



*Journal of Advances in
Science and Technology*

*Vol. VI, Issue No. XI,
November-2013, ISSN
2230-9659*

AN ANALYSIS ON TECHNICAL SCALABILITY AND MANAGEABILITY OF STORAGE SYSTEMS

AN
INTERNATIONALLY
INDEXED PEER
REVIEWED &
REFEREED JOURNAL

An Analysis on Technical Scalability and Manageability of Storage Systems

Dr. Shailendra Singh Sikarwar¹ Mahesh Bansal²

¹Assistant Professor, P. G. V. College, Gwalior

²Assistant Professor, P. G. V. College, Gwalior

Abstract – This paper presents a deterministic placement approach for distributing orthogonal redundancy on distributed heterogeneous disk arrays, which is able to adapt on-line the storage system to the capacity/performance demands by only moving a fraction of data layout.

The evaluation reveals that our proposal achieve data layouts delivering an improved performance and increased capacity while keeping the effectiveness of the redundancy scheme even after several migrations. Finally, it keeps the complexity of the data management at an acceptable level. This paper describes the performance and manageability of scalable storage systems based on Object Storage Devices (OSD). Object-based storage was invented to provide scalable performance as the storage cluster scales in size. For example, in our large file tests a 10-OSD system provided 325 MB/sec read bandwidth to 5 clients (from disk), and a 299-OSD system provided 10,334 MB/sec read bandwidth to 151 clients. This shows linear scaling of 30x speedup with 30x more client demand and 30x more storage resources. However, the system must not become more difficult to manage as it grows.

This architecture has several advantages. A file's physical location is decoupled from its location in the namespace. This decoupling enables a powerful and flexible mechanism for the placement of file system objects. For example, different types of files, e.g., text or video, may reside anywhere in the namespace while being hosted by servers best suited to handling their content type. DiFFS also provides lightweight protocols for online dynamic reconfiguration (volume reassignment and object migration) to address fluctuating demand and potentially mobile file system entities. A DiFFS prototype has been implemented in Linux. Performance results indicate that the architecture achieves its flexibility and scalability goals without sacrificing performance.

INTRODUCTION

Delivering improved performance I/O, increased capacity and strong data availability, have been the traditional aims when designing storage systems. Scalability of storage systems is as important, or even more important to current applications than the aims above-mentioned mainly because of the constant growth of new data (at an annual rate of 30% and even 50% for several applications) and the rapid decline in the cost of storage per GByte, which have led to an increased interest in storage systems able to upgrade their capacity online.

Since applications such as scientific or database are particularly sensitive to storage performance because of their imposing I/O requirements (in some cases can account for between 20 to 40 percent of total execution time), they are requiring storage systems capable of upgrading their bandwidth in order to deliver fast service times.

The large-scale adoption of the Internet as a means of personal communication, societal interaction, and a successful venue for conducting business has raised the need for applications and services that can support Internet-scale communities of users. Cloud computing has recently offered the infrastructural support necessary for small, medium and large enterprises alike to deploy such services. However both application developers and Cloud computing providers are in need of the infrastructural services and platforms that can support the scalability requirements of distributed applications. Compounded with the need for scalability is the need for rapid prototyping. Today's planet-scale social networks such as Facebook, and application marketplaces such as Apple Store have brought applications (and the "garage innovator" behind them) closer to large communities of users. This trend has fueled a race among developers to bring new ideas to market as soon as possible without sacrificing scalability and availability along the way. The combination of the above needs (scalability and rapid

prototyping of application-specific designs) was certainly a challenging undertaking a decade ago. As the core theme of this paper indicates however, recent advances in scalable infrastructure services have improved on the state of the art.

THE IDEAL STORAGE SYSTEM

The ideal scalable storage system is a large, seamless storage pool that grows incrementally without performance degradation and is shared uniformly by all clients of the system under a common access control scheme. As the system scales in size, however, issues arise in two general areas: traditional storage management and internal resource management. Both of these areas are affected by the distributed system implementation of the storage system itself. To external clients, the storage system should feel like one large, high-performance system with essentially no physical boundaries imposed by the implementation. Internally, the system must manage a large and growing collection of computing and storage resources and shield the administrator from chore of administering individual resources. Ideally, a distributed file service should scale arbitrarily, with new resources adding to system capabilities without diminishing marginal returns. Also, the service should have simple mechanisms for aggregation and (re)configuration of system resources. To meet these goals, DiFFS takes a partitioning approach to resource sharing, by dividing resources into storage partitions. Resources in a partition are controlled exclusively by a single partition server.

