Performance Improvement of Sequential Access to IP-Storage using IP-SAN analysis tools

Masato Oguchi [1,2], Saneyasu Yamaguchi [2], and Masaru Kitsuregawa [2]

[1] Department of Information Sciences, Ochanomizu University
2-1-1 Otsuka Bunkyo-ku, Tokyo 112-8610, Japan
oguchi@computer.org

[2] Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba Meguro-ku, Tokyo 153-8505, Japan

Abstract

In terms of the storage consolidation of the cluster computing systems, Storage Area Network (SAN) plays a significant role. Although SAN has already become an important tool in the business field, the current generation SAN based on Fibre Channel (FC) has some demerits, such that it is considerably expensive, the number of FC engineers is small, and so on. The next generation SAN based on IP is expected to remedy those problems. iSCSI is a hopeful standard of IP-SAN.

iSCSI has a structure of multi-layer protocols. A typical configuration of protocols to realize the system is; SCSI over iSCSI over TCP/IP over Ethernet. Thus, in order to improve the performance of the system, it is inevitable to analyze the complicated behavior of each layer precisely. In this paper, we present an IP-SAN analysis tool that monitors iSCSI layers from various points of view. Performance improvement of sequential access to IP-storage is achieved under some conditions, using this analysis tool.

1 Introduction

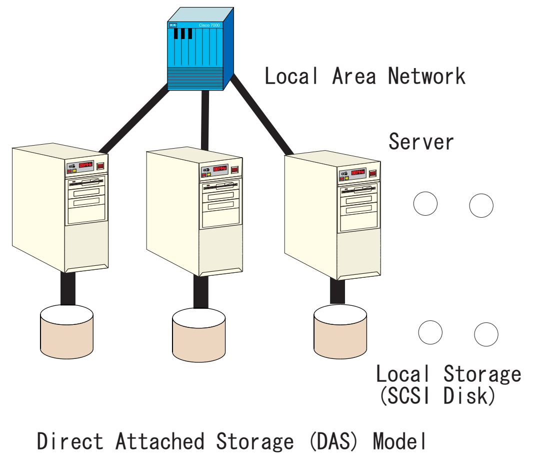
PC/WS cluster, a system connecting a large number of computers with a high-speed LAN, is used in the field of high performance computing[1][2][3]. A mass volume of data is processed in those clusters, for example, in databases and data mining applications[4][5].

However, it is not efficient to execute such data-intensive applications in a shared-nothing system, which has neither shared memory nor shared storage. A large amount of data must be transferred through the network in such a case. More serious problem is its management costs. As the size of system grows and the volume of data processed in the system becomes huge, it is extremely difficult to manage the data dispersed among a large number of storage units. Although the hardware of the system becomes inexpensive as the computer and storage technology advances, its management cost is not negligible anymore.

In order to consolidate the back-end of the system, Storage Area Network (SAN) is introduced[6]. The manageability of the storage improves greatly by the deployment of SAN. An overview of the system with shared storage using SAN is shown in Figure 1. In this figure, the conventional cluster that has only Direct Attached Storage (DAS) is also presented. In the DAS case, storage connected to another node can only be accessed through LAN and the node, which tends to become a bottleneck of the system. Compared to DAS, SAN provides better manageability of the system as well as better performance.

Although SAN has already become an important tool in the business field, the current generation SAN based on Fibre Channel (FC)[7][8] has some demerits, for example, it is considerably expensive, the number of FC engineers is small, and so on. The next generation SAN based on IP is expected to remedy those defects. iSCSI is a hopeful standard of IP-SAN, which is approved by IETF in February 2003[9][10][11][12]. It encapsulates SCSI protocol into TCP/IP working on Gigabit/10Gigabit Ethernet. Since the IP-SAN is based on the commodity technologies such as TCP/IP and Ethernet, it has an excellent cost/performance ratio, and thus it will be widely used in the near future.

