Performance, Reliability, and Operational Issues for High Performance NAS Storage on Cray Platforms

John Kaitschusk, *Cray, Inc.*; **James Reaney** *BlueArc Corporation;* **Matthew O'Keefe** *and* **Christopher Hertel**, *Alvarri, Inc.*

ABSTRACT: *This report summarizes performance and failure analyses results for the BlueArc Titan NFS server in the Cray CASA test bed in Chippewa Falls, Wisconsin. The goals of this project are: to observe and describe the file transfer performance and fault recovery behavior of the BlueArc under light and heavy loads with varying file sizes and system access patterns; to measure the BlueArc Titan NAS server for reliability, uniformity, performance, and scalability in operational scenarios; and to develop configuration guidelines and best practices to achieve the highest performance, most efficient utilization and effective data management using the BlueArc platform capabilities.*

KEYWORDS: CASA (Cray Advanced Storage Architecture), NFS (Network File System), Network-Attached Storage (NAS), EVS (Enterprise Virtual Server), Storage Area Network (SAN), DVS (Distributed Virtual Server), SpecSFS.

Introduction

Cray Inc. is creating solutions that deliver more reliable, faster, and more scalable storage systems to its customers. This strategy is based on a new integrated storage system architecture, the Cray Advanced Storage Architecture (CASA). CASA leverages scalable cluster file system technology for fast IO transfer speeds within Cray's large supercomputer systems, and combines it with fast, interoperable, highly-manageable network-attached storage (NAS). The following figure shows a typical Cray CASA configuration:

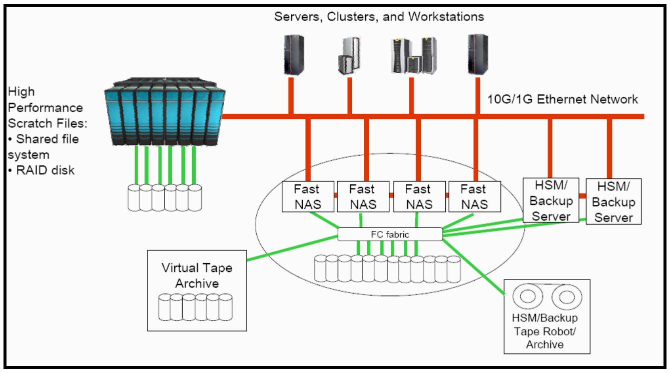

Figure 1: Cray Advanced Storage Architecture

With CASA, Cray is focusing on two critical storage elements:

- ever faster paths between its compute nodes and external NAS storage; and
- fast backup capability from the shared NAS pool using MAID (Massive Arrays of Idle Disks) technology.

While the highest speed attainable is important for NAS, MAID storage consists of large, fractionally-powered, high-density disk arrays that emulate tape devices – but faster and more consistently – without the complexity and errors of mechanical tape systems.

In a CASA configuration, the shared pool of fast NAS stages data into the Cray supercomputer for initial processing, and stages data out of supercomputer after a calculation completes. The NAS typically contains shared home directories, and can be used as a centralized local file store for all machines in a data center. Since supercomputer workloads require the processing of large numbers of both large and small files and large amounts of data, the speed and reliability of the NAS in support of this goal is critical. Recent improvements in the NFS protocol, including NFS RDMA[1], NFSv4, and pNFS, combined with massive industry investments in technologies to improve NFS performance, means that future implementations can scale in both performance per client and the number of clients supported over past NFS implementations.[2]

In response to these performance needs, Cray has chosen the BlueArc Titan NAS server for its initial CASA configurations, including Cray's CASA test bed in Chippewa Falls, Wisconsin (See Appendix A). In this report, we describe the performance and reliability results achieved for the CASA test bed configuration.

Benchmark System and the Bringsel Benchmark Test

For the BlueArc NAS benchmarking effort, the following configuration was used:

- 24 dual-core Opteron nodes connected through a Cisco 100-port 6509 switch to two BlueArc Titan servers (each with 6x1 Gigabit Ethernet ports supporting link aggregation)
- SuSE SLES OS running on the Opteron nodes:
  - primary OS image: SLES9 (kernel version 2.6.5-7.244)
  - secondary OS image: SLES10 (kernel version 2.6.16.21-0.8)
- each Opteron node running the Bringsel benchmarking and storage measurement software tool, developed by John Kaitschuck, a senior Cray field analyst and author of this paper.

Bringsel Description

Bringsel is a primary IO testing program that enables the use of either POSIX or MPI-IO calls to perform benchmarking for file systems and storage technologies, and to evaluate them for reliability, uniformity, performance and scalability.[3] Bringsel coordinates testing by enabling the creation of a large number of directories and files using both a threading model (POSIX) and the MPI library for multiple nodes.

Bringsel has run on large-scale SGI Origin systems, Sun enterprise-scale SMP systems and various clusters. The Bringsel feature set currently includes:

- flexible benchmarking setup and measurement via a configuration file parser and command line interface
- an environment variable interface for parameter passing
- ability to perform file operations in parallel to determine performance in this critical (but often overlooked) area
- ability to checksum (via the Haval algorithm) to verify the integrity of a given file and the associated writing or reading operations within the test space

Other IO performance measurement tools include `bonnie++`, `ior`, `iozone` and `xdd`. Bringsel differs from these in its emphasis on reliability, uniformity, and scalability:

- Reliability: for all files written, a checksum is computed and checked when a read access is performed.
- Uniformity: each test can be run several times to verify that bandwidth and latency is uniform across different nodes and at different run times for the same node.
- Scalability: bandwidth and operations per second of a single node and across many parallel nodes can be measured.

