no complicated procedures required for creating or mounting the file system and so on. GPFS appeared as a seamless integrated file system, and it was agnostic to the operating system through which applications accessed it or commands maintained it. It should also be observed that no file system “crashes”, “job hangs” or other abnormal terminations or behaviors occurred in this environment that were attributable to GPFS during these tests; it was very stable and performed well.

Summary

This benchmark study was constructed with two goals: to assess GPFS performance in a mixed AIX/Linux environment, and to assess GPFS scaling regarding increased numbers of DS4500s.

The results of this study established the following conclusions:

1. Jobs running only on the AIX/p690 system that accessed the GPFS file system in this mixed AIX/Linux environment met or exceeded performance expectations.
2. Read jobs running only on the Linux/x335 system that accessed the GPFS file system in this mixed AIX/Linux environment met performance expectations. Write jobs, however, accessed the GPFS file system at 40% of the read rate. While this is good enough for many applications, it requires further investigation.
3. Jobs running simultaneously on the AIX/p690 and Linux/x335 systems that accessed the GPFS file system in this mixed AIX/Linux environment met or exceeded performance expectations. Note that these tests were done using two multi-task jobs, where each job ran exclusively on one system or the other and accessed its own file.[15]
4. GPFS performance scaled linearly on the p690 with increasing numbers of DS4500 storage controllers; the number of LUNs per DS4500 was constant.
5. GPFS runs stably in a mixed Linux/AIX environment. This mixed environment does not impose restrictions upon normal GPFS operations, nor does it require complex procedures to maintain and configure GPFS.

Using GPFS in a mixed AIX/Linux environment is a useful configuration alternative for processing centers running both AIX- and Linux-based systems. Properly configured, GPFS in this mixed environment can be expected to perform well.

Additional material

The author has made the code for this benchmark, ibm.v3g (I/O Benchmark, version v3g) available on an as-is basis. It can be downloaded from the IBM Redbooks™ Web server at:

`ftp://www.redbooks.ibm.com/redbooks/REDP3962`

At the Web site, click **Additional material** on the right side of the window. Right-click the .zip file (REDP3962.ZIP). Select the **Save file as** option to download the file. Unzip into a directory of your choice.

[15] This is not due to a GPFS restriction; rather it is an artifact attributable to the fact that PVM was not configured to support parallel jobs spanning both systems.

Appendix A: The RAID5 write penalty

There is a close relationship between the GPFS block size, the DS4500 segment size, and another parameter called the Logical Track Group (LTG) for a RAID5 configuration. The LTG is the maximum amount of data that can be transferred in one I/O request to the disk(s) in an LUN[16]. The key to understanding this relationship is the RAID5 write penalty.

Suppose you have 4+P RAID5 array. Logically, you can think of it as four data disks plus one disk containing error recovery information. When writing to a RAID5 array, regardless of the data unit size, a “stripe” of data is written across all four of the data disks at the same time.

Suppose the size of this stripe is 1 MB. Then it will consist of four segments, each 256 KB in size. Suppose a 600 KB data unit is to be written to the RAID5 array. Since its size is less than 1 MB, the entire stripe must first be read, the portion of the stripe corresponding to the 600 KB unit updated, and the whole stripe written to disk. Hence, this write operation requires a read/update/write action. By contrast, suppose the data unit is exactly 1 MB; then it is not necessary to first read and update the stripe. The controller can simply overwrite the stripe, thus saving a significant amount of time.

Consider a RAID5 configuration with N+P parity (N data disks plus 1 parity disk). Then for optimum performance, the GPFS block size should be Q * N * segment size (Q and N are integers), and at the same time the LTG should be J * N * segment size (J is an integer).

For example, suppose the parity is 4+P and the LTG is 1 MB; then N is already determined to be 4 and the values Q = 1, J = 1, segment size = 256 KB and GPFS block size = 1 MB yield optimum performance. Or, for another example, suppose the parity is 4+P and LTG is 128 KB; then N is already determined to be 4 and the values Q = 8, J = 1, segment size = 32 KB and GPFS block size = 1 MB yield optimum performance.