Storage within a partition is divided into volumes. Each volume contains a single physical file system. Another basic design principle of the architecture is to decouple the names of file and directory objects (location in some namespace) from the objects' actual physical location (volume, partition). The aim is to allow for maximum flexibility of placing objects across the resources available in the system, while presenting a virtual file space to the clients. Files and directories can be placed arbitrarily on any volume and in any partition in the system.

BACKGROUND

The storage subsystems and tape libraries are typically connected to switch ports and provide block storage access in the form of volumes to the hosts. The admissibility of particular data paths is vetted using access control provided by storage controller known as masking and mapping. This section contains The historical importance of this drive is that it was the first to support real time online transaction processing, a capability which revolutionized the computer industry. A magnetic disk provides rapid data access, a feature that necessitated a finite separation between the head and the medium to avoid wear at the high surface speeds required.

Noncontact data recording, being a major departure from the typical tape recording of analog signals at that time, generated interest in exploring unique head designs. In this section describe the details of The central processing unit (CPU) is the portion of a computer system that carries out the instructions of a computer program, to perform the basic arithmetical, logical, and input/output operations of the system.

This section is intended to give a general introduction to the LIN. It's a compilation of informations from the lin specifications. It describes the features and highlights the main advantages of this communication bus. In this Sections contains a quick refresher on RAID is an acronym for Redundant Array of Independent Disks. With RAID enabled on a storage system you can connect two or more drives in the system so that they act like one big fast drive or set them up so that one drive in the system is used to automatically and instantaneously duplicate (or mirror) your data for real-time backup.

NETWORK-ATTACHED STORAGE DEVICES

Network attached storage systems must provide highly available access to data while maintaining high performance, easy management, and maximum scalability.

The traditional storage solution has typically been direct attached storage (DAS) where the actual disk hardware is directly connected to the application server through high-speed channels such as SCSI or IDE. With the proliferation of local area networks, the use of network file servers has increased, leading to the development of several distributed file systems that make the local server DAS file system visible to other machines on the network. These include AFS/Coda, NFS, Sprite, CIFS, amongst others. The desire to increase the performance and simplify the administration of these file servers has led to the development of dedicated machines known as network attached storage (NAS) appliances by companies such as Network Appliance, Auspex, and EMC. In addition to specialized file systems, these NAS appliances are also characterized by specialized hardware components to address scalability and reliability. In an effort to remove the bottleneck of the single server model of NAS servers, there has lately been significant work in the area of distributed or clustered storage systems.

Networked storage also simplifies storage management by centralizing storage under a consolidated manager interface that is increasingly Web based, storage-specific, and easy to use. Inherent availability, at least in systems in which all components are provided by the same or cooperating vendors, is improved, because all hardware and software in a networked storage system is specifically developed and tested to run together.

Most networked storage systems fall into one of two technology families: NAS systems (such as the F700 file-server systems from Network Appliance of Sunnyvale, Calif.) typically accessed via Ethernet networks; and SAN systems (such as the Symmetrix disk array from EMC of Hopkinton, Mass.) typically accessed via Fibre Channel networks. Both NAS and SAN storage architectures provide consolidation, rapid deployment, central management, more convenient backup, high availability, and, to varying degrees, data sharing. It is therefore no surprise that an IT executive might view NAS and SAN as solutions to the same problems and the selection of networking infrastructure as the principle differentiator among them. The technology trends we discuss here are likely to blur this simplistic, network-centric differentiation between NAS and SAN, so we begin with the principle technological difference.

SHARED STORAGE ARRAYS

In such systems, clients perform access tasks (read and write) and management tasks (storage migration and reconstruction of data on failed devices). Each task translates into multiple phases of low-level device I/Os, so that concurrent clients accessing shared devices can corrupt redundancy codes and cause hosts to read inconsistent data. This paper is devoted to the problem of ensuring correctness in a shared storage array. The challenge is guaranteeing correctness without compromising scalability.

