Redbooks Paper

Raymond L. Paden

I/O Benchmark Using x345/GPFS/Linux Clients and x345/Linux/NSD File Servers

A series of four benchmark tests were run at the IBM® Benchmarking Center in Poughkeepsie, NY. The primary focus of these tests was to evaluate GPFS performance for HPC applications on Linux® clusters. The clusters in this test were IBM @server® xSeries® 345-based systems using IBM TotalStorage® FAStT900 disk controllers and EXP700 disk drawers.

Each test was conducted on a different cluster network configuration (Gigabit Ethernet (GbE), bonded GbE, 100 Megabit Ethernet (MbE), and Myrinet). Some of these test configurations are officially supported, and others are not.

The author

Raymond L. Paden, PhD, is an HPC Technical Architect working in Deep Computing for the IBM Systems Group. Prior to joining IBM, he worked for six years as a team manager and systems programmer developing seismic processing applications and 13 years as a professor of Computer Science. He holds a PhD in Computer Science from the Illinois Institute of Technology. His areas of technical expertise include disk and tape I/O, performance optimization, and operating systems on parallel systems, such as IBM @server pSeries® and xSeries clusters. He has written on topics including parallel and combinatorial optimization and I/O. He co-authored the IBM Redbook *GPFS on AIX Clusters: High Performance File System Administration Simplified*, SG24-6035.

Technical contributors to this benchmark

Lerone Latouche
David Watson
IBM Poughkeepsie

Executive overview

The primary focus of these four tests was to evaluate General Parallel File System (GPFS) performance for HPC applications on Linux clusters. The clusters in this test were IBM @server xSeries 345-based (x345) systems using IBM TotalStorage FAStT900 disk controllers and EXP700 disk drawers. Each test was conducted on a different cluster network configuration (Gigabit Ethernet (GbE), bonded GbE, 100 Megabit Ethernet (MbE), and Myrinet). Some of these test configurations are officially supported, and others are not.

In this paper, we summarize and interpret the results of these benchmark tests. We also propose untested alternative designs that were suggested by these results.

Overall, the results of the tests were positive, most meeting or exceeding expectations. A few others, although below expectations, were acceptable. Only one test, in an unsupported, infrequently-used configuration, yielded unacceptable results.

Equally important to the performance results is that there were no system failures attributable to GPFS or the GPFS/Linux combination. We did experience several node failures on one of the storage server nodes, but the *apparent* cause was an intermittent adapter failure under heavy loads. This presented no significant setbacks to the benchmark tests and had the added value of validating the failover features of the storage subsystem.

System configuration

This benchmark consisted of four tests. Each was associated with a unique cluster network configuration. The node and storage subsystem configuration remained constant across all completed tests.

Components defined

The *General Parallel File System (GPFS)* for Linux is a high-performance shared-disk file system that can provide data access from all nodes in a Linux cluster environment. Parallel and serial applications can readily access shared files using standard UNIX® file system interfaces, and the same file can be accessed concurrently from multiple nodes. GPFS provides high availability through logging and replication and can be configured for failover from both disk and server malfunctions

High Performance Computing (HPC) is a cluster configuration designed to provide greater computational power than one computer alone could provide. HPC clusters connect the computing powers of the nodes involved to provide higher scalability and more computing power.

IBM @server xSeries 345 is an expandable, high-availability, high-performance server for e-business applications and mail and collaborative applications in a rack-mounted package. The x345 2-way server delivers computing power with high availability and expansion capability. The 2U x345 has the latest Intel® Xeon processors, DDR memory, and Dual Ultra320 SCSI controllers to deliver high performance.

IBM TotalStorage FAStT900 Storage Server delivers breakthrough disk performance and outstanding reliability for demanding applications in compute intensive environments. The FAStT900 offers investment protection with advanced functions and flexible features, including up to 32 TB of Fibre Channel (FC) disk storage capacity with the EXP700. It also offers advanced replication services to support business continuance and disaster recovery.

IBM TotalStorage FAStT EXP700 Storage Expansion Unit is a 14-bay, rack-mountable Fibre Channel disk drive enclosure, designed for use with many of the FAStT Storage Servers. It provides a full end-to-end 2 Gbps FC highly scalable, high-performance storage solution. Used with the FAStT900 Storage Server[1] in particular, this option provides incremental support for more than 1 TB of disk storage per unit. It supports four new, slim-profile, dual-loop, hot-swappable disk drive modules.

Node, disk, and file system configurations

The test cluster consisted of 32 nodes as follows:

- xSeries 345, dual CPU, 2.8 GHz, 4 GB RAM
- Red Hat Linux 8.3, Kernel 2.4.20-19.8 pok
- GPFS 1.3
- PVM 3.4

The nodes were divided into two sets; there were 28 client nodes and four storage server[2] nodes. All 32 nodes were part of the same GPFS node set.

The storage subsystem consisted of four server nodes (x345), two disk controllers (FAStT900), 12 disk drawers (EXP700, six per controller), 16 FC adapters (two per server, four per controller, all at 2 Gbps) in a Network Shared Disk (NSD) configuration. See the test configuration diagrams throughout the paper for illustrations.

The disks attached to each FAStT900 were configured as 16 LUNs (4+P RAID groups). The FAStT900 parameters were set as follows.

- Read-ahead multiplier = 0
- Write caching = off
- Read caching = off
- Segment size = 32 KB
- Cache block size = 16 KB

The GPFS settings were left at their default settings for the most part. The only exceptions were the page pool size, which was set to 128 MB, and the block size, which was set to 256 KB.

Cluster network configurations for each test

Each test had a different test cluster network configuration, each offering alternative price and performance profiles. GPFS, in an NSD configuration, makes extensive use of the cluster network for passing file data/transactions between the client and server nodes.

Test #1 - GPFS/NSD over 100 MbE for client nodes

In this test, each client node was connected to the cluster network using 100 MbE (Megabit Ethernet), and each server node was connected using 1 GbE (Gigabit Ethernet or 1000 MbE). This is not an officially supported configuration by GPFS, but it configured quite naturally, and all tests using it completed normally. See Figure 1 on page 7.

[1] http://www.storage.ibm.com/disk/fastt/fast900/

[2] In this document, the term “server node” refers to a “storage server node” unless stated otherwise.

Test #2 - GPFS/NSD over 1 GbE on servers and clients

There were two related but distinct configurations evaluated in this test.

In the first test (#2A), each of the client nodes and server nodes was connected to the cluster network with a single GbE adapter. This configuration is officially supported and commonly used by GPFS.