Bringsel can determine if a particular storage system, either serial or parallel, can meet Cray customer requirements for stability, robustness, predictability and reliability, in addition to speed.

Test Plan

To date (March 2007), ~25 separate test cases have been executed. These vary in the number of client nodes, number of threads per node, number of directories and their structure, number and size of files, block sizes, and sequential versus random access patterns. In general, the tests progressively add more configuration features (varying node counts, varying block sizes, larger files, more files, more directories, different OS and NFS clients, etc.) to find regressions against our four operational requirements for storage systems (reliability, uniformity, performance and scalability).

A description of the CASA test plan can be found in Appendix B of this document.

Organization of the Bringsel Benchmark Runs

The following Bringsel Benchmark tests have been or are being conducted on the CASA test bed at Cray's

[1] B. Callaghan, "NFS over RDMA," *FAST 02 Conference,* 2002.

[2] R. Martin and D. Culler, "NFS Sensitivity to High Performance Networks," *Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modelling of computer systems*, pp. 71-82*,* Atlanta, Georgia, 1999.

[3] J. Kaitschusk and M. O'Keefe, "Bringsel: A Tool for Measuring Storage System Reliability, Uniformity, Performance and Scalability," *CUG 2007 Proceedings,* (this publication), Seattle, Washington, 2007.

Chippewa Falls, Wisconsin facility. (See Section 3 for test results.)

1) Simple single file write (per node in all benchmark runs), single directory (tests 01.00 to 01.01)

   In these Bringsel benchmark runs, a single node writes a 500 MB file using 8K blocks, followed by a run with all nodes writing their own 500 MB file to the same directory.

2) Simple single file write, moderate size directory tree (tests 01.02)

   Each node writes a 150 MB file, one to each of 12883 directories in a symmetric tree structure (1,3,3,5,5,7,7).

3) Single file write, vary block sizes, single directory (test 01.03 to 01.05)

   Each node writes a 500 MB file using varying block sizes, from 16K to 1024K. Write operations vary from sequential to random access.

4) Multiple file writes, varying block sizes, moderate size directory tree (test 01.06)

   Two files of 100 MB size each are written into a directory tree with 283 directories using roughly 1.5 Terabytes of space. One goal is to measure uniformity of access speeds across the directory structure.

5) Single large file write, varying block sizes, single directory (01.07 to 01.09)

   These runs simulate the large file writes (varying from 125 GB to 500 GB) commonly performed at the end of a large simulation. Block sizes are varied and the operation is repeated twice to verify uniformity.

6) Multiple file writes and reads (one file per node) per directory, large directory tree (01.10 to 01.14)

   A large symmetric directory tree structure is created with 55,392 directories, each containing 24,310 MB files (requiring a total of about 16.5 TB, which is roughly 97% of the available Fibre Channel storage on the Titan). A series of benchmark runs is performed over this directory tree structure with sequential and random reads and writes, and a sequential directory tree walk.

7) Single and multiple node (moderate and large file) writes, single directory, whole cluster tests (01.16 to 01.20)

   This series of tests uses 250 MB files per node and 512 KB block sizes across all 24 nodes. Link aggregation is enabled in most tests but turned off in one test to measure its effect. File creates, reads and writes are performed. The goal of these tests is to determine the maximum bandwidth through the BlueArc Titan server.

8) Multiple tests to measure storage metrics under various failure scenarios (failed drive, failed network, failed NFS server, etc.) (01.21 to 01.26)

   These tests measured several things: (1) the BlueArc system's reaction time to induced faults (time to detect the fault, time to recover from the fault), (2) BlueArc performance while recovering from the fault, (3) BlueArc performance after recovery from the induced fault, and most importantly, (4) data integrity, to make sure files written before, during and after recovery can be read and contain the same file data written.

9) Tests to measure data migration performance from Fibre Channel to SATA tier on the BlueArc; also various caching and access pattern tests (including read-ahead and metadata caching) (01.15, 01.21 to 01.32)

   Some of the data migration tests are still pending. The goal is to determine if the operational storage metrics are met under various failure scenarios and workloads.

Selected Results from Initial Bringsel Runs

The Bringsel benchmark runs started in August 2006, beginning with a SLES9 Linux kernel (2.6.5), and have been run continually since.

Given the demands of the Bringsel NFS benchmark, problems with NFS client memory management resulted in excessive memory usage. A kernel patch was applied that fixed this problem (this kernel patch also worked for AWE, a Cray customer in the UK also using a BlueArc system).

Another Linux NFS client issue was discovered when several small file read and write benchmarks yielded very low performance (low bandwidth and high latency per operation). Several weeks of investigation led to the decision to upgrade to SLES10 (2.6.16) to take advantage of improvements to the Linux NFS client in that kernel. This upgrade yielded a large improvement in small file performance.

The initial BlueArc NFS tests were performed without link aggregation turned on in the Cisco switch, and with all 24 nodes accessing one BlueArc Titan server over a single network link. Later tests turned link aggregation on. Tests are also being performed without link aggregation but with multiple nodes sharing all the network links into the BlueArc Titan server. This allows us to test the maximum bandwidth achievable on the platform.

