Redbooks Paper

Raymond L. Paden

GPFS/NSD on Linux/xSeries using a 1 Gigabit Ethernet Network

A series of four benchmark tests were completed during 22–26 September 2003 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 x345-based systems using 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, while others are not.

This paper summarizes and interprets the results of the second benchmark test. It also proposes alternative untested designs that were suggested by these results.

The author

Raymond Paden 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 Ph.D. 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's 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.com/redbooks   **1**

Overview

Overall, the results of the tests were positive, most meeting or exceeding expectations. A few others, while 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.

Our experiences with these benchmarks led us to propose several additional, untested configurations, which we describe on page 10.

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. This paper describes the second test.

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 eServer™ xSeries 345 is an expandable, high-availability, high-performance server for e-business applications and mail/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 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 (FC) 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.

---

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

Node, disk, and file system configurations

The test cluster consisted of 32 nodes, as follows:

► x345, dual CPU, 2.8 GHz, 4 GB RAM

► Red Hat Linux 8.0, Kernel 2.4.20-19.8pok

► 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), and 16 FC adapters (two per server, four per controller, all at 2 Gbit/sec) in an NSD configuration. See Figure 1 on page 6 for an illustration.

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 configuration for the test

Each test had a different cluster network configuration for the test, each offering alternative price/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 #2 (A and B) - GPFS/NSD over 1 GbE on servers and clients

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

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

In the second variation (#2B), each of the client nodes were connected to the cluster network with a single GbE adapter. However, each of the server nodes were connected to the cluster network using two bonded GbE adapters forming a single channel (IP address). While 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 Figure 1 on page 6 and Figure 2 on page 7.

---

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

For reference, the other tests were:

► Test #1 - GPFS/NSD over 100 MbE for client nodes (see REDP-3796)

► Test #3 - GPFS/NSD over Myrinet on servers and clients (see REDP-3798)

► Test #4 - Miscellaneous tests (see REDP-3799)

Benchmark code description

The benchmark codes used in all tests 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

Let us explain the measurements and access patterns that we used, then we will examine the test results.

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 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. Since 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 (decision making under uncertainty) nature for the various queuing systems associated with parallel tasks, under normal circumstances tasks can and do terminate with some variance[3], though 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 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 perspective. From a system perspective, this rate is often lower than the actual system rate since it is heavily biased by outliers.

---

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

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.

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 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 that 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 this test

This section summarizes key results and their analysis from this test. Complete results for each experiment are listed in a spreadsheet accompanying this report.

Test #2 (A and B) - GPFS/NSD over GbE on servers and clients

Two variations of this test were performed. The first, variation #2A, was the very standard configuration where all nodes, both server and client, provided a single GbE adapter for GPFS traffic. In the second, variation #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. Figures Figure 1 on page 6 and Figure 2 on page 7 show the configurations for variations 2A and 2B.

First consider a large record sequential test for the single GbE adapter per server configuration. It will be used to establish baseline peak performance. See Table 1. 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.
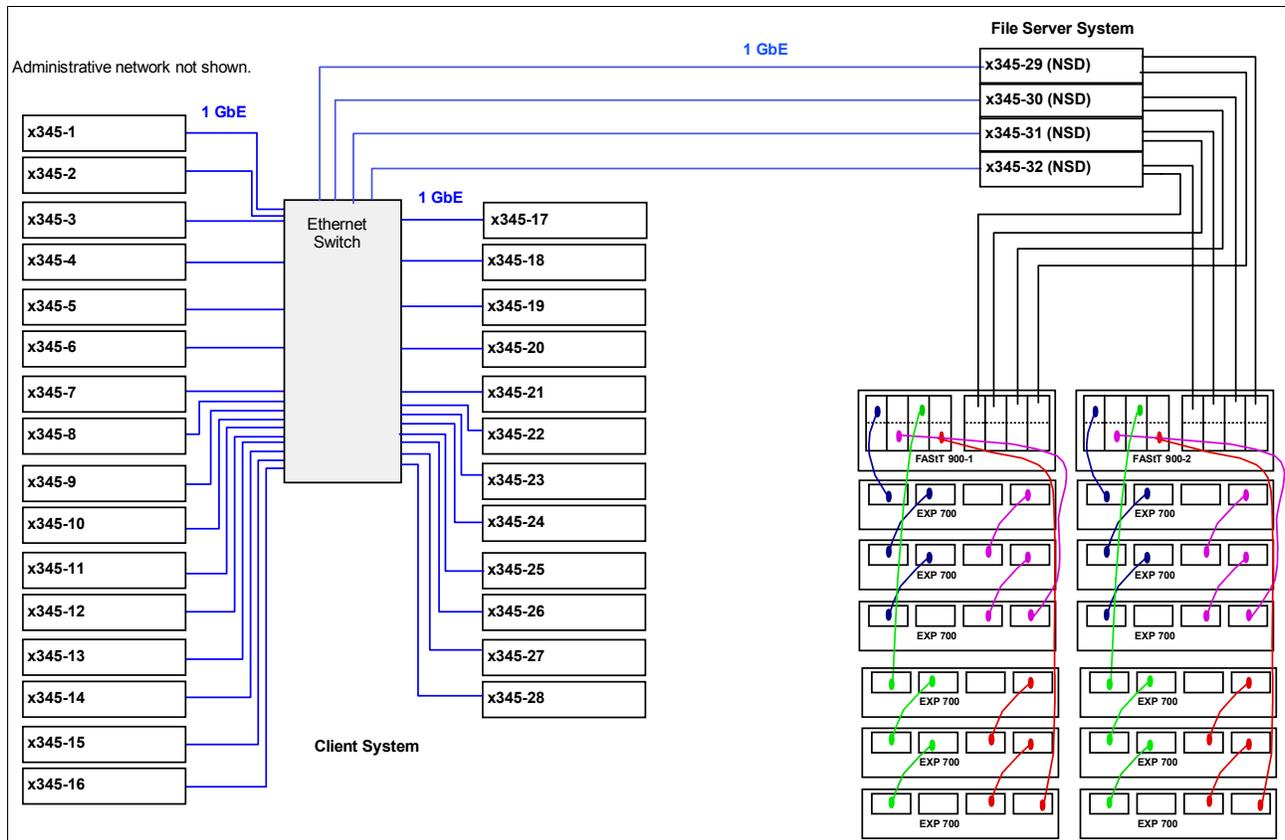

