PMC-SIERRA

Storage Area Network Architectures

Technology White Paper

Heng Liao
Technical Advisor

Issue 1: April, 2003

PMC-2022178

Abstract

This PMC-Sierra white paper provides an architectural overview of storage technology, SAN switches and Storage Elements. The systems discussed are partitioned into the functional building blocks needed to address individual functional requirements of the systems. Finally the paper explores the technology trends in the SAN industry and their evolution with the changing trend in IC technology and SAN protocols.

About the Author

Heng Liao is a Technical Advisor in the Product Research Group and oversees ICs for Enterprise Networks and Storage Systems at PMC-Sierra, Inc. Heng has a Ph.D. in Computer Engineering. Prior to joining PMC, he has worked at Princeton University on VLIW video signal processor architecture. Heng's interest includes storage systems, switching, switch fabrics, traffic management technologies, network processor architecture, microprocessor architectures, instruction level parallelism, cluster computing, storage systems, IP layer processing, flow classification. He has 14 patents issued or pending and several published papers. Heng Liao is an Adjunct Professor at the Department of Computer Science and Technology, Tsinghua University, China.

Revision History

Issue No.	Issue Date	Details of Change
1	April, 2003	Document created

Contents

List of Figures

List of Tables

1 Introduction

This white paper provides an overview of key technologies that have evolved around data storage and storage networking. The paper focuses on analyzing the system architectures of the different building blocks of storage networks.

In recent years, enterprise data storage has seen explosive growth in demand from users. This growth is driven by increasingly more sophisticated applications that generate more rich and numerous quantities of content data, and an increasingly larger number of users/consumers of this rich content data. The rapid advancement of networking technology both in the LAN and the WAN has enable new applications that generate large demands on data storage. The rapid growth of information content is fueled by a profound change in the underlying infrastructure of computer networks that facilitates acquisition, processing, exchange and delivery of information among the processing units and the users.

These new applications drive the data storage demand in the following areas:

- ***Capacity*** – the amount of data storage space is increasing. The historic data shows the growth of enterprise data storage has surpassed the exponential growth rate projected by Moore's law (doubling every 18 months).

- ***Performance*** – the bandwidth for delivering storage content is growing to match the increased speed of computer processing power, the speed of data communication networks, and the speed requirement of emerging applications such as multimedia applications.

- ***Availability*** – as people and enterprises become more and more reliant on the content in the data storage, the reliability and availability of data storage systems networks must be dramatically increased to prevent the severe consequences that may result from loss of data content and loss of access to data. Mission critical storage networks are required to achieve "5-9's " (99.999%) availability and the capability to recover from catastrophic disasters via mirroring and backup techniques that protect the content through geographic diversity.

- ***Scalability*** – the data storage solution must not only be able to satisfy the current storage requirements, but also be easy to grow to address the increased demand of future applications.

- ***Cost*** – the cost of ownership needs to be reduced. That includes not only the cost of hardware system, but also the cost of maintaining and managing the data storage.

Driven by the above requirements, various storage-networking technologies have undergone a fairly rapid adoption to become mainstream enterprise solutions. This paper provides a brief introduction to various storage models and technologies. In addition, it provides an overview of various functional entities involved in building storage networks and the reference hardware architectures.

2 Storage Models

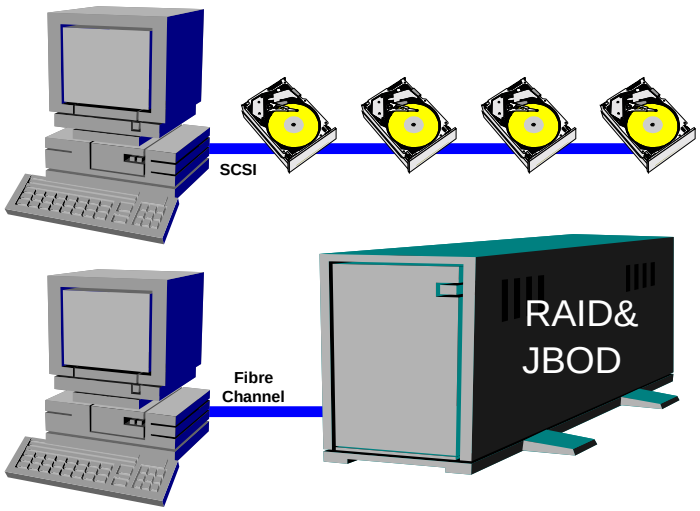
The following discussion examines three categories of data storage technologies including Direct Attached Storage (DAS), NAS (Network Attached Storage), and SAN (Storage Area Networks). Out of this comparison will appear the benefit of sharing storage resources over the network, and how different schemes can be used to accomplish the task of sharing storage resources.

2.1 Direct Attached Storage (DAS)

Direct attached storage is the simplest and most commonly used storage model found in most standalone PCs, workstations and servers. A typical DAS configuration consists of a computer that is directly connected to one or several hard disk drives (HDDs) or disk arrays. Standard buses are used between the HDDs and the computers, such as SCSI, ATA, Serial-ATA (SATA), or Fibre Channel (FC). Some of the bus cabling definitions allow for multiple HDDs to be daisy chained together on each host bus adapter (HBA), host channel adapter, or integrated interface controller on the host computer.

Two examples of DAS systems are given in Figure 1. In the first example, 4 SCSI HDDs are attached to the host computer via a daisy chain of SCSI cabling. The second example uses Fibre Channel cabling to connect the host computer and a RAID/JBOD[1] storage system together.

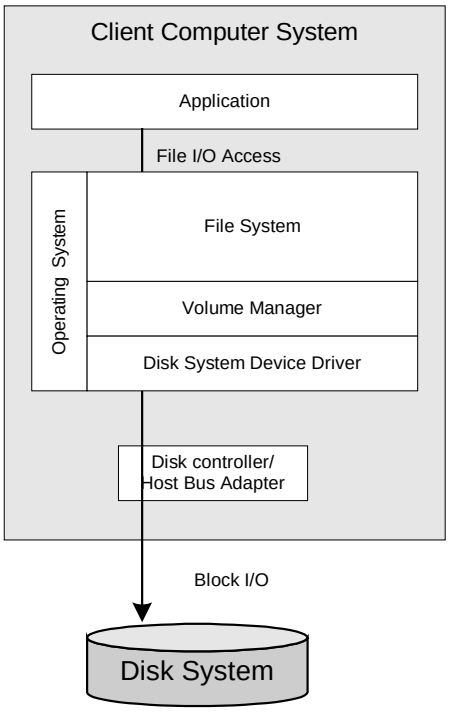
Figure 1 Direct Attached Storage



[1] JBOD – Just a Bunch Of Disks. See Section 3.2.2 for more details.

DAS is a widely deployed technology in enterprise networks. It is easy to understand, acquire and install, and is low cost. It is well suited to the purpose of attaching data storage resources to a computer or a server when capacity, administration, backup, high-availability, high performance are not key requirements. For home PC and small enterprise network applications, DAS is still the dominant choice, as the low-end requirements for growth in capacity, performance and reliability can be easily addressed by the advancements in HDD and bus technologies. The past few years have seen 2x sequential increase in HDD capacity per year, while maintaining the low cost point of HDDs targeting the personal computer market. The advent of Ultra-ATA, SATA, SATA-2, and Serial Attached SCSI (SAS) and FC bus standards alleviates the performance bottleneck on the bus interface. The quality of the HDDs has also much improved over the years. These technology advancements have helped DAS systems address the requirements of low-end data storage users.

Figure 2 DAS Software Architecture



The software layers of a DAS system are illustrated in Figure 2. The directly attached storage disk system is managed by the client operating system. Software applications access data via file I/O system calls into the *Operating System*. The file I/O system calls are handled by the *File System*, which manages the directory data structure and mapping from files to disk blocks in an abstract logical disk space. The *Volume Manager* manages the block resources that are located in one or more physical disks in the *Disk System* and maps the accesses to the logical disk block space to the physical volume/cylinder/sector address. The *Disk System Device Driver* ties the *Operating System* to the *Disk controller* or *Host Bus Adapter* hardware that is responsible for the transfer of commands and data between the client computer and the *disk system*. The file level I/O

initiated by the client application is mapped into block level I/O transfers that occurred over the interface between the client computer and the disk system.

One of the key characteristics of DAS is the binding of storage resources to the individual computers/servers. The shortcomings from such a resource binding become apparent when applications demand higher requirements on the data storage. The DAS suffers from the following severe limitations.

The storage capacity of the DAS is limited by the number of HDDs supported by the bus (e.g. 15 devices for SCSI). Adding/removing a disk drive may disrupt the access to all the disks on the SCSI chain, thus making the storage resource unavailable for the duration of the maintenance period. The maximum capacity of a DAS system tops out when the SCSI bus is loaded with the maximum number of HDDs supported.

The efficiency of the storage resource is low, as the storage capacity is bound to a given computer/server. The distributed nature of the storage resource not only means more content replication, but also means the free resources on one computer can not be used by another computer/server whose disk space is running low. The computer service department of an enterprise has to constantly monitor the disk space usage level of each computer to add disks to individual computers or move data around manually to ensure the request for disk space is satisfied. This quickly becomes an administrative nightmare as the number of computers in the enterprise grows.

The availability of storage content of DAS is limited – any server failure results in the content on the attached storage resources becoming inaccessible. If the storage resource is decoupled from the server, then a backup server can be used to take over control of the storage and provide access to the data.

The performance of DAS applications is limited by the processing speed of the individual server. As the content is only accessible by the attached server, parallel processing to share the workload among multiple servers is not possible.

The maintenance work on a large computer network consisting of DAS systems is tedious. To protect the data on the DAS systems, backup/recovery of data is required for each computer. This is a time-consuming process that affects the performance of the computers, but also requires a lot of human intervention. Repairing the failures on the individual computers requires even more manual work. All these factors increase the total cost of ownership of DAS systems.