Because iSCSI has a structure of multi-layer protocols, it is difficult to improve the system performance without analyzing the behavior of each layer precisely. Although iSCSI systems are discussed in some


Local Area Network
Server
Local Storage (SCSI Disk)
Direct Attached Storage (DAS) Model

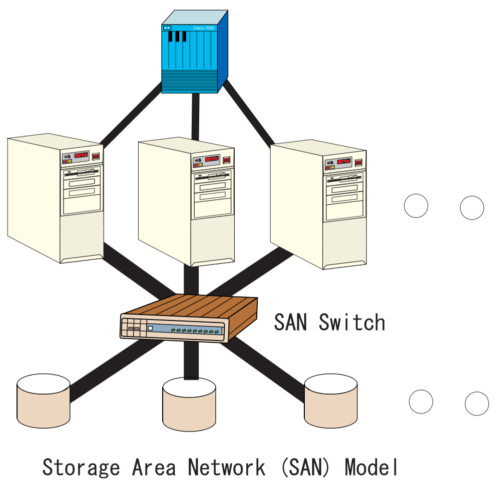


Figure 1. DAS and SAN

literatures[13][14][15], they do not analyze each layer of the iSCSI protocol. In this paper, we have developed an iSCSI analysis tool, and performance improvement of sequential access to IP-storage is achieved using this tool. Sequential read is important operation for SAN, because it is used for the backup of the system and some data-intensive applications like data mining.

The rest of the paper is organized as follows. In Section 2, iSCSI protocol and its format are presented. Our iSCSI analysis tool is introduced in Section 3. In Section 4, sequential access to IP-storage is analyzed using the tool, and performance improvement is achieved in a long-latency environment. Final remarks are stated in Section 5.

2 iSCSI protocol and its format

iSCSI encapsulates SCSI packets into TCP/IP frame and transfers through a network. The protocol stack is

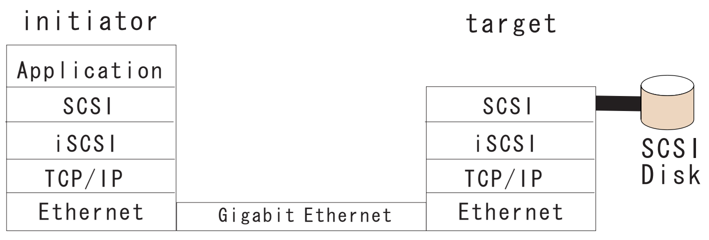


Figure 2. Each layer of iSCSI protocol

SCSI over iSCSI over TCP/IP. In most cases, it is SCSI over iSCSI over TCP/IP over Ethernet as shown in Figure 2.

The node beginning to access storage beyond the network is called "initiator", and the node that receives the request and provides its local storage to the initiator is called "target". Both nodes have the multi-layer protocol as shown in the figure.

SCSI interface is provided to the application. The access to the remote storage progresses as follows: The SCSI commands issued by the application are encapsulated in PDU of the iSCSI layer, and passed to the TCP/IP layer. Normally, the data is divided and put into the data field of multiple TCP segments. The TCP segments are encapsulated into the IP datagram, which is put into the Ethernet packet in turn, and finally transferred through the network.

At the target side, the received Ethernet packet goes to the IP layer, which is passed to the TCP layer after the header is removed. The TCP layer gives this segment to the iSCSI layer, and then the SCSI command inside it is taken out and executed finally. For example, if a SCSI Read command is issued from the initiator, it will reach to the SCSI layer of the target, and the contents of the SCSI disk will be returned in response to the command. In Figure 3, "iSCSI Data-in" Protocol Data Unit (PDU) is returned from target in response to 32Kbytes SCSI Read command. In this figure, the sizes of each field in the TCP/IP header are the case of our experimental system[1].

