BMS
Institute of Technology and Management

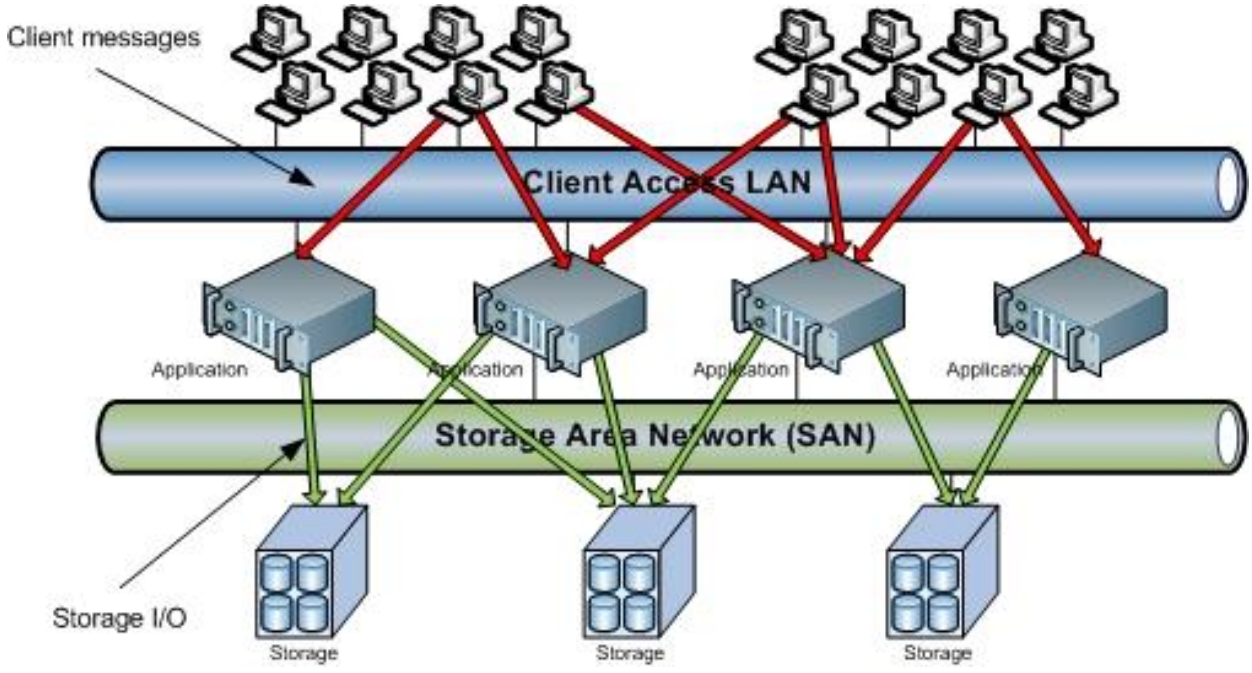
Avalahalli, Doddaballapur Main Road, Bengaluru – 560064

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Storage Area Networks

(17CS754)

SANs are primarily used to access storage devices, such as disk arrays and tape libraries from servers so that the devices appear to the operating system as direct-attached storage.



STORAGE AREA NETWORKS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018) SEMESTER – VII

Subject Code	17CS754	IA Marks	40
Number of Lecture Hours/Week	3	Exam Marks	60
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Module – 1	**Teaching Hours**
Storage System Introduction to evolution of storage architecture, key data centre Elements, virtualization, and cloud computing. Key data centre elements – Host (or compute), connectivity, storage, and application in both classic and virtual Environments. RAID implementations, techniques, and levels along with the Impact of RAID on application performance. Components of intelligent storage systems and virtual storage provisioning and intelligent storage system Implementations.	**8 Hours**
Module – 2	
Storage Networking Technologies and Virtualization Fibre Channel SAN components, connectivity options, and topologies including access protection mechanism 'zoning", FC protocol stack, addressing and operations, SAN-based virtualization and VSAN technology, iSCSI and FCIP(Fibre Channel over IP) protocols for storage access over IP network, Converged protocol FCoE and its components, Network Attached Storage (NAS) - components, protocol and operations, File level storage virtualization, Object based storage and unified storage platform.	**8 Hours**
Module – 3	
Backup, Archive, and Replication This unit focuses on information availability and business continuity solutions in both virtualized and non-virtualized environments. Business continuity terminologies, planning and solutions, Clustering and multipathing architecture to avoid single points of failure, Backup and recovery - methods, targets and topologies, Data deduplication and backup in virtualized environment, Fixed content and data archive, Local replication in classic and virtual environments, Remote replication in classic and virtual environments, Three-site remote replication and continuous data protection	**8 Hours**
Module – 4	
Cloud Computing Characteristics and benefits This unit focuses on the business drivers, definition, essential characteristics, and phases of journey to the Cloud. ,Business drivers for Cloud computing, Definition of Cloud computing, Characteristics of Cloud computing, Steps involved in transitioning from Classic data center to Cloud computing environment Services and deployment models, Cloud infrastructure components, Cloud migration considerations	**8 Hours**
Module – 5	
Securing and Managing Storage Infrastructure This chapter focuses on framework and domains of storage security along with covering security. implementation at storage networking. Security threats, and countermeasures in various domains Security solutions for (Fiber channel)FC-SAN, IP-SAN and NAS environments, Security in virtualized and cloud environments, Monitoring and	**8 Hours**

managing various information infrastructure components in classic and virtual environments, Information lifecycle management (ILM) and storage tiering, Cloud service management activities	

Course outcomes: The students should be able to:

- Identify key challenges in managing information and analyze different storage networking technologies and virtualization
- Explain components and the implementation of NAS
- Describe CAS architecture and types of archives and forms of virtualization
- Illustrate the storage infrastructure and management activities

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Information Storage and Management, Author :EMC Education Services, Publisher: Wiley ISBN: 9781118094839
2. Storage Virtualization, Author: Clark Tom, Publisher: Addison Wesley Publishing Company ISBN: 9780321262516

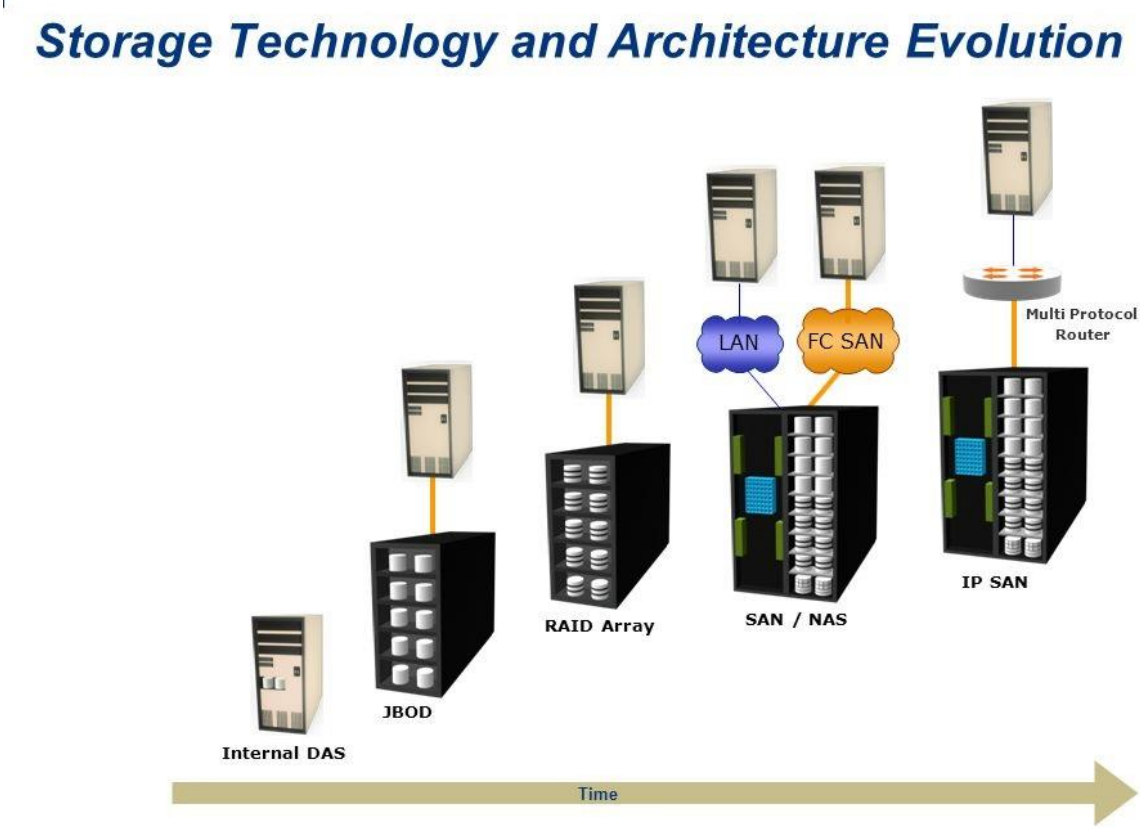
Table of Content

Module-1

Introduction to evolution of storage architecture:

Introduction

Information is increasingly important in our daily lives. We have become information dependents of the twenty-first century, living in an on-command, on-demand world that means we need information when and where it is required. We access the Internet every day to perform searches, participate in social networking, send and receive e-mails, share pictures and videos, and scores of other applications. Equipped with a growing number of content-generating devices, more information is being created by individuals than by businesses.



Key data centre Elements:

Uninterrupted operation of data centers is critical to the survival and success of a business. It is necessary to have a reliable infrastructure that ensures data is accessible at all times. While the requirements, , are applicable to all elements of the data centre infrastructure, our focus here is on storage systems.

1 Availability: All data center elements should be designed to ensure accessibility. The inability of users to access data can have a significant negative impact on a business.

2 Security: Polices, procedures, and proper integration of the data center core elements that will prevent unauthorized access to information must be established. In addition to the security measures for client access, specific mechanisms must enable servers to access only their allocated resources on storage arrays.

3 Scalability: Data center operations should be able to allocate additional processing capabilities or storage on demand, without interrupting business operations. Business growth often requires deploying more servers, new applications, and additional databases. The storage solution should be able to grow with the business.

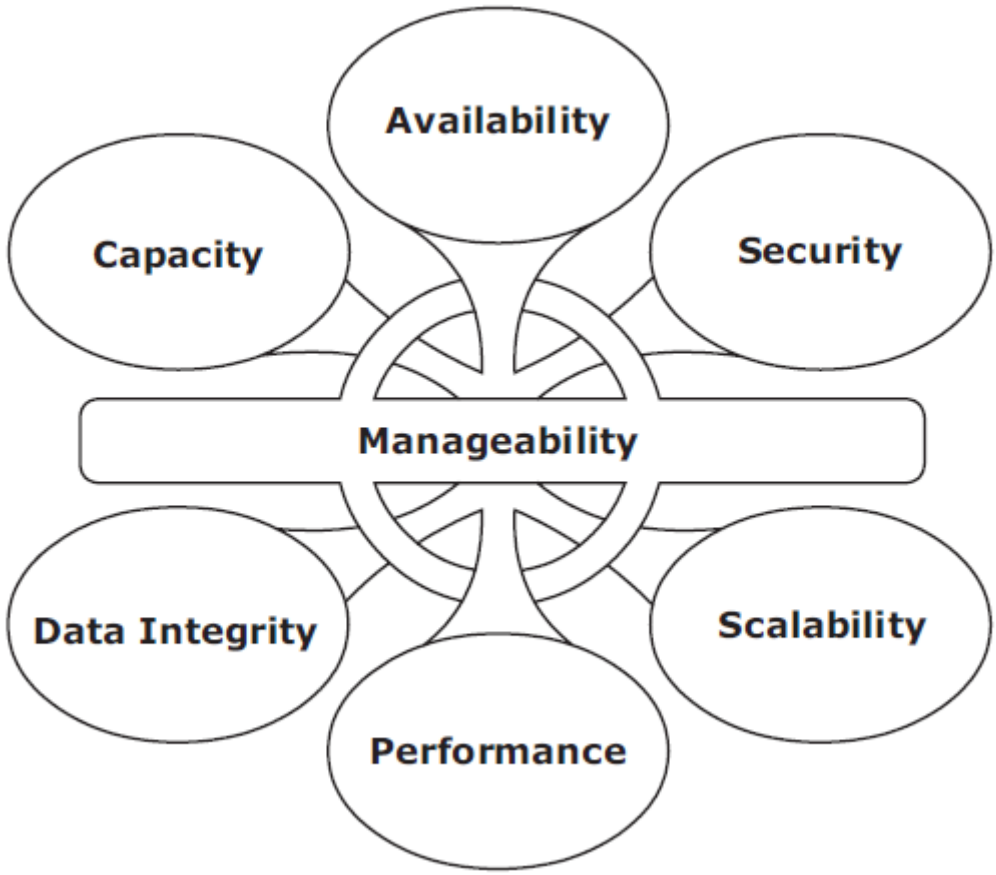
4 Performance: All the core elements of the data center should be able to provide optimal performance and service all processing requests at high speed. The infrastructure should be able to support performance requirements.

5 Data integrity: Data integrity refers to mechanisms such as error correction codes or parity bits which ensure that data is written to disk exactly as it was received. Any variation in data during its retrieval implies corruption, which may affect the operations of the organization.

6 Capacity: Data center operations require adequate resources to store and process large amounts of data efficiently. When capacity requirements increase, the data center must be able to provide additional capacity without interrupting availability, or, at the very least, with minimal disruption.

Capacity may be managed by reallocation of existing resources, rather than by adding new resources.

7 Manageability: A data center should perform all operations and activities in the most efficient manner. Manageability can be achieved through automation and the reduction of human (manual) intervention in common tasks.



Virtualization and cloud computing:

What Is Cloud Computing?

The National Institute of Standards defines cloud computing as "enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services)."

To be a cloud, NIST has determined it must have the following five essential characteristics:

- On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage.
- Broad network access: Capabilities are available over the network through multiple clients and devices.
- Resource pooling: The provider's computing resources are pooled to serve numerous consumers using a multi-tenant model.
- Rapid elasticity: Users can add or reduce capacity through software
- Measured service: Automatic control and optimization of resources detailing who is using what and how much.

Without those five essential characteristics, it is technically not a cloud.
The cloud model is comprised of three service models:

- Software as a Service (SaaS): The consumer can use the provider's applications running on a cloud infrastructure.
- Platform as a Service (PaaS): The consumer can deploy on the cloud infrastructure, applications created using programming languages, libraries, services or tools supported by the provider.
- Infrastructure as a Service (IaaS): The consumer can provision processing, storage, networks, and other computer resources to deploy and run arbitrary software.

There are four deployment models of the cloud:

- Private Cloud: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers.
- Community Cloud: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
- Public Cloud: The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization.
- Hybrid Cloud: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public).

What Is Virtualization?

Contrary to what some believe, virtualization is not cloud computing. It is, however, a fundamental technology that makes cloud computing work. While cloud computing and virtualization rely on similar models and principles, they are intrinsically different.
Simply put, virtualization can make one resource act like many, while cloud computing lets different users access a single pool of resources.
With virtualization, a single physical server can become multiple virtual machines, which are essentially isolated pieces of hardware with plenty of processing, memory, storage, and network capacity.
Each virtual machine can run independently while sharing the resources of a single host machine because they've been loaded into hypervisors. Hypervisors, also known as the abstraction layer, are used to separate physical resources from their virtual environments. Once resources are pooled together, they can be divided across many virtual environments as needed.

Cloud Computing vs. Virtualization

Deciding which to implement for your business depends on the type of business and the requirements you have.
For smaller companies, cloud computing is easier and more cost-effective to implement. Resources are accessed via the Internet rather than added to the network.
Many small businesses are turning to the cloud for applications such as customer relationship management (CRM), hosted voice over IP (VoIP) or off-site storage. The cost of using the cloud is much lower than implementing virtualization. Cloud computing also offers easier installation of applications and hardware, access to software they couldn't otherwise afford, and the ability to try software before they buy it. It requires a small investment to implement a cloud-based application.
For some businesses, virtualization is the smarter choice and can save money in several different ways:

- Adding many guests to one house maximizes resources, which means the business needs fewer servers. This cuts down on operational costs.
- Fewer servers mean fewer people to look after and manage servers. This helps to consolidate management, thereby reducing costs.
- Virtualization also adds another layer of protection for business continuity, since virtual machines will limit the damage to itself.

Software RAID

Software RAID uses host-based software to provide RAID functions. It is imple- mented at the operating-system level and does not use a dedicated hardware controller to manage the RAID array.

Software RAID implementations offer cost and simplicity benefits when compared with hardware RAID. However, they have the following limitations:

- **Performance:** Software RAID affects overall system performance. This is due to additional CPU cycles required to perform RAID calculations.
- **Supported features:** Software RAID does not support all RAID levels.
- **Operating system compatibility:** Software RAID is tied to the host operat- ing system; hence, upgrades to software RAID or to the operating system should be validated for compatibility. This leads to inflexibility in the data-processing environment.

Hardware RAID

In *hardware RAID* implementations, a specialized hardware controller is implemented either on the host or on the array.

Controller card RAID is a host-based hardware RAID implementation in which a specialized RAID controller is installed in the host, and disk drives are con- nected to it. Manufacturers also integrate RAID controllers on motherboards. A host-based RAID controller is not an efficient solution in a data center environ- ment with a large number of hosts.

The external RAID controller is an array-based hardware RAID. It acts as an interface between the host and disks. It presents storage volumes to the host, and the host manages these volumes as physical drives. The key functions of the RAID controllers are as follows:

- Management and control of disk aggregations
- Translation of I/O requests between logical disks and physical disks
- Data regeneration in the event of disk failures

RAID Array Components

A *RAID array* is an enclosure that contains a number of disk drives and sup- porting hardware to implement RAID. A subset of disks within a RAID array can be grouped to form logical associations called logical arrays, also known as a *RAID set* or a *RAID group* (see Figure 3-1).

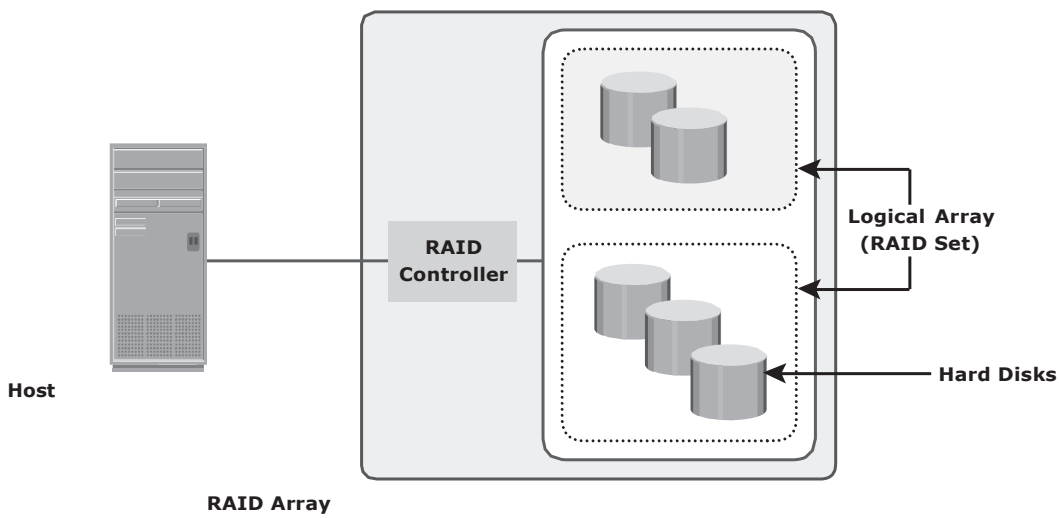


Figure 3-1: Components of a RAID array

RAID Techniques

RAID techniques — striping, mirroring, and parity — form the basis for defin-

ing various RAID levels. These techniques determine the data availability and performance characteristics of a RAID set.

Striping

Striping is a technique to spread data across multiple drives (more than one) to use the drives in parallel. All the read-write heads work simultaneously, allowing

more data to be processed in a shorter time and increasing performance, compared to reading and writing from a single disk.

Within each disk in a RAID set, a predefined number of contiguously addressable disk blocks are defined as a *strip*. The set of aligned strips that spans across all the disks within the RAID set is called a *stripe*. Figure 3-2 shows physical and logical representations of a striped RAID set.

Strip size (also called *stripe depth*) describes the number of blocks in a strip and is the maximum amount of data that can be written to or read from a single disk in the set, assuming that the accessed data starts at the beginning of the strip. All strips in a stripe have the same number of blocks. Having a smaller strip size means that data is broken into smaller pieces while spread across the disks. Stripe size is a multiple of strip size by the number of *data* disks in the RAID set. For example, in a five disk striped RAID set with a strip size of 64 KB, the stripe size is 320 KB(64KB ∞ 5). *Stripe width* refers to the number of data strips in a stripe. Striped RAID does not provide any data protection unless parity or mirroring is used, as discussed in the following sections.

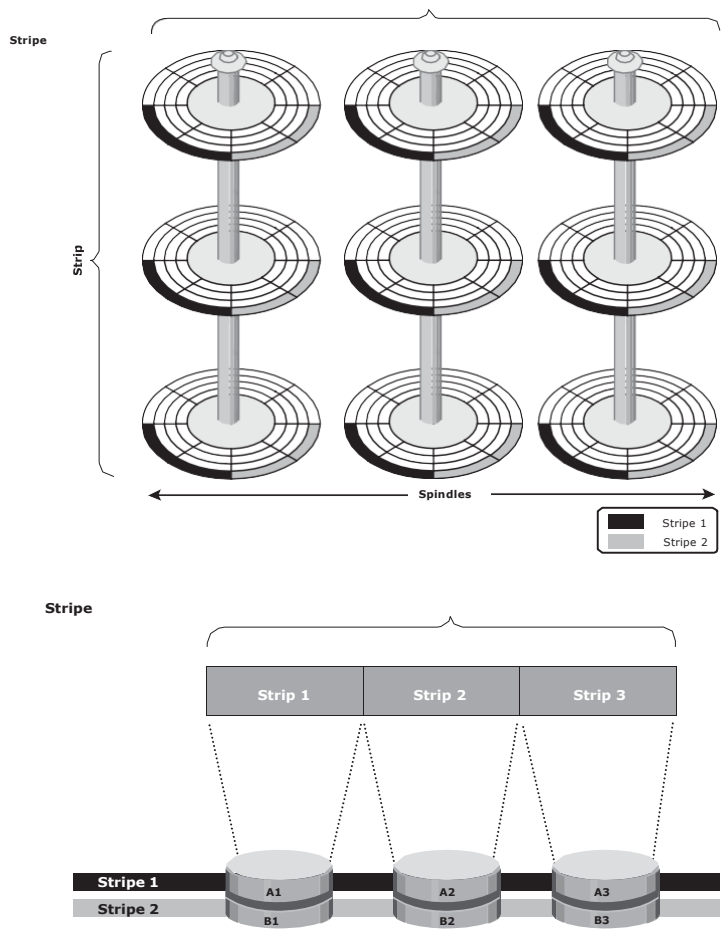


Figure 3-2: Striped RAID set

Mirroring

Mirroring is a technique whereby the same data is stored on two different disk drives, yielding two copies of the data. If one disk drive failure occurs, the data is intact on the surviving disk drive (see Figure 3-3) and the controller continues to service the host's data requests from the surviving disk of a mirrored pair.

When the failed disk is replaced with a new disk, the controller copies the data from the surviving disk of the mirrored pair. This activity is transparent to the host. In addition to providing complete data redundancy, mirroring enables fast recovery from disk failure. However, disk mirroring provides only data protection and is not a substitute for data backup. Mirroring constantly captures changes in the data, whereas a backup captures point-in-time images of the data.

Mirroring involves duplication of data — the amount of storage capacity needed is twice the amount of data being stored. Therefore, mirroring is considered expensive and is preferred for mission-critical applications that cannot afford the risk of any data loss. Mirroring improves read performance because read requests can be serviced by both disks. However, write performance is slightly lower than that in a single disk because each write request manifests as two writes on the disk drives. Mirroring does not deliver the same levels of write performance as a striped RAID.

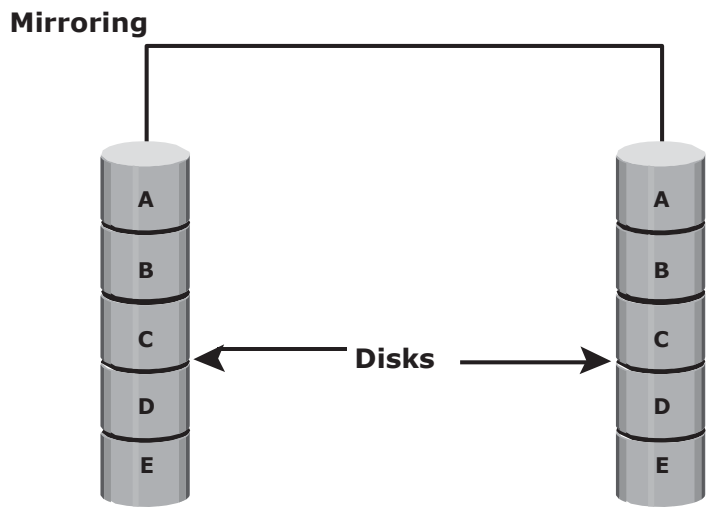


Figure 3-3: Mirrored disks in an array

Parity

Parity is a method to protect striped data from disk drive failure without the cost of mirroring. An additional disk drive is added to hold parity, a mathematical construct that allows re-creation of the missing data. Parity is a redundancy technique that ensures protection of data without maintaining a full set of duplicate data. Calculation of parity is a function of the RAID controller.

Parity information can be stored on separate, dedicated disk drives or distributed across all the drives in a RAID set. Figure 3-4 shows a parity RAID set. The first four disks, labeled "Data Disks," contain the data. The fifth disk, labeled "Parity Disk," stores the parity information, which, in this case, is the sum of the elements in each row. Now, if one of the data disks fails, the missing value can be calculated by subtracting the sum of the rest of the elements from the parity value. Here, for simplicity, the computation of parity is represented as an arithmetic sum of the data. However, parity calculation is a *bitwise XOR* operation.

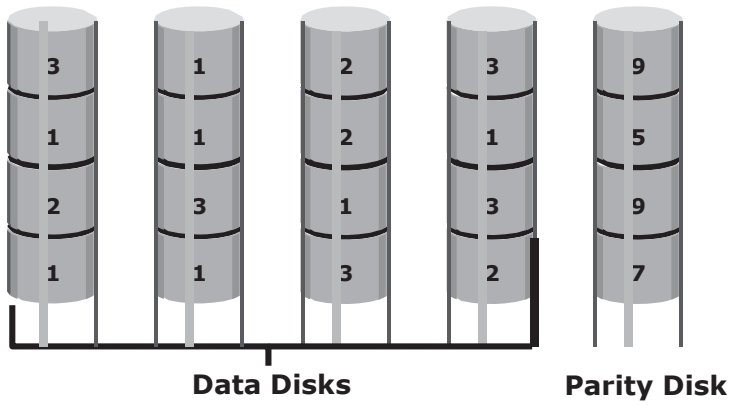


Figure 3-4: ParityRAID

Compared to mirroring, parity implementation considerably reduces the cost associated with data protection. Consider an example of a parity RAID configuration with five disks where four disks hold data, and the fifth holds the parity information. In this example, parity requires only 25 percent extra disk space compared to mirroring, which requires 100 percent extra disk space. However, there are some disadvantages of using parity. Parity information is generated from data on the data disk. Therefore, parity is recalculated every time there is a change in data. This recalculation is time-consuming and affects the performance of the RAID array.

For parity RAID, the stripe size calculation does not include the parity strip. For example in a five (4 + 1) disk parity RAID set with a strip size of 64 KB, the stripe size will be 256 KB (64 KB ∞ 4).

RAID Levels

Application performance, data availability requirements, and cost determine the RAID level selection. These RAID levels are defined on the basis of striping, mirroring, and parity techniques. Some RAID levels use a single technique, whereas others use a combination of techniques. Table 3-1 shows the commonly used RAID levels.

Table 3-1: Raid Levels

LEVELS	BRIEF DESCRIPTION
RAID 0	Striped set with no fault tolerance
RAID 1	Disk mirroring
Nested	Combinations of RAID levels. Example: RAID 1 + RAID 0
RAID 3	Striped set with parallel access and a dedicated parity disk
RAID 4	Striped set with independent disk access and a dedicated parity disk
RAID 5	Striped set with independent disk access and distributed parity
RAID 6	Striped set with independent disk access and dual distributed parity

RAID 0

RAID 0 configuration uses data striping techniques, where data is striped across all the disks within a RAID set. Therefore it utilizes the full storage capacity of a RAID set. To read data, all the strips are put back together by the controller. Figure 3-5 shows RAID 0 in an array in which data is striped across five disks. When the number of drives in the RAID set increases, performance improves

because more data can be read or written simultaneously. RAID 0 is a good option for applications that need high I/O throughput. However, if these applications require high availability during drive failures, RAID 0 does not provide data protection and availability.

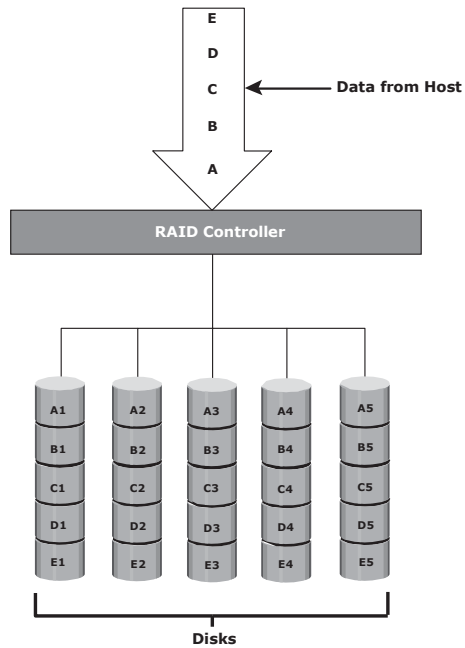


Figure 3-5: RAID 0

RAID 1

RAID 1 is based on the mirroring technique. In this RAID configuration, data is mirrored to provide fault tolerance (see Figure 3-6). A RAID 1 set consists of two disk drives and every write is written to both disks. The mirroring is transparent to the host. During disk failure, the impact on data recovery in RAID 1 is the least among all RAID implementations. This is because the RAID controller

uses the mirror drive for data recovery. RAID 1 is suitable for applications that require high availability and cost is no constraint.

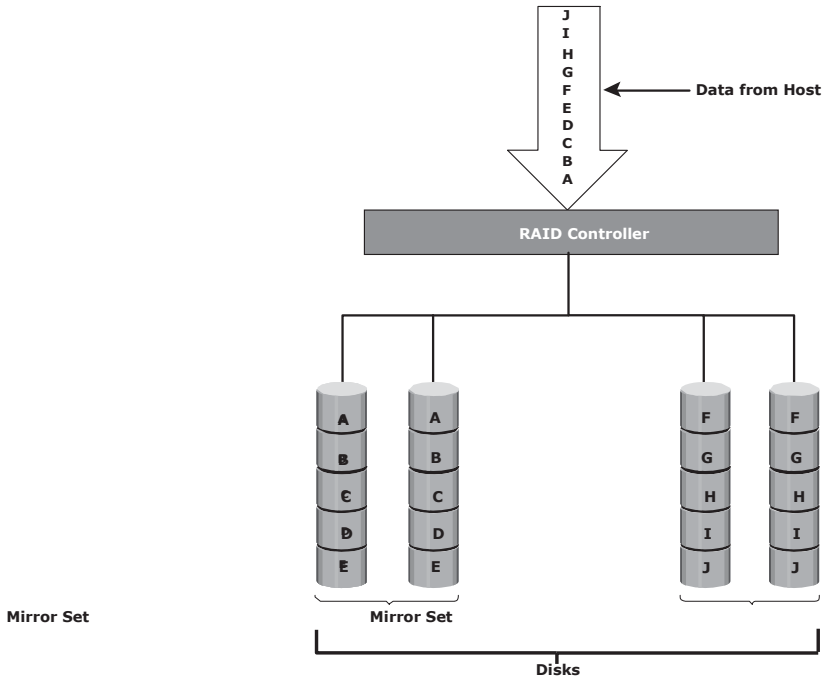


Figure 3-6: RAID 1

Nested RAID

Most data centers require data redundancy and performance from their RAID arrays. RAID 1+0 and RAID 0+1combine the performance benefits of RAID 0 with the redundancy benefits of RAID 1. They use striping and mirroring techniques and combine their benefits. These types of RAID require an even number of disks, the minimum being four (see Figure 3-7).

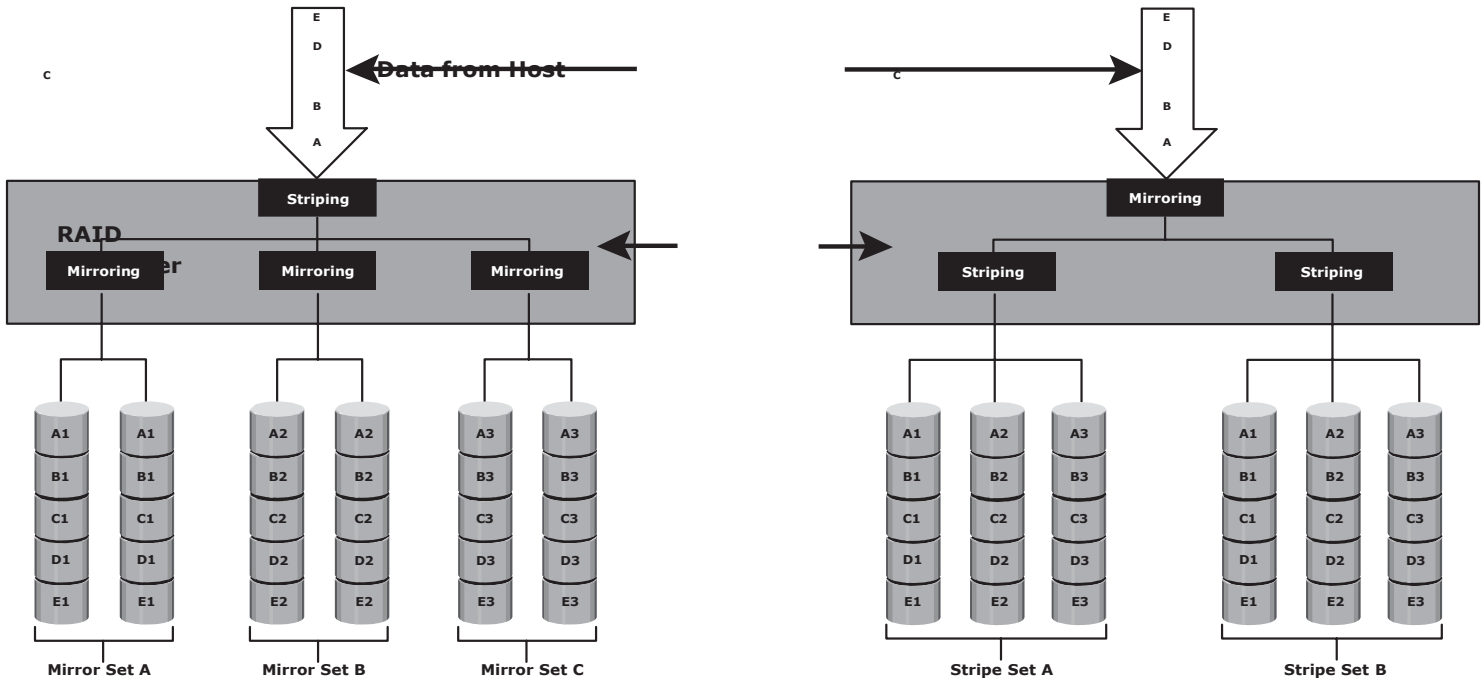


Figure 3-7: Nested RAID

RAID 1+0 is also known as RAID 10 (Ten) or RAID 1/0. Similarly, RAID 0+1 is also known as RAID 01 or RAID 0/1. RAID 1+0 performs well for workloads with small, random, write-intensive I/Os. Some applications that benefit from RAID 1+0 include the following:

- High transaction rate Online Transaction Processing (OLTP)
- Large messaging installations
- Database applications with write intensive random access workloads

A common misconception is that RAID 1+0 and RAID 0+1 are the same. Under normal conditions, RAID levels 1+0 and 0+1 offer identical benefits. However, rebuild operations in the case of disk failure differ between the two.

RAID 1+0 is also called striped mirror. The basic element of RAID 1+0 is a mirrored pair, which means that data is first mirrored and then both copies of the data are striped across multiple disk drive pairs in a RAID set. When replacing a failed drive, only the mirror is rebuilt. In other words, the disk array controller uses the surviving drive in the mirrored pair for data recovery and continuous operation. Data from the surviving disk is copied to the replacement disk.

To understand the working of RAID 1+0, consider an example of six disks forming a RAID 1+0 (RAID 1 first and then RAID 0) set. These six disks are paired into three sets of two disks, where each set acts as a RAID 1 set (mirrored pair of disks). Data is then striped across all the three mirrored sets to form RAID 0. Following are the steps performed in RAID 1+0 (see Figure 3-7 [a]):

Drives 1+2 = RAID 1 (Mirror Set A)
Drives 3+4 = RAID 1 (Mirror Set B)
Drives 5+6 = RAID 1 (Mirror Set C)

Now, RAID 0 striping is performed across sets A through C. In this configuration, if drive 5 fails, then the mirror set C alone is affected. It still has drive 6 and continues to function and the entire RAID 1+0 array also keeps functioning. Now, suppose drive 3 fails while drive 5 was being replaced. In this case the array still continues to function because drive 3 is in a different mirror set. So, in this configuration, up to three drives can fail without affecting the array, as long as they are all in different mirror sets.

RAID 0+1 is also called a mirrored stripe. The basic element of RAID 0+1 is a stripe. This means that the process of striping data across disk drives is performed initially, and then the entire stripe is mirrored. In this configuration if one drive fails, then the entire stripe is faulted. Consider the same example of six disks to understand the working of RAID 0+1 (that is, RAID 0 first and then RAID 1). Here, six disks are paired into two sets of three disks each. Each of these sets, in turn, act as a RAID 0 set that contains three disks and then these

two sets are mirrored to form RAID 1. Following are the steps performed in RAID 0+1 (see Figure 3-7 [b]):

Drives 1 + 2 + 3 = RAID 0 (Stripe Set A)
Drives 4 + 5 + 6 = RAID 0 (Stripe Set B)

Now, these two stripe sets are mirrored. If one of the drives, say drive 3, fails, the entire stripe set A fails. A rebuild operation copies the entire stripe, copying the data from each disk in the healthy stripe to an equivalent disk in the failed stripe. This causes increased and unnecessary I/O load on the surviving disks and makes the RAID set more vulnerable to a second disk failure.

RAID 3

RAID 3 stripes data for performance and uses parity for fault tolerance. Parity information is stored on a dedicated drive so that the data can be reconstructed if a drive fails in a RAID set. For example, in a set of five disks, four are used for data and one for parity. Therefore, the total disk space required is 1.25 times the size of the data disks. RAID 3 always reads and writes complete stripes of data across all disks because the drives operate in parallel. There are no partial writes that update one out of many strips in a stripe. Figure 3-8 illustrates the RAID 3 implementation.

RAID 3 provides good performance for applications that involve large sequential data access, such as data backup or video streaming.

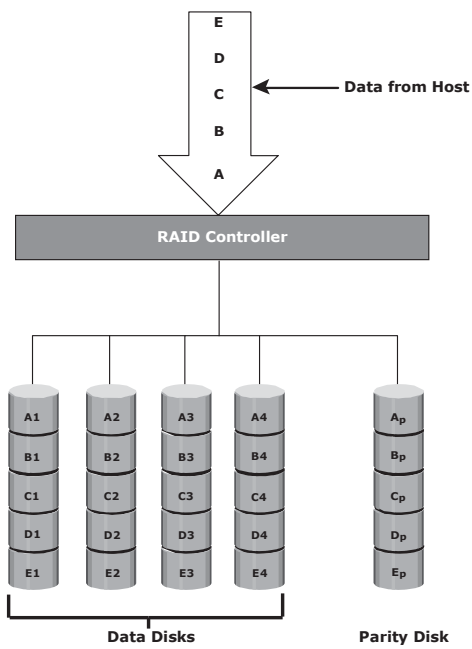


Figure 3-8: RAID 3

RAID 4

Similar to RAID 3, RAID 4 stripes data for high performance and uses parity for improved fault tolerance. Data is striped across all disks except the parity disk in the array. Parity information is stored on a dedicated disk so that the data can be rebuilt if a drive fails.

Unlike RAID 3, data disks in RAID 4 can be accessed independently so that specific data elements can be read or written on a single disk without reading or writing an entire stripe. RAID 4 provides good read throughput and reasonable write throughput.

RAID 5

RAID 5 is a versatile RAID implementation. It is similar to RAID 4 because it uses striping. The drives (strips) are also independently accessible. The difference between RAID 4 and RAID 5 is the parity location. In RAID 4, parity is written to a dedicated drive, creating a write bottleneck for the parity disk. In RAID 5, parity is distributed across all disks to overcome the write bottleneck of a dedicated parity disk. Figure 3-9 illustrates the RAID 5 implementation.

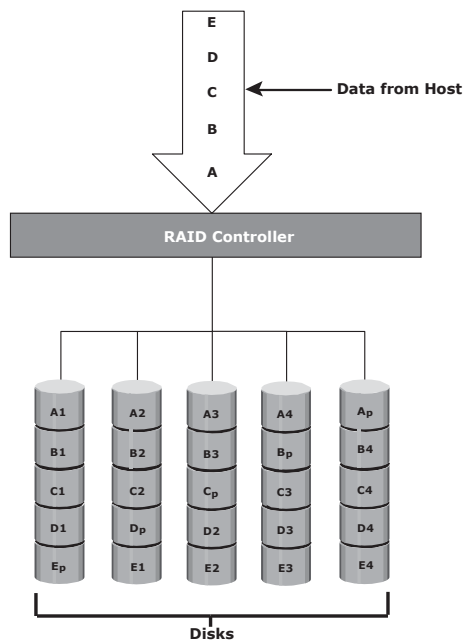


Figure 3-9: RAID 5

RAID 5 is good for random, read-intensive I/O applications and preferred for messaging, data mining, medium-performance media serving, and relational database management system (RDBMS) implementations, in which database administrators (DBAs) optimize data access.

RAID 6

RAID 6 works the same way as RAID 5, except that RAID 6 includes a second parity element to enable survival if two disk failures occur in a RAID set (see Figure 3-10). Therefore, a RAID 6 implementation requires at least four disks. RAID 6 distributes the parity across all the disks. The write penalty (explained later in this chapter) in RAID 6 is more than that in RAID 5; therefore, RAID 5 writes perform better than RAID 6. The rebuild operation in RAID 6 may take longer than that in RAID 5 due to the presence of two parity sets.

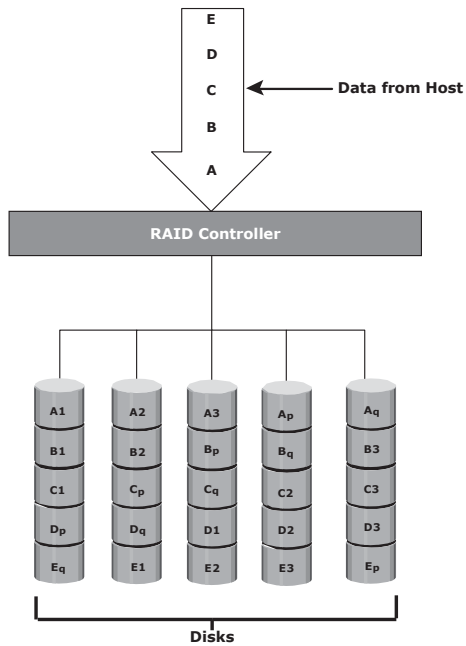


Figure 3-10: RAID 6

3.2 RAID Impact on Disk Performance

When choosing a RAID type, it is imperative to consider its impact on disk performance and application IOPS.

In both mirrored and parity RAID configurations, every write operation translates into more I/O overhead for the disks, which is referred to as a *write penalty*. In a RAID 1 implementation, every write operation must be performed on two disks configured as a mirrored pair, whereas in a RAID 5 implementation, a write operation may manifest as four I/O operations. When performing I/Os to a disk configured with RAID 5, the controller has to read, recalculate, and write a parity segment for every data write operation.

Figure 3-11 illustrates a single write operation on RAID 5 that contains a group of five disks.