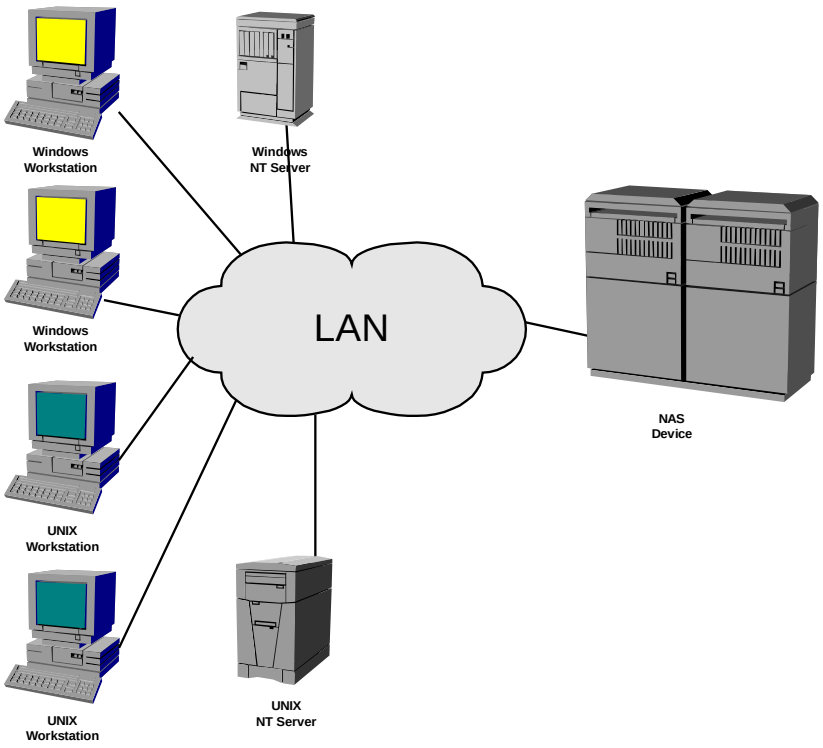
2.2 Network Attached Storage (NAS)

After seeing the consequences of binding storage to individual computers in the DAS model, the benefits of sharing storage resources over the network become obvious. NAS and SAN are two ways of sharing storage over the network. NAS is generally referred to as storage that is directly attached to a computer network (LAN) through network file system protocols such as NFS and CIFS.

The difference between NAS and SAN is that NAS does “file-level I/O” while SAN does “block-level I/O” over the network. For practical reasons, the distinction between block level access and file level access is of little importance and can be easily dismissed as implementation details.

Network file systems, after all, reside on disk blocks. A file access command referenced by either the file name or file handle is translated into a sequence of block access commands on the physical disks. The difference between NAS and SAN is in whether the data is transferred across the network to the recipient in blocks directly (SAN), or in a file data stream that was processed from the data blocks (NAS). As the file access model is built on a higher abstraction layer, it requires an extra layer of processing both in the host (file system redirector) computer, and in the function of translation between file accesses and block accesses in the NAS box. The NAS processing may result in extra overhead affecting the processing speed, or additional data transfer overhead across the network; both can be easily overcome as technology advances with Moore's law. The one overhead that can not be eliminated is the extra processing latency, which has direct impact on the performance of I/O throughput in many applications. Block level access can achieve higher performance, as it does not require this extra layer of processing in the operating systems.

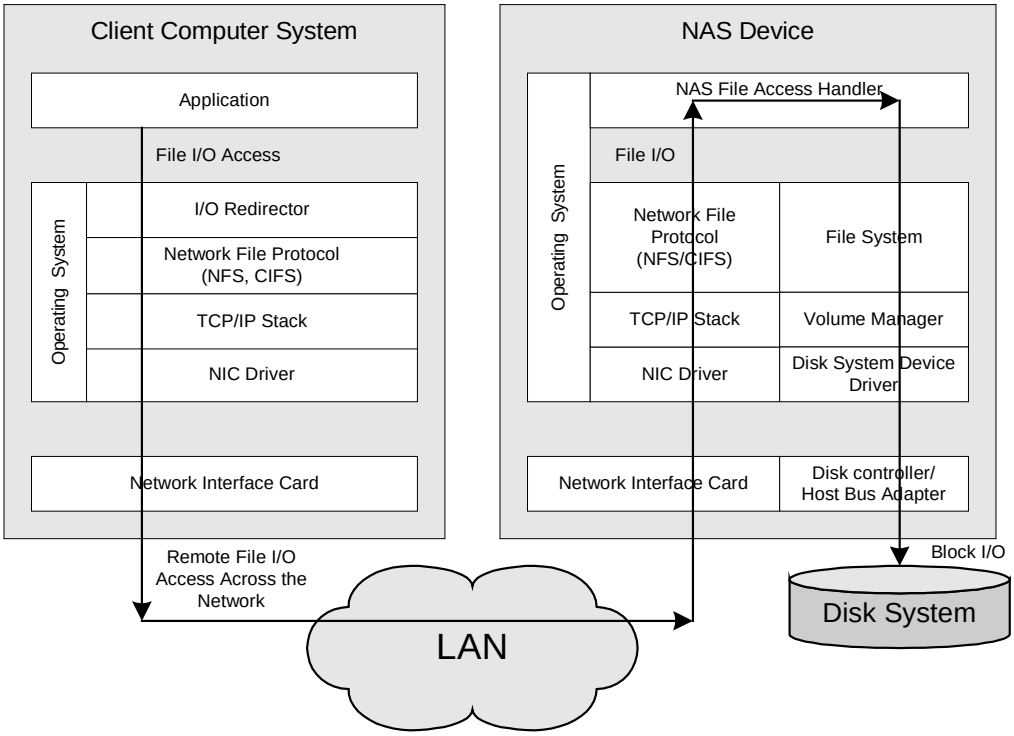
Figure 3 Network Attached Storage (File Oriented)



The benefit that comes with the higher layer abstraction in NAS is ease-of-use. Many operating systems, such as UNIX and LINUX, have embedded support for NAS protocols such as NFS. Later versions of Windows OS have also introduced support for the CIFS protocol. Setting up a NAS system, then, involves connecting the NAS storage system to the enterprise LAN (e.g. Ethernet) and configuring the OS on the workstations and servers to access the NAS filer. The many benefits of shared storage can then be easily realized in a familiar LAN environment without introducing a new network infrastructure or new switching devices. File-oriented access also makes it easy to implement a heterogeneous network across multiple computer operating system platforms. An example of NAS is shown in Figure 3. In this example, there are a number

of computers and servers running a mixture of Windows and UNIX OS. The NAS device attaches directly to the LAN and provides shared storage resources.

Figure 4 NAS Software Architecture



The generic software architecture of NAS storage is illustrated in Figure 4. Logically, the NAS storage system involves two types of devices: the client computer systems, and the NAS devices. There can be multiple instances of each type in a NAS network. The NAS devices present storage resources onto the LAN network that are shared by the client computer systems attached to the LAN. The client *Application* accesses the virtual storage resource without knowledge of the whereabouts of the resource.

In the client system, the application File I/O access requests are handled by the client *Operating System* in the form of systems calls, identical to the systems calls that would be generated in a DAS system. The difference is in how the systems calls are processed by the *Operating System*. The systems calls are intercepted by an *I/O redirector* layer that determines if the accessed data is part of the remote file system or the local attached file system. If the data is part of the DAS system, the systems calls are handled by the local file system (as described in Section 2.1 above). If the data is part of the remote file system, the file director passes the commands onto the *Network File System* Protocol stack that maps the file access system calls into command messages for accessing the remote file servers in the form of NFS or CIFS messages. These remote file access messages are then passed onto the TCP/IP protocol stack, which ensures reliable transport of the message across the network. The NIC driver ties the TCP/IP stack to the

Ethernet Network Interface card. The *Ethernet NIC* provides the physical interface and media access control function to the LAN network.

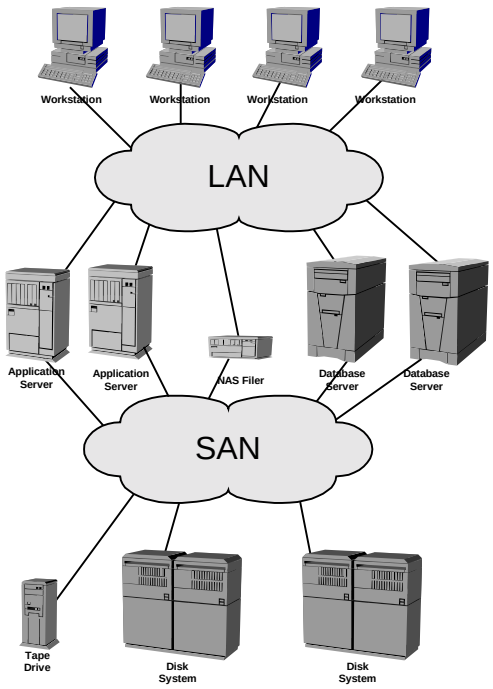
In the NAS device, the Network Interface Card receives the Ethernet frames carrying the remote file access commands. The NIC driver presents the datagrams to the TCP/IP stack. The TCP/IP stack recovers the original NFS or CIFS messages sent by the client system. The NFS file access handler processes the remote file commands from the NFS/CIFS messages and maps the commands into file access system calls to file system of the NAS device. The NAS file system, the volume manager and disk system device driver operate in a similar way as the DAS file system, translating the file I/O commands into block I/O transfers between the *Disk Controller/ HBA* and the *Disk System* that is either part of the NAS device or attached to the NAS device externally. It is important to note that the Disk System can be one disk drive, a number of disk drives clustered together in a daisy-chain or a loop, an external storage system rack, or even the storage resources presented to a SAN network that is connected with the HBA of the NAS device. In all cases, the storage resources attached to the NAS device can be accessed via the HBA or Disk controller with block level I/O.

2.3 Storage Area Network (SAN)

SAN provides block-orient I/O between the computer systems and the target disk systems. The SAN may use Fibre Channel or Ethernet (iSCSI) to provide connectivity between hosts and storage. In either case, the storage is physically decoupled from the hosts. The storage devices and the hosts now become peers attached to a common SAN fabric that provides high bandwidth, longer reach distance, the ability to share resources, enhanced availability, and other benefits of consolidated storage.

Figure 5 gives an example of a typical SAN network. The SAN is often built on a dedicated network fabric that is separated from the LAN network to ensure the latency-sensitive block I/O SAN traffic does not interfere with the traffic on the LAN network. This examples shows an dedicated SAN network connecting multiple application servers, database servers, NAS filers on one side, and a number of disk systems and tape drive system on the other. The servers and the storage devices are connected together by the SAN as peers. The SAN fabric ensures a highly reliable, low latency delivery of traffic among the peers.