In the second test (#2B), each of the client nodes was connected to the cluster network with a single GbE adapter. However, each of the server nodes was connected to the cluster network using two bonded GbE adapters forming a single channel (IP address). Although it worked, and all tests performed on it completed normally, this configuration is not officially supported for use by GPFS at this time. The reason for considering it is that it can compensate for the communication load imbalance presented by having a large number of client nodes sharing a small number of server nodes. See Figures 2 and 3 beginning on page 8.

Test #3 - GPFS/NSD over Myrinet on servers and clients

In this test (#3), each client and server node was connected to the cluster network using Myrinet adapters and the Myrinet switch. This configuration is officially supported and commonly used by GPFS. A Linux cluster with the Myrinet switch is architecturally similar to an SP system. See Figure 4 on page 12.

Test #4 - Miscellaneous tests

There were two miscellaneous tests conducted, more as a proof of concept than a rigorous benchmarking test, to examine using the Linux storage server nodes for multiple purposes. The Myrinet-based configuration was used for these tests, but extended as necessary.

In the first test (#4A), client tasks were executed both on the server nodes and the client nodes at the same time. This is very common on SP systems. The system configuration in this case was identical to Test #3. See Figure 4 on page 12.

In the second test (#4B), the NSD server nodes for GPFS were also used as NFS server nodes, although it was the Linux ext3 file system that was exported, not GPFS.[3] The NFS clients used the 100 MbE network to communicate with the server nodes, and GPFS used the Myrinet network. A client node with an NFS mount did not have access to the GPFS file system and vice versa This was done for reasons of convenience, not necessity. See Figure 5 on page 14.

Benchmark code description

The benchmark codes used in this study are custom codes that we designed and programmed to evaluate common storage I/O paradigms used in seismic processing and most HPC customers. These codes have been used many times to evaluate I/O designs for various customers. 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.v3e (I/O Benchmark, version v3e). Contact the author for further details.

Analyzing the results

We explain the measurements and access patterns that we used, and then we examine the test results.

[3] For reasons not entirely clear, we were never able to NFS mount the GPFS file system to the client nodes. The TCP mount option simply failed, so we were forced to use UDP. The job would then fail upon trying to access the file; a write() call failed with EEXIST and a read() call failed with ENOENT.

Metrics used to assess results

Bandwidth in units of megabytes per second (MB/s) is the fundamental measure of performance in these tests. To measure bandwidth, one must therefore measure data volume and time.

For all of the reported tests, each task in a job accessed at least 1 GB of data with the total data for the job being the number of tasks times the data per task. For example, a four-task job accesses 4 GB of data.

Time is assessed using wall clock time starting at a beginning and terminating at an end. Thus no distinction is made between user, system, and real time. Because the test jobs are parallel, time is collected on a per task basis, and no task starts its timer until all tasks have been 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 is stopped after the last record has been accessed and the file has been closed. Because of the stochastic (involving chance or probability) nature for the various queuing systems associated with parallel tasks, under normal circumstances, tasks can and do terminate with some variance,[4] although on the whole they remain uniformly active.

Three different measures of bandwidth are 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. (Note: It is very consistent with performance monitors such as 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 perspective. From a system perspective, this rate is often lower than the actual system rate because 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 are 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.
Semi-random	The next four records are randomly accessed within a working set of 32 records.
Random	Each record within the file is accessed once and only once using a uniform random distribution.
Hint	Use the GPFS Multiple Access Hint mechanism to pre-fetch records.

[4] The variances were not that extreme in this study.

Access patterns have a significant impact on 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 are tested.

At the other extreme in terms of performance is small record irregular patterns (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 that 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

This section summarizes key results and their analysis from each of the various tests. Complete results for each experiment are listed in a spreadsheet. See “Associated documents” on page 18.

Test #1 - NSD over 100 MbE for client nodes

100 MbE is an older technology, but provides “good enough” performance in a cost effective manner for many HPC applications with low storage I/O performance requirements. For example, it is commonly used to provide file access through NFS in very large Linux clusters. Because of its popularity, GPFS was evaluated using this technology, even though it is not officially supported. See Figure 1.

Table 1 lists the results for a large record sequential access pattern test. For a single 100 MbE adapter, 7 MB/s of effective throughput is a reasonable target. Looking at the harmonic mean in the read test, you can easily see that this was achieved and that aggregate performance scales linearly. Because the servers’ access to the cluster network was over GbE, they were not a bottleneck. Write performance, however, was much lower for reasons that are not clear. It is felt that a configuration anomaly or some other artificial problem is the cause. Further investigation is needed.

Table 1 Large record sequential using 100 MbE for the client nodes

		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**
1	1	0.141	7.87				
2	1	0.209	15.44	0.209	15.49	0.105	7.74
4	1	0.296	30.59	0.307	30.74	0.077	7.68
8	1	0.357	56.27	0.393	60.22	0.049	7.53
16	1	0.429	111.89	0.498	116.41	0.031	7.28

Because of the intrinsically low access rate for 100 MbE, time only allowed one more test, a small record sequential access test. The data rate per task dropped down to a little over 4 MB/s for reads; writes were not tested. See the spreadsheet listed in “Associated documents” on page 18 for further details.

