THE USE AND INTEGRATION OF DISTRIBUTED AND OBJECT-BASED FILE-SYSTEMS AT BROOKHAVEN NATIONAL LABORATORY

R. Petkus*, M. Askinazi, D. Free, B. Gibbard, J. Lauret, Z. Liu, O. Rind,
T. Throwe, Y. Wu
Brookhaven National Laboratory, Upton, NY, USA

Abstract

The roles of centralized and distributed storage at the RHIC/USATLAS Computing Facility have been undergoing a redefinition as the size and demands of computing resources continues to expand. Traditional NFS solutions, while simple to deploy and maintain, are marred by performance and scalability issues, whereas distributed software solutions such as PROOF and rootd are application specific, non-POSIX compliant, and do not present a unified namespace.

Hardware and software-based storage offer differing philosophies with respect to administration, data access, and how I/O bottlenecks are resolved. Panasas, a clustered, load-balanced storage appliance utilizing an object-based file system, has been key in mitigating the problems inherent in NFS centralized storage. Conversely, distributed software storage implementations such as dCache and Xrootd have enabled individual compute nodes to actively participate as a unified "file server", thus allowing one to reap the benefits of inexpensive hardware without sacrificing performance.

This talk will focus on the architecture of these file servers, how they are being utilized, and the specific issues each attempt to address.

INTRODUCTION

The RHIC Computing Facility (RCF) and the ATLAS Computing Facility (ACF) at Brookhaven National Laboratory (BNL) provide a complete, robust, and uninterruptible data processing environment for collaborators at the Relativistic Heavy Ion Collider (RHIC) and the U.S. Tier 1 Center for ATLAS Computing (USATLAS), both large-scale programs devoted to new discoveries in high-energy nuclear physics.

Currently, the RCF comprises a 2000+ node dual-processor Linux farm for central analysis and reconstruction with over 680 TB of local disk partially dedicated to dCache and Xrootd, 220 TB of centralized, NFS SAN storage, 100 TB of clustered Panasas storage, and a minimal AFS deployment of 6 TB.

The ACF, ramping up for the activation of the LHC (Large Hadron Collider), has a growing deployment of 330 dual-processor compute nodes utilizing 146 TB of local disk as a dedicated dCache storage pool, 20 TB of centralized, NFS SAN storage, and an AFS repository of 1 TB [1].

As the computing potential of the RCF and ACF continues to expand with each procurement of cluster nodes, so do the demands placed on storing and making accessible tremendous amounts of data, already exceeding 3.2 PB on mass storage, in a cost-effective way.

TRADITIONAL FILE SERVERS

NFS

Providing fast, reliable storage has always been a key challenge. Almost from the outset, however, it was clear that NFS, while ubiquitous,compatible, and mature, was destined to fall prey to CPU, I/O, and network bottlenecks when connected to thousands of nodes over a fast network. NFS is difficult to scale both vertically (performance) and horizontally (management), does not effectively utilize available resources (load-balance), and offers little in the arena of security. But for lack of a more satisfactory or cost-competitive solution, each NFS file server, typically a Solaris v240 server attached to a SAN back-end of fibre-channel disks in RAID 5 arrays, was able to achieve sustained network throughput of 70-80 MB/sec using a single 1 Gb NIC. Although this transfer rate may be sufficient for many applications, more frequently it equates to severe client-side latency and wasted CPU cycles spent in an I/O-wait state when a filesystem is in high demand.

AFS

AFS usage at the RCF/ACF occupies a completely different storage niche than NFS. AFS performs most ably when used as repository for static data (software, web content) and is well-suited for secure WAN access. Read-only data can be replicated across many AFS fileservers to increase availability, redundancy, and read performance. The caveat, however, is that AFS is not particularly fast, especially during a write operation, which can not take advantage of replication. Therefore, AFS is not suitable as a container for dynamic data and is unable to satisfy all mass storage needs.

NEXT GENERATION FILE SERVERS

Criteria

When evaluating a new file server, what features constitute a better file server, and should all criterion be given equal weight for each experiment? Is it reasonable to expect that one file server alone can meet all current needs universally? At a minimum, a competitive offering would have to be fast, scalable, reliable, and fault-tolerant. And fundamental niceties of a next generation file server need to address load-balancing, built-in security, centralized management, and transparent, POSIX access to a single, global namespace.

To this end, three new file servers, each embracing a different implementation philosophy, have been put into production at the RCF/ACF to augment and offer alternatives to traditional storage. These file servers are Panasas, dCache, and Xrootd.

PANASAS – OBJECT-BASED STORAGE

Panasas offers an integrated hardware/software high-performance storage solution. Panasas is sold as a "shelf" which is a bladed server system consisting of at least 1 "Director" blade and a maximum of 10 "Storage" blades for a total of 11 blades. Each shelf has a maximum raw storage capacity of 8 TB with 4 Gb/sec of external connectivity via 4 bonded network interfaces.

Storage blades, or Object Storage Devices (OSDs), simply store and retrieve data objects. Objects, which can be arbitrary

*rpetkus@bnl.gov

in size, encapsulate data along with attributes like RAID attributes, ownership, etc. The ActiveScale operating system, run on the Director blade, coordinates between the Linux client and the OSDs, manages metadata, distributes workload across OSDs, and slices files into data objects. ActiveScale provides to the Linux client, via a "DirectFlow" kernel module, the PanFS global namespace. All components within the Panasas framework are redundant and provide for on-line fail-over.