Figure 5 Storage Area network (Block Oriented)

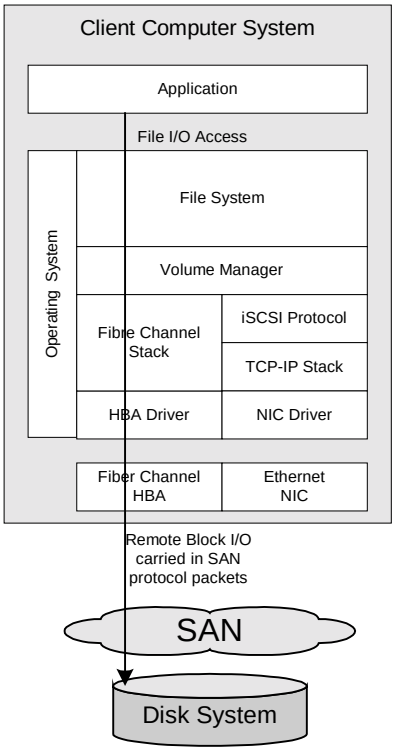


Although it is possible to share the network infrastructure between LAN and SAN in an iSCSI environment, there are a couple of reasons for maintaining the separation. First of all, the LAN network and the SAN network often exist in physically different parts of the Enterprise network. The SAN network is often restricted to connecting the servers and the storage devices that are typically located close to each other in a centralized environment. The LAN often covers the connectivity between the servers and the desktop workstations or PCs, which spans a much wider area in the enterprise. Secondly, the traffic load on the LAN and the SAN, and the Quality of Service requirement are different. The SAN traffic typically demands higher dedicated bandwidth and higher availability with lower latency, which is difficult to ensure in a LAN network. Additionally, the SAN many create high bandwidth demand for applications such as backup and mirroring for sustained periods of time, which can easily disrupt the performance of the LAN traffic when they share common network resources. Lastly, the SAN network is often built on a different network protocol, such as Fibre Channel, that is different from the prevalent LAN protocol of Ethernet. Even when iSCSI SAN runs over Ethernet technology, the SAN may still be separated from the LAN either physically, or logically via VLAN to ensure the security and the QoS on the SAN traffic.

The SAN software architecture required on the computer systems (servers), shown in Figure 6, is essentially the same as the software architecture of a DAS system. The key difference here is that the disk controller driver is replaced by either the Fibre Channel protocol stack, or the iSCSI/TCP/IP stack that provides the transport function for block I/O commands to the remote disk system across the SAN network. Using Fibre Channel as an example, the block I/O SCSI commands are mapped into Fibre Channel frames at the FC-4 layer (FCP). The FC-2 and FC-1 layer provides the signaling and physical transport of the frames via the HBA driver and the HBA hardware. As the abstraction of storage resources is provided at the block level, the applications

that access data at the block level can work in a SAN environment just as they would in a DAS environment. This property is a key benefit of the SAN model over the NAS, as some high-performance applications, such as database management systems, are designed to access data at the block level to improve their performance. Some database management systems even use proprietary file systems that are optimized for database applications. For such environments, it is difficult to use NAS as the storage solution because NAS provides only abstraction of network resources at the file system level for standard file systems that the Database Management System may not be compatible with. However, such applications have no difficulty migrating to a SAN model, where the proprietary file systems can live on top of the block level I/O supported by the SAN network. In the SAN storage model, the operating system views storage resources as SCSI devices. Therefore, the SAN infrastructure can directly replace Direct Attach Storage without significant change to the operating system.

Figure 6 SAN Software Architecture



Fibre Channel is the first network architecture to enable block level storage networking applications. The Fibre Channel standards are developed in the National Committee of Industrial Technology Standards (NCITS) T11 standards organization. The standards define a layered architecture for transport of block level storage data over a network infrastructure. The protocols are numbered from FC-0 to FC-4, corresponding to the first four layers of the OSI layered network model: physical (FC-0), data link (FC-1, FC-2), network (FC-3), and transport (FC-4). The FC-0 layer defines the specification for media types, distance, and signal electrical and optical characteristics. The FC-1 layer defines the mechanism for encoding/decoding data for transmission over the intended media and the command structure for accessing the media. The

FC-2 layer defines how data blocks are segmented into frames, how the frames are handled according to the class of service, and the mechanisms for flow control and ensuring frame data integrity. The FC-3 layer defines facilities for data encryption and compression. The FC-4 layer is responsible for mapping SCSI-3 protocols (FCP) and other higher layer protocols/services into Fibre Channel commands. Together, the FC protocols provide a purpose-built mechanism for transporting block level storage data across the network efficiently at gigabit rates. As the SAN model can easily replace the DAS storage without changes in the operating system, since its emergence Fibre Channel has enabled the rapid deployment of SAN systems. However, as Fibre Channel is a new ground-up networking technology, its deployment faces the challenge of requiring a dedicated and new networking infrastructure to be built for the storage application. As with any new networking technology, Fibre Channel networking products will take significant time and effort before they reach maturity and full interoperability. Prior to that time, early adopters of Fibre Channel will struggle with such interoperability difficulties. Adding to this challenge, the Fibre Channel protocols introduce a new set of concepts, terminology, and management issues that the network administrators (or users) will have to learn. Collectively, these factors have formed barriers to mainstream adoption of the technology. Consequently, Fibre Channel deployment has been limited to mostly large corporations that have a pressing need for the higher performance that Fibre Channel offers and can afford the price of early adoption. As the technology and products gradually reach higher maturity, affordability, and availability, adoption of Fibre Channel will expand towards more mainstream applications.

IP storage technologies such as iSCSI and FCIP have emerged to take advantage of the ubiquity of IP and Ethernet network infrastructures both in the LAN, MAN, and WAN environments. Ethernet dominates the enterprise network as the lowest cost, most deployed, and most understood technology in the world. IP has become the dominant protocol in the wide area data network that provides connectivity from anywhere to anywhere globally, and the TCP/IP protocol stack is the de facto protocol that most application software are built on. It is only natural, then, to combine these technologies to find a solution to the problem of transporting block level storage I/Os over the existing network infrastructure. The benefit of using these common technologies is multi-fold. First, these technologies are very mature. The R&D investment and years of effort that have been put into these networking technologies is unsurpassed. The results are TCP/IP/Ethernet products that are very mature, have good interoperability, and are well supported in any operating system. Second, the scale of deployment has helped to lower the cost of TCP/IP/Ethernet networking devices. Riding the low cost curve of the mass-market products helps to reduce the cost of SAN infrastructure. Third, there is no shortage of skilled people who understand these protocols well. Not only is it easier to put the SAN network together, but it is also lower cost to manage a network infrastructure that is based on mainstream technologies.

FCIP provides a means of encapsulation Fibre Channel frames within TCP/IP for linking Fibre Channel SAN islands over a wide area IP network. Each Fibre Channel SAN island is connected to the IP backbone via a FCIP device that is identified by an IP address. The FCIP devices establish TCP/IP connections among each other. The FCIP tunnels runs over the TCP connections. When a Fibre Channel node in one SAN island needs to communication to another node that belongs to a different SAN island, the FCIP device at the source island encapsulates the Fibre Channel frame in TCP/IP and sends it across the IP network to the destination FCIP device. The destination FCIP device decapsulates the FCIP packets to recover the original Fibre Channel frame. In this way, the different Fibre Channel islands are connected together to form a virtual SAN that encompass all the islands via the FCIP tunnels over the IP infrastructure.

iSCSI uses TCP/IP to provide reliable transport of SCSI commands directly over a IP/Etherent network among the SCSI initiators and the SCSI targets. Because each host and storage device supports the Ethernet interface and the iSCSI stack, these devices can plug directly into an Ethernet or IP network infrastructure. From the network perspective, the iSCSI devices are simply regarded as normal IP or Ethernet nodes. The network infrastructure need not be different than normal enterprise LAN or IP network. Significant cost savings can be achieved by constructing an iSCSI SAN using mass-market enterprise Ethernet switching devices. Additionally the SAN infrastructure can be seamlessly integrated with the Enterprise LAN to further reduce the cost of building and managing the network. In the iSCSI protocol layers, the iSCSI layer maps SCSI commands into TCP packets directly. As in FCIP, the TCP ensures reliable transport of packets from the source to the destination. iSCSI also specifies the IPSec protocol for data security. At the data link and physical layer, Ethernet or any other protocol that can handle IP traffic may be used to carry the traffic on the physical media.

iFCP is a gateway-to-gateway protocol for providing Fibre Channel Fabric services to Fibre Channel end devices over a TCP/IP network. An iFCP network emulates the services provided by a Fibre Channel Switch Fabric over TCP/IP. Fibre Channel end nodes, including hosts and storage devices, can be directly connected to an iFCP gateway via Fibre Channel links, and operate as if they were connected by a virtual Fibre Channel Fabric. Normal Ethernet/IP network is used to connect the iFCP gateways together to provide the abstract view of a virtual Fibre Channel Fabric. These design considerations establish iFCP as a migration path from Fibre Channel SANs to IP SANs.

3 Storage Network Elements

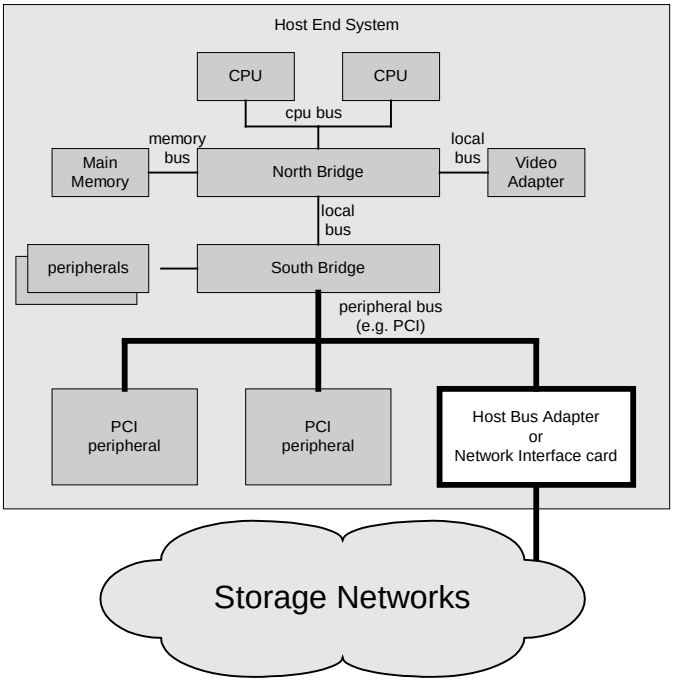
A storage network generally consists of a variety of hardware and software elements. As the scope of this paper is focus on hardware architectures, the software elements that mostly involved with management functions are not discussed in detail. The hardware elements can be divided into three categories: Host systems, storage systems, and switches and bridges that provide the connectivity.

The following sections discuss these major building blocks of storage networks and highlight the hardware architectures of these building blocks.

3.1 Host Systems

The Host Systems of storage networking is any computer device that is attached to the storage network to access the storage resources over the network. Typical host end systems include personal computers, workstations, a variety of servers, or other network appliances that have access to the storage resources.

