STUDY OF THE EFFECT OF THE iSCSI PARAMETERS ON THE PERFORMANCE

Smita Vishwakarma
Computer Networks and Internet Engineering
Centre for Development of Advanced Computing
Kharghar, India – 400614
Email: smitav@cdacmumbai.in

Abstract. Internet SCSI (iSCSI) is a communication protocol that works over Transmission Control Protocol/Internet Protocol or TCP/IP on standard Ethernet networks and is used to access data storage blocks over the network. For an IP SAN i.e., a Storage Area Network working on the IP network, performance is a major concern. In this paper, we study the effect of the various iSCSI parameters on the IPSAN's performance so that the throughput of the iSCSI – based IP SAN can be maximized.

NS2 is an event driven simulator very popular in the research of networking. We have made a simulation model for the iSCSI-based storage system to study its performance with changes in the iSCSI protocol's various parameters. This is done by initiating and testing the communication between the iSCSI initiator and the iSCSI target.

1. Introduction. Computer storage has evolved from dedicated computers to client – server models to recently, network storage systems. In a network – based storage system, the storage devices are distributed over the network. They could be of different types, from different vendors and working with different protocols. The underlying network has some controllers which direct the data from the device to where it is needed.

FC SANs or storage networks [8] over fibre channel are fast and work over dedicated fiber optical cables. Its disadvantages are its large deployment time, high total cost of operation and requirement of special maintenance. On the other hand, iSCSI – based SANs use standard Ethernet network, are cheaper and easily deployable. It also supports the mirroring applications, the remote backup, disaster recovery etc. And, moreover, a well-designed IP SAN using iSCSI protocol should be as fast as FC SAN.

To build fast and efficient storage systems based on iSCSI Protocol [5], we need to study its performance by varying the iSCSI parameters and study its response and throughput. The best way to study the performance is obviously to build the prototype of the system and measure its performance. However, the actual measurement approach is time – consuming, expensive and restricted to a limited number of settings.

Simulation offers an easier and cheaper alternative to study the storage system. NS2 [3] is an event driven simulator very popular in the research of networking. It provides support for the simulation of TCP/UDP, multicast and routing protocols over wireless and wired networks, etc. The storage system under the study consists of an iSCSI initiator and an iSCSI target. The response time for each block access (R/W) made by the initiator to the target is found and the throughput is calculated. The system's performance is studied for a wide range of iSCSI parameter values.

This paper is organized as follows: Section 2 presents the iSCSI Protocol and the SAN. Sections 3 and 4 describe how various parameters affect Read and Write operations. Section 5 discusses the implementation of the iSCSI protocol in NS2. Section 6 consists of the graphs and the results. Section 7 tells us about the related work in this field. Finally this paper concludes in Section 8.

2. iSCSI Protocol. The iSCSI-based storage is very different from a traditional one. A traditional storage system is usually physically restricted to a limited environment, e.g. a data center. Transport protocols specific to that environment are used, e.g. Fibre Channel, parallel SCSI bus, etc.

On the other hand, in an iSCSI storage, the data transfer is not restricted to a small area. The initiator and the target can be quite far from each other. The networking technology in between the initiator and target can be diverse and heterogeneous, e.g. it can be Ethernet, Optical DWDM, ATM, Wireless, satellite or a combination of these.

The iSCSI protocol transfers the SCSI block oriented storage data over standard Ethernet based TCP/IP networks. iSCSI causes data transfer between the initiator(s) and target(s) over one or many connections. SCSI CDBs (Command Descriptor Blocks) are passed from SCSI layer to the iSCSI transport layer. The iSCSI layer encapsulates CDB in an iSCSI PDU (Protocol Data Unit) and passes it to TCP. When received, the iSCSI layer removes the CDB from PDU and forwards it to SCSI layer.

TCP guarantees reliable and in-order delivery of data packets. TCP automatically sends the requests for resending of the data if an error like acknowledgement not received within time-out period occurs or when the loss of the data occurs. And that is why, the iSCSI Protocol has been developed over TCP Protocol as storages requires reliable and in-order delivery of the packets. Even multiple connections can be made between the iSCSI initiator and iSCSI target.