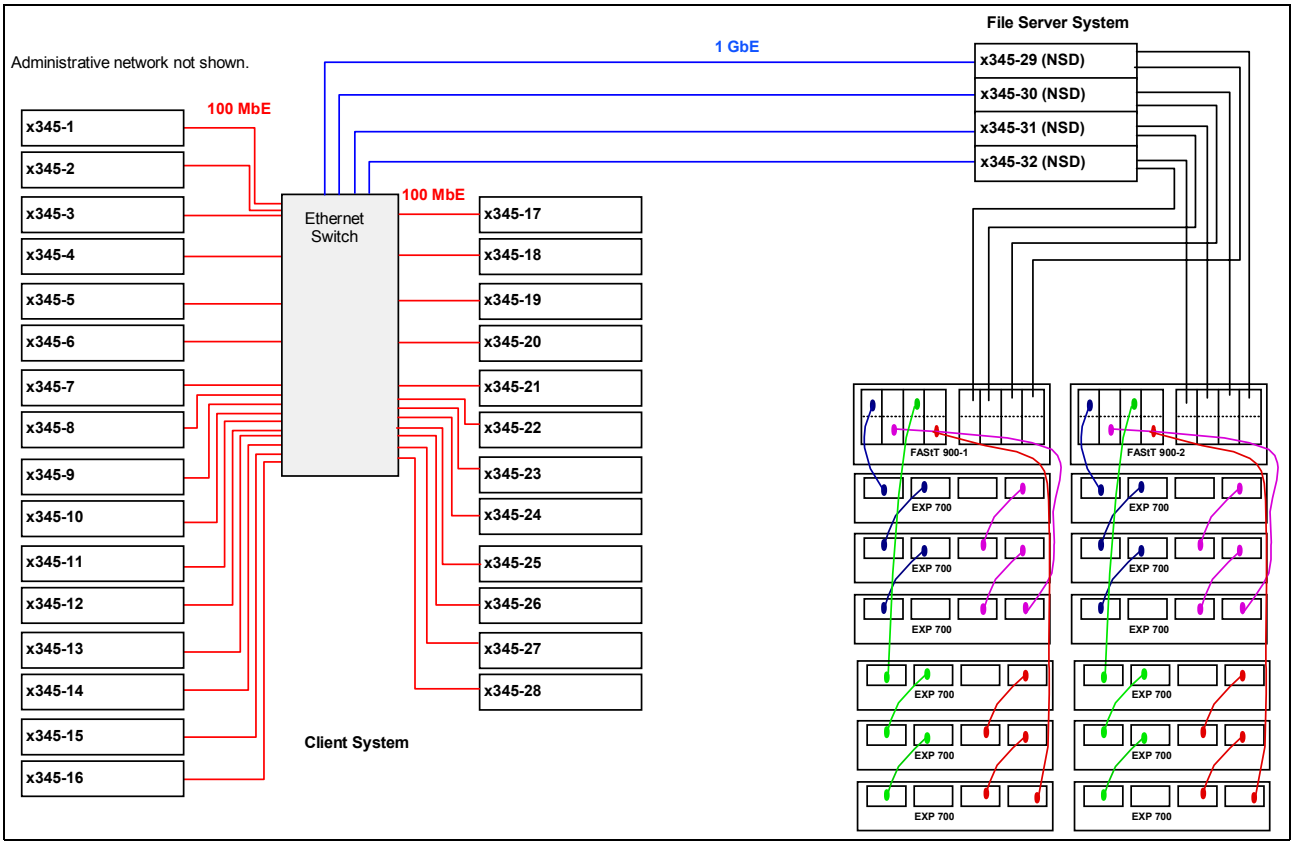

Figure 1 Test #1 configuration

Test#2 - GPFS/NSD over GbE on servers and clients

Two variations of this test were performed. The first, Test #2A, was the very standard configuration where all nodes, both server and client, provided a single GbE adapter for GPFS traffic. See Figure 2. In the second, Test #2B, the server nodes had dual bonded GbE adapters in order to compensate for the large number of clients per server; this configuration is not yet officially supported in GPFS.

First consider a large record sequential test for the single GbE adapter per server configuration. It is used to establish baseline peak performance. See Table 2. A reasonable upper limit estimate of GbE adapter performance is 75 MB/s. As the single and dual client tests show, this was significantly surpassed. However, as the task counts increased, performance leveled off somewhere around 350 MB/s, and the harmonic mean steadily dropped rather than remaining constant.

Part of the problem is that the servers are gating the performance. There are only four servers, each with a single GbE adapter, but there are 28 clients in the configuration, which in aggregate are capable of yielding 7X the bandwidth that the servers can provide. Using the 75 MB/s target, the aggregate performance exceeds expectation for what this configuration can provide given that there are only four servers. However, using a 100 MB/s target, suggested as feasible by the single and dual task jobs, one would expect an aggregate closer to 400 MB/s. A possible reason that we are below this is attributable to network congestion

being sorted out by the various software layers (such as inevitable parallel overhead). For example, there are 16 clients dumping packets to only four servers. In any case, scaling appears to be linear in the number of servers, being O(n). See “Linear scaling in the number of storage servers” on page 14.

