IP Storage: The Challenge Ahead

Prasenjit Sarkar, Kaladhar Voruganti
IBM Almaden Research Center
San Jose, CA 95120
{psarkar,kaladhar}@almaden.ibm.com
tel +1-408-927-1417
fax +1-408-927-3497

Abstract

Advanced networking technology has led to the genesis of the storage area network model, where host servers can access storage as a service from various devices connected to the network. While the initial approach to storage area networks has involved specialized networking technology, the emergence of Gigabit Ethernet technology has raised the question of whether we can use commodity IP networks for storage. This paper examines the issues involving IP storage networks and presents a performance analysis to dispel some of the myths and outline some of the challenges.

1 Introduction

With the steady increase in the storage needs of most organizations, block storage management is becoming an important storage management problem. Application servers, databases and file systems ultimately rely on the presence of an efficient and scalable block storage management system.

In the past, the storage model assumed the presence of storage attached to every host server. This type of host server-attached storage relied on the Small Computer System Interface (SCSI) protocol. The SCSI protocol emerged as the predominant one inside host servers due to its clean, well-standardized message-based interface. Moreover, in later years, it supported command queuing at the storage devices and allowed for overlapping commands. In particular, since the storage was local to the server, the preferred SCSI transport used was Parallel SCSI where multiple storage devices were connected to the host server using cable-based bus. However, as the need for storage and servers grew, the limitations of this technology became obvious. First, the use of parallel cables limits the number of storage devices and the distance of the storage devices from the host server. The limits imply that adding storage devices might mean the need to purchase a host server for attaching the storage. Second, the concept of attaching storage to every host server means that the storage had to be managed on a per-host server basis, a costly implication for centers with a large number of host servers. Finally, the technology does not allow for an easy sharing of storage between host servers, nor typically does the technology allow for easy addition or removal of storage without host server downtime.

The lack of scalability and manageability of the host server-attached storage model led to the evolution of the concept of a storage area network. Storage devices are assumed to be independent machines that provide storage service via a network to a multitude of host servers. The attraction of this approach is that host servers can share a pool of storage devices leading to easier storage administration. The advent of networking infrastructure capable of gigabit speeds further facilitates the service of storage over the network.

Furthermore, storage can be added, removed or upgraded without causing any host server downtime. In addition, the distance limitation of the host server-attached storage model is also removed.

Approaches to storage area networks have involved specialized technology such as HIPPI, VaxClusters, Fibre Channel and Infiniband [3][6][7]. The motivation behind the design is to construct a network that meets all the performance and connectivity requirements of a storage area network. The downside to these storage area networks is the requirement to purchase specialized adapters, switches and wiring for equipping the network. Furthermore, since storage area networks are not expected to be very high-volume, the cost of these components tends to be on the higher side in comparison to commodity Ethernet networks. Finally, all these specialized networks have very limited support for wide area networking and security. In fact, accessing such specialized storage area networks over long distances requires an IP network bridge.

The question then arises – is it possible to transport the SCSI storage protocol over commodity Ethernet IP networks [2] and still satisfy the performance requirements of storage area networks?

The advantages of IP networks are obvious. The presence of well tested and established protocols such as TCP/IP allow IP networks both wide-area connectivity as well as proven bandwidth sharing capabilities. Furthermore, the emergence of Gigabit Ethernet and the future arrival of 10 Gigabit Ethernet seems to indicate that the bandwidth requirements of serving storage over a network should not be an issue [1]. Finally, the commodity availability of IP networking infrastructure indicates the cost of building a storage area network will not be prohibitive.

This paper examines the issues involved in developing a high performance storage area networking solution. We present a performance analysis of a software-based IP Storage Area network. First, we measure the latency of block transfers to show that the protocol overhead of TCP/IP is minimal. Second, we do throughput measurements to show that while it is theoretically possible to saturate a Gigabit Ethernet network but that the CPU utilization is high compared to that in specialized storage area networks. We conclude this paper with an assessment of various hardware and software techniques that can help obtain high bandwidth at low CPU utilizations.

2 IP Storage

With the steady increase in the storage needs of most organizations, block storage management is becoming an important storage management problem. Both databases as well as file systems ultimately rely on the presence of an efficient and scalable block storage management system. The Small Computer System Interface (SCSI), rather than Advanced Technology Attachment (ATA), is the block management protocol of choice for most storage area network solutions because it supports command queuing at the storage devices and allows for overlapping commands. The SCSI protocol is mostly implemented over the parallel SCSI cable technology where multiple storage devices are connected to a SCSI bus via a cable. Though parallel SCSI technology supports gigabit network speeds, the distance (few meters) and the connectivity limitations (16 devices to a channel) are hampering its acceptance as the gigabit networking transport layer of choice for the emerging large storage area networks. In addition, the parallel SCSI technology is more suited to attach to a specific host rather than being available as a network service which can be managed separately. Thus, specialized networking protocols such as Fibre Channel [3] and Infiniband [5] have been developed to overcome these limitations while still providing network-attached block storage at gigabit speeds.

