Comparative Performance Evaluation of iSCSI Protocol over Metropolitan, Local, and Wide Area Networks

Ismail Dalgic   Kadir Ozdemir   Rajkumar Velpuri   Jason Weber
Intransa, Inc
2870 Zanker Rd.
San Jose, CA 95134-2114
Tel: +1-408-678-8600
{ismail.dalgic, kadir.ozdemir, rajkumar.velpuri, jason.weber}@intransa.com

Helen Chen
Sandia National Laboratories, California
PO Box 969
Livermore, CA 94551
Tel: +1- 925-294-3000
hycsw@ca.sandia.gov

Umesh Kukreja
Atrica, Inc.
3255-3 Scott Blvd
Santa Clara, CA 95054
Tel: +1-408-562-9400
umesh_kukreja@atrica.com

Abstract

We identify the tunable parameters of iSCSI and TCP that affect the performance characteristics for local, metropolitan, and wide area networks. Through measurements, we determine the effect of these parameters on the throughput. We conclude that with the appropriate tuning of those parameters, iSCSI and TCP protocols maintain a good level of throughput for all types of networks.

1. Introduction

iSCSI [1] is a promising new technology, which overcomes the distance limitations of other storage networking technologies such as Fibre Channel and Infiniband, and thereby enables globally distributed mass storage systems. Wide Area Ethernet services, at the same time are emerging as a strong contender for wide area connectivity among multiple enterprise locations. While some studies exist on the performance characteristics of the iSCSI protocol [2] [3], the performance characteristics for metropolitan area and wide area networks are yet to be understood. The iSCSI protocol and the underlying TCP/IP and Ethernet protocols have some configurable parameters which impact performance. In this paper, we investigate the effect of some of these parameters on iSCSI throughput.

2. Parameters

At the iSCSI level, the parameters of interest from a performance point of view are: (i) command request size, (ii) iSCSI command window credit amount, (iii) the number of simultaneous iSCSI connections in a session, and (iv) the option of sending solicited vs. unsolicited data. The command request size, the command window credit amount, and the number of simultaneous connections may impact both read and write performance. The choice of solicited vs. unsolicited data may impact write performance, but it has no impact on reads.

At the TCP/IP level, the most important parameters are the send and receive window sizes especially in networks with a large bandwidth-delay product such as high speed WANs.

The iSCSI command request size is the amount of data that is sent or received as part of a SCSI command encapsulated in iSCSI. The iSCSI command window credit amount, dynamically set by the target, determines the maximum number of iSCSI commands that can be outstanding at a given time. The product of these two parameters will determine the maximum amount of data that can be pipelined in the network to deal with the network latency. Increasing this amount will generally improve throughput.

The primary reason for iSCSI to support multiple connections per session is to take advantage of trunking in Gigabit LAN switches [4]; each TCP connection may utilize a different link, thus improving the overall throughput of the session. However, even on a WAN or MAN link where only a single path is available between an initiator and a target, the number of simultaneous connections in an iSCSI session may impact the performance due to the behavior of the TCP protocol where each TCP connection adjusts its transfer rate so as to share fairly a congested path. By allowing multiple connections per iSCSI session, the iSCSI traffic is effectively given priority over other TCP traffic. Furthermore, a packet loss in a TCP connection triggers the TCP slow-start and congestion avoidance algorithms, resulting in a drop in the throughput which takes some time to reach back to the maximum possible level [5]. By using multiple connections in a session, the overall impact of this temporary drop in throughput is reduced. On the other hand, the iSCSI protocol has to obey the SCSI command ordering rules that may reduce the parallelism among multiple connections.