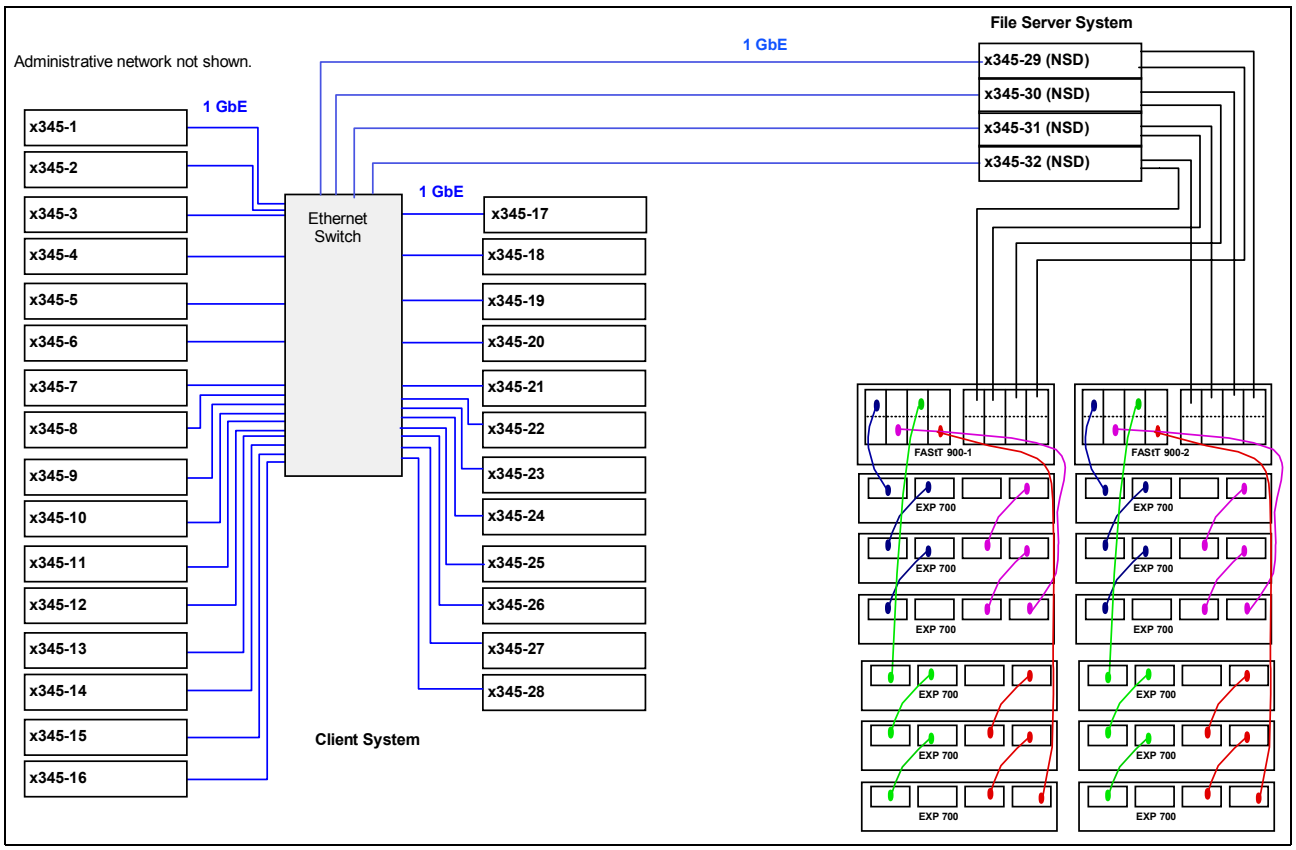

Figure 2 Test #2A configuration

Table 2 Large record sequential in the single GbE adapter/server configuration

		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**
1	1	92.26	111.27				
2	1	178.02	209.94	178.03	220.60	89.02	110.30
4	1	292.54	320.46	296.06	337.92	74.02	84.48
8	1	383.69	338.26	383.86	360.21	47.98	45.03
16	1	315.12	347.32	335.75	357.07	20.98	22.32

In order to diminish the effect of the single GbE adapter/server gating factor, we tried another set of tests using a set of dual bonded GbE adapters per server; see Figure 3. Table 3 summarizes the large record test. As a target, one would expect to double the bandwidth per server node, yet this is not the case. Performance is roughly the same for the writes and slightly improved for the reads. A plausible explanation for this is that these x345 systems have only two CPUs, so there is insufficient horsepower to keep up with the load. If this is true, perhaps a four-CPU x360 could improve the performance.

Tip: A guideline for older pSeries systems such as the p660-6m1 is to allow 1 CPU for every GbE adapter. Testing is needed to see if this applies to xSeries systems also.

A good validation of this assertion could be based on CPU utilization. Unfortunately, the CPU utilization record was unavailable for this test. However, the CPU utilization for the single GbE adapter tests was as high as 86% for peak bandwidth jobs. It is, therefore, quite plausible that having dual GbE adapters per server, albeit bonded into a single channel, required more than the additional 14% CPU bandwidth to fully use their bandwidth. Further testing is needed to fully validate this point.

Table 3 Large record sequential in dual bonded GbE adapter/server configuration

		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**
1	1	109.67	110.29				
2	1	212.36	214.51	212.36	215.50	89.02	110.30
4	1	344.26	398.28	365.75	398.94	74.02	84.48
8	1	326.76	414.49	342.01	426.82	47.98	45.03
16	1	308.77	388.94	312.33	424.90	20.98	22.32
28	1	192.20	206.89	235.88	207.65	8.42	7.42

Under Test #2A (single GbE adapter per server), a number of other access patterns were also tested and performed according to expectation. For example, the small record random pattern yielded the worst performance; the 16 KB record size only used 6% of the GPFS block and the random pattern thwarted cache optimization algorithms (see Table 4). In spite of low absolute performance, reads have the useful property of scaling linearly in the number of clients instead of leveling off at four tasks, as in the sequential pattern case being gated by the servers. This yields a relatively acceptable aggregate performance across a larger number of client nodes. This can be seen very clearly by looking at the harmonic mean. However, due to the RAID penalty, small record writes performed significantly worse than reads did, and they did not scale linearly. By using the Multiple Access Hint feature in GPFS, the absolute performance can be increased significantly for random reads (for example, 5X for eight client tasks; see the spreadsheet listed in “Associated documents” on page 18 for details).

Note: Further testing is needed to determine to how many clients this linear scaling will extend. Because GPFS clients are sending large, though partially used, blocks, the server must process entire blocks. However, if the records are small enough (that is, if a record is 1/32 of the block size), only sub-blocks are sent. This could increase the degree of linear scaling. In this test, the record size is 16 KB and block size is 256 KB, thus the records exceeded the sub-block threshold.

Table 4 Small record random in a single GbE adapter/server configuration

		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**
1	1	2.86	2.09				
2	1	1.49	4.16	1.58	4.16	0.79	2.08
4	1	2.09	8.37	2.35	8.43	0.59	2.11
8	1	1.80	16.78	2.03	16.95	0.25	2.12
16	1	1.98	32.42	2.12	32.87	0.13	2.05