Figure 1   Variation #2A configuration

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 seven times 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; see Table 1. 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 9.

Table 1   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

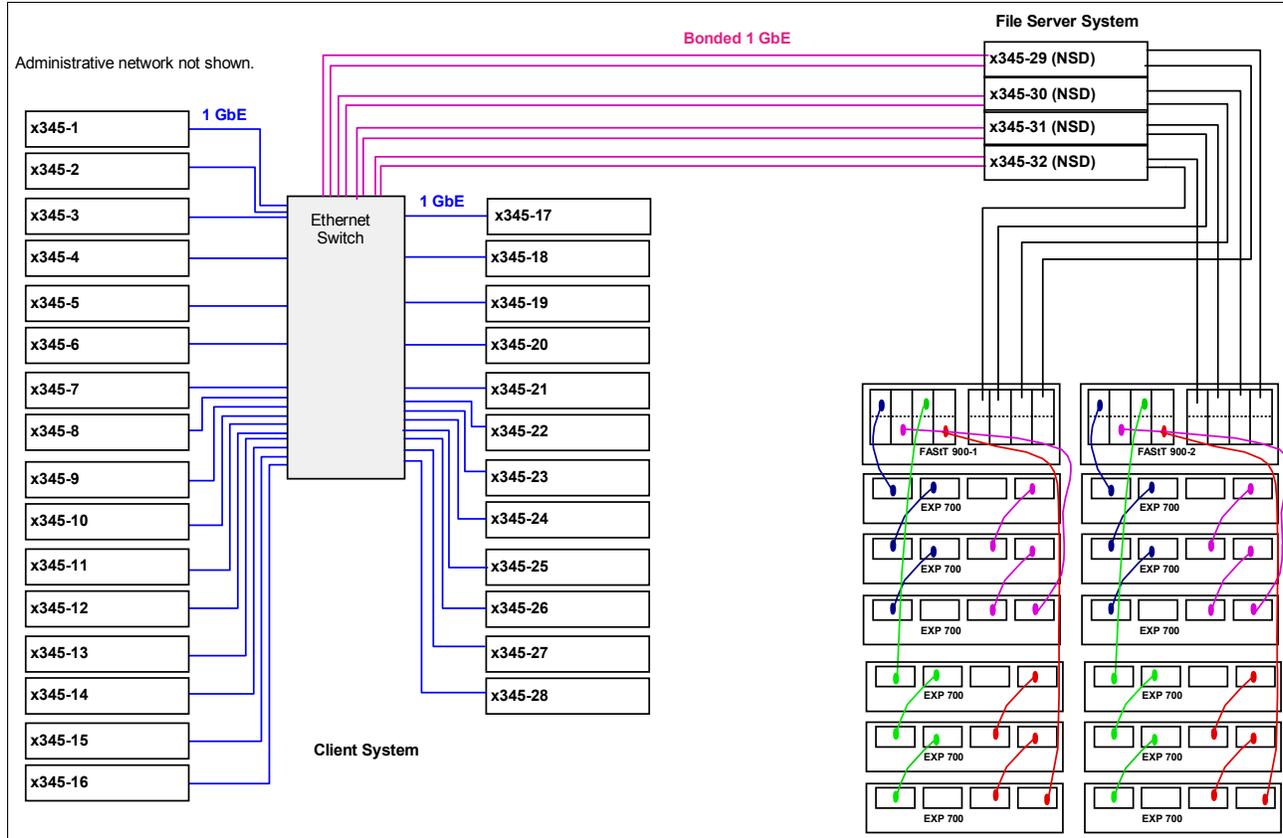| 4 | 1 | 292.54 | 320.46 | 296.06 | 337.92 | 74.02 | 84.48 |
| 8 | 1 | 383.69 | 338.26 | 383.86 | 360.21 | 47.98 | 45.03 |
| 16 | 1 | 315.12 | 347.32 | 335.75 | 357.07 | 20.98 | 22.32 |