The performance characteristics depend on the behavior of each protocol layer of SAN. It is difficult to achieve performance improvement unless the behavior of each layer and the interactions between the layers are clarified. Because TCP/IP is designed as a general-purpose communication protocol, its performance tends to degrade drastically when it is used for burst traffic in IP-SAN. Although it is possible to design a special protocol only for a storage network, such a proprietary system will not be widely used. It is inevitable to adopt TCP/IP as a lower layer of SAN, and the efficient multi-layer protocol should be designed. For this purpose, we analyze the

[1]Timestamp option is used in the TCP header in this figure.

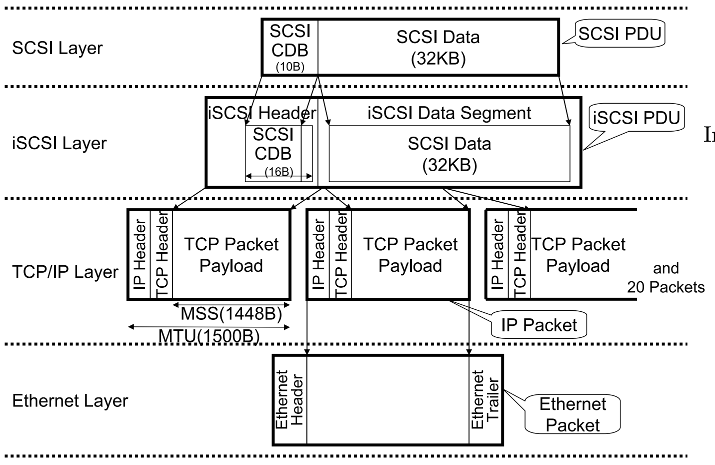


Figure 3. iSCSI packet processed at each layer

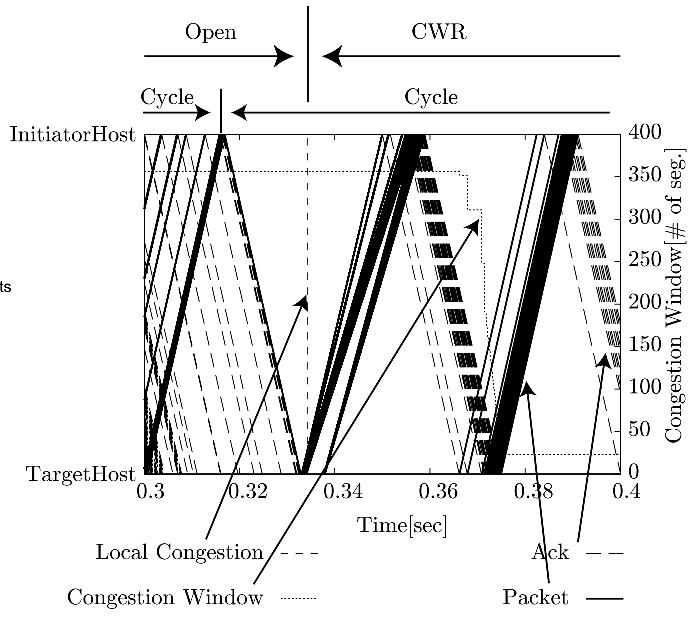


Figure 4. Visualization of packet transfer

behavior of the complicated protocol layers of the iSCSI system precisely.

3 iSCSI analysis tool

3.1 An overview of analysis tool

In order to investigate the behavior of each layer when the iSCSI system is used, we have developed an iSCSI analysis tool. It is difficult to understand the behavior of the whole system only monitoring a part of it, since iSCSI has a complicated multi-layer configuration. Thus, the tool analyzes the system from various points of view. It has the following functions.

1. Protocol translation of each layer (SCSI, iSCSI, TCP/IP)
2. Visualization of packet transfer
3. Monitoring TCP flow control
4. Detection of lost packet
5. Simple iSCSI storage access generator

Some of these functions are explained in the following subsections.

3.2 Visualization of packet transfer

In Figure 4, the transferred iSCSI packet is captured and visualized.