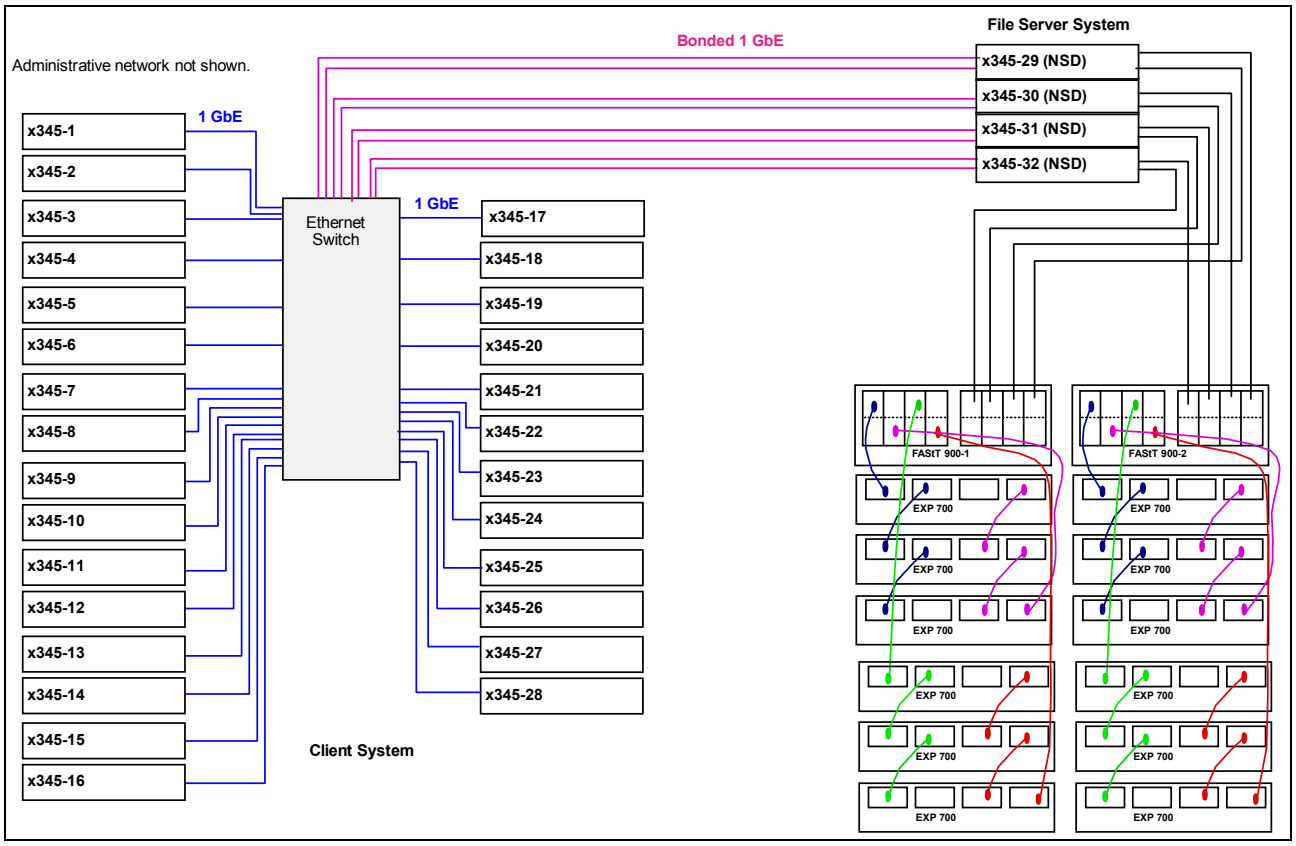

Figure 3 Test #2B configuration

The small record strided access pattern yielded performance rates in between the two extremes of the small record random and large record sequential patterns (see Table 5). Its performance is degraded because it is only using 6% of the block for each record, but it gets a performance uplift from optimization algorithms based on the fact that GPFS recognizes the pattern; for example, it can pre-fetch records in a read job. Due to the RAID write penalty for small records, the write performance is not as good as the read performance.

Table 5 Small record strided in a single GbE adapter/server configuration

		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**
1	1	8.46	24.73				
2	1	8.83	30.72	8.84	44.89	4.42	22.44
4	1	13.22	51.02	13.25	79.13	3.31	19.78
8	1	18.14	86.32	18.44	117.54	2.31	14.69
16	1	22.08	76.30	22.67	91.95	1.42	5.75

Test#3 - GPFS/NSD over Myrinet on servers and clients

The Myrinet switch is a very standard Linux xSeries cluster network configuration for GPFS. It is architecturally similar to SP systems, although it uses only the IP stack for communication. This test configuration used the first generation Myrinet technology. Therefore, single adapter target bandwidth is 125 to 150 MB/s, although testing suggests that this might be slightly pessimistic. Compared to GbE, Myrinet delivers higher performance. However, the general shape of the performance profiles is very similar; large record sequential is faster than small record strided is faster than small record random. Although the absolute price for Myrinet is higher, fewer servers may be needed from a storage standpoint, especially for the faster second generation Myrinet technology, thus improving its price/performance characteristics.

Note: This assertion relative to the second generation Myrinet technology needs further testing to see if the server CPUs can handle the load. It might be necessary to use a four-CPU system like an x360 optimized for file service, rather than a dual CPU x345, which will work well for a single GbE adapter.

Consider the large record sequential access pattern (see Table 6 and Figure 4). As in the GbE case, the four-server configuration was used, and each server and each client had a single Myrinet adapter. And as with the GbE adapter case, performance was gated by the limited number of servers, each with one Myrinet adapter. However, what is different is that the write performance is significantly less than the read performance, although in absolute terms, it is quite acceptable. For example, the relative error between the read and write performance in the Myrinet case is roughly 30% for 8 or 16 client tasks, although it is only 6% for the GbE case.

Table 6 Large record sequential in Myrinet adapter configuration

		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**
1	1	159.75	184.63				
2	1	273.85	324.94	288.82	341.13	144.41	170.56
4	1	320.76	394.52	420.34	483.16	105.09	120.79
8	1	382.95	536.70	385.03	559.73	48.13	69.97
16	1	388.43	516.92	396.57	584.77	24.79	36.55
28	1	380.49	560.52	385.31	563.83	13.76	20.14

The performance profiles of the other Myrinet test cases follow expectation in a similar manner to the GbE test cases. Small record strided access is less than large record sequential, but it is better than the GbE case. Random and semi-random patterns using hints is slower than small record strided, but again, it is better than the GbE case. The one slight surprise is the small record random with no hints case. Although it is the slowest access pattern for Myrinet, its rate relative to GbE is nearly the same. Because of these similarities, these test cases are not considered in detail in this report, but the spreadsheet (listed in “Associated documents” on page 18) documents them.

