Journal of Network and Computer Applications 32 (2009) 1064–1072

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Deconstructing Network Attached Storage systems

Yuhui Deng *

EMC Research China, Beijing 100084, PR China

ARTICLE INFO

Article history:
Received 8 April 2008
Received in revised form
17 November 2008
Accepted 26 February 2009

Keywords:
Network Attached Storage
Performance bottleneck
Storage stack
Network stack
RAID

ABSTRACT

Network Attached Storage (NAS) has been gaining general acceptance, because it can be managed easily and files shared among many clients, which run different operating systems. The advent of Gigabit Ethernet and high speed transport protocols further facilitates the wide adoption of NAS. A distinct feature of NAS is that NAS involves both network I/O and file I/O. This paper analyzes the layered architecture of a typical NAS and the data flow, which travels through the layers. Several benchmarks are employed to explore the overhead involved in the layered NAS architecture and to identify system bottlenecks. The test results indicate that a Gigabit network is the system bottleneck due to the performance disparity between the storage stack and the network stack. The tests also demonstrate that the performance of NAS has lagged far behind that of the local storage subsystem, and the CPU utilization is not as high as imagined. The analysis in this paper gives three implications for the NAS, which adopts a Gigabit network: (1) The most effective method to alleviate the network bottleneck is increasing the physical network bandwidth or improving the utilization of network. For example, a more efficient network file system could boost the NAS performance. (2) It is unnecessary to employ specific hardware to increase the performance of the storage subsystem or the efficiency of the network stack because the hardware cannot contribute to the overall performance improvement. On the contrary, the hardware methods could have side effect on the throughput due to the small file accesses in NAS. (3) Adding more disk drives to an NAS when the aggregate performance reaches the saturation point can only contribute to storage capacity, but not performance. This paper aims to guide NAS designers or administrators to better understand and achieve a cost-effective NAS.

1. Introduction

With the explosive growth of the Internet and its increasingly important role in our daily lives, traffic on the Internet is increasing dramatically, more than doubling every year. Under such circumstances, the enormous amount of digital data is identified as the key drivers to escalate storage requirements including capacity, performance, and quality of service. Due to the explosive data growth, three typical storage system architectures have been adopted to satisfy the increasing storage requirements, namely Direct Attached Storage (DAS), Storage Area Network (SAN), and Network Attached Storage (NAS) (Deng et al., 2005, 2006; Gibson and Meter, 2000; Mesnier et al., 2003; Patterson et al., 1988; Riedel, 2003; Varki et al., 2004). Different implementation of each of the storage architectures has a different effect on the overall system performance and application circumstances. DAS cannot be scaled because the number and type of storage devices that can be attached to a server are limited. The server could also quickly become a major system bottleneck due to store-and-forward data copying between the server and the attached storage devices (Deng et al., 2005). The complex administration and cost are indicated by many researches as the key barriers to adopt SAN solutions. Service-oriented storage (Chuang and Sirbu, 2000) and storage grid (Deng et al., 2008a; Deng and Wang, 2007) are also emerging as new storage system architectures to tackle the large-scale, geographically distributed storage demands. However, it is still in its infancy.

NAS has recently been gaining general acceptance, because it can be managed easily and files shared among many clients that run different operating systems. It also offers some other advantages such as parallel I/O, incremental scalability and lower operating costs (Sohan and Hand, 2005). The advent of Gigabit Ethernet and high speed transport protocols further facilitates the adoption of NAS. Recently, the community is very active in designing a large storage system that involves multiple NAS nodes, while providing scalability and simplified storage management. X-NAS (Yasuda et al., 2003) is a highly scalable and distributed file system designed to virtualize multiple NAS systems into a single file system view for different kinds of clients. NAS switch (Katsurashima et al., 2003) is proposed and designed as an in-band switch between clients and NAS systems to provide a single virtual NAS system for users and

* Tel.: +86 10 86108216.
E-mail addresses: deng_derek@emc.com, yuhuid@hotmail.com (Y. Deng).

1084-8045/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2009.02.006

administrators. Bright and Chandy (2003) designed a clustered storage system with a unified file system image across multiple NAS nodes. The system provides simplified storage management and scalable performance. CoStore (Chen et al., 2003) is a serverless storage cluster architecture that evenly distributes system responsibilities across all collaborating NAS nodes without a separate central file manager. The architecture has the potential to provide scalable performance and storage capacity with strong reliability and availability. RISC (Deng, 2008) divides storage nodes into multiple partitions to facilitate the data access locality. Multiple spare links between any two storage nodes are employed to offer strong resilience to reduce the impact of the failures of links, switches, and storage nodes. The scalability is guaranteed by plugging in additional switches and storage nodes without reconfiguring the overall system. Two I/O-aware load-balancing methods are proposed in Qin et al. (2005a) to improve the overall performance of a cluster system running I/O intensive applications. The proposed approaches dynamically identify I/O load imbalance on nodes of a cluster, and determine whether to migrate some I/O load from overloaded nodes to other underloaded nodes to alleviate the system bottleneck. Traditional load-balancing approaches normally employ static configuration for the weights of resources. These methods cannot be adjusted automatically for the dynamic workload. A feedback control mechanism is proposed to improve overall performance of a cluster with I/O-intensive and memory-intensive workload (Qin et al., 2005b). However, the aggregate performance of a NAS cluster can be dominated by the performance of a single NAS node.