Key results from the tests run so far on the CASA test bed hardware include the following:

- A total of over 400 TB of data has been written without data corruption or access failures.
- There have been no major hardware failures in ~120 days of operation (as of January 4, 2007). The only active test component to fail was a single FC DDU, which occurred with no data loss.
- Generally, the results are predictable and relatively uniform. Repeating the same tests yields similar performance results. Uniformity of performance in non-oversubscribed mode, for the clients, was good given dedicated transport resources.
- With some exceptions, the BlueArc aggregate performance generally scales with the number of clients.
- Recovery from injected faults was fast and relatively transparent to clients: clients waited until the BlueArc system returned to service and then continued operation. As expected, NFS behaved as a resilient protocol.
- 32 test cases have been prepared, about 25 of varying length have been run, all file checksums to date have been valid
- Early SLES9 NFS client problems occurred under load. These were detected and corrected via kernel patch. This same patch was used at AWE, a Cray customer site, which experienced the same problem
- There was one storage array disk failure, but with no data loss or significant performance impact.
- Spatial use is from 0% to 100%+ during various test cases, so that all storage was accessed during some of the benchmark runs.
- Test case durations are from several minutes to several days.

Bandwidth and IOs-per-second Performance

The following sections provide results for bandwidth and IOs-per-second tests for BlueArc running both SLES9 and SLES10.

Test 1: TC01.07 (SLES10)

These results show that BlueArc sequential write performance is not dependent on block size. The write performance for a single node is approximately 112 MBytes/second, or close to peak line speed (that is, the link is nearly 100% utilized); the 125 GB file is transferred in approximately 1200 seconds (20 minutes).

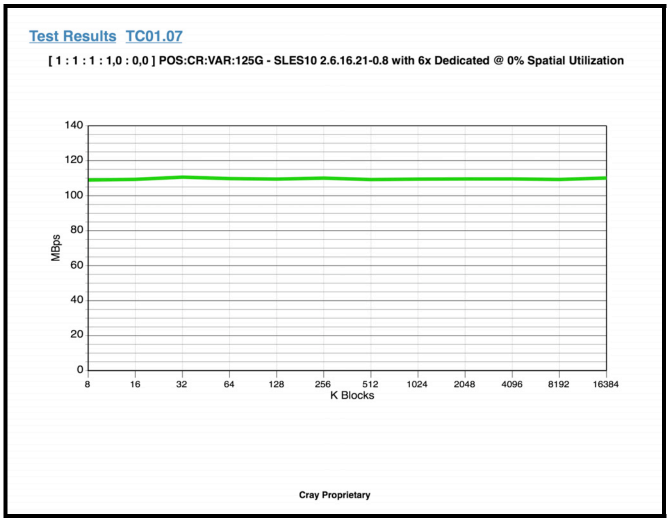

Figure 2: TC01.07 (SLES 10)

Test 2: TC01.01 (SLES10)

In this test, each node creates its own 500 MByte file (single stream) using sequential write operations and 8K blocks. The operations are scaled to use all the nodes in the cluster starting with one node and scaling by 2.

The average aggregate bandwidth for these runs was ~300 to ~360 MBytes/sec across all nodes – about 60% of the peak line speed available. These speeds for a write-oriented workload are consistent with the amount of storage array bandwidth in the BlueArc configuration.

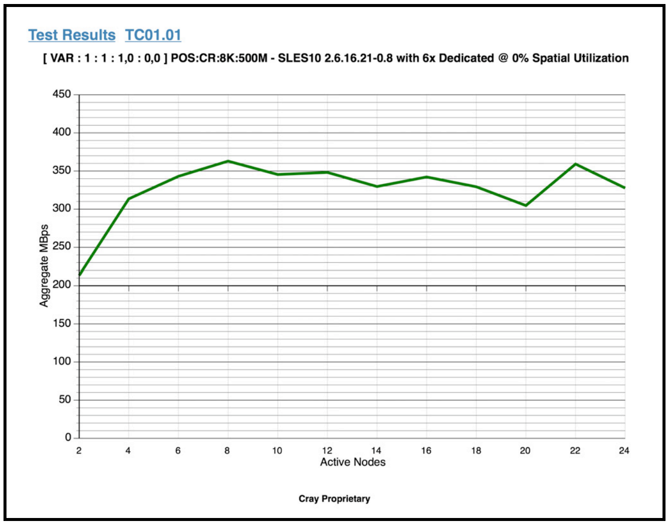

Figure 3: TC01.01 (SLES 10)

Test 3: TC01.05 (SLES10)

In this test, each node creates a 500 MByte file using random write operations and variable-sized blocks. The operations are scaled to use all the nodes in the cluster starting with two nodes and scaling by 2. The operations were iterated across the following block values: 16K, 32K, 64K, 128K, 256K, 512K and 1024K.

The average aggregate bandwidth for these tests was ~450 MBytes/sec. Beyond 64K block transfers, transfer rates per node were not affected by block size.

In general, the BlueArc performance was consistent, without noticeable anomalies, except for certain small block access patterns.

