http://www.zju.edu.cn/jzus; E-mail: jzus@zju.edu.cn
ISSN 1009-3095 *Journal of Zhejiang University SCIENCE* V.5, No.1, P.1-7, Jan., 2004

Performance analysis of an iSCSI-based unified storage network*

FU Xiang-lin(傅湘林)[†1], ZHANG Kun(张琨)[2], XIE Chang-sheng(谢长生)[2]

([1] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*)

([2] *National Storage System Laboratory, School of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China*)

[†]E-mail: fu_xianglin@sina.com.cn

Received Jan.26, 2003; revision accepted May 20, 2003

Abstract: In this paper, we introduced a novel storage architecture "Unified Storage Network", which merges NAC(Network Attached Channel) and SAN(Storage Area Network), and provides the file I/O services as NAS devices and provides the block I/O services as SAN. To overcome the drawbacks from FC, we employ iSCSI to implement the USN(Unified Storage Network). To evaluate whether iSCSI is more suitable for implementing the USN, we analyze iSCSI protocol and compare it with FC protocol from several components of a network protocol which impact the performance of the network. From the analysis and comparison, we can conclude that the iSCSI is more suitable for implementing the storage network than the FC under condition of the wide-area network. At last, we designed two groups of experiments carefully.

Key words: iSCSI, Network Attached Channel (NAC), Unified Storage Networks (USN), NAS (Network Attached Storage), SAN (Storage Area Network)

Document code: A **CLC number:** TP303

INTRODUCTION

NAS is a term used to refer to storage elements that connect to a network and provide file access services to computer systems. In common usage, a NAS system is a special-purpose device that is designed to serve files to clients over a LAN. There are many benefits in the architecture: heterogeneous file sharing; internal resource pooling; exploitation of the existing infrastructure; simplicity of implementation; connectivity; improved manageability; reduction of total cost of ownership; and so on. But in practice, there are still many drawbacks: (1) the access speed is too low and not suitable for situations where high access speed is needed; (2) backup consumes bandwidth and other network resources of the LAN, even worse, it lowers the overall LAN performance greatly; (3) it can only consolidate the disks of the same NAS device, but cannot consolidate the disks of different NAS devices into a single storage pool; the storage must be managed separately, etc.

SAN is a network with primary purpose of transferring data between computer systems and storage elements and among storage elements. Currently the prominent building technology of the SAN is Fibre Channel (FC-SAN). The architecture introduces a number of opportunities: high performance, high availability, high scalability, improved manageability, storage sharing, and so on. Unfortunately, there are still several drawbacks preventing fast development of FC-SAN. Firstly, vendors of FC-SAN all have different implementations giving rise to compatibility problem of storage devices from different vendors. Now some vendors have made an alliance to increase the compatibility in the alliance, but the scope of the alliance is limited, so it is still inefficient. Secondly, to build SAN, experienced and special training professionals are necessary, and the management tool is insufficient. As a result, the cost of building and maintaining a SAN is too high to many enterprises. Additionally, the maximum distance a FC-SAN can spread across is limited to 10 km.

* Project supported by the National Natural Science Foundation of China (No.60173043) and the National Basic Research Program(973) of China(No. G1999033006)

NAS and SAN are different, but in fact they are complementary to each other to provide access to different types of data. They are used on different occasions. SAN is optimized for high-volume block-oriented data transfers while NAS is designed to provide data access at file level. It is possible that both NAS and SAN are needed in the same company. This paper introduces a novel storage architecture called USN (Unified Storage Network) to reduce the TOC (total of cost) and overcome the drawbacks of FC.

UNIFIED STORAGE NETWORK

Characteristics of USN

Firstly, USN(in Fig.1) uses iSCSI to build the storage network. iSCSI integrates the existing storage protocols (SCSI) with the IP protocol directly. With the integration, the storage and network can be merged seamlessly. That means people can use IP networking devices (hub, switch, router and so on) to build native IP-based storage network. It simplifies the implementation of the storage network and decreases the total cost greatly. Currently IP and Ethernet are used for all form of networking, and the skills that people have developed to control and manage IP network will be applied to the USN, where the existing software applications and tools can be used well. Especially, with iSCSI, USN can spread across the MAN/WAN and break the distance limitation of FC-SAN (10 km).