3. Read Operation in Iscsi. A Read Request is to be sent by the initiator to the target. A session is established between the iSCSI initiator and the iSCSI target. Then, the connection is built within the session. The command PDU with the Read operation is sent to the target by the initiator. The target receives the command PDU and calculates how many data PDUs are to be sent. The data PDUs are sent via the TCP layer. After sending the data PDUs, the target sends the response PDU. We may choose not to send the response PDU in the end, if the initiator and the target have initially agreed to interpret the last data PDU as the response PDU too i.e., phase collapse occurs.

3.1. Effect of parameters on Read Performance

3.1.1. MaxRecvDataSegmentLength. The MaxRecvDataSegmentLength defines the size of the data that is to be sent with the data PDU. As MaxRecvDataSegmentLength increases, the PDU size increases and thus the time required to send it. When the data PDU size increases, the throughput also increases because more data is being sent with lesser overhead.

3.1.2. MaxConnections. As the number of connections between an iSCSI Initiator and iSCSI Target increases, the throughput increases. This is because command requests can be sent in parallel. The bandwidth of the connection is better utilized. But, the disk accesses are done serially and as the disk access is much slower than the network, the effective increase in the throughput will not be in the multiples of the number of connections.

Response time (Read) = Time to send Command PDU + Disk access time (serial) + Time to send data to target through the 'MaxConnections' no. of connections + Time taken to receive the command response

4. Write Operation in iSCSI. A Write Request is to be sent by the initiator to the target. A session is established between the iSCSI initiator and the iSCSI target. Then, the connection is built within the session. The command PDU with Write operation is sent to the target by the initiator. The target receives the command PDU. The initiator calculates how many data PDUs are to be sent. The data PDUs are sent via the TCP layer. After receiving the data PDUs, the target sends the response PDU.

4.1. Effect of parameters on Write Performance

4.1.1. MaxRecvDataSegmentLength. The MaxRecvDataSegmentLength defines the size of the data that is to be sent with the data PDU. As MaxRecvDataSegmentLength increases, the PDU size increases and thus the time required to send it. When the data PDU size increases, the throughput also increases because more data is being sent with lesser overhead.

4.1.2. ImmediateData. If the ImmediateData is Yes, an immediate data equivalent to the min {MaxRecvDataSegmentLength, FirstBurstLegth) can be sent along with the command PDU. It increases the overall throughput.

4.1.2. InitialR2T. If the InitialR2T is No, the initiator may send the data, the size given by the FirstBurstLength, without waiting for the initial R2T. This enhances the overall throughput, since each R2T includes a round trip delay.

4.1.3. FirstBurstLength. The FirstBurstLength gives the size of the unsolicited data that may be sent immediately after the sending the command PDU, if the InitialR2T is yes. Since data is sent without waiting for the initial R2T, thereby avoiding the round trip delay, it increases the overall throughput.

4.1.4. MaxBurstLength. The MaxBurstLength gives the maximum amount of the solicited data that maybe sent as a sequence. One R2T has to be received per sequence from the target. A sequence consists of the PDUs of the size given by the MaxRecvDataSegmentLength.

4.1.5. MaxConnections. As the number of the connections between an iSCSI Initiator and an iSCSI Target increases, the throughput increases. This is because the commands can then be sent in parallel. The bandwidth of the connections is better utilized. But, the disk accesses are done serially and as the disk access is much slower than the network, the effective increase in the throughput would not be in the multiples of the MaxConnections.

Response time (Write) = time to send a command request along with Immediate data + time to send unsolicited data + solicited data over MaxConnections + time to receive command response from the target + Disk Access Time at the initiator and the target.

5. Description of Simulation of iSCSI Protocol in NS2. We have simulated iSCSI protocol using NS to study the behavior of its parameters. In our Simulation we have created two nodes named as the iSCSIInitiator and the iSCSITarget. Figure 1 gives the class diagram. These nodes are linked with the duplex link.

Implementation: Our implementation of the iSCSI protocol in ns-2 supports the following classes:

iSCSIInitiator. iSCSIInitiator class has been inherited from the Agent class. It works as an iSCSI initiator in the simulation. The function of the Agent class is to transmit the packet from the source to the destination and calculate the Round Trip time. In NS2, an implicit acknowledgement of the packet is automatically sent by the destination to the sender of the packet. With the help of this acknowledgement, the Agent class calculates the Round Trip Time.

iSCSITarget. This class has been inherited from the Agent class. It behaves like an iSCSI target. As mentioned above, the function of the Agent class is to transmit the packet from the source to the destination and calculate the Round Trip time. In our implementation we have sent different PDUs which are required for the iSCSI protocol and calculated the Round Trip Time of each PDU by varying different parameters like MaxRecvDataSegmentLength, FirstBurstLength, MaxBurstLength under different working conditions, i.e. different bandwidths and latencies.

iSCSIInitiatorTimer. For the Time Handling functions, we have used the iSCSIInitiatorTimerHandler class. Using its objects, we can measure the time at which the packet is sent and received.

iSCSITargetTimer. This class works in the same way as iSCSIInitiatorTimerHandler class, described above.