As far as solicited vs. unsolicited data transfer is concerned, three independent parameters determine the transfer type: *FirstBurstLength*, *MaxBurstLength,* and *MaxRecvDataSegmentLength* [1]. *FirstBurstLength* determines the maximum amount of unsolicited data that the initiator can send per command. *MaxBurstLength* determines the maximum amount of solicited data that the initiator can send per command. *MaxRecvDataSegmentLength* is the maximum data segment size that can be sent in each protocol data unit (PDU). There are many ways that these 3 parameters can be set. In this study, we consider two cases which produce results in the two extremes: most-unsolicited and most-solicited data allowed by the iSCSI protocol. Most-unsolicited data implies that the *FirstBurstLength* is greater than or equal to the maximum write command request size that the initiator generates. Most-solicited data implies that the unsolicited data mode is disabled during the login negotiation, effectively equivalent to setting the *FirstBurstLength* to zero. In this mode, the target will notify the initiator when it is ready to receive data for a given command. This will give the target more control in the receive buffer allocation, but it will introduce extra round trip delays as compared to the fully unsolicited mode.

3. Experimental Setup

In this paper, we study the effect of the aforementioned parameters on iSCSI performance for different network types between the initiators and the targets. Note however that we did not study scenarios with multiple TCP connections per iSCSI session because targets and initiators that support this feature are not yet widely available. In addition, the test configurations we study do not have multiple paths.

In order to isolate the effect of the network latency and not to be affected by the idiosyncrasies of different commercial products, we used a WAN/MAN emulator. This allowed us to vary the network latency while keeping all other parameters unchanged. More specifically, our experimental setup consisted of an open source software initiator by Cisco running on a 933 MHz two processor Intel Pentium III SMP machine, and an Intransa IP5000 iSCSI target, interconnected by a LANforge ICE WAN emulator by Candela Technologies. As traffic generators, we used two open source tools, *xdd* for block IO and *ettcp* for tcp traffic.

In our experiments, we set the network bandwidth to the OC3 rate, 155Mb/s. This rate is the maximum supported by the WAN emulator. Since this paper's focus is network performance, we configured the IP5000 in write-back mode and we set the traffic patterns such that all reads are served from the cache. This allowed us to eliminate any possible disk IO bottleneck.

We studied four values of round trip latency: 0 ms as baseline, 2 ms for MAN and 10 and 50 ms for WAN. In addition, we performed some LAN measurements by using a Gigabit Ethernet connection between the initiator and target, without the LANforge.

4. Results

4.1 Effect of the TCP Window Size

We first studied the effects of TCP window size setting. By using the default 64KB settings of the Linux kernel 2.4.19, we obtained the results shown in Table 1. The first row corresponds to data being sent from the iSCSI initiator machine to the target, and the second row corresponds to the data sent in the other direction. As can be seen, even at the small values of round trip latency, the default TCP window size settings are inadequate to maintain a good level of throughput. We then increased the maximum send and receive window sizes to 10 MBytes for both the initiator and the target, and achieved the wire speed for all the latency values under consideration as shown in Table 2. In the remainder of this paper, we kept the maximum window sizes at 10 MBytes.

Table 1: TCP throughput results in MBytes/s with default TCP send and receive window sizes (64KB)

Transfer Direction	**Round Trip Latency**			
	0 ms	**2 ms**	**10 ms**	**50 ms**
I → T	19.0	18.7	4.4	0.9
T → I	19.0	13.5	3.2	0.7

Table 2: TCP Throughput results in MBytes/s with maximum TCP send and receive window sizes set to 10 MBytes

Transfer Direction	**Round Trip Latency**			
	0 ms	**2 ms**	**10 ms**	**50 ms**
I → T	19.3	19.3	19.3	19.3
T → I	19.3	19.3	19.3	19.3

4.2 Effect of the iSCSI Parameters

After eliminating the TCP bottleneck, we studied the effect of the iSCSI parameters. Table 3 presents the iSCSI throughput results for most solicited writes using different iSCSI command window and request sizes. It is clear that the throughput is adversely affected when the product of window and request size is small.

Table 4 shows similar results to Table 3, but for most unsolicited writes. Clearly, the elimination of the extra round trip delays help to improve the throughput.