Traditional I/O subsystems, such as RAID arrays, use a single centralized component to coordinate access to storage when the system includes multiple storage devices. A single storage controller receives an application's read and write requests and coordinates them so that applications see the appearance of a single shared disk. In addition to performing storage access on behalf of clients, the storage controller also performs other "management" tasks. Storage management tasks include migrating data to balance load or utilize new devices, adapting storage representation to access pattern, backup, and the reconstruction of data on failed devices. One of the major limitations of today's I/O subsystems is their limited scalability caused by shared controllers that data must pass through, typically from server to RAID controller, and from RAID controller to device. Emerging shared, network-attached storage arrays, enhance scalability by eliminating the shared controllers and enable direct host access to potentially thousands of storage devices over cost-effective switched networks. In these systems, each host acts as the storage controller on behalf of the applications running on it, achieving scalable storage access bandwidths.

Unfortunately, such shared storage arrays lack a central point to effect coordination. Because data is

striped across several devices and often stored redundantly, a single logical I/O operation initiated by an application may involve sending requests to several devices. Unless proper concurrency control provisions are taken, these I/Os can become interleaved so that hosts see inconsistent data or corrupt the redundancy codes. These consistencies can occur even if the application processes running on the hosts are participating in an application-level concurrency control protocol, because storage systems can impose hidden relationships among the data they store, such as shared parity blocks.

CONCLUSION

The ability of storage systems built on the Object Storage Architecture to scale capacity and performance addresses a key requirement for HPC Linux clusters. Panasas' Objectbased storage cluster demonstrates scalability with 32-shelf systems providing 30x the bandwidth of a single shelf, and 30 shelf NAS benchmarks providing 6x the throughput of 5-shelf runs of the same benchmark. While we want performance and capacity to grow linearly as resources are added to a storage cluster, we do not want administrator effort to grow anywhere near linearly. Object Storage Architectures are designed to abstract physical limitations, making virtualization easier to provide, so that larger systems can be managed with little more effort than small systems. Panasas object-based storage clusters use distributed intelligence, a single namespace interface, file-level striping and RAID, and transparent rebalancing to realize the manageability advantages of Object-based Storage.

Shared storage systems are facing a new era as the network hardware takes a new leap in bandwidth. The old trusty, centralized NAS storage will have to step aside for new scalable, distributed concepts. The surface starts to crack when the NAS is exposed to demanding multi-user environments and bandwidths that exceeds the current 1 GbE standard. SAN systems on the other hand, has shown that the light protocols and distributed coordination makes it possible to achieve an affordable shared storage system that is much faster than before.

REFERENCES

- Amiri, K., Gibson, G., and Golding, R. (2000). Highly concurrent shared storage. In Proceedings of the 20th International Conference on Distributed Computing Systems, Taipei, Taiwan, Republic of china.
- Arpaci-Dusseau, R. H., Anderson, E., Treuhaft, N., Culler, D. E., Hellerstein, J. M., Patterson, D., and Yelick, K. (2007). Cluster I/O with River: Making the fast case common.

In Proceedings of the Sixth Workshop on Input/Output in Parallel and Distributed Systems, pages 10-22, Atlanta, GA.

- Cobalt Networks (2005). Cobalt networks delivers network-attached storage solution with the new NASRaQ. Press release, <http://www.cobaltnet.com/company/press/-press32.html>.
- Gibson, G. A., Nagle, D. F., Courtright, W., Lanza, N., Mazaitis, P., Unangst, M., and Zelenka, J. (2002). NASD scalable storage systems. In Proceedings of the USENIX '99 Extreme Linux Workshop, Monterey, CA.
- Golding, R., Shriver, E., Sullivan, T., and Wilkes, J. (2005). Attribute- managed storage. In Workshop on Modeling and Specification of I/O, San Antonio, TX.
- Hartman, J. H., Murdock, I., and Spalink, T. (2006). The Swarm scalable storage system. In Proceedings of the 19th IEEE International Conference on Distributed Computing Systems.
- In Proceedings of the 14th ACM Symposium on Operating Systems Principles, pages 1-14.
- Rambus (2000). Rambus for small, high performance memory systems. http://www.rambus.com/dc/clopccr/do/nloads/Valuc_Proposition_Nctworking.html.