Optimization of the Switches in Storage Networks

Nianmin Yao[1], Xiuli Zhao, Huimin Meng[2], and Xinlei Jiang[3]

[1] College of Computer Science and Technology,
Harbin Engineering University, Harbin, China
yaonianmin@hrbeu.edu.cn
[2] Dalian Institute of Light Industry
[3] China Construction Bank

Abstract. In the Storage Area Network (SAN), the mature network technology is used to substitute for the IO bus. Usually, the common switch is used in the storage network especially in the iSCSI storage system. Because the common switches are not aware of the environment it is used in, it can not make optimization for the storage network. In this paper, an optimization of storage network is presented which can greatly improve the performance and resource utilization of the whole storage system. In this optimization, the IO acknowledgment commands passing through the switch which is used in storage network are transferred in highest priority. This optimization of SAN has been certified by the simulation experiments.

Keywords: SAN, performance, network, switch.

1 Introduction

With the developing of the Internet, a dramatic growth of enterprise data storage capacity can be observed in the last couple of years. Many things including a lot of enterprise data coming onto the internet; data warehouse, e-business and especially the Web contents contribute the growth of storage. People require more and more performance, capacity and manageability of the storage with the time goes by. So the relative slow developing speed of the storage compared to the other parts of the computer system such as CPU, network bandwidth and so on are becoming the bottleneck of IT. Now, the SAN (Storage Area Network) is a popular technology to solve these problems. It can ensure the reliability, serviceability, scalability and availability of the storage. But now it obviously can not satisfy the increasing need of market, so many researchers are working on improving the technology of SAN. There are three main components in an SAN environment such as server systems, storage devices and interconnect devices. So methods to improve the SAN are mainly focused on these three components. For example, many data placement algorithms, for example [1][2][3][4] and so on, which mainly improve the storage devices are proposed to get more performance and scalability of the storage system. In this paper, we focus on the optimization

Y. Shi et al. (Eds.): ICCS 2007, Part III, LNCS 4489, pp. 66–73, 2007.
© Springer-Verlag Berlin Heidelberg 2007

of the switches which are used to connect the servers and storage devices. So far, there are not many works on improving the switches, because their implementation details are not much clear because they are commercial secrets for the company selling the products. But we may assert that the switches now being sold are not optimized specially for the use of storage networks because most of the switches which can be used to connect the servers can also be used in the storage networks. So the swiches even can not know which environment they are used in, storage network or normal intranet. In fact, there are a lot of properties of switches can be optimized specially for the storage networks. This paper presented a change of the priorities of the routing packages. The theory analysis and simulation tests all ensure that this optimization can improve the response speed of the IO requests and resource utilization. And more, it can be implemented easily in switches being sold.

2 Related Works

There are many works trying to improve the performance of the SAN. But they are most focused on the other two components of the SAN environments such as servers and storage devices. In this paper, we focus on the optimization of the switches which are used as the interconnect devices in the iSCSI storage system. As for switches, they have been well studied in the past two decades. In the traditional method, the switches internally works with fixed-size cells according to which a tremendous amount of scheduling algorithms have been proposed, such as [5][6][7][8]. The packet mode scheduling was first proposed in article[9], in which switches internally operate on variable-length packets and restrict that the packets are transferred without interruption. After then, many packet-based scheduling algorithms have been proposed, such as [10][11]. In [10], an algorithm which is stable for arbitrary admissible arrival processes was prented. In [11], it showed that making short packets which may preempt the long packets scheduled in high priority can reduce the average waiting time of packet and thus improve the switching performance. All the above works are helpful to promote the performance of the switch when used in normal intranet as well as storage network. But for being used in storage networks, they are too general since they mostly did not take storage networks for granted. In fact, we could do much optimization in the switch specially for the storage network. According to the characteristics of storage networks, we proposed an optimization of scheduling algorithm of switches to improve the performance of the whole storage system.