[16] In AIX, the size of LTG for a volume group can be seen using the `lsvg` command. It is also the same as the max transfer size in an hdisk and can be viewed using the `lsattr` command.

Appendix B - Summary of benchmark tests

Test #1	**p690**	**1 T900**	**write caching = off, pattern = seq, bsz = 1 MB**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	325.05	388.97					4
	1	2	336.30	397.73	325.74	388.19	162.87	194.10	8
	1	4	330.86	402.15	319.38	387.25	79.84	96.81	16
	1	8	328.93	417.80	316.73	385.73	39.59	48.22	32
	1	16	328.04	392.83	316.22	390.09	19.76	24.38	64
	1	32					0.00	0.00	
Test #2	**x335**	**1 T900**	**write caching = off, pattern = seq, bsz = 1 MB**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	see test_5 *						
	2	1	180.27	219.73	181.18	219.81	90.59	109.91	8
	4	1	256.28	388.48	257.37	389.79	64.34	97.45	16
	8	1	260.75	390.53	260.93	392.65	32.62	49.08	32
	16	1	266.58	390.52	267.51	392.00	16.72	24.50	64
	32	1					0.00	0.00	
note *	since x335 node is "gated" by GbE adapter, rates will ~= single task job in test_5								
Test #3	**p690**	**1 T900**	**write caching = off, pattern = strd, bsz = 16k**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	15.64	25.66					1
	1	2	9.25	36.92	9.25	37.15	4.63	18.58	2
	1	4	10.12	13.49	10.23	14.26	2.56	3.57	4
	1	8	10.06	16.82	10.21	17.48	1.28	2.19	8
	1	16					0.00	0.00	
	1	32					0.00	0.00	
Test #4	**x335**	**1 T900**	**write caching = off, pattern = strd, bsz = 16k**						
I skipped this test: I simply ran out of time and the test was not that interesting.									
Test #5		**1 T900**	**write caching = off, pattern = seq, bsz = 1 MB**						
			Natural Aggregate		Harmonic Aggregate		Harmonic aggregate over x335 and p690		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
x335 only	4	1	253.85	389.03	254.11	390.08	N/A	N/A	16
p690 only	1	4	313.95	381.33	324.05	402.66	N/A	N/A	16
x335 with p690	4	1	167.23	267.50	167.23	268.30	369.01*	341.64*	28**
p690 with x335	1	4	196.76	199.97	195.32	200.35			28**
note *	job times are not identical; iostat showed aggregate rate while both jobs were running ~= 400 MB/s								
note **	Size of combined files from each job; file sizes adjusted so job times were approximately equal.								

Figure 11 Single DS4500 tests - tests using only one DS4500, write caching DISabled.