Figure 2   Variation #2B configuration

In order to diminish the effect of the single GbE adapter/server gating factor, another set of tests were tried using a set of dual bonded GbE adapters per server. Table 2 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 rule of thumb for older pSeries systems like 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 percent for peak bandwidth jobs. It is therefore quite possible that having dual GbE adapters per server, albeit bonded into a single channel, required more than the additional 14 percent CPU bandwidth to fully utilize their bandwidth. Further testing is needed to fully validate this point.

Table 2   Large record sequential in dual bonded GbE adapter/server configuration

Number of client nodes	Number of tasks per client node	Natural Aggregate		Harmonic Aggregate		Harmonic Mean	
		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 utilized 6 percent of the GPFS block, and the random pattern thwarted cache optimization algorithms (see Table 3). In spite of low absolute performance, reads have the useful property that they scale 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 instance, 5X for eight client tasks; see the spreadsheet 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) then 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 3   Small record random in a single GbE adapter/server configuration

Number of client nodes	Number of tasks per client node	Natural Aggregate		Harmonic Aggregate		Harmonic Mean	
		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

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 4). Its performance is degraded since it is only utilizing 6 percent of the block for each record, but it gets 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 4   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

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 may 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.

Another measure providing evidence that is consistent with this linear scaling observation can be seen by considering Table 1 on page 6. Consider read performance in particular. Referencing a single client task job, the data rate is 111 MB/s for the GbE adapter case, while 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 1. This test is using 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, though the exact nature of these problems was never determined.[4] 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.

Summary

Two configurations were tested. The first one used GPFS in NSD mode with a single GbE adapter in each client node and in each server node. The tests on this configuration demonstrated effective and efficient file service. However, performance was restricted by having a single GbE adapter in each of four server nodes against 28 client nodes each having a single GbE adapter. Therefore a second configuration, while not officially supported, was tested as a means of addressing this load imbalance. It was like the first configuration, except that each of the four server nodes had dual bonded GbE adapters. As predicted, performance was improved, but not as much as expected. Further testing is warranted to explore this configuration, given the promise it shows in the area of load balancing.

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, then the controllers are being under-utilized and their cost wasted. The one place where this rule regarding balance may 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 pdisks can deliver. In such cases, the number of pdisks attached to the controllers will exceed the bandwidth capability of the controllers.[5]

According to this design criterion, the configurations tested in this benchmark are unbalanced (see Figures 1 and 2). 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[6]. 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 resources 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 that the following configurations were not tested in this benchmark.

---

[4] Regarding the benchmark schedule, this was merely an inconvenience.
[5] For the FAStT900 disk controller and EXP700 disk drawers (14 pdisks/drawer), up to 16 drawers may be physically attached, yet the controller can only harvest the full potential bandwidth of at most 5 or 6 drawers.
[6] Paden, R.L. Benchmark Report: I/O Benchmark Using x345 Linux Clients and p655/AIX/GPFS File servers. Technical Report, IBM, August 2003.