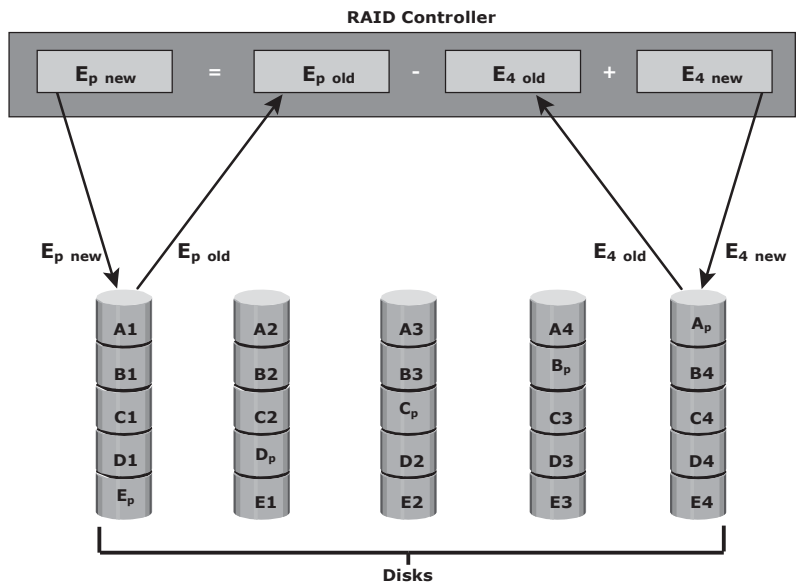


Figure 3-11: Write penalty in RAID 5

The parity (P) at the controller is calculated as follows:

$$E_p = E_1 + E_2 + E_3 + E_4 \text{ (XOR operations)}$$

Whenever the controller performs a write I/O, parity must be computed by reading the old parity (E_p old) and the old data ($E_{4\ old}$) from the disk, which means two read I/Os. Then, the new parity (E_p new) is computed as follows:

$$E_{p\ new} = E_{p\ old} - E_{4\ old} + E_{4\ new} \text{ (XOR operations)}$$

After computing the new parity, the controller completes the write I/O by writing the new data and the new parity onto the disks, amounting to two write I/Os. Therefore, the controller performs two disk reads and two disk writes for every write operation, and the write penalty is 4.

In RAID 6, which maintains dual parity, a disk write requires three read operations: two parity and one data. After calculating both new parities, the

controller performs three write operations: two parity and an I/O. Therefore, in a RAID 6 implementation, the controller performs six I/O operations for each write I/O, and the write penalty is 6.

Application IOPS and RAID Configurations

When deciding the number of disks required for an application, it is important to consider the impact of RAID based on IOPS generated by the application. The total disk load should be computed by considering the type of RAID configuration and the ratio of read compared to write from the host.

The following example illustrates the method to compute the disk load in different types of RAID.

Consider an application that generates 5,200 IOPS, with 60 percent of them being reads.

The disk load in RAID 5 is calculated as follows:

RAID 5 disk load (reads + writes) = 0.6 ∞ 5,200 + **4** ∞ (0.4 ∞ 5,200) [because the write penalty for RAID 5 is 4]

= 3,120 + 4 ∞ 2,080

= 3,120 + 8,320

= 11,440 IOPS

The disk load in RAID 1 is calculated as follows:

RAID 1 disk load = 0.6 ∞ 5,200 + **2** ∞ (0.4 ∞ 5,200) [because every write manifests as two writes to the disks]

= 3,120 + 2 ∞ 2,080

= 3,120 + 4,160

= 7,280 IOPS

The computed disk load determines the number of disks required for the application. If in this example a disk drive with a specification of a maximum 180 IOPS needs to be used, the number of disks required to meet the workload for the RAID configuration would be as follows:

RAID 5: 11,440/180 = 64 disks

RAID 1: 7,280/180 = 42 disks (approximated to the nearest even number)

RAID Comparison

Table 3-2 compares the common types of RAID levels.

Table 3-2: Comparison of Common RAID Types

RAID	MIN. DISKS	STORAGE EFFICIENCY %	COST	READ PERFORMANCE	WRITE PERFORMANCE	WRITE PENALTY	PROTECTION
0	2	100	Low	Good for both random and sequential reads	Good	No	No protection
1	2	50	High	Better than single disk	Slower than single disk because every write must be committed to all disks	Moderate	Mirror protection
3	3	[(n-1)/n] ∞ 100 where n= number of disks	Moderate	Fair for random reads and good for sequential reads	Poor to fair for small random writes and fair for large, sequential writes	High	Parity protection for single disk failure
4	3	[(n-1)/n] ∞ 100 where n= number of disks	Moderate	Good for random and sequential reads	Fair for random and sequential writes	High	Parity protection for single disk failure
5	3	[(n-1)/n] ∞ 100 where n= number of disks	Moderate	Good for random and sequential reads	Fair for random and sequential writes	High	Parity protection for single disk failure
6	4	[(n-2)/n] ∞ 100 where n= number of disks	Moderate but more than RAID 5.	Good for random and sequential reads	Poor to fair for random writes and fair for sequential writes	Very High	Parity protection for two disk failures
1+0 and 0+1	4	50	High	Good	Good	Moderate	Mirror protection

Hot Spares

A *hot spare* refers to a spare drive in a RAID array that temporarily replaces a failed disk drive by taking the identity of the failed disk drive. With the hot spare, one of the following methods of data recovery is performed depending on the RAID implementation:

- If parity RAID is used, the data is rebuilt onto the hot spare from the parity and the data on the surviving disk drives in the RAID set.
- If mirroring is used, the data from the surviving mirror is used to copy the data onto the hot spare.

When a new disk drive is added to the system, data from the hot spare is copied to it. The hot spare returns to its idle state, ready to replace the next failed drive. Alternatively, the hot spare replaces the failed disk drive perma- nently. This means that it is no longer a hot spare, and a new hot spare must be configured on the array.

A hot spare should be large enough to accommodate data from a failed drive. Some systems implement multiple hot spares to improve data availability.

A hot spare can be configured as automatic or user initiated, which specifies how it will be used in the event of disk failure. In an automatic configuration, when the recoverable error rates for a disk exceed a predetermined threshold, the disk subsystem tries to copy data from the failing disk to the hot spare auto- matically. If this task is completed before the damaged disk fails, the subsystem switches to the hot spare and marks the failing disk as unusable. Otherwise, it uses parity or the mirrored disk to recover the data. In the case of a user-initiated configuration, the administrator has control of the rebuild process. For example, the rebuild could occur overnight to prevent any degradation of system perfor- mance. However, the system is at risk of data loss if another disk failure occurs.

Module-2

Storage Networking Technologies and Virtualization

Fibre Channel: Overview

The FC architecture forms the fundamental construct of the FC SAN infrastructure. *Fibre Channel* is a high-speed network technology that runs on high-speed optical fiber cables and serial copper cables. The FC technology was developed to meet the demand for increased speeds of data transfer between servers and mass storage systems. Although FC networking was introduced in 1988, the FC standardization process began when the American National Standards Institute (ANSI) chartered the Fibre Channel Working Group (FCWG). By 1994, the new high-speed computer interconnection standard was developed and the Fibre Channel Association (FCA) was founded with 70 charter member companies. Technical Committee T11, which is the committee within International Committee for Information Technology Standards (INCITS), is responsible for Fibre Channel interface standards.

High data transmission speed is an important feature of the FC networking technology. The initial implementation offered a throughput of 200 MB/s (equivalent to a raw bit rate of 1Gb/s), which was greater than the speeds of Ultra SCSI (20 MB/s), commonly used in DAS environments. In comparison with Ultra SCSI, FC is a significant leap in storage networking technology. The latest FC implementations of 16 GFC (Fibre Channel) offer a throughput of 3200 MB/s (raw bit rates of 16 Gb/s), whereas Ultra640 SCSI is available with a throughput of 640 MB/s. The FC architecture is highly scalable, and theoretically, a single FC network can accommodate approximately 15 million devices.

The SAN and Its Evolution

A SAN carries data between servers (or *hosts*) and storage devices through Fibre Channel network (see Figure 5-1). A SAN enables storage consolidation and enables storage to be shared across multiple servers. This improves the utilization of storage resources compared to direct-attached storage architecture and reduces the total amount of storage an organization needs to purchase and manage. With consolidation, storage management becomes centralized and less complex, which further reduces the cost of managing information. SAN also enables organizations to connect geographically dispersed servers and storage.

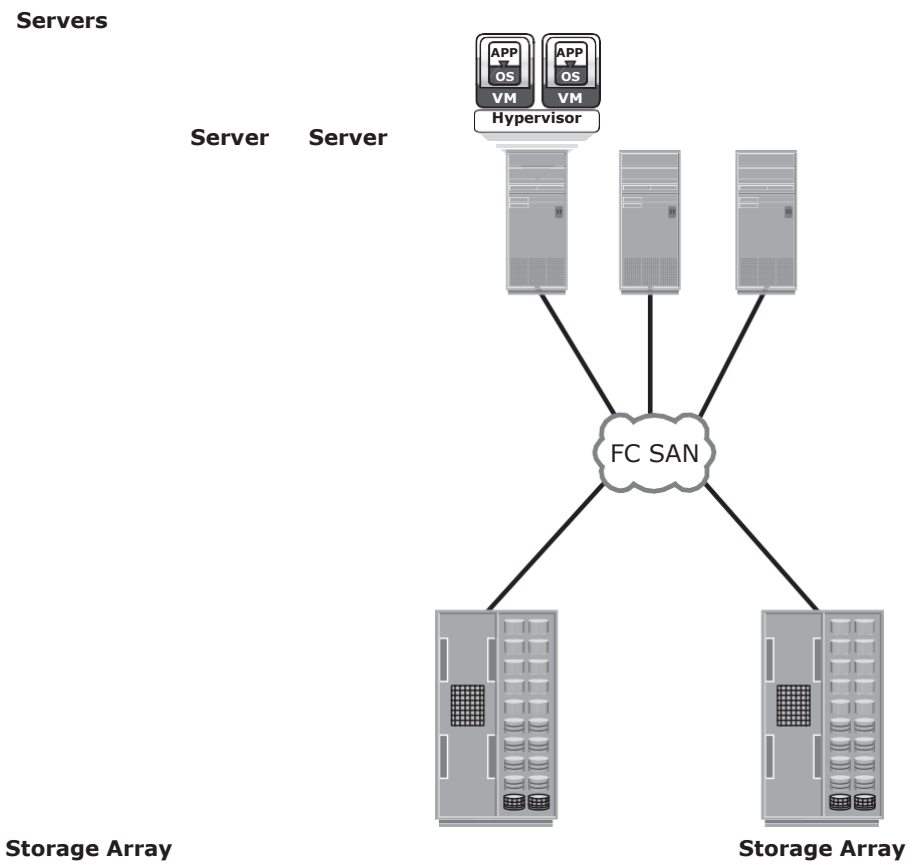


Figure 5-1: FC SAN implementation

In its earliest implementation, the FC SAN was a simple grouping of hosts and storage devices connected to a network using an FC hub as a connectivity device. This configuration of an FC SAN is known as a *Fibre Channel Arbitrated*

Loop (FC-AL). Use of hubs resulted in isolated FC-AL SAN islands because hubs provide limited connectivity and bandwidth.

The inherent limitations associated with hubs gave way to high-performance FC *switches*. Use of switches in SAN improved connectivity and performance and enabled FC SANs to be highly scalable. This enhanced data accessibility to applications across the enterprise. Now, FC-AL has been almost abandoned for FC SANs due to its limitations but still survives as a back-end connectivity option to disk drives. Figure 5-2 illustrates the FC SAN evolution from FC-AL to enterprise SANs.

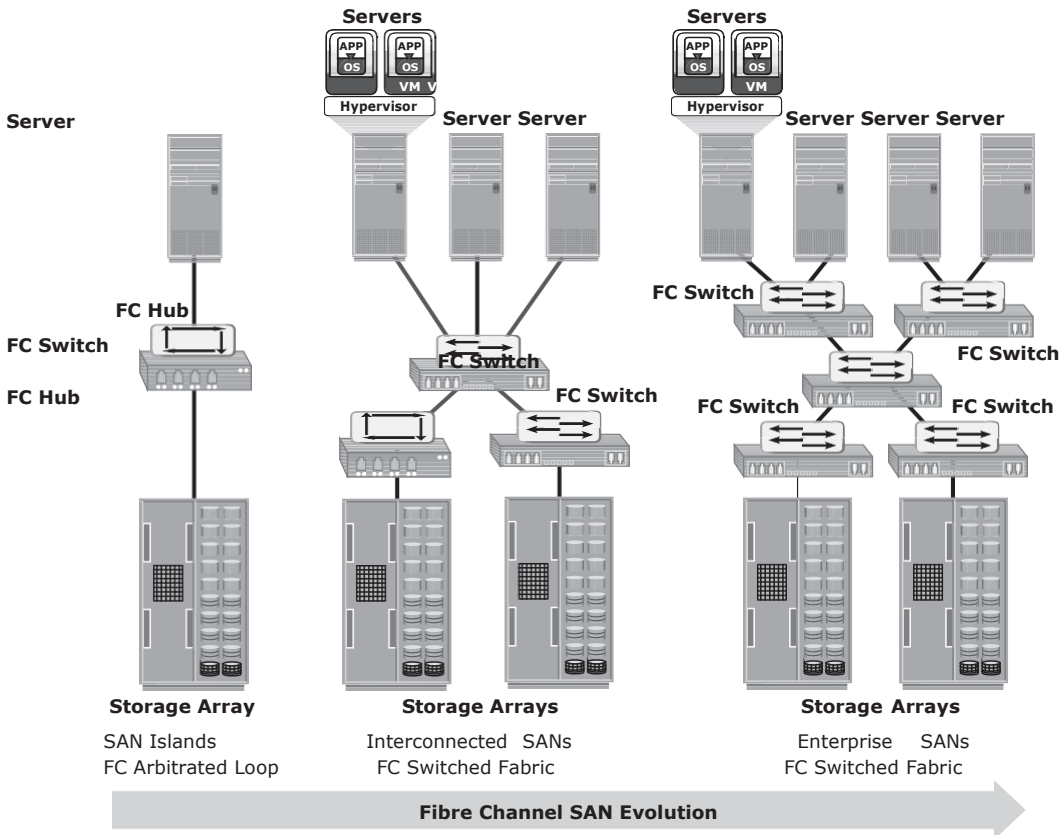


Figure 5-2: FC SAN evolution

Components of FC SAN

FC SAN is a network of servers and shared storage devices. Servers and storage are the end points or devices in the SAN (called *nodes*). FC SAN infrastructure consists of node ports, cables, connectors, and interconnecting devices (such as FC switches or hubs), along with SAN management software.

Node Ports

In a Fibre Channel network, the end devices, such as hosts, storage arrays, and tape libraries, are all referred to as *nodes*. Each node is a source or destination of information. Each node requires one or more ports to provide a physical interface for communicating with other nodes. These ports are integral components of host adapters, such as HBA, and storage front-end controllers or adapters. In an FC environment a port operates in full-duplex data transmission mode with a *transmit* (Tx) link and a *receive* (Rx) link (see Figure 5-3).

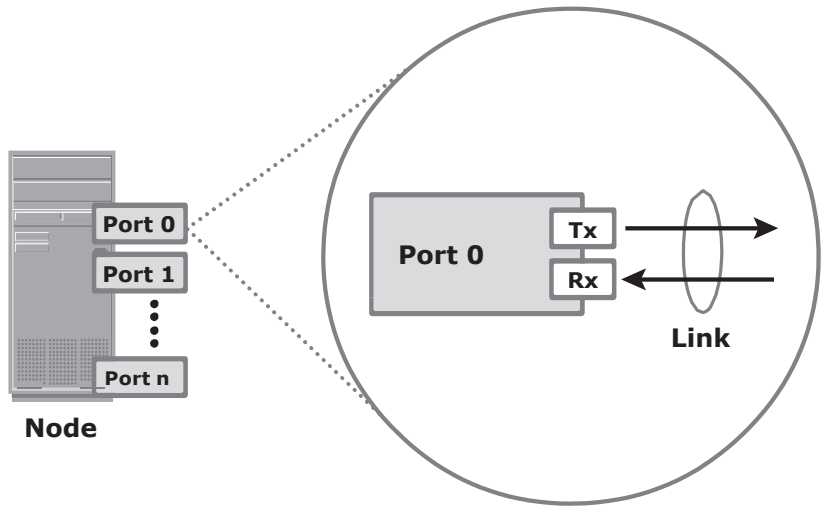


Figure 5-3: Nodes, ports, and links

Cables and Connectors

SAN implementations use optical fiber cabling. Copper can be used for shorter distances for back-end connectivity because it provides an acceptable signal-to-noise ratio for distances up to 30 meters. Optical fiber cables carry data in the form of light. There are two types of optical cables: multimode and single-mode. *Multimode fiber* (MMF) cable carries multiple beams of light projected at different angles simultaneously onto the core of the cable (see Figure 5-4 [a]). Based on the bandwidth, multimode fibers are classified as OM1 (62.5µm core), OM2 (50µm core), and laser-optimized OM3 (50µm core). In an MMF transmission, multiple light beams traveling inside the cable tend to disperse and collide. This collision weakens the signal strength after it travels a certain distance — a process known as *modal dispersion*. An MMF cable is typically used for short distances because of signal degradation (attenuation) due to modal dispersion. *Single-mode fiber* (SMF) carries a single ray of light projected at the center of the core (see Figure 5-4 [b]). These cables are available in core diameters of 7 to 11microns;

the most common size is 9 microns. In an SMF transmission, a single light beam travels in a straight line through the core of the fiber. The small core and the single light wave help to limit modal dispersion. Among all types of fiber cables, single-mode provides minimum signal attenuation over maximum distance (up to 10 km). A single-mode cable is used for long-distance cable runs, and distance usually depends on the power of the laser at the transmitter and sensitivity of the receiver.

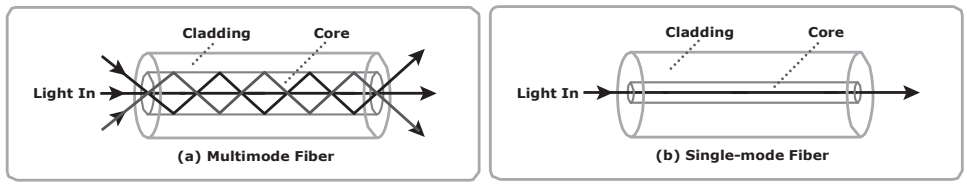


Figure 5-4: Multimode fiber and single-mode fiber

MMFs are generally used within data centers for shorter distance runs, whereas SMFs are used for longer distances.

A connector is attached at the end of a cable to enable swift connection and disconnection of the cable to and from a port. A *Standard connector* (SC) (see Figure 5-5 [a]) and a *Lucent connector* (LC) (see Figure 5-5 [b]) are two commonly used connectors for fiber optic cables. *Straight Tip* (ST) is another fiber-optic connector, which is often used with fiber patch panels (see Figure 5.5 [c]).

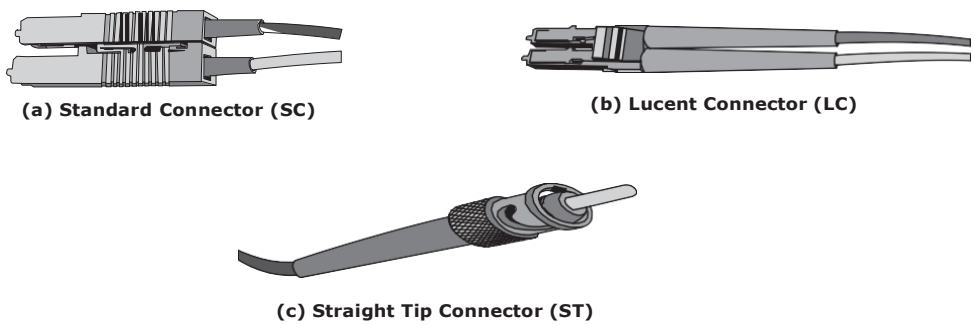


Figure 5-5: SC, LC, and ST connectors

Interconnect Devices

FC hubs, switches, and directors are the interconnect devices commonly used in FC SAN.

Hubs are used as communication devices in FC-AL implementations. Hubs physically connect nodes in a logical loop or a physical star topology. All the nodes must share the loop because data travels through all the connection points. Because of the availability of low-cost and high-performance switches, hubs are no longer used in FC SANs.

Switches are more intelligent than hubs and directly route data from one physical port to another. Therefore, nodes do not share the bandwidth. Instead, each node has a dedicated communication path.

Directors are high-end switches with a higher port count and better fault-tolerance capabilities.

Switches are available with a fixed port count or with modular design. In a modular switch, the port count is increased by installing additional port cards to open slots. The architecture of a director is always modular, and its port count is increased by inserting additional line cards or blades to the director's chassis. High-end switches and directors contain redundant components to provide high availability. Both switches and directors have management ports (Ethernet or serial) for connectivity to SAN management servers.

A port card or blade has multiple ports for connecting nodes and other FC switches. Typically, a Fibre Channel transceiver is installed at each port slot that houses the transmit (Tx) and receive (Rx) link. In a transceiver, the Tx and Rx links share common circuitry. Transceivers inside a port card are connected to an application specific integrated circuit, also called port ASIC. Blades in a director usually have more than one ASIC for higher throughput.

SAN Management Software

SAN management software manages the interfaces between hosts, interconnect devices, and storage arrays. The software provides a view of the SAN environment and enables management of various resources from one central console.

It provides key management functions, including mapping of storage devices, switches, and servers, monitoring and generating alerts for discovered devices, and *zoning* (discussed in section 5.9 "Zoning" later in this chapter).

FC Connectivity

The FC architecture supports three basic interconnectivity options: point-to-point, arbitrated loop, and Fibre Channel switched fabric.

5.1.1 Point-to-Point

Point-to-point is the simplest FC configuration — two devices are connected directly to each other, as shown in Figure 5-6. This configuration provides a dedicated connection for data transmission between nodes. However, the point-to-point configuration offers limited connectivity, because only two devices can communicate with each other at a given time. Moreover, it cannot be scaled to accommodate a large number of nodes. Standard DAS uses point-to-point connectivity.

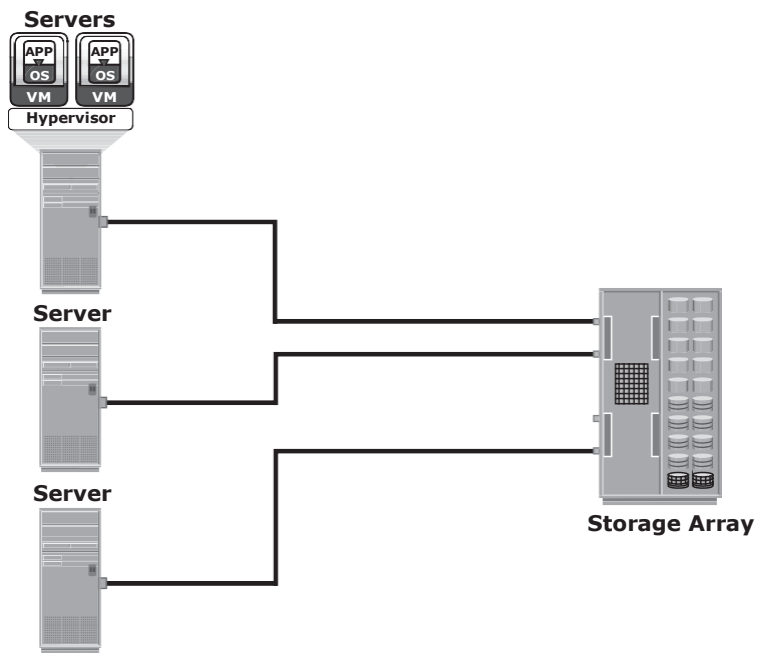


Figure 5-6: Point-to-point connectivity

Fibre Channel Arbitrated Loop

In the FC-AL configuration, devices are attached to a shared loop. FC-AL has the characteristics of a token ring topology and a physical star topology. In FC-AL, each device contends with other devices to perform I/O operations. Devices on the loop must "arbitrate" to gain control of the loop. At any given time, only one device can perform I/O operations on the loop (see Figure 5-7).

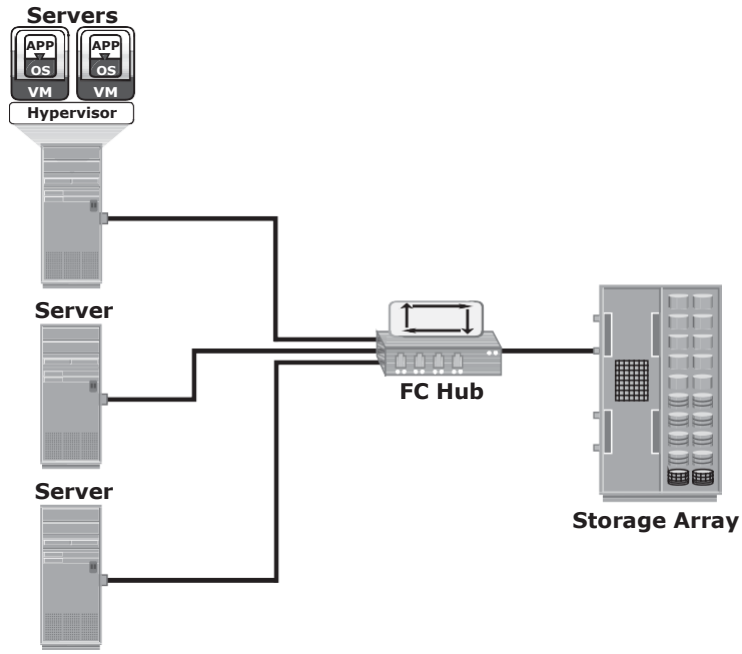


Figure 5-7: Fibre Channel Arbitrated Loop

As a loop configuration, FC-AL can be implemented without any interconnecting devices by directly connecting one device to another two devices in a ring through cables.

However, FC-AL implementations may also use hubs whereby the arbitrated loop is physically connected in a star topology.

The FC-AL configuration has the following limitations in terms of scalability:

- FC-AL shares the loop and only one device can perform I/O operations at a time. Because each device in a loop must wait for its turn to process an I/O request, the overall performance in FC-AL environments is low.
- FC-AL uses only 8-bits of 24-bit Fibre Channel addressing (the remaining 16-bits are masked) and enables the assignment of 127 valid addresses to the ports. Hence, it can support up to 127 devices on a loop. One address is reserved for optionally connecting the loop to an FC switch port. Therefore, up to 126 nodes can be connected to the loop.
- Adding or removing a device results in loop re-initialization, which can cause a momentary pause in loop traffic.

Fibre Channel Switched Fabric

Unlike a loop configuration, a Fibre Channel switched fabric (FC-SW) network provides dedicated data path and scalability. The addition or removal of a device

in a switched fabric is minimally disruptive; it does not affect the ongoing traffic between other devices.

FC-SW is also referred to as *fabric connect*. A fabric is a logical space in which all nodes communicate with one another in a network. This virtual space can be created with a switch or a network of switches. Each switch in a fabric contains a unique domain identifier, which is part of the fabric's addressing scheme. In FC-SW, nodes do not share a loop; instead, data is transferred through a dedicated path between the nodes. Each port in a fabric has a unique 24-bit Fibre Channel address for communication. Figure 5-8 shows an example of the FC-SW fabric. In a switched fabric, the link between any two switches is called an *Interswitch link* (ISL). ISLs enable switches to be connected together to form a single, larger fabric. ISLs are used to transfer host-to-storage data and fabric management traffic from one switch to another. By using ISLs, a switched fabric can be expanded to connect a large number of nodes.

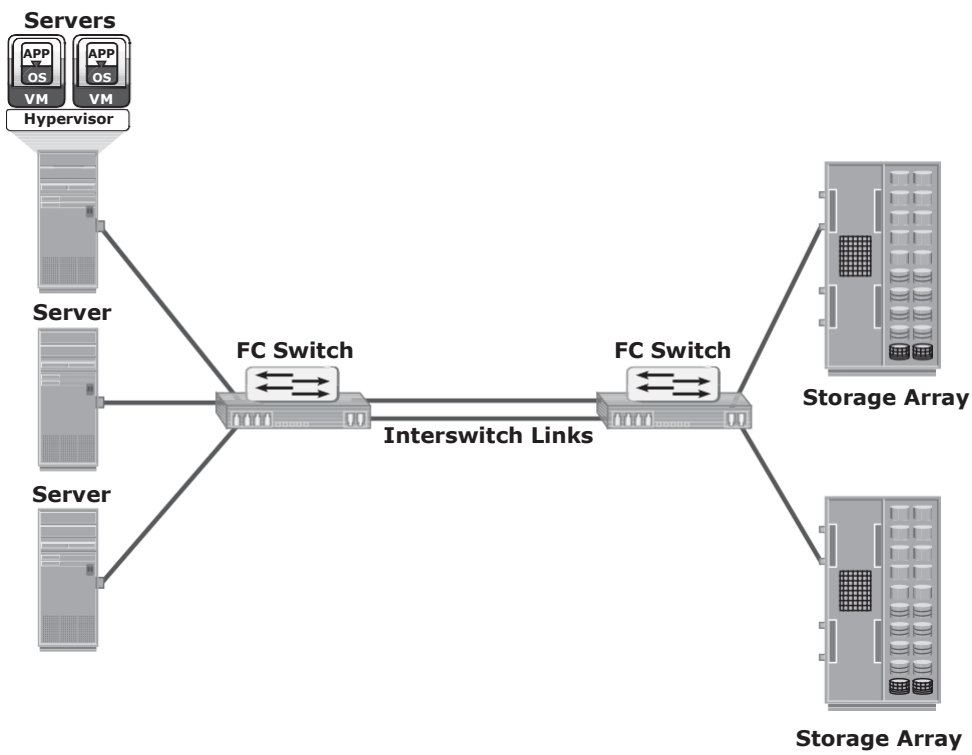


Figure 5-8: Fibre Channel switched fabric

A fabric can be described by the number of tiers it contains. The number of tiers in a fabric is based on the number of switches traversed between two points that are farthest from each other. This number is based on the infrastructure

constructed by the fabric instead of how the storage and server are connected across the switches.

When the number of tiers in a fabric increases, the distance that the fabric management traffic must travel to reach each switch also increases. This increase in the distance also increases the time taken to propagate and complete a fabric reconfiguration event, such as the addition of a new switch or a zone set propagation event. Figure 5-9 illustrates two-tier and three-tier fabric architecture.

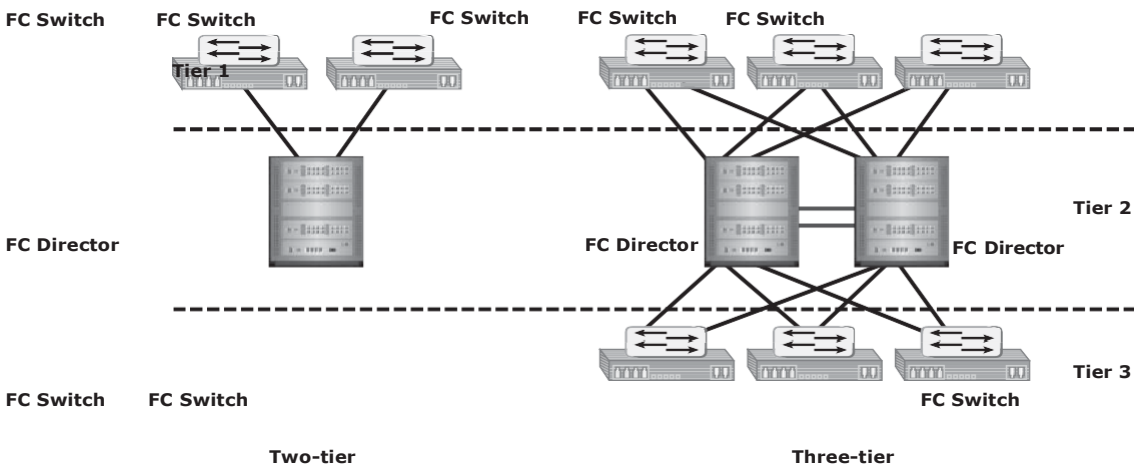


Figure 5-9: Tiered structure of Fibre Channel switched fabric

FC-SW Transmission

FC-SW uses switches that can switch data traffic between nodes directly through switch ports. Frames are routed between source and destination by the fabric. As shown in Figure 5-10, if node B wants to communicate with node D, the nodes should individually login first and then transmit data via the FC-SW. This link is considered a dedicated connection between the initiator and the target.

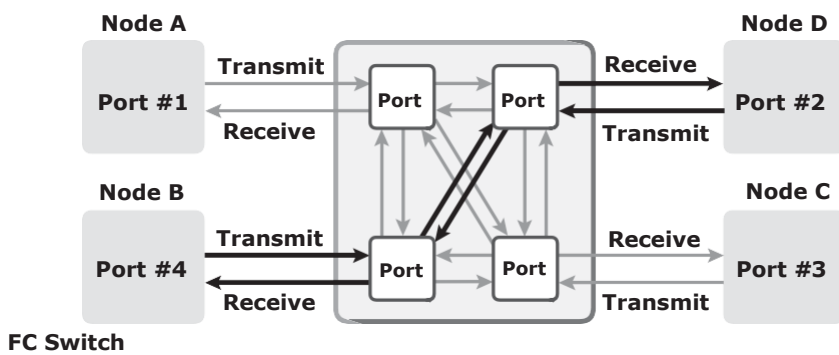


Figure 5-10: Data transmission in Fibre Channel switched fabric

Switched Fabric Ports

Ports in a switched fabric can be one of the following types:

- **N_Port:** An end point in the fabric. This port is also known as the *node port*. Typically, it is a host port (HBA) or a storage array port connected to a switch in a switched fabric.
- **E_Port:** A port that forms the connection between two FC switches. This port is also known as the *expansion port*. The E_Port on an FC switch connects to the E_Port of another FC switch in the fabric through ISLs.
- **F_Port:** A port on a switch that connects an N_Port. It is also known as a *fabric port*.
- **G_Port:** A generic port on a switch that can operate as an E_Port or an F_Port and determines its functionality automatically during initialization.

Figure 5-11 shows various FC ports located in a switched fabric.

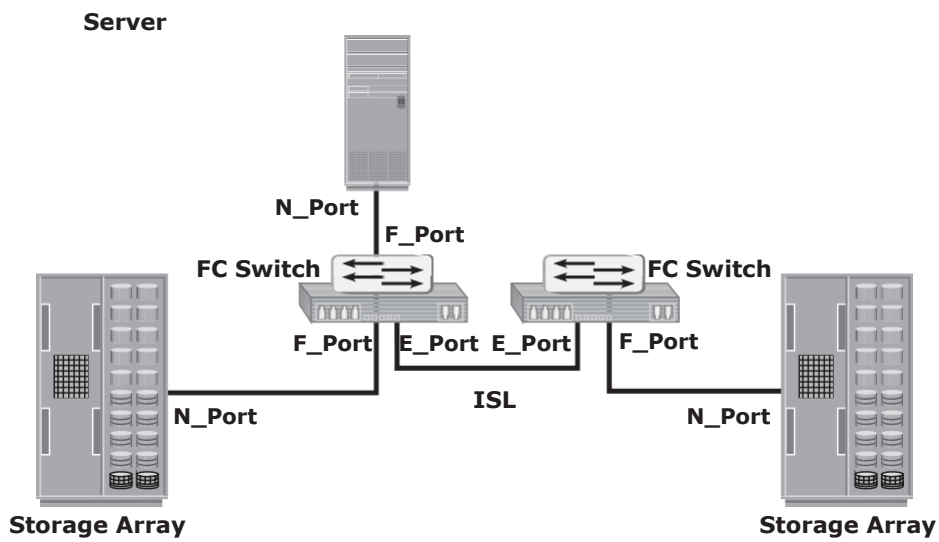


Figure 5-11: Switched fabric ports

Fibre Channel Architecture

Traditionally, host computer operating systems have communicated with peripheral devices over channel connections, such as ESCON and SCSI. Channel technologies provide high levels of performance with low protocol overheads. Such performance is achievable due to the static nature of channels and the high level of hardware and software integration provided by the channel technologies.

However, these technologies suffer from inherent limitations in terms of the number of devices that can be connected and the distance between these devices. In contrast to channel technology, network technologies are more flexible and provide greater distance capabilities. Network connectivity provides greater scalability and uses shared bandwidth for communication. This flexibility results in greater protocol overhead and reduced performance.

The FC architecture represents true channel/network integration and captures some of the benefits of both channel and network technology. FC SAN uses the *Fibre Channel Protocol* (FCP) that provides both channel speed for data transfer with low protocol overhead and scalability of network technology.

FCP forms the fundamental construct of the FC SAN infrastructure. Fibre Channel provides a serial data transfer interface that operates over copper wire and optical fiber. FCP is the implementation of serial SCSI over an FC network. In FCP architecture, all external and remote storage devices attached to the SAN appear as local devices to the host operating system. The key advantages of FCP are as follows:

- Sustained transmission bandwidth over long distances.
- Support for a larger number of addressable devices over a network. Theoretically, FC can support more than 15 million device addresses on a network.
- Support speeds up to 16 Gbps (16 GFC).

Fibre Channel Protocol Stack

It is easier to understand a communication protocol by viewing it as a structure of independent layers. FCP defines the communication protocol in five layers: FC-0 through FC-4 (except FC-3 layer, which is not implemented). In a layered communication model, the peer layers on each node talk to each other through defined protocols. Figure 5-12 illustrates the Fibre Channel protocol stack.

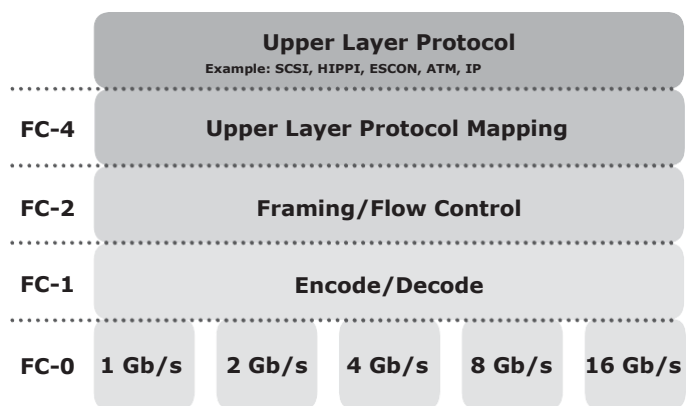


Figure 5-12: Fibre Channel protocol stack

FC-4 Layer

FC-4 is the uppermost layer in the FCP stack. This layer defines the application interfaces and the way *Upper Layer Protocols* (ULPs) are mapped to the lower FC layers. The FC standard defines several protocols that can operate on the FC-4 layer (see Figure 5-12). Some of the protocols include SCSI, High Performance Parallel Interface (HIPPI) Framing Protocol, Enterprise Storage Connectivity (ESCON), Asynchronous Transfer Mode (ATM), and IP.

FC-2 Layer

The FC-2 layer provides Fibre Channel addressing, structure, and organization of data (frames, sequences, and exchanges). It also defines fabric services, classes of service, flow control, and routing.

FC-1 Layer

The FC-1 layer defines how data is encoded prior to transmission and decoded upon receipt. At the transmitter node, an 8-bit character is encoded into a 10-bit transmissions character. This character is then transmitted to the receiver node. At the receiver node, the 10-bit character is passed to the FC-1 layer, which decodes the 10-bit character into the original 8-bit character. FC links with speeds of 10 Gbps and above use 64-bit to 66-bit encoding algorithms. The FC-1 layer also defines the transmission words, such as FC frame delimiters, which identify the start and end of a frame and primitive signals that indicate events at a transmitting port. In addition to these, the FC-1 layer performs link initialization and error recovery.

FC-0 Layer

FC-0 is the lowest layer in the FCP stack. This layer defines the physical interface, media, and transmission of bits. The FC-0 specification includes cables, connectors, and optical and electrical parameters for a variety of data rates. The FC transmission can use both electrical and optical media.

Mainframe SANs use *Fibre Connectivity* (FICON) for a low-latency, high-bandwidth connection to the storage controller. FICON was designed as a replacement for *Enterprise System Connection* (ESCON) to support mainframe-attached storage systems.

Fibre Channel Addressing

An FC address is dynamically assigned when a node port logs on to the fabric. The FC address has a distinct format, as shown in Figure 5-13. The addressing mechanism provided here corresponds to the fabric with the switch as an interconnecting device.

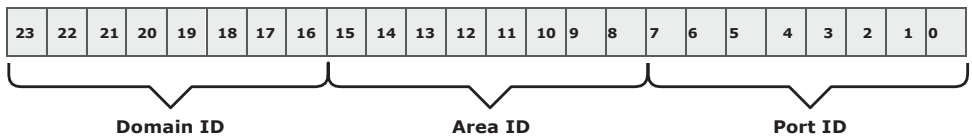


Figure 5-13: 24-bit FC address of N_Port

The first field of the FC address contains the domain ID of the switch. A *domain ID* is a unique number provided to each switch in the fabric. Although this is an 8-bit field, there are only 239 available addresses for domain ID because some addresses are deemed special and reserved for fabric management services. For example, FFFFFC is reserved for the name server, and FFFFFE is reserved for the fabric login service. The *area ID* is used to identify a group of switch ports used for connecting nodes. An example of a group of ports with a common area ID is a port card on the switch. The last field, the *port ID*, identifies the port within the group.

Therefore, the maximum possible number of node ports in a switched fabric is calculated as:

239 domains ∞ 256 areas ∞ 256 ports = 15,663,104

World Wide Names

Each device in the FC environment is assigned a 64-bit unique identifier called the *World Wide Name* (WWN). The Fibre Channel environment uses two types of WWNs: *World Wide Node Name* (WWNN) and *World Wide Port Name* (WWPN). Unlike an FC address, which is assigned dynamically, a WWN is a static name

for each node on an FC network. WWNs are similar to the Media Access Control (MAC) addresses used in IP networking. WWNs are *burned* into the hardware or assigned through software. Several configuration definitions in a SAN use WWN for identifying storage devices and HBAs. The name server in an FC environment keeps the association of WWNs to the dynamically created FC addresses for nodes. Figure 5-14 illustrates the WWN structure examples for an array and an HBA.

World Wide Name - Array															
5	0	0	6	0	1	6	0	0	0	6	0	0	1	B	2
0101	0000	0000	0110	0000	0001	0110	0000	0000	0000	0110	0000	0000	0001	1011	0010
Format Type	Company ID 24 bits						Port	Model Seed 32 bits							

World Wide Name - HBA															
1	0	0	0	0	0	0	0	c	9	2	0	d	c	4	0
Format Type	Reserved 12 bits			Company ID 24 bits						Company Specific 24 bits					

Figure 5-14: World Wide Names

5.1.2 FC Frame

An FC frame (Figure 5-15) consists of five parts: *start of frame* (SOF), *frame header*, *data field*, *cyclic redundancy check* (CRC), and *end of frame* (EOF).

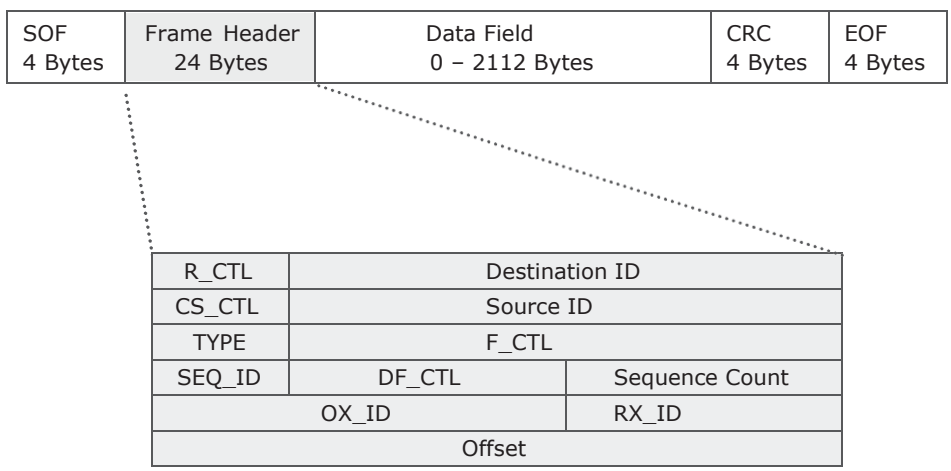


Figure 5-15: FC frame

The SOF and EOF act as delimiters. In addition to this role, the SOF also indicates whether the frame is the first frame in a sequence of frames.

The frame header is 24 bytes long and contains addressing information for the frame. It includes the following information: Source ID (S_ID), Destination ID (D_ID), Sequence ID (SEQ_ID), Sequence Count (SEQ_CNT), Originating Exchange ID (OX_ID), and Responder Exchange ID (RX_ID), in addition to some control fields.

The S_ID and D_ID are FC addresses for the source port and the destination port, respectively. The SEQ_ID and OX_ID identify the frame as a component of a specific sequence and exchange, respectively.