Figure 7 Host System Architecture



An example of a Host System is shown in Figure 7. In the example, the system is composed of multiple CPUs, a north bridge system controller that deals with high speed local bus resources such as main memory and a video adapter, and a south bridge system controller that deals with lower speed I/O devices and the peripheral bus (PCI). For the purposes of discussing storage networking, the element of interest in the host end system is the interface to the storage network. This interface device is often called a Host Bus Adapter (HBA), or Network Interface Card (NIC).

The HBA or NIC provides the interface from the internal bus (such as PCI) and the external network (Fibre Channel, or Ethernet). Often, the software drivers for the host operating system are supplied for the HBA and NIC. The drivers and the HBA combine together to provide the mapping of storage access commands (e.g. SCSI or NFS commands) to the network protocol packets.

Figure 8 depicts the basic functions of a Fibre Channel Host Bus Adapter. Realizing these functions physically is achieved through a combination of hardware and software, and depends on the choice of available IC devices. Most state-of-the-art implementations partition the hardware function into four entities: the optical transceiver device; the SERDES (serializer/deserializer) device that deals with the conversion between parallel data and the higher speed serial data, and the clock/data recovery function and 8b/10b encoding/decoding; the Fibre Channel protocol logic (FC-1, FC-2) that handles the order sets, signaling protocol and link services; and some datapath function of FC-4 as well as the interface to the host bus. The exception handling and control function of layers FC-2/3/4 are often handled by the device driver software. With the advancement of IC technology, integration of all HBA electronics hardware into a single silicon device has become technically feasible and advantageous.

Figure 8 Fibre Channel HBA Functional Diagram

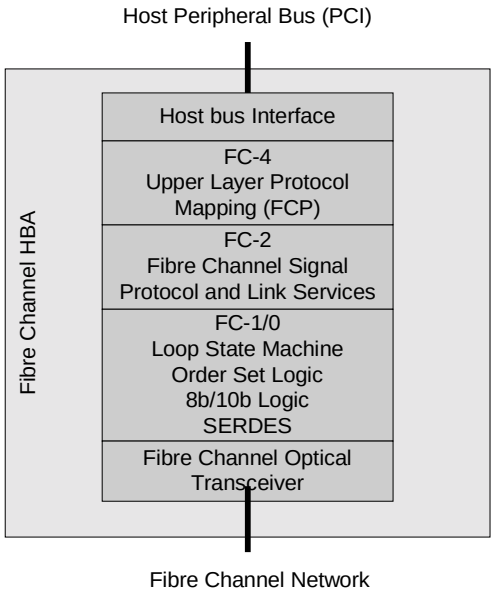


Figure 9 Ethernet NIC Functional Diagrams

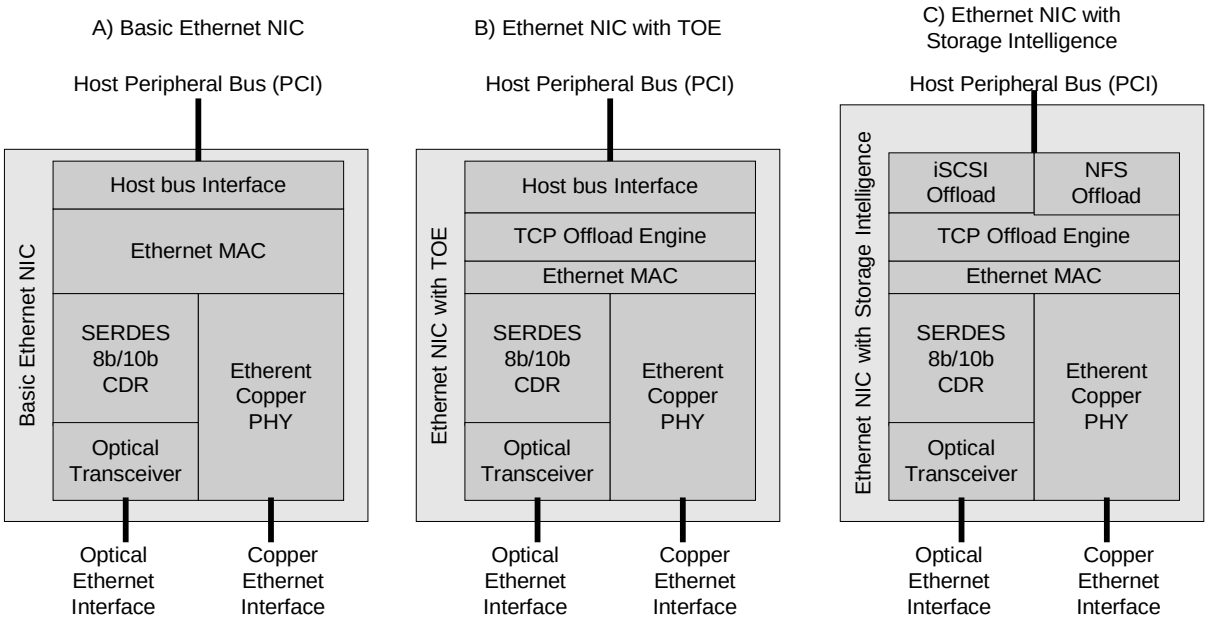


Figure 9 depicts 3 variants of Ethernet Network Interface Card design. Diagram A shows a classic Ethernet NIC that contains the Ethernet physical layer function (copper PHY device, or SERDES and optical transceiver), the Ethernet Media Access Control (MAC) function, and the host bus interface. This type of Ethernet NIC provides the basic function of sending, receiving and checking the integrity of Ethernet frames. The Host system can implement the TCP/IP protocol stack in the operating system software based on the data link layer services and facilities provided by the NIC. Furthermore, IP storage protocols such as NFS, CIFS, iSCSI, FCIP, iFCP can be implemented in the operating system software on top of TCP/IP layer.

Diagram B incorporates the TCP/IP protocol processing into the NIC hardware. The TCP protocol consumes significant processor resources when processed in operating system software. In a host system with the classic NIC Design A, a high percentage of host CPU cycles must be dedicated just for sending and receiving TCP packets. Design B offloads this processing burden onto a TCP offload engine (TOE) on the NIC card. Typically, the TOE is a high performance, general-purpose embedded processor or a special-purpose processor that is customized for the task of handling TCP datapath traffic. As the TCP protocol stack is an integral part of the traditional operating system kernel, moving the TCP function into the NIC has some software impact on the operating system. First of all, most TOE only implement the fast-path function of TCP, all the control function and exception condition handling is still processed by the driver software that is running on the host processor. Second, the data transfer between the NIC, the Operating System kernel, and the user process involves significant data movement overhead. The entire system has to be optimized to fully realize the benefit of the hardware TOE. Third, some TOE are optimized for a small number of long live TCP connections required by storage applications such as iSCSI or NFS. The TCP stack implemented on the NIC may not be suitable for all TCP traffic that the host may initiate. As a result, the host OS may still implement a full TCP /IP stack for the normal TCP traffic of other applications and reserve the TOE for storage traffic only. Such parallel TCP stacks need to be coordinated properly to prevent system anomalies. One popular way of tying the TOE

NIC to the operating system efficiently for specific application is to provide a customized SOCKET layer library in the user mode as the driver. The library can be linked into the application for direct access to the TOE resources without going through time-consuming kernel mode context switching and data copying. The socket library can also be linked to the iSCSI or NFS stack to give high performance TCP services to the selected component of the operating system.

Diagram C goes one step further by incorporating the storage-processing layer (such as the iSCSI and NFS protocol) into the NIC. Again, such function are often implemented in an embedded processor or a special storage network processor that is located on the NIC. Also, as before for Design B, to tie the intelligent NIC functions into the operating system seamlessly requires similar software patches to be added to the operating system.

3.2 Storage Systems

The basic function of storage systems is to provide storage resources to the network for primary data store, mirror data store, or backup data store. A wide variety of storage systems are available in the marketplace and serve different purposes with varied performance and features. The storage devices include RAID disk arrays, Just a Bunch Of Disks (JBODs), tape systems, Network Attached Storage Systems, optical storage systems etc. The type of interfaces provided on these devices include SCSI, Fibre Channel, Ethernet.

In order to understand the storage systems, it is important to understand the disk drives, the most common building block of system systems, and the disk drive interface technologies used. The next few sections will discuss the key characteristics of disk drives and internal architecture of various storage systems.

3.2.1 Disk Drive Interfaces

The disk drive industry is a rapidly changing industry that turns out a large variety of disk drive products each year. These products are differentiated by drive capacity, disk speed (RPM), access time, reliability, and mostly importantly interface type. Looking at the disk drive industry today, the interface type divides the market into 3 broad categories:

- The low-end PC market, with relatively low performance and reliability but at very economical price, dominated by ATA interfaces;
- The mid-range enterprise market, with higher performance and reliability, serviced by SCSI interfaces;
- The high-end enterprise application market, with the highest performance, reliability and scalability, provided by Fibre Channel Drives.

It is important to keep in mind that these disk drive interfaces have gone through several generations of evolution (IDE/ATA/EIDE/UDMA/Ultra-ATA, Narrow/Wide SCSI-1/2/3) and are still evolving rather quickly. Serial ATA (SATA) is emerging to replace ATA for higher performance. Serial Attached SCSI (SAS) was invented to be the next generation SCSI interface. Fibre Channel is evolving from 1Gbit/s to 2Gbit/s to 4Gbit/s and 10Gbit/s to satisfy the ever-increasing demand for higher bandwidth. Table 1 compares the key characteristics of each of these important disk interfaces.