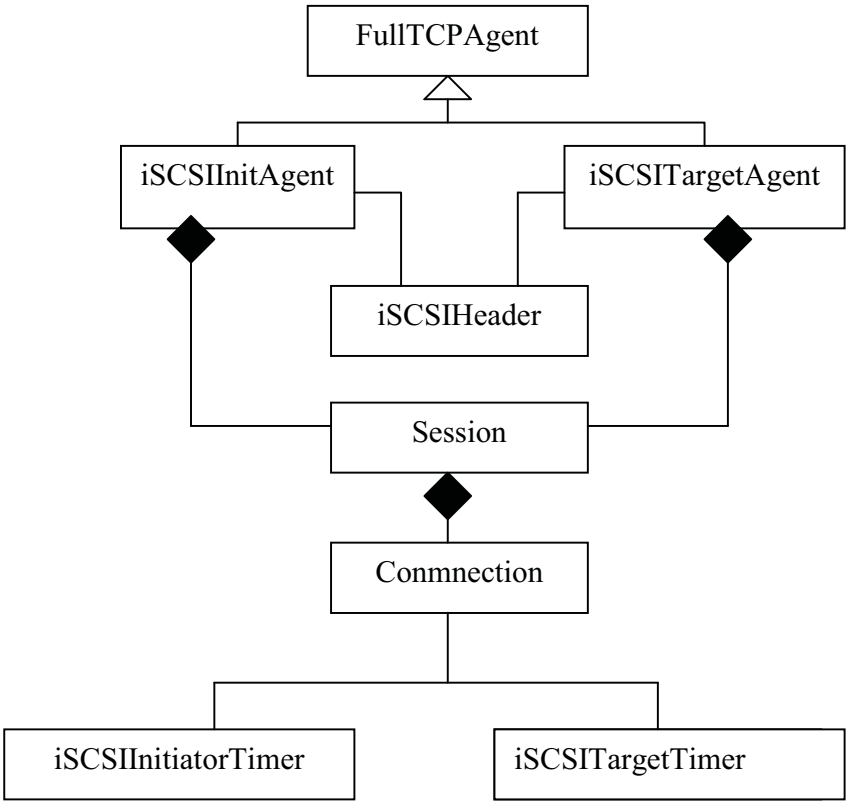

Fig. 1: Class Diagram for NS simulation of iSCSI Protocol

Our implementation supports the following features of the iSCSI:

- Sequence number of Text Command PDUs.
- Random Generation of Read and Write Request.
- Random Generation of Data Size.
- Any combination of Immediate, unsolicited and solicited data.

- Arbitrary values of the MaxRecvDataSegmentLength, FirstBurstLength, MaxBurstLength.

6. Performance Analysis. The response time is the time difference between the submission of the request and receipt of its response. It not only depends on the available bandwidth of the physical connection but also on the values of the various iSCSI parameters.

The throughput is the number of megabytes transferred per second. We have calculated it as the size of the data requested divided by the time taken to complete that request, i.e. the response time. As the request size of the data increases, the throughput increases because the overhead to send the data decreases.

6.1 Type of Workload – Sequential or Random. The number of blocks to be read from or written to is random. Workload could be sequential or random. That is, the starting block, from which data is read or data is written to, can also be random. If sequential data is accessed, the disk access time decreases and thus, the throughput increases.