The frame header also defines the following fields:

- **Routing Control (R_CTL):** This field denotes whether the frame is a link control frame or a data frame. Link control frames are frames that do not carry any user data. These frames are used for setup and messaging. In contrast, data frames carry the user data.
- **Class Specific Control (CS_CTL):** This field specifies link speeds for class 1 and class 4 data transmission. (Class of service is discussed in section 5.6.7 "Classes of Service" later in the chapter.)
- **TYPE:** This field describes the upper layer protocol (ULP) to be carried on the frame if it is a data frame. However, if it is a link control frame, this field is used to signal an event such as "fabric busy." For example, if the TYPE is 08, and the frame is a data frame, it means that the SCSI will be carried on an FC.
- **Data Field Control (DF_CTL):** A 1-byte field that indicates the existence of any optional headers at the beginning of the data payload. It is a mechanism to extend header information into the payload.
- **Frame Control (F_CTL):** A 3-byte field that contains control information related to frame content. For example, one of the bits in this field indicates whether this is the first sequence of the exchange.

The data field in an FC frame contains the data payload, up to 2,112 bytes of actual data with 36 bytes of fixed overhead.

The CRC checksum facilitates error detection for the content of the frame. This checksum verifies data integrity by checking whether the content of the frames are received correctly. The CRC checksum is calculated by the sender before encoding at the FC-1 layer. Similarly, it is calculated by the receiver after decoding at the FC-1 layer.

Structure and Organization of FC Data

In an FC network, data transport is analogous to a conversation between two people, whereby a frame represents a word, a sequence represents a sentence, and an exchange represents a conversation.

- **Exchange:** An exchange operation enables two node ports to identify and manage a set of information units. Each upper layer protocol has its protocol-specific information that must be sent to another port to perform certain operations. This protocol-specific information is called an information unit. The structure of these information units is defined in the FC-4 layer. This unit maps to a sequence. An exchange is composed of one or more sequences.
- **Sequence:** A sequence refers to a contiguous set of frames that are sent from one port to another. A sequence corresponds to an information unit, as defined by the ULP.
- **Frame:** A frame is the fundamental unit of data transfer at Layer 2. Each frame can contain up to 2,112 bytes of payload.

Flow Control

Flow control defines the pace of the flow of data frames during data transmission. FC technology uses two flow-control mechanisms: buffer-to-buffer credit (BB_Credit) and end-to-end credit (EE_Credit).

BB_Credit

FC uses the *BB_Credit* mechanism for flow control. BB_Credit controls the maximum number of frames that can be present over the link at any given point in time. In a switched fabric, BB_Credit management may take place between any two FC ports. The transmitting port maintains a count of free receiver buffers and continues to send frames if the count is greater than 0. The BB_Credit mechanism uses *Receiver Ready* (R_RDY) primitive that indicates a buffer has been freed on the port that transmitted the R_RDY.

EE_Credit

The function of end-to-end credit, known as *EE_Credit*, is similar to that of BB_Credit. When an initiator and a target establish themselves as nodes communicating with each other, they exchange the EE_Credit parameters (part of Port login). The EE_Credit mechanism provides the flow control for class 1 and class 2 traffic only.

Classes of Service

The FC standards define different classes of service to meet the requirements of a wide range of applications. Table 5-1 shows three classes of services and their features.

Table 5-1: FC Class of Services

	CLASS 1	CLASS 2	CLASS 3
Communication type	Dedicated connection	Nondedicated connection	Nondedicated connection
Flow control	End-to-end credit	End-to-end credit B-to-B credit	B-to-B credit
Frame delivery	In order delivery	Order not guaranteed	Order not guaranteed
Frame acknowledgment	Acknowledged	Acknowledged	Not acknowledged
Multiplexing	No	Yes	Yes
Bandwidth utilization	Poor	Moderate	High

Another class of service is *class F*, which is used for fabric management. Class F is similar to Class 2 and provides notification of nondelivery of frames.

Fabric Services

All FC switches, regardless of the manufacturer, provide a common set of services as defined in the Fibre Channel standards. These services are available at certain predefined addresses. Some of these services are Fabric Login Server, Fabric Controller, Name Server, and Management Server (see Figure 5-16).

The *Fabric Login Server* is located at the predefined address of FFFFFE and is used during the initial part of the node's fabric login process.

The *Name Server* (formally known as *Distributed Name Server*) is located at the predefined address FFFFFC and is responsible for name registration and management of node ports. Each switch exchanges its Name Server information with other switches in the fabric to maintain a synchronized, distributed name service.

Each switch has a *Fabric Controller* located at the predefined address FFFFFD. The Fabric Controller provides services to both node ports and other switches. The Fabric Controller is responsible for managing and distributing Registered State Change Notifications (RSCNs) to the node ports registered with the

Fabric Controller. If there is a change in the fabric, RSCNs are sent out by a switch to the attached node ports. The Fabric Controller also generates Switch Registered State Change Notifications (SW-RSCNs) to every other domain (switch) in the fabric. These RSCNs keep the name server up-to-date on all switches in the fabric.

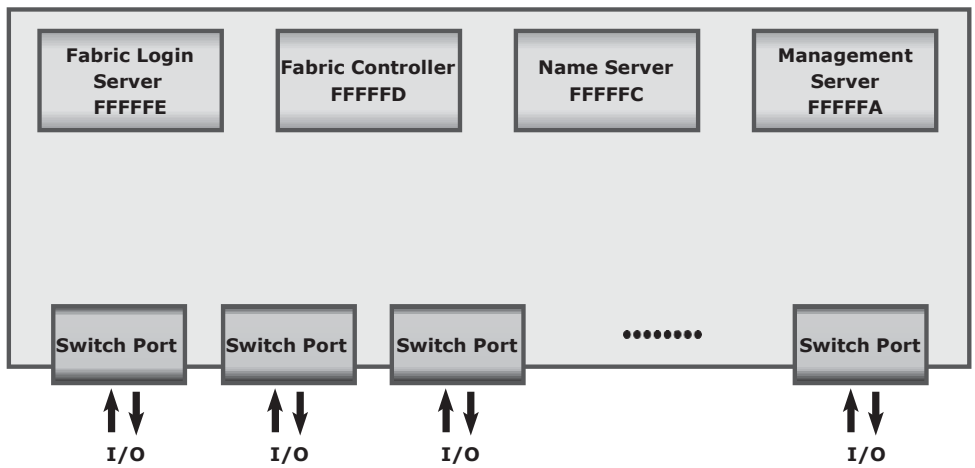


Figure 5-16: Fabric services provided by FC switches

FFFFFA is the Fibre Channel address for the *Management Server*. The Management Server is distributed to every switch within the fabric. The Management Server enables the FC SAN management software to retrieve information and administer the fabric.

Switched Fabric Login Types

Fabric services define three login types:

- ***Fabric login* (FLOGI):** Performed between an N_Port and an F_Port. To log on to the fabric, a node sends a FLOGI frame with the WWNN and WWPN parameters to the login service at the predefined FC address FFFFFE (Fabric Login Server). In turn, the switch accepts the login and returns an Accept (ACC) frame with the assigned FC address for the node. Immediately after the FLOGI, the N_Port registers itself with the local Name Server on the switch, indicating its WWNN, WWPN, port type, class of service, assigned FC address and so on. After the N_Port has logged in, it can query the name server database for information about all other logged in ports.

- ***Port login* (PLOGI):** Performed between two N_Ports to establish a session. The initiator N_Port sends a PLOGI request frame to the target N_Port, which accepts it. The target N_Port returns an ACC to the initiator N_Port. Next, the N_Ports exchange service parameters relevant to the session.
- ***Process login* (PRLI):** Also performed between two N_Ports. This login relates to the FC-4 ULPs, such as SCSI. If the ULP is SCSI, N_Ports exchange SCSI-related service parameters.

Zoning

Zoning is an FC switch function that enables node ports within the fabric to be logically segmented into groups and to communicate with each other within the group (see Figure 5-17).

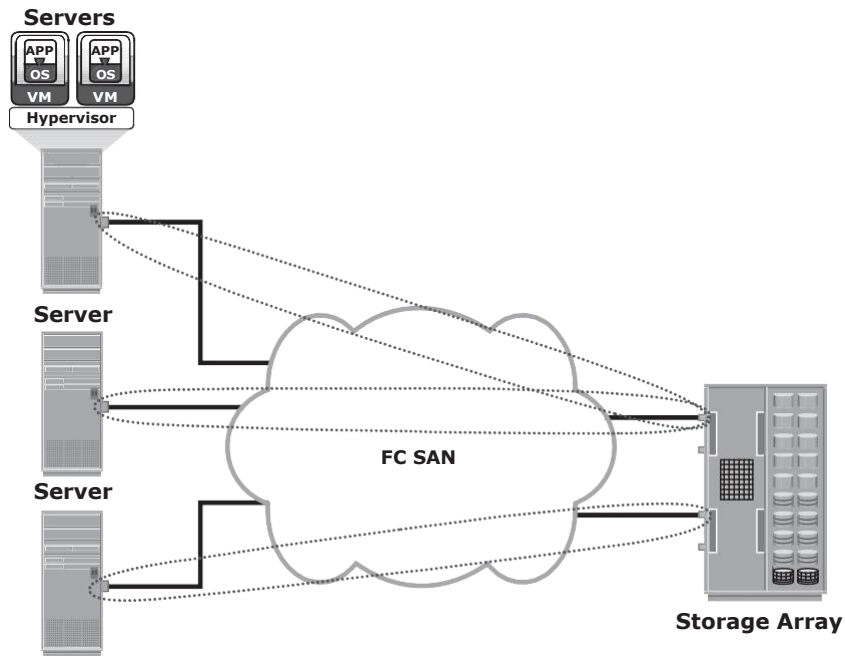


Figure 5-17: Zoning

Whenever a change takes place in the name server database, the fabric controller sends a Registered State Change Notification (RSCN) to all the nodes impacted by the change. If zoning is not configured, the fabric controller sends an RSCN to all the nodes in the fabric. Involving the nodes that are not impacted by the change results in increased fabric-management traffic. For

a large fabric, the amount of FC traffic generated due to this process can be significant and might impact the host-to-storage data traffic. Zoning helps to limit the number of RSCNs in a fabric. In the presence of zoning, a fabric sends the RSCN to only those nodes in a zone where the change has occurred. Zone members, zones, and zone sets form the hierarchy defined in the zoning process (see Figure 5-18). A *zone set* is composed of a group of zones that can be activated or deactivated as a single entity in a fabric. Multiple zone sets may be defined in a fabric, but only one zone set can be active at a time. *Members* are nodes within the SAN that can be included in a zone. Switch ports, HBA ports, and storage device ports can be members of a zone. A port or node can be a member of multiple zones. Nodes distributed across multiple switches in a switched fabric may also be grouped into the same zone. Zone sets are also referred to as *zone configurations*.

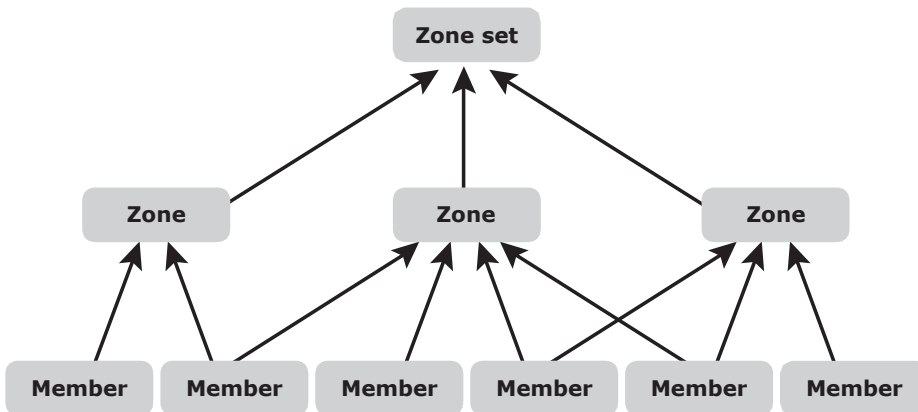


Figure 5-18: Members, zones, and zone sets

Zoning provides control by allowing only the members in the same zone to establish communication with each other.

Types of Zoning

Zoning can be categorized into three types:

- **Port zoning:** Uses the physical address of switch ports to define zones. In port zoning, access to node is determined by the physical switch port to which a node is connected. The zone members are the port identifier (switch domain ID and port number) to which HBA and its targets (storage devices) are connected. If a node is moved to another switch port in the

fabric, then zoning must be modified to allow the node, in its new port, to participate in its original zone. However, if an HBA or storage device port fails, an administrator just has to replace the failed device without changing the zoning configuration.

- **WWN zoning:** Uses World Wide Names to define zones. The zone members are the unique WWN addresses of the HBA and its targets (storage devices). A major advantage of WWN zoning is its flexibility. WWN zoning allows nodes to be moved to another switch port in the fabric and maintain connectivity to its zone partners without having to modify the zone configuration. This is possible because the WWN is static to the node port.
- **Mixed zoning:** Combines the qualities of both WWN zoning and port zoning. Using mixed zoning enables a specific node port to be tied to the WWN of another node.

Figure 5-19 shows the three types of zoning on an FC network.

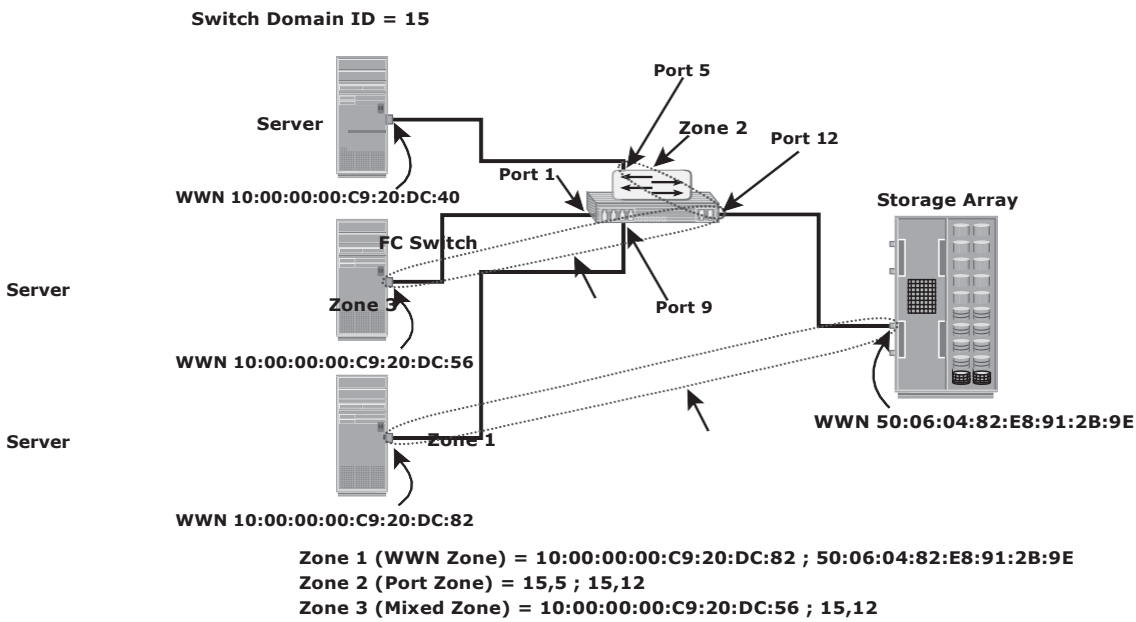


Figure 5-19: Types of zoning

Zoning is used with LUN masking to control server access to storage. However, these are two different activities. Zoning takes place at the fabric level and LUN masking is performed at the array level.

FC SAN Topologies

Fabric design follows standard topologies to connect devices. Core-edge fabric is one of the popular topologies for fabric designs. Variations of core-edge fabric and mesh topologies are most commonly deployed in FC SAN implementations.

Mesh Topology

A mesh topology may be one of the two types: full mesh or partial mesh. In a *full mesh*, every switch is connected to every other switch in the topology. A full mesh topology may be appropriate when the number of switches involved is small. A typical deployment would involve up to four switches or directors, with each of them servicing highly localized host-to-storage traffic. In a full mesh topology, a maximum of one ISL or hop is required for host-to-storage traffic. However, with the increase in the number of switches, the number of switch ports used for ISL also increases. This reduces the available switch ports for node connectivity.

In a *partial mesh topology*, several hops or ISLs may be required for the traffic to reach its destination. Partial mesh offers more scalability than full mesh topology. However, without proper placement of host and storage devices, traffic management in a partial mesh fabric might be complicated and ISLs could become overloaded due to excessive traffic aggregation. Figure 5-20 depicts both partial mesh and full mesh topologies.

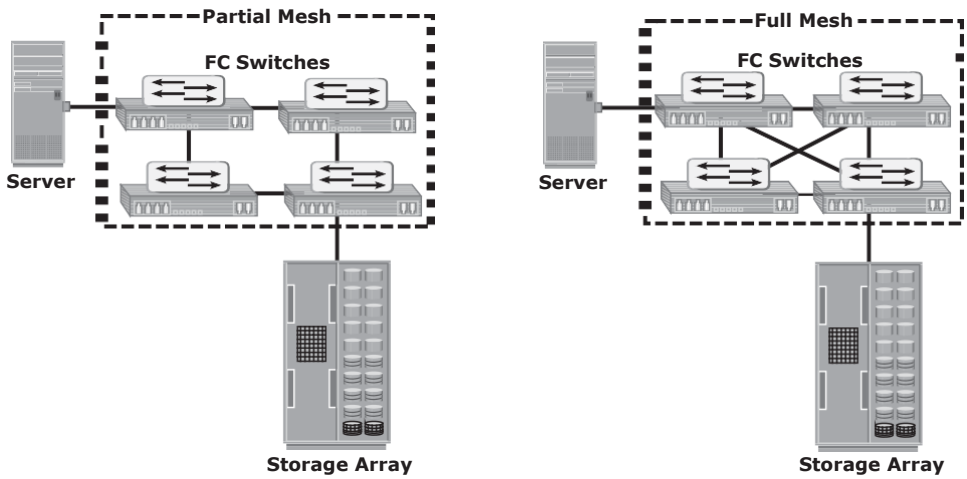


Figure 5-20: Partial mesh and full mesh topologies

Core-Edge Fabric

The *core-edge fabric* topology has two types of switch tiers. The *edge tier* is usually composed of switches and offers an inexpensive approach to adding more hosts in a fabric. Each switch at the edge tier is attached to a switch at the core tier through ISLs.

The *core tier* is usually composed of enterprise directors that ensure high fabric availability. In addition, typically all traffic must either traverse this tier or terminate at this tier. In this configuration, all storage devices are connected to the core tier, enabling host-to-storage traffic to traverse only one ISL. Hosts that require high performance may be connected directly to the core tier and consequently avoid ISL delays.

In *core-edge topology*, the edge-tier switches are not connected to each other. The core-edge fabric topology increases connectivity within the SAN while conserving the overall port utilization. If fabric expansion is required, additional edge switches are connected to the core. The core of the fabric is also extended by adding more switches or directors at the core tier. Based on the number of core-tier switches, this topology has different variations, such as, *single-core topology* (see Figure 5-21) and *dual-core topology* (see Figure 5-22). To transform a single-core topology to dual-core, new ISLs are created to connect each edge switch to the new core switch in the fabric.

Benefits and Limitations of Core-Edge Fabric

The core-edge fabric provides maximum one-hop storage access to all storage devices in the system. Because traffic travels in a deterministic pattern (from the edge to the core and vice versa), a core-edge provides easier calculation of the ISL load and traffic patterns. In this topology, because each tier's switch port

is used for either storage or hosts, it's easy to identify which network resources are approaching their capacity, making it easier to develop a set of rules for scaling and apportioning.

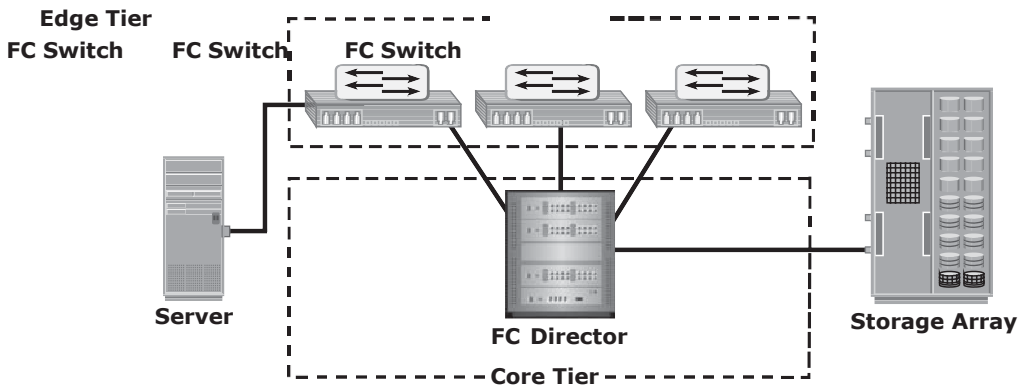


Figure 5-21: Single-core topology

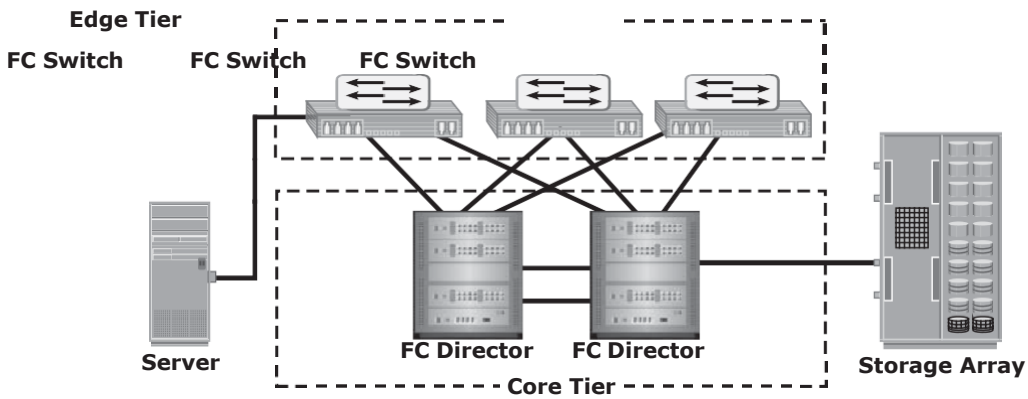


Figure 5-22: Dual-core topology

Core-edge fabrics are scaled to larger environments by adding more core switches and linking them, or adding more edge switches. This method enables extending the existing simple core-edge model or expanding the fabric into a compound or complex core-edge model.

However, the core-edge fabric might lead to some performance-related problems because scaling a core-edge topology involves increasing the number of hop counts in the fabric. *Hop count* represents the total number of ISLs traversed by a packet between its source and destination. A common best practice is to keep the number of host-to-storage hops unchanged, at one hop, in a core-edge. Generally, a large hop count means a high data transmission delay between the source and destination.

As the number of cores increases, it is prohibitive to continue to maintain ISLs from each core to each edge switch. When this happens, the fabric design is changed to a compound or complex core-edge design (see Figure 5-23).

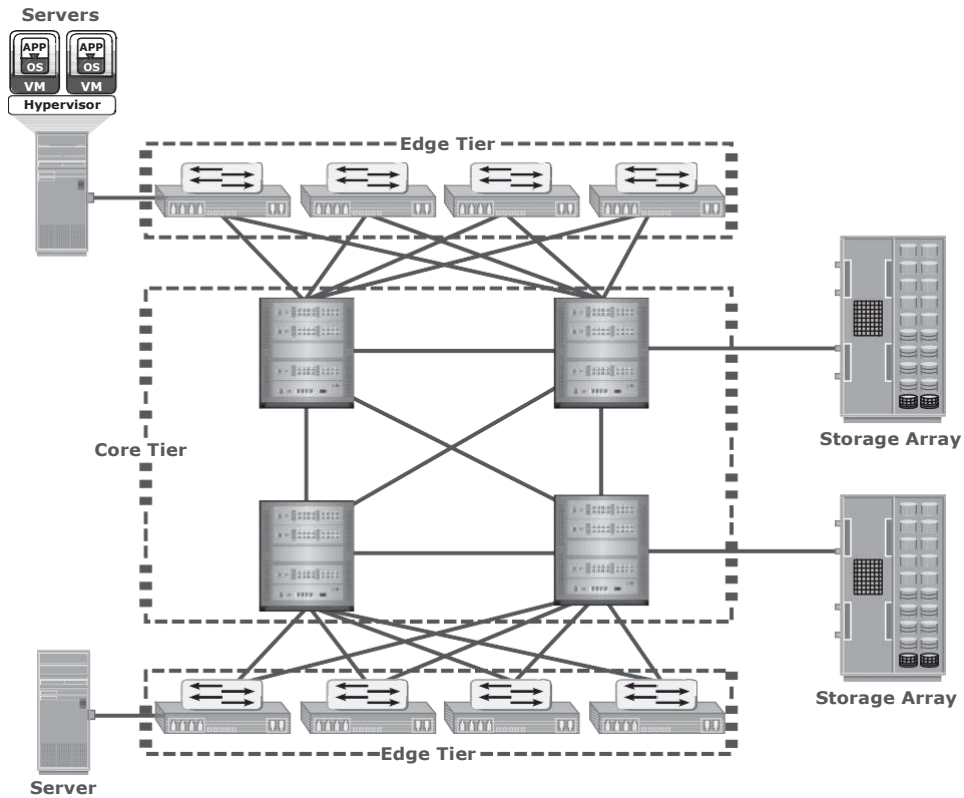


Figure 5-23: Compound core-edge topology

Virtualization in SAN

This section details two network-based virtualization techniques in a SAN environment: block-level storage virtualization and virtual SAN (VSAN).

Block-level Storage Virtualization

Block-level storage virtualization aggregates block storage devices (LUNs) and enables provisioning of virtual storage volumes, independent of the underlying physical storage. A virtualization layer, which exists at the SAN, abstracts the identity of physical storage devices and creates a storage pool from heterogeneous storage devices. Virtual volumes are created from the storage pool and assigned to the hosts. Instead of being directed to the LUNs on the individual storage arrays, the hosts are directed to the virtual volumes provided by the virtualization layer. For hosts and storage arrays, the virtualization layer appears as the target and initiator devices, respectively. The virtualization layer maps the virtual volumes to the LUNs on the individual arrays. The hosts remain unaware of the mapping operation and access the virtual volumes as if they were accessing the physical storage attached to them. Typically, the virtualization layer is managed via a dedicated virtualization appliance to which the hosts and the storage arrays are connected.

Figure 5-24 illustrates a virtualized environment. It shows two physical servers, each of which has one virtual volume assigned. These virtual volumes are used by the servers. These virtual volumes are mapped to the LUNs in the storage arrays. When an I/O is sent to a virtual volume, it is redirected through the virtualization layer at the storage network to the mapped LUNs. Depending on the capabilities of the virtualization appliance, the architecture may allow for more complex mapping between array LUNs and virtual volumes.

Block-level storage virtualization enables extending the storage volumes online to meet application growth requirements. It consolidates heterogeneous storage arrays and enables transparent volume access.

Block-level storage virtualization also provides the advantage of nondisruptive data migration. In a traditional SAN environment, LUN migration from one array to another is an offline event because the hosts needed to be updated to reflect the new array configuration. In other instances, host CPU cycles were required to migrate data from one array to the other, especially in a multivendor environment. With a block-level virtualization solution in place, the virtualization layer handles the back-end migration of data, which enables LUNs to remain online and accessible while data is migrating. No physical changes are required because the host still points to the same virtual targets on the virtualization layer. However, the mappings information on the virtualization

layer should be changed. These changes can be executed dynamically and are transparent to the end user.

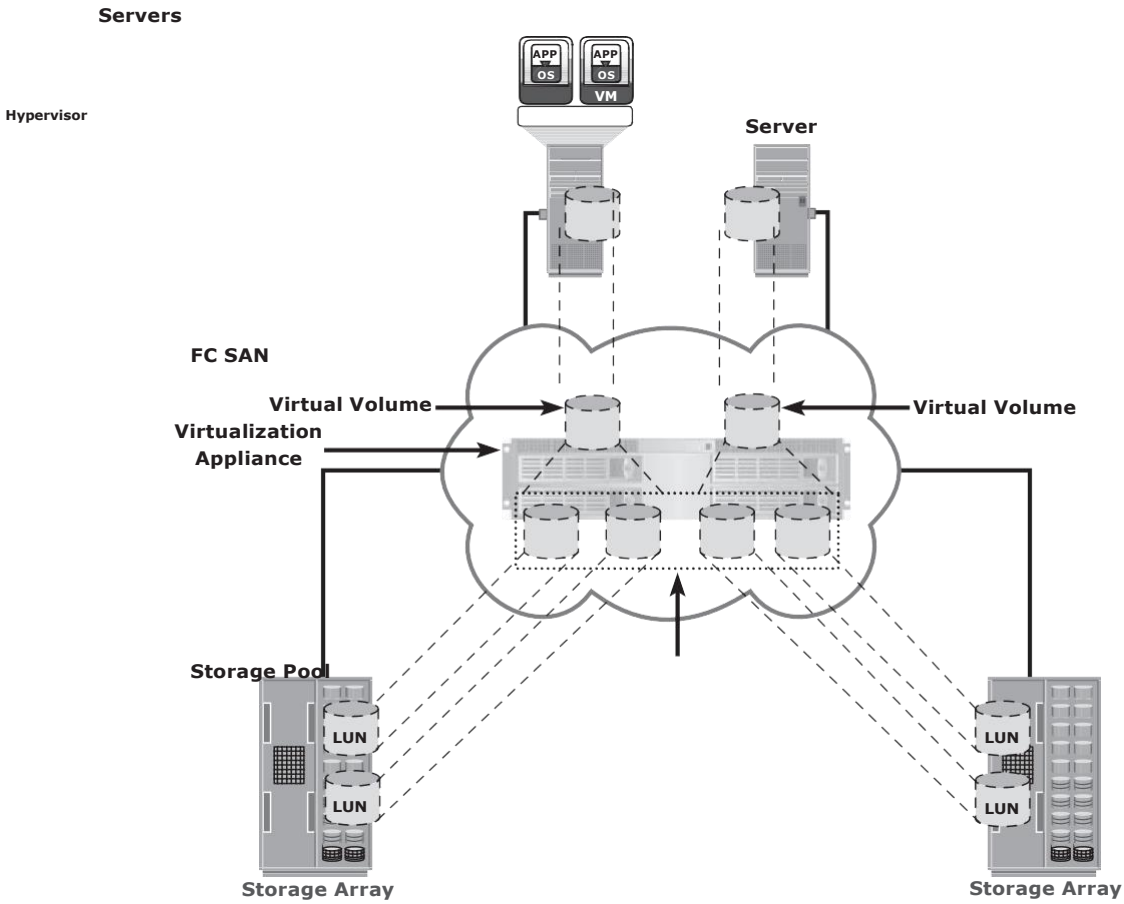


Figure 5-24: Block-level storage virtualization

Previously, block-level storage virtualization provided nondisruptive data migration only within a data center. The new generation of block-level storage virtualization enables nondisruptive data migration both within and between data centers. It provides the capability to connect the virtualization layers at multiple data centers. The connected virtualization layers are managed centrally and work as a single virtualization layer stretched across data centers (see Figure 5-25). This enables the federation of block-storage resources both within and across data centers. The virtual volumes are created from the federated storage resources.

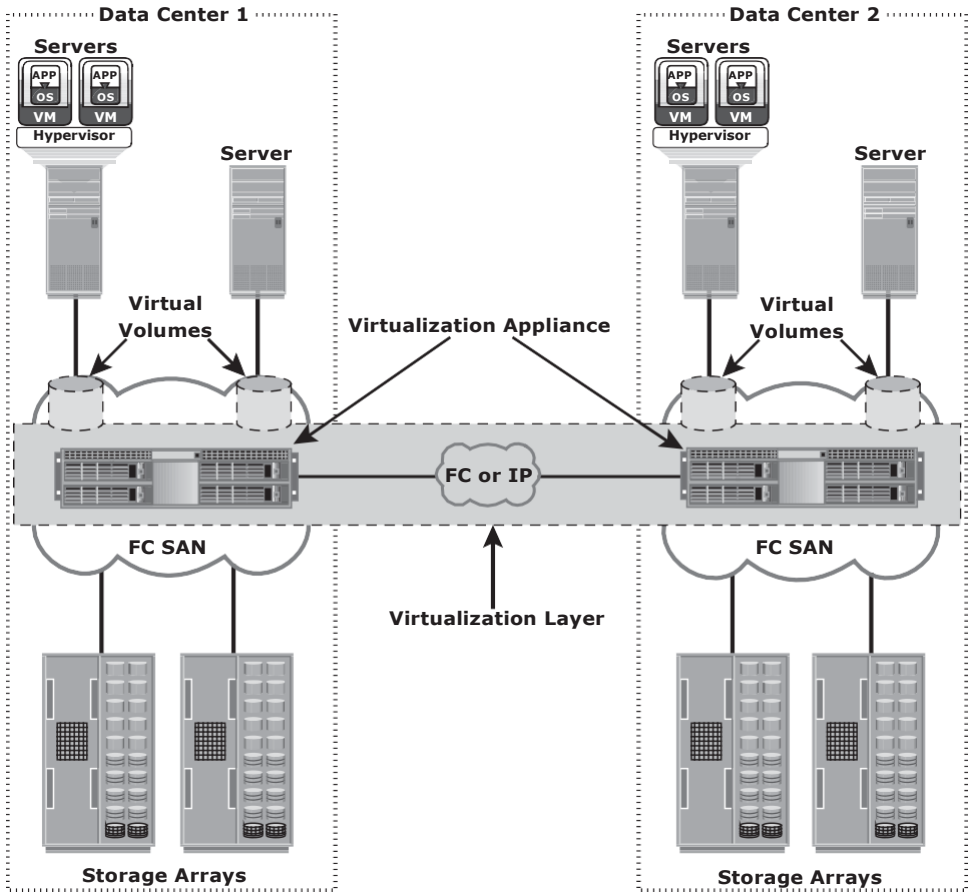


Figure 5-25: Federation of block storage across data centers

Virtual SAN (VSAN)

Virtual SAN (also called *virtual fabric*) is a logical fabric on an FC SAN, which enables communication among a group of nodes regardless of their physical location in the fabric. In a VSAN, a group of hosts or storage ports communicate with each other using a virtual topology defined on the physical SAN. Multiple VSANs may be created on a single physical SAN. Each VSAN acts as an independent fabric with its own set of fabric services, such as name server, and zoning. Fabric-related configurations in one VSAN do not affect the traffic in another. VSANs improve SAN security, scalability, availability, and manageability.
VSANs provide enhanced security by isolating the sensitive data in a VSAN and by restricting access to the resources located within that VSAN. The same Fibre Channel address can be assigned to nodes in different VSANs, thus increasing the fabric scalability. Events causing traffic disruptions in one VSAN are contained

within that VSAN and are not propagated to other VSANs. VSANs facilitate an easy, flexible, and less expensive way to manage networks. Configuring VSANs is easier and quicker compared to building separate physical FC SANs for various node groups. To regroup nodes, an administrator simply changes the VSAN configurations without moving nodes and recabling. VSAN is further discussed in Chapter 14.

Concepts in Practice: EMC Connectrix and EMC VPLEX

The EMC Connectrix family represents the industry's most extensive selection of networked storage connectivity products. Connectrix integrates high-speed Fibre Channel connectivity, highly resilient switching technology, options for intelligent IP storage networking, and I/O consolidation with products that support Fibre Channel over Ethernet.

EMC VPLEX is the next-generation solution for block-level virtualization and data mobility within, across, and between data centers. EMC VPLEX provides storage federation by aggregating storage arrays that can be located either in a single data center or multiple data centers. VPLEX is also used as the data mobility solution for environments like cloud computing.

For the latest information on Connectrix connectivity products and VPLEX, visit `www.emc.com`.

EMC Connectrix

EMC offers the following connectivity products under the Connectrix brand (see Figure 5-26):

- Enterprise directors
- Departmental switches
- Multi-purpose switches

Enterprise directors are ideal for large enterprise connectivity. They offer high port density and high component redundancy. Enterprise directors are deployed in high-availability or large-scale environments. Connectrix directors offer several hundred ports per domain. Departmental switches are best suited for workgroup, mid-tier environments. Multi-purpose switches support various protocols such as iSCSI, FCIP, FCoE, FICON, in addition to FC protocol. In addition to FC ports, Connectrix switches and directors have Ethernet ports and serial ports for communication and switch management functions. The Connectrix management software enables configuration, monitoring, and management of Connectrix switches.

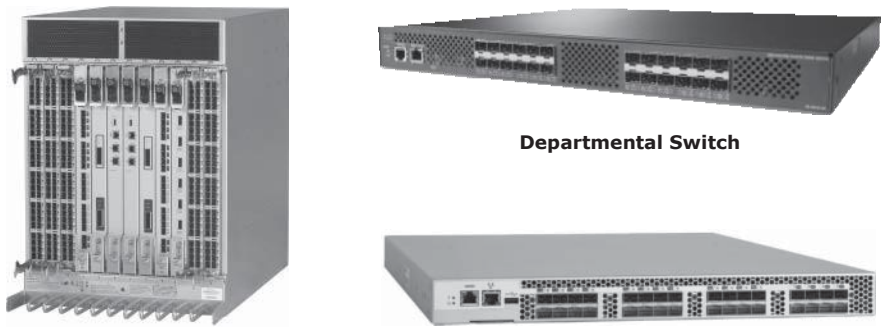


Figure 5-26: EMC Connectrix

Connectrix Switches

B-series and MDS-series make up the Connectrix family of switches offered by EMC. These switches are designed to meet workgroup, department-level, and enterprise-level requirements. They are designed with a nonblocking architecture and can operate in heterogeneous environments. Nonblocking architecture refers to the capability of a switch to handle independent packets simultaneously because the switch has sufficient internal resources to handle maximum transfer rates from all ports. The features of these switches that ensure their high availability are their nondisruptive software and port upgrade, and redundant and hot-swappable components. These switches can be managed through CLI, HTTP, and standalone GUI applications.

Connectrix Directors

EMC offers the high-end Connectrix family of directors. Their modular architectural design offers high scalability by providing over 500 ports. They are suitable for server and storage consolidation across enterprises. These directors have redundant components for high availability and provide multiprotocol connectivity for both mainframe and open system environments. Connectrix directors offer high speeds (up to 16 Gb/s) and support ISL aggregation. Similar to switches, directors can also be managed through CLI or with other GUI tools.

Connectrix Multi-purpose Switches

Multi-purpose switches provide support for multiple protocols, such as FC, FCIP, iSCSI, FCoE, and FICON. They perform protocol translation and route frames between two dissimilar networks, such as FC and IP. These multiprotocol

capabilities offer many benefits, including long-distance SAN extension, greater resource sharing, and simplified management. Connectrix multi-purpose switches include FCoE switches, FCIP routers, iSCSI gateways, and so on.

Connectrix Management Tools

There are several ways to monitor and manage FC switches in a fabric. Individual switch management is accomplished through the CLI or browser-based tools. Command-line utilities such as Telnet and Secure Shell (SSH) are used to log on to the switch over IP and issue CLI commands. The primary purpose of the CLI is to automate the management of a large number of switches or directors with the use of scripts. The browser-based tools provide GUIs. These tools also display the topology map.

Fabric-wide management and monitoring is accomplished by using vendor-specific tools and Simple Network Management Protocol (SNMP)-based, third-party software.

EMC ControlCenter SAN Manager provides a single interface for managing a Storage Area Network. With SAN Manager, an administrator can discover, monitor, manage, and configure complex heterogeneous SAN environments. It streamlines and centralizes SAN management operations across multivendor storage networks and storage devices. It enables storage administrators to manage SAN zones and LUN masking consistently across multivendor SAN arrays and switches. EMC ControlCenter SAN Manager also supports virtual environments, including VMware, and virtual SANs.

EMC ProSphere is a newly launched tool with additional features specifically for the cloud computing environment. A future release of EMC ProSphere will include all the functionalities of EMC ControlCenter.

EMC VPLEX

EMC VPLEX provides a virtual storage infrastructure that enables federation of heterogeneous storage resources both within and across datacenters. The VPLEX appliance resides between the servers and heterogeneous storage devices. It forms a pool of distributed block storage resources and enables creating virtual storage volumes from the pool. These virtual volumes are then allocated to the servers. The virtual-to-physical-storage mapping remains hidden to the servers. VPLEX provides nondisruptive data mobility among physical storage devices to balance the application workload and to enable both local and remote data access. The mapping of virtual volumes to physical volumes can be changed dynamically by the administrator. This allows for a virtual volume to be moved across storage arrays while still in production.

VPLEX uses a unique clustering architecture and distributed cache coherency that enable multiple hosts located across two locations to access a single copy of data. This eliminates the operational overhead and time required to copy and distribute data across locations VPLEX also provides the capability to mirror data of a virtual volume both within and across locations. This enables hosts at different data centers to access cache-coherent copies of the same virtual volume. Practical applications of this capability include mobility, load-balancing, and high availability across data centers.

To avoid application downtime due to outage at a data center, the workload can be moved quickly to another data center. Applications continue accessing the same virtual volume and remain uninterrupted by the data mobility.

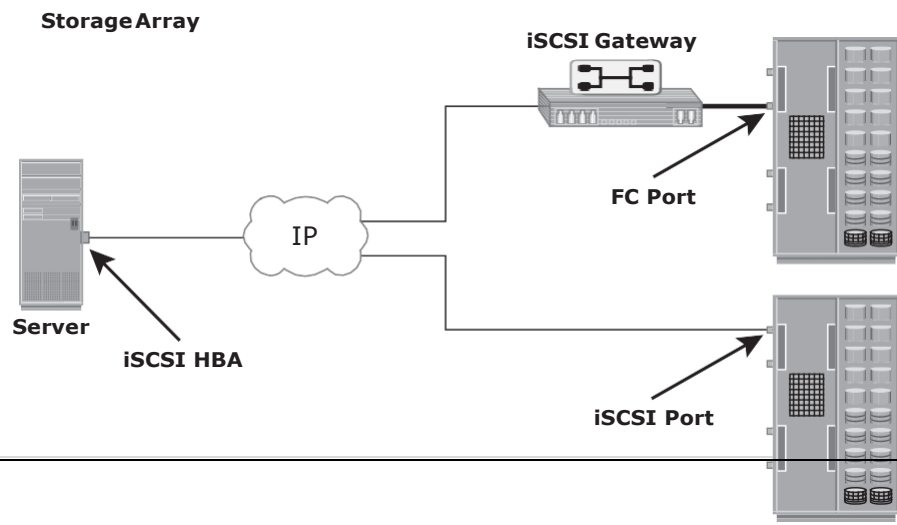
VPLEX Family of Products

The VPLEX family consists of three products: VPLEX Local, VPLEX Metro, and VPLEX Geo.

EMC VPLEX Local delivers local federation, which provides simplified management and nondisruptive data mobility across heterogeneous arrays within a data center. EMC VPLEX Metro delivers distributed federation, which provides data access and mobility between two VPEX clusters within synchronous distances that support round-trip latency up to 5 ms. EMC VPLEX Geo delivers data access and mobility between two VPLEX clusters within asynchronous distances (that support round-trip latency up to 50 ms).

iSCSI

iSCSI is an IP based protocol that establishes and manages connections between host and storage over IP, as shown in Figure 6-1. iSCSI encapsulates SCSI commands and data into an IP packet and transports them using TCP/IP. iSCSI is widely adopted for connecting servers to storage because it is relatively inexpensive and easy to implement, especially in environments in which an FC SAN does not exist.



Storage Array

Figure 6-1: iSCSI implementation

Components of iSCSI

An initiator (host), target (storage or iSCSI gateway), and an IP-based network are the key iSCSI components. If an iSCSI-capable storage array is deployed, then a host with the iSCSI initiator can directly communicate with the storage array over an IP network. However, in an implementation that uses an existing FC array for iSCSI communication, an iSCSI gateway is used. These devices perform

the translation of IP packets to FC frames and vice versa, thereby bridging the connectivity between the IP and FC environments.

iSCSI Host Connectivity

A standard NIC with software iSCSI initiator, a TCP offload engine (TOE) NIC with software iSCSI initiator, and an iSCSI HBA are the three iSCSI host connectivity options. The function of the iSCSI initiator is to route the SCSI commands over an IP network.