Table 1 Comparison of Disk Interfaces

Features	IDE/ATA/EIDE/UDMA/ Ultra-ATA	SATA	SCSI & SAS	Fibre Channel (Arbitrated Loop)
Application	PC, Macintosh low-end server low-end workstation low-end NAS low-end RAID	PC, Macintosh low-end server low-end workstation low-end NAS low-end RAID	PC, Macintosh Mid/high range server NAS RAID Storage Systems	High-end Server High-end NAS Storage Systems
Device Types	Hard Disk Drive CD-ROM, DVD, Low-end Tape Drive	Hard Disk Drive CD, DVD, Tap devices	Hard Disk Drive CD-ROM, DVD Optical WORM Tape Drive Scanner	High End Hard Disk Drive
Maximum number of devices supported (per Bus/channel)	2	Point to point, support multiple devices via RSM	Narrow SCSI: 7 Wide: 15 SAS: point to point, support up to 128 devices via expander	FC-AL: 126 FC fabric: unlimited
External Device Support	No	No?	Yes	Yes
Maximum Burst Transfer Rate	EIDE (PIO) = 3~16MB/s EIDE (DMA) = 2~ 8MB/s UDMA = up to 33MB/s Ultra-ATA = 100MB/s	1.5G SATA (150MB/s) 3G SATA (300MB/s) 6G SATA (600MB/s)	SCSI-1 = 5MB/s Fast-10 = 10MB/s Fast-20 Wide = 40MB/s Fast-80 Wide = 160MB/s SAS: 150MB/s, 300MB/s, 600MB/s leverage SATA physical layer	1G FC 2G FC 4G FC 10G FC
Multitasking	Only one active device per bus	Tag Command queuing allows parallel tasks within a HDD. Lack the support for multi-initiator in HDD	Multiple active disks per bus Tag queuing Bus master DMA	Supports SCSI multitasking
Error Detection	Data protected by CRC, control unprotected	CRC-32 for data and control	Bus Parity	Frame CRC
Cabling/ Connector	40-pin dual row header 32 signals + 7 grounds Ultra ATA: 80-wire cable	7-pin 4 signals + 3 grounds hot pluggable	SCSI: 50-pin or 68-pin connector SAS: same cabling as SATA over 10m distance	Optical Fibre
Signaling	Ultra ATA: 3.3V DDR signals, 5V tolerant	LVDS 0.25 common mode voltage, 0.125 swing	Single ended or Low voltage differential SAS: LVDS	Optical
Hard Disk Cost	Inexpensive	Similar to ATA	Relative expensive: more sophisticated protocol, significant firmware, higher-end target application	Most Expensive: FC/SCSI protocol processing, higher performance market

ATA

ATA is the primary internal storage interface for the PC, connecting the host system to peripherals such as hard drives, optical drives, CD-ROMs. Ultra ATA is an extension of the original parallel ATA interface introduced in the mid 1980's and maintains backwards compatibility with all previous versions. The latest version of the Ultra ATA specification accepted by the NCITS T13 committee is ATA/ATAPI-6. The Ultra-ATA specification supports up to 100Mbytes/sec data transfer rates using double edge clocking. ATA is a relatively simple protocol that accesses the disk through register maps. This simplicity reduces the cost of disk implementation and simplifies integration and test. As the dominant hard disk technology driven by the PC market, the ATA disk drives have significantly lower cost than higher performance SCSI/FC disk drives. ATA disks also often have the highest volumetric density and lower power consumption. But inexpensive ATA drives often come with lower reliability (shorter MTBF than SCSI drives) and lower performance (lower RPM spin speed, smaller cache, slower access time). The ATA protocol also suffers in performance due to the serial access nature of the protocol for heavy-duty multitask applications.

SATA

Serial ATA is the next generation internal storage interconnect designed to replace Ultra ATA. The SATA interface is an evolution of the ATA interface from parallel bus to serial bus architecture. The serial bus architecture overcomes the difficult electrical constraints hindering continued speed enhancement of the parallel ATA bus. The first generation SATA-I technology is designed to be a direct serial transport mechanism for ATA protocol data at 150Mbyte/sec that is fully compliant with the ATA protocol at the software level. SATA II is defining protocol enhancements to further speed increases to 3Gbit/s and to improve the protocol efficiency in a multitasking environment. A future SATA device, with the appropriate drivers, could set up a host adapter DMA engine to execute DMA transfers, execute commands out of a queue located in the host memory, report command results into a table/list in host memory, execute commands and deliver data out of order to optimize performance, and do data scatter/gather in the host memory. With these enhancements, the protocol efficiency for enterprise and storage systems applications would approach the performance of SCSI/FC-based disk drives. But even in SATA II, the support for multi-initiator in a target device has not been specified.

SCSI

The Small Computer System Interface (SCSI) came to life in 1986 as ANSI X3.131-1986 standard with long history of proprietary hard disk interfaces for high performance computers dated back to the 1970s. SCSI defines a universal, parallel system interface, called the SCSI bus, for connecting up to eight devices along a single cable. SCSI is an independent and intelligent local I/O bus through which a variety of different devices and one or more controllers can communicate and exchange information independent of the rest of the system. SCSI has gone through a long evolution, with SCS-I, SCSI-2, and SCSI-3 over different types of cabling, and including Wide SCSI and Narrow SCSI. Presently, the various forms of SCSI interfaces dominate the heavy-duty enterprise applications that require high performance for multitasking, high reliability, and scalability to grow capacity with multiple disks. The primary benefits of SCSI include: cross platform interoperability; support for a large number of devices; easy expandability (up to seven devices with Narrow SCSI, and 15 devices with Wide SCSI); long cabling distances for external device connectivity; very good support for multitasking disk accesses that allow for interleaving of multiple concurrent transfers; tag queue and out-of-order data delivery. Some of these advanced features are being advocated as enhancements to the SATA-2 standards, as they

have a large benefit on the system performance for enterprise and storage systems applications. SCSI disk drives are optimized for enterprise applications that can tolerate moderately higher cost than the PC market, therefore SCSI drives can adopt more advanced technologies such as higher disk spin speed, better mechanics, and larger caches to achieve higher performance and reliability. As the protocol is significantly more complicated than ATA, SCSI devices require longer design cycles and more rigorous testing for the multitasking operations. The result is that SCSI products tend to cost quite a bit more than the ATA disk drives for the same disk capacity.

SAS

To overcome the bandwidth barrier of parallel bus cabling, the Serial Attached SCSI (SAS) is being defined to replace the physical layer of SCSI with serial bus technology. The goal is to achieve higher data transfer rates and yet maintain the protocol compatibility at the command set level. SAS is a new near-cabinet and disk interface technology that leverages the best of the SCSI and serial ATA interfaces to bring new capabilities, higher levels of availability, and more scalable performance to future generations of servers and storage systems. SAS uses a serial, point-to-point topology to overcome the performance barriers associated with storage systems based on parallel bus or arbitrated loop architectures. With a SAS port expander device, SAS can support large fan-outs in storage systems with a large number of devices. The expander also allows SAS to connect with lower-cost, high-capacity serial ATA disks as well as high-availability, high-performance Serial Attached SCSI drives. This ability to support two classes of disk drives offers system managers new levels of flexibility in managing and scaling storage resources. The SAS technology is positioned as an enhancement to the widely deployed SCSI technology for mid-range storage solutions that demand high performance and higher reliability.

Fibre Channel

Fibre Channel Disk Drives typically come with single or dual Fibre Channel Arbitrated Loop (FC-AL) interfaces. The dual FC-AL interface is useful in storage systems for providing redundant cabling. Some drives even allow concurrent access from the two interfaces to increase the bandwidth to the drive.

The FC-AL is a loop interconnection topology that allows up to 126 participating node ports to communicate with one another without the need for a separate switch fabric. Using the FC-AL protocol, a large number of disk drives can be connected together for large capacity storage systems. All the devices on the loop share the bandwidth of the loop. To prevent a single point of failure in the physical loop topology, the disk drives in a storage system are typically connected to a central Port Bypass Controller (PBC) in a star topology. The PBC logically implements the FC-AL topology by chaining the ports together. In the case a drive failure, the PBC can be configured to bypass the failed drive and allow the remaining drives to maintain loop connectivity. The Fibre Channel protocol provides a transport mechanism for SCSI commands, but also provides enhancements in several areas. First, the FC disk drives can take advantage of the higher data rates offered by the Fibre Channel physical links. Second, the FC-AL allows for much longer cabling distance than the parallel SCSI interface. Third, the FC-AL can accommodate up to 126 devices for each loop, a big improvement over the 15 maximum devices supported by each SCSI interface. Combined with the redundancy offered by the dual-loop configuration, the FC-AL provides a perfect solution for higher-end/large storage systems.

FC-AL provides simple, low-cost connectivity without requiring a separate fabric switch; it provides performance and connectivity that is 5x-10x the capabilities of fast-wide SCSI at

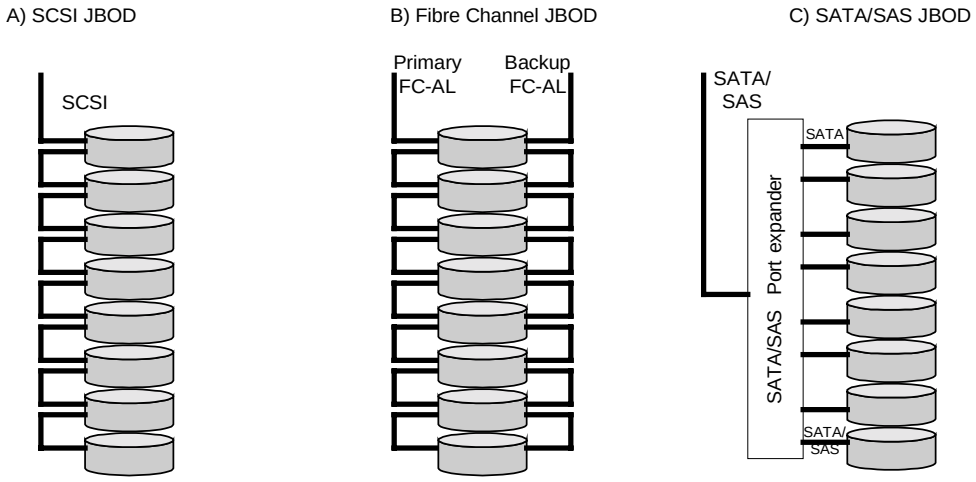
comparable system level costs; and it provides a good path for performance improvement as Fibre Channel Protocol grows towards higher speeds. Typically, FC disk drives are manufactured with the same hard disk technology platform as the SCSI disk drives but with the addition of Fibre Channel interface. Hence the mechanical performance and reliability are similar to SCSI disk drives of the same make. The cost is often slightly higher than SCSI disk drives. However, due to the many benefits of the FC-AL interfaces, it is currently the dominant disk drive technology in large storage systems.

3.2.2 JBOD

A JBOD is an enclosure with multiple disk drives installed in a common backplane. Since JBOD has no front-end logic that manages the distribution of data over the disks. The disks are addressed individually as separate resources. The JBOD can be used as direct attached storage that is connected to the host server directly, as the storage array that is attached to a NAS filer, or it can be used on a storage network with Fibre Channel Interface.