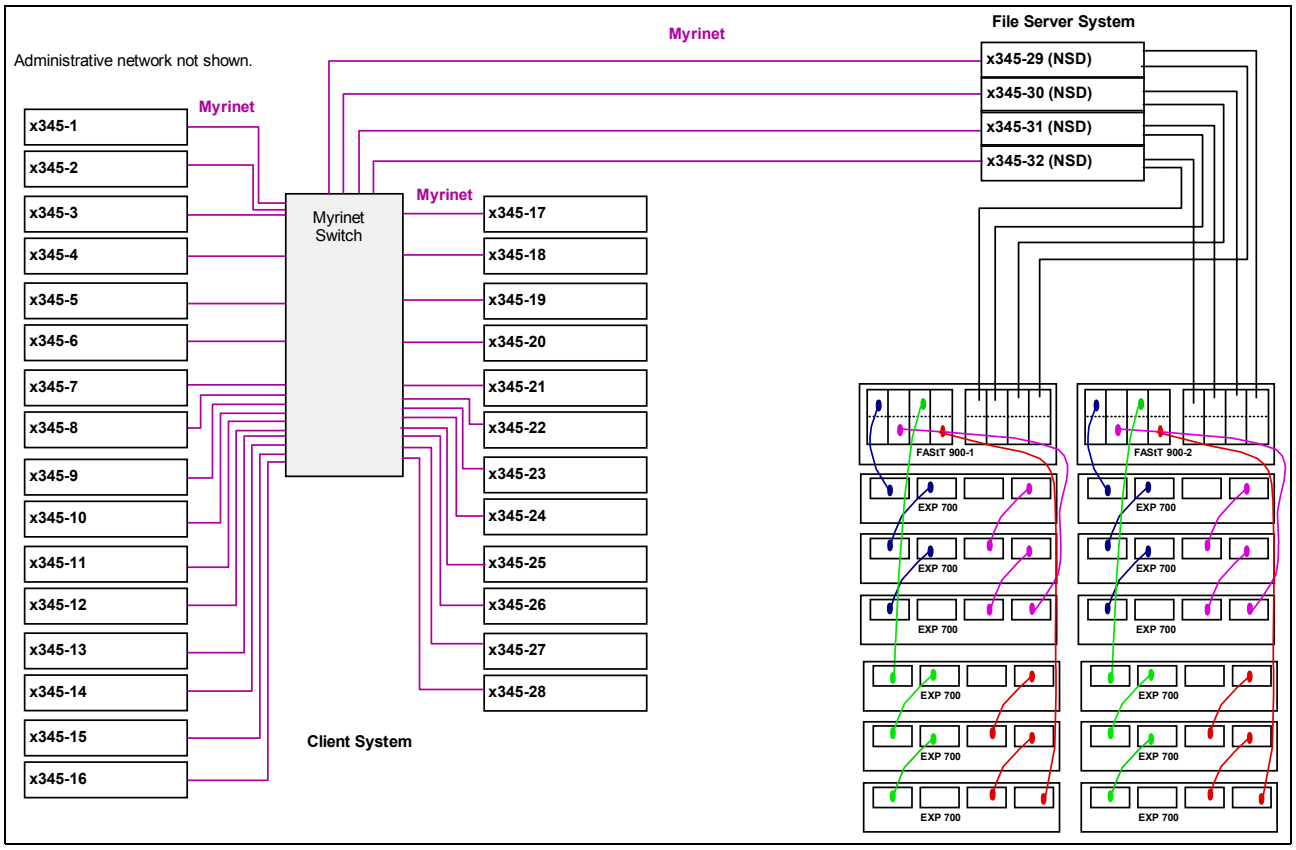

Figure 4 Tests #3 and #4A configuration

Test #4 - Miscellaneous tests

Several smaller, restricted scope tests[5] were conducted as time allowed. The purpose of these tests was to evaluate multiple simultaneous uses of the servers. They were completed using the Myrinet test environment, extending it as necessary. The significance of this test is that NSD servers do a lot of work for GPFS clients, especially in the case where there are a small number of servers relative to a large number of clients. However, the NSD servers do not always use all of their available bandwidth. Therefore, can this bandwidth be devoted to other tasks without adversely affecting NSD service?

Test #4A (see Table 7 and Figure 4) shows the result of one experiment running “client-on-server tasks” at the same time that regular client-on-client tasks are running (these are all large record sequential read tests). This is common strategy in an SP environment, but there are typically a large number of servers relative to the clients, unlike this case. The first three lines of Table 7 are various baseline measurements. First compare the client-on-client

[5] Because these tests were only fit into the benchmarking schedule as time allowed, they were not as rigorously executed as the other tests in this benchmark. Therefore, they should be considered only “proof of concept” tests, requiring further benchmark validation before drawing final conclusions.

and client-on-server baseline cases. The client-on-server case is significantly slower, probably because these nodes are doing both client and server work. Therefore, local client activities *appear* to interfere with local NSD service activity. But then compare the 8-way client-only baseline with the mixed client and client-on-server case, eight tasks in total). The results are *not* significantly different. This test was repeated several times, and the results were generally similar. (See the spreadsheet listed in “Associated documents” on page 18.)

Table 7 Client on server tests

Client task on NSD server node		**Client task on client node**				
Number of tasks	**Read rate MB/s**	**Number of tasks**	**Read rate MB/s**	**Total number of tasks**	**Aggregate rate**	
4	336	0		4	336	Client on server baseline
0		483		4	483	Client on client baseline
8	560			8	560	8-way client only baseline
4	212	4	364	8	576	Mixed client and client on server

Test #4B (see Table 8 and Figure 5) is similar in spirit to the previous test, except that the NSD servers are given NFS service duties instead of GPFS client duties (these are all large record sequential read tests). Also note that the NFS service is based on the local ext3 file system. In this test, because 100 MbE is used, the NFS client rates are quite low. Comparing the GPFS client rate in the mixed NSD/NFS test (third row, fourth column) to the GPFS client only baseline rate (second row), the difference is not significant. However, the NFS client rate in the NSD/NFS mixed test (third row, second column) showed a 23% drop in performance compared with its baseline (first row). Because the absolute value of this rate is small relative to the GPFS client rates, it is hard to draw definitive conclusions; however, the results show enough promise to warrant further testing.