Test #6	**p690**	**2 T900s**	**write caching = off, pattern = seq, bsz = 1 MB**							
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean			
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB	
	1	1	640.10	765.28					4	
	1	2	641.07	760.75	648.02	761.71	324.01	380.85	8	
	1	4	631.84	739.31	646.62	758.13	161.65	189.53	16	
	1	8	649.14	724.88	670.83	755.63	83.85	94.45	32	
	1	16	646.97	721.26	671.29	788.48	41.96	49.28	64	
	1	32					0.00	0.00		
Test #7	**x335**	**2 T900s**	**write caching = off, pattern = seq, bsz = 1 MB**							
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean			
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB	
	1	1	109.27	110.89					4	
	2	1	198.92	218.65	198.92	219.51	99.46	109.76	8	
	4	1	269.91	435.37	269.95	437.68	67.49	109.42	16	
	8	1	282.44	624.19	283.44	626.81	35.43	78.35	32	
	16	1	253.23	595.50	281.78	598.18	17.61	37.39	64	
	32	1	269.77	577.53	269.83	581.23	8.43	18.16	128	
Test #8	**p690**	**2 T900s**	**write caching = off, pattern = strd, bsz = 16k**							
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean			
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB	
	1	1	25.44	35.79					2	
	1	2	18.14	54.53	18.16	55.00	9.08	27.50	4	
	1	4	17.34	23.22	17.44	23.87	4.36	5.97	8	
	1	8	18.88	26.80	19.18	28.01	2.40	3.50	16	
	1	16	20.48	34.95	20.71	36.06	1.29	2.25	32	
	1	32					0.00	0.00		
Test #9	**x335**	**2 T900s**	**write caching = off, pattern = strd, bsz = 16k**							
I did not do this test since there was little to gain from such a test in light of the one I did with write caching Enabled. See test_9.										
Test #10		**2 T900s**	**write caching = off, pattern = seq, bsz = 1 MB**							
			Natural Aggregate		Harmonic Aggregate		Harmonic aggregate			
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB	
x335 only ***	4	1	267.33	435.72	268.29	437.23	N/A	N/A	16	
p690 only	1	4	632.45	771.00	650.98	788.25	N/A	N/A	16	
x335 with p690	4	1	233.03	386.27	233.03	389.61	707.48*	801.66*	24**	
p690 with x335	1	4	470.22	407.79	477.30	412.81			32**	
note *	Job times are nerely identical; therefore, iostat measured rate was very close to harmonic aggregate rate.									
note **	Size of combined files from each job; file sizes adjusted so job times were approximately equal.									
	Combined files for write = 24 GB, combined files for the read = 32 GB.									
note ***	x335 aggregate read rates were gated by the 4 GbE at a little over 100 MB/s per adapter.									
	Should do 8 task job. Compare with test_C.									

Figure 12 Two DS4500 tests - tests using two DS4500s, write caching DISabled.

Test #11	**p690**	**2 T900s**	**write caching = ON, pattern = seq, bsz = 1 MB**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	768.21	620.57					4
	1	2	774.97	771.37	775.80	773.59	387.90	386.80	8
	1	4	767.94	735.78	784.99	751.72	196.25	187.93	16
	1	8	776.23	720.27	810.23	751.22	101.28	93.90	32
	1	16	721.80	724.33	755.05	781.48	47.19	48.84	64
	1	32					0.00	0.00	
Test #12	**x335**	**2 T900s**	**write caching = ON, pattern = seq, bsz = 1 MB**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	110.19	110.88					4
	2	1	189.97	219.18	189.97	219.76	94.99	109.88	8
	4	1	271.39	433.87	272.78	435.99	68.19	109.00	16
	8	1	270.35	648.30	301.89	654.39	37.74	81.80	32
	16	1	286.21	645.56	287.15	666.74	17.95	41.67	64
	32	1	280.56	580.72	282.29	587.63	8.82	18.36	128
Test #13	**p690**	**2 T900s**	**write caching = ON, pattern = strd, bsz = 16k**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1							
	1	2	11.63	49.86	11.63	49.87	5.82	24.93	0.5
	1	4	15.11	34.48	15.12	35.36	3.78	8.84	1.0
	1	8	23.32	55.96	23.96	59.41	2.99	7.43	2.0
	1	16					0.00	0.00	
	1	32					0.00	0.00	
Test #14	**x335**	**2 T900s**	**write caching = ON, pattern = strd, bsz = 16k**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	1.77	62.44					0.25
	2	1	5.80	8.85	6.17	15.09	3.08	7.55	0.5
	4	1	7.58	14.87	7.77	24.88	1.94	6.22	1.0
	8	1	7.67	79.85	7.91	86.53	0.99	10.82	2.0
	16	1					0.00	0.00	
	32	1					0.00	0.00	
Test #15		**2 T900s**	**write caching = off, pattern = seq, bsz = 1 MB**						
I did not do this test… not enough time and not a recommended config (i.e., write caching ENabled not recommend).									

Figure 13 Two DS4500 tests - tests using two DS4500s, write caching ENabled.