The Fibre Channel protocol covers the physical, link, network and transport layers of the OSI network stack. Fibre Channel provides support for many different service classes. The Fibre Channel protocol contains a SCSI over Fibre Channel definition called FCP. The FCP protocol optimizes data transfer by enabling zero-copy transfers to the receiving host and reduces buffering requirements by making every frame self-describing. The FCP protocol also contains a simple and conservative flow control mechanism.

The Infiniband protocol also covers the physical, link, network and transport layers of the OSI network stack. The Infinband protocol provides support for many different service classes like Fibre Channel. In addition, the Infiniband protocol provides the QueuePair programming abstraction that allows application programs to transfer data directly from the network card into the application. The protocol provides the notion of verbs that allows application programs to send and receive data. The Infiniband protocol is similar to Fibre Channel in that it also supports a simple and conservative flow control mechanism.

Storage over IP is currently driven primarily by the iSCSI protocol [4] that defines the operation of SCSI over TCP and tries to leverage the existing TCP over IP over Gigabit Ethernet infrastructure. The goal of iSCSI is to leverage TCP flow control, congestion control, segmentation mechanisms, and build upon the IP addressing and discovery mechanisms to create a seamless and scalable storage area network. iSCSI can be implemented as a combination of network adapter card with the TCP/IP and iSCSI layers in software. This approach has the appeal of benefiting from the commodity appeal of existing network adapters and switches, an important factor in lowering infrastructure costs.

The challenges of building a storage area network over IP are not trivial. Detractors of IP storage area networks point out that the overhead of using TCP is prohibitive enough to result in poor latency for transaction-oriented benchmarks. It is also pointed out that common network application programming interfaces such as sockets do not allow for zero-copy transmits and receives of data leading to the overhead of multiple data copying [5]. Such data copying is considered harmful for overall throughput and will affect bulk-data scientific and video applications. Finally, data is transferred from the network adapter to the host machine using frame-size transfers. This means that every bulk data transfer may involve multiple interrupts instead of at most one interrupt in the case of specialized storage networks. Consequently, the interrupt overhead can be the limiting factor in peak throughput if the storage device or host server CPU spends the majority of its cycles processing interrupts.

3 Performance Analysis

We present a performance evaluation of a software implementation of IP storage and point out the performance characteristics that meet the requirements of storage area networks and those that do not. Our test-bed aims to determine the latency and throughput characteristics of a host server connected to a storage device over a Gigabit Ethernet network.

We use the iSCSI protocol [4] to transfer SCSI blocks between the storage device and the host server. The iSCSI protocol is a standard for transporting SCSI blocks over TCP/IP and is expected to be an IETF standard by early 2002. The key features of the iSCSI protocol are:

- Explicit login with the option to negotiate features such as security
- Authentication using SRP and other optional algorithms
- Trunking using multiple TCP/IP connections between storage endpoints
- Digests using CRC-32C and other optional schemes
- Encryption using IPSEC based algorithms
- Framing for faster recovery at high gigabit speeds
- Scalable discovery mechanisms using SLP and other protocols

The storage device is a dual-733 MHz Pentium III with 128 MB of memory and running iSCSI server software on top of Linux 2.4.2. The host server is an 800 MHz Pentium III with 256 MB of memory and running iSCSI client software on top of Linux 2.2.19. The two entities are connected via a Gigabit Ethernet connection over an Alteon 180 switch. The Ethernet frame size used was the regular 1500 bytes and no Jumbo frames were used. In addition, TCP/IP zero copy optimizations were not used. Instead, we relied on the standard socket interface that meant that the TCP copy-and-checksum routines were performed on both the host server and the storage device.