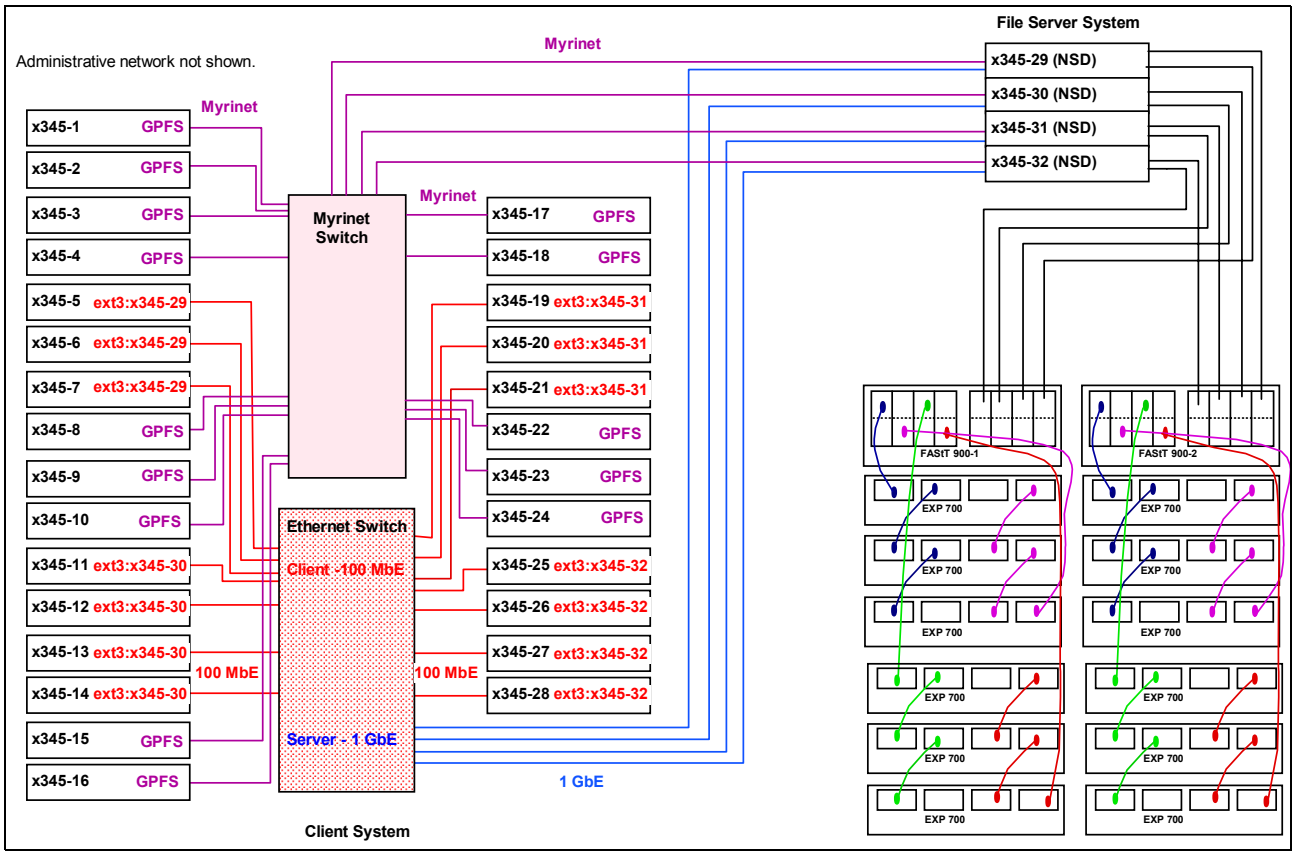

Figure 5 Test #4B configuration

Table 8 Simultaneous NFS and NSD server tests

NFS client tasks		GPFS client tasks				
Number of tasks	**Read rate MB/s**	**Number of tasks**	**Read rate MB/s**	**Total number of tasks**	**Aggregate rate**	
4	22.84				22.84	NFS clients only - baseline
0			413.88		413.88	GPFS clients only - baseline
4	17.92		428.33		446.25	Mixed NSD/NFS server test

Linear scaling in the number of storage servers

One of the well-known properties of GPFS in AIX systems is its ability to scale linearly in the number of VSD server nodes. Mathematically, this property is represented as O(n), where n is the number of servers. In more specific terms, if X is the I/O rate for one server, then the rate for n servers is $n * X - k_n$ where k_n is an overhead term. k_n can be unique for each n, but it is not necessarily linear in n.

Given the broad architectural similarities between GPFS/VSD/AIX/pSeries and GPFS/NSD/Linux/xSeries, it is reasonable to expect that this behavior would likewise be evident in the latter system. However, this assertion needs to be validated. Given the limited time that this test system was available, there was insufficient time to rigorously test this. However, some proof of concept tests give support to this assertion, as do several other indirect measures.

During initial testing to validate optimum FAStT900 parameters, we conducted a two-server test. The results of this test can then be compared with the baseline four-server tests used for the bulk of this study. These results are shown in Table 9. These were write tests on the Myrinet configuration (see “Test#3 - GPFS/NSD over Myrinet on servers and clients” on page 11) writing 1 GB per task to a common file. The numbers are consistent with the O(n) relationship.

Table 9 Ad hoc test evaluating scaling in the number of servers

			Natural aggregate	Harmonic aggregate
Number of server nodes	**Number of client nodes**	**Number of tasks per client node**	**Write rate MB/s**	**Write rate MB/s**
2	4	1	215	231
4	4	1	356	384

Another measure providing evidence that is consistent with this linear scaling observation can be seen by considering Tables 2 and 6. Consider the read performance in particular. Referencing a single client task job, the data rate is 111 MB/s for the GbE adapter case, and it is 185 MB/s for a single Myrinet adapter case. From this, we can infer for the GbE case that the performance is gated by the adapter at 111 MB/s, which would also be an upper bound data rate on a single server test where the server had only a single GbE adapter.

Next consider the multitask jobs in Table 2. This test uses four servers, each with a single GbE adapter. In this test, the peak read rates are approximately 360 MB/s, obeying the O(n) relationship to the 111 MB/s for a single server.

Again, this assertion needs more rigorous testing. But based on these preliminary tests and inferences, as well as the architectural similarity GPFS on Linux/xSeries to GPFS on AIX/pSeries systems where this relationship is well understood, this assertion seems plausible enough to warrant further investigation.

Failover validated by unexpected server failures

During these tests, one of the server nodes experienced several failures, forcing it to shut down and be rebooted. This problem appears to have been caused by intermittent adapter failures under heavy load, although the exact nature of these problems was never determined.[6] But more to the point, the failover features of this system performed according to expectation in each case. The jobs continued to run, albeit at a reasonably reduced data rate.

Alternative GbE configurations that are balanced

Ideally, the aggregate bandwidth of the various components of a storage subsystem (disk servers, disk controllers, disk drawers, and adapters) should be balanced. For example, if the aggregate bandwidth of the GbE adapters in the disk servers can sustain a data rate of 360 MB/s, yet the controllers can yield 700 MB/s, the controllers are being under-used and their cost wasted. The one place where this rule regarding balance might not apply is if the total disk volume needed requires a large number of physical disks (pdisks) to achieve it, yet the data rates needed by the application do not require the peak bandwidth that this number of

[6] Regarding the benchmark schedule, this was merely an inconvenience.

pdisks can deliver. In such cases, the number of pdisks attached to the controllers will exceed the bandwidth capability of the controllers.[7]

According to this design criterion, the configurations tested in this benchmark are unbalanced (see Figures 1 through 5). In earlier tests using four GPFS/VSD/AIX/p655 servers with the exact same FAStT900 configuration described in this paper, sustained peak aggregate data rates were measured as high as 740 MB/s. However, due to the gating effect of having either a single GbE or Myrinet adapter per server, the peak sustained bandwidths measured in this benchmark was much less. To circumvent this gating factor and thereby balance the design, a configuration using dual bonded GbE adapters was also tested. However, the performance of this configuration *appears* to have been gated by having insufficient CPU resource in the x345 servers to fully harvest the available bandwidth, thus leaving the design unbalanced.

This observation suggests two alternative configurations that can theoretically eliminate this gating effect and balance the design.

Note: The following configurations were not tested in this benchmark.

A 4:1 server/controller ratio

The configurations tested in this benchmark used a 2:1 server/controller ratio delivering at best 50% of the potential bandwidth from controllers, being gated by the single GbE adapter in each x345 server. Because it appears that an x345 server cannot fully drive more than 1 GbE adapter (see “Test#2 - GPFS/NSD over GbE on servers and clients” on page 7), then by increasing the server/controller ratio to 4:1, the bandwidth can be potentially balanced while maintaining a single GbE adapter per server. However, this requires attaching eight FC adapters to each controller in order to support the customary failover design (see Figure 6). Based on extractions from the results of this benchmark, it is reasonable to expect that four x345 servers, each with a single GbE adapter, should be able to harvest most of the bandwidth that a FAStT900 can deliver. But what is not tested is whether the FAStT900 has the processing power to effectively manage eight FC adapters, although only four FC adapters are needed for bandwidth purposes.[8]

Using four-CPU x360 servers

As previously observed, the x345 servers do not appear to be able to handle the load of dual bonded GbE adapters. However, because an x360 can be configured with up to four CPUs (twice as many as an x345), as well as having a more robust I/O capability, then perhaps they can be used in place of the x345 servers and achieve bandwidth balance with a 2:1 server/controller ratio (see Figure 7).