A 4:1 server/controller ratio

The configurations tested in this benchmark used a 2:1 server/controller ratio delivering at best 50 percent of the potential bandwidth from controllers, being gated by the single GbE adapter in each x345 server. Since it appears that an x345 server can not fully drive more than 1 GbE adapter (see "Test #2 (A and B) - GPFS/NSD over GbE on servers and clients" on page 5), 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 3). 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, though only four FC adapters are needed for bandwidth purposes[7].
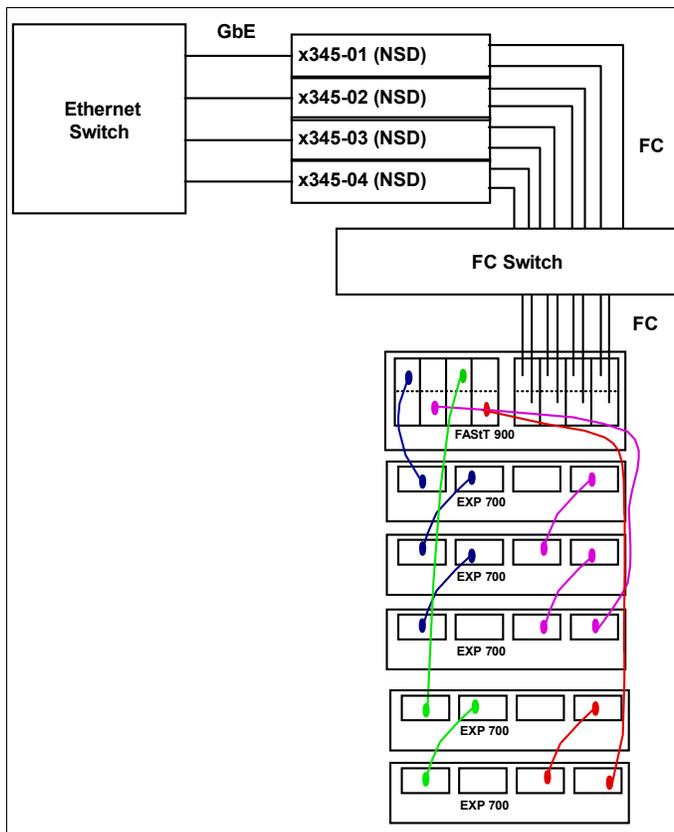

Figure 3   Configuration using a 4:1 server/controller ratio

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, since 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 4).

---

[7]   Since these are 2 Gbit/S FC adapters, capable of effectively delivering at least 120 MB/s of effective bandwidth, then only 1 FC adapter per server and 4 FC adapters per controller are actually needed for BW purposes.
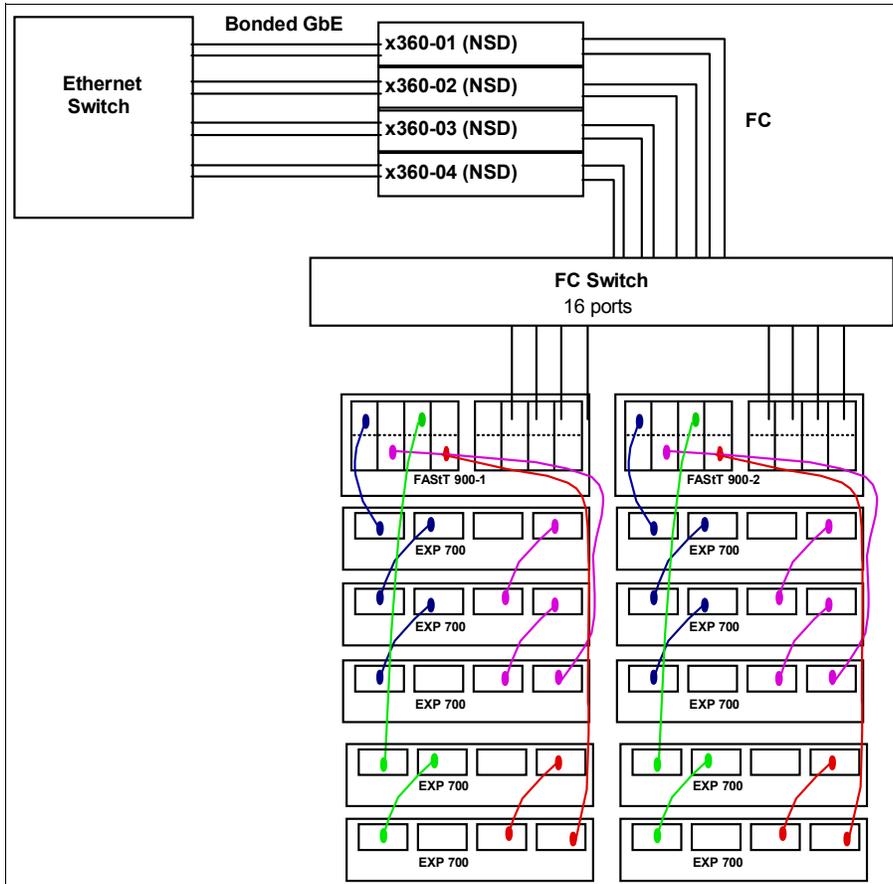
Figure 4   Balanced configuration using bonded GbE with a 4-CPU x360 server in a 4:1 server/controller ratio

Associated documents

More details on this and related tests are available from the following documents, which can be downloaded from the Web site:

ftp://www.redbooks.ibm.com/redbooks/REDP3797

This zip file contains the following documents:

► xSeries/Linux/GPFS: Benchmark Plan (BM_Plan.pdf)

► Benchmark Results: GPFS on Linux/x345 Cluster (BM_results.xls)

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.

This document created or updated on December 12, 2003.

IBM®

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 Internet note 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®	TotalStorage®
AIX®	pSeries®	@server™
eServer™	Redbooks(logo) ™	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.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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