A standard NIC with a software iSCSI initiator is the simplest and least expensive connectivity option. It is easy to implement because most servers come with at least one, and in many cases two, embedded NICs. It requires only a software initiator for iSCSI functionality. Because NICs provide standard IP function, encapsulation of SCSI into IP packets and decapsulation are carried out by the host CPU. This places additional overhead on the host CPU. If a standard NIC is used in heavy I/O load situations, the host CPU might become a bottleneck. TOE NIC helps alleviate this burden. A TOE NIC offloads TCP management functions from the host and leaves only the iSCSI functionality to the host processor. The host passes the iSCSI information to the TOE card, and the TOE card sends the information to the destination using TCP/IP. Although this solution improves performance, the iSCSI functionality is still handled by a software initiator that requires host CPU cycles. An iSCSI HBA is capable of providing performance benefits because it offloads the entire iSCSI and TCP/IP processing from the host processor. The use of an iSCSI HBA is also the simplest way to boot hosts from a SAN environment via iSCSI. If there is no iSCSI HBA, modifications must be made to the basic operating system to boot a host from the storage devices because the NIC needs to obtain an IP address before the operating system loads. The functionality of an iSCSI HBA is similar to the functionality of an FC HBA.

iSCSI Topologies

Two topologies of iSCSI implementations are native and bridged. *Native topology* does not have FC components. The initiators may be either directly attached to targets or connected through the IP network. *Bridged topology* enables the coexistence of FC with IP by providing iSCSI-to-FC bridging functionality. For example, the initiators can exist in an IP environment while the storage remains in an FC environment.

Native iSCSI Connectivity

FC components are not required for iSCSI connectivity if an iSCSI-enabled array is deployed. In Figure 6-2 (a), the array has one or more iSCSI ports configured with an IP address and is connected to a standard Ethernet switch.

After an initiator is logged on to the network, it can access the available LUNs on the storage array. A single array port can service multiple hosts or initiators as long as the array port can handle the amount of storage traffic that the hosts generate.

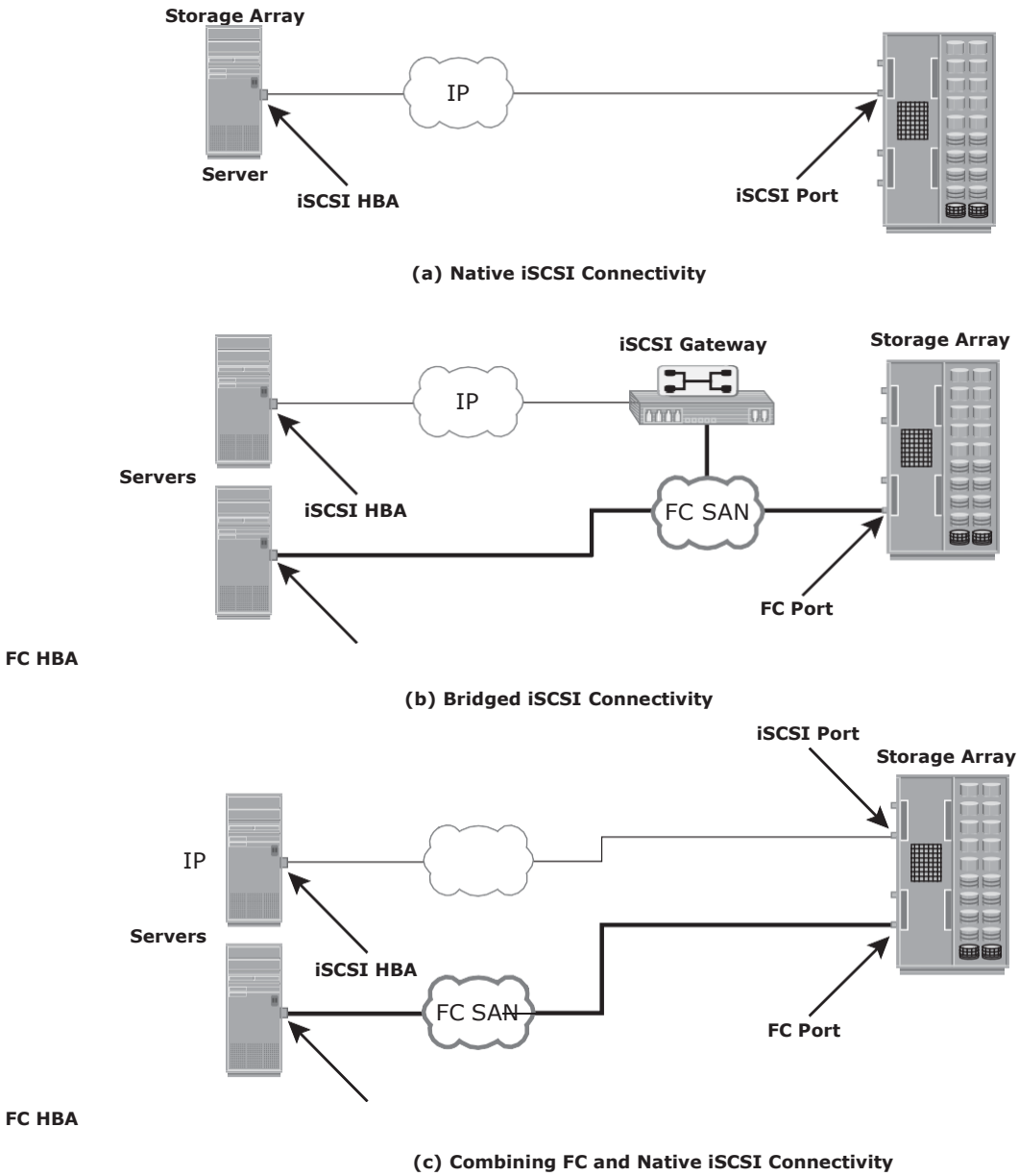


Figure 6-2: iSCSI Topologies

Bridged iSCSI Connectivity

A bridged iSCSI implementation includes FC components in its configuration. Figure 6-2 (b) illustrates iSCSI host connectivity to an FC storage array.

In this case, the array does not have any iSCSI ports. Therefore, an external device, called a gateway or a multiprotocol router, must be used to facilitate the communication between the iSCSI host and FC storage. The gateway converts IP packets to FC frames and vice versa. The bridge devices contain both FC and Ethernet ports to facilitate the communication between the FC and IP environments.

In a bridged iSCSI implementation, the iSCSI initiator is configured with the gateway's IP address as its target destination. On the other side, the gateway is configured as an FC initiator to the storage array.

Combining FC and Native iSCSI Connectivity

The most common topology is a combination of FC and native iSCSI. Typically, a storage array comes with both FC and iSCSI ports that enable iSCSI and FC connectivity in the same environment, as shown in Figure 6-2 (c).

iSCSI Protocol Stack

Figure 6-3 displays a model of the iSCSI protocol layers and depicts the encapsulation order of the SCSI commands for their delivery through a physical carrier.

SCSI is the command protocol that works at the application layer of the Open System Interconnection (OSI) model. The initiators and targets use SCSI commands and responses to talk to each other. The SCSI command descriptor blocks, data, and status messages are encapsulated into TCP/IP and transmitted across the network between the initiators and targets.

iSCSI is the session-layer protocol that initiates a reliable session between devices that recognize SCSI commands and TCP/IP. The iSCSI session-layer interface is responsible for handling login, authentication, target discovery, and session management. TCP is used with iSCSI at the transport layer to provide reliable transmission.

TCP controls message flow, windowing, error recovery, and retransmission. It relies upon the network layer of the OSI model to provide global addressing and connectivity. The Layer 2 protocols at the data link layer of this model enable node-to-node communication through a physical network.

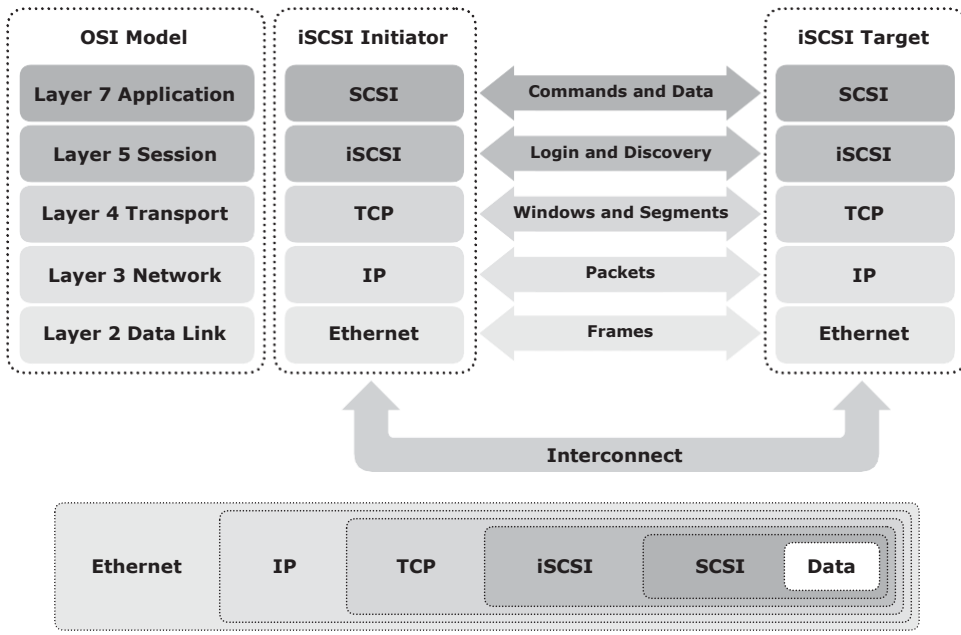


Figure 6-3: iSCSI protocol stack

iSCSI PDU

A *protocol data unit* (PDU) is the basic "information unit" in the iSCSI environment. The iSCSI initiators and targets communicate with each other using iSCSI PDUs. This communication includes establishing iSCSI connections and iSCSI sessions, performing iSCSI discovery, sending SCSI commands and data, and receiving SCSI status. All iSCSI PDUs contain one or more header segments followed by zero or more data segments. The PDU is then encapsulated into an IP packet to facilitate the transport.

A PDU includes the components shown in Figure 6-4. The IP header provides packet-routing information to move the packet across a network. The TCP header contains the information required to guarantee the packet delivery to the target. The iSCSI header (basic header segment) describes how to extract SCSI commands and data for the target. iSCSI adds an optional CRC, known as the *digest*, to ensure datagram integrity. This is in addition to TCP checksum and Ethernet CRC. The header and the data digests are optionally used in the PDU to validate integrity and data placement.

As shown in Figure 6-5, each iSCSI PDU does not correspond in a 1:1 relationship with an IP packet. Depending on its size, an iSCSI PDU can span an IP packet or even coexist with another PDU in the same packet.

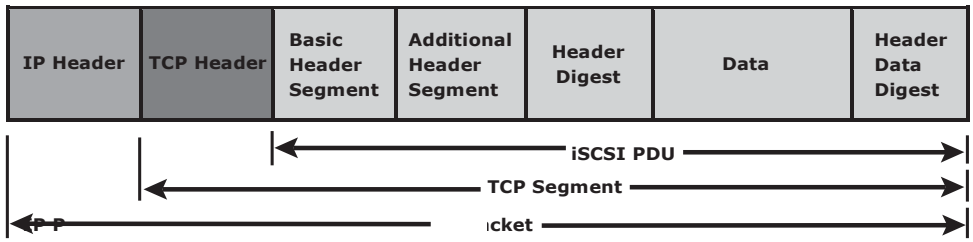


Figure 6-4: iSCSI PDU encapsulated in an IP packet

A message transmitted on a network is divided into a number of packets. If necessary, each packet can be sent by a different route across the network. Packets can arrive in a different order than the order in which they were sent. IP only delivers them; it is up to TCP to organize them in the right sequence. The target extracts the SCSI commands and data on the basis of the information in the iSCSI header.

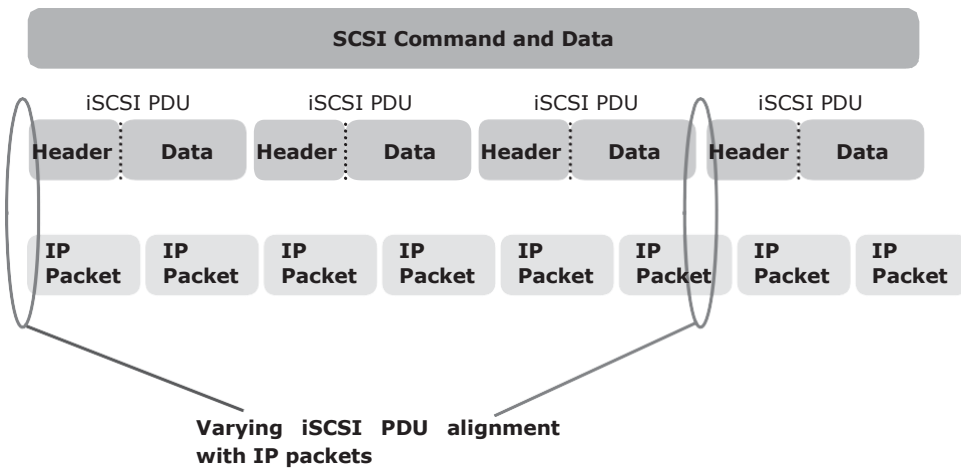


Figure 6-5: Alignment of iSCSI PDUs with IP packets

To achieve the 1:1 relationship between the IP packet and the iSCSI PDU, the maximum transmission unit (MTU) size of the IP packet is modified. This eliminates fragmentation of the IP packet, which improves the trans- mission efficiency.

iSCSI Discovery

An initiator must discover the location of its targets on the network and the names of the targets available to it before it can establish a session. This discov- ery can take place in two ways: *SendTargets discovery* or *internet Storage Name Service* (iSNS).

In *SendTargets discovery*, the initiator is manually configured with the target's network portal to establish a discovery session. The initiator issues the `SendTargets` command, and the target network portal responds with the names and addresses of the targets available to the host.

iSNS (see Figure 6-6) enables automatic discovery of iSCSI devices on an IP network. The initiators and targets can be configured to automatically register themselves with the iSNS server. Whenever an initiator wants to know the targets that it can access, it can query the iSNS server for a list of available targets.

The discovery can also take place by using service location protocol (SLP). However, this is less commonly used than `SendTargets` discovery and iSNS.

iSCSI Names

A unique worldwide iSCSI identifier, known as an *iSCSI name*, is used to identify the initiators and targets within an iSCSI network to facilitate communication. The unique identifier can be a combination of the names of the department, application, or manufacturer, serial number, asset number, or any tag that can be used to recognize and manage the devices. Following are two types of iSCSI names commonly used:

- **iSCSI Qualified Name (IQN):** An organization must own a registered domain name to generate iSCSI Qualified Names. This domain name does not need to be active or resolve to an address. It just needs to be reserved to prevent other organizations from using the same domain name to generate iSCSI names. A date is included in the name to avoid potential conflicts caused by the transfer of domain names. An example of an IQN is `iqn.2008-02.com.example:`*`optional_string`*.

  The *optional_string* provides a serial number, an asset number, or any other device identifiers. An iSCSI Qualified Name enables storage administrators to assign meaningful names to iSCSI devices, and therefore, manage those devices more easily.

- **Extended Unique Identifier (EUI):** An EUI is a globally unique identi- fier based on the IEEE EUI-64 naming standard. An EUI is composed of the eui prefix followed by a 16-character hexadecimal name, such as `eui.0300732A32598D26`.

In either format, the allowed special characters are dots, dashes, and blank spaces.

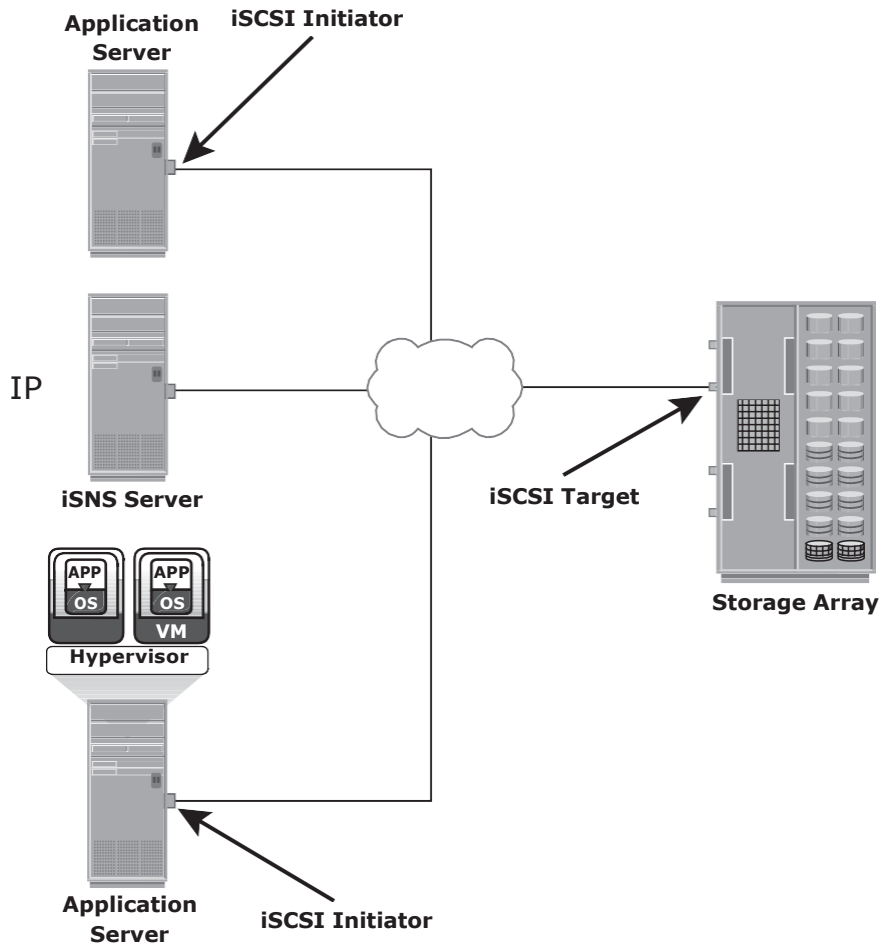


Figure 6-6: Discovery using iSNS

iSCSI Session

An iSCSI session is established between an initiator and a target, as shown in Figure 6-7. A session is identified by a session ID (SSID), which includes part of an initiator ID and a target ID. The session can be intended for one of the following:

- The discovery of the available targets by the initiators and the location of a specific target on a network
- The normal operation of iSCSI (transferring data between initiators and targets)

There might be one or more TCP connections within each session. Each TCP connection within the session has a unique connection ID (CID).

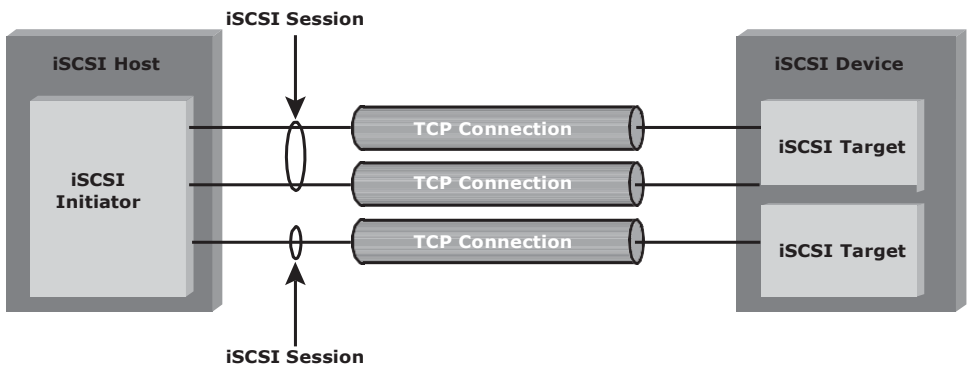


Figure 6-7: iSCSI session

An iSCSI session is established via the iSCSI login process. The login process is started when the initiator establishes a TCP connection with the required target either via the well-known port 3260 or a specified target port. During the login phase, the initiator and the target authenticate each other and negotiate on various parameters.

After the login phase is successfully completed, the iSCSI session enters the full-feature phase for normal SCSI transactions. In this phase, the initiator may send SCSI commands and data to the various LUNs on the target by encapsulating them in iSCSI PDUs that travel over the established TCP connection.

The final phase of the iSCSI session is the connection termination phase, which is referred to as the logout procedure. The initiator is responsible for commencing the logout procedure; however, the target may also prompt termination by sending an iSCSI message, indicating the occurrence of an internal error condition. After the logout request is sent from the initiator and accepted by the target, no further request and response can be sent on that connection.

iSCSI Command Sequencing

The iSCSI communication between the initiators and targets is based on the request-response command sequences. A command sequence may generate multiple PDUs. A *command sequence number* (CmdSN) within an iSCSI session is used for numbering all initiator-to-target command PDUs belonging to the session. This number ensures that every command is delivered in the same order in which it is transmitted, regardless of the TCP connection that carries the command in the session.

Command sequencing begins with the first login command, and the CmdSN is incremented by one for each subsequent command. The iSCSI target layer is responsible for delivering the commands to the SCSI layer in the order of their CmdSN. This ensures the correct order of data and commands at a target even when there are multiple TCP connections between an initiator and the target that use portal groups.

Similar to command numbering, a *status sequence number* (StatSN) is used to sequentially number status responses, as shown in Figure 6-8. These unique numbers are established at the level of the TCP connection.

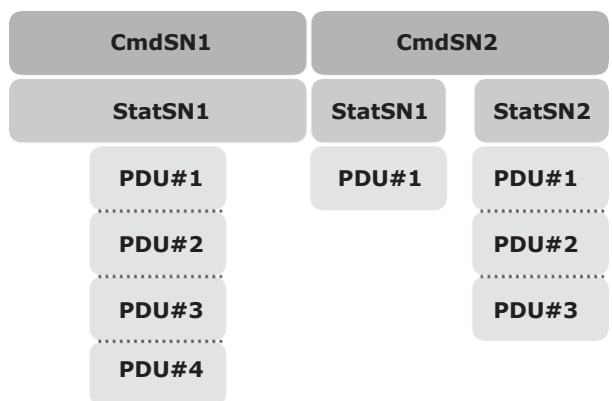


Figure 6-8: Command and status sequence number

A target sends *request-to-transfer* (R2T) PDUs to the initiator when it is ready to accept data. A *data sequence number* (DataSN) is used to ensure in-order delivery of data within the same command. The DataSN and R2TSN are used to sequence data PDUs and R2Ts, respectively. Each of these sequence numbers is stored locally as an unsigned 32-bit integer counter defined by iSCSI. These numbers are communicated between the initiator and target in the appropriate iSCSI PDU fields during command, status, and data exchanges.

For read operations, the DataSN begins at zero and is incremented by one for each subsequent data PDU in that command sequence. For a write operation, the first unsolicited data PDU or the first data PDU in response to an R2T begins with a DataSN of zero and increments by one for each subsequent data PDU. R2TSN is set to zero at the initiation of the command and incremented by one for each subsequent R2T sent by the target for that command.

FCIP

FCSAN provides a high-performance infrastructure for localized data movement. Organizations are now looking for ways to transport data over a long distance between their disparate SANs at multiple geographic locations. One of the best ways to achieve this goal is to interconnect geographically dispersed SANs through reliable, high-speed links. This approach involves transporting the FC block data over the IP infrastructure. FCIP is a tunneling protocol that enables distributed FC SAN islands to be interconnected over the existing IP-based networks.

The FCIP standard has rapidly gained acceptance as a manageable, cost-effective way to blend the best of the two worlds: FC SAN and the proven, widely deployed IP infrastructure. As a result, organizations now have a better way to store, protect and move their data by leveraging investments in their existing IP infrastructure. FCIP is extensively used in disaster recovery implementations in which data is duplicated to the storage located at a remote site.

FCIP might require high network bandwidth when replicating or backing up data. FCIP does not handle data traffic throttling or flow control; these are controlled by the communicating FC switches and devices within the fabric.

6.1.2 FCIP Protocol Stack

The FCIP protocol stack is shown in Figure: 6-9. Applications generate SCSI commands and data, which are processed by various layers of the protocol stack.

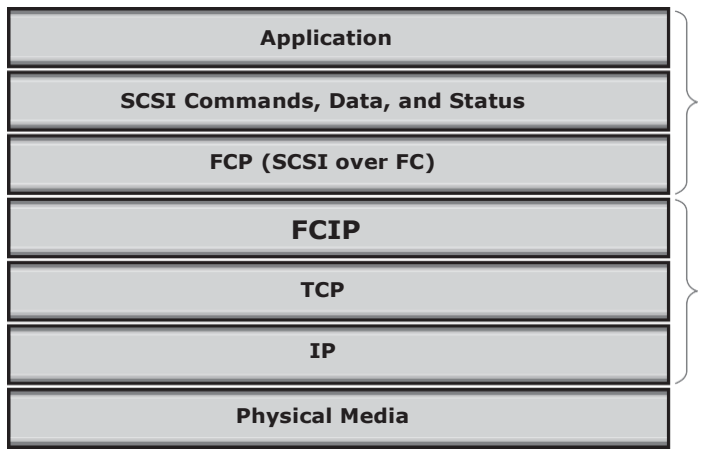


Figure 6-9: FCIP protocol stack

The upper layer protocol SCSI includes the SCSI driver program that executes the read-and-write commands. Below the SCSI layer is the Fibre Channel Protocol (FCP) layer, which is simply a Fibre Channel frame whose payload is SCSI. The FCP layer rides on top of the Fibre Channel transport layer. This enables the FC frames to run natively within a SAN fabric environment. In addition, the FC frames can be encapsulated into the IP packet and sent to a remote SAN over the IP. The FCIP layer encapsulates the Fibre Channel frames onto the IP payload and passes them to the TCP layer (see Figure 6-10). TCP and IP are used for transporting the encapsulated information across Ethernet, wireless, or other media that support the TCP/IP traffic.

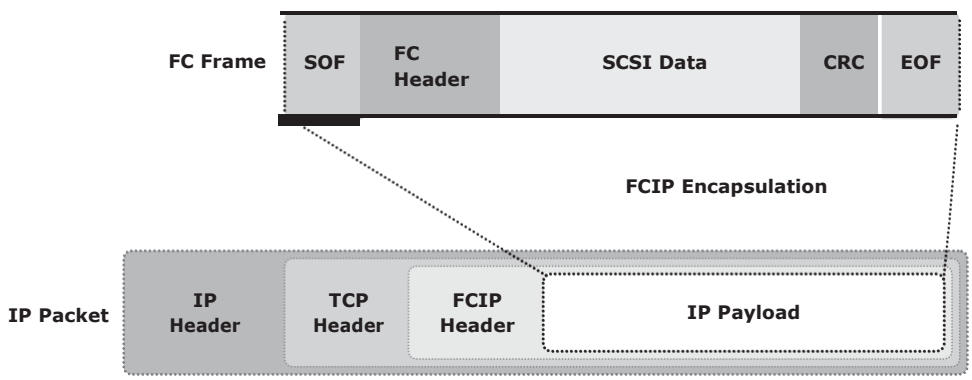


Figure 6-10: FCIP encapsulation

Encapsulation of FC frame into an IP packet could cause the IP packet to be fragmented when the data link cannot support the maximum transmission unit

(MTU) size of an IP packet. When an IP packet is fragmented, the required parts of the header must be copied by all fragments. When a TCP packet is segmented, normal TCP operations are responsible for receiving and re-sequencing the data prior to passing it on to the FC processing portion of the device.

FCIP Topology

In an FCIP environment, an FCIP gateway is connected to each fabric via a standard FC connection (see Figure 6-11). The FCIP gateway at one end of the IP network encapsulates the FC frames into IP packets. The gateway at the other end removes the IP wrapper and sends the FC data to the layer 2 fabric. The fabric treats these gateways as layer 2 fabric switches. An IP address is assigned to the port on the gateway, which is connected to an IP network. After the IP connectivity is established, the nodes in the two independent fabrics can communicate with each other.

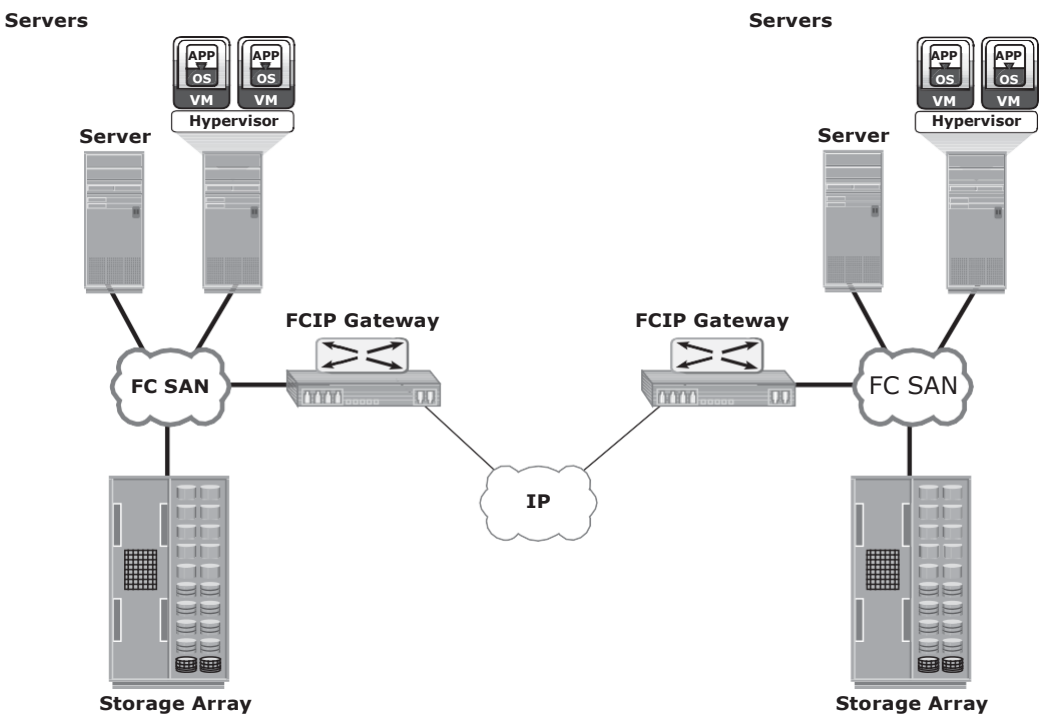


Figure 6-11: FCIP topology

FCIP Performance and Security

Performance, reliability, and security should always be taken into consideration when implementing storage solutions. The implementation of FCIP is also subject to the same considerations.

From the perspective of performance, configuring multiple paths between FCIP gateways eliminates single points of failure and provides increased bandwidth. In a scenario of extended distance, the IP network might be a bottleneck if sufficient bandwidth is not available. In addition, because FCIP creates a unified fabric, disruption in the underlying IP network can cause instabilities in the SAN environment. These instabilities include a segmented fabric, excessive RSCNs, and host timeouts.

The vendors of FC switches have recognized some of the drawbacks related to FCIP and have implemented features to enhance stability, such as the capability to segregate the FCIP traffic into a separate virtual fabric.

Security is also a consideration in an FCIP solution because the data is transmitted over public IP channels. Various security options are available to protect the data based on the router's support. IPSec is one such security measure that can be implemented in the FCIP environment.

FCoE

Data centers typically have multiple networks to handle various types of I/O traffic — for example, an Ethernet network for TCP/IP communication and an FC network for FC communication. TCP/IP is typically used for client-server communication, data backup, infrastructure management communication, and so on. FC is typically used for moving block-level data between storage and servers. To support multiple networks, servers in a data center are equipped with multiple redundant physical network interfaces — for example, multiple Ethernet and FC cards/adapters. In addition, to enable the communication, different types of networking switches and physical cabling infrastructure are implemented in data centers. The need for two different kinds of physical network infrastructure increases the overall cost and complexity of data center operation.

Fibre Channel over Ethernet (FCoE) protocol provides consolidation of LAN and SAN traffic over a single physical interface infrastructure. FCoE helps organizations address the challenges of having multiple discrete network infrastructures. FCoE uses the Converged Enhanced Ethernet (CEE) link (10 Gigabit Ethernet) to send FC frames over Ethernet.

I/O Consolidation Using FCoE

The key benefit of FCoE is I/O consolidation. Figure 6-12 represents the infrastructure before FCoE deployment. Here, the storage resources are accessed using HBAs, and the IP network resources are accessed using NICs by the servers. Typically, in a data center, a server is configured with 2 to 4 NIC cards and redundant HBA cards. If the data center has hundreds of servers, it would

require a large number of adapters, cables, and switches. This leads to a complex environment, which is difficult to manage and scale. The cost of power, cooling, and floor space further adds to the challenge.

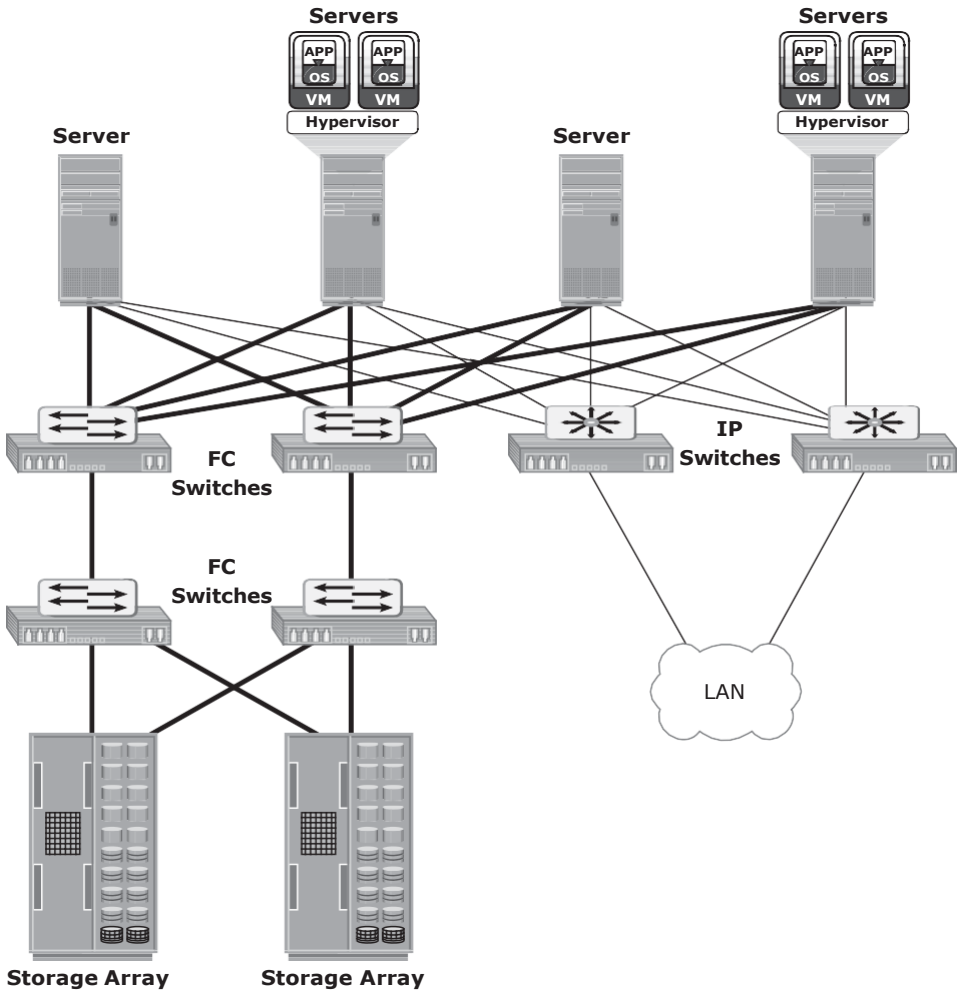


Figure 6-12: Infrastructure before using FCoE

Figure 6-13 shows the I/O consolidation with FCoE using FCoE switches and Converged Network Adapters (CNAs). A CNA (discussed in the section "Converged Network Adapter") replaces both HBAs and NICs in the server and consolidates both the IP and FC traffic. This reduces the requirement of multiple network adapters at the server to connect to different networks. Overall, this reduces the requirement of adapters, cables, and switches. This also considerably reduces the cost and management overhead.

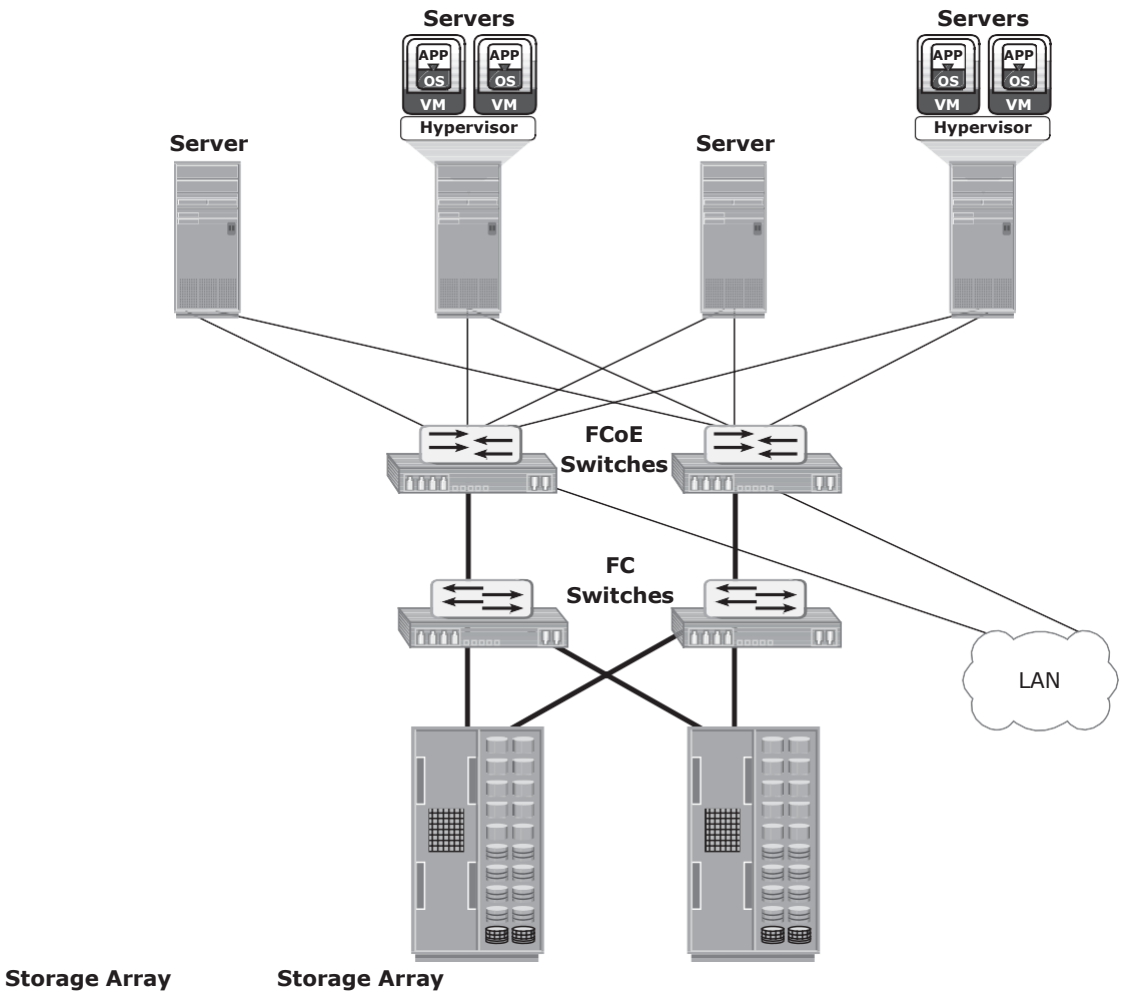


Figure 6-13: Infrastructure after using FCoE

Components of an FCoE Network

This section describes the key physical components required to implement FCoE in a data center. The key FCoE components are:

- Converged Network Adapter (CNA)
- Cables
- FCoE switches

Converged Network Adapter

A CNA provides the functionality of both a standard NIC and an FC HBA in a single adapter and consolidates both types of traffic. CNA eliminates the need to deploy separate adapters and cables for FC and Ethernet communications, thereby reducing the required number of server slots and switch ports. CNA offloads the FCoE protocol processing task from the server, thereby freeing the server CPU resources for application processing. As shown in Figure 6-14, a CNA contains separate modules for 10 Gigabit Ethernet, Fibre Channel, and FCoE Application Specific Integrated Circuits (ASICs). The FCoE ASIC encapsulates FC frames into Ethernet frames. One end of this ASIC is connected to 10GbE and FC ASICs for server connectivity, while the other end provides a 10GbE interface to connect to an FCoE switch.

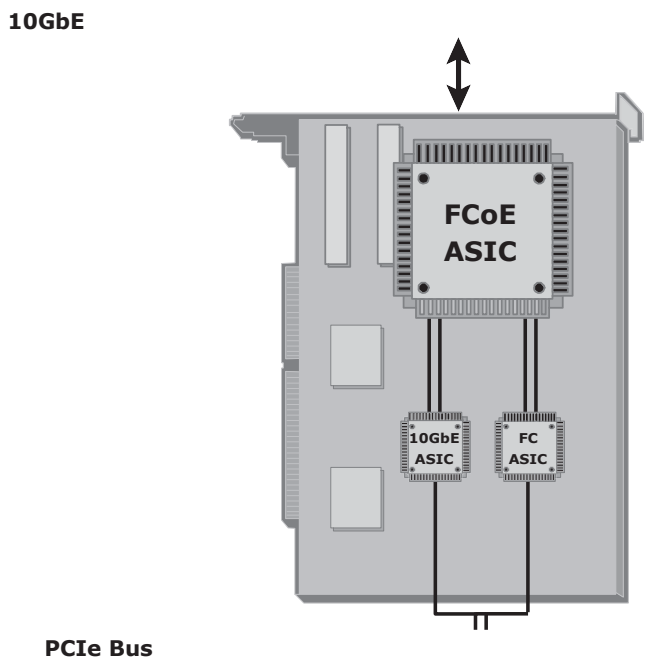


Figure 6-14: Converged Network Adapter

Cables

Currently two options are available for FCoE cabling: Copper based Twinax and standard fiber optical cables. A Twinax cable is composed of two pairs of copper cables covered with a shielded casing. The Twinax cable can transmit data at the speed of 10 Gbps over shorter distances up to 10 meters. Twinax cables require less power and are less expensive than fiber optic cables. The Small Form Factor Pluggable Plus (SFP+) connector is the primary connector used for FCoE links and can be used with both optical and copper cables.

FCoE Switches

An FCoE switch has both Ethernet switch and Fibre Channel switch functionalities. The FCoE switch has a Fibre Channel Forwarder (FCF), Ethernet Bridge, and set of Ethernet ports and optional FC ports, as shown in Figure 6-15. The function of the FCF is to encapsulate the FC frames, received from the FC port, into the FCoE frames and also to de-encapsulate the FCoE frames, received from the Ethernet Bridge, to the FC frames.

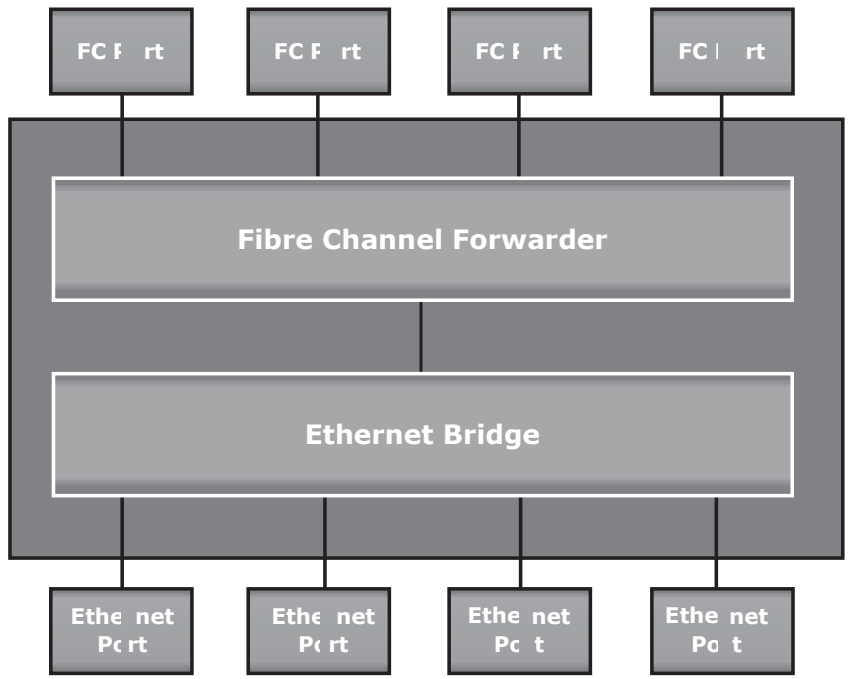


Figure 6-15: FCoE switch generic architecture

Upon receiving the incoming traffic, the FCoE switch inspects the Ethertype (used to indicate which protocol is encapsulated in the payload of an Ethernet frame) of the incoming frames and uses that to determine the destination. If the Ethertype of the frame is FCoE, the switch recognizes that the frame contains an FC payload and forwards it to the FCF. From there, the FC is extracted from the FCoE frame and transmitted to FC SAN over the FC ports. If the Ethertype is not FCoE, the switch handles the traffic as usual Ethernet traffic and forwards it over the Ethernet ports.