3 Optimization of the Switches

As mentioned above, the switches which can be used to connect servers can also be used in the storage networks. And more than that, the switches can not get any information about the environments they are in. So we can conclude that the switches being sold in the market must have not been optimized for being used in storage networks. Obviously, not distinguishing the purposes of switches,

being used in normal intranet or storage network, is good for the products costs and applicability. But in fact, when used in the real environments, people always need more performance and the switches are seldom required to connect both servers and storage devices. So, we may do some optimization in the switches when they are specially used in the storage networks.

To illustrate the characteristics of the storage networks which common switches are used in, consider a reading and a writing request in the iSCSI protocol. As for a reading request, it triggers the transmission of iSCSI Command PDU to the target. The target on receiving the command finds the requested data from the buffers and sends out a sequence of Data-In PDUs which contain the data requested. The initiator on receiving the Data-In PDUs, allocates the data into buffers. When all the data for the request have been transferred from the target to the initiator, the target sends an iSCSI Response PDU to the initiator, indicating successful completion of the command. Then both of the initiator and the target release the resources allocated for this request. The interaction of iSCSI Reading Commands is illustrated in fig. 1.

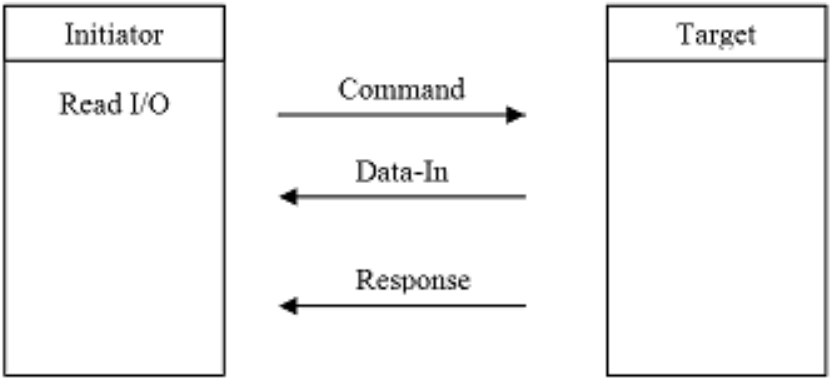

Fig. 1. Interaction of iSCSI Reading Commands

As for a writing request, the initiator transfers an iSCSI Command PDU to the target. After the target receives the command, it allocates buffers for transfer and responds with Ready to Transfer (R2T) PDUs, indicating permission for the initiator to transfer the data. The initiator responds to a R2T PDU by sending out a sequence of Data-Out PDUs which contain the data requested. The target on receiving the Data-Out PDUs, allocates buffer for data. When all the data for the request has been transferred from the initiator to the target, the target sends an iSCSI Response PDU to the initiator, indicating successful completion of the request. Then both of the initiator and the target release the resources allocated for this request. The interaction of iSCSI Writing Commands is illustrated in fig. 2.

From the read and write process of iSCSI protocol described above, we can see that there are two types of iSCSI PDUs transferred between the initiator