There are a number of ways of connecting the disk drives together in JBOD systems, depending on the type of disk drive the JBOD systems are based on. Current JBOD products are usually marketed as 19 inch enclosures with SCSI or Fibre Channel Interfaces, since both interfaces have the capability of supporting a relatively large number of disk drives on each interface. Using wide SCSI cabling, each SCSI bus can support a daisy-chain of up to 15 disks, sufficient for chaining together the number of standard 3.5 inch disk drives that a typical 19 inch enclosure can house (Figure 10 A). The Fibre Channel interface can support up to 126 disks on each arbitrated loop, not only sufficient for connecting disks within an enclosure, but also allowing multiple JBOD enclosures to be connected together in a single loop (Figure 10 B). The SATA or Serial Attached SCSI (SAS) technologies are design to support point to point connections. The SATA/SAS port expander device helps to solve the fan-out problem of a disk array by allowing one host controller to communicate with a large number of disk drives (Figure 10 C). The SAS expander not only supports a 1:N fan-out for SAS, but also supports communication between a SAS host and a SATA disk drive via the SATA Tunneling Protocol.

Figure 10 JBOD Disk Configurations



It is also possible to build a JBOD with a Fibre Channel fabric or with a hierarchy of FC switches and PBCs as the interconnect between the host and the disk drive disks. However, as the number of PBCs and disks grows, a problem arises with the length of the resultant loop. Normal FC_AL requires that all Fibre Channel frames propagate through all devices on the loop in a hop by hop fashion. As the loop length grows, so does the latency incurred by a frame from the HBA going through all the disk drives on the path to the destination device. An enhancement is required in the PBC architecture to essentially cut out part of the loop length based on the knowledge of the intended frame's destination. In addition to being able to take the shortest path from the source to the destination for latency reduction, the FC switches can also provide the additional benefit of supporting multiple concurrent data transfer to several disk drives. The parallelism can significantly improve the overall system I/O throughput, especially when coupled with an intelligent cache system design.

3.2.3 RAID

RAID (Redundant Array of Independent Disks) is a technology to combine multiple small, independent disk drivers into an array that looks like a single, big disk drive to the system. Simply putting *n* disk drives together (as in JBOD) results in a system with a failure rate that is *n* times the failure rate of a single disk. The high failure rate makes the disk array concept impractical for addressing the high reliability and large capacity needs of enterprise storage.

The system reliability is measured in MTTF (Mean time to Failure) or FITS (Failures in Time). The MTTF is the mean value of life distribution for the population of devices under operation or the expected lifetime of an individual device. Usually MTTF is measured in hours. The FITS is the measure of failure rates in 10^9 device hours.

Assuming $MTTF_{Disk}$ and $FITS_{Disk}$ to be the MTTF and FITS value of a single disk device, and $MTTF_{JBOD}$ and $FITS_{JBOD}$ to be the measurement of a JBOD system with *n* disks, the resulting failure rate parameters follow the equations below:

$$FITS_{Disk} = \frac{10^9}{MTTF_{Disk}}$$

$$FITS_{JBOD} = n \times FITS_{Disk} = \frac{n \times 10^9}{MTTF_{Disk}}$$

$$MTTF_{JBOD} = \frac{10^9}{FITS_{JBOD}} = \frac{MMTF_{Disk}}{n}$$

With the current typical Fibre Channel disk drive MTTF parameter of 1.5M hours, a JBOD system with 100 disk drives has a MTTF of 15000 hours, or 1.7 years. Without the use of RAID technology, the raw MTTF number is clearly too low for applications that demands high reliability. This is dismal when considering the scalability requirements of enterprise storage is multi-shelf. Later analysis will show that RAID technologies can dramatically increase the MTTF of disk array systems.

The basic concepts of RAID were described in [9] as five types of array architectures, called RAID levels – each providing disk fault tolerance and each offering different feature sets and performance trade-offs. Later, industry introduced proprietary RAID implementations that included various combinations and variations of the basic RAID levels. Some of the most popular extensions will be discussed in the following paragraphs in addition to the basic RAID levels.

RAID 0 – Data Striping

After the initial concept of RAID was introduced, RAID 0 was adopted to describe non-redundant disk arrays, wherein the data striping was used only to increase capacity and data throughput of the disk array.

RAID 1 - Mirroring

RAID level 1 describes data mirroring system. Mirroring provides data redundancy by writing the same data to both sides of the mirror. Level 1 is simple to implement, provides good data reliability, and doubles read performance of the array, all at the cost of doubling the storage capacity required.

RAID 2 – Striping with ECC

RAID Level 2 uses Hamming codes to generate ECC (Error Correction Code) checksums. The data and the ECC checksums are striped across multiple disk drives. The basic idea is for the ECC to correct single or multiple bit failure across the disk drives.

In practice, although ECC codes are often used inside the disk drive for correcting bit errors from the physical storage media, it is unsuitable for protecting data across multiple disks. Typical disk drive failures are often caused by catastrophic mechanical failures – the drive either works properly, or it does not work at all. To protect the disk array against single disk drive failure, a simple XOR (Exclusive OR) parity is as good as the more complicated ECC code. For this reason, XOR parity is the primary checksum mechanism for RAID architectures including RAID 3, 4 and 5. RAID 2 is conceptually intuitive, but has rarely or never been implemented in any real storage systems.

RAID 3 – Byte Striping with Parity

RAID Level 3 stripes data bytes across a group of disks in a similar way as RAID 2, and generates parity information over the data on a dedicated parity disk. If a disk drive fails, the data can be restored on the fly by calculating the exclusive OR (XOR) of the data from the remaining drives. RAID provides high reliability at the cost of one additional parity drive per RAID group. The storage capacity requirement of RAID 3 is much lower than mirroring (RAID 1).

The major drawback to level 3 is that every read or write operation needs to access all drives in a group, so only one request can be pending at a time (i.e., sequential access). As the disk access (seek) time dominates the overhead for random disk accesses, the sequential access pattern of RAID 3 makes it impossible to hide the access latency by overlapping multiple I/O transactions over time. The byte striping methods of RAID3 also imply certain restrictions on the number of disks in a RAID group and the size of the logical block. The most efficient block size is now dependent on the number of disks (Group_size x Sector_size). Some RAID configurations can result in unusual block sizes that are difficult for the operating systems to deal with.

RAID 4 – Block Striping with Parity

RAID 4 uses the same XOR parity checksum technique to provide data redundancy as RAID 3, except the checksum is calculated over disk blocks. A dedicated drive is assigned to store the checksum blocks. As the data blocks are stripped across the data drives in the RAID group, RAID 4 can simultaneously perform multiple asynchronous read transactions on the different disk drives, giving a very good read transaction rate.

However, the write performance is still limited to the transaction rate of a single disk, as any block write requires the corresponding block in the parity disk to be updated. The parity disk becomes the system performance bottleneck for write accesses.

RAID 5 – Block Striping with Distributed Parity

RAID 5 addresses the write bottleneck issue of RAID 4 by distributing the parity data across all member drives of the group in a round robin fashion. RAID 5 still requires any write transaction to update the parity block and the data block. To calculate the new checksum, RAID 5 requires the XOR operation to be applied to the old data block (1st block read), the old checksum (2nd block read) and the new data block to generate the new checksum block. Then the new data block and the new checksum blocks are written back (2 block writes) to their respective drives. Each write transaction requires 2 block reads and 2 block writes to be executed. Since the parity blocks are even distributed across all the member drives, the probability of access conflict for the parity drive from simultaneous write transactions is much reduced. As the result, RAID 5 has the full benefit of RAID 4 for data redundancy and high read transaction rate, the write performance is also very high. RAID 5 is commonly regarded as the best compromise of all RAID levels among read performance, write performance, data availability, cost of capacity. It is the most commonly used RAID level today.

Other RAID Architectures

Over the years, the storage industry has implemented various proprietary storage array architectures, often by combining multiple techniques outlined in the basic RAID levels. RAID 0+1 is a combination of level 0 (striping) and 1 (mirroring). RAID 0+1 benefits from the performance gain of striping and the data redundancy provided by mirroring. Because no parity needs to be calculated, write operations are very fast. In some proprietary implementations, extensions are made to RAID 5 and the result called RAID 6. In addition to the parity, RAID 6 includes the additional checksum generated over all the data and parity using Reed-Solomon coding on another drive. The RAID 6 design can allow the failure of any two drives at the same time, thereby dramatically increasing the survival capability beyond what can be offered by RAID 5. RAID 6 requires additional storage space for the additional checksum, and the write performance is slower from having to generate and write to the two checksums. But clearly it may be applied to protect mission critical data that requires very high fault tolerance.

RAID Reliability Calculation

To illustrate the reliability enhancement provided by RAID techniques, it is necessary to revisit the reliability calculation of a RAID system. First, the following terms must be defined:

n = total number of disks with data

g = number of data blocks in a group

c = number of checksum blocks in a group

$m = n/g$ = number of groups

$MTTR_{Disk}$ = mean time to repair a failed disk

$MTTF_{Disk}$ = mean time to failure

Next, assume the disk failures are independent and occur at a uniform rate, then the mean time to failure is given by:

$$MTTF_{Group} = \frac{MTTF_{Disk}}{(g+c)} \times \frac{MTTF_{Disk}}{((g+c-1)\times MTTR_{Disk})} = \frac{MTTF_{Disk}^2}{(MTTR_{Disk}\times(g+c-1)\times(g+c))}$$

$$MTTF_{RAID} = \frac{MTTF_{Group}}{m} = \frac{MTTF_{Disk}^2 \times g}{(MTTR_{Disk}\times(g+c-1)\times(g+c)\times n)}$$

Now, calculate the improvement in reliability provided by using the example of a RAID 5 system with 100 data disks divided into groups of 10 disks. 10 additional disks are required for the checksum (total number of disks = 110; n =100; g = 10; c =1). Assuming it takes 10 hours to repair a failed disk (replace the faulty disk and repopulate the data based on the checksum (MTTR=10 hr) and a normal disk mean time to failure of 1.5M hours ($MTTF_{Disk}$ = 1.5Mhr) like before. Plugging these numbers into the equation yields:

$$MTTF_{RAID} = 2.045\times10^8 \text{ hours} = 23349.9 \text{years}$$

Contrasting this result to the MTTF for a 100-disk JBOD system (only about 1.7 years), it is clear that the RAID 5 technology has dramatically increased the reliability of the storage array to over 23 thousand years. Because of this, RAID technology has established itself to be corner stone of highly reliable storage array systems.