FCoE Frame Structure

An FCoE frame is an Ethernet frame that contains an FCoE Protocol Data Unit. Figure 6-16 shows the FCoE frame structure. The first 48-bits in the frame are used to specify the destination MAC address, and the next 48-bits specify the source MAC address. The 32-bit IEEE 802.1Q tag supports the creation of multiple virtual networks (VLANs) across a single physical infrastructure. FCoE has its own Ethertype, as designated by the next 16 bits, followed by the 4-bit version field. The next 100-bits are reserved and are followed by the 8-bit Start of Frame and then the actual FC frame. The 8-bit End of Frame delimiter is followed by 24 reserved bits. The frame ends with the final 32-bits dedicated to the Frame Check Sequence (FCS) function that provides error detection for the Ethernet frame.

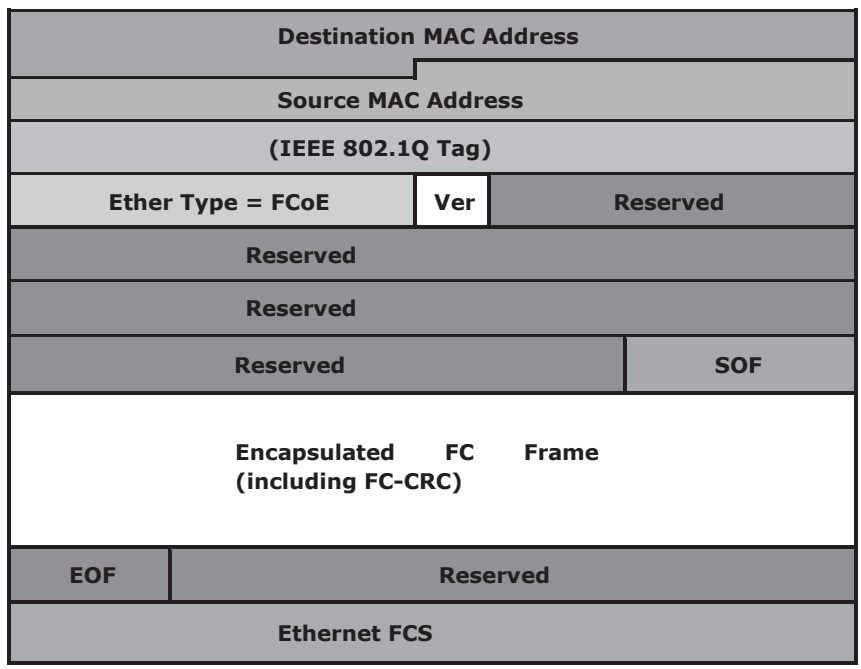


Figure 6-16: FCoE frame structure

The encapsulated Fibre Channel frame consists of the original 24-byte FC header and the data being transported (including the Fibre Channel CRC). The FC frame structure is maintained such that when a traditional FC SAN is connected to an FCoE capable switch, the FC frame is de-encapsulated from the FCoE frame and transported to FC SAN seamlessly. This capability enables FCoE to integrate with the existing FC SANs without the need for a gateway.

Frame size is also an important factor in FCoE. A typical Fibre Channel data frame has a 2,112-byte payload, a 24-byte header, and an FCS. A standard Ethernet frame has a default payload capacity of 1,500 bytes. To maintain good performance, FCoE must use jumbo frames to prevent a Fibre Channel frame from being split into two Ethernet frames. The next chapter discusses jumbo frames in detail. FCoE requires Converged Enhanced Ethernet, which provides lossless Ethernet and jumbo frame support.

FCoE Frame Mapping

The encapsulation of the Fibre Channel frame occurs through the mapping of the FC frames onto Ethernet, as shown in Figure 6-17. Fibre Channel and traditional networks have stacks of layers where each layer in the stack represents a set of functionalities. The FC stack consists of five layers: FC-0 through FC-4. Ethernet is typically considered as a set of protocols that operates at the physical and data link layers in the seven layer OSI stack. The FCoE protocol specification replaces the FC-0 and FC-1 layers of the FC stack with Ethernet. This provides the capability to carry the FC-2 to the FC-4 layer over the Ethernet layer.

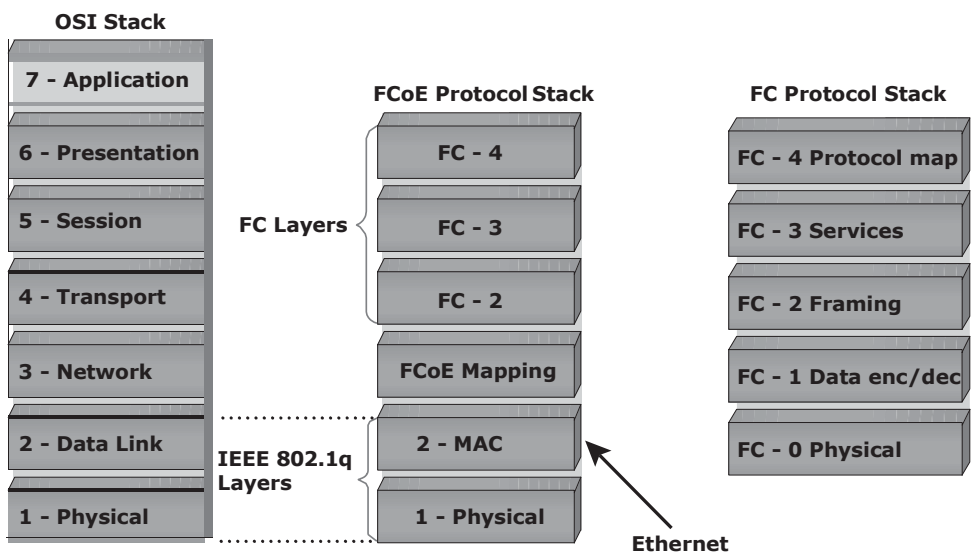


Figure 6-17: FCoE frame mapping

FCoE Enabling Technologies

Conventional Ethernet is lossy in nature, which means that frames might be dropped or lost during transmission. *Converged Enhanced Ethernet* (CEE), or lossless Ethernet, provides a new specification to the existing Ethernet standard that eliminates the lossy nature of Ethernet. This makes 10 Gb Ethernet a viable storage networking option, similar to FC. Lossless Ethernet requires certain functionalities. These functionalities are defined and maintained by the data center bridging (DCB) task group, which is a part of the IEEE 802.1 working group, and they are:

- Priority-based flow control
- Enhanced transmission selection

- Congestion Notification
- Data center bridging exchange protocol

Priority-Based Flow Control (PFC)

Traditional FC manages congestion through the use of a link-level, credit-based flow control that guarantees no loss of frames. Typical Ethernet, coupled with TCP/IP, uses a packet drop flow control mechanism. The packet drop flow control is not lossless. This challenge is eliminated by using an IEEE 802.3x Ethernet PAUSE control frame to create a lossless Ethernet. A receiver can send a PAUSE request to a sender when the receiver's buffer is filling up. Upon receiving a PAUSE frame, the sender stops transmitting frames, which guarantees no loss of frames. The downside of using the Ethernet PAUSE frame is that it operates on the entire link, which might be carrying multiple traffic flows.

PFC provides a link level flow control mechanism. PFC creates eight separate virtual links on a single physical link and allows any of these links to be paused and restarted independently. PFC enables the pause mechanism based on user priorities or classes of service. Enabling the pause based on priority allows creating lossless links for traffic, such as FCoE traffic. This PAUSE mechanism is typically implemented for FCoE while regular TCP/IP traffic continues to drop frames. Figure 6-18 illustrates how a physical Ethernet link is divided into eight virtual links and allows a PAUSE for a single virtual link without affecting the traffic for the others.

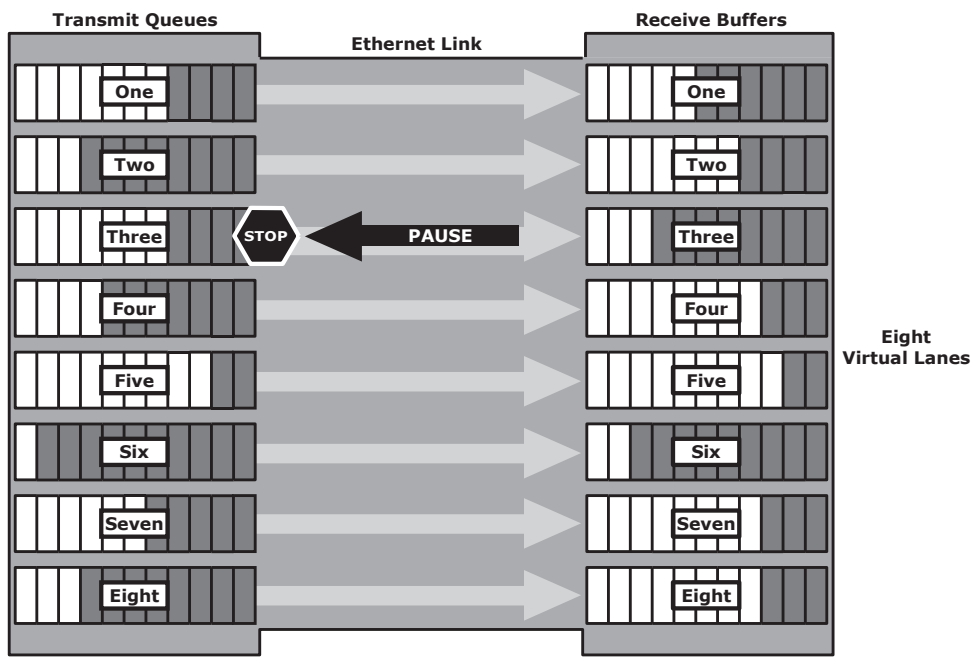


Figure 6-18: Priority-based flow control

Enhanced Transmission Selection (ETS)

Enhanced transmission selection provides a common management framework for the assignment of bandwidth to different traffic classes, such as LAN, SAN, and Inter Process Communication (IPC). When a particular class of traffic does not use its allocated bandwidth, ETS enables other traffic classes to use the available bandwidth.

Congestion Notification (CN)

Congestion notification provides end-to-end congestion management for protocols, such as FCoE, that do not have built-in congestion control mechanisms. Link level congestion notification provides a mechanism for detecting congestion and notifying the source to move the traffic flow away from the congested links. Link level congestion notification enables a switch to send a signal to other ports that need to stop or slow down their transmissions. The process of congestion notification and its management is shown in Figure 6-19, which represents the communication between the nodes A (sender) and B (receiver). If congestion at the receiving end occurs, the algorithm running on the switch generates a congestion notification message to the sending node (Node A). In response to the CN message, the sending end limits the rate of data transfer.

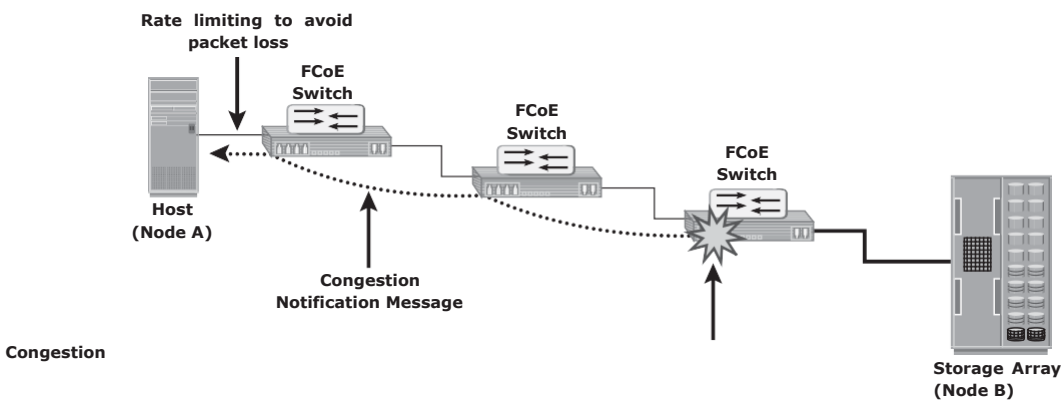


Figure 6-19: Congestion Notification

Data Center Bridging Exchange Protocol (DCBX)

DCBX protocol is a discovery and capability exchange protocol, which helps Converged Enhanced Ethernet devices to convey and configure their features with the other CEE devices in the network. DCBX is used to negotiate capabilities

between the switches and the adapters, and it allows the switch to distribute the configuration values to all the attached adapters. This helps to ensure consistent configuration across the entire network.

General-Purpose Servers versus NAS Devices

A NAS device is optimized for file-serving functions such as storing, retrieving, and accessing files for applications and clients. As shown in Figure 7-1, a general-purpose server can be used to host any application because it runs a general-purpose operating system. Unlike a general-purpose server, a NAS device is dedicated to file-serving. It has specialized operating system dedicated to file serving by using industry-standard protocols. Some NAS vendors support features, such as native clustering for high availability.

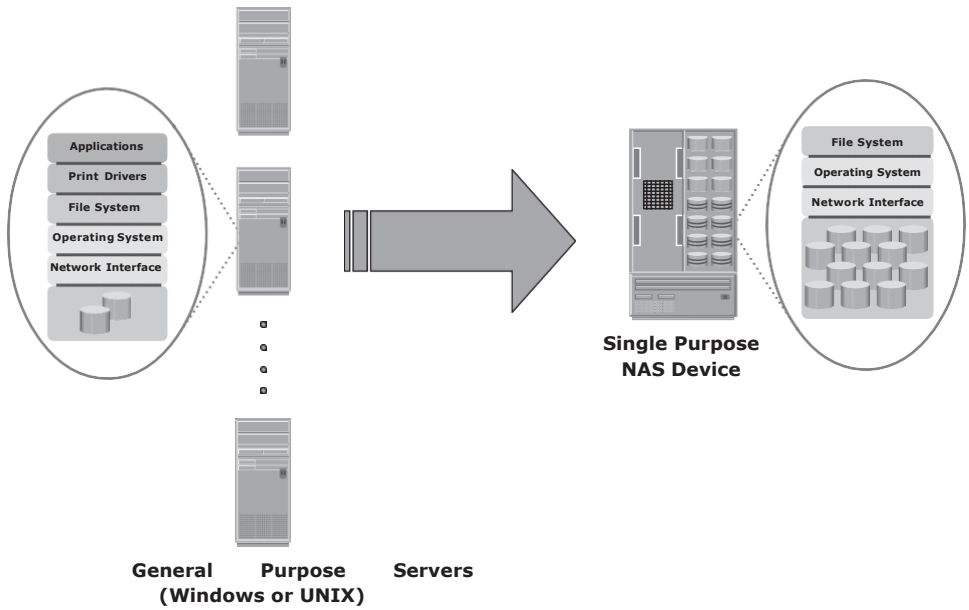


Figure 7-1: General purpose server versus NAS device

Benefits of NAS

NAS offers the following benefits:

- **Comprehensive access to information:** Enables efficient file sharing and supports many-to-one and one-to-many configurations. The many-to-one configuration enables a NAS device to serve many clients simultaneously. The one-to-many configuration enables one client to connect with many NAS devices simultaneously.
- **Improved efficiency:** NAS delivers better performance compared to a general-purpose file server because NAS uses an operating system specialized for file serving.
- **Improved flexibility:** Compatible with clients on both UNIX and Windows platforms using industry-standard protocols. NAS is flexible and can serve requests from different types of clients from the same source.
- **Centralized storage:** Centralizes data storage to minimize data duplication on client workstations, and ensure greater data protection
- **Simplified management:** Provides a centralized console that makes it possible to manage file systems efficiently

- **Scalability:** Scales well with different utilization profiles and types of business applications because of the high-performance and low-latency design
- **High availability:** Offers efficient replication and recovery options, enabling high data availability. NAS uses redundant components that provide maximum connectivity options. A NAS device supports clustering technology for failover.
- **Security:** Ensures security, user authentication, and file locking with industry-standard security schemas
- **Low cost:** NAS uses commonly available and inexpensive Ethernet components.
- **Ease of deployment:** Configuration at the client is minimal, because the clients have required NAS connection software built in.

File Systems and Network File Sharing

A *file system* is a structured way to store and organize data files. Many file systems maintain a file access table to simplify the process of searching and accessing files.

Accessing a File System

A file system must be mounted before it can be used. In most cases, the operating system mounts a local file system during the boot process. The mount process creates a link between the file system on the NAS and the operating system on the client. When mounting a file system, the operating system organizes files and directories in a tree-like structure and grants the privilege to the user to access this structure. The tree is rooted at a mount point. The mount point is named using operating system conventions. Users and applications can traverse the entire tree from the root to the leaf nodes as file system permissions allow. Files are located at leaf nodes, and directories and subdirectories are located at intermediate roots. The access to the file system terminates when the file system is unmounted. Figure 7-2 shows an example of a UNIX directory structure.

Network File Sharing

Network file sharing refers to storing and accessing files over a network. In a file-sharing environment, the user who creates a file (the creator or owner of a file) determines the type of access (such as read, write, execute, append, and

delete) to be given to other users and controls changes to the file. When multiple users try to access a shared file at the same time, a locking scheme is required to maintain data integrity and, at the same time, make this sharing possible.

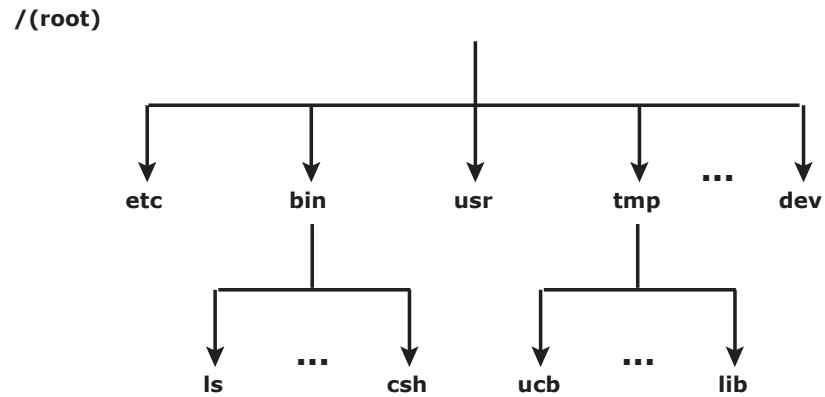


Figure 7-2: UNIX directory structure

Some examples of file-sharing methods are file transfer protocol (FTP), Distributed File System (DFS), client-server models that use file-sharing protocols such as NFS and CIFS, and the peer-to-peer (P2P) model

FTP is a client-server protocol that enables data transfer over a network. An FTP server and an FTP client communicate with each other using TCP as the transport protocol. FTP, as defined by the standard, is not a secure method of data transfer because it uses unencrypted data transfer over a network. FTP over Secure Shell (SSH) adds security to the original FTP specification. When FTP is used over SSH, it is referred to as Secure FTP (SFTP).

A *distributed file system* (DFS) is a file system that is distributed across several hosts. A DFS can provide hosts with direct access to the entire file system, while ensuring efficient management and data security. Standard client-server file-sharing protocols, such as NFS and CIFS, enable the owner of a file to set the required type of access, such as read-only or read-write, for a particular user or group of users. Using this protocol, the clients mount remote file systems that are available on dedicated file servers.

A name service, such as Domain Name System (DNS), and directory services such as Microsoft Active Directory, and Network Information Services (NIS), helps users identify and access a unique resource over the network. A *name service protocol* such as the Lightweight Directory Access Protocol (LDAP) creates a namespace, which holds the unique name of every network resource and helps recognize resources on the network.

A *peer-to-peer* (P2P) file sharing model uses a peer-to-peer network. P2P enables client machines to directly share files with each other over a

network. Clients use a file sharing software that searches for other peer clients. This differs from the client-server model that uses file servers to store files for sharing.

Components of NAS

A NAS device has two key components: NAS head and storage (see Figure 7-3). In some NAS implementations, the storage could be external to the NAS device and shared with other hosts. The NAS head includes the following components:

- CPU and memory
- One or more network interface cards (NICs), which provide connectivity to the client network. Examples of network protocols supported by NIC include Gigabit Ethernet, Fast Ethernet, ATM, and Fiber Distributed Data Interface (FDDI).
- An optimized operating system for managing the NAS functionality. It translates file-level requests into block-storage requests and further converts the data supplied at the block level to file data.
- NFS, CIFS, and other protocols for file sharing
- Industry-standard storage protocols and ports to connect and manage physical disk resources

The NAS environment includes clients accessing a NAS device over an IP network using file-sharing protocols.

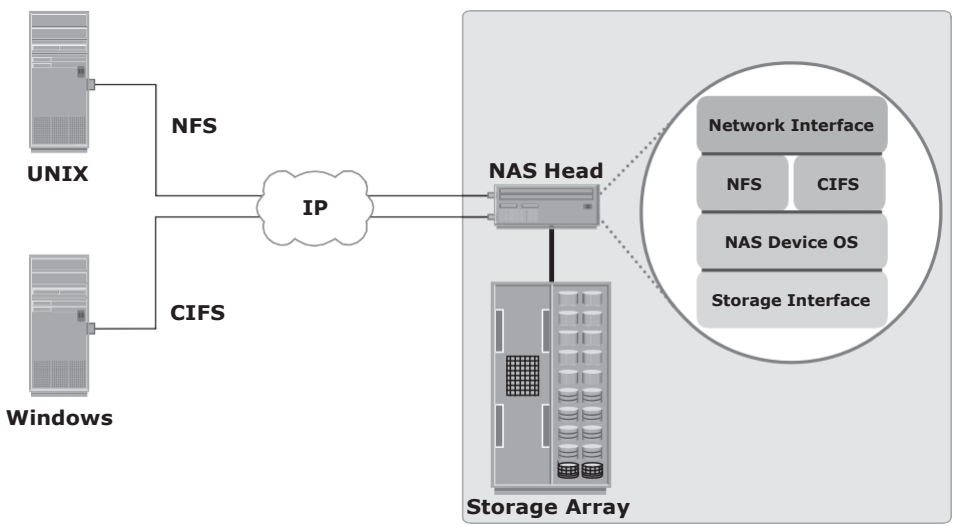


Figure 7-3: Components of NAS

NAS I/O Operation

NAS provides file-level data access to its clients. File I/O is a high-level request that specifies the file to be accessed. For example, a client may request a file by specifying its name, location, or other attributes. The NAS operating system keeps track of the location of files on the disk volume and converts client file I/O into block-level I/O to retrieve data. The process of handling I/Os in a NAS environment is as follows:

1. The requestor (client) packages an I/O request into TCP/IP and forwards it through the network stack. The NAS device receives this request from the network.
2. The NAS device converts the I/O request into an appropriate physical storage request, which is a block-level I/O, and then performs the operation on the physical storage.
3. When the NAS device receives data from the storage, it processes and repackages the data into an appropriate file protocol response.
4. The NAS device packages this response into TCP/IP again and forwards it to the client through the network.

Figure 7-4 illustrates this process.

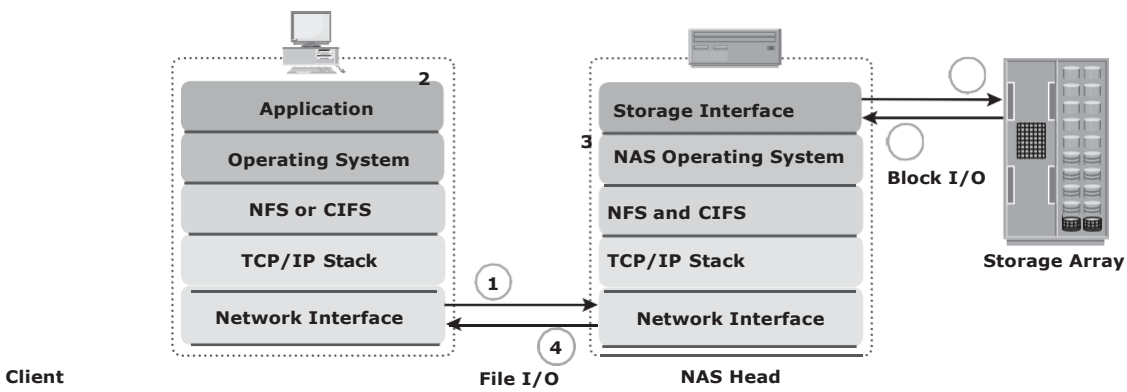


Figure 7-4: NAS I/O operation

NAS Implementations

Three common NAS implementations are unified, gateway, and scale-out. The *unified* NAS consolidates NAS-based and SAN-based data access within a unified storage platform and provides a unified management interface for managing both the environments.

In a *gateway* implementation, the NAS device uses external storage to store and retrieve data, and unlike unified storage, there are separate administrative tasks for the NAS device and storage.

The *scale-out* NAS implementation pools multiple nodes together in a cluster. A node may consist of either the NAS head or storage or both. The cluster performs the NAS operation as a single entity.

Unified NAS

Unified NAS performs file serving and storing of file data, along with providing access to block-level data. It supports both CIFS and NFS protocols for file access and iSCSI and FC protocols for block level access. Due to consolidation of NAS-based and SAN-based access on a single storage platform, unified NAS reduces an organization's infrastructure and management costs.

A unified NAS contains one or more NAS heads and storage in a single system. NAS heads are connected to the storage controllers (SCs), which provide access to the storage. These storage controllers also provide connectivity to iSCSI and FC hosts. The storage may consist of different drive types, such as SAS, ATA, FC, and flash drives, to meet different workload requirements.

Unified NAS Connectivity

Each NAS head in a unified NAS has front-end Ethernet ports, which connect to the IP network. The front-end ports provide connectivity to the clients and service the file I/O requests. Each NAS head has back-end ports, to provide connectivity to the storage controllers.

iSCSI and FC ports on a storage controller enable hosts to access the storage directly or through a storage network at the block level. Figure 7-5 illustrates an example of unified NAS connectivity.

Gateway NAS

A gateway NAS device consists of one or more NAS heads and uses external and independently managed storage. Similar to unified NAS, the storage is shared with other applications that use block-level I/O. Management functions in this type of solution are more complex than those in a unified NAS environment because there are separate administrative tasks for the NAS head and the storage. A gateway solution can use the FC infrastructure, such as switches and directors for accessing SAN-attached storage arrays or direct-attached storage arrays.

The gateway NAS is more scalable compared to unified NAS because NAS heads and storage arrays can be independently scaled up when required.

For example, NAS heads can be added to scale up the NAS device performance. When the storage limit is reached, it can scale up, adding capacity on the SAN, independent of NAS heads. Similar to a unified NAS, a gateway NAS also enables high utilization of storage capacity by sharing it with the SAN environment.

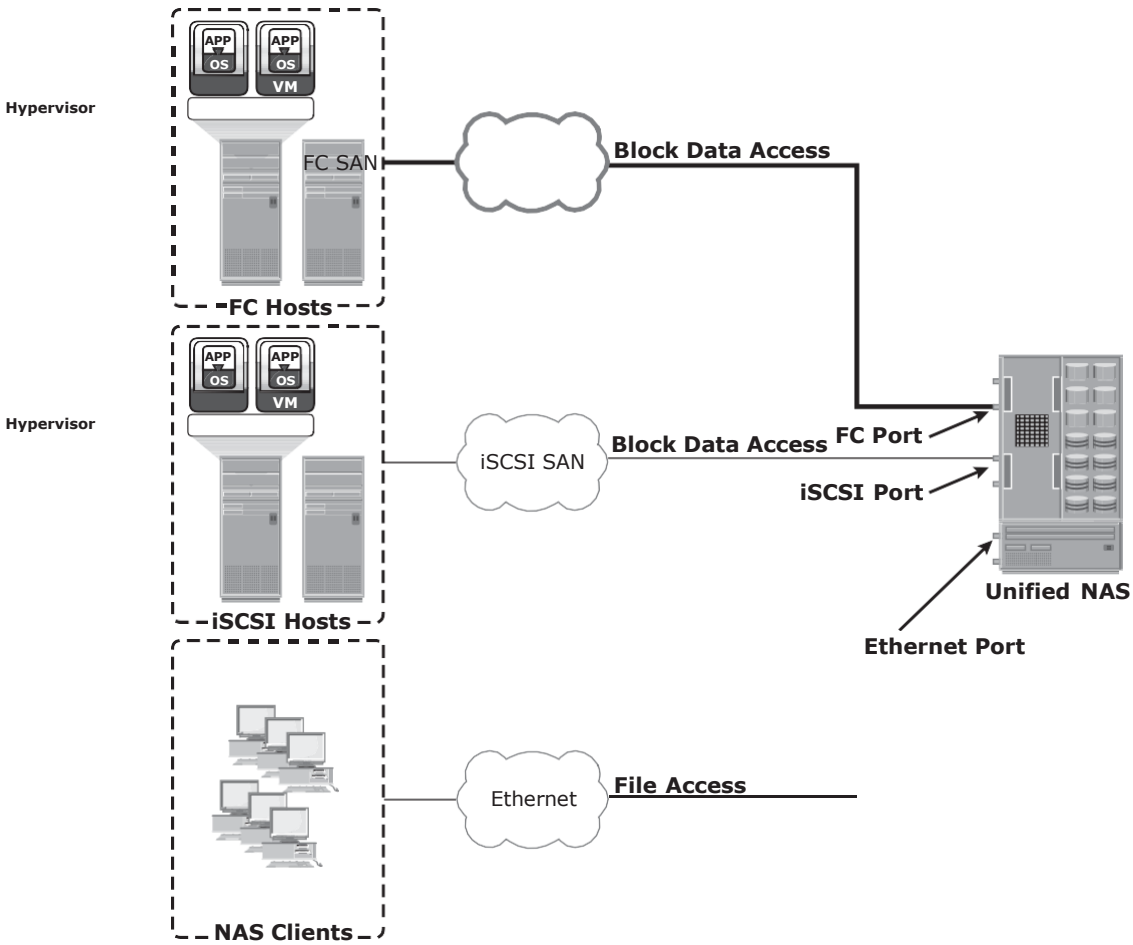


Figure 7-5: Unified NAS connectivity

Gateway NAS Connectivity

In a gateway solution, the front-end connectivity is similar to that in a unified storage solution. Communication between the NAS gateway and the storage system in a gateway solution is achieved through a traditional FC SAN. To deploy a gateway NAS solution, factors, such as multiple paths for data, redundant

fabrics, and load distribution, must be considered. Figure 7-6 illustrates an example of gateway NAS connectivity.

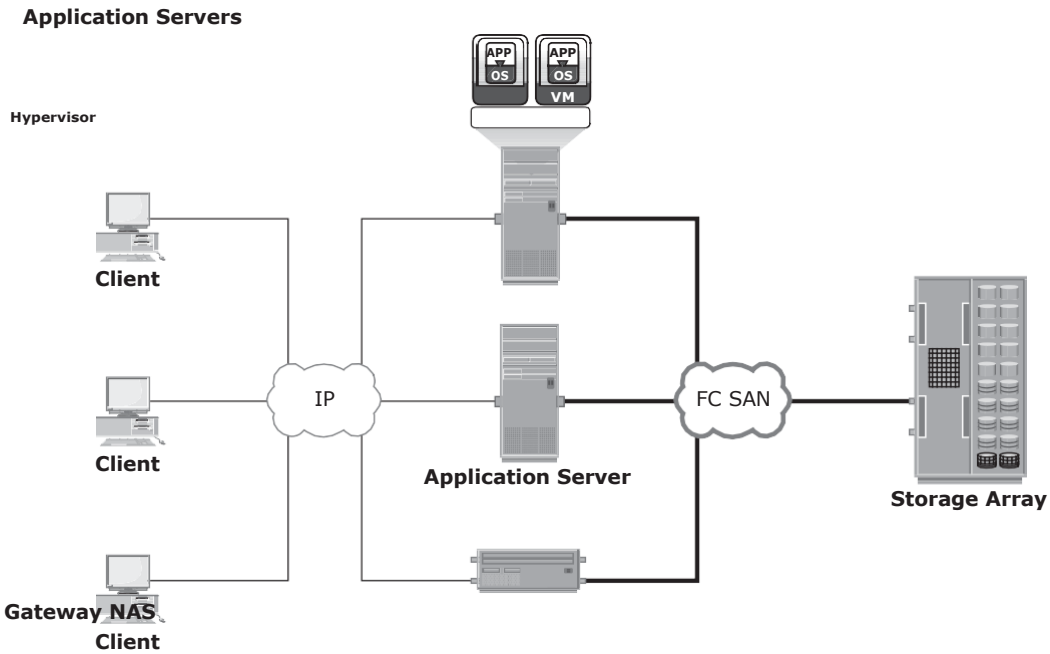


Figure 7-6: Gateway NAS connectivity

Implementation of both unified and gateway solutions requires analysis of the SAN environment. This analysis is required to determine the feasibility of combining the NAS workload with the SAN workload. Analyze the SAN to determine whether the workload is primarily read or write, and if it is random or sequential. Also determine the predominant I/O size in use. Typically, NAS workloads are random with small I/O sizes. Introducing sequential workload with random workloads can be disruptive to the sequential workload. Therefore, it is recommended to separate the NAS and SAN disks. Also, determine whether the NAS workload performs adequately with the configured cache in the storage system.

Scale-Out NAS

Both unified and gateway NAS implementations provide the capability to scale-up their resources based on data growth and rise in performance requirements. Scaling up these NAS devices involves adding CPUs, memory, and storage to

the NAS device. Scalability is limited by the capacity of the NAS device to house and use additional NAS heads and storage.

Scale-out NAS enables grouping multiple nodes together to construct a clustered NAS system. A scale-out NAS provides the capability to scale its resources by simply adding nodes to a clustered NAS architecture. The cluster works as a single NAS device and is managed centrally. Nodes can be added to the cluster, when more performance or more capacity is needed, without causing any downtime. Scale-out NAS provides the flexibility to use many nodes of moderate performance and availability characteristics to produce a total system that has better aggregate performance and availability. It also provides ease of use, low cost, and theoretically unlimited scalability.

Scale-out NAS creates a single file system that runs on all nodes in the cluster. All information is shared among nodes, so the entire file system is accessible by clients connecting to any node in the cluster. Scale-out NAS stripes data across all nodes in a cluster along with mirror or parity protection. As data is sent from clients to the cluster, the data is divided and allocated to different nodes in parallel. When a client sends a request to read a file, the scale-out NAS retrieves the appropriate blocks from mul- tiple nodes, recombines the blocks into a file, and presents the file to the client. As nodes are added, the file system grows dynamically and data is evenly distributed to every node. Each node added to the cluster increases the aggregate storage, memory, CPU, and network capacity. Hence, cluster performance also increases.

Scale-out NAS is suitable to solve the "Big Data" challenges that enterprises and customers face today. It provides the capability to manage and store large, high-growth data in a single place with the flexibility to meet a broad range of performance requirements.

Scale-Out NAS Connectivity

Scale-out NAS clusters use separate internal and external networks for back-end and front-end connectivity, respectively. An internal network provides connections for intracluster communication, and an external network connection enables clients to access and share file data. Each node in the cluster connects to the internal network. The internal network offers high throughput and low latency and uses high-speed networking technology, such as InfiniBand or Gigabit Ethernet. To enable clients to access a node, the node must be connected to the external Ethernet network. Redundant internal or external networks may be used for high availability. Figure 7-7 illustrates an example of scale-out NAS connectivity.

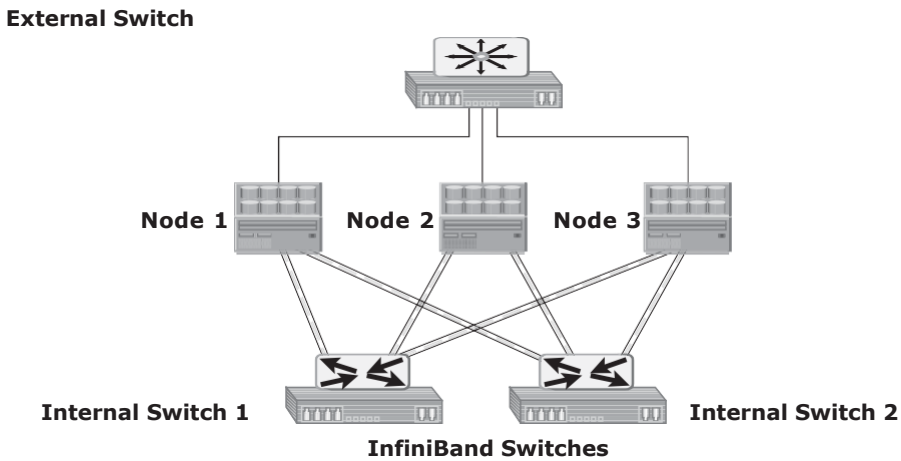


Figure 7-7: Scale-out NAS with dual internal and single external networks

NAS File-Sharing Protocols

Most NAS devices support multiple file-service protocols to handle file I/O requests to a remote file system. As discussed earlier, NFS and CIFS are the common protocols for file sharing. NAS devices enable users to share file data across different operating environments and provide a means for users to migrate transparently from one operating system to another.

NFS

NFS is a client-server protocol for file sharing that is commonly used on UNIX systems. NFS was originally based on the connectionless *User Datagram Protocol* (UDP). It uses a machine-independent model to represent user data. It also uses Remote Procedure Call (RPC) as a method of inter-process communication between two computers. The NFS protocol provides a set of RPCs to access a remote file system for the following operations:

- Searching files and directories
- Opening, reading, writing to, and closing a file
- Changing file attributes
- Modifying file links and directories

NFS creates a connection between the client and the remote system to transfer data. NFS (NFSv3 and earlier) is a *stateless protocol*, which means that it does not maintain any kind of table to store information about open files and associated pointers. Therefore, each call provides a full set of arguments to access files on the server. These arguments include a file handle reference to the file, a particular position to read or write, and the versions of NFS.

Currently, three versions of NFS are in use:

- **NFS version 2 (NFSv2):** Uses UDP to provide a stateless network connection between a client and a server. Features, such as locking, are handled outside the protocol.
- **NFS version 3 (NFSv3):** The most commonly used version, which uses UDP or TCP, and is based on the stateless protocol design. It includes some new features, such as a 64-bit file size, asynchronous writes, and additional file attributes to reduce refetching.
- **NFS version 4 (NFSv4):** Uses TCP and is based on a stateful protocol design. It offers enhanced security. The latest NFS version 4.1 is the enhancement of NFSv4 and includes some new features, such as session model, parallel NFS (pNFS), and data retention.

(Continued)

CIFS

CIFS is a client-server application protocol that enables client programs to make requests for files and services on remote computers over TCP/IP. It is a public, or open, variation of Server Message Block (SMB) protocol.

The CIFS protocol enables remote clients to gain access to files on a server. CIFS enables file sharing with other clients by using special locks. Filenames in CIFS are encoded using unicode characters. CIFS provides the following features to ensure data integrity:

- It uses file and record locking to prevent users from overwriting the work of another user on a file or a record.
- It supports fault tolerance and can automatically restore connections and reopen files that were open prior to an interruption. The fault tolerance features of CIFS depend on whether an application is written to take advantage of these features. Moreover, CIFS is a stateful protocol because the CIFS server maintains connection information regarding every connected

client. If a network failure or CIFS server failure occurs, the client receives a disconnection notification. User disruption is minimized if the application has the embedded intelligence to restore the connection. However, if the embedded intelligence is missing, the user must take steps to reestablish the CIFS connection.

Users refer to remote file systems with an easy-to-use file-naming scheme: `\\server\share` or `\\servername.domain.suffix\share`.

The file naming scheme in an NFS environment is:
`Server:/export` or **`Server.domain.suffix:/export`.**

Factors Affecting NAS Performance

NAS uses IP network; therefore, bandwidth and latency issues associated with IP affect NAS performance. Network congestion is one of the most significant sources of latency (Figure 7-8) in a NAS environment. Other factors that affect NAS performance at different levels follow:

1. **Number of hops:** A large number of hops can increase latency because IP processing is required at each hop, adding to the delay caused at the router.
2. **Authentication with a directory service such as Active Directory or NIS:** The authentication service must be available on the network with enough resources to accommodate the authentication load. Otherwise, a large number of authentication requests can increase latency.
3. **Retransmission:** Link errors and buffer overflows can result in retransmission. This causes packets that have not reached the specified destination to be re-sent. Care must be taken to match both speed and duplex settings on the network devices and the NAS heads. Improper configuration might result in errors and retransmission, adding to latency.
4. **Overutilized routers and switches:** The amount of time that an overutilized device in a network takes to respond is always more than the response time of an optimally utilized or underutilized device. Network administrators can view utilization statistics to determine the optimum utilization of switches and routers in a network. Additional devices should be added if the current devices are overutilized.

5. **File system lookup and metadata requests:** NAS clients access files on NAS devices. The processing required to reach the appropriate file or directory can cause delays. Sometimes a delay is caused by deep directory structures and can be resolved by flattening the directory structure. Poor file system layout and an overutilized disk system can also degrade performance.
6. **Over utilized NAS devices:** Clients accessing multiple files can cause high utilization levels on a NAS device, which can be determined by viewing utilization statistics. High memory, CPU, or disk subsystem utilization levels can be caused by a poor file system structure or insufficient resources in a storage subsystem.
7. **Over utilized clients:** The client accessing CIFS or NFS data might also be over utilized. An overutilized client requires a longer time to process the requests and responses. Specific performance-monitoring tools are available for various operating systems to help determine the utilization of client resources.

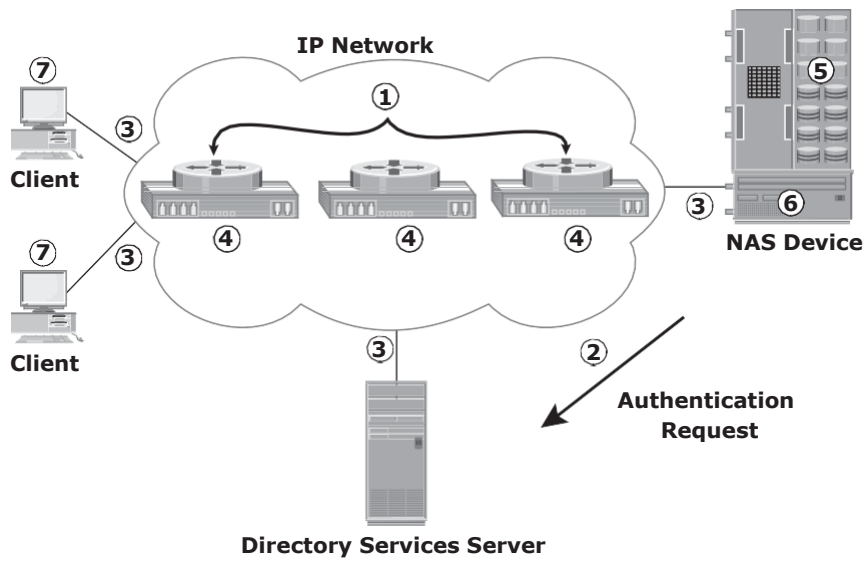


Figure 7-8: Causes of latency

Configuring *virtual LANs* (VLANs), setting proper Maximum Transmission Unit (MTU) and TCP window sizes, and link aggregation can improve NAS performance. Link aggregation and redundant network configurations also ensure high availability.

A VLAN is a logical segment of a switched network or logical grouping of end devices connected to different physical networks. An end device could be a client or a NAS device. The segmentation or grouping can be done based on business functions, project teams, or applications. VLAN is a Layer 2 (data link layer) construct and works similar to a physical LAN. A network switch can be logically divided among multiple VLANs, enabling better utilization of the switch and reducing the overall cost of deploying a network infrastructure.

The broadcast traffic on one VLAN is not transmitted outside that VLAN, which substantially reduces the broadcast overhead, makes bandwidth available for applications, and reduces the network's vulnerability to broadcast storms.

VLANs also provide enhanced security by restricting user access, flagging network intrusions, and controlling the size and composition of the broadcast domain. The *MTU* setting determines the size of the largest packet that can be transmitted without data fragmentation. *Path maximum transmission unit discovery* is the process of discovering the maximum size of a packet that can be sent across a network without fragmentation. The default MTU setting for an Ethernet interface card is 1,500 bytes. A feature called *jumbo frames* sends, receives, or transports Ethernet frames with an MTU of more than 1,500 bytes. The most common deployments of jumbo frames have an MTU of 9,000 bytes. However not all vendors use the same MTU size for jumbo frames. Servers send and receive larger frames more efficiently than smaller ones in heavy network traffic conditions. Jumbo frames ensure increased efficiency because it takes fewer, larger frames to transfer the same amount of data. Larger packets also reduce the amount of raw network bandwidth being consumed for the same amount of payload. Larger frames also help to smooth sudden I/O bursts.