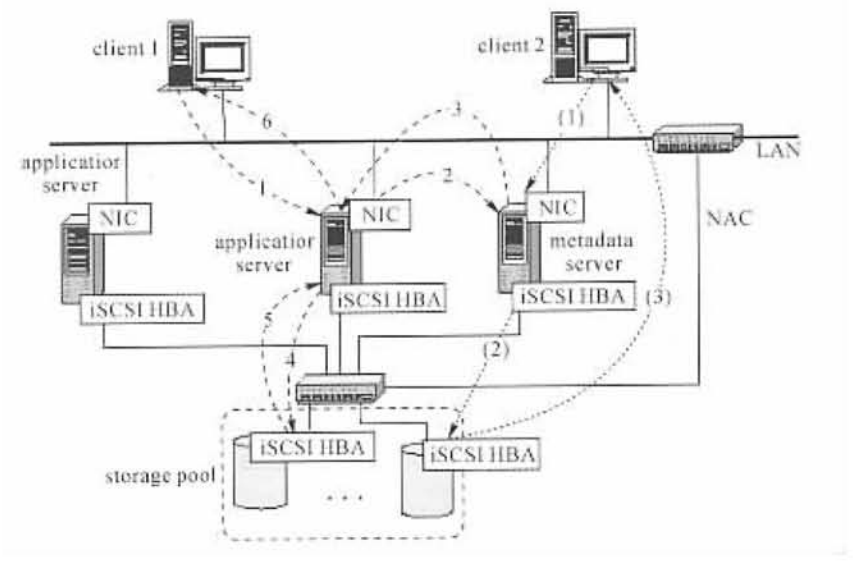

Fig.1 Architecture and communication pattern of USN

Secondly, USN merging of NAS and SAN provides file I/O and block I/O services simultaneously, so that all the storage space can be divided into the FILE space and BLOCK space. USN merging of LAN and NAS, reaps benefits of the NAS: heterogeneous file sharing; internal resource pooling; exploitation of the existing infrastructure; simplicity of implementation; extensive connectivity; improved manageability and so on. Different from the traditional NAS, it changes the tight coupling between the OS and the storage subsystem and separates the OS (file system) component from the storage system. In USN, there is a special server to perform the metadata managing tasks. It can also supplies the block I/O services to the client directly. In addition, there are application servers to supply the file I/O services to the client directly. In USN, the storage subsystem adopts the SAN model; all storage devices can be connected to each other through network but not through bus. So USN has the benefits of the SAN, for example, the high scalability and so on. Especially, all storage devices can be virtualized and consolidated to a single storage pool by the storage virtualization.

Most importantly, a high speed NAC (Network Attached Channel) is introduced in USN,

which connects storage devices and LAN directly and moves data between the storage devices and the client without using the server. When the USN provides block I/O services, if data block must be processed by the metadata server before transferring to client, it is sent to the metadata server; if not, it can be transferred to the client via the NAC without using the metadata server. It can decrease the workload of the metadata server, and shorten the response time of the client request. This approach enables direct data transfer between the client and storage subsystems.

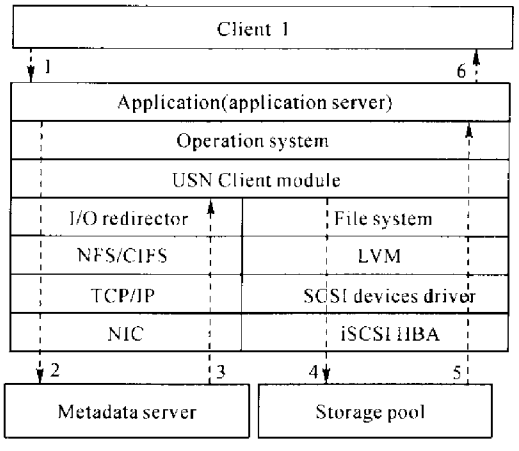

Fig.2 Software layers of the application server

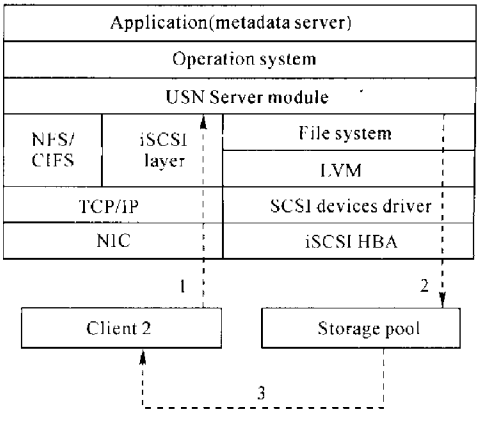

Fig.3 Software layers of the metadata server

The communication pattern

In practice, two special software modules must be developed and installed on the application server (USN Client Module) and the metadata server (USN Server Module), respectively.