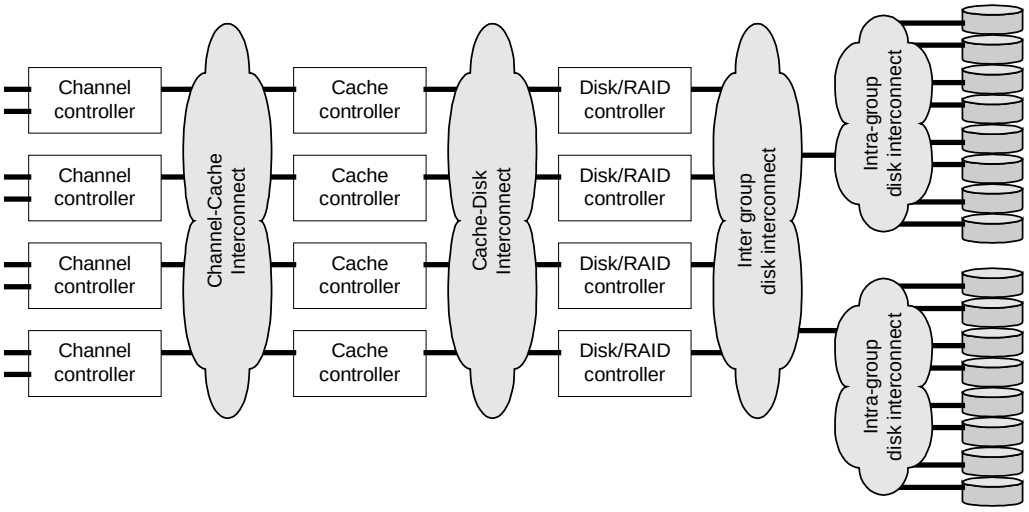
3.2.4 Storage Systems – Various Front-end Architectures

The previous discussion focussed on the various disk drive technologies, JBOD technologies, and RAID technologies. These technologies are the common building block of the back-end storage arrays that provides the storage resources to various storage systems. The back-end storage resources are presented to the outside world via the front-end interface of the storage systems. To the user, what differentiate the various storage systems are the front-end interfaces and the protocol processing associated with the front-end network interfaces.

The storage systems started out as dedicated devices for the different storage models of DAS, NAS, and SAN. These different devices have a lot in common: they often employ the same JBOD or RAID technologies to manage the disk array; the disk/RAID controllers are common across the platforms; and caches are employed to enhance the performance. The key difference is in the network interface at the front end and the protocol processing associated with the front-end interfaces. For NAS devices, the front-end interface needs to support Ethernet ports and the protocol function of NFS and CIFS. For SAN devices, the front-end interface must support Fibre Channel and FCP and SCSI protocols. Most of the storage arrays (RAID or JBOD) on the market today are designed to support block level I/O over SCSI or Fibre Channel interfaces. These arrays can be used as DAS or SAN storage systems. A NAS server can be used at the front end of the disk array to provide NAS protocol processing required to translate between the front-end file level I/O transactions and the block level I/O transactions at the back-end. Some storage vendors market the NAS server and the storage array as separate products; others attempt to package them together as a NAS storage system.

The storage system market can be roughly categorized into monolithic and modular storage systems. The modular systems are built from block modules that are often housed in standard size chassis that can be mounted on a rack to form a complete system. The typical modules include disk enclosures, storage processor units, and switching units. The system is form by connecting a collection of modules together via inter-enclosure cabling. Adding or removing the number of modules, such as disk enclosures, can flexibly scale the system. By partitioning the system into a number of modules with relative simple interfaces among them as the interconnect mechanism, the modular systems can offer economical and flexible solutions to mid-range and low-end applications. But the modular design also limits the choice of architecture and the performance of the system. Monolithic storage systems are typically higher performance storage systems that are housed in a single large shelf unit with the building blocks flexibly arranged inside as needed. Without the limitation of modular systems, the monolithic systems often can employ higher performance interconnect among the building blocks, implement more parallelism for performance improvement, and offer higher redundancy. The end result is a better performing system at a higher cost point.

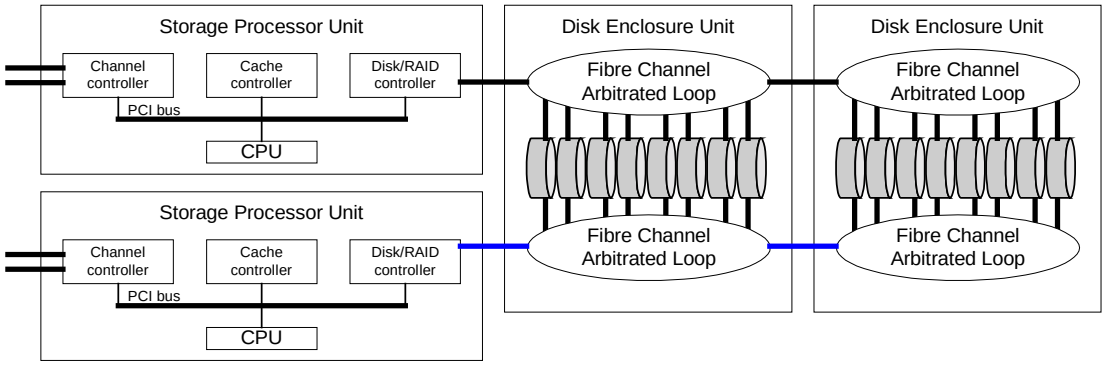
Figure 11 Storage System Reference Architecture



Despite the difference in functional partitioning and physical realization among the various categories of storage systems, the general concept remains the same. Figure 11 provides a reference architecture framework of the datapath of storage systems. The processing units in the storage systems can be 3 types, corresponding to the 4 major stages of processing: channel controller, cache controller, disk controller, and the actual disk drives. Connecting these 4 stages are interconnection networks: channel-cache interconnect, cache-disk interconnect, and the disk interconnect (between the disk controller, and the actual disks). Often, the disk interconnect can be further partitioning into two hierarchies: the intra-group disk interconnect (often corresponding to the interconnect within a disk enclosure in a modular system or a group of adjacent disks in a monolithic design), and the inter-group disk interconnect (for connecting the enclosures/groups of disk drivers to the disk controllers).

The architecture framework is useful for studying and comparing the various storage system implementations. Figure 12 shows a typical implementation of modular storage system. In this implementation, the channel controller, cache controller and the disk controller function are all combined into a single storage processor unit. The channel-cache interconnect and cache-disk interconnect are all provided by a common PCI bus. The channel/cache/disk controller functions in the storage processor unit are implemented with a combination of hardware logic and software functions running on a higher performance CPU. The simplified implementation of the storage processor unit helps to lower the system cost. However, the performance of the modular system is also limited by the performance of the CPU and the throughput of the bus interconnect. At the back-end, the disk drives are organized as a number of cascaded disk enclosures. Within each disk enclosures, the disk drives are connected together with dual fibre channel arbitrated loops (implemented as a star topology using port bypass controller devices) for redundancy.

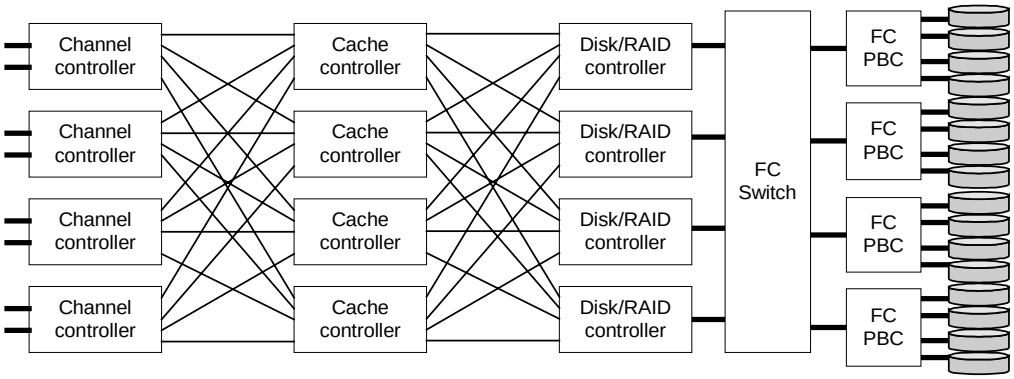
Figure 12 Typical Modular Storage System



Different performance requirements at various price points drive the choice of implementation method for the internal interconnection networks. Figure 13 provides an example of a higher performance monolithic storage system architecture. In this example, full mesh connections are provided both at the stage between the channel controllers and the cache controllers, and at the stage between cache controller and the disk controllers. Using multiple concurrent cache controllers increases the size and the throughput of the cache subsystem, thereby significantly reducing the percentage of read/write accesses that have to go through to the actual hard disk subsystem. Cache memory not only provides higher access bandwidth, but also provides shorter access latency than the hard disk subsystem. It is essential to improve the hit ratio of the cache. Modern high-performance storage systems incorporate gigabytes of DRAM in the cache

controllers and advanced page management algorithms. Combined with the high-speed interconnection network, the majority of the data accesses happen concurrently between the channel controllers and the cache controllers. At the back-end, the parallel disk controllers coupled with the high speed full-mesh interconnection network provide efficient paths for data fetching between the cache controller and the disk subsystem. This example uses a combination of a FC switch (between the enclosures) and Port Bypass controllers (FC-AL within the disk enclosures) between the disk controllers and the actual hard disk drives. This allows for an optimal cost and performance trade-off. The FC-switch allows for concurrent accesses between the multiple disk controllers, while the Port Bypass controller provide inexpensive ways to connect a large number of hard drive devices in a system. Overall, such an example shows how a higher performance system can be constructed with reasonable cost based on matured technologies that are available today.

Figure 13 High Performance Monolithic Storage System



Storage systems are constantly evolving. It is essential for system architects to take advantage of the latest technology advancements to achieve increasing performance, higher capacity and expanding feature sets while lowering system costs.

On the network side, new protocols are emerging that can cause a paradigm shift. Different types of storage systems can converge to a common architecture. For example, the emergence of iSCSI over Ethernet requires new type of channel controller to handle the iSCSI protocol in addition to traditional Fibre Channel Interfaces. NAS and SAN storage systems can be converged to a common hardware platform by providing a NFS/CIFS protocol capable channel controller.

The interconnection network at various stages could take advantage of the new high speed serial processor buses such RAPID IO, HyperTransport, or PCI-Express, which are all progressing towards standards. Some of these buses provide coherent way of sharing distributed cache resources over a high-bandwidth switched network among multiple processors. It is conceivable that the bus evolution will impact the design of channel-cache interconnection and the cache-disk interconnection networks.