In the TCP/IP communications, the data flow is controlled in the TCP layer using the window size. TCP has two kinds of window; one is an advertised window and the other is a congestion window. The advertised window size is designated in the header of TCP, notified from the receiver to the sender. The congestion window size is decided in the TCP implementation, contained in the kernel of the sender. The smaller one works as the window size that controls the data flow. The sender is allowed to transfer packets as much as this number, without receiving any acknowledgment (Ack). The transferred packets and their Acks are visualized in Figure 4. The congestion window size and some events occurred in the TCP implementation are also shown in the figure. This will be explained in the next subsection.

3.3 Monitoring TCP flow control

In most cases, the TCP flow control function is implemented in the kernel, and thus, the behavior of the function depends on TCP implementation in the operating system. We have developed a monitoring tool of the Linux TCP flow control mechanism.

The Linux TCP flow control is realized as a state machine. It has several states including TCP_CA_OPEN, TCP_CA_Recovery, TCP_CA_CWR, and TCP_CA_Loss. For example, TCP_CA_OPEN is a normal state that has no congestion, in which the congestion window size increases step by step. When congestion is detected by an event such as a packet loss, the state will move to TCP_CA_Recovery, TCP_CA_CWR, or TCP_CA_Loss. In those cases, the congestion window size decreases drastically, and the performance of the upper layer like iSCSI is severely influenced. The analysis tool shows the transition of those states, which is marked as "Open"

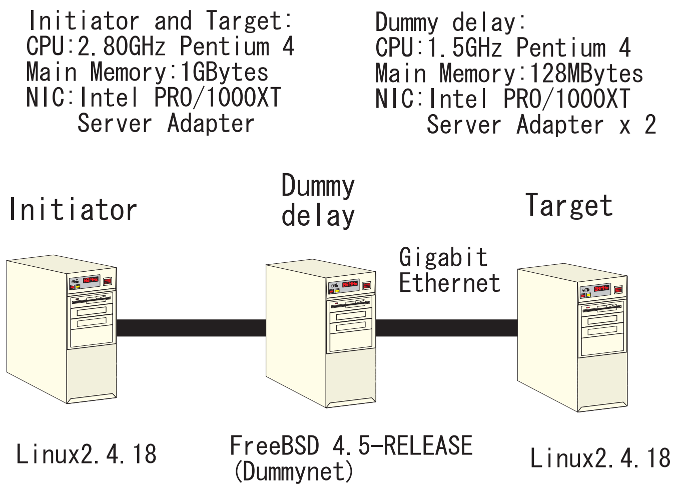


Figure 5. An overview of the experimental system

(stands for TCP_CA_OPEN) and "CWR" (stands for TCP_CA_CWR) in Figure 4. The congestion window size is also shown in this figure.

4 Performance improvement of sequential access to IP-Storage in a long-latency environment

4.1 An overview of our experimental system

In Figure 5, our iSCSI experimental system is presented. This is used for the iSCSI storage access analysis in a long-latency environment.

Three nodes of PCs are used in the system. They are used as the initiator and the target of iSCSI, and the dummy delay node that provides an artificial network delay. The initiator and the target have 2.80GHz Pentium 4 CPU, 1Gbytes main memory, and Intel PRO/1000 XT Server Adapter as a Gigabit Ethernet card, respectively. The dummy delay has 1.5GHz Pentium 4 CPU, 128Mbytes main memory, and 2 connections of Intel PRO/1000 XT Server Adapter. Linux 2.4.18-3 is used in the initiator and the target, and FreeBSD 4.5-RELEASE is used in the dummy delay node.

As the iSCSI initiator and target drivers, a reference implementation released from InterOperability Lab in the University of New Hampshire[16] is used. This is a reference implementation ver.3 based on the iSCSI Draft 18. As parameters in the iSCSI specification, MaxRecvDataSegmentLength, MaxBurstLength, and FirstBurstLength are 16,777,215bytes, respectively.