USN can provide both file I/O and block I/O services. In regard to the file I/O, all the file I/O requests sent to the application server OS are intercepted by the USN Client Module, and transferred to metadata server via LAN. The communication patterns are displayed in Fig. 2: (1) client 1 sends a file I/O request to the application server via the LAN; (2) when the file I/O request arrives at the application server, the application server has no knowledge about the metadata of the file. So the USN Client Module on the application server intercepts and forwards the request to the metadata server via LAN; (3) after the request has arrived at the metadata server, the metadata server (USN Server Module) performs such functions as: security check and authentication check; provides a locking mechanism, prevents the current file from being accessed by the subsequent applications and causing consistency problem; returns the metadata information back to the application server and so on; (4) the application server (USN Client Module) accepts the metadata and sends the corresponding block I/O request to the storage subsystems; (5) the storage subsystems performs the block I/O operations. If the request is a write operation, the data are written into the storage subsystem and the application server is acknowledged. If the request is a read operation, the data blocks retrieved are sent to the application server; (6) for read operation, the block data returned from the storage subsystem is transformed to file by the file system of the application server; (7) The application server returns the file to client 1.

USN can also use iSCSI protocol to provide the block I/O service. The communication patterns are shown in Fig. 3. (1) Block I/O requests (SCSI commands) are sent from client 2, encapsulated by the iSCSI HBA on client 2, and become IP packets (iSCSI commands) which are transmitted in the TCP/IP network. (2) Metadata server (USN Server Module) accepts iSCSI commands, and performs the functions such as authentication, authorization, allocation, lock and so on, then forwards the iSCSI commands to the corresponding storage subsystems. (3) In the storage subsystem, the iSCSI commands are de-encapsulated and become the SCSI com-

mands. If the SCSI command is a write operation, the storage subsystem performs the operation, and the metadata server sends the relevant metadata to the client. If the SCSI command is a read operation, the storage subsystems performs the operation, and the retrieved data blocks are encapsulated by the iSCSI HBA and converted to the IP packets again. The IP packets are transmitted to client 2 via the high-speed NAC directly, de-encapsulated by its iSCSI HBA of the client 2 and become the block data needed by the client 2. In the model, the functions of the file system are finished by the client 2 and metadata server respectively, the client 2 finishes the mapping between the file and block, and the directory operations must be finished by the metadata server to maintain a single and global view of all the BLOCK spaces, so the metadata of all the data in the BLOCK space must be stored in the client and metadata server simultaneously.

ANALYSIS OF iSCSI

Overview

iSCSI is an emerging Internet Engineering Task Force (IETF) specification that defines how SCSI and Ethernet work together to perform SCSI data transfers across TCP/IP networks. It allows the block storage SCSI commands to be carried by the standard TCP/IP protocols over Ethernet wires. iSCSI is the convergence of SCSI, the dominant protocol for block storage I/O, with IP, the dominant protocol for computer internetworking and leverages the existing TCP/IP infrastructure. With the development of the TCP/IP, the IP-based network is more and more becoming the main infrastructure of the Internet and Intranet. Building the storage network with the iSCSI protocol fits into the development trend. To evaluate whether iSCSI protocol is suitable for building the storage network, we analyze the protocol and compare it with the Fibre Channel.

I/O channel overhead

In TCP/IP, data are copied from the NIC into the kernel memory first, then the user application memory. But in the case of zero copy, the data is copied directly from the NIC into the user application memory without going through the kernel memory. The benefits of zero copy are obvious, it shortens the path length of the data flow and decrease the CPU utilization of the host. In FC, the FC Host Bus Adapter performs the zero copy processing with the specialized tag aproaches (Kaladhar and Prasenjit, 2001). The TCP protocol is a stream-based protocol, and it is difficult to implement the zero copy semantics because the TCP segments could be spread across multiple Ethernet frames (Julian, 2003).