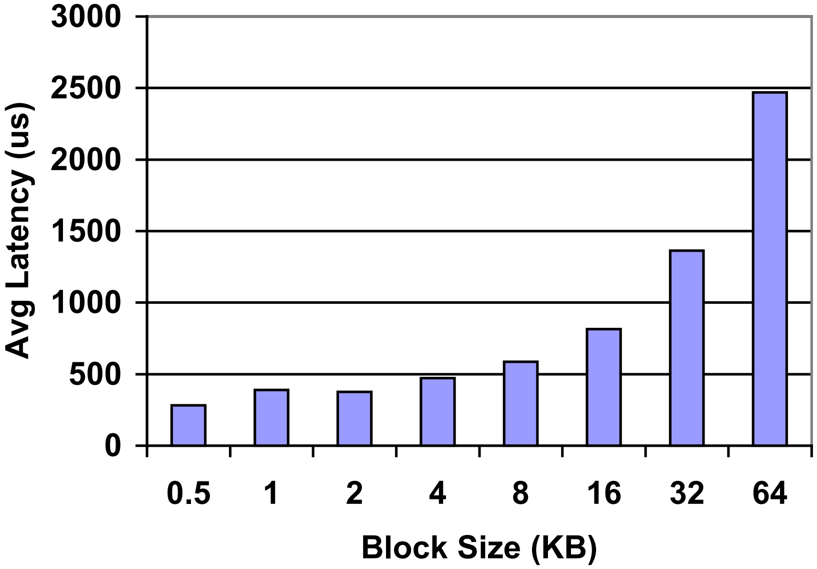
The test application resided on the host server and read raw SCSI blocks off a SCSI volume exported by the storage device. Since we wanted to isolate the efficiency of the transport, the application always read the same block so as to ensure a cache-hit. Otherwise, a cache miss would involve the RAID subsystem of the storage device and make it difficult to analyze the results. Writes were not measured as they can be done using various means (immediate, unsolicited, solicited) and add unneeded complexity to the analysis.

3.1 Latency Analysis

To measure latency, we used a single thread in the application to read raw SCSI blocks of various sizes from the storage device. For a particular block size, the same block was read 10,000 times and the average latency determined from the time required to perform the experiment. To measure throughput, we used 8 concurrent threads to read SCSI blocks of various sizes from the storage device. 8 threads were used because that is the concurrency limit imposed by the iSCSI client software in the host server. For a particular block size, each thread read a block 10,000 times and the throughput was calculated based on the time taken for all threads to finish reading the blocks. For the throughput experiment, we measured the CPU utilizations of the host server and storage device using the *vmstat* utility.

The latency measurements shown in Figure 1 indicate a variation of average latency for 283 us for a 512-byte block to a high of 2469 us for a 64 KB block. The average latency values provide no meaning by themselves but are comparable (within 5%) of latency numbers obtained from the specification sheet of a Fibre Channel storage device for all block sizes [8]. We had expected the cost of TCP/IP segmentation to have an adverse effect on latency for the larger block sizes, but it appears that the Gigabit Ethernet adapter is doing a reasonable job of interrupt coalescing. This indicates that the TCP/IP fast path for transmits and receives does not impose a prohibitive overhead on latency. Consequently, we do not expect IP storage (even in its software incarnation with no optimizations) to have an adverse effect of transaction-oriented applications and benchmarks.

Figure 1. Latency Measurements



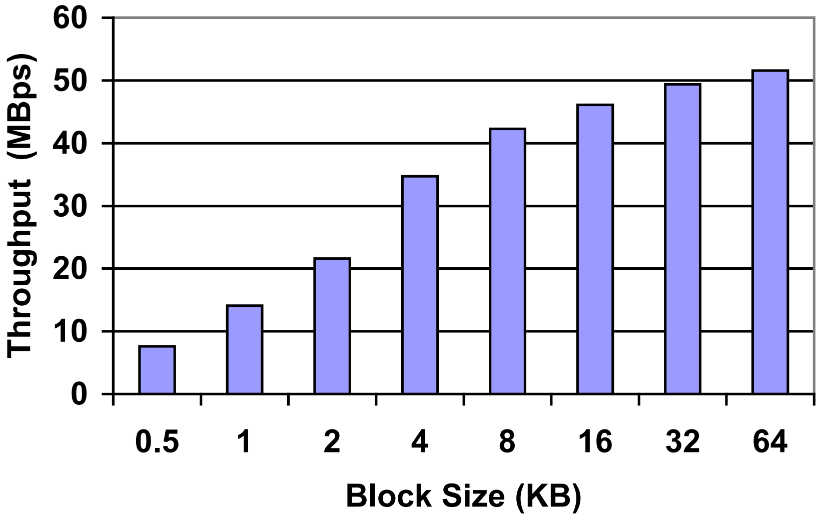
3.2 Throughout Analysis

However, the throughput measurements indicate a different story. Figure 2 indicates that while the average throughput from the storage device is competitive for the lower block sizes in comparison to that obtained from a Fibre Channel storage device, the peak throughput is about 60% less than what is obtainable from a Fibre Channel storage device[8]. In these experiments, the peak throughput is about 52 MBps for the 64 KB block size and is constrained by the CPU of the host server whose utilization is at 100%. A profiling of the CPU utilization of the host server indicated that the primary components were interrupt overhead (72%) and TCP copy-and-checksum (23%).

