Shared Object-Based Storage and the HPC Data Center

Jim Glidewell, Boeing Information Technology

ABSTRACT: *Providing a high performance storage solution is an essential requirement for most HPC data centers. This paper will focus on our experiences with the Panasas object-based storage system in a shared Cray X1 and Linux cluster environment. We will present an overview of our strategy for employing Panasas storage and the results we have obtained.*

Our HPC Environment

Boeing's Puget Sound High Performance Computing (HPC) environment consists of a Cray X1 with 128 MSPs and 1 terabyte of main memory, and a number of Linux clusters. The Cray X1's permanent storage is managed by ADIC's StorNext HSM (Hierarchical Storage Manager), in addition to direct attached temporary disk storage.

There are six Linux clusters currently: two 128 node dual-Xeon clusters, two 128 node dual-Opteron clusters, and three 256 node dual-Opteron systems. All of the systems use Myrinet as their MPI communication fabric, and all nodes have Gigabit Ethernet connections for TCP traffic.

The Linux cluster nodes share access to a single, separate ADIC StorNext system, which serves as primary permanent storage for the clusters. Individual nodes get access to the storage via a small number of NFS servers, which mount the ADIC storage directly.

Background and Current Issues

Boeing has made use of Cray/SGI's DMF (Data Migration Facility) since the early-1990s. DMF had provided us the ability to minimize the amount of expensive disk by automatically moving inactive data off to tape. Unfortunately, the option of using DMF with the Cray X1 was not available. Since our users were very comfortable and somewhat dependent on the facilities that an HSM provides, we felt it was imperative that we provide them with this functionality. We selected ADIC StorNext as the HSM for our Cray X1, as it was the only HSM that was supported directly, without going through an NFS connection, which we believed would not provide the necessary performance.

Our selection of ADIC StorNext was initially limited to the Cray X1, but we were starting to acquire Linux clusters, and we needed an HSM solution for them as well. We felt we could minimize labor and training costs by using the same HSM vendor for both Cray and clusters. We contracted Cray to provide hardware and support for the new ADIC system, so that we had a single point of contact for both the Cray and cluster ADIC systems.

Unfortunately, we radically underestimated the I/O demands that the cluster would put on our ADIC storage system, as well as the rate of growth of the Linux cluster environment. As the cluster complex has grown, we have seen continuing and worsening performance problems. This is particularly evident to our interactive users, who use the cluster head nodes for editing, pre- and post-processing, and job submission. Many of these

performance issues are linked to the fact that ADIC StorNext is both a SAN and an HSM. There are interactions between the NFS servers and the HSM which cause extended delays while a file is being retrieved from tape, as well as problems with fragmentation due to user errors or atypical application I/O write behavior.

Based on our experiences with StorNext, we came to the conclusion that expecting any vendor to provide *both* best of breed performance (which we clearly needed) *and* HSM functionality was simply expecting too much. We needed to reassess our strategy regarding storage.

Reassessing Hierarchical Storage Management

Over the years, HSMs have provided a number of benefits to our users, our admin staff, and the company as a whole. The most obvious benefit is cost reduction. HSMs make very efficient use of disk space, which has traditionally been very expensive for HPC systems. It also provides users a storage environment that appears to be "unlimited." This eliminates (or greatly reduces) the need for user disk quotas, and reduces the need for users to focus on storage management on a day to day basis. Finally, HSMs provide benefits to the system administrators by reducing the complexity of storage forecasting and planning decisions, and can provide a mechanism for making backups less time consuming.

HSMs also have downsides, however. The administration of an HSM is complex and time consuming, and requires a highly skilled administrator. User storage access patterns are not predictable, so users often see delays when retrieving files. The fact that users do not need to do regular file housekeeping leads to data building up without bounds, and serious cleanup is only addressed when a system is being retired. Moving data from one HSM to another is a very time-consuming and difficult task, requiring that all data be brought back from tape and moved into the new storage system. This problem will only get worse as disk capacities continue to outpace those of tape storage and data volume grows in general.

HSMs had served us very well, but were becoming an increasing burden for both the admin staff and our users. We needed to reassess the use of an HSM as our primary storage system.

A New Storage Strategy

Over the years, storage in our HPC datacenter has always been tightly coupled to a single computing platform, with no central common storage repository common to all systems. This strategy had not been a significant problem in the past, but had been becoming less palatable as time went on. We have seen a significant increase in the use of multiple platforms in a single engineering analysis. In our environment, this had led to significant duplication of data and increasing user dissatisfaction with the status quo. And, as previously mentioned, the performance of our existing SAN system was causing considerable frustration in our user community.

We needed to solve two problems – high performance storage for our Linux clusters, and a shared storage server for all HPC systems. After surveying the field of vendors and products, and an extended evaluation period, we selected Panasas ActiveStor storage clusters as our solution to both of these functions.