The *TCP window size* is the maximum amount of data that can be sent at any time for a connection. For example, if a pair of hosts is talking over a TCP connection that has a TCP window size of 64 KB, the sender can send only 64 KB of data and must then wait for an acknowledgment from the receiver. If the receiver acknowledges that all the data has been received, then the sender is free to send another 64 KB of data. If the sender receives an acknowledgment from the receiver that only the first 32 KB of data has been received, which can happen only if another 32 KB of data is in transit or was lost, the sender can send only another 32 KB of data because the transmission cannot have more than 64 KB of unacknowledged data outstanding.

In theory, the TCP window size should be set to the product of the available bandwidth of the network and the round-trip time of data sent over the network.

For example, if a network has a bandwidth of 100 Mbps and the round-trip time is 5 milliseconds, the TCP window should be as follows:

100 Mb/s × .005 seconds = 524,288 bits or 65,536 bytes

The size of the TCP window field that controls the flow of data is between 2 bytes and 65,535 bytes.

Link aggregation is the process of combining two or more network interfaces into a logical network interface, enabling higher throughput, load sharing or load balancing, transparent path failover, and scalability. Due to link aggregation, multiple active Ethernet connections to the same switch appear as one link. If a connection or a port in the aggregation is lost, then all the network traffic on that link is redistributed across the remaining active connections.

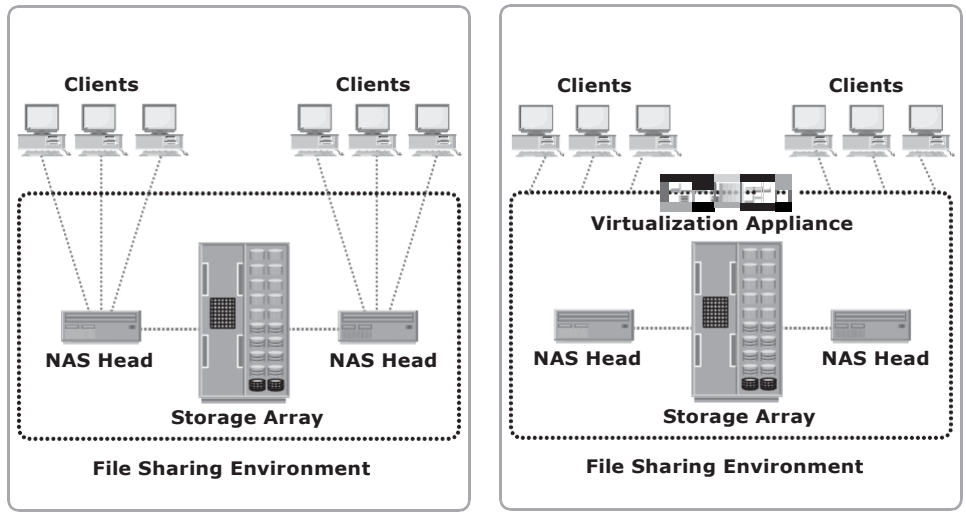
File-Level Virtualization

File-level virtualization eliminates the dependencies between the data accessed at the file level and the location where the files are physically stored. Implementation of file-level virtualization is common in NAS or file-server environments. It provides non-disruptive file mobility to optimize storage utilization.

Before virtualization, each host knows exactly where its file resources are located. This environment leads to underutilized storage resources and capacity problems because files are bound to a specific NAS device or file server. It may be required to move the files from one server to another because of performance reasons or when the file server fills up. Moving files across the environment is not easy and may make files inaccessible during file movement. Moreover, hosts and applications need to be reconfigured to access the file at the new location. This makes it difficult for storage administrators to improve storage efficiency while maintaining the required service level.

File-level virtualization simplifies file mobility. It provides user or application independence from the location where the files are stored. File-level virtualization creates a logical pool of storage, enabling users to use a logical path, rather than a physical path, to access files. File-level virtualization facilitates the movement of files across the online file servers or NAS devices. This means that while the files are being moved, clients can access their files nondisruptively. Clients can also read their files from the old location and write them back to the new location without realizing that the physical location has changed. A global namespace is used to map the logical path of a file to the physical path names.

Figure 7-9 illustrates a file-serving environment before and after the implementation of file-level virtualization.



(a) Before File-Level Virtualization (b) After File-Level Virtualization

Figure 7-9: File-serving environment before and after file-level virtualization

Concepts in Practice: EMC Isilon and EMC VNX Gateway

EMC Isilon is the scale-out NAS solution. Isilon offers high scalability of both performance and storage capacity. It provides the capability to address big-data challenges.

The VNX Gateway, a member of the EMC VNX family, provides a gateway NAS solution. It provides multiprotocol file access, dynamic expansion of file systems, high availability, and high performance.

For more information on EMC Isilon and VNX Gateway, visit `www.emc.com`.

EMC Isilon

Isilon has a specialized operating system called OneFS that enables the scale-out NAS architecture. OneFS combines the three layers of traditional storage architectures — file system, volume manager, and RAID — into one unified software layer, creating a single file system that spans across all nodes in an Isilon cluster. OneFS enables data protection and automated data balancing. It provides the ability to seamlessly add storage and other resources without system downtime. With OneFS, throughput scales linearly with the number of nodes in a cluster.

OneFS enables different node types to be mixed in a single cluster through the addition of the SmartPools application software. SmartPools enables deploying a single file system to span multiple nodes that have different performance characteristics and capacities. Isilon offers different types of nodes, such as the X-Series, S-Series, NL-Series, and Accelerator. These nodes have different prices, performance levels, and storage capabilities. Each type of node is optimized for handling a specific type of workload.

OneFS enables the storage system administrator to specify the access pattern (random, concurrent, or sequential) on a per-file or per-directory basis. This unique capability enables OneFS to tailor data layout decisions, cache-retention policies, and data prefetch policies to maximize performance of individual workflows.

OneFS constantly monitors the health of all files and disks within a cluster, and if components are at risk, the file system automatically flags the problem components for replacement and transparently relocates those files to healthy components. OneFS also ensures data integrity if the file system has an unexpected failure during a write operation.

When a new storage node is added, the Autobalance feature of OneFS automatically moves data onto this new node via the Infiniband based internal network. This automatic rebalancing ensures that the new node does not become a hot spot for new data. The Autobalance feature is transparent to the clients and can be adjusted to minimize the impact on high-performance workloads.

OneFS includes a core technology, called FlexProtect, to provide data protection. FlexProtect provides protection for up to four simultaneous failures of either nodes or individual drives per stripe. FlexProtect ensures minimal data reconstruction time if a failure occurs. FlexProtect provides file-specific protection capabilities. Different protection levels can be assigned to individual files, directories, or to portions of a file system. These protection levels are aligned based on the importance of data and workflow.

EMC VNX Gateway

The VNX Series Gateway contains one or more NAS heads, called X-Blades, that access external storage arrays, such as Symmetrix, block-based VNX, or CLARiiON storage array, via SAN. X-Blades run the VNX operating environment that is optimized for high-performance and multiprotocol network file system access. Each X-Blade consists of processors, redundant data paths, power supplies, Gigabit Ethernet, and 10-Gigabit Ethernet optical ports. All the X-Blades in a VNX gateway system are managed by Control Station, which provides a single point for configuring VNX Gateway. The VNX Gateway supports both pNFS and EMC patented Multi-Path File System (MPFS) protocols, which further improves the VNX Gateway performance.

VNX Series Gateway offers two models: VG2 and VG8. VG8 supports up to eight X-Blades, whereas VG2 supports up to two. X-Blades may be configured as either primary or standby. A primary X-Blade is the operating NAS head, whereas a standby X-Blade becomes operational if the primary X-Blade fails. The Control Station handles an X-Blade failover. The Control Station also provides other high-availability features, such as fault monitoring, fault reporting, call home, and remote diagnostics.

Object-Based Storage Devices

An OSD is a device that organizes and stores unstructured data, such as movies, office documents, and graphics, as objects. Object-based storage provides a scalable, self-managed, protected, and shared storage option. OSD stores data in the form of *objects*. OSD uses flat address space to store data. Therefore, there is no hierarchy of directories and files; as a result, a large number of objects can be stored in an OSD system (see Figure 8-1).

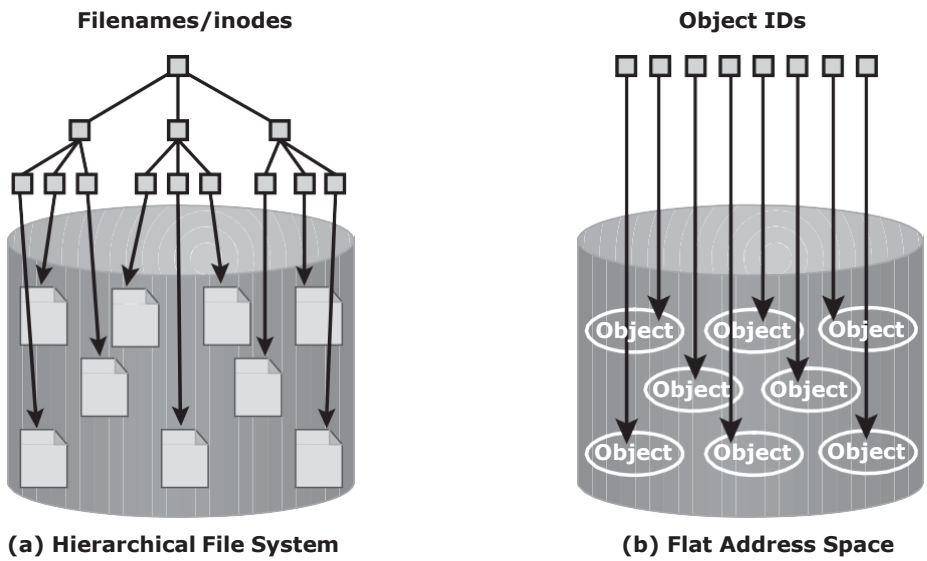


Figure 8-1: Hierarchical file system versus flat address space

An object might contain user data, related metadata (size, date, ownership, and so on), and other attributes of data (retention, access pattern, and so on); see Figure 8-2. Each object stored in the system is identified by a unique ID called the *object ID*. The object ID is generated using specialized algorithms such as hash function on the data and guarantees that every object is uniquely identified.

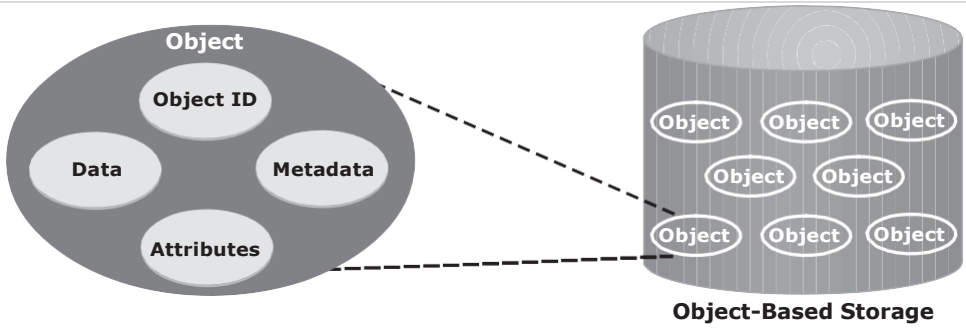


Figure 8-2: Object structure

Object-Based Storage Architecture

An I/O in the traditional block access method passes through various layers in the I/O path. The I/O generated by an application passes through the file system, the channel, or network and reaches the disk drive. When the file system receives the I/O from an application, the file system maps the incoming I/O to the disk blocks. The block interface is used for sending the I/O over the channel or network to the storage device. The I/O is then written to the block allocated on the disk drive. Figure 8-3 (a) illustrates the block-level access.

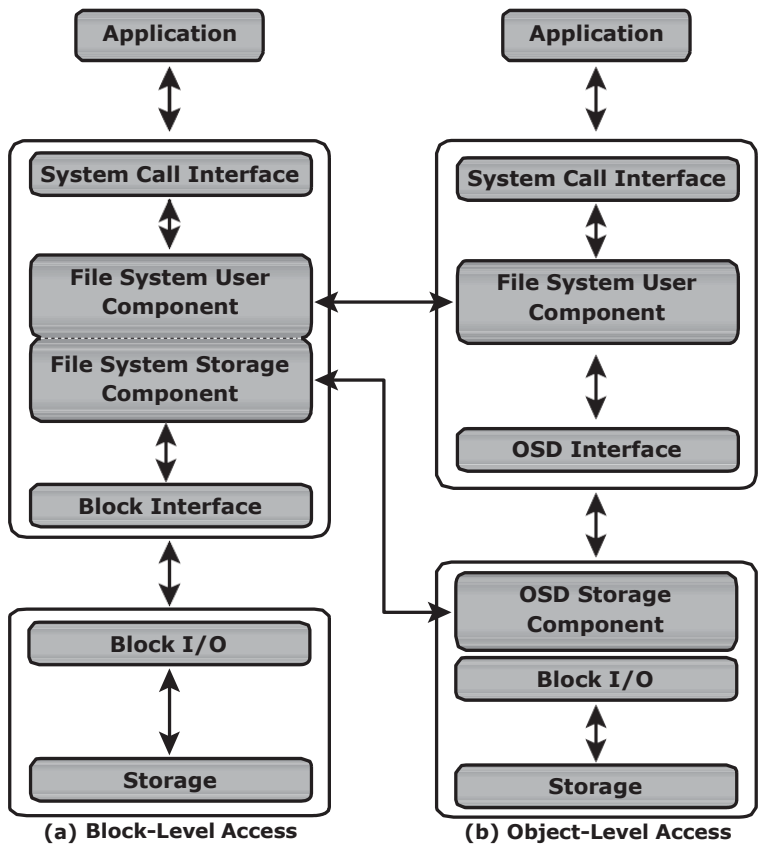


Figure 8-3: Block-level access versus object-level access

The file system has two components: user component and storage component. The user component of the file system performs functions such as hierarchy management, naming, and user access control. The storage component maps the files to the physical location on the disk drive.

When an application accesses data stored in OSD, the request is sent to the file system user component. The file system user component communicates to the OSD interface, which in turn sends the request to the storage device. The storage device has the OSD storage component responsible for managing the access to the object on a storage device. Figure 8-3 (b) illustrates the object-level access. After the object is stored, the OSD sends an acknowledgment to the application server. The OSD storage component manages all the required low-level storage and space management functions. It also manages security and access control functions for the objects.

Components of OSD

The OSD system is typically composed of three key components: nodes, private network, and storage. Figure 8-4 illustrates the components of OSD.

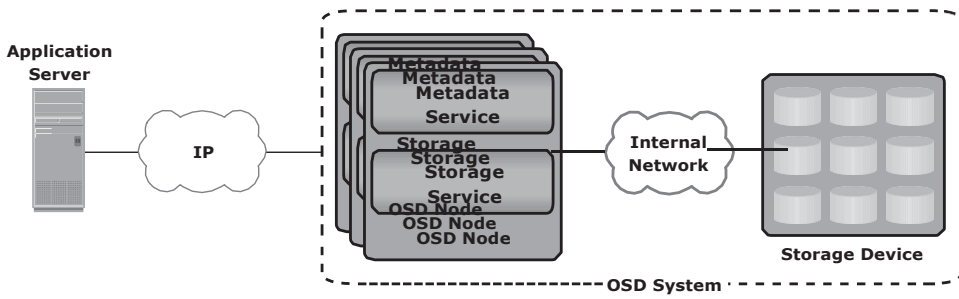


Figure 8-4: OSD components

The OSD system is composed of one or more *nodes*. A node is a server that runs the OSD operating environment and provides services to store, retrieve, and manage data in the system. The OSD node has two key services: metadata service and storage service. The metadata service is responsible for generating the object ID from the contents (and can also include other attributes of data) of a file. It also maintains the mapping of the object IDs and the file system namespace. The storage service manages a set of disks on which the user data is stored. The OSD nodes connect to the storage via an internal network. The internal network provides node-to-node connectivity and node-to-storage connectivity. The application server accesses the node to store and retrieve data over an external network. In some implementations, such as CAS, the metadata service might reside on the application server or on a separate server.

OSD typically uses low-cost and high-density disk drives to store the objects. As more capacity is required, more disk drives can be added to the system.

Object Storage and Retrieval in OSD

The process of storing objects in OSD is illustrated in Figure 8-5. The data storage process in an OSD system is as follows:

1. The application server presents the file to be stored to the OSD node.
2. The OSD node divides the file into two parts: user data and metadata.
3. The OSD node generates the object ID using a specialized algorithm. The algorithm is executed against the contents of the user data to derive an ID unique to this data.
4. For future access, the OSD node stores the metadata and object ID using the metadata service.
5. The OSD node stores the user data (objects) in the storage device using the storage service.
6. An acknowledgment is sent to the application server stating that the object is stored.

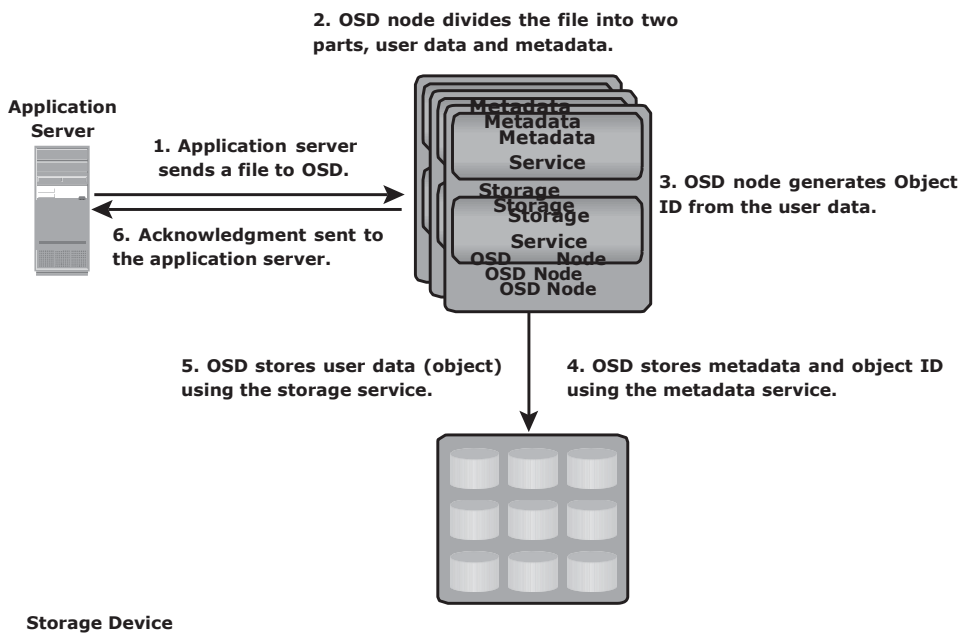


Figure 8-5: Storing objects on OSD

After an object is stored successfully, it is available for retrieval. A user accesses the data stored on OSD by the same filename. The application server retrieves the stored content using the object ID. This process is transparent to the user.

The process of retrieving objects in OSD is illustrated in Figures 8-6. The process of data retrieval from OSD is as follows:

1. The application server sends a read request to the OSD system.
2. The metadata service retrieves the object ID for the requested file.
3. The metadata service sends the object ID to the application server.
4. The application server sends the object ID to the OSD storage service for object retrieval.
5. The OSD storage service retrieves the object from the storage device.
6. The OSD storage service sends the file to the application server.

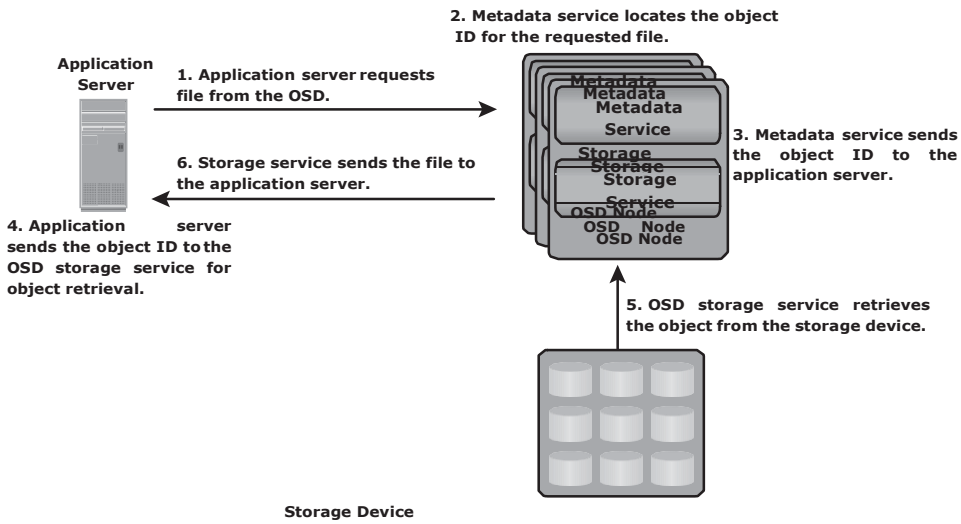


Figure 8-6: Object retrieval from an OSD system

Benefits of Object-Based Storage

For unstructured data, object-based storage devices provide numerous benefits over traditional storage solutions. An ideal storage architecture should provide performance, scalability, security, and data sharing across multiple platforms. Traditional storage solutions, such as SAN and NAS, do not offer all these benefits as a single solution. Object-based storage combines benefits of both the worlds. It provides platform and location independence, and at the same time, provides scalability, security, and data-sharing capabilities. The key benefits of object-based storage are as follows:

- **Security and reliability:** Data integrity and content authenticity are the key features of object-based storage devices. OSD uses specialized algorithms

to create objects that provide strong data encryption capability. In OSD, request authentication is performed at the storage device rather than with an external authentication mechanism.

- **Platform independence:** Objects are abstract containers of data, including metadata and attributes. This feature allows objects to be shared across heterogeneous platforms locally or remotely. This platform-independence capability makes object-based storage the best candidate for cloud computing environments.
- **Scalability:** Due to the use of flat address space, object-based storage can handle large amounts of data without impacting performance. Both storage and OSD nodes can be scaled independently in terms of performance and capacity.
- **Manageability:** Object-based storage has an inherent intelligence to manage and protect objects. It uses self-healing capability to protect and replicate objects. Policy-based management capability helps OSD to handle routine jobs automatically.

Common Use Cases for Object-Based Storage

A data archival solution is a promising use case for OSD. Data integrity and protection is the primary requirement for any data archiving solu- tion. Traditional archival solutions — CD and DVD-ROM — do not provide scalability and performance. OSD stores data in the form of objects, associates them with a unique object ID, and ensures high data integrity. Along with integrity, it provides scalability and data protection. These capabilities make OSD a viable option for long term data archiving for fixed content. Content addressed storage (CAS) is a special type of object-based storage device purposely built for storing fixed content. CAS is covered in the following section.

Another use case for OSD is cloud-based storage. OSD uses a web interface to access storage resources. OSD provides inherent security, scalability, and automated data management. It also enables data sharing across heterogeneous platforms or tenants while ensuring integrity of data. These capabilities make OSD a strong option for cloud-based storage. Cloud service providers can leverage OSD to offer storage-as-a-service.

OSD supports web service access via *representational state transfer* (REST) and *simple object access protocol* (SOAP). REST and SOAP APIs can be easily integrated with business applications that access OSD over the web.

Content-Addressed Storage

CAS is an object-based storage device designed for secure online storage and retrieval of fixed content. CAS stores user data and its attributes as an object. The stored object is assigned a globally unique address, known as a *content address* (CA). This address is derived from the object's binary representation. CAS provides an optimized and centrally managed storage solution. Data access in CAS differs from other OSD devices. In CAS, the application server access the CAS device only via the CAS API running on the application server. However, the way CAS stores data is similar to the other OSD systems.

CAS provides all the features required for storing fixed content. The key features of CAS are as follows:

- **Content authenticity:** It assures the genuineness of stored content. This is achieved by generating a unique content address for each object and validating the content address for stored objects at regular intervals. Content authenticity is assured because the address assigned to each object is as unique as a fingerprint. Every time an object is read, CAS uses a hashing algorithm to recalculate the object's content address as a validation step and compares the result to its original content address. If the object fails validation, CAS rebuilds the object using a mirror or parity protection scheme.
- **Content integrity:** It provides assurance that the stored content has not been altered. CAS uses a hashing algorithm for content authenticity and integrity. If the fixed content is altered, CAS generates a new address for the altered content, rather than overwrite the original fixed content.
- **Location independence:** CAS uses a unique content address, rather than directory path names or URLs, to retrieve data. This makes the physical location of the stored data irrelevant to the application that requests the data.
- **Single-instance storage (SIS):** CAS uses a unique content address to guarantee the storage of only a single instance of an object. When a new object is written, the CAS system is polled to see whether an object is already available with the same content address. If the object is available in the system, it is not stored; instead, only a pointer to that object is created.
- **Retention enforcement:** Protecting and retaining objects is a core requirement of an archive storage system. After an object is stored in the CAS system and the retention policy is defined, CAS does not make the object available for deletion until the policy expires.
- **Data protection:** CAS ensures that the content stored on the CAS system is available even if a disk or a node fails. CAS provides both local and remote

protection to the data objects stored on it. In the local protection option, data objects are either mirrored or parity protected. In mirror protection, two copies of the data object are stored on two different nodes in the same cluster. This decreases the total available capacity by 50 percent. In parity protection, the data object is split in multiple parts and parity is generated from them. Each part of the data and its parity are stored on a different node. This method consumes less capacity to protect the stored data, but takes slightly longer to regenerate the data if corruption of data occurs.

In the remote replication option, data objects are copied to a secondary CAS at the remote location. In this case, the objects remain accessible from the secondary CAS if the primary CAS system fails.

- **Fast record retrieval:** CAS stores all objects on disks, which provides faster access to the objects compared to tapes and optical discs.
- **Load balancing:** CAS distributes objects across multiple nodes to provide maximum throughput and availability.
- **Scalability:** CAS allows the addition of more nodes to the cluster without any interruption to data access and with minimum administrative overhead.
- **Event notification:** CAS continuously monitors the state of the system and raises an alert for any event that requires the administrator's attention. The event notification is communicated to the administrator through SNMP, SMTP, or e-mail.
- **Self diagnosis and repair:** CAS automatically detects and repairs corrupted objects and alerts the administrator about the potential problem. CAS systems can be configured to alert remote support teams who can diagnose and repair the system remotely.
- **Audit trails:** CAS keeps track of management activities and any access or disposition of data. Audit trails are mandated by compliance requirements.

CAS Use Cases

Organizations have deployed CAS solutions to solve several business challenges. Two solutions are described in detail in the following sections.

Healthcare Solution: Storing Patient Studies

Large healthcare centers examine hundreds of patients every day and generate large volumes of medical records. Each record might be composed of one

or more images that range in size from approximately 15 MB for a standard digital X-ray to more than 1 GB for oncology studies. The patient records are stored online for a specific period of time for immediate use by the attending physicians. Even if a patient's record is no longer needed, compliance requirements might stipulate that the records be kept in the original format for several years.

Medical image solution providers offer hospitals the capability to view medical records, such as X-ray images, with acceptable response times and resolution to enable rapid assessments of patients. Figure 8-7 illustrates the use of CAS in this scenario. Patients' records are retained on the primary storage for 60 days after which they are moved to the CAS system. CAS facilitates long-term storage and at the same time, provides immediate access to data, when needed.

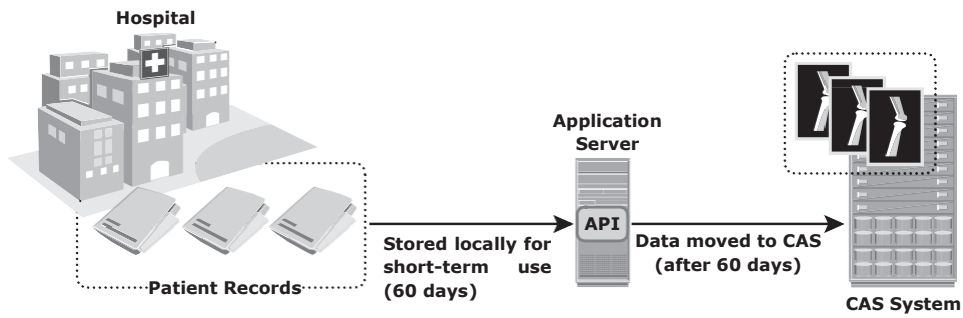


Figure 8-7: Storing patient studies on a CAS system

Finance Solution: Storing Financial Records

In a typical banking scenario, images of checks, each approximately 25 KB in size, are created and sent to archive services over an IP network. A check imaging service provider might process approximately 90 million check images per month. Typically, check images are actively processed in transactional systems for about 5 days.

For the next 60 days, check images may be requested by banks or individual consumers for verification purposes; beyond 60 days, access requirements drop drastically. Figure 8-8 illustrates the use of CAS in this scenario. The check images are moved from the primary storage to the CAS system after 60 days, and can be held there for long term based on retention policy. Check imaging is one example of a financial service application that is best serviced with CAS. Customer transactions initiated by e-mail, contracts, and security transaction records might need to be kept online for 30 years; CAS is the preferred storage solution in such cases.

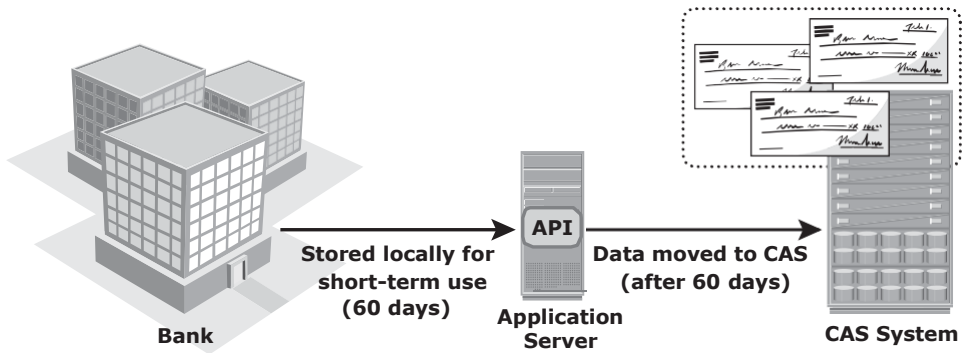


Figure 8-8: Storing financial records on a CAS system

Unified Storage

Unified storage consolidates block, file, and object access into one storage solution. It supports multiple protocols, such as CIFS, NFS, iSCSI, FC, FCoE, REST (representational state transfer), and SOAP (simple object access protocol).

Components of Unified Storage

A unified storage system consists of the following key components: storage controller, NAS head, OSD node, and storage. Figure 8-9 illustrates the block diagram of a unified storage platform.

The *storage controller* provides block-level access to application servers through iSCSI, FC, or FCoE protocols. It contains iSCSI, FC, and FCoE front-end ports for direct block access. The storage controller is also responsible for managing the back-end storage pool in the storage system. The controller configures LUNs and presents them to application servers, NAS heads, and OSD nodes. The LUNs presented to the application server appear as local physical disks. A file system is configured on these LUNs and is made available to applications for storing data.

A *NAS head* is a dedicated file server that provides file access to NAS clients. The NAS head is connected to the storage via the storage controller typically using a FC or FCoE connection. The system typically has two or more NAS heads for redundancy. The LUNs presented to the NAS head appear as physical disks. The NAS head configures the file systems on these disks, creates a NFS, CIFS, or mixed share, and exports the share to the NAS clients.

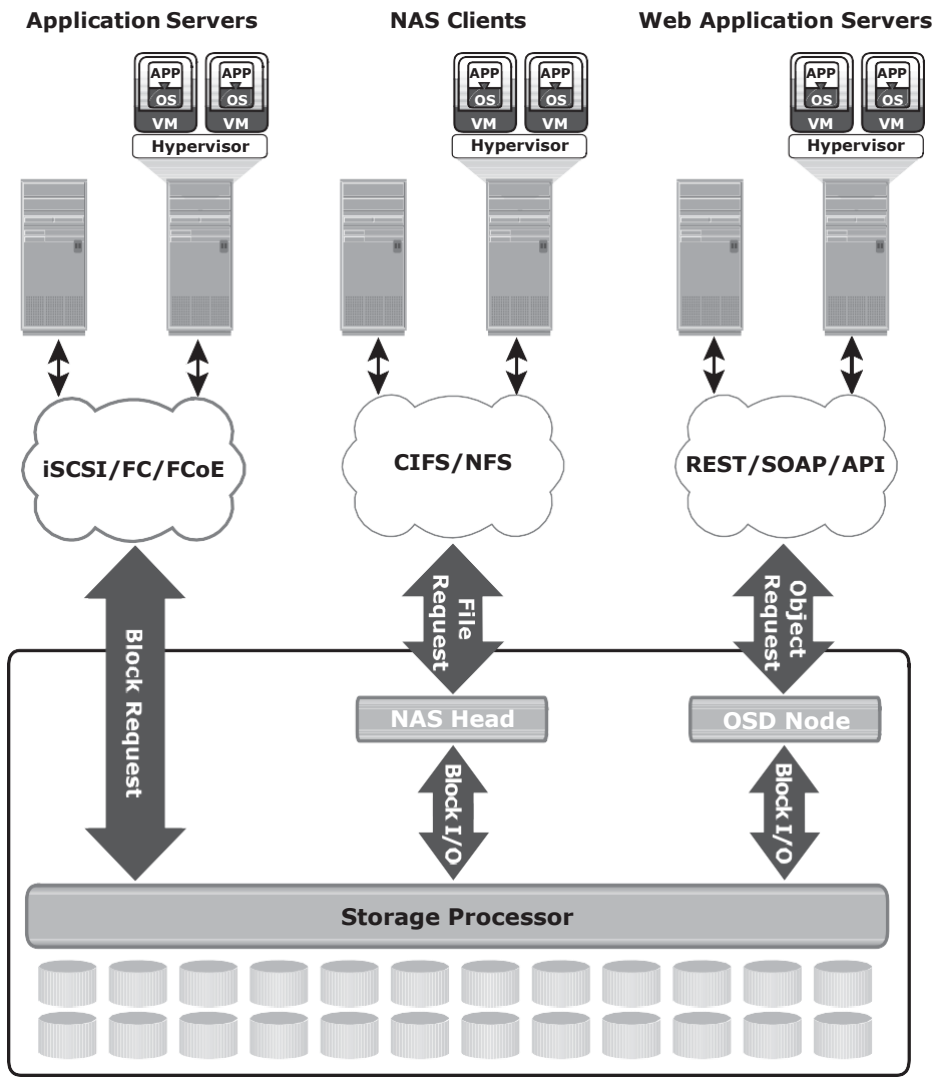


Figure 8-9: Unified storage platform

The *OSD node* accesses the storage through the storage controller using a FC or FCoE connection. The LUNs assigned to the OSD node appear as physical disks. These disks are configured by the OSD nodes, enabling them to store the data from the web application servers.

Data Access from Unified Storage

In a unified storage system, block, file, and object requests to the storage travel through different I/O paths. Figure 8-9 illustrates the different I/O paths for block, file, and object access.

- **Block I/O request:** The application servers are connected to an FC, iSCSI, or FCoE port on the storage controller. The server sends a block request over an FC, iSCSI, or FCoE connection. The storage processor (SP) processes the I/O and responds to the application server.
- **File I/O request:** The NAS clients (where the NAS share is mounted or mapped) send a file request to the NAS head using the NFS or CIFS protocol. The NAS head receives the request, converts it into a block request, and forwards it to the storage controller. Upon receiving the block data from the storage controller, the NAS head again converts the block request back to the file request and sends it to the clients.
- **Object I/O request:** The web application servers send an object request, typically using REST or SOAP protocols, to the OSD node. The OSD node receives the request, converts it into a block request, and sends it to the disk through the storage controller. The controller in turn processes the block request and responds back to the OSD node, which in turn provides the requested object to the web application server.

Concepts in Practice: EMC Atmos, EMC VNX, and EMC Centera

EMC Atmos supports object-based storage for unstructured data, such as pictures and videos. Atmos combines massive scalability with specialized intelligence to address the cost, distribution, and management challenges associated with vast amounts of unstructured data.

EMC VNX is a unified storage platform that consolidates block, file, and object access in one solution. It implements a modular architecture that integrates hardware components for block, file, and object access. EMC VNX delivers file access (NAS) functionality via X-Blades (Data Movers) and block access functionality via storage processors. Optionally, it offers object access to the storage using EMC Atmos Virtual Edition (Atmos VE).

EMC Centera is a simple, affordable, and secure repository for information archiving. EMC Centera is designed and optimized specifically to deal with the storage and retrieval of fixed content by meeting performance, compliance, and regulatory requirements. Compared to traditional archive storage, EMC Centera provides faster record retrieval, Single instance storage (SIS), guaranteed content authenticity, self-healing, and support for numerous industry and regulatory standards.

For the latest information on EMC Atmos, EMC VNX, and EMC Centera, visit www.emc.com.

EMC Atmos

Atmos can be deployed in two ways: as a purpose-built hardware appliance or as software in VMware environments, where AtmosVE can leverage the existing servers and storage.

Figure 8-10 illustrates the EMC Atmos hardware appliance. The hardware appliance is comprised of servers (nodes) connected to standard disk enclosures. The rack includes a 24-port Gigabit Ethernet switch to provide internode communication. The Atmos software is installed on each node.

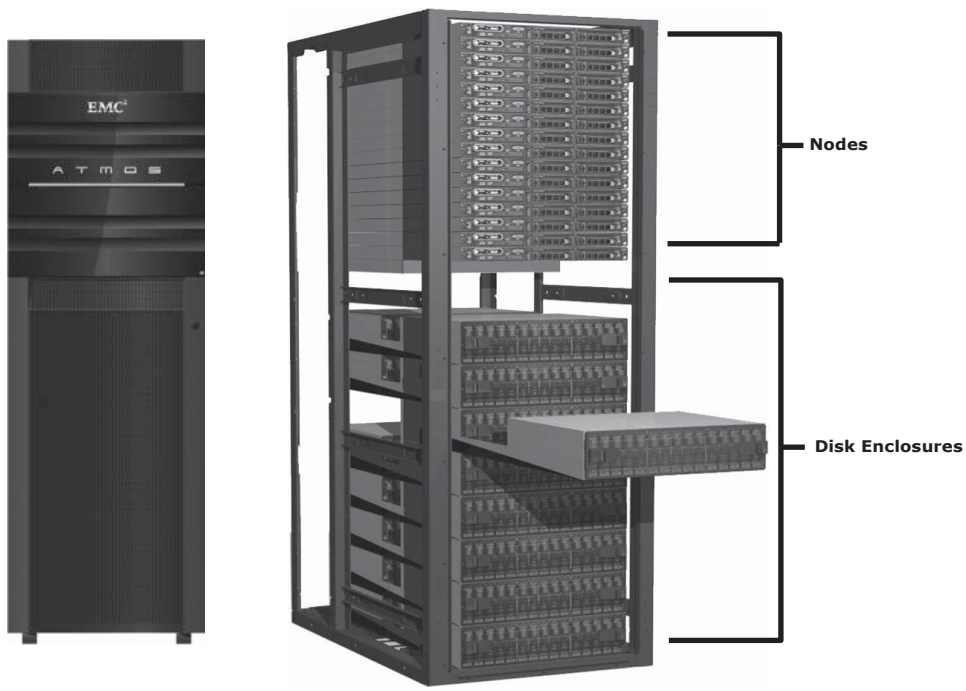


Figure 8-10: EMC Atmos storage system

Atmos VE enables users to exploit the power of Atmos in a virtualized environment. It can be deployed on a virtual machine in VMware ESXi hosts and configured with the VMware certified back-end storage.

Following are the key features offered by EMC Atmos:

- **Policy-based management:** EMC Atmos improves operational efficiency by automatically distributing content based on business policy. The administrator-defined policies dictate how, when, and where the information resides.

- **Protection:** Atmos offers two options to protect the objects, replication and Geo Parity:
  - *Replication* ensures that the content is available and accessible by creating redundant copies of an object at redundant designated locations.
  - *Geo Parity* ensures that the content is available and accessible by dividing objects into multiple segments plus parity segments and distributing them to one or more designated locations.
- **Data services:** EMC Atmos includes the data services, such as compression and deduplication. These features are native to Atmos and can be managed and accessed via a policy.
- **Web services and legacy protocols:** EMC Atmos provides flexible web services access (REST/SOAP) for web-scale applications and file access (CIFS/NFS/Installable File System/Centera API) for traditional applications.
- **Automated system management:** EMC Atmos provides auto-configuring, auto-managing, and auto-healing capabilities to reduce administration and downtime.
- **Multitenancy:** EMC Atmos enables multiple applications to be served from the same infrastructure. Each application is securely partitioned and cannot access the other application's data. Multitenancy is ideal for service providers or large enterprises that want to provide cloud computing services to multiple customers or departments allowing logical and secure separation within a single infrastructure.
- **Flexible administration:** EMC Atmos can be managed via a graphical user interface (GUI) or command-line interface (CLI).

EMC VNX

VNX is EMC's unified storage product offering. Figure 8-11 illustrates the EMC VNX storage array.

VNX storage systems include the following components:

- *Storage processors* (SPs) support block I/O access to storage with FC, iSCSI, and FCoE protocols.
- *X-Blades* access data from the back end and provide host access with NFS, CIFS, MPFS, pNFS, and FTP protocols. The X-Blades in each array are scalable and provide redundancy to ensure no single point of failure.
- *Control Stations* provide management functions to the X-Blades. The Control Station is also responsible for X-Blade failover. The Control Station may optionally be configured with a matching secondary Control Station to ensure management redundancy on the VNX array.

- *Standby power supplies* provide enough power to each storage processor and first DAE to ensure that any data in flight is stored in the vault area if a power failure occurs. This ensures that no writes are lost.
- *Disk-array enclosures* (DAEs) house the drives used in the array. Different sized DAEs are available that can each hold a maximum of 15, 25, or 60 drives. More DAEs can be added to meet growing storage demands.

Figure 8-11: EMC VNX storage system

EMC Centera

EMC Centera is offered in three different models to meet different types of user requirements — EMC Centera Basic, EMC Centera Governance Edition, and EMC Centera Compliance Edition Plus (CE+):

- **EMC Centera Basic:** Provides all functionalities without the enforcement of retention periods.

- **EMC Centera Governance Edition:** Provides the retention capabilities required by organizations to manage digital records in addition to the features provided by EMC Centera Basic.
- **EMC Centera Compliance Edition Plus:** Provides extensive compliance capabilities. CE+ is designed to meet the requirements of the most stringent regulated business environments for electronic storage media, as established by regulations from the Securities and Exchange Commission (SEC), or other national and international regulatory groups.

EMC Centera Architecture

The Centera architecture is shown in Figure 8-12. A client accesses the Centera over a LAN. The client can access Centera only through the server that runs the Centera API (application programming interface). The Centera API is responsible for performing functions that enable an application to store and retrieve the data.

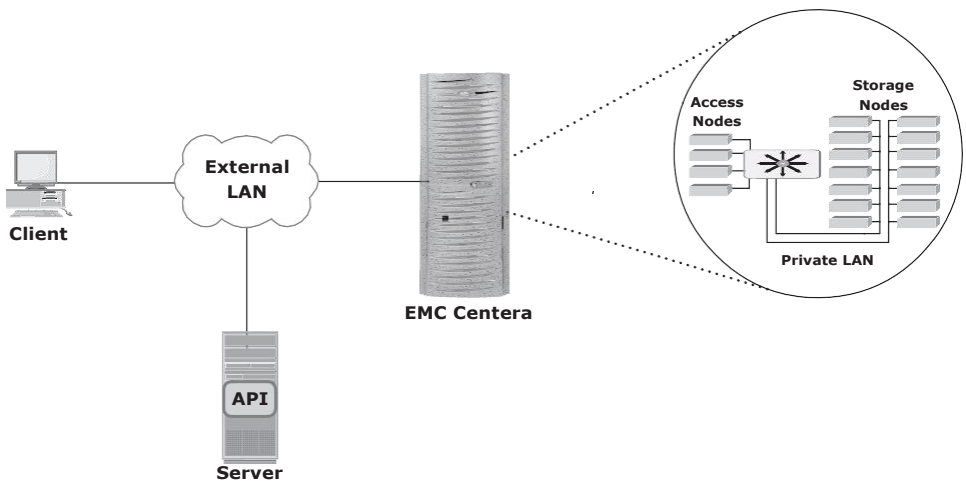


Figure 8-12: Centera architecture

Centera architecture is a *Redundant Array of Independent Nodes* (RAIN). It contains storage nodes and access nodes that are networked as a cluster by using a private LAN. The internal LAN reconfigures automatically when it detects configuration changes, such as the addition of storage or access nodes. The application server accesses the Centera via an external LAN.

The nodes are configured with low-cost, high-capacity SATA disk drives. These nodes run CentraStar, the operating environment for Centera, which provides the features and functionalities required in a Centera system.