Test #16	**p690**	**4 T900s**	**write caching = off, pattern = seq, bsz = 1 MB**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	1233.84	1259.48					4
	1	2	1205.22	1507.61	1207.57	1515.52	603.79	757.76	8
	1	4	1220.48	1541.19	1221.24	1543.55	305.31	385.89	16
	1	8	1230.79	1435.37	1255.06	1478.34	156.88	184.79	32
	1	16	1261.76	1388.82	1314.27	1472.10	82.14	92.01	64
	1	32					0.00	0.00	
Test #17	**x335**	**4 T900s**	**write caching = off, pattern = seq, bsz = 1 MB**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	see test_5 *						
	2	1	177.47	215.71	177.61	217.44	88.80	108.72	8
	4	1	207.42	427.55	236.79	432.29	59.20	108.07	16
	8	1	244.68	692.17	245.77	698.91	30.72	87.36	32
	16	1	249.66	574.91	250.03	610.65	15.63	38.17	64
	32	1					0.00	0.00	
note *	since x335 node is "gated" by GbE adapter, rates will ~= single task job in test_5								
Test #18	**p690**	**4 T900s**	**write caching = off, pattern = strd, bsz = 16k**						
			Natural Aggregate		Harmonic Aggregate		Harmonic Mean		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
	1	1	35.90	48.88					1
	1	2	30.12	109.31	30.17	109.80	15.08	54.90	2
	1	4	33.34	78.74	33.50	79.44	8.38	19.86	4
	1	8	35.19	61.40	35.43	64.04	4.43	8.01	8
	1	16					0.00	0.00	
	1	32					0.00	0.00	
Test #19	**x335**	**4 T900s**	**write caching = off, pattern = strd, bsz = 16k**						
I skipped this test: I simply ran out of time and the test was not that interesting.									
Test #20		**4 T900s**	**write caching = off, pattern = seq, bsz = 1 MB**						
			Natural Aggregate		Harmonic Aggregate		Harmonic aggregate		
	Number of Client Nodes	Number of tasks per client node	Write Rate MB/s	Read Rate MB/s	Write Rate MB/s	Read Rate MB/s	write	read	sizeof(file) GB
x335 only	8	1	224.45	625.55	253.99	683.67	N/A	N/A	32
p690 only	1	8	1219.18	1358.67	1253.33	1404.92	N/A	N/A	32
x335 with p690	8	1	223.25	590.99	226.07	646.43	1231.42	1702.23	40*
p690 with x335	1	8	1067.09	991.81	1081.95	1011.34			48*
note *	Size of combined files from each job; file sizes adjusted so job times were approximately equal.								
	Combined files for write = 40 GB, combined files for the read = 48 GB.								

Figure 14 Quad DS4500 tests - tests using four DS4500s, write caching DISabled.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper:

- IBM Redpaper *Benchmark Using x345/GPFS/Linux Clients and x345/Linux/NSD File Servers*, REDP-3909-00, Paden, R.L. [Paden 04a], available at:

  http://w3.itso.ibm.com/

- "GPFS: Programming, Configuration and Performance Perspectives", tutorial presented at ScicomP 10, Austin, TX, July 04, Paden, R.L. [Paden04b]
- *GPFS for Clusters V2.2 Administration and Programming Reference*, SA22-7967-01, [IBM04a]
- *GPFS for Clusters V2.2 Concepts, Planning, and Installation Guide*, GA22-7968-01, [IBM04b]
- Pfister, G.F. *In Search of Clusters, 2nd ed*. Prentice Hall, 1998, ISBN 0138997098 [Pfister98]

The cited GPFS documentation is available at:

http://publib.boulder.ibm.com/clresctr/windows/public/gpfsbooks.html#aix_rsctpd22wo

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	IBM®	Redbooks (logo)™
@server®	POWER™	Redbooks (logo) ™
AIX®	POWER4™	
AIX/L®	Redbooks™	

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.