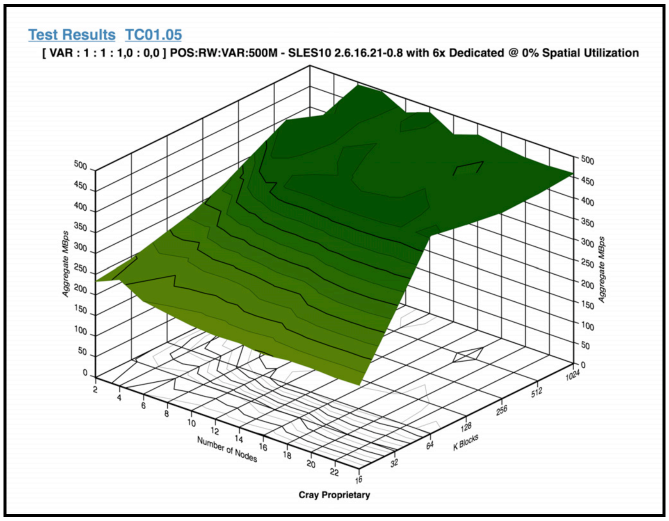

Figure 4: TC01.05 (SLES10)

Test 4: Random Write Results (SLES9)

The following results show the performance drop for small block access performance, in this case, random writes on 310-Megabyte files contained in the upper directories of a large directory tree. For 1K small block accesses using the SLES9 Linux kernel, only 0.3 MBytes/sec bandwidth is achieved, and bandwidth only slowly improves to a maximum of 6 MBytes/sec as the block size increases. After discussions with BlueArc regarding this poor performance, it was decided to upgrade to the SLES10 Linux kernel to determine if the Linux NFS client and VM changes within that kernel might improve performance.

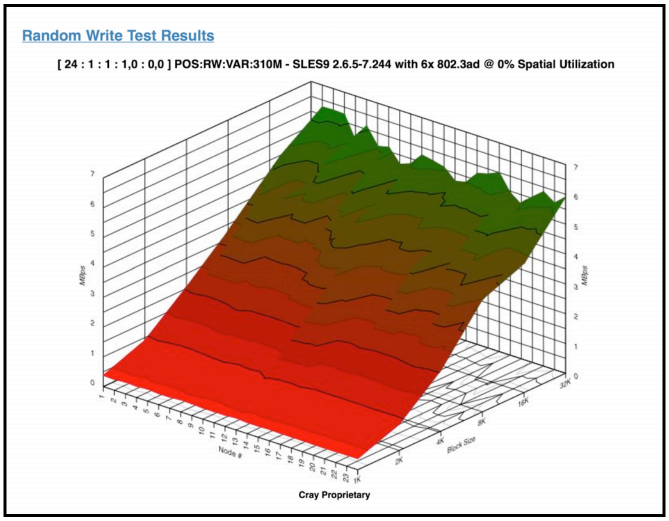

Figure 5: Random Write Test – SLES9 2.6.5

Test 5: Random Write Results (SLES10)

The following figure shows the results from the same tests (Test 4, above) with the SLES10 (Linux 2.6.16) kernel. The random write results are much improved, though still somewhat below expectations.

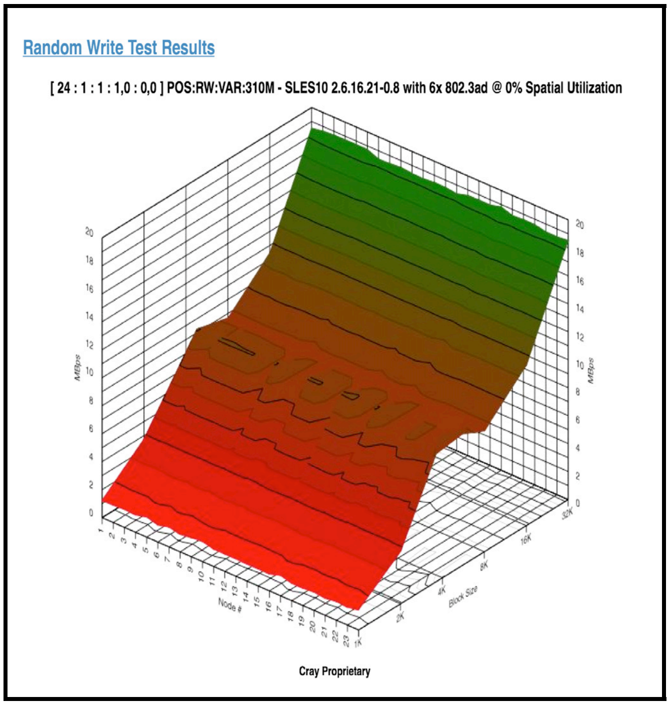

Figure 6: Random Write Test with SLES10

Field-relevant Issues

Several issues were uncovered during testing that would affect Cray deployments of BlueArc systems in the field.

SLES9-versus-SLES10 Kernels

The Linux kernels used in the SLES10 distribution have significantly better NFS client performance and behavior than those in the SLES9 kernels. This is because of major changes in the NFS client and related kernel support (virtual memory, block device drivers, vfs layer, and so on).

The NFS client directory in SLES9 is 524kb. The same directory in SLES10 is 708kb. The patch between them is 666kb. A significant portion of that patch includes NFSv4 features, plus there has been a lot of work on the read and write paths as well as the `open()/close()` paths. The change logs for just the kernel.org kernel from 2.6.5 to 2.6.16 mention NFS over 1000 times.

In addition, without a patch, the SLES9 NFS client interactions with the virtual memory system were unstable. Under slightly heavier loads on large block/file sizes, the Linux clients would hang or crash due to memory starvation. While stability improved with a patch (nodes no longer hang or crash under load), residual performance issues can still be expected.

The same small block random IO tests were run under a variety of mount options and MTU sizes (the BlueArc can support jumbo frames) with no noticeable impact on performance. Further SLES10 testing showed differences between SLES9 and SLES10 of from ~3x to ~10x, depending on the small block random access case in question.