The Panasas system has been configured to provide: shared permanent storage for each

user, a home directory for each cluster user, shared HPC temporary storage, and cluster temporary storage.

Selection Criteria

In selecting a storage system for our environment, there were a number of criteria that we felt were necessary for providing the best possible service to our end users.

After our experiences with I/O performance issues on the existing SAN, performance was a central selection criterion. Since this storage server was to be used by all current and future HPC systems, and needed to service the Cray X1 today, it needed to provide good NFS performance. NFS is the only option for a server that lacks Cray X1-specific client software. For the Linux clusters, we strongly preferred a non-NFS solution for performance reasons.

Panasas met our performance expectations. The storage system is able to support a large number of concurrent active clients, provides very good single-node performance, and a high aggregate bandwidth. One very important aspect of Panasas performance is that it is scalable – adding additional storage improves the performance of the system. Further results of our performance testing will be addressed later in this paper.

Since we are expecting user applications and processes on every HPC system to be making use of the storage server at any time, it is critical that the system be available 24 hours per day, 365 days per year, with minimal downtime – both scheduled and unscheduled.

Single points of failure are minimized, storage redundancy protects against single disk failure, and the effects of a double disk failure are limited in scope by the storage hierarchy. With the release of version 3.0 of the ActiveStor software, it is possible to do concurrent software maintenance on the server without any scheduled downtime on the clients.

A final criterion for the storage system was *manageability*. Storage management, including care and feeding of the HSM systems, has been an increasing burden on our HPC staff. Since we were abandoning HSMs for primary user storage, it was essential that the system we chose would allow us to accommodate storage growth with minimal impact on us and our users. As much as possible, we were looking for a storage *appliance*, and for one which would let us expand storage with minimal effort and downtime.

When new storage hardware is brought online, it automatically attaches itself to the local Panasas *realm*, as unassigned capacity. A web-based interface allows the administrator to assign storage where it is needed. This causes an immediate increase in capacity for that group of storage volumes, and triggers an automated rebalancing of the data resident on the now-larger storage pool. The administrative interface also provides visibility of current usage, and allows the creation of volumes, quotas, etc. and overall monitoring of system health.

Panasas Architecture

Logically, Panasas presents itself as a hierarchy of realms, bladesets, and volumes.

A *realm* is the top of the hierarchy. The realm provides a single, uniform namespace for

all storage underneath it. While a site may choose to have multiple realms, the administrative tool can only administer a single realm.

A realm consists of one or more *bladesets*. A bladeset is the "double disk failure domain" - the group of drives which contain redundant copies of the data. The bladeset presents itself as a directory in the realm hierarchy.

Each bladeset contains one or more *volumes*. Volumes serve primarily as a mechanism for monitoring disk usage and enforcing soft and hard disk quotas. If quotas are not defined, the entire free space on the enclosing bladeset is available to any volume residing on that bladeset. Volumes also appear as directories, just below the containing bladeset.

Physically, Panasas comes in standard rack-mounted *shelves*. Each shelf contains 11 *blades*. The blades come in two types: *director blades* manage metadata traffic and provide NFS services, while *storage blades* contains a pair of hard drives and provide the actual storage capacity.

A typical installation will use shelves containing one director blade and eleven storage blades, but shelves can have multiple director blades or none at all. Bladesets are made up of one or more shelves, which is the minimum allocation unit for storage.

When a new unallocated shelf is presented to the Panasas realm, it can be either allocated to an existing bladeset, or used to define a new bladeset.

The Panasas system uses software, in the director and storage blades, to provide striping and redundancy of the data in the bladeset. Small files are mirrored, while larger files are striped across multiple blades along with sufficient parity data to allow reconstruction in case of a single disk failure. By relying on intelligence in the blades, rather than hardware based RAID controllers, Panasas allows for easy expansion of bladesets, with automatic data movement between old drives and new to optimize performance.

Our Panasas Configuration

Our acquisition of Panasas storage came in two phases. Initially, we brought in a three shelf evaluation system, on which we did training, familiarization, and testing. Testing included verifying conformance to established standard behavior, performance, reliability, and administrative functions.

After verifying that Panasas could meet our needs, we acquired our production storage system, which consists of 52 shelves of Panasas ActivStor 3000 storage. Each shelf consists of two director blades and nine storage blades (a "2+9" configuration). At 500 gigabytes of storage per storage blade, each shelf contains 4.5 terabytes of raw capacity, for a total of 234 terabytes of total capacity, mounted in seven standard racks.

Panasas Performance

Panasas performance is a function of multiple factors, including network speed, concurrent usage of the bladeset, number of shelves in the bladeset, and the access method. Our performance to the Cray X1 is limited primarily by the network speed to the

Cray when using NFS, and we typically see 35 megabytes per second transfer rate to the Cray.