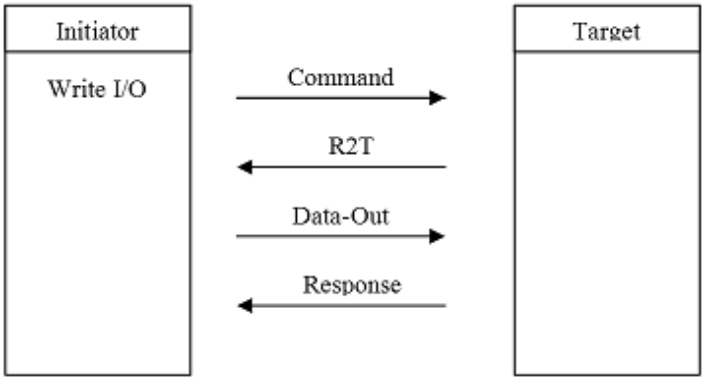

Fig. 2. Interaction of iSCSI Writing Commands

and the target. One type includes the commands which contain the reading or writing data, such as Data-In PDUs and Data-Out PDUs. The other type of the commands are the acknowledgement commands, for example, iSCSI Command PDU, Ready to Transfer (R2T) PDU and iSCSI Response PDU. Were these acknowledgements accepted, the resource allocated for the request would be released. For example, when all the data for the IO request has been transferred between the initiator and the target, the target sends an iSCSI Response PDU to the initiator. Only after the initiator receives this command, both of the initiator and the target can release the resources allocated for the IO request. So by intuition, we can assert that if the second type of commands have higher priority to be transferred, the time waiting for releasing resource will be reduced and the resource utilization will be improved.

We can also conclude that if we give the acknowledgement commands higher priority to be transfered, the performance will be exalted. This is because most of the commands, including the iSCSI command PDU, the Ready to Transfer (R2T) PDU and the iSCSI Response PDU, are so short that they have not extra room for reading or writing data. During the IO commands processing, if these short commands have the higher priority of being transfered, the performance of the system will be improved, which has been tested in other works [11].

4 Simulation Tests

Simpy [12] is used to test the performance promotion of the system with the optimization. Using Simpy, we build a simulation model which is shown in fig. 3. The model for storage networks simulates a closed queueing network, assuming that there are always 1000 requests in the system, 50% of which are reading requests and the others are writing requests. The arrival distribution of the requests is influenced by the time slot which is variable in the simulation model, which means that two requests arrived at the system with a equal time internal. It is assumed that each request takes equal resources of server. As for the switches,

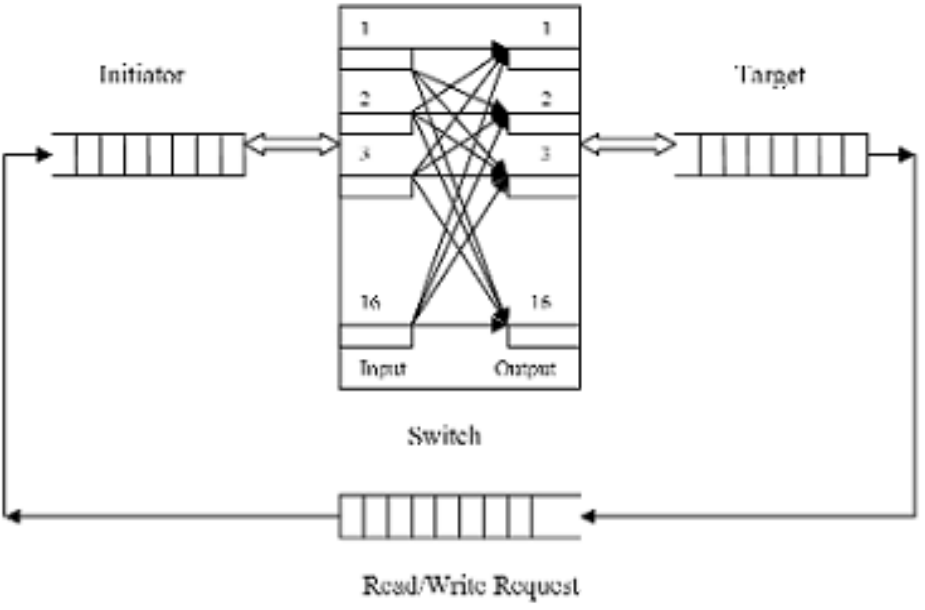

Fig. 3. Simulation Model

they are assumed to have 16 pairs of input and output interfaces and operate on the situation with or without the optimization. The resources of the server are infinite. Some simulation results are showed below.

Fig. 4 shows the comparison of the average processing time of the requests in the two systems. It indicates that when the arrival time slot is less than about 0.3, the processing time of the requsts with the optimization is much smaller. When the arrival time slot is more than 0.3, the performance of the two systems is nearly the same. It is because that the system is not busy at that time and there are less requests competing for the resources in the system.