The block I/O request issued by the storage application normally varies in size from 4 K to 64 K; correspondingly the size of the Ethernet (iSCSI) frame and FC frame is 1.5 K and 2 K. If the storage application has issued a block I/O request for a 16 K block, fragmentation and assembly must be performed for the request. There are more overheads with fragmenting and assembling the smaller frame size than the larger frame size. On the other hand, in FC, the FC host adapter performs the fragmentation and assembly processing and thus, they offload these operations from the host CPU. In TCP/IP, the same operations performed by the host increases the overhead and CPU utilization of the host machine. Therefore, from the point of the overhead on the I/O channel, the iSCSI is less suitable than the FC for building the storage network. In most cases, the above mentioned issues can be resolved by using specialized host adapter to implement the iSCSI, which have built-in-zero-copy support, and performs the fragmentation and assembly processing, etc.

Flow and congestion control mechanism

FC uses a credit based flow control mechanism (Roger and Dal, 2001; Kaladhar and Prasenjit, 2001). The receiver allocates the credits to the sender, when it has the necessary buffer space to store the sender's data. It allocates the credits on the basis of the sender's request. Only when the sender has not used up its credits, can it send data. The receiver returns the credits to the sender when it sends an acknowledgement to the sender. The credit based flow control mechanism ensures that there is never dropping of packets due to data congestion. In iSCSI, the end to end flow control mechanism is employed (Julian, 2003). That is, the two endpoints of a connection negotiate a windows size that is based on the buffer space available at

their respective ends. The window represents the number of messages that can be sent without receiving an acknowledgment from the receiver. In iSCSI, the congestion will occur at the networking devices and the end-points. iSCSI reacts to congestion by dropping packers. The end to end flow control mechanism of iSCSI is more scalable than the credit-based flow control mechanism of FC. In the WAN/MAN, the sender has to wait for a long time to get credits from the receiver for injecting new data into the network; thus the credit-based flow control mechanism decreases the utilization of the network and it is only adequate when the network delay is small. However, the end to end flow control mechanism of iSCSI is more suitable than the FC in the wide-area network, because the senders can dynamically increase or decrease their data transfer rates at the expense of packer drops at the network nodes during periods of high congestion. Currently the distance of SAN spread across is getting further; so the advantages of iSCSI are obvious.

Discovery mechanism

In FC, when a new device comes on-line, it contacts its fabric manager. The fabric manager, in turn, informs all the devices that have registered with the fabric manager and those that want to be informed about this event (Steven and Scott, 2001). Furthermore, in FC, when a device comes on-line, it performs a login with all the other devices that are present in the same zone, the switch to which this device connects informs all the other switches in the fabric about the event. Therefore this mechanism can result in the sending of many messages. Environments with thousands of switches and devices lower the performance of the network greatly. Therefore the FC is only suitable for the smaller scale network. In iSCSI, an iSCSI initiator can discover an iSCSI target in the following different ways (Julian, 2003): (1) By configuring the target's address on the initiator; (2) By configuring a default target address on the initiator and the initiator connects to the target and requests a list of iSCSI Names, via a separate SendTargets command; (3) By issuing Service Location Protocol (SLP) multicast requests, to which the targets may respond; (4) By querying a storage name server for a list of targets that it can access. In large network with thousands of devices, the storage node will use the mechanism of querying a storage name server rather than the multicast approach. Once the initiator receives the IP address and TCP port number of the target from the storage name server, the initiator establishes a connection. For addressing, FC employs the 24 bit address, and iSCSI employs the 128 bit address. Thus the iSCSI is more scalable than the FC. Therefore, from the point of view of discovery mechanism and addressing, the iSCSI is more suitable than the FC in the wide-area networks.

Timeout and retranslation mechanism

iSCSI uses an adaptive timeout and retranslation mechanism of the TCP/IP protocol stack (Julian, 2003). In TCP/IP protocol, TCP monitors the performance of each connection and deduces the reasonable values of timeout. As the performance of a connection changes, TCP revises its timeout value. Every time, TCP records the time at which the segment is sent, and the time at which an acknowledgement arrives for the data in the segment. From the two times, TCP computes an elapsed time known as a sample round trip time or round trip sample, whenever it obtains a new round trip sample, TCP adjusts its notions of the average round trip time for the connection. Usually, TCP software stores the estimated round trip time, RTT, as a weighted average and use new round trip samples to change the average slowly. Whereas the FC has a static timeout and retranslation mechanism (Kaladhar and Prasenjit, 2001), which does not adjust itself dynamically according to the network conditions, so the sender may timeout and retranslate the message either too soon or too late, and this can negatively impact the overall performance

Conclusion of the section

In conclusion, for traditional LAN-based SAN, FC is more suitable for internetworking protocol than the iSCSI because of its zero-copy and the fragmentation and assembly mechanism. But with the growth of storage applications, the storage network can be deployed across longer distances (Garth and Rodney, 2000). iSCSI is more suitable because of its flow and congestion control mechanism, discovery and addressing mechanism, timeout and retranslation mechanism and so on.