When a Linux client requests a file from PanFS, the Director validates the request, and grants a token along with a map of which OSDs contain the data. The client then retrieves the data directly and in-parallel from each OSD.

There are 20 Panasas shelves at the RCF totaling 100 TB of raw storage. Each shelf at the RCF utilizes 2 bonded network interfaces equaling 2 Gb/sec sustained throughput per 5 TB of data. Shelves are organized into a logical unit named a "Bladeset" composed of volumes. Bladesets become members of a realm, a global namespace similar to an AFS cell.

Unfortunately, in practice Panasas has proven itself to be far less worthy of praise. Frequent client-side crashes, hardware failures, and data corruption sully what would otherwise be a favorable experience. We've also discovered that Panasas volumes approaching 100% capacity quickly become unstable resulting in offline file systems. A remedy, proposed by Panasas, is to set volume quotas at 95% of usable capacity as a protective measure. Ultimately, this merely diminishes usable capacity even further.

In short, Panasas is indeed fast but this glossy is undermined by software instabilities that prevent it from gaining further traction at the RCF/ACF.

dCache

dCache is a distributed disk caching system that allows transparent client access to heterogeneous storage (mass storage and local disk). dCache provides load-balancing by detecting "Hot Spots", heavily accessed data, and dynamically replicating this data onto a less utilized storage pool node. The dCache administrator is able to define what cost metrics (cpu load, disk space, configuration) will determine how, when, and where data replication will occur. Furthermore, dCache is secure, scalable, redundant, and fault tolerant.

Both the RCF and ACF have implemented a hybrid deployment of dCache wherein compute nodes also act as read pool nodes. The PHENIX experiment at RHIC has 25 TB of local storage on 128 compute nodes allocated for dCache while USATLAS has 146 TB of local disk reserved on 322 compute nodes.

dCache provides multiple and redundant points of data access called "doors". Doors exist for local POSIX-like access (dCAP), secure WAN access (GsiFTP), site-specific storage protocol negotiation (SRM), and even Xrootd [2].

General UNIX filesystem access is made available using PNFS, a metadata database exported as a NFS filesystem. Note that PNFS only stores metadata, not the actual files. Although rare, a scenario exists where a PNFS server crash during file transfer could result in improperly registered or invalid files being written into the system requiring a manual cleanup of residual or orphaned data.

In practice, dCache is a well-regarded, high performance storage element at the RCF and especially USATLAS. Peak daily data transfer rates at PHENIX exceed 50 TB and 60K files per day. During the LHC Service Challenge in January 2006, the disk-disk transfer rate from CERN to USATLAS was sustained at 90 MB/sec [3].

A limitation of dCache is that it only offers read or write-only file operations. Since a file can not be opened for both read/write access, dCache is not a suitable replacement in certain domains where NFS excels such as providing home directories and scratch space. Another limitation is that there are no mechanisms or policies available to throttle user activity such as a quota system.

Xrootd

Xrootd offers high-performance, security, multiplexed data access, load-balancing, and a global namespace. Load-balancing is achieved via open load balancers (OLBs), which determine which server is best suited to serve data to a particular client [4].

There is no single point of failure in Xrootd's architecture. Each element in the system is completely redundant and fault-tolerant. The components of Xrootd are 1) OLB Managers and Servers, which monitor load, data availability, and a data location map, 2) Redirectors, which act as traffic managers directing clients to data, 3) Data Servers, which retrieve a master copy of data from mass storage for use by clients, and 4) the Data Clients, which have transparent access to data [5].

The STAR experiment at RHIC utilizes a hybrid model of Xrootd wherein compute nodes also act as Data Servers. With 650 Xrootd Data Servers serving 132 TB of local storage, STAR has the largest Xrootd deployment in the world.

However, unlike dCache, Xrootd is limited in its data access repertoire, although the development of a SRM, middleware which can invoke a transfer service such as GridFTP, for use with Xrootd seeks to address this. Xrootd always copies data from mass storage but not from another file server [4]. There is no access via a general UNIX file system-like interface such as with PNFS, so the user is denied the perhaps psychological convenience of perusing a familiar file system tree. Other limitations include read or write-only access and the inability to manage usage based on quotas or transfer limits.

CONCLUSION

Although no single implementation is a panacea providing all solutions, the following trend is clear – storage will continue to move toward a distributed rather than centralized model. The dCache and Xrootd deployment at the RCF and ACF will be further expanded while use of NFS will diminish as less investment is made to enhance and update that infrastructure. In the interim, candidate file servers will continue to be evaluated and revisited. Promising candidates include GPFS, NFSv4.1, ZFS, and Lustre.

REFERENCES

[1] A. Withers, "BNL Site Report", HEPIX, SLAC, USA, Fall 2005.

[2] P. Fuhrmann, "dCache, the Upgrade", CHEP06, Mumbai, India, Feb. 2006.

[3] Z. Liu, "Large Scale, Grid-Enabled, Distributed Disk Storage Systems at Brookhaven National Lab RHIC/ATLAS Computing Facility", CHEP06, Mumbai, India, Feb. 2006.

[4] J. Lauret, et. al., "From rootd to xrootd, from PFN to LFN", CHEP06, Mumbai, India, Feb. 2006.

[5] A. Hanushevsky and H. Stockinger, "A Proxy Service for the Xrootd Data Server", CERN, 2005.