Fig. 5 shows the comparison of the number of the request finished in the two systems. There are much more requests finished in the system with our optimization and the difference between the two systems becomes greater when the arrival time slot is smaller. It can be infered that when there are more competitions happened in the scheduling the optimization will play a dominant role in affecting the performance of the storage networks.

Next, we studied the average resource occupancy in the system whose results were shown in fig. 6. It indicates that the average resource occupancy of the system with optimization is lower when the arrival time slot is smaller. We can get that the busier the system is, the more effectively the resources are used. With the increase of the arrival time slot, the difference is less which nearly none when the arrival time slot is more than 0.3. It is because that there are less requests

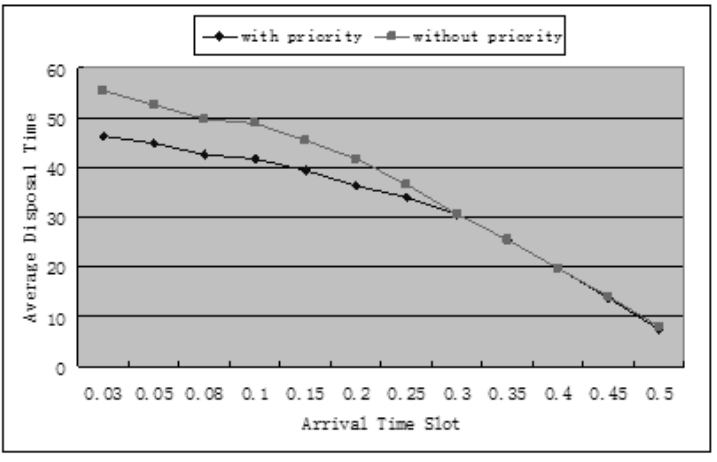

Fig. 4. Comparison of Average Processing Time

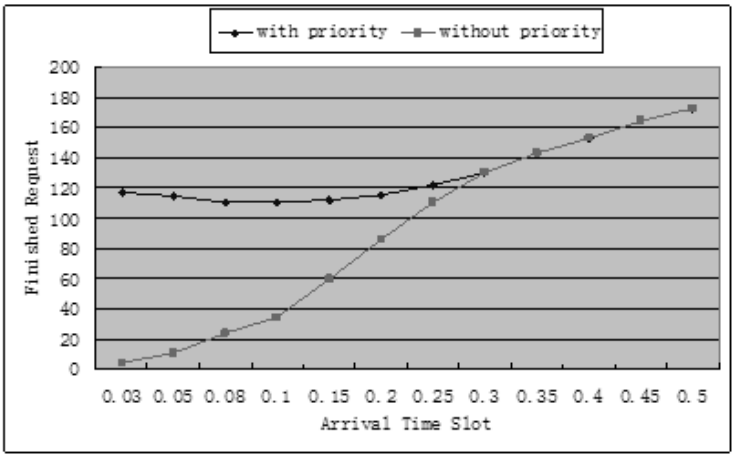

Fig. 5. Comparison of Finished Requests

waiting for being allocated in the server, so there are less requests competing for resources. It can be deduced that our optimiztion can improve the resource utilization of the storage networks.

The results are gained under the assuming that the arrival distribution of the requests is described as the time slot. But in reality, the IO requests happened at random, and the status which indicates in the simulation that two requests are generated more than 0.3 seconds is nearly rare. When there are a great lot of IO requests in the system, the improvement of the performance using the optimization will be more conspicuous.