PERFORMANCE EVALUATIONS

Methodology

To evaluate the performance of our solution, we build an experimental USN and compare it with the existing FC-SAN of our lab. Additionally, we employ a software packet to simulate the iSCSI HBA. Currently, Intel has implemented the iSCSI under Linux using the software approach (Intel iSCSI protect, 2002). In these experiments, we employ Intel's iSCSI target for the sake of the simplicity. However, considering that the existing FC-SAN is under Windows 2000, we implement the iSCSI initiator based on Windows 2000 by using the combination of iSCSI miniport driver and class filter driver, to implement the kernel network transport by TDI. Therefore, we perform two tests. The first is for USN, and the second is for the existing FC-SAN. Fig.5 gives a highly simplified view of the two storage networks. To test the performance of iSCSI, the network bandwidth bottleneck should be removed, so in the experiment, the 1000 M Ethernet is employed.

To compare fairly, we try our best to create similar experiment environment. In the experiment, there are three host machines whose configurations are shown in Table 1. Host 1 acts as both application server and metadata server that connects the IP-based switch with the traditional NIC, and additionally, simulates the iSCSI HBA (Initiator) as shown in Fig.4a. Host 2 simulates the iSCSI-supported storage device that connects the switch with the traditional NIC and connects the local disk drive with SCSI controller. Host 3 acts as application server that connects the FC switch (HP-FC16) with the FC HBA, as shown in Fig.4b. On the other side of the FC16 is the FC-based disk array (HP-FC60 disk array).

Table 1 The machine configurations

	CPU	RAM	OS	Disk	NIC/HBA
Host 1	PIII 450 (double)	256MB	Windows2000	Maxtor 91020D6 (IDE)	AGE-1000SX (NIC)
Host 2	PIII 450 (double)	256MB	Linux 7.1	ST318437LW(SCSI)	AGE-1000SX (NIC)
Host 3	PIII 450 (double)	256MB	Windows2000	ST318437LW(SCSI)	Qlogic 300

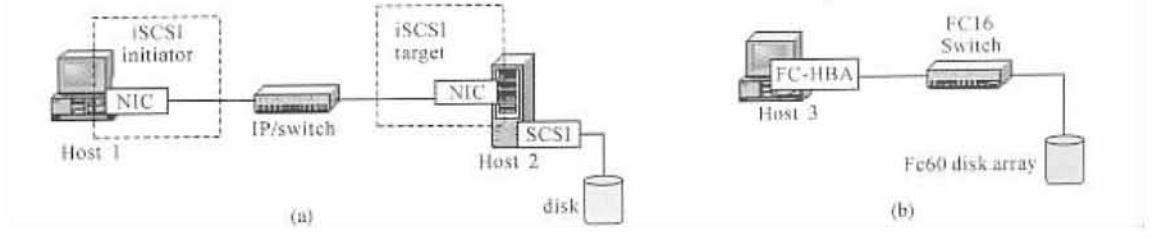

Fig.4 Simplified view of the experiment
(a) Host 1 and Host 2; (b) Host 3

Results and discussions

In the experiments, we used the Iometer as measurement tool (current version is 1998.10.08. Intel does not endorse any Iometer results). Fig.5 and Fig.6 show the effect of block size on the throughput. As seen from Fig.5, when the transfer request size is increased, the throughput (MB/s) is increased (but the IO/s is decreased). Most importantly, the throughput of iSCSI is higher than that of FC. As observed in Fig.5, when the transfer request size is small (smaller than 64K), the throughput of iSCSI is obviously higher than that of FC. In most cases, the write/read requests issued by the application is decomposed into small I/O requests with sizes of 4 K – 64 K (Xavier *et al*., 2000). So from the point of view of throughput, iSCSI is suitable for implementing storage networks. When the transfer request size is larger than 64 K, the difference of throughput is decreased. At this point, the performance of the storage subsystem becomes more and more dependent on the disk controller. When the transfer request size is 1024 K, the throughputs of iSCSI and FC become close. It can be understood easily that, in

the experiment, to create similar experiment environment, the maximum throughput of the two disk subsystems are similar, and that when the transfer request size reaches 1024 K, the throughput of the storage subsystem reaches the maximum throughput of the disks subsystems. In fact, after the throughput reaches a maximum, increasing transfer request size does not guarantee increased throughput, but decreases the throughput instead.