Cisco Link Aggregation

During the early and middle stages of testing, link aggregation was used to create one large, logical (6x) connection into the BlueArc Titan server. Client accesses would load balance across these aggregated links. Though link aggregation simplifies system configuration and management, the performance penalty was significant, such that only 35%-45% of peak line performance (one-way) was achieved when using aggregation. In contrast, by statically assigning clients to particular IP addresses (and hence Ethernet connections), 60% of peak line performance (balanced) could be achieved. This performance anomaly will be further investigated, but it is believed that with 100 or more clients it would more likely to saturate the connection and therein utilize the full bandwidth of both the Titan server and Cisco switch.

Failure Recovery Study

The BlueArc fault recovery study tested the following failures (each injected manually). All failures were detected very quickly, within one second.

Disk drive failures

For the disk drive failures, a hot spare was enabled within one second. Rebuild times took approximately 3.5 hours, during which time Titan read performance degraded by 20%, and small write performance degraded by 50%.

SAN path and switch failures

SAN path and switch failures were followed, within 5-10 seconds, with a failover to a redundant, operational path. The performance impact after the path failover was negligible.

Titan server failures

Tens of seconds were required to transfer EVS state information from a failed (in our case, powered-off) Titan server to its redundant partner server. The performance impact after Titan server failover was negligible, but in current performance studies, only a single head is used at one time.

The Titan logging and error reporting mechanisms were clear, concise, and easy-to-understand. These mechanisms made it possible to quickly identify the problem cause, the time the problem happened, and the recovery mechanism and any other associated steps necessary to complete recovery.

At no time in any of the associated failure testing were the files being written corrupted in any way. Data integrity tests were run both before and after fault injection on the Haval checksums to insure data integrity.

Migration Study

As of March 2007, the data migration tests on the BlueArc system are in progress.

Performance Analysis

The initial set of tests on the BlueArc focused on reliability and uniformity, but achievable peak performance was important as well, where performance expectations were higher than what were achieved. However, in reviewing the results internally and with BlueArc, it was determined that the BlueArc Titan-2 in the test bed has too few storage array controllers to saturate the Titan-2 FSM and NIM components.

In the performance study described in the following section, the Caltech Bandwidth Challenge Results with BlueArc Titan, five 2882 Engenio controllers were used. The newer Engenio 3994 controllers have higher performance and can manage more drives per controller. BlueArc expects the performance of two or three 3994 controllers (now shipping with the Titan) to match the performance of five 2882 Engenio controllers.

In addition, the test bed Titan was configured with a file system volume mapped to a particular controller and set of storage shelves. It is expected that more performance can be gained by striping the file system volume across multiple controllers, and the plan is to test this configuration with future systems.

BlueArc also suggests enlarging the file system block size for a possible positive impact on performance. Other performance enhancement techniques are expected as well that can be applied in the next set of tests.

Caltech Bandwidth Challenge Results with BlueArc Titan

For comparison purposes, data has been included from the SC|06 Bandwidth Challenge that used BlueArc Titan servers. This data was provided by James Reaney of BlueArc.

The configuration used for the SC|06 Bandwidth Challenge included the following:

- 130x146GB 15k FC drives, with five pairs of the 2882 controllers (the same controllers on the Titan in the Cray test bed);
- a dual Titan 2200 cluster with the "Razor" firmware update; and the Cluster Name Space enabled
- a file system workload of 32kB blocks, large 2GB+ files, and a 50/50 read-write mix.
- a Cisco 6509 network switch with Sup720-3B and 6724 line cards, and jumbo frames enabled.
- 36 compute node clients, 144 cores total (2-CPU, dual-core Woodcrest), running 8 processes each.

  Each client mounted a single NFS export. The EVS from both Titans appeared as sub-folders under the mounted root; hence, from the client perspective, it was a single file system.

- peak bandwidth of nearly 1.4 GBytes/second, and a maximum of 842 MBytes/sec from a single Titan during the testing.
- IOPS (IOs per second) was 44,000.

  Note that these tests focused on bandwidth, not IOPS; more spindles would have been used to increase metadata and small file performance.

The following figure shows bandwidth over the course of the five days of the Supercomputing '06 conference, measured in real time. The peak bandwidth of 1.4 GB/second was achieved on November 18th. The workloads vary during each day based on the workload generated by other systems participating in the bandwidth challenge.

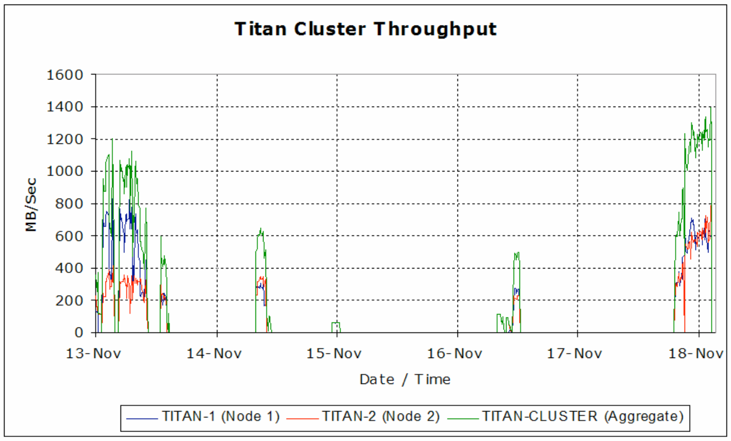

Figure 7: Caltech Bandwidth Challenge Results

Summary and Future Work

Initial results showed that the BlueArc reliability, uniformity, performance, and scalability meet Cray's goals for NAS in the CASA storage solution.