Response time = P(read) [average time taken to complete one read request] + (1 -P(read)) [average time taken to complete one write request]

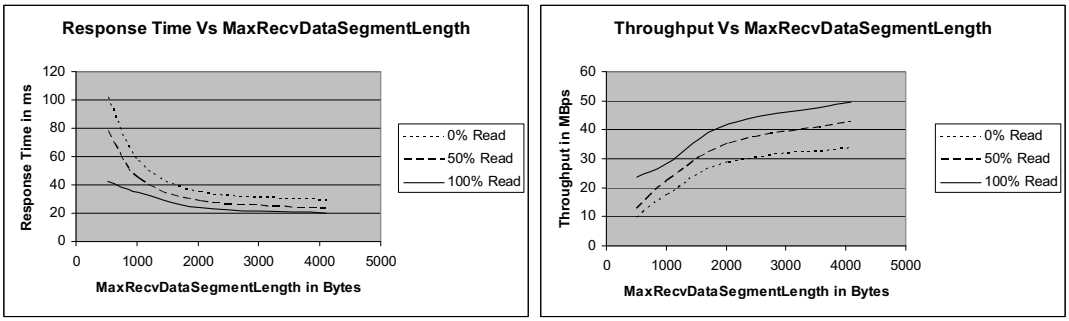

Fig. 2: Graph showing iSCSI performance as MRDSL changes for different Read percentages

As the MaxRecvDataSegmentLength increases, the size of the PDU increases and thus the time required to transfer one PDU increases. The throughput also increases with the MaxRecvDataSegmentLength as can be

observed from the figures 2 and 3. This is so because as the PDU size increases, the overhead in fulfilling the given request decreases. As the proportion of read requests increases, throughput increases because R2Ts are not exchanged in read requests.

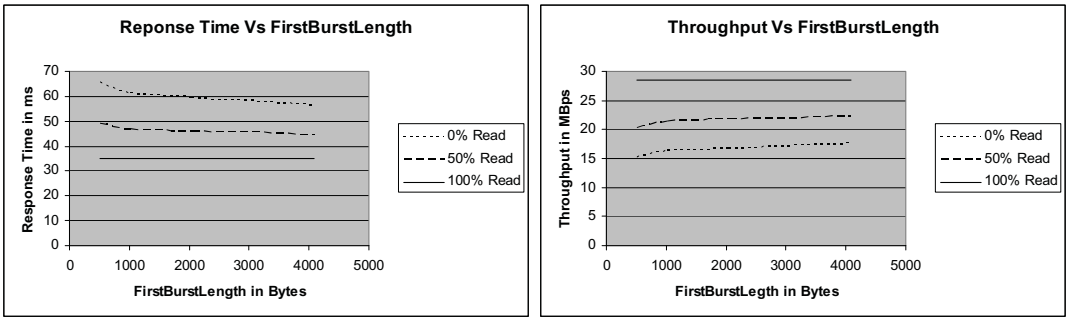

Fig. 3: Graph showing iSCSI performance as FBL changes for different Read percentages

As the FirstBurstLength increases, the size of the unsolicited data that can be sent for a given write request, without waiting for the first or the initial R2T from the target, increases. Thus, the round trip delay decreases and the throughput increases, as we see from the figures 4 and 5. We also see from the graphs that the size of the FirstBurstLength doesn't affect the Read Request because the R2Ts are not required to be sent for the read request.

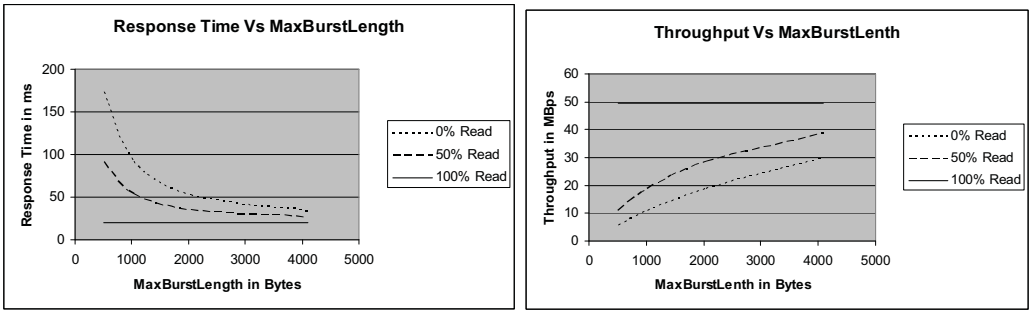

Fig. 4: Graph showing iSCSI performance as MBL changes for different Read percentages

The MaxBurstLength gives the amount of the data that can be sent for a single R2T for solicited data for a write request. Thus, the throughput increases as the MaxBurstLength increases because R2T need not be sent for every PDU, saving on the round trip delay. It won't affect the read requests because in a read request, the data is sent one after another without waiting for any R2T. This is evident from figures 6 and 7.