When nodes are installed, they are configured with a "role" that defines the functionality provided to the node. A node can be configured as a storage node, an access node, or a dual-role node.

Storage nodes store and protect data objects. They are sometimes referred to as *back-end nodes*.

Access nodes provide connectivity to application servers through an external LAN. They establish connectivity with the storage nodes in the cluster through a private LAN. The number of access nodes is determined by the amount of throughput required from the cluster. If a node is configured solely as an "access node," its disk space cannot be used to store data objects. Storage and retrieval requests are sent to the access node via the external LAN.

Dual-role nodes provide both storage and access-node capabilities. This configuration is more common than a pure access-node configuration.

Module-3

Backup, Archive, and Replication

Information Availability

Information availability (IA) refers to the ability of an IT infrastructure to function according to business expectations during its specified time of operation. IA ensures that people (employees, customers, suppliers, and partners) can access information whenever they need it. IA can be defined in terms of accessibility, reliability, and timeliness of information.

- **Accessibility:** Information should be accessible at the right place, to the right user.
- **Reliability:** Information should be reliable and correct in all aspects. It is "the same" as what was stored, and there is no alteration or corruption to the information.
- **Timeliness:** Defines the exact moment or the time window (a particular time of the day, week, month, and year as specified) during which information must be accessible. For example, if online access to an application is required between 8:00 a.m. and 10:00 p.m. each day, any disruptions to data availability outside of this time slot are not considered to affect timeliness.

Causes of Information Unavailability

Various planned and unplanned incidents result in information unavailability. *Planned outages* include installation/integration/maintenance of new hardware, software upgrades or patches, taking backups, application and data restores, facility operations (renovation and construction), and refresh/migration of the testing to the production environment. *Unplanned outages* include failure caused by human errors, database corruption, and failure of physical and virtual components.

Another type of incident that may cause data unavailability is natural or man-made disasters, such as flood, fire, earthquake, and contamination. As illustrated in Figure 9-1, the majority of outages are planned. Planned outages are expected and scheduled but still cause data to be unavailable. Statistically, the cause of information unavailability due to unforeseen disasters is less than 1 percent.

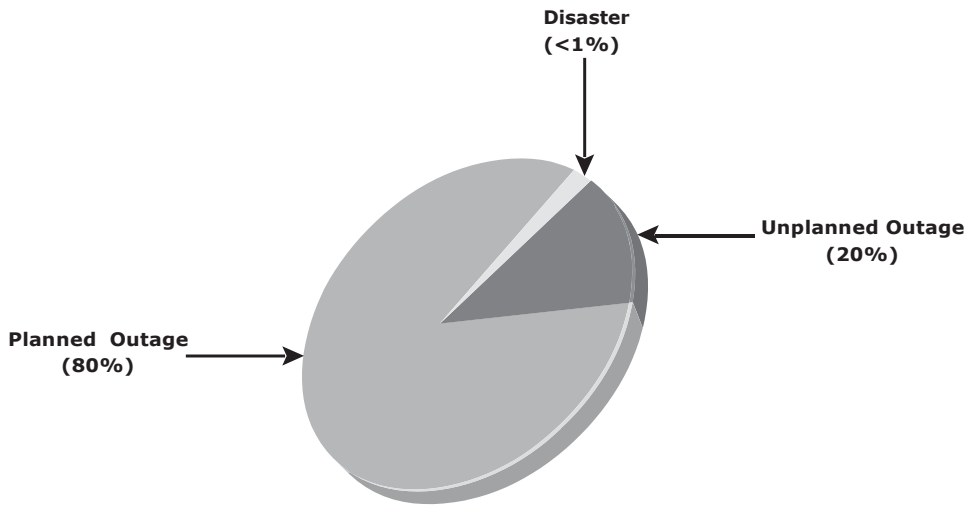


Figure 9-1: Disruptors of information availability

Consequences of Downtime

Information unavailability or downtime results in loss of productivity, loss of revenue, poor financial performance, and damage to reputation. Loss of productivity includes reduced output per unit of labor, equipment, and capital. Loss of revenue includes direct loss, compensatory payments, future revenue loss, billing loss, and investment loss. Poor financial performance affects revenue recognition, cash flow, discounts, payment guarantees, credit rating, and stock price. Damages to reputations may result in a loss of confidence or credibility with customers, suppliers, financial markets, banks, and business partners. Other possible consequences of downtime include the cost of additional equipment rental, overtime, and extra shipping.

The business impact of downtime is the sum of all losses sustained as a result of a given disruption. An important metric, *average cost of downtime per hour*, provides a key estimate in determining the appropriate BC solutions. It is calculated as follows:

Average cost of downtime per hour = average productivity loss per hour + average revenue loss per hour

Where:

Productivity loss per hour = (total salaries and benefits of all employees per week)/(average number of working hours per week)

Average revenue loss per hour = (total revenue of an organization per week)/(average number of hours per week that an organization is open for business)

The average downtime cost per hour may also include estimates of projected revenue loss due to other consequences, such as damaged reputations, and the additional cost of repairing the system.

Measuring Information Availability

IA relies on the availability of both physical and virtual components of a data center. Failure of these components might disrupt IA. A failure is the termination of a component's capability to perform a required function. The component's capability can be restored by performing an external corrective action, such as a manual reboot, repair, or replacement of the failed component(s). Repair involves restoring a component to a condition that enables it to perform a required function. Proactive risk analysis, performed as part of the BC planning process, considers the component failure rate and average repair time, which are measured by mean time between failure (MTBF) and mean time to repair (MTTR):

- **Mean Time Between Failure (MTBF):** It is the average time available for a system or component to perform its normal operations between failures. It is the measure of system or component reliability and is usually expressed in hours.
- **Mean Time To Repair (MTTR):** It is the average time required to repair a failed component. While calculating MTTR, it is assumed that the fault responsible for the failure is correctly identified and the required spares and personnel are available. A fault is a physical defect at the component level, which may result in information unavailability. MTTR includes the total time required to do the following activities: Detect the fault, mobilize the maintenance team, diagnose the fault, obtain the spare parts, repair, test, and restore the data. Figure 9-2 illustrates the various information availability metrics that represent system uptime and downtime.

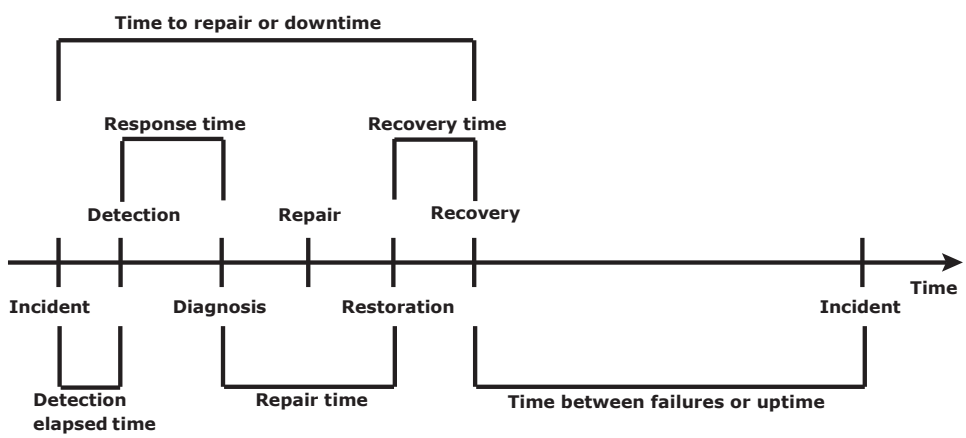


Figure 9-2: Information availability metrics

IA is the time period during which a system is in a condition to perform its intended function upon demand. It can be expressed in terms of system uptime and downtime and measured as the amount or percentage of system uptime:

IA = system uptime/(system uptime + system downtime)

Where *system uptime* is the period of time during which the system is in an accessible state; when it is not accessible, it is termed as *system downtime*. In terms of MTBF and MTTR, IA could also be expressed as

IA = MTBF/(MTBF + MTTR)

Uptime per year is based on the exact timeliness requirements of the service. This calculation leads to the number of "9s" representation for availability metrics. Table 9-1 lists the approximate amount of downtime allowed for a service to achieve certain levels of 9s availability.

For example, a service that is said to be "five 9s available" is available for 99.999 percent of the scheduled time in a year (24 × 365).

Table 9-1: Availability Percentage and Allowable Downtime

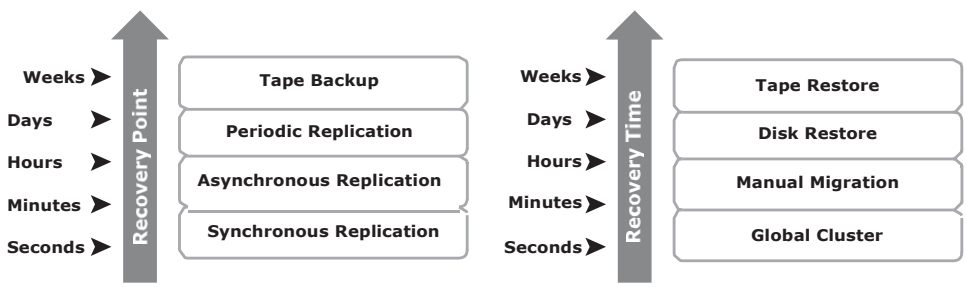
UPTIME (%)	DOWNTIME (%)	DOWNTIME PER YEAR	DOWNTIME PER WEEK
98	2	7.3 days	3 hr, 22 minutes
99	1	3.65 days	1 hr, 41 minutes
99.8	0.2	17 hr, 31 minutes	20 minutes, 10 secs
99.9	0.1	8 hr, 45 minutes	10 minutes, 5 secs
99.99	0.01	52.5 minutes	1 minute
99.999	0.001	5.25 minutes	6 secs
99.9999	0.0001	31.5 secs	0.6 secs

BC Terminology

This section introduces and defines common terms related to BC operations, which are used in the next few chapters to explain advanced concepts:

- **Disaster recovery:** This is the coordinated process of restoring systems, data, and the infrastructure required to support ongoing business operations after a disaster occurs. It is the process of restoring a previous copy of the data and applying logs or other necessary processes to that copy to bring it to a known point of consistency. After all recovery efforts are completed, the data is validated to ensure that it is correct.

- **Disaster restart:** This is the process of restarting business operations with mirrored consistent copies of data and applications.
- **Recovery-Point Objective (RPO):** This is the point in time to which systems and data must be recovered after an outage. It defines the amount of data loss that a business can endure. A large RPO signifies high tolerance to information loss in a business. Based on the RPO, organizations plan for the frequency with which a backup or replica must be made. For example, if the RPO is 6 hours, backups or replicas must be made at least once in 6 hours. Figure 9-3 (a) shows various RPOs and their corresponding ideal recovery strategies. An organization can plan for an appropriate BC technology solution on the basis of the RPO it sets. For example:
  - **RPO of 24 hours:** Backups are created at an offsite tape library every midnight. The corresponding recovery strategy is to restore data from the set of last backup tapes.
  - **RPO of 1 hour:** Shipping database logs to the remote site every hour. The corresponding recovery strategy is to recover the database to the point of the last log shipment.
  - **RPO in the order of minutes:** Mirroring data asynchronously to a remote site
  - **Near zero RPO:** Mirroring data synchronously to a remote site



(a) Recovery-point objective (b) Recovery-time objective

Figure 9-3: Strategies to meet RPO and RTO targets

- **Recovery-Time Objective (RTO):** The time within which systems and applications must be recovered after an outage. It defines the amount of downtime that a business can endure and survive. Businesses can optimize disaster recovery plans after defining the RTO for a given system. For example, if the RTO is 2 hours, it requires disk-based backup because it enables a faster restore than a tape backup. However, for an RTO of 1 week, tape backup will likely meet the requirements. Some examples

of RTOs and the recovery strategies to ensure data availability are listed here (refer to Figure 9-3 [b]):

- **RTO of 72 hours:** Restore from tapes available at a cold site.
- **RTO of 12 hours:** Restore from tapes available at a hot site.
- **RTO of few hours:** Use of data vault at a hot site
- **RTO of a few seconds:** Cluster production servers with bidirectional mirroring, enabling the applications to run at both sites simultaneously.

- **Data vault:** A repository at a remote site where data can be periodically or continuously copied (either to tape drives or disks) so that there is always a copy at another site
- **Hot site:** A site where an enterprise's operations can be moved in the event of disaster. It is a site with the required hardware, operating system, application, and network support to perform business operations, where the equipment is available and running at all times.
- **Cold site:** A site where an enterprise's operations can be moved in the event of disaster, with minimum IT infrastructure and environmental facilities in place, but not activated
- **Server Clustering:** A group of servers and other necessary resources coupled to operate as a single system. Clusters can ensure high availability and load balancing. Typically, in failover clusters, one server runs an application and updates the data, and another server is kept as standby to take over completely, as required. In more sophisticated clusters, multiple servers may access data, and typically one server is kept as standby. Server clustering provides load balancing by distributing the application load evenly among multiple servers within the cluster.

BC Planning Life Cycle

BC planning must follow a disciplined approach like any other planning process. Organizations today dedicate specialized resources to develop and maintain BC plans. From the conceptualization to the realization of the BC plan, a life cycle of activities can be defined for the BC process. The BC planning life cycle includes five stages (see Figure 9-4):

1. Establishing objectives
2. Analyzing
3. Designing and developing
4. Implementing
5. Training, testing, assessing, and maintaining

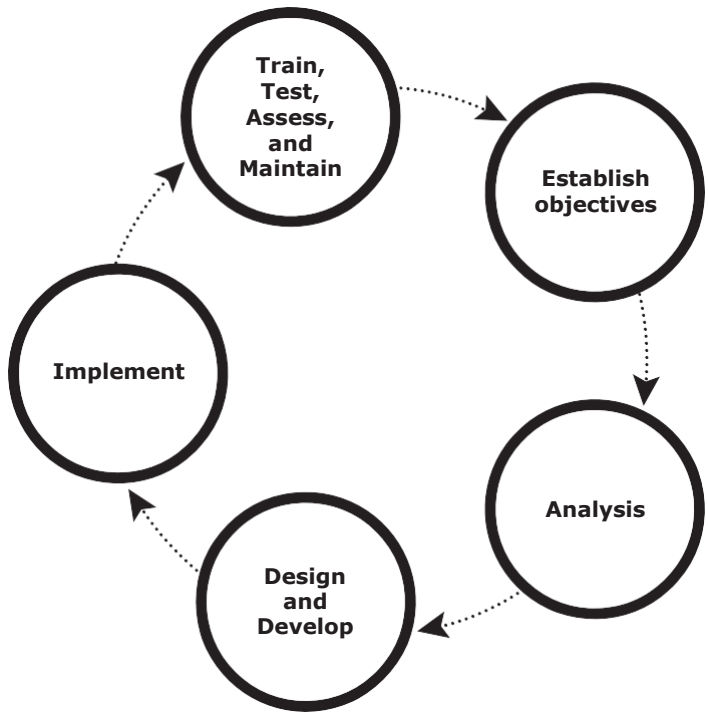


Figure 9-4: BC planning life cycle

Several activities are performed at each stage of the BC planning life cycle, including the following key activities:

1. Establish objectives:
   - Determine BC requirements.
   - Estimate the scope and budget to achieve requirements.
   - Select a BC team that includes subject matter experts from all areas of the business, whether internal or external.
   - Create BC policies.
2. Analysis:
   - Collect information on data profiles, business processes, infrastructure support, dependencies, and frequency of using business infrastructure.
   - Conduct a Business Impact Analysis (BIA).
   - Identify critical business processes and assign recovery priorities.
   - Perform risk analysis for critical functions and create mitigation strategies.

- Perform cost benefit analysis for available solutions based on the mitigation strategy.
- Evaluate options.

3. Design and develop:
   - Define the team structure and assign individual roles and responsibilities. For example, different teams are formed for activities, such as emergency response, damage assessment, and infrastructure and application recovery.
   - Design data protection strategies and develop infrastructure.
   - Develop contingency solutions.
   - Develop emergency response procedures.
   - Detail recovery and restart procedures.

4. Implement:
   - Implement risk management and mitigation procedures that include backup, replication, and management of resources.
   - Prepare the disaster recovery sites that can be utilized if a disaster affects the primary data center.
   - Implement redundancy for every resource in a data center to avoid single points of failure.

5. Train, test, assess, and maintain:
   - Train the employees who are responsible for backup and replication of business-critical data on a regular basis or whenever there is a modification in the BC plan.
   - Train employees on emergency response procedures when disasters are declared.
   - Train the recovery team on recovery procedures based on contingency scenarios.
   - Perform damage-assessment processes and review recovery plans.
   - Test the BC plan regularly to evaluate its performance and identify its limitations.
   - Assess the performance reports and identify limitations.
   - Update the BC plans and recovery/restart procedures to reflect regular changes within the data center.

Failure Analysis

Failure analysis involves analyzing both the physical and virtual infrastructure components to identify systems that are susceptible to a single point of failure and implementing fault-tolerance mechanisms.

Single Point of Failure

A *single point of failure* refers to the failure of a component that can terminate the availability of the entire system or IT service. Figure 9-5 depicts a system setup in which an application, running on a VM, provides an interface to the client and performs I/O operations. The client is connected to the server through an IP network, and the server is connected to the storage array through an FC connection.

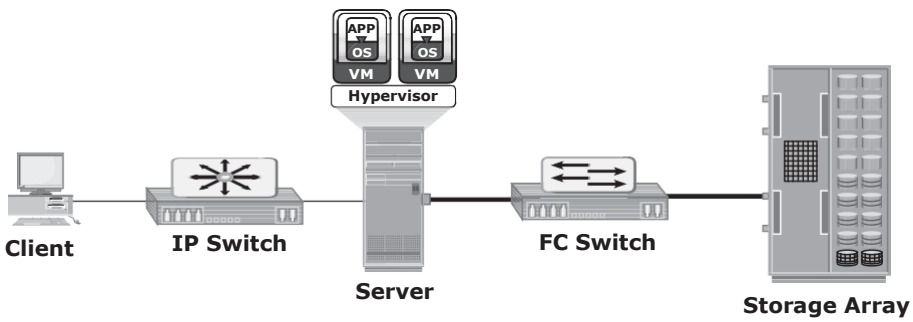


Figure 9-5: Single point of failure

In a setup in which each component must function as required to ensure data availability, the failure of a single physical or virtual component causes the unavailability of an application. This failure results in disruption of business operations. For example, failure of a hypervisor can affect all the running VMs and the virtual network, which are hosted on it. In the setup shown in Figure 9-5, several single points of failure can be identified. A VM, a hypervisor, an HBA/NIC on the server, the physical server, the IP network, the FC switch, the storage array ports, or even the storage array could be a potential single point of failure.

Resolving Single Points of Failure

To mitigate single points of failure, systems are designed with redundancy, such that the system fails only if all the components in the redundancy group fail. This ensures that the failure of a single component does not affect data availability. Data centers follow stringent guidelines to implement fault tolerance for uninterrupted information availability. Careful analysis is performed to eliminate every single point of failure. The example shown in Figure 9-6 represents all enhancements in the infrastructure to mitigate single points of failure:

- Configuration of redundant HBAs at a server to mitigate single HBA failure
- Configuration of NIC teaming at a server allows protection against single physical NIC failure. It allows grouping of two or more physical NICs and treating them as a single logical device. With NIC teaming, if one of the underlying physical NICs fails or its cable is unplugged, the traffic is redirected to another physical NIC in the team. Thus, NIC teaming eliminates the single point of failure associated with a single physical NIC.
- Configuration of redundant switches to account for a switch failure
- Configuration of multiple storage array ports to mitigate a port failure
- RAID and hot spare configuration to ensure continuous operation in the event of disk failure
- Implementation of a redundant storage array at a remote site to mitigate local site failure
- Implementing server (or compute) clustering, a fault-tolerance mechanism whereby two or more servers in a cluster access the same set of data volumes. Clustered servers exchange a *heartbeat* to inform each other about their health. If one of the servers or hypervisors fails, the other server or hypervisor can take up the workload.
- Implementing a VM Fault Tolerance mechanism ensures BC in the event of a server failure. This technique creates duplicate copies of each VM on another server so that when a VM failure is detected, the duplicate VM can be used for failover. The two VMs are kept in synchronization with each other in order to perform successful failover.

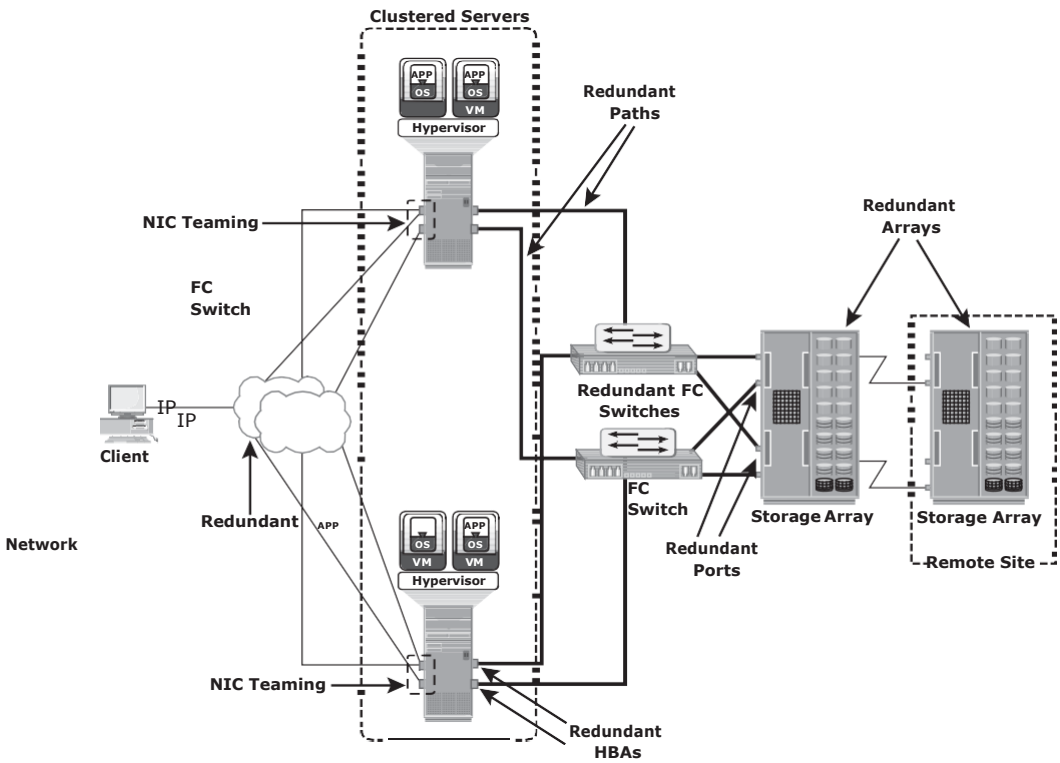


Figure 9-6: Resolving single points of failure

Multipathing Software

Configuration of multiple paths increases the data availability through path failover. If servers are configured with one I/O path to the data, there will be no access to the data if that path fails. Redundant paths to the data eliminate the possibility of the path becoming a single point of failure. Multiple paths to data also improve I/O performance through load balancing among the paths and maximize server, storage, and data path utilization.

In practice, merely configuring multiple paths does not serve the purpose. Even with multiple paths, if one path fails, I/O does not reroute unless the system recognizes that it has an alternative path. Multipathing software provides the functionality to recognize and utilize alternative I/O paths to data. Multipathing software also manages the load balancing by distributing I/Os to all available, active paths. Multipathing software intelligently manages the paths to a device by sending I/O down the optimal path based on the load balancing and failover policy setting for the device. It also takes into account path usage and availability before deciding the path through which to send the I/O. If a path to the device fails, it automatically reroutes the I/O to an alternative path.

In a virtual environment, multipathing is enabled either by using the hypervisor's built-in capability or by running a third-party software module, added to the hypervisor.

Business Impact Analysis

A *business impact analysis* (BIA) identifies which business units, operations, and processes are essential to the survival of the business. It evaluates the financial, operational, and service impacts of a disruption to essential business processes. Selected functional areas are evaluated to determine resilience of the infrastructure to support information availability. The BIA process leads to a report detailing the incidents and their impact over business functions. The impact may be specified in terms of money or in terms of time. Based on the potential impacts associated with downtime, businesses can prioritize and implement countermeasures to mitigate the likelihood of such disruptions. These are detailed in the BC plan. A BIA includes the following set of tasks:

- Determine the business areas.
- For each business area, identify the key business processes critical to its operation.
- Determine the attributes of the business process in terms of applications, databases, and hardware and software requirements.
- Estimate the costs of failure for each business process.
- Calculate the maximum tolerable outage and define RTO and RPO for each business process.
- Establish the minimum resources required for the operation of business processes.
- Determine recovery strategies and the cost for implementing them.
- Optimize the backup and business recovery strategy based on business priorities.
- Analyze the current state of BC readiness and optimize future BC planning.

BC Technology Solutions

After analyzing the business impact of an outage, designing the appropriate solutions to recover from a failure is the next important activity. One or more copies of the data are maintained using any of the following strategies so that

data can be recovered or business operations can be restarted using an alternative copy:

- **Backup:** Data backup is a predominant method of ensuring data availability. The frequency of backup is determined based on RPO, RTO, and the frequency of data changes.
- **Local replication:** Data can be replicated to a separate location within the same storage array. The replica is used independently for other business operations. Replicas can also be used for restoring operations if data corruption occurs.
- **Remote replication:** Data in a storage array can be replicated to another storage array located at a remote site. If the storage array is lost due to a disaster, business operations can be started from the remote storage array.

Concept in Practice: EMC PowerPath

EMC PowerPath is host-based multipathing software that provides path failover and load-balancing functionality for SAN environments. PowerPath resides between the operating system and device drivers. EMC PowerPath/VE software allows optimizing virtual environments with PowerPath multipathing features.

Refer to `www.emc.com` for the latest information.

PowerPath Features

PowerPath provides the following features:

- **Dynamic path configuration and management:** PowerPath provides the flexibility to define some paths to a device as "active" and some as "standby." The standby paths are used when all active paths to a logical device have failed. Paths can be dynamically added and removed by setting them in standby or active mode.
- **Dynamic load balancing across multiple paths:** PowerPath intelligently distributes I/O requests across all available paths to the logical storage device. This reduces path bottlenecks and improves application performance.
- **Automatic path failover:** In the event of a path failure, PowerPath fails over seamlessly to an alternative path without disrupting application operations. PowerPath redistributes I/O to the best available path to achieve optimal host performance.
- **Proactive path testing and automatic path recovery:** PowerPath uses the autoprobe and autorestore functions to proactively test the dead

and restored paths, respectively. The PowerPath *autoprobe* function periodically probes all the paths to check failed paths before sending the application I/O. This process enables PowerPath to proactively close paths before an application experiences a timeout when sending I/O over failed paths. The PowerPath *autorestore* function runs every 5 minutes and tests every failed or closed path to determine whether it has been restored.

- **Cluster support:** The deployment of PowerPath in a server cluster eliminates invoking cluster failover due to a path failure.

Dynamic Load Balancing

PowerPath provides significant performance improvement in environments where the I/O workload is not balanced. For every I/O, the PowerPath filter driver selects the path based on the load-balancing policy and failover setting for the logical storage device. The driver identifies all available paths to a device and builds a routing table, called a volume path set, for the devices. PowerPath supports certain user-specified load-balancing policies such as the following:

- **Round-Robin policy:** I/O requests are assigned to each available path in rotation.
- **Least I/Os policy:** I/O requests are routed to the path with the fewest queued I/O requests, regardless of the total number of I/O blocks.
- **Least Blocks policy:** I/O requests are routed to the path with the fewest queued I/O blocks, regardless of the number of requests involved.
- **Priority-Based policy:** I/O requests are balanced across multiple paths based on the composition of reads, writes, user-assigned devices, or application priorities.

I/O Operation without PowerPath

Figure 9-7 illustrates I/O operations in a storage system in the absence of PowerPath. The applications running on a host have four paths to the storage array. This example illustrates how I/O throughput is unbalanced without PowerPath. Two paths get high I/O traffic and are highly loaded, whereas the other two paths are less loaded. As a result, applications cannot achieve optimal performance.

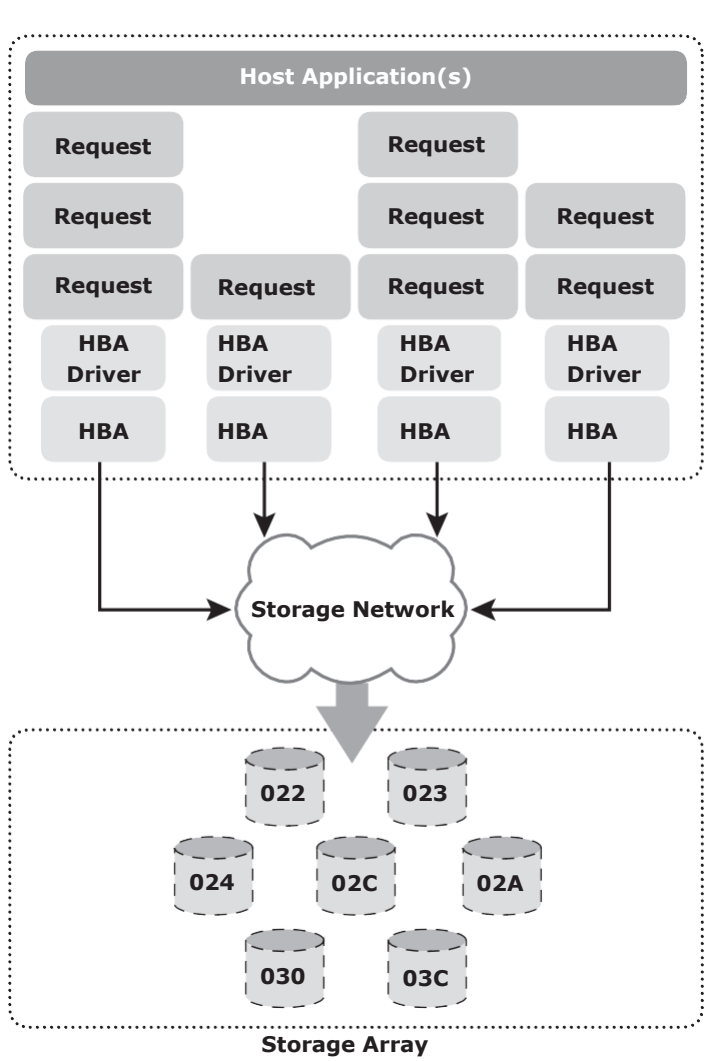


Figure 9-7: I/O without PowerPath

I/O Operation with PowerPath

Figure 9-8 shows I/O operations in a storage system environment that has PowerPath. PowerPath ensures that I/O requests are balanced across all the paths to storage, based on the load-balancing algorithm chosen. As a result, the applications can effectively utilize all the paths, thereby improving their performance.

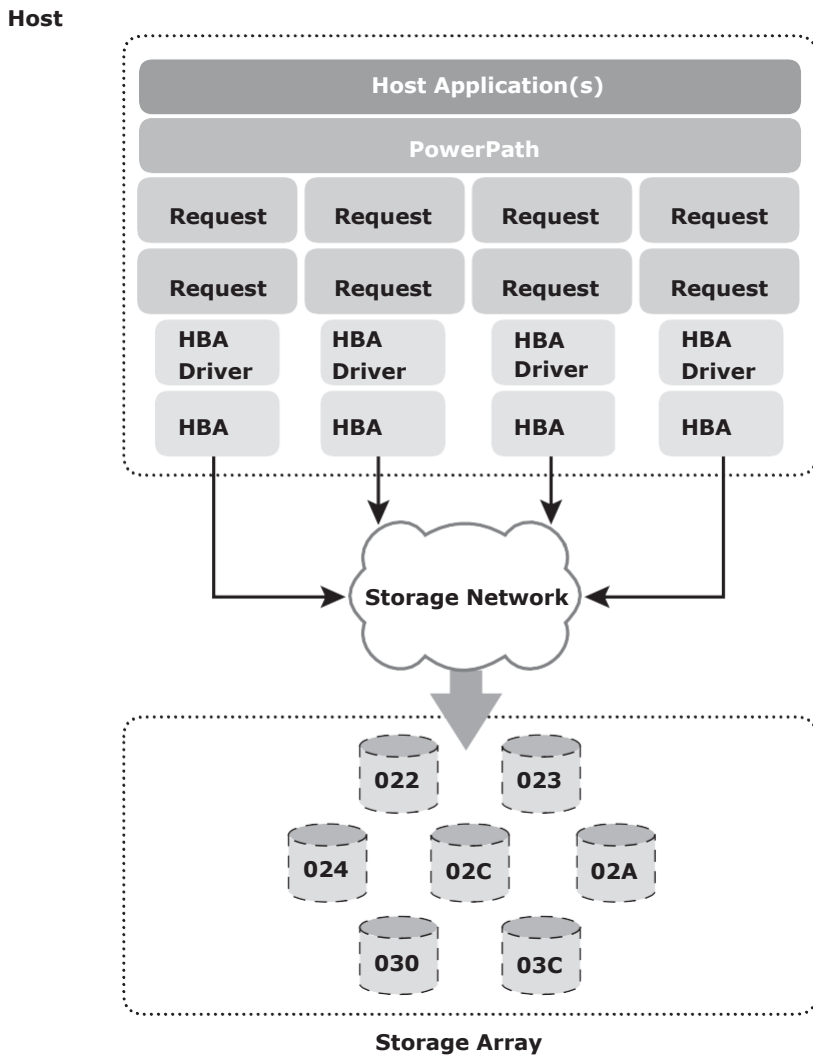


Figure 9-8: I/O with PowerPath

Automatic Path Failover

The next two examples demonstrate how PowerPath performs path failover operations if a path failure occurs for active-active and active-passive array configurations.

Path Failure without PowerPath

Figure 9-9 shows a scenario without PowerPath. The loss of a path (the path failure is marked by a cross "X") due to single points of failure, such as the loss of an HBA, storage array front-end connectivity, switch port, or a failed cable, can result in an outage for one or more applications that use that path.

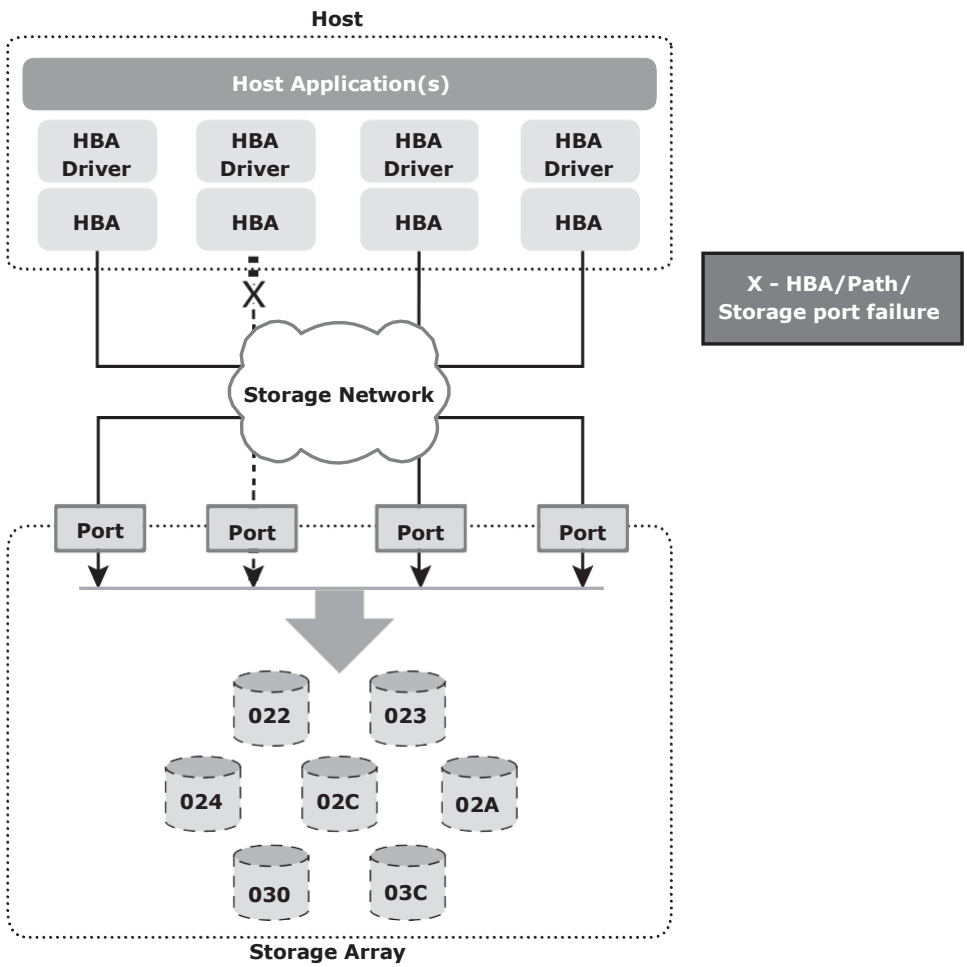


Figure 9-9: Path failure without PowerPath

Path Failover with PowerPath: Active-Active Array

Figure 9-10 shows a storage system environment in which an application uses PowerPath with an active-active array configuration to perform I/O operations. In an active-active storage array, if multiple paths to a logical device exist, they

all are active and provide access to the device. If a path to the device fails, PowerPath redirects the application I/Os through an alternative active path therefore preventing any application outage.

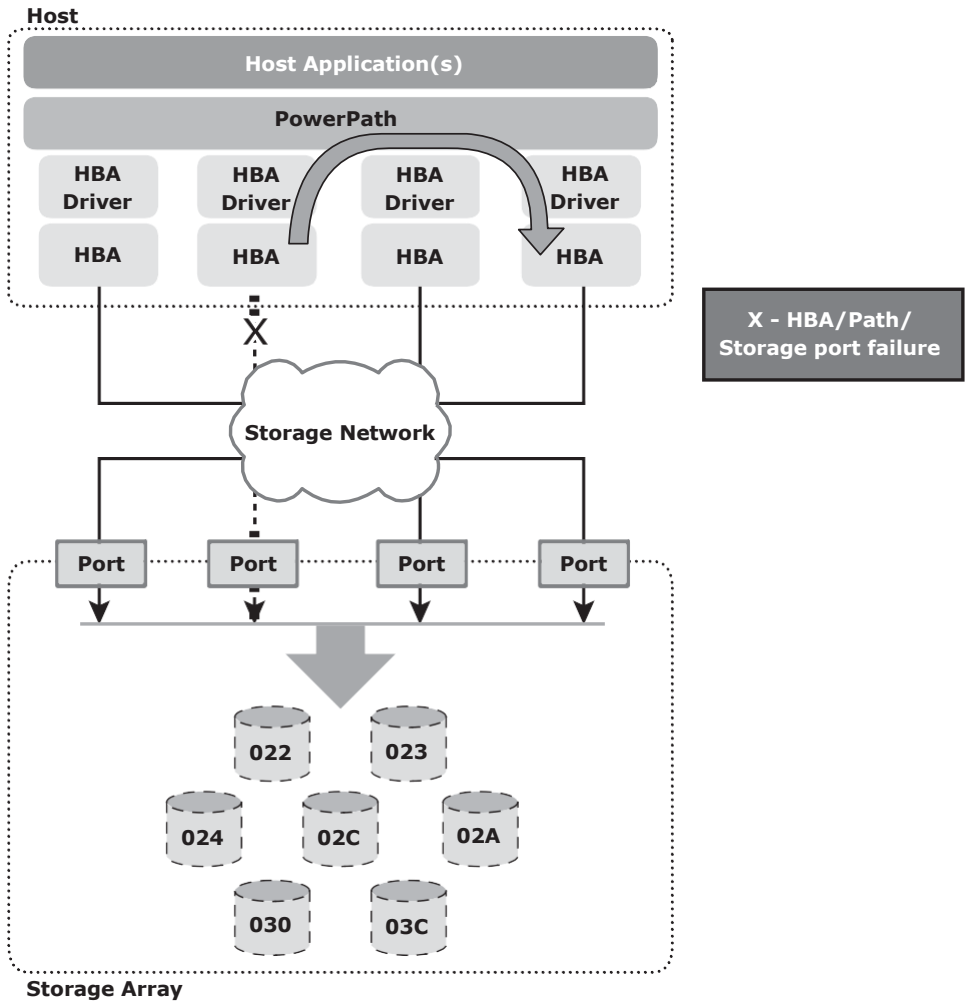


Figure 9-10: Path failover with PowerPath for an active-active array

Path Failover with PowerPath: Active-Passive Array

Figure 9-11 shows a scenario in which a logical device is assigned to a storage processor B (SP B) and therefore, all I/Os are directed down the path through SP B to the device. The logical device can also be accessed through SP A but only after SP B is unavailable and the device is re-assigned to SP A.

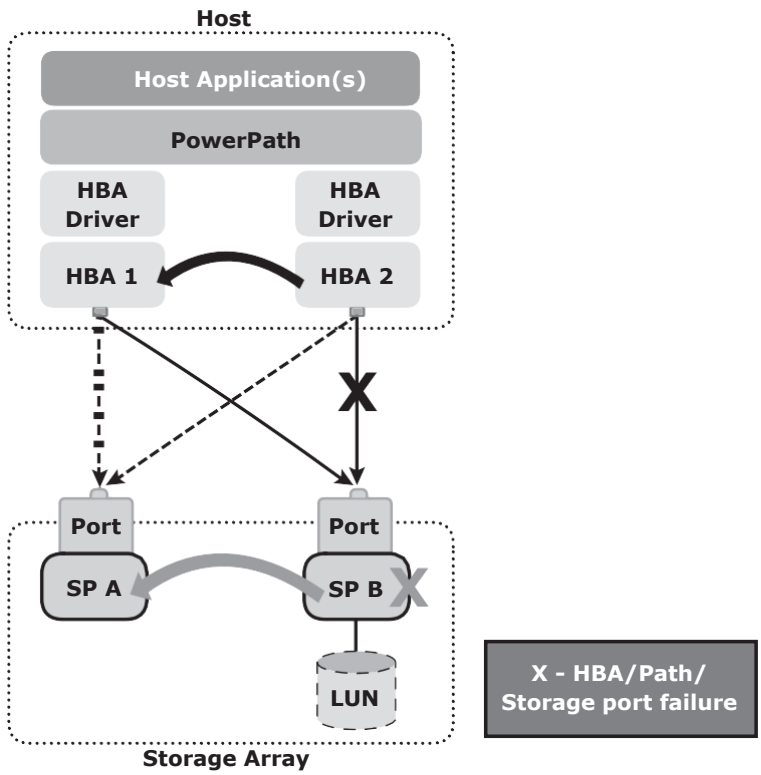


Figure 9-11: Path failover with PowerPath for an active-passive array

Path failure can occur due to a failure of the link, HBA, or storage processor (SP). If a path failure occurs, PowerPath with an active-passive configuration performs the path failover operation in the following way:

- If an I/O path to SP B either through HBA 2 or through HBA 1 fails, PowerPath uses the remaining available path to SP B to send all the I/Os.
- If SP B fails, PowerPath stops all I/O to SP B and *trespasses* the device over to SP A. All I/O is sent down the paths to SP A (paths which were previously standby but are now active for the given LUN). This process is referred as *LUN trespassing*. When SP B is brought back online, PowerPath recognizes that it is available and resumes sending I/O down to SP B after the LUN has been trespassed back to SP B.

Backup Purpose

Backups are performed to serve three purposes: disaster recovery, operational recovery, and archival. These are covered in the following sections.

Disaster Recovery

One purpose of backups is to address disaster recovery needs. The backup copies are used for restoring data at an alternate site when the primary site is incapacitated due to a disaster. Based on recovery-point objective (RPO) and recovery-time objective (RTO) requirements, organizations use different data protection strategies for disaster recovery. When tape-based backup is used as a disaster recovery option, the backup tape media is shipped and stored at an offsite location. Later, these tapes can be recalled for restoration at the disaster recovery site. Organizations with stringent RPO and RTO requirements use remote replication technology to replicate data to a disaster recovery site. This allows organizations to bring production systems online in a relatively short period of time if a disaster occurs. Remote replication is covered in detail in Chapter 12.

Operational Recovery

Data in the production environment changes with every business transaction and operation. Backups are used to restore data if data loss or logical corruption occurs during routine processing. The majority of restore requests in most organizations fall in this category. For example, it is common for a user to accidentally delete an important e-mail or for a file to become corrupted, which can be restored using backup data.

Archival

Backups are also performed to address archival requirements. Although content addressed storage (CAS) has emerged as the primary solution for archives (CAS is discussed in Chapter 8), traditional backups are still used by small and medium enterprises for long-term preservation of transaction records, e-mail messages, and other business records required for regulatory compliance.