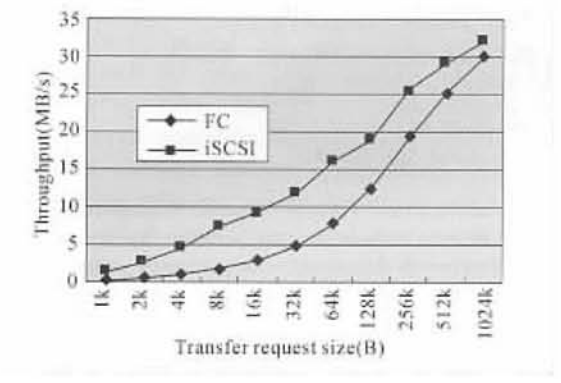

Fig.5 Throughput (MB/s) vs transfer request size

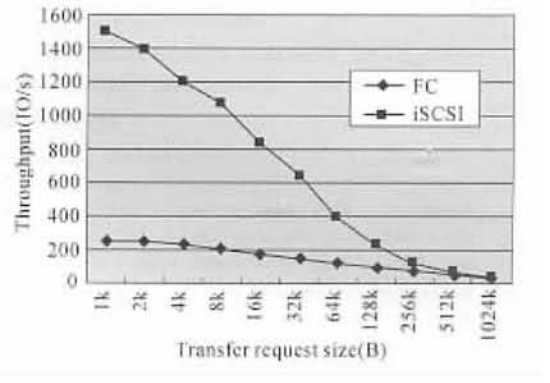

Fig.6 Throughput (IO/s) vs transfer request size

Fig.7 displays the effect of transfer request size on the average response time. Notice that

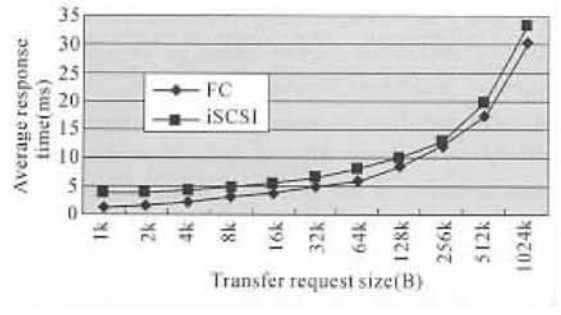

Fig.7 Average response time vs transfer request size

when the transfer request size is increased, the average response time is increased. When the transfer request size is more than 1024 K, the performance of the storage subsystem decreased markedly. Most importantly, when the transfer request size is smaller than 1024 K, the average response time of iSCSI is lower than that of FC. It indicates that the performance of iSCSI is better than FC in terms of average response time.

CONCLUSIONS

From Table 2 and Fig.5, it can be concluded that when iSCSI and FC are employed to implement the storage network, the performance of iSCSI is better than that of the FC from the point of throughput and average response time (but the conditions of the two experiments are different). Most importantly, in most cases the read/write request initiated by the application is transmitted into the I/O request with the 64 K block size. Fig.5 shows that the throughput of the iSCSI is bigger and the average response time is smaller than those of the FC when the block size is 64 K, so we can conclude that the iSCSI is more suitable for implementing the storage network; and that the architecture of the USN is reasonable and feasible.

References

Garth, A. and Rodney, M., 2000. Network attached storage architecture. *Communication of the ACM*, **43**(11): 37 – 45.

Intel iSCSI protect, 2002. https://sourceforge.net/projects/intel-iscsi

Julian, S., iSCSI Standard, 2003. IETF RFC2026; http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-13.pdf

Kaladhar, V. and Prasenjit, S., 2001. An analysis of three gigabit networking protocols for storage area networks. *IEEE Computer*, **29**(4): 259 – 265.

Roger, C. and Dal, A., 2001. Fibre Channel - Generic Services 3(FC-GS-3), ANSI NCITS 348.

Steven, W. and Scott, K., 2001. Fibre Channel - Switch Fabric - 2(FC-SF-2), ANSI NCITS 355 – 2001.

Xavier, M., Federico, S. and Vicente, S., 2000. Performance Analysis of Storage Area Networks Using High-Speed LAN Interconnects. Processing of the 8th ISPAN Conference. IEEE Computer Society, Texas, USA, p. 474 – 478.