The imminent release of the BlueArc Titan 2 includes the following upgrades:

- from 2-Gbit/second FC interfaces to 4-Gbit/second interfaces
- from Engenio 2882 storage array controllers to 3994 controllers
- from 2-way to 4-way Titan server support for clustered name spaces
- from 6x1-Gigabit Ethernet interfaces to 2x10-Gigabit Ethernet interfaces per Titan server

BlueArc and other vendors (like Agami) continue to trend towards faster and more scalable NFS.

This report shows that NFS has the potential to become a capable protocol for the supercomputing data center. NFS as implemented in the BlueArc Titan server is highly resilient and uniform, can be tuned to support large transfers at line speed, and is balanced enough to support large file transfers simultaneously with small file, metadata-intensive workloads.[4]

The tests described here were intended only as a baseline to determine the basic capabilities of the BlueArc Titan. More extensive benchmarking with a larger client cluster and directly with Cray systems, more robust client nodes, faster storage arrays, and a larger tier of Titan servers is just a start. The CASA test bed, combined with the Bringsel benchmarking tool, provides an infrastructure for testing other NAS systems, cluster file systems, and storage array hardware in a controlled environment. Future work could include the following testing:

- extend Bringsel capabilities:
  - include better post-processing of data (including automatic generation of graphics)
  - include a conveniently accessible archive data format, and automated archiving of runs
  - include support for different file sizes and different operations (like spec) in the same test
  - coordinate multiple runs against a tier of NFS servers
- make extensive additional performance and scalability runs:
  - try to recreate Reaney's Supercomputing 06 benchmark run (see Appendix C) and,
  - test the setup to verify the high performance achieved in those tests including, if possible, tests with new, faster controllers
- run benchmarks against a tier of at least 4 Titan servers running with cluster name space enabled to determine its performance overhead
- test new storage arrays to track performance differences and to build a performance database to assist customers in configuring their storage
  - in general, test with different physical drive configurations
  - run rebuild time tests with different drives and under different workload conditions
- execute the same Bringsel-based NFS workload against other NFS server technology (including low-cost Linux servers) to compare the results to BlueArc Titan and other NFS solutions
- perform more studies on link aggregation and its affect on system performance
- test new BlueArc Titan server product releases

-

[4] The latest BlueArc SpecSFS performance results can be found at the spec web site: http://www.spec.org/osg/sfs97r1/results/sfs97r1.html

Appendix A: CASA Test Bed Description

This is a description of the CASA test bed at the Cray, Inc. manufacturing facility in Chippewa Falls, Wisconsin. The test bed includes hardware and software tools that can be configured and reconfigured so that a variety of storage models can be tested. The following sections describe the components of the CASA test bed.

Cray XT3 System

The Cray XT3 system used in the CASA test bed is a one-cabinet system with 86 processors. It includes the following:

- 76 compute nodes
- 10 SIO nodes
- processor speed is 2.4Ghz
- memory size is 4Gb per node on compute nodes
- QLA2342 HBAs
- Intel pro Ethernet cards.
- the system has two Lustre filesystems
  - a single node mds/ost, /lus//lus/nid00036, 785G
  - a (5)OST (1)MDS parallel, /lus/nid00008, 3.9T
- the system uses the PBS batch system.

PC Cluster

There are 30 PCs in a rack located next to the BlueArc cabinets. This cluster will be expanded to include a set of more powerful PCs. It will be used to drive I/O testing against the NAS gear, the disk arrays, and the MAID system, among other storage products.

The higher-powered PCs that will be added to the cluster will have PCI-Express (PCIe) buses and can be used as Lustre OSTs and MetaData servers, or as NAS heads for NFS/RDMA and pNFS testing, or as intermediaries between the other PCs and the storage systems. Future Cray products will include PCIe bus support, so this set of PCs will be useful in testing NIC and HBA compatibility as well as I/O capabilities.

All of these PCs are equipped with two or more GigE ports.

BlueArc Titan NAS

The lab is equipped with two BlueArc Titan NAS appliances, bound together in an active/active cluster. The Titan cluster provides NFS file sharing services (and can also provide CIFS and iSCSI).

One of the Engenio arrays (described below) is directly attached to and is therefore essentially part of the BlueArc NAS cluster. The BlueArc cabinet includes a pair of FibreChannel switches which interconnect the disk array with the Titans.

Copan MAID Array

The Copan MAID system is another disk array, but of a different type. The Copan uses commodity SATA drives, densely packed into canisters (drawers). Each canister holds fourteen drives and each shelf holds eight canisters. The Copan system in the CASA test bed has four shelves populated with a total of 448 drives. This density makes the Copan system quite heavy – the rack sits flat with no feet because the concentrated weight could puncture the floor tiles.

IO speed is not one of Copan's guiding ideals. Instead, the Copan MAID system aims at providing very reliable—though slower—archive-class storage. Its target niche is as a replacement for tape library systems. It can emulate a wide variety of tape libraries, and it can run several emulations at once. The Copan achieves disk reliability and longevity by systematically spinning drives up and down to ensure that they are neither running continuously, nor idle for long periods of time, but are properly exercised so that they do not seize up. They call this “Disk Aerobics”.

Like the Disk Arrays, the Copan presents storage via the FibreChannel SAN.

Disk Arrays

The Lab design specifies several disk arrays. As of this writing (March 2007), some are installed, and some remain to be installed. The disk array systems allow us to combine disks into different RAID configurations, each with different IO speed and latency characteristics. The resulting storage is then parcelled out and made available via a FibreChannel network.