When accessed by multiple clients, the per-shelf bandwidth is roughly 300 megabytes per second. To an individual Linux client (dual-CPU Opteron, connected via gigabit Ethernet), typical large sequential transfers can achieve up to 85 megabytes per second, which compares favorably to the EIDE or SATA typically found in a Linux cluster compute node. Our testing included a test where twenty clients read and wrote to a single four-shelf bladeset, which generated a total of 1.2 gigabytes per second total bandwidth – an average of 60 megabytes per second for each client.

Backup Issues

The growth in disk capacity and user storage requirements has put considerable strain on the ability of site administrators to keep user data properly backed up. Tape capacity growth has not kept pace with the growth in drive capacity, nor have tape transfer rates kept up with user storage growth. This makes the traditional "weekly base plus daily incremental" file system dumps impractical for many large sites.

We investigated using our corporate backup service as an alternative to handling backups ourselves, as we had always done. This idea was abandoned in the face of a number of issues. Adding the HPC backup requirements would roughly double the amount of data they were backing up, which would require significant hardware upgrades. Even with these upgrades, the time for base dumps was prohibitive. In conjunction with the enterprise backup organization, we examine the option of doing "synthetic" base dumps, which would use a previous base plus subsequent incremental dumps to create a new base dump. But this was a new and unproven option. Even with synthetic base dumps, there was still an enormous amount of tape to tape copying of unchanged data, which all concerned felt was wasteful. And finally, it was determined that in the case of a catastrophic failure of the disk storage system, recovery time could stretch significantly past a week. This was unacceptable. After discussions with Panasas, we came up with an alternative – using an HSM as our backup server.

HSM as a Backup Server

Our basic strategy was to use an HSM for what it was good for – managing data movement from disk to tape, while hiding the HSM from direct user access (and the recall delays and data churning which result from user access to a heavily overcommitted HSM). The HSM sits "behind" the Panasas storage system. It is configured with a directory hierarchy which mirrors that of the Panasas storage server. On a period basis (normally daily) each of the Panasas volumes is used as the source for a one-way synchronization from that volume to the corresponding directory on the HSM server. In the case of large volumes, synchronization can be done on a directory by directory basis. This synchronization is always disk to disk, once the data has been copied to the HSM server's disk, it can be migrated off to tape over time, and in parallel with synchronizations of other volumes. Once the data is tape resident, backups of the inodes on the HSM system can be done using a mechanism like the "xfsdump –a" option, which ignores data that is tape-resident.

Given the size of our existing Panasas system, our expected growth of that system, and the predicted daily "churn" rate (new or modified data) of two to three terabytes per day, it was important to select a system which could handle both the network traffic involved

in the synchronization step, but also had the capability of supporting the I/O required to drive a large number of tape drives. For the server, we selected an SGI Altix 450 with 16 cores, attached to a Sun/STK SL8500 tape library with six T10000 drives and 1500 tape slots.

The choice of hardware was driven to a great extent by our selection of an HSM that we were very familiar with and whose capabilities and stability we were very confident about: DMF. We are fortunate to have staff members with nearly twenty years of DMF experience on multiple platforms, so we are well aware of its strengths and weaknesses, and know the techniques for optimizing its behavior to meet our needs.

The Benefits of HSM-based Backup

We needed a backup system that could accommodate hundreds of terabytes of online storage, which is likely to grow past a petabyte. Why should we choose an HSM over a more conventional commercial backup offering? What we observed was that almost all commercial tape-based backup solutions involve a large of amount of repetitive and wasteful I/O – copying and recopying unchanged data. HSM backup offers significant advantages over these backup methods.

HSMs are designed to write data to tape *once*, then track changes to the file system in case the data in that file changes. This makes close to optimal use of tape drives. Media usage is also optimized; in that all HSM managed tapes can be written to full capacity. For files that have been deleted or modified, DMF, like many HSMs, has a "tape merge" facility which will copy files from tapes with obsolete data to new tapes. HSM systems, and DMF in particular, are a mature technology which we are very familiar with.

Finally, using an HSM as a backup server offers a fast recovery option in the case of a catastrophic failure of the primary storage system. If our primary storage system fails in such a way that recovery is not immediately possible, we can use the HSM as a direct replacement for the primary storage by mounting it on the client systems directly. As storage systems grow, and recovery times in the case of a major disaster grow also, this option becomes increasing desirable.

Summary

The Panasas ActivStor storage system is serving as the central shared storage for the Cray X1 in our HPC data center, as well as serving as primary storage for our Linux clusters. Its combination of performance, scalability, and manageability has met our needs and expectations. With our adoption of an HSM-based backup system, we feel confident that we can accommodate our user's requirements for continued data growth and sharing in the HPC environment.

About the Author

Jim Glidewell has been a member of Boeing's HPC group for over twenty years, working on a variety of systems from Cray, SGI, CDC, and others. He is currently serving as the CUG X1/E Systems SIG Chair. He can be reached at The Boeing Company, P.O. Box 3707 MC 7J-04, Seattle WA 98124-2207; E-mail: james.glidewell@boeing.com