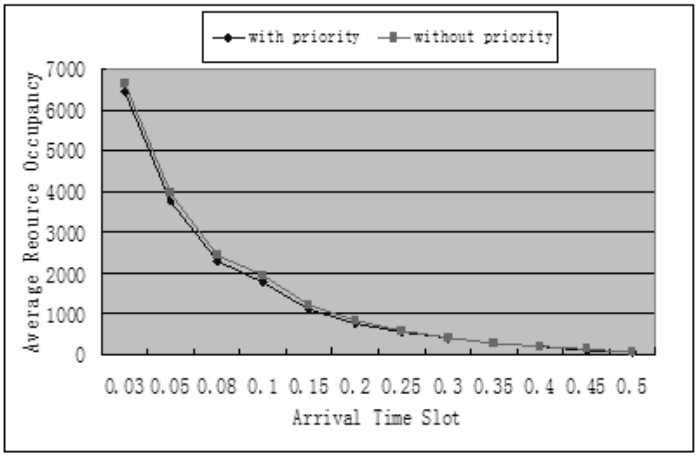

Fig. 6. Comparison of Average Resource Occupancy

5 Conclusions

This paper focused on improving the performance of the storage networks by optimizing the switches which are used as the interconnection devices. Common switches used in intranet can also be used in storage networks, so few optimizations special for the storage networks have been done. The processing steps of iSCSI commands are analyzed, and then based upon that, improving the transferring priority of the acknowledgement commands is proposed. This method can greatly promote the performance of the storage system and improve the resource utilization, for example buffers. Finally, a simulation model implemented by Simpy is constructed to test our conclusion.

Though we take iSCSI as an example, the optimization can also be used in other storage systems, such as FC-SAN. More than that, we wonder that if this optimization could effectively be applied in the whole IO path, including servers, switches, and storage devices. The further theoretical and experimental works are being done.

Acknowledgments. This research is supported by the National Natural Science Foundation of China (No: 60503055) and the Basic Research Foundation of Harbin Engineering University(No: HEUFT05011).

References

1. Yao Nian-Min, Shu Ji-Wu, Zheng Wei-Min: Improving the Data Placement Algorithm of Randomization in SAN. V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3516, pp. 415C422,2005.
2. D. A. Patterson, G. Gibson and R. H. Katz: A case for Redundant Arrays of Inexpensive Disks(RAID). In Proceedings of the 1988 ACM Conference on Management of Data(SIGMOD), pages 109-116, June 1988.

3. G. Weikum, P. Zabback, and P. Scheuermann: Dynamic File Allocation in Disk Arrays. In Proc. of ACM SIGMOD, pp. 406–415, May 1991
4. R. J. Honicky, E. L. Miller: A fast algorithm for online placement and reorganization of replicated data. 17th International Parallel and Distributed Processing Symposium (IPDPS), 2003.
5. N. McKeown, V. Anantharam, and J. Walrand: Achieving 100
6. P. Giaccone, B. Prabhakar, and D. Shah: Toward simple, high-performance schedulers for high-aggregate bandwidth switches. In Proc. IEEE INFOCOM, 2002, pp. 1160-1169
7. C. Partridge et al: A 50-Gb/s IP router. IEEE/ACM Trans. Networking, vol. 6, pp. 237-248, June 1998
8. T. Anderson, S. Owicki, J. Saxe, and C. Thacker: High speed switch scheduling for local area networks. ACM Trans. Comput. Syst., vol. 11, no. 4, pp. 319-352, Nov. 1993
9. M. A. Marsan, A. Bianco, P. Giaccone et al: Packet-mode scheduling in input - queued cell-based switches. IEEE/ACM Trans. Networking, vol. 10, no. 5, pp. 666 - 678. Oct. 2002
10. Y. Ganjali, A. Keshavarzian and D. Shah: Input queued switches: cell switching vs. packet switching. IEEE/ACM Trans. Networking, vol. 13, no. 4, pp. 782 - 789. Aug. 2005
11. Li Wen-Jie, Liu Bin: Preemptive Short-Packet-First Scheduling in Input Queuing Switches. Acta Electronic Sinica, vol. 33, no. 4, pp. 577-583. Apr. 2005
12. Develop group of Simpy: Purpose of the Simpy Laboratory pages. Simpy Homepage [Online]. Available at http://simpy.sourceforge.net