Most of the disk drives in the array are high-speed, high reliability drives. There are, however, some slower SATA drives that have been added to the array that is directly connected to the BlueArc Titan systems. These will be used for experiments with data migration, and to verify their manufacturer's MTBF claims.

The arrays purchased for the CASA test bed are from Engenio Corporation and DDN. As needs and testing goals change, these systems can be moved in and out of the CASA test bed.

Networks

There are two networks: The FibreChannel Storage Area Network (SAN) and the Ethernet-based network.

The SAN will be organized around a pair of QLogic FibreChannel (FC) switches. In addition to the FC ports already present in the Copan MAID, BlueArc Titan systems, and disk arrays, we have purchased a set of eight PCIe FibreChannel HBAs so that the PCs can access SAN disk space directly.

On the Ethernet side, there are three switches that will be used in the lab:

- a 48-port Netgear 10/100 switch which is used for network management connections (e.g., connecting to the management ports on the BlueArc Titans)
- a 48-port Netgear GigE switch (that is currently idle)
- a Cisco 6509 enterprise-class switch – the core of the CASA test bed network

FibreChannel Network

The BlueArc system is bundled with two Titan NAS heads, two Brocade FibreChannel switches, and an Engenio disk array – a "SAN-in-a-box".

The plan is to use the QLogic switches to build a second SAN that will connect the Copan MAID system, the FC HBAs in the PCs, and a separate Engenio disk array. The two SANs will be linked together by connecting the Brocade switches to the QLogic switches. The main reason that the two SANs need to be connected is that the BlueArc needs to be able to transfer data to and from the Copan MAID array.

The QLogic switches are not in place yet, so as an interim measure the Copan MAID has been directly attached to the Brocade switches included with the BlueArc system.

Note that FibreChannel networks are almost always built as redundant pairs ("dual fabrics"). Disk arrays typically have dual controllers and client PCs either have two HBAs or one dual-ported HBA. SANs carry raw block data, and clients typically do not handle service interruptions gracefully. The dual fabric ensures continuation of service in the event of a controller or switch failure.

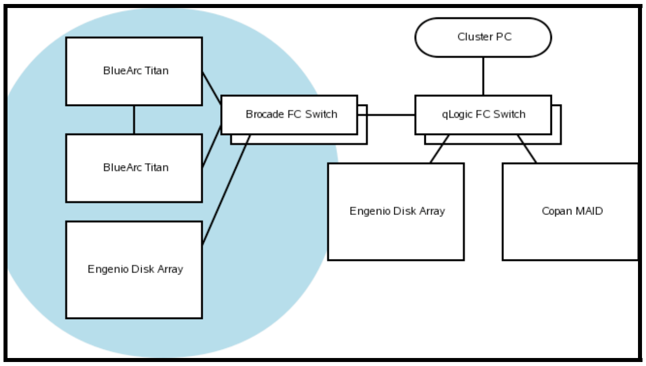

Figure A-1: FibreChannel Network

Ethernet Networks

There are several independent networks running in the CASA test bed. Networks are separated based on functionality and access restrictions. This helps manage physical and logical connectivity as well as access controls.

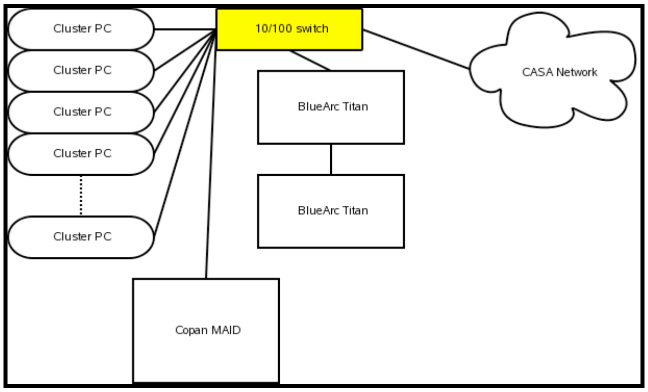

Figure A-2: Ethernet Networks

1 - Management Network

The 30 single-core, dual-processor PCs currently in the lab each have a 10/100 Ethernet port set aside for network management. That port connects to a small, embedded Linux-based terminal server which can be used to perform various management functions including power cycling the PC. The Copan, the BlueArc Titans, the Cisco 6509 and, in fact, most gear in the room has some sort of management port.

Instead of using expensive GigE connections on the Cisco, we purchased a cheap 48-port 10/100 switch that has GigE uplinks. Most management connections will be distributed out of the 10/100 switch. There are also a few serial console ports on some of this gear, and there is a Cyclades Console Server to access those.

2 - Data Network

The high-speed ports on the Cisco switch are for use in building the testing fire hose. That is, a separate data network that can pass very large amounts of data all at once. This network will be kept logically separate from the management network so that the two traffic types do not interfere with one another and so that users with access to the management network cannot also gain access to systems (such as Supercomputers) that are on the data network.