The emergence of new hard disk interface standard such as SAS and SATA will have the largest impact on the back-end design of the storage systems. In an evolutionary approach, it may be possible to maintain Fibre channel as the protocol choice between the disk controller and the disk enclosures and use FC switch among the disk enclosures. At the same time, a Fibre channel to SATA bridge device can be used within the disk enclosure to do the translation between FC and

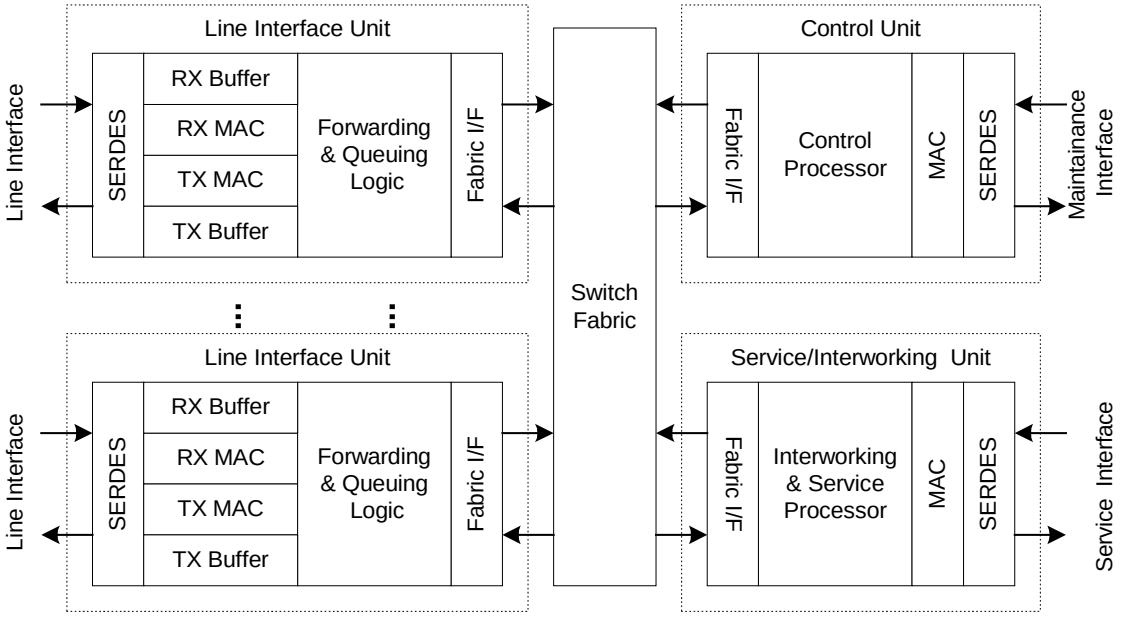
SATA protocols. This approach requires little change to the disk controller and system software design, yet it enables the cost reduction of the disk enclosures by following the low cost curve of SATA hard disk drives. In a more revolutionary approach, the disk controller can speak SATA natively and use a SATA RSM (Routing/Switching/Multiplexing) network for inter-enclosure and intra-enclosure connectivity.

3.3 Storage Switches

Storage switches are basic building blocks to storage network topology. Each switch consists of several ports interconnected by a switch construct. The basic function of a storage switch is to forward frames among the ports according to the protocol supported by the storage network. In a Fibre Channel Network, the switch obeys the rules specified by the FC-SW standard to handle the frames. In an iSCSI storage network, the switching is typically done at the Ethernet layer according to the Ethernet bridging standards, such as 802.1D or the IP layer. Although the specific protocols are different, the various types of storage switches share the common basic structure. However, because of the difference in network protocols, different types of switches are still required: Fibre Channel switches for FC SANs, the Ethernet switches or IP routers for iSCSI network. New multiservice switches are being developed to help the convergence of heterogeneous storage networks running multiple network protocols.

The storage switch market is segmented into two main product classes to address the needs of different types of customers. The small storage switches typically provide between eight and 64 ports while larger director-class switches provide 64 to 256 ports. The director switches not only excel in large port count, but also often provide higher reliability by use of redundant fabrics, fan units and power supplies.

Figure 14 Storage Switch Reference Architecture



The functional blocks of a typical storage switch are shown in Figure 14. The key functional blocks are:

- A number of line interface units that provide the external switch ports.
- The control unit, which is either attached directly to the switch fabric or attached via a dedicated control interconnection network, and provides control plane and management plane functions to the rest of the system.
- The service or interworking units that provide the processing power for complicated service functions or interworking functions. The user traffic from the line interface units that requires special processing is often groomed into the service/interworking units for further processing. The service unit can operate in a one-arm fashion, or have its own external line interfaces. One example is an iSCSI Ethernet service card on a Fibre Channel director switch. The iSCSI service card provides interworking between Fibre Channel and iSCSI as well as the Ethernet interface to the iSCSI network.
- Lastly, the switch fabric is the central unit that ties all the units together and provides the means for passing traffic among the units. Most popular switching fabric technologies employed by recent storage switches include shared memory switches, digital crossbar switches and analog crosspoint switches. There are examples of large storage switches that are put together using smaller switch elements in a multistage fashion.

Although the logical concepts of the various types of storage switches are common, their physical implementations are mainly determined by the target capacity (port count), feature set, and reliability.

Typical small enterprise storage switches are engineered to implement one prevalent switching protocol (Fibre Channel, or Ethernet/IP/iSCSI) efficiently to achieve the low system cost target. The interface type is often limited to the same protocol with different media types at various rates. The additional services or interworking features are often limited or non-existent. Small enterprise storage switches have also taken advantage of System-On-a-Chip (SOC) technologies to achieve high-density designs to lower the cost per port. Some switch SOC integrates the logic of a number of line interface units, the frame buffers, the crossbar or shared memory switch cores and/or additional logic for expansion interface that allows multiple switch devices to be connected together to form larger systems. Many switch devices require the use of multi-channel external SERDES to implement the transceiver function externally, while some switch devices have gone one step further by integrating the SERDES function into the SOC as well. The integration of SERDES helps to reduce the number of I/O on the switch device, which is a limiting factor in high density SOC implementations. Some switch devices provide an interface to an external control processor. This allows the control path function to be implemented externally in software running on the external processor.

The director class switches are often designed to handle high port count, complex services and interworking functions with high reliability. The director switch is often implemented in a variety of distributed modules that can be plugged into a backplane or midplane on a chassis to form a working system. Different modules implement subsystems that serve specific purposes in the system. For example, redundant switch fabric modules provide the switching capability and fail over capability – when the working fabric modules experiences a failure, the system will

automatically switch over to the protection module to allow service to continue without disruption while the failed module is being replaced. The redundant control modules are typically cards where the main control processors reside. The control processor handles the control protocols such as Fibre Channel services, the Ethernet bridging protocols and IP routing protocols as well as the coordination among the distributed modules of the system. Again the redundancy in control modules allows the system to withstand a failure of one of the control units.

As the linecards are now separated from the switch fabric, high-speed serial links are often employed to pass data and control messages between the switch cards and the line cards. The queue status on the linecards is communicated to the switch fabric and the switch fabric makes arbitration decisions to schedule traffic across the system. Such information is communicated between the line cards and the fabric cards via an internal protocol. The line cards typically implement the line interface function for a number of ports. Some design uses separate components for each stage of the line card processing such as line SERDES, MAC function, forwarding engine, fabric interface, and switch fabric SERDES. Many recent designs use advanced IC technologies to integrate the line interface function for many ports into a SOC device. This allows the linecards to achieve higher density, lower cost and lower power consumption. Due to the challenge in achieving good signal integrity over the backplane at very high speed, many designs uses high speed retiming devices on the line card fabric interface to clean up the signal on the backplane.

There are a number of ways to implement the switch fabric for a director switch. Some designs favor very high-density crosspoint switches that have no protocol awareness. Such designs rely on a scheduler device that is designed to gather queue status information from the distributed linecards and make arbitration decisions to control the traffic directions on the crosspoint switch. Other designs employ switch fabric chipsets (mostly based on a digital crossbar) that have more protocol intelligence. The scheduler function is often part of the fabric chipset. In band protocols are used to communicate the request grant information or flow control/scheduling information between the linecard and the fabric card. As the chipset gets more intelligent, it is usually not possible to achieve the same level of port density as the dumb crosspoint switches in a single device using the same IC technology. To support large switch systems, the intelligent switches often have to support multi-slice or multi-stage networks to allow expanding the switch fabric beyond the capacity of a single device. In comparison, the dumb crosspoint switch offers good performance at low cost but the switch capacity is limited by the capacity of the crosspoint switch device; the intelligent switches offers good scalability, but the cost, system complexity and power consumption is often higher.

4 Conclusions

This white paper provides an introduction to the technologies in storage area networking. It touches on the fundamental ideas underlying each storage models, the prevailing protocols and the trends of technology evolution. It also describes how networks are put together with various network elements. Reference architectures are also provided to help the understanding of the key system considerations when designing these network elements.

5 References

[1] M. Krueger, R. Haagens, et al, IETF RFC 3347, Small Computer Systems Interface protocol over the Internet (iSCSI) Requirements and Design Considerations

[2] Julian Satran, Kalman Meth, et al, iSCSI IETF Internet Draft: iSCSI, draft-ietf-ips-iscsi-20.txt

[3] Charles Monia, Rod Mullendore et al, IETF draft: iFCP - A Protocol for Internet Fibre Channel Networkin, draft-ietf-ips-ifcp-14.txt

[4] M. Rajagopal, et al, IETF draft: Fibre Channel Over TCP/IP (FCIP), draft-ietf-ips-fcovertcpip-12.txt

[5] ANSI X3.297-1997, Fibre Channel Physical and Signalling Interface-2

[6] NCITS T11/Project 1508-D/Rev 6.01, Fiber Channel Switch Fabric –3 (FC-SW-3)

[7] Tom Clark, “IP SANs – A guide to iSCSI, iFCP, and FCIP Protocols for Storage Area Networks”, 2002, Pearson Education, Inc.

[8] Gary Field, Peter Ridge, et al. “The book of SCSI: I/O for the new millennium”, 2nd edition, 1999, No Starch Press

[9] David A Patterson, Garth Gibson, and Randy H Katz, “A Case for Redundant Arrays of Inexpensive Disks (RAID)”, Proceedings of the 1988 ACM SIGMOD international conference on Management of data. 1988 , Chicago, Illinois, United States

Head Office:
PMC-Sierra, Inc.
#105 - 8555 Baxter Place
Burnaby, B.C. V5A 4V7
Canada
Tel: 604.415.6000
Fax: 604.415.6200

To order documentation,
send email to:
document@pmc-sierra.com
or contact the head office,
Attn: Document Coordinator

All product documentation is
available on our web site at:
http://www.pmc-sierra.com/
http://www.pmc-sierra.com/processors/
http://www.pmc-sierra.com/serdes/

For corporate information,
send email to:
mailto:info@pmc-sierra.com

PMC-2022178 (1)