Table 3: iSCSI throughput results in MBytes/s for most solicited writes

Request Size	Window Size	Round Trip Latency			
		0ms	2ms	10ms	50ms
1KB	1	1.3	0.2	0.05	0.01
	32	11.3	3.3	0.8	0.2
8KB	1	7.7	1.7	0.4	0.08
	32	19.0	18.9	5.8	1.2
64KB	1	19.2	10.9	2.9	0.6
	32	19.2	19.3	19.3	4.5
256KB	1	19.2	19.2	7.8	1.7
	32	19.3	19.3	19.3	4.7

Table 4: iSCSI throughput results in MBytes/s for most unsolicited writes

Request Size	Window Size	Round Trip Latency			
		0ms	2ms	10ms	50ms
1KB	1	2.2	0.4	0.09	0.02
	32	17.4	6.6	1.5	0.3
8KB	1	10.3	3.2	0.7	0.2
	32	19.2	19.2	11.6	2.5
64KB	1	19.3	17.6	5.4	1.2
	32	19.3	19.3	19.3	4.8
256KB	1	19.3	19.3	11.6	2.5
	32	19.3	19.3	19.3	4.9

Table 5: iSCSI throughput results in MBytes/s for reads

Request Size	Window Size	Round Trip Latency			
		0ms	2ms	10ms	50ms
1KB	1	2.3	0.4	0.1	0.02
	32	17.6	8.0	2.3	0.5
8KB	1	10.2	3.1	0.8	0.2
	32	19.2	19.2	19.1	4.7
64KB	1	19.2	17.2	5.4	1.2
	32	19.2	19.3	19.3	19.2
256KB	1	19.2	19.2	18.3	4.7
	32	19.3	19.3	19.3	19.3

Table 5 shows similar results for read requests. Considering that both the read requests and the unsolicited write requests involve one round trip latency per request, the results in Table 5 match the results in Table 4 in many cases. However, for some other cases, read throughput seems to significantly exceed the most unsolicited write throughput.

Finally, our LAN measurement results are shown in Table 6 for writes and reads, using various iSCSI command window and request sizes. It is interesting to note that even in a low latency LAN environment, the product of the iSCSI window size and request size impacts the performance significantly. In addition, the unsolicited writes provide a significant increase in performance.

Table 6: iSCSI throughput results in MBytes/s in Gb/s LAN environment

Request Size	Window Size	Most Solicited Writes	Most Unsol. Writes	Reads
1KB	1	3.4	5.5	5.4
	32	17.1	23.4	19.9
8KB	1	17.4	25.0	22.5
	32	68.6	84.3	72.3
64KB	1	56.1	65.9	61.3
	32	96.5	99.8	91.9
256KB	1	71.3	79.9	74.7
	32	97.3	100.4	98.4

5. Conclusions

We have observed that the default TCP parameter values are inadequate for the high speed MAN and WAN environments, and therefore require tuning. We have seen that the product of the iSCSI command window and request sizes has a very significant effect on the performance as well. Furthermore, using the most solicited writes has a major performance penalty, and should be avoided whenever possible. With appropriate performance tuning, the iSCSI and TCP protocols are capable of achieving good throughput in all types of networks.

6. Acknowledgements

We would like to thank Mr. Ben Greear for kindly allowing us to use the LANforge LAN/MAN/WAN emulator for this study. We would also like to thank Mr. Robert Gilligan and Mr. Kenny Speer for their valuable feedback.

References

[1] IETF Internet Draft, "iSCSI", draft-ietf-ips-iscsi-20.txt, J Satran et al, Jan 2003

[2] "A Performance Analysis of the iSCSI Protocol," S Aiken et al, *proceedings of 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies,* San Diego, CA, USA, Apr 2003, pp123-134

[3] "IP SAN – From iSCSI to IP-Addressable Ethernet Disks," P Wang et al, *proceedings of 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies,* San Diego, CA, USA, Apr 2003, pp189-193

[4] IEEE 802.3-2002 "Information Technology - Telecommunication & Information Exchange Between Systems - LAN/MAN - Specific Requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," 2002, ISBN 0-7381-3089-3

[5] IETF RFC 2001, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," W Stevens, Jan 1997