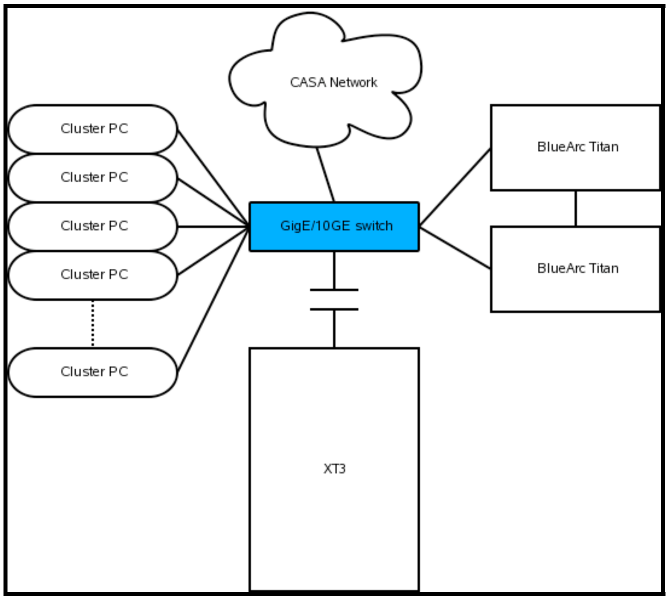

Figure A-3: Data Network

Once again, outside users who need access to serial ports or management ports on CASA test bed equipment will not be able to access the CASA data network. As an additional measure, outside users will be disconnected and locked out whenever non-CASA equipment (again, Supercomputers) are connected, and vice versa.

CASA Test Bed Network Access

Access to the CASA test bed network is via a PC configured as a gateway and firewall. One of the cluster PCs has been allocated to fill this role, although a less-powerful, less-expensive system should be used. The gateway/firewall PC has several duties, none of which are taxing or requires high speed.

The gateway/firewall system has a connection to the Internet so that it can access common network services such as (but not limited to) DNS, Network Time Protocol, software update services, web and FTP services, and so on. Inbound connections (that is, traffic not initiated from the gateway) may be restricted to SSH and OpenVPN. External ping traffic is also be permitted.

1 - SSH Access to the Gateway/Firewall

A number of people must access the CASA gateway/firewall system. Their tasks may include loading new OS images on the PC cluster nodes or running lab experiments. These users have been given an interactive shell account on the gateway/firewall system and can log on using SSH.

Interactive shell access via SSH gives user access to the entire CASA network. It is recommended, therefore, that SSH authentication for shell accounts use public/private keys rather than passwords – something that is easy to set up and not bothersome to the user. The SSH service on the gateway should also deny root logins. Administrators needing root access must use the `su` command after they have logged onto their own account.

In addition to the shell accounts listed above, one non-interactive account will be created to allow third parties to upload and/or download software using the SCP and SFTP utilities that are built on top of SSH. This non-interactive account will utilize password authentication.

2 - OpenVPN Access to the Gateway/Firewall

SSH can provide access to, but not through, the CASA gateway/firewall system. Many of the people who will be administering the equipment in the CASA test bed will not be on-site. So in order for them to manage systems within the CASA test bed it will be necessary to bridge or route network traffic through the gateway (that is, in order to access the web interface for BlueArc configuration). So, in addition to SSH access to the gateway PC, OpenVPN is used to create network connections from end-user workstations to the CASA management network.

Like SSH, OpenVPN is designed as a security tool. It uses strong authentication methods and encrypts all network traffic.

3 - Outgoing Connections

The CASA Gateway/Firewall system will act a as web and FTP proxy for the systems inside the CASA network, so that those systems can download necessary files, software updates, etc.

4 - Other Gateway Services

The CASA Gateway/Firewall will act as the local DHCP, DNS, and NTP server for the CASA test bed.

5 - Non-Cray Access to CASA Systems

It is sometimes necessary for a vendor or partner to gain access to a system in the CASA test bed so that they can work with Cray to resolve problems or perform testing. The goal here is to provide that partner with access to a specific system without permitting access to any other system.

This can be fairly easily done by setting up a separate vLAN with appropriate access controls on the Etherswitch. Outside access to the isolated vLAN is then granted by running a separate instance of OpenVPN on the gateway/firewall, and bridging it to the isolated vLAN.

Each instance of OpenVPN requires a new port number. The default port number for OpenVPN is 1194, but that port will be in use by the primary instance. It is recommended that a small range of unused ports (e.g. 10500..10510) be designated for use as additional OpenVPN instances. These would need to be left open on any external router that is between the CASA Gateway/Firewall and the Internet.

About the Authors

John Kaitschuck is currently a Senior Systems Engineer with Cray Federal. He has previously served in a variety of technical and consulting positions in industry and government as both analyst and developer. He has worked with a wide range of HPC issues around systems and system software. He can be reached at jkaitsch@cray.com.

James Reaney has sixteen years of experience as an IT Director and Computing/Networking/Storage Analyst with various HPC research environments. Prior to joining BlueArc Corporation, Dr. Reaney was Network and Server Operations Manager for research computing at Harvard University. His experience provides him with a solid working knowledge of research customers' day-to-day operations, the challenges faced by local IT staff, how to better meet the fiscal requirements of research administrations, and how BlueArc's products help accelerate research applications.

Matthew O'Keefe is a founder and Vice-President of Engineering at Alvarri Inc., a start-up focusing on storage management software. Previously, Matthew founded Sistina Software, sold to Red hat in late 2003; he spent 10 years as a tenured Professor at the University of Minnesota, where he is currently a Research Associate Professor. He can be reached at okeefe@alvarri.com.

Christopher R. Hertel is a well-known Open Source developer; a member of the Samba team, a founding member of the jCIFS Team, and author of *Implementing CIFS--the Common Internet File System*. For ten years designed and deployed campus-wide networks and storage networks as a network design engineer at the University of Minnesota. He is currently a storage architect with Alvarri, Inc.