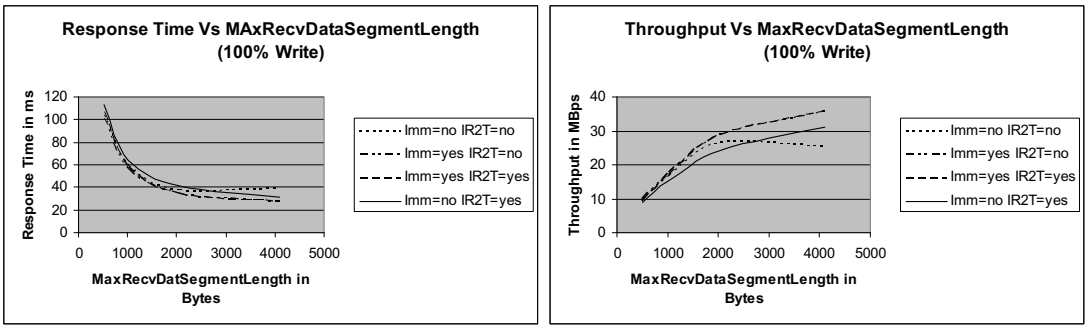

Fig. 5: Graph showing iSCSI performance as MRDSL changes while sending solicited and unsolicited data

According to the iSCSI protocol, the immediate data is sent if the parameter ImmediateData is yes. If InitialR2T is no, the unsolicited data is sent, without waiting for the initial R2T. Solicited data is sent in all cases, after sending the data of the size FirstBurstLength. As we can see from the figures 8 and 9, for sufficiently large data, the throughput is maximum when the immediate data and the unsolicited data are sent before sending the solicited data as compared to the case when solicited data is only sent.

7. Related Work. Paper [1] describes how various TCP protocols will fare when used along with the iSCSI Protocol. Paper [2] describes a useful model for simulating the iSCSI in the NS2. [3] is the official guide for the NS simulator. [5] is a popular book on the iSCSI protocol. [7] is a study on some of the iSCSI parameters. [8] is an introductory guide to SANs. [9] is a previous paper of us on simulating iSCSI protocol.. This paper shows the performance characteristics of the iSCSI as simulated in the NS2. It involves studying the effect of all the parameters which can have an impact on the iSCSI performance.

8. Conclusion. This paper describes the implementation of the iSCSI protocol in ns-2. It provides the details of the parameters and their effect on response time and throughput. Average throughput is calculated for randomly generated R/W PDUs. Both the round trip time and the throughput increases with the size of the data PDU. The throughput is the maximum when the unsolicited data is also sent.

References. [1] Girish Motwani and K. Gopinath : Evaluation of Advanced TCP Stacks in the iSCSI Environment using Simulation Model in the Proceedings of the 22nd IEEE/ 13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST 05)

[2] Yingping Lu, Farrukh Noman, David H.C. Du: Simulation Study of iSCSI-based Storage System in http://www.dtc.umn.edu/publications/reports/2005_06.pdf

[3] K. Fall and K. Varadhan. "The Network Simulator ns-2", NS – Manual : VINT project (formerly ns Notes and Documentation)

[4] M. Chadalapka, H. Shah, U. Elzur,P. Thaler, and M. Ko. "A study of iSCSI extensions for RDMA (iSER)."Proc. ACM SIGCOMM 2003, Workshop on Network I/O Convergence, pp. 209-219.

[5] iSCSI-The Universal Storage Connection by John L. Hufferd.

[6] S. Hussain and R.D. McLeod: iSCSI Simulation for Internet Applications http://www.comtec.e-technik.uni-kassel.de/ICOMP/IC2004/ConfMan/SUBMISSIONS/58-obumabaoda.pdf

[7] Yamini Shastry, Steve Klotz, Robert D. Russell: Evaluating the Effect of iSCSI Protocol Prameters on Performance in the Proceeding (456) Parallel and Distributed Computing and Networks - 2005

[8] Jon Tate, Fabiano Lucchese, Richard Moore: IBM Redbook Introduction to Storage Area Networks

[9] Sankalp Bagaria, Smita Vishwakarma: iSCSI Simulation Study of Storage System in Proceedings of *uksim*, pp. 703-707, Tenth International Conference on Computer Modeling and Simulation (uksim 2008), 2008