A lot of research efforts have been invested in designing and developing performance optimization methods, which may be able to improve the performance of NAS. TCP offload is proposed to offload the entire TCP/IP stack to the network adapter. The method can reduce the overheads of interrupts and TCP copy-and-checksum (Mogul, 2003; Sarkar et al., 2003). Remote Direct Memory Access (RDMA) moves data directly from the memory of one computer into the memory of another computer without involving the operating systems at both sides. Zero copy is used to directly transfer data between the application memory and network adapter. VI Architecture (Compaq, Intel, Microsoft, 1997) is a user-level memory mapped architecture. By avoiding the kernel involvement, the architecture is designed to eliminate the software overhead imposed by traditional communication models, thus achieving low latency and high bandwidth. DAFS (Magoutis et al., 2002) is a network file system that allows applications to transfer data while bypassing the potential performance bottlenecks such as operating system control, buffering, network protocol, etc. DAFS works with any interconnection that supports Virtual Interface (VI) including Fibre Channel and Ethernet. Brustoloni (1999) investigated a solution that allows data to be passed between networks and file systems without copying and without changing the existing interfaces. Sohan and Hand (2005) showed that the current NAS architecture performs poorly mainly because of multiple data copies, poor kernel resource accounting, inadequate process isolation and poor application customisation facilities. They proposed and designed a user-level NAS architecture that does all file and buffer layer processing in user space without any specific hardware support. The experimental results illustrate that this architecture produces better performance than the traditional architecture. Because the traditional NAS adopts general computer system architecture, NAS may also benefit from other generic performance enhancing methods such as intelligent scheduling, caching, and prefetching.

A distinct feature of NAS is that NAS involves both network I/O and file I/O. Gigabit Ethernet further propels the wide adoption of NAS. It seems that the performance requirements of providing storage over a network should not be a problem due to the involved Gigabit network. Modern operating systems tend to be structured in distinct, self-contained layers, with little inter-layer state or data sharing. Communication between layers uses well defined interfaces that are difficult to circumvent (Sohan and Hand, 2005). An NAS normally consists of several key layers including storage devices, logical volume manager, local disk file system, and network file system. Each layer in the system takes its cut off performance from the layer below. Each layer has an amount of overhead that reduces the performance, which is available to the next higher layer of the system (Riedel, 2003). Due to the complexity of NAS system, how to identify the system bottlenecks and employ the most effective approach to eliminate the bottlenecks and boost the performance are challenging problems.

In this paper, we discuss the system architecture of a typical NAS and isolate some key components within the NAS by tracking the data flow. We use a variety of benchmarks to explore the overhead involved in the layered NAS architecture and identify the system bottlenecks, because we only need to speed a small portion (system bottleneck) of the overall system to achieve the maximum performance gains in terms of Amdahl's law. The key contribution of this paper is to identify some cost-effective methods that can boost the performance of NAS from a large number of optimization methods.

The remainder of the paper is organized as follows. Section 2 introduces the architecture and the data flow of a typical NAS. The testbed and performance measurements are depicted in Section 3 in detail. There are some discussions of the work and the indications of future research in Section 4. Section 5 concludes the paper with remarks on the contributions of the paper.

2. System overview

The hierarchy of storage in current computer architectures is designed to take advantage of data access locality to improve overall performance. Each level of the hierarchy has higher speed, lower latency, and smaller size than lower levels. For decades, the hierarchical arrangement has suffered from significant bandwidth, latency, and cost gaps between the RAM and disk drive (Pugh, 1971). The performance gap has been widened to six orders of magnitude in 2000 and continues to widen by about 50% per year (Schlosser et al., 2000). The disk I/O subsystem is repeatedly identified as a major bottleneck to system performance in many computing systems. The bottleneck of disk I/O could significantly impact the application performance. NAS employs RAID subsystems, which adopt multiple disk drives working in parallel to achieve high performance, thus alleviating or even eliminating the I/O bottleneck. The performance of the storage subsystem scales with the number of disk drives. The performance bottleneck of a NAS could be migrated from disk I/O to other components with the increase in number of disk drives.

2.1. Architecture overview of NAS

Currently, there are two popular data sharing mechanisms including network file system and an iSCSI. The two mechanisms are fundamentally different. Network file system enables files to be shared among multiple client machines. iSCSI is a block level protocol that encapsulates SCSI commands into TCP/IP packets and transfers the data through TCP/IP network. iSCSI permits applications running on a single client machine to share remote data, but it is not directly suitable for sharing data across multiple machines (Radkov et al., 2004). An NAS is a file server that normally presents a file interface to the network by employing