Backup Considerations

The amount of data loss and downtime that a business can endure in terms of RPO and RTO are the primary considerations in selecting and implementing a specific backup strategy. RPO refers to the point in time to which data must be recovered, and the point in time from which to restart business operations. This specifies the time interval between two backups. In other words, the RPO determines backup frequency. For example, if an application requires an RPO of 1 day, it would need the data to be backed up at least once every day. Another consideration is the retention period, which defines the duration for which a business needs to retain the backup copies. Some data is retained for years and some only for a few days. For example, data backed up for archival is retained for a longer period than data backed up for operational recovery.

The backup media type or backup target is another consideration, that is driven by RTO and impacts the data recovery time. The time-consuming operation of starting and stopping in a tape-based system affects the backup performance, especially while backing up a large number of small files.

Organizations must also consider the granularity of backups, explained later in section "10.3 Backup Granularity." The development of a backup strategy must include a decision about the most appropriate time for performing a backup to minimize any disruption to production operations. The location, size, number of files, and data compression should also be considered because they might affect the backup process. Location is an important consideration for the data to be backed up. Many organizations have dozens of heterogeneous platforms locally and remotely supporting their business. Consider a data warehouse environment that uses the backup data from many sources. The backup process must address these sources for transactional and content integrity. This process must be coordinated with all heterogeneous platforms at all locations on which the data resides.

The file size and number of files also influence the backup process. Backing up large-size files (for example, ten 1 MB files) takes less time, compared to backing up an equal amount of data composed of small-size files (for example, ten thousand 1 KB files).

Data compression and data deduplication (discussed later in section "10.11 Data Deduplication for Backup") are widely used in the backup environment because these technologies save space on the media. Many backup devices have built-in support for hardware-based data compression. Some data, such as application binaries, do not compress well, whereas text data does compress well.

Backup Granularity

Backup granularity depends on business needs and the required RTO/RPO. Based on the granularity, backups can be categorized as full, incremental and cumulative (differential). Most organizations use a combination of these three backup types to meet their backup and recovery requirements. Figure 10-1 shows the different backup granularity levels.

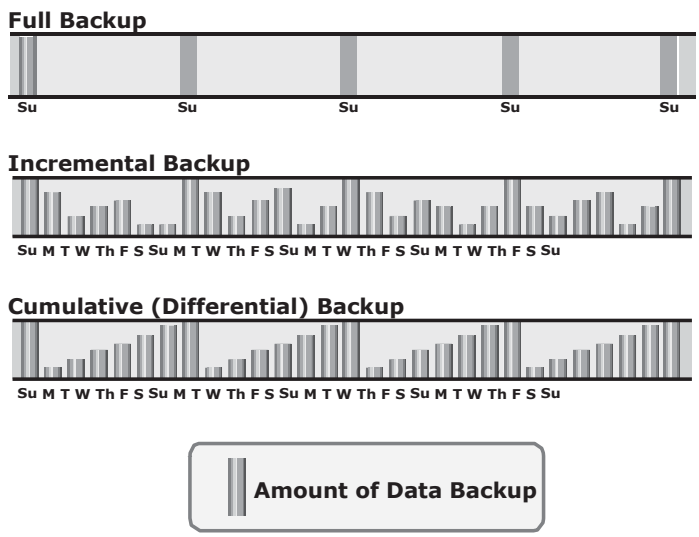


Figure 10-1: Backup granularity levels

Full backup is a backup of the complete data on the production volumes. A full backup copy is created by copying the data in the production volumes to a backup storage device. It provides a faster recovery but requires more storage space and also takes more time to back up. *Incremental backup* copies the data that has changed since the last full or incremental backup, whichever has occurred more recently. This is much faster than a full backup (because the volume of data backed up is restricted to the changed data only) but takes longer to restore. *Cumulative backup* copies the data that has changed since the last full backup. This method takes longer than an incremental backup but is faster to restore.

Restore operations vary with the granularity of the backup. A full backup provides a single repository from which the data can be easily restored. The process of restoration from an incremental backup requires the last full backup and all the incremental backups available until the point of restoration. A restore from a cumulative backup requires the last full backup and the most recent cumulative backup.

Figure 10-2 shows an example of restoring data from incremental backup.

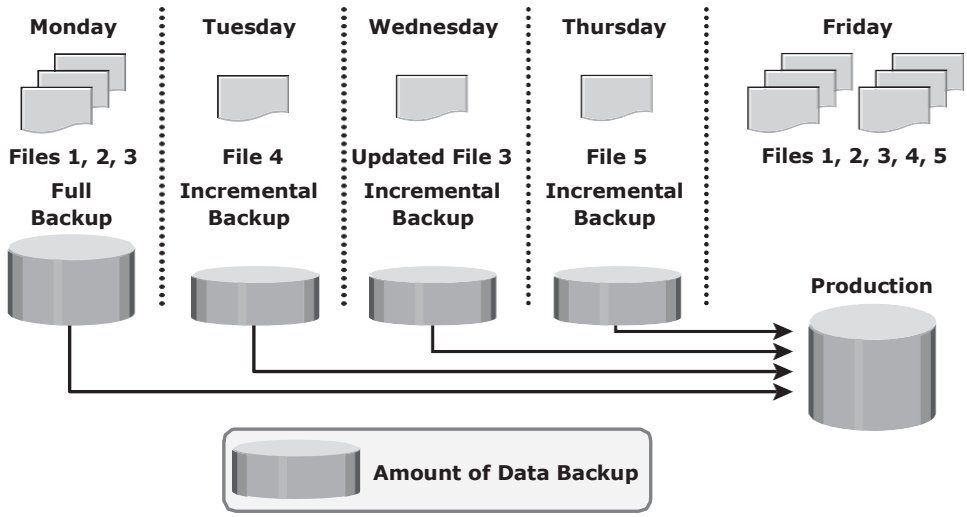


Figure 10-2: Restoring from an incremental backup

In this example, a full backup is performed on Monday evening. Each day after that, an incremental backup is performed. On Tuesday, a new file (File 4 in the figure) is added, and no other files have changed. Consequently, only File

4 is copied during the incremental backup performed on Tuesday evening. On Wednesday, no new files are added, but File 3 has been modified. Therefore, only the modified File 3 is copied during the incremental backup on Wednesday evening. Similarly, the incremental backup on Thursday copies only File 5. On Friday morning, there is data corruption, which requires data restoration from the backup. The first step toward data restoration is restoring all data from the full backup of Monday evening. The next step is applying the incremental backups of Tuesday, Wednesday, and Thursday. In this manner, data can be successfully recovered to its previous state, as it existed on Thursday evening.

Figure 10-3 shows an example of restoring data from cumulative backup.

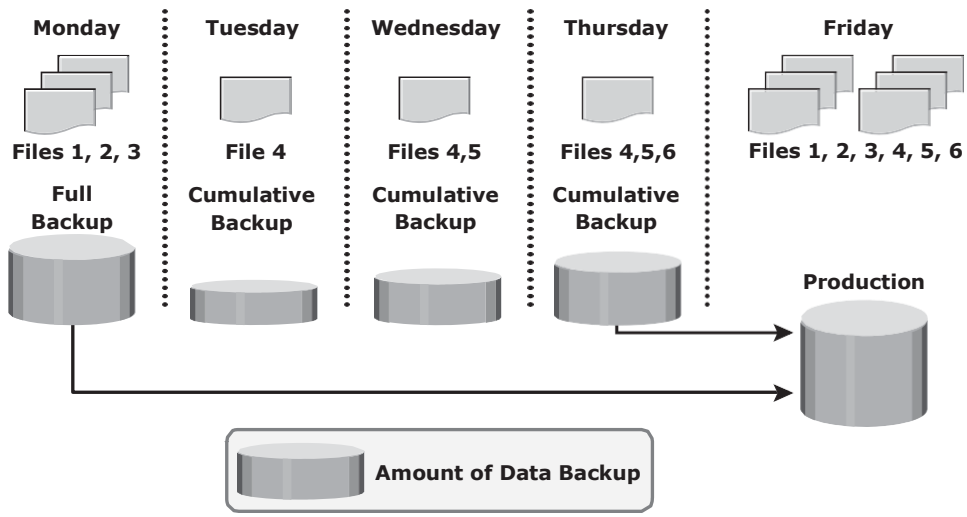


Figure 10-3: Restoring a cumulative backup

In this example, a full backup of the business data is taken on Monday evening. Each day after that, a cumulative backup is taken. On Tuesday, File 4 is added and no other data is modified since the previous full backup of Monday evening. Consequently, the cumulative backup on Tuesday evening copies only File 4. On Wednesday, File 5 is added. The cumulative backup taking place on Wednesday evening copies both File 4 and File 5 because these files have been added or modified since the last full backup. Similarly, on Thursday, File 6 is added. Therefore, the cumulative backup on Thursday evening copies all three files: File 4, File 5, and File 6. On Friday morning, data corruption occurs that requires data restoration using backup copies. The first step in restoring data is to restore all the data from the full backup of Monday evening. The next step is to apply only the latest cumulative backup, which is taken on Thursday evening. In this way, the production data can be recovered faster because its needs only two copies of data — the last full backup and the latest cumulative backup.

Recovery Considerations

The retention period is a key consideration for recovery. The retention period for a backup is derived from an RPO. For example, users of an application might request to restore the application data from its backup copy, which was created a month ago. This determines the retention period for the backup. Therefore, the minimum retention period of this application data is one month. However, the organization might choose to retain the backup for a longer period of time because of internal policies or external factors, such as regulatory directives.

If the recovery point is older than the retention period, it might not be possible to recover all the data required for the requested recovery point. Long retention periods can be defined for all backups, making it possible to meet any RPO within the defined retention periods. However, this requires a large storage space, which translates into higher cost. Therefore, while defining the retention period, analyze all the restore requests in the past and the allocated budget.

RTO relates to the time taken by the recovery process. To meet the defined RTO, the business may choose the appropriate backup granularity to minimize recovery time. In a backup environment, RTO influences the type of backup media that should be used. For example, a restore from tapes takes longer to complete than a restore from disks.

Backup Methods

Hot backup and cold backup are the two methods deployed for a backup. They are based on the state of the application when the backup is performed. In a *hot backup*, the application is up-and-running, with users accessing their data during the backup process. This method of backup is also referred to as an *online backup*. A *cold backup* requires the application to be shut down during the backup process. Hence, this method is also referred to as an *offline backup*. The hot backup of online production data is challenging because data is actively used and changed. If a file is open, it is normally not backed up during the backup process. In such situations, an *open file agent* is required to back up the open file. These agents interact directly with the operating system or application and enable the creation of consistent copies of open files. In database environments, the use of open file agents is not enough, because the agent should also support a consistent backup of all the database components. For example, a database is composed of many files of varying sizes occupying several file systems. To ensure a consistent database backup, all files need to be backed up in the same state. That does not necessarily mean that all files need to be backed up at the same time, but they all must be synchronized so that the database can be restored with consistency. The disadvantage associated with a hot backup is that the agents usually affect the overall application performance.

Consistent backups of databases can also be done by using a cold backup. This requires the database to remain inactive during the backup. Of course, the disadvantage of a cold backup is that the database is inaccessible to users during the backup process.

A *point-in-time* (PIT) copy method is deployed in environments in which the impact of downtime from a cold backup or the performance impact resulting from a hot backup is unacceptable. The PIT copy is created from the production volume and used as the source for the backup. This reduces the impact on the production volume. This technique is detailed in Chapter 11.

To ensure consistency, it is not enough to back up only the production data for recovery. Certain attributes and properties attached to a file, such as permissions, owner, and other metadata, also need to be backed up. These attributes are as important as the data itself and must be backed up for consistency.

In a disaster recovery environment, *bare-metal recovery* (BMR) refers to a backup in which all metadata, system information, and application configurations are appropriately backed up for a full system recovery. BMR builds the base system, which includes partitioning, the file system layout, the operating system, the applications, and all the relevant configurations. BMR recovers the base system first before starting the recovery of data files. Some BMR technologies — for example server configuration backup (SCB) — can recover a server even onto dissimilar hardware.

Backup Architecture

A backup system commonly uses the client-server architecture with a backup server and multiple backup clients. Figure 10-4 illustrates the backup architecture. The backup server manages the backup operations and maintains the backup catalog, which contains information about the backup configuration and backup metadata. Backup configuration contains information about when to run backups, which client data to be backed up, and so on, and the backup metadata contains information about the backed up data. The role of a backup client is to gather the data that is to be backed up and send it to the storage node. It also sends the tracking information to the backup server.

The storage node is responsible for writing the data to the backup device. (In a backup environment, a *storage node* is a host that controls backup devices.) The storage node also sends tracking information to the backup server. In many cases, the storage node is integrated with the backup server, and both are hosted on the same physical platform. A backup device is attached directly or through a network to the storage node's host platform. Some backup architecture refers to the storage node as the *media server* because it manages the storage device.

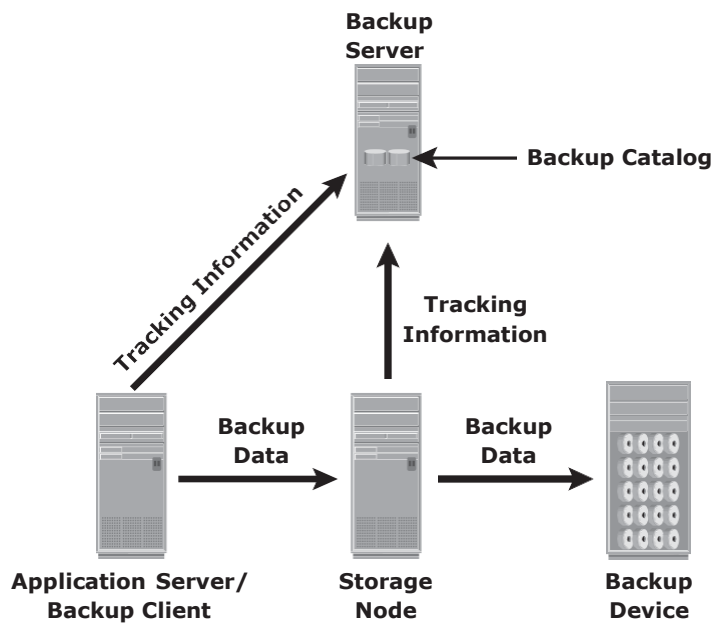


Figure 10-4: Backup architecture

Backup software provides reporting capabilities based on the backup catalog and the log files. These reports include information, such as the amount of data backed up, the number of completed and incomplete backups, and the types of errors that might have occurred. Reports can be customized depending on the specific backup software used.

Protecting backup metadata is an important aspect of backup. If the backup catalog is lost, data recovery will be a challenge. Therefore, an updated copy of the backup catalog should be maintained separately all the time.

Backup and Restore Operations

When a backup operation is initiated, significant network communication takes place between the different components of a backup infrastructure. The backup operation is typically initiated by a server, but it can also be initiated by a client. The backup server initiates the backup process for different clients based on the backup schedule configured for them. For example, the backup for a group of clients may be scheduled to start at 11:00 p.m. every day.

The backup server coordinates the backup process with all the components in a backup environment (see Figure 10-5). The backup server maintains the information about backup clients to be backed up and storage nodes to be used in a backup operation. The backup server retrieves the backup-related information from the backup catalog and, based on this information, instructs the storage node to load the appropriate backup media into the backup devices. Simultaneously, it instructs the backup clients to gather the data to be backed up and send it over the network to the assigned storage node. After the backup data is sent to the storage node, the client sends some backup metadata (the number of files, name of the files, storage node details, and so on) to the backup server. The storage node receives the client data, organizes it, and sends it to the backup device. The storage node then sends additional backup metadata (location of the data on the backup device, time of backup, and so on) to the backup server. The backup server updates the backup catalog with this information.

After the data is backed up, it can be restored when required. A restore process must be manually initiated from the client. Some backup software has a separate application for restore operations. These restore applications are usually accessible only to the administrators or backup operators. Figure 10-6 shows a restore operation.

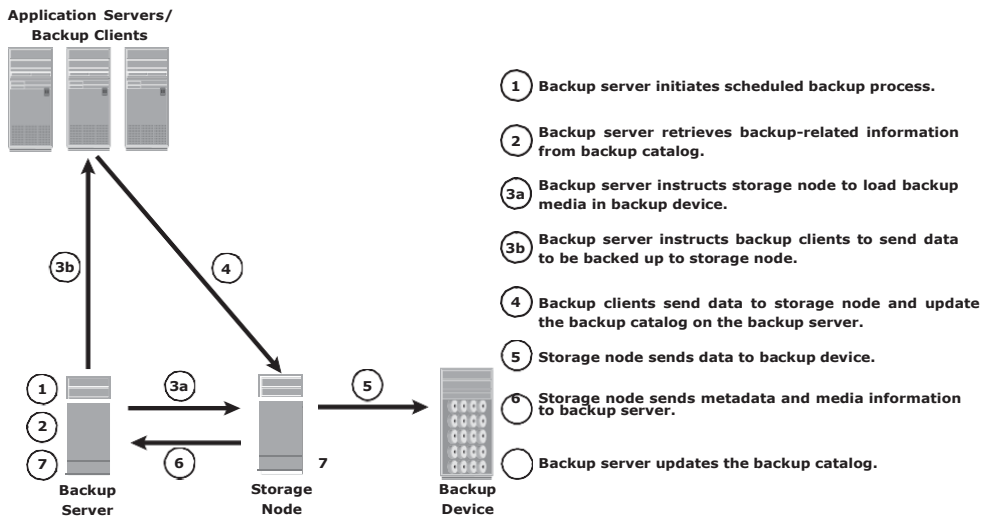


Figure 10-5: Backup operation

Upon receiving a restore request, an administrator opens the restore application to view the list of clients that have been backed up. While selecting the client for which a restore request has been made, the administrator also needs to identify the client that will receive the restored data. Data can be restored on the same client for whom the restore request has been made or on any other client. The administrator then selects the data to be restored and the specified point in time to which the data has to be restored based on the RPO. Because all this information comes from the backup catalog, the restore application needs to communicate with the backup server.

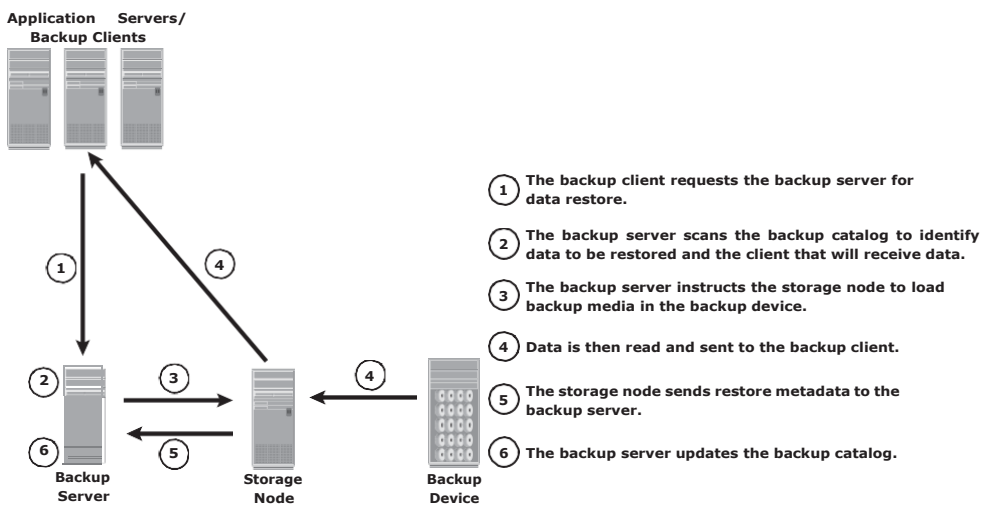


Figure 10-6: Restore operation

The backup server instructs the appropriate storage node to mount the specific backup media onto the backup device. Data is then read and sent to the client that has been identified to receive the restored data.

Some restorations are successfully accomplished by recovering only the requested production data. For example, the recovery process of a spreadsheet is completed when the specific file is restored. In database restorations, additional data, such as log files, must be restored along with the production data. This ensures consistency for the restored data. In these cases, the RTO is extended due to the additional steps in the restore operation.

Backup Topologies

Three basic topologies are used in a backup environment: direct-attached backup, LAN-based backup, and SAN-based backup. A mixed topology is also used by combining LAN-based and SAN-based topologies.

In a *direct-attached backup*, the storage node is configured on a backup client, and the backup device is attached directly to the client. Only the metadata is sent to the backup server through the LAN. This configuration frees the LAN from backup traffic. The example in Figure 10-7 shows that the backup device is directly attached and dedicated to the backup client. As the environment grows, there will be a need for centralized management and sharing of backup devices to optimize costs. An appropriate solution is required to share the backup devices among multiple servers. Network-based topologies (LAN-based and SAN-based) provide the solution to optimize the utilization of backup devices.

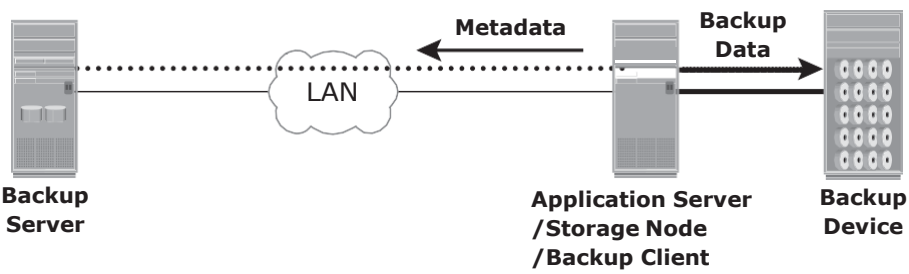


Figure 10-7: Direct-attached backup topology

In a *LAN-based backup*, the clients, backup server, storage node, and backup device are connected to the LAN. (see Figure 10-8). The data to be backed up is

transferred from the backup client (source) to the backup device (destination) over the LAN, which might affect network performance.

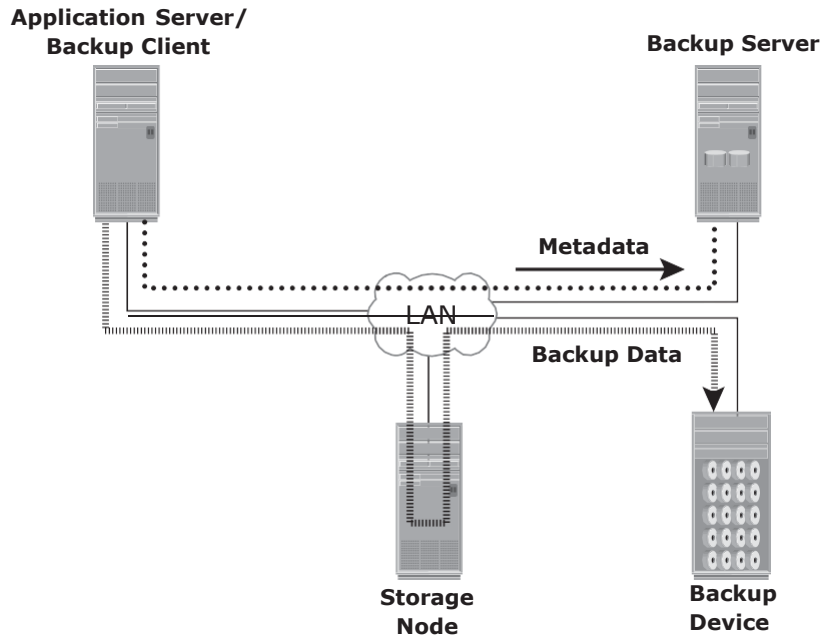


Figure 10-8: LAN-based backup topology

This impact can be minimized by adopting a number of measures, such as configuring separate networks for backup and installing dedicated storage nodes for some application servers.

A *SAN-based backup* is also known as a *LAN-free backup*. The SAN-based backup topology is the most appropriate solution when a backup device needs to be shared among clients. In this case, the backup device and clients are attached to the SAN. Figure 10-9 illustrates a SAN-based backup.

In this example, a client sends the data to be backed up to the backup device over the SAN. Therefore, the backup data traffic is restricted to the SAN, and only the backup metadata is transported over the LAN. The volume of metadata is insignificant when compared to the production data; the LAN performance is not degraded in this configuration.

The emergence of low-cost disks as a backup medium has enabled disk arrays to be attached to the SAN and used as backup devices. A tape backup of these data backups on the disks can be created and shipped offsite for disaster recovery and long-term retention.

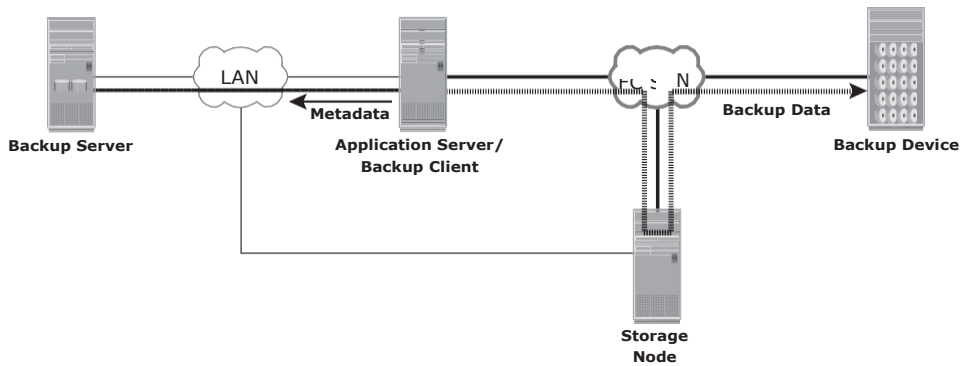


Figure 10-9: SAN-based backup topology

The *mixed topology* uses both the LAN-based and SAN-based topologies, as shown in Figure 10-10. This topology might be implemented for several reasons, including cost, server location, reduction in administrative overhead, and performance considerations.

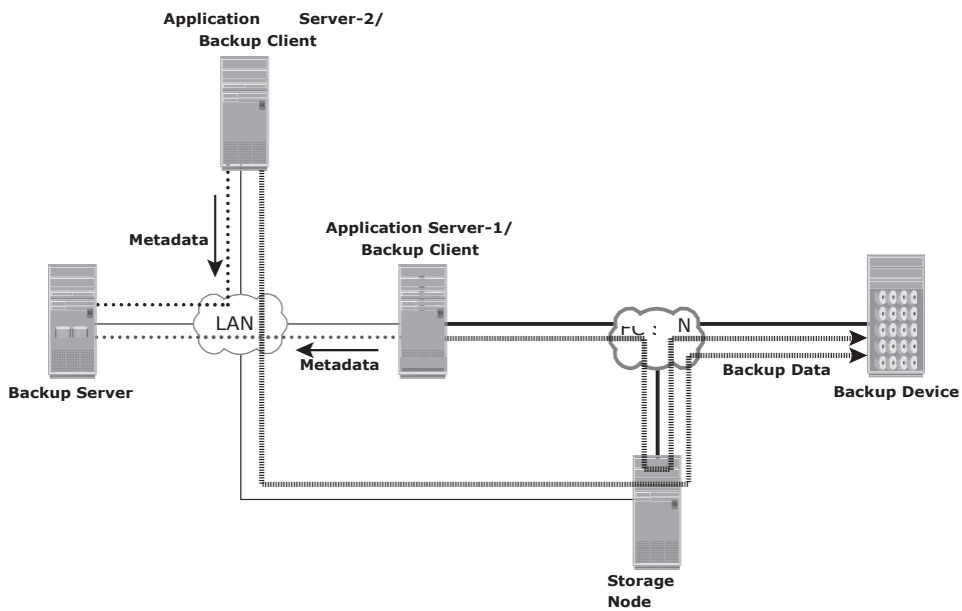


Figure 10-10: Mixed backup topology

Backup in NAS Environments

The use of a NAS head imposes a new set of considerations on the backup and recovery strategy in NAS environments. NAS heads use a proprietary operating system and file system structure that supports multiple file-sharing protocols. In the NAS environment, backups can be implemented in different ways: server based, serverless, or using Network Data Management Protocol (NDMP). Common implementations are NDMP 2-way and NDMP 3-way.

Server-Based and Serverless Backup

In an *application server-based backup*, the NAS head retrieves data from a storage array over the network and transfers it to the backup client running on the application server. The backup client sends this data to the storage node, which in turn writes the data to the backup device. This results in overloading the network with the backup data and using application server resources to move the backup data. Figure 10-11 illustrates server-based backup in the NAS environment.

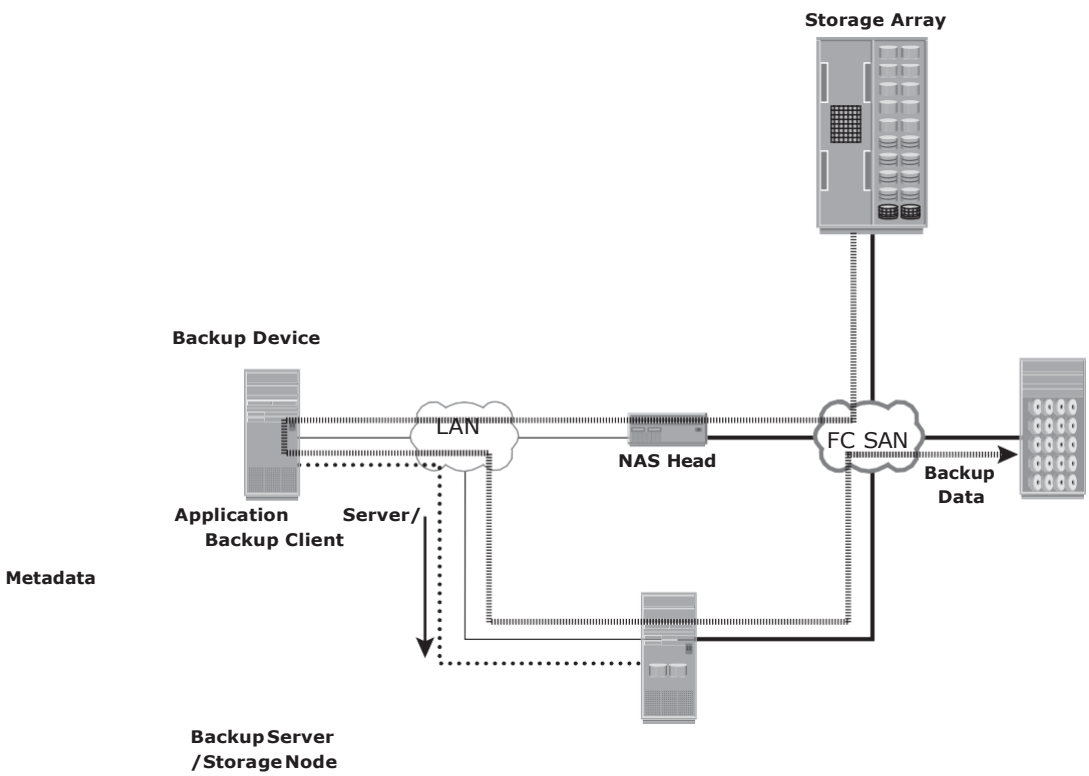


Figure 10-11: Server-based backup in a NAS environment

In a *serverless backup*, the network share is mounted directly on the storage node. This avoids overloading the network during the backup process and eliminates the need to use resources on the application server. Figure 10-12 illustrates serverless backup in the NAS environment. In this scenario, the storage node, which is also a backup client, reads the data from the NAS head and writes it to the backup device without involving the application server. Compared to the previous solution, this eliminates one network hop.

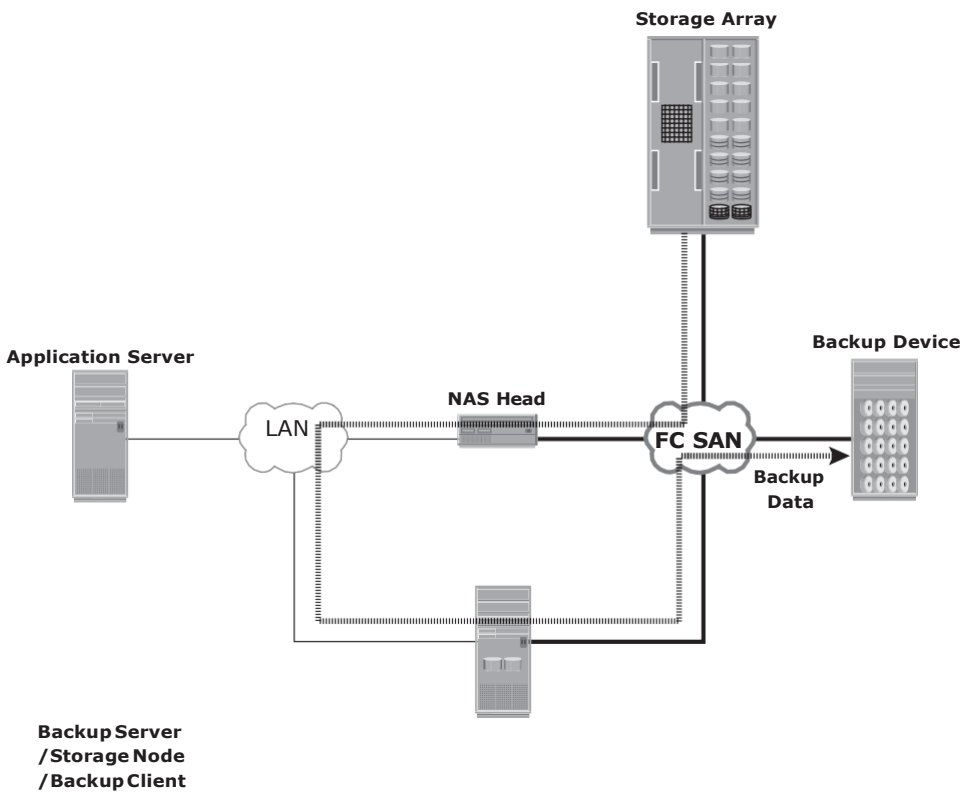


Figure 10-12: Serverless backup in a NAS environment

10.1.1 NDMP-Based Backup

NDMP is an industry-standard TCP/IP-based protocol specifically designed for a backup in a NAS environment. It communicates with several elements in the backup environment (NAS head, backup devices, backup server, and so on) for data transfer and enables vendors to use a common protocol for the backup architecture. Data can be backed up using NDMP regardless of the operating

system or platform. Due to its flexibility, it is no longer necessary to transport data through the application server, which reduces the load on the application server and improves the backup speed.

NDMP optimizes backup and restore by leveraging the high-speed connection between the backup devices and the NAS head. In NDMP, backup data is sent directly from the NAS head to the backup device, whereas metadata is sent to the backup server. Figure 10-13 illustrates a backup in the NAS environment using NDMP 2-way. In this model, network traffic is minimized by isolating data movement from the NAS head to the locally attached backup device. Only metadata is transported on the network. The backup device is dedicated to the NAS device, and hence, this method does not support centralized management of all backup devices.

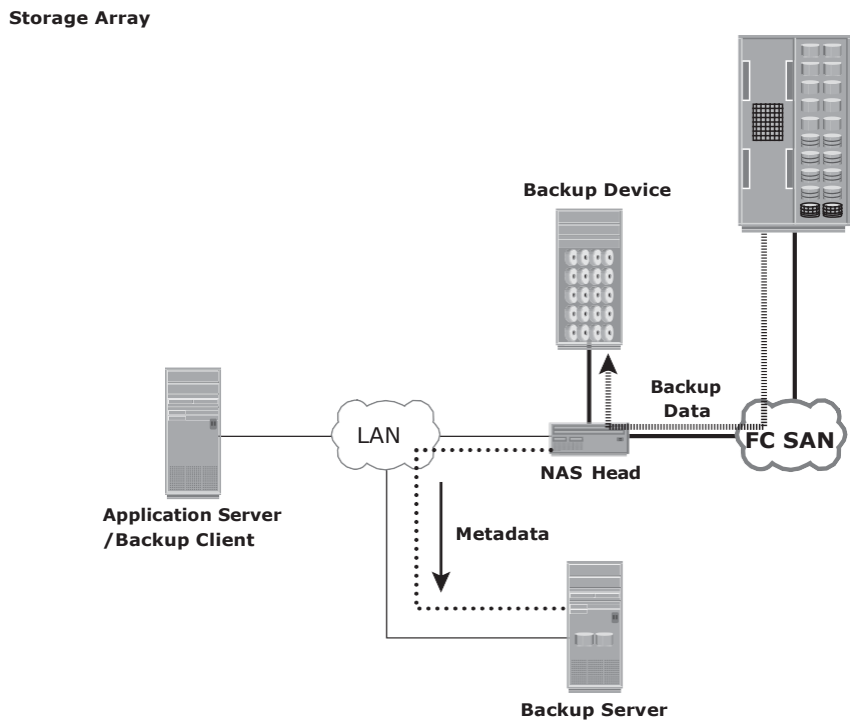


Figure 10-13: NDMP 2-way in a NAS environment

In the *NDMP 3-way* method, a separate private backup network must be established between all NAS heads and the NAS head connected to the backup device. Metadata and NDMP control data are still transferred across the public network. Figure 10-14 shows a NDMP 3-way backup.

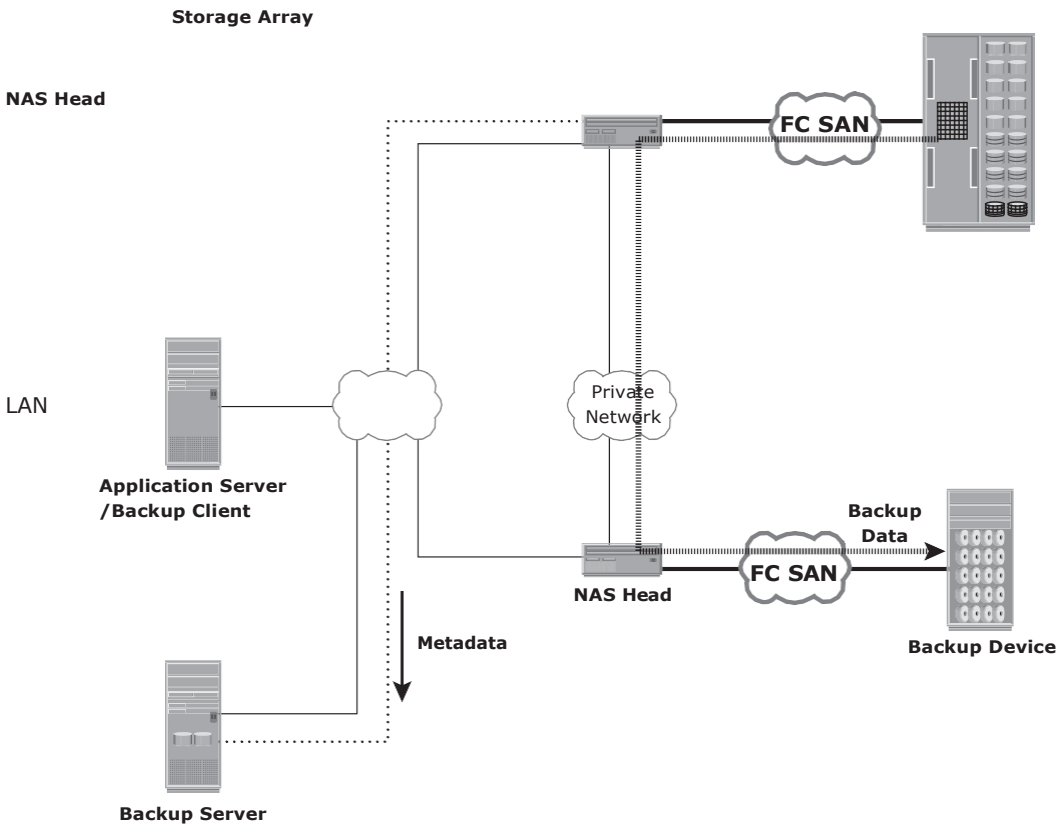


Figure 10-14: NDMP 3-way in a NAS environment

An NDMP 3-way is useful when backup devices need to be shared among NAS heads. It enables the NAS head to control the backup device and share it with other NAS heads by receiving the backup data through the NDMP.

10.2 Backup Targets

A wide range of technology solutions are currently available for backup targets. Tape and disk libraries are the two most commonly used backup targets. In the past, tape technology was the predominant target for backup due to its low cost. But performance and management limitations associated with tapes and the availability of low-cost disk drives have made the disk a viable backup target. A virtual tape library (VTL) is one of the options that uses disks as a backup medium. VTL emulates tapes and provides enhanced backup and recovery capabilities.

Backup to Tape

Tapes, a low- cost solution, are used extensively for backup. Tape drives are used to read/write data from/to a tape cartridge (or cas- sette). Tape drives are referred to as sequential, or linear, access devices because the data is written or read sequentially. A tape cartridge is com- posed of magnetic tapes in a plastic enclosure. *Tape mounting* is the process of inserting a tape cartridge into a tape drive. The tape drive has motorized controls to move the magnetic tape around, enabling the head to read or write data.

Several types of tape cartridges are available. They vary in size, capacity, shape, density, tape length, tape thickness, tape tracks, and supported speed.

Physical Tape Library

The physical tape library provides housing and power for a large number of tape drives and tape cartridges, along with a robotic arm or picker mechanism. The backup software has intelligence to manage the robotic arm and entire backup process. Figure 10-15 shows a physical tape library.

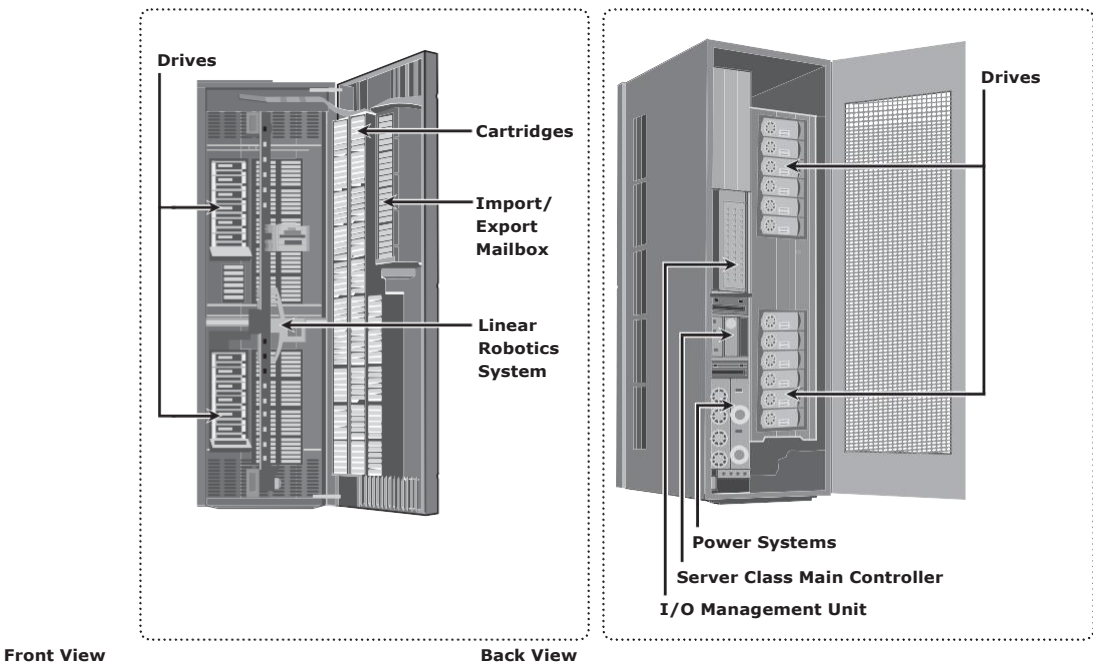


Figure 10-15: Physical tape library

Tape drives read and write data from and to a tape. Tape *cartridges* are placed in the *slots* when not in use by a tape drive. *Robotic arms* are used to move tapes between cartridge slots and tape drives. *Mail* or *import/export slots* are used to add or remove tapes from the library without opening the access doors (refer to Figure 10-15 Front View).

When a backup process starts, the robotic arm is instructed to load a tape to a tape drive. This process adds delay to a degree depending on the type of hardware used, but it generally takes 5 to 10 seconds to mount a tape. After the tape is mounted, additional time is spent to position the heads and validate header information. This total time is called *load to ready time*, and it can vary from several seconds to minutes. The tape drive receives backup data and stores the data in its internal buffer. This backup data is then written to the tape in blocks. During this process, it is best to ensure that the tape drive is kept busy continuously to prevent gaps between the blocks. This is accomplished by buffering the data on tape drives. The speed of the tape drives can also be adjusted to match data transfer rates.

Tape drive *streaming* or *multiple streaming* writes data from multiple streams on a single tape to keep the drive busy. As shown in Figure 10-16, multiple streaming improves media performance, but it has an associated disadvantage. The backup data is interleaved because data from multiple streams