In addition, during the throughput experiments for the 64 KB block size, the CPU utilization of the storage device is at 51% indicating that the storage device is capable of delivering additional throughput. In fact, by using multiple initiators, we are able to obtain a throughput of 100 MBps at around 98% CPU utilization in the storage device. At this throughput, the constraining factor was the limit imposed by the network adapter. The CPU utilization figures were not available for the Fibre Channel storage device [8].

The CPU utilization of the host server is greater than that of the storage device because the host server is the receiver of bulk data. The receiving of data involves interrupting the host server every time a frame arrives and increases the interrupt overhead even if interrupt coalescing is used. This implies that if the experiments above involved writes, then the CPU utilization of the storage device would be higher.

Figure 2. Throughput Measurements



The results indicate that the main performance bottleneck in meeting the requirements of storage area networks is the high CPU utilization involved with bulk data transfers. The two main components of the high CPU utilization are:

- Interrupt overhead due to frame size transfers from the adapter to the host at high rates.
- The overhead due to TCP copy-and-checksum in standard TCP/IP stacks for bulk data.

4 Improvement Techniques

There are four potential avenues to reduce the high CPU utilization issues in an IP storage subsystem.

First, the interrupt overhead can be reduced by using 9KB Jumbo Ethernet frames, because this reduces the number of interrupts per bulk data transfer. For example, transferring a 32 KB data payload using the standard Ethernet frame may involve as many as 22 interrupts in the worst case whereas using the 9KB Jumbo Ethernet frame only 4 interrupts may be involved. However, the Jumbo Ethernet frames are not standardized and are not likely to be present in 10 Gigabit Ethernet.

Second, modified TCP/IP stacks with zero-copy transmit capability can be used to reduce the TCP copy-and-checksum overhead; the responsibility of generating the checksum is off-loaded to the network adapter. However, zero-copy receives are not possible on such stacks because the network adapters are typically unaware of the final destination of any frame.

Third, network adapters with TCP/IP offload engines (TOE) have been released [9] where the entire TCP/IP stack is offloaded onto the network adapter. This also reduces the TCP copy-and-checksum overhead. However, zero-copy receives are not possible on such stacks because the TCP/IP stack is also typically unaware of the final destination of any TCP/IP packet. There is proposed work to add enough application hints to the TCP/IP header to make zero-copy receives possible.

The fourth and most promising approach is the anticipated emergence of specialized adapters that have an iSCSI interface. This approach will reduce the interrupt overhead, as the iSCSI adapter will ensure at most one interrupt per data transfer. In addition, offloading the protocol processing to the adapter will eliminate TCP/IP copy-and-checksum overhead. The disadvantage of this approach is that the use of such specialized adapters implies that commodity network adapters cannot be used in IP storage area networks. However, one can still use the existing switches and wiring present in commodity Ethernet networks.

5 Conclusions

Advanced networking technology has led to the concept of storage networks where pooled storage is available as a service to host servers. The emergence of Gigabit Ethernet technology has raised the question of whether we can use commodity IP networks for storage instead of specialized storage area networks. This paper examines the issues involving IP storage networks and presents a performance analysis focusing on latency and throughput. The results indicate that the main performance bottleneck in meeting the requirements of storage area networks is the high CPU utilization involved with bulk data transfers. The two main components of the high CPU utilization are the interrupt overhead due to the bulk data transfers as well as the TCP copy-and-checksum overhead. We finally present four potential avenues to reduce the high CPU utilization issues in an IP storage subsystem.

References

[1] A. Gallatin, J. Chase, and K. Yocum, "Trapeze/IP: TCP/IP at Near-Gigabit Speeds", *Proceedings of USENIX Technical Conference (FreeNix Track)*, June 1999.
[2] R. Van Meter, G. Finn and Steve Hotz, "VISA: Netstation's Virtual Internet SCSI Adapter", *ASPLOS-VIII*, October 1998.
[3] A. Benner*, "Fibre Channel: Gigabit Communications and I/O For Computer Networks"*, McGraw-Hill, 1996.
[4] J. Satran et al., "iSCSI", IETF Work in Progress (IPS group), http://www.ietf.org/html.charters/ips-charter.html, 2001.
[5] Hsiao Keng, and J. Chu, "Zero-copy TCP in Solaris", *Proceedings of the USENIX 1996 Annual Technical Conference*, January 1996.
[6] http://www.infinibandta.org
[7] K.Voruganti, and P. Sarkar, "An Analysis of Three Gigabit Networking Protocols for Storage Area Networks'. *20th IEEE International Performance, Computing, and Communications Conference"*, April 2001.
[8] Mylex Corp., "White Paper on the Performance of the Mylex SanArray Pro FF2 Storage Controller", Mylex Technical Report, 2001.
[9] http://www.alacritech.com