[7] For the FAStT900 disk controller and EXP700 disk drawers (14 pdisks/drawer), up to 16 drawers can be physically attached, yet the controller can only harvest the full potential bandwidth of at most five or six drawers.

[8] Because these are 2 Gbit/s FC adapters, capable of effectively delivering at least 120 MB/s of effective bandwidth, only one FC adapter per server and four FC adapters per controller is actually needed for bandwidth purposes.

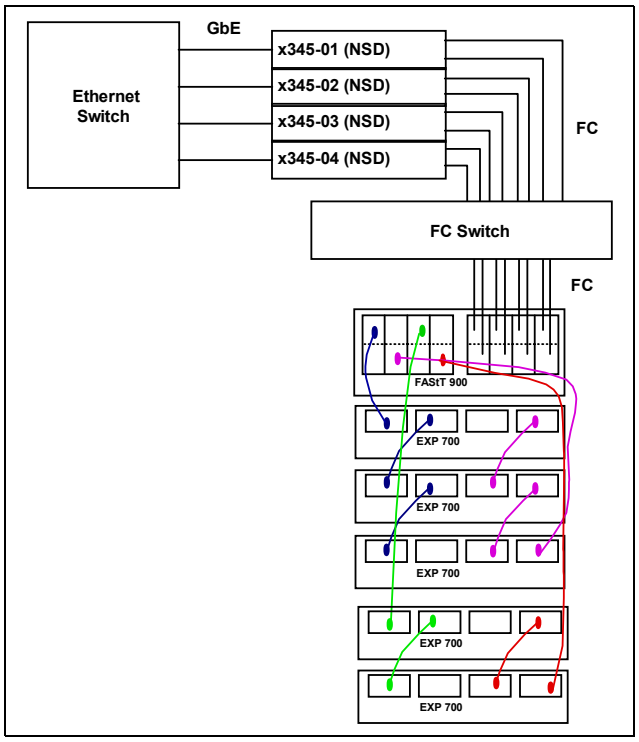

Figure 6 Configuration using a 4:1 server/controller ratio

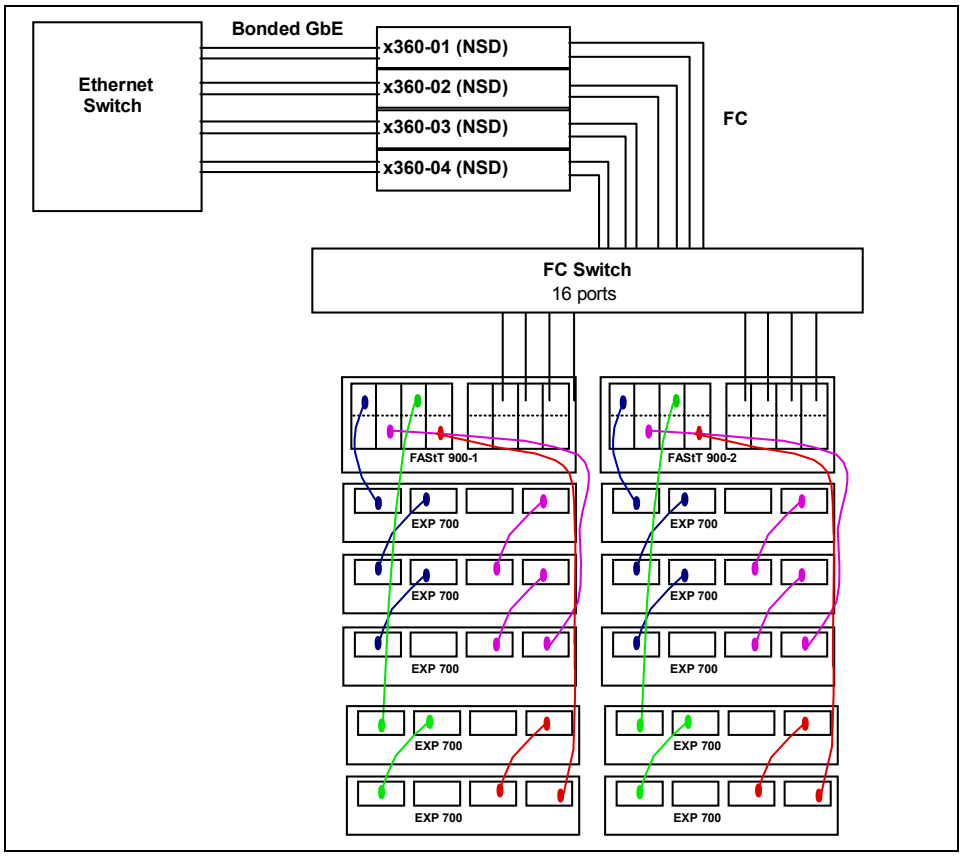

Figure 7 Balanced configuration using bonded GbE with a 4-CPU x360 in a 4:1 server/controller ratio

Summary

The GPFS file system in a Linux/x345 cluster configured in an NSD mode is shown to provide effective and efficient file service when using single GbE adapters per node or single Myrinet switch adapters per node. Alternative designs based on presently unsupported configurations (100 MbE and bonded Ethernet) show promise but require further investigation. The configurations in this study were not, however, bandwidth balanced. Therefore, alternative configurations are proposed, subject to validation.

Associated documents

More details about this test are available in the following documents:

- Benchmark Results: GPFS on Linux/x345 Cluster

  See spreadsheet file: BM_results.xls

- xSeries/Linux/GPFS: Benchmark Plan

  See file: BM_Plan.pdf

These documents can be downloaded from the IBM Redbooks™ Web server at:

`ftp://www.redbooks.ibm.com/redbooks/REDP3909`

At the Web site, click **Additional Material** on the right side of the window. Right-click the .zip file (REDP3909.ZIP). Select the **Save file as** option to download the file. Unzip into a directory of your choice.

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 email 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:

AIX®	IBM®	TotalStorage®
@server®	pSeries®	xSeries®
ibm.com®	Redbooks™	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.