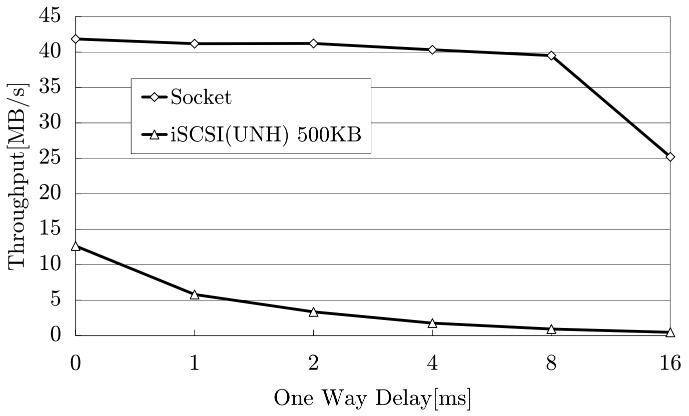


Figure 6. Throughput of iSCSI sequential read (default parameters)

4.2 iSCSI performance in a long-latency environment

The experimental result is shown in Figure 6. The x-axis is the artificial one way delay inserted by the dummy delay node shown in Figure 5. "0ms" is a case, in which the dummy delay node is used but the delay is not inserted actually, and the real delay time is about 140μs in this case.

The lower line, indicated "iSCSI(UNH) 500KB" in the figure, is the throughput of the iSCSI sequential read from the raw device using 500Kbytes block size. The upper line shows the throughput of the simple socket communication, which is the maximum limit value of the data transfer on this connection. The advertised window size of the receiver is 2Mbytes.

According to this result, the throughput of the iSCSI sequential read is much lower than that of the socket communication, and it decreases as the one way delay increases. The performance deterioration of using multiple iSCSI layers is extremely large in this case. We have investigated the behavior of each layer using the iSCSI analysis tool.

4.3 Performance analysis using the tool

The experiment of data transfer in the previous subsection is monitored using the protocol translation function of the iSCSI analysis tool. Several captured packets include 32Kbytes SCSI Read command. When an application issues a system call like read() function, this is going through file system, block device, character device, SCSI driver, and iSCSI driver, and then passed to the TCP/IP layer. The block size of a system call and the iSCSI PDU block size are not necessarily equal, depend-

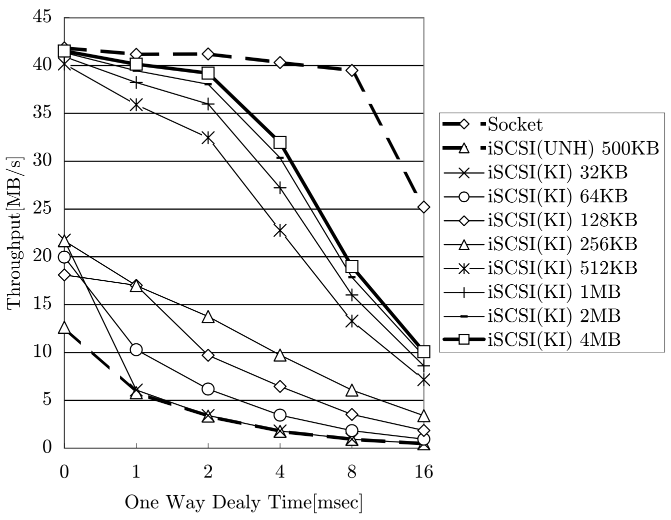


Figure 7. Throughput of iSCSI sequential read (using large iSCSI PDU)

ing of the iSCSI implementation. In this experiment, although the application issues 500Kbytes read() function, it is divided into many 32Kbytes blocks and sent to the target. In the long-latency environment, this causes long waiting time, since each block is transferred one by one, and each cycle of the transfer includes the access delay.

Thus, it is a good idea to enlarge the iSCSI PDU block size in such a case. Unfortunately, the iSCSI initiator driver used in this experiment has no option to change the value. Therefore, we use the Simple iSCSI storage access generator that produces pseudo iSCSI access traffic. The throughput with a large iSCSI PDU block size using this tool is measured. The result is shown in Figure 7.

In this figure, 8 new lines indicated "iSCSI(KI)" are the throughput when the iSCSI PDU block size is changed. The rest 2 lines, "Socket" and "iSCSI(UNH) 500KB", are the same with the previous case. As this figure shows, higher throughput is achieved when larger iSCSI PDU block size is used. The improvement from the default "iSCSI(UNH)" case is extensively high, especially when the iSCSI PDU block size is larger than 1Mbytes.

5 Conclusion

As the importance of data-intensive applications increases more and more, storage becomes significant part of the high performance computing systems. SAN is an essential mechanism to consolidate a large number of storage units dispersed in the cluster computing systems. This works for reducing the management cost of storage. While the FC-SAN already plays a principal role in the business field currently, IP-SAN, the next generation SAN, is ready to be adopted.

Since the IP-SAN has multi-layer protocols, it is inevitable to analyze each layer of the protocols precisely, in order to achieve higher performance using IP-SAN. An iSCSI analysis tool is developed, which monitors the behavior of each layer of the protocols. We have analyzed iSCSI sequential access in a long-latency environment using this tool. Based on the result of analysis, higher throughput is achieved with larger iSCSI PDU in this experiment. We will investigate various other cases using this tool.

Acknowledgment

This project is partly supported by the Ministry of Education, Culture, Sports, Science and Technology, under Grant 13224014 of Grant-in-Aid for Scientific Research on Priority Areas.

References

[1] D. E. Culler, A. A. Dusseau, R. A. Dusseau, B. Chun, S. Lumetta, A. Mainwaring, R. Martin, C. Yoshikawa, and F. Wong, "Parallel Computing on the Berkeley NOW," *Proc. 1997 Joint Symp. Parallel Processing(JSPP '97)*, pp. 237-247, May 1997.

[2] T. Tamura, M. Oguchi, and M. Kitsuregawa, "Parallel Database Processing on a 100 Node PC Cluster: Cases for Decision Support Query Processing and Data Mining," *Proc. SC97: High Performance Networking and Computing (SuperComputing '97)*, Nov. 1997.

[3] G. Bell and J. Gray, "What's Next in High-Performance Computing?," *Communications of the ACM*, Vol.45, No.2, pp.91-95, February 2002.

[4] V. Ganti, J. Gehrke, and R. Ramakrishnan, "Mining Very Large Databases," *IEEE Computer*, vol. 32, no. 8, pp. 38-45, Aug. 1999.

[5] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, MIT Press, 2001.

[6] B. Phillips, "Have Storage Area Networks Come of Age?," *IEEE Computer*, Vol.31, No.7, pp.10-12, July 1998.

[7] ANSI, "Fibre Channel - Arbitrated Loop," Standard X3.272-1996, 1996.

[8] ANSI, "Fibre Channel - Switch Fabric," Standard NCITS 320-1998, 1998.

[9] IETF, http://www.ietf.org/

[10] IETF IP Storage (ips) Charter, http://www.ietf.org/html.charters/ips-charter.html

[11] iSCSI Draft, http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.txt

[12] Storage Networking Industry Association, http://www.snia.org/

[13] W. T. Ng, B. Hillyer, E. Shriver, E. Gabber, and B. Ozden, "Obtaining High Performance for Storage Outsourcing," *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, pp. 145-158, Jan. 2002.

[14] P. Sarkar and K. Voruganti, "IP Storage: The Challenge Ahead", *Proc. Tenth NASA Goddard Conference on Mass Storage Systems and Technologies*, pp. 35-42, Apr. 2002.

[15] P. Sarkar, S. Uttamchandani, and K. Voruganti, "Storage over IP: When Does Hardware Support help?", *Proc. FAST 2003, USENIX Conference on File and Storage Technologies*, pp. 231-244, Jan. 2002.

[16] InterOperability Lab in the University of New Hampshire, http://www.iol.uhn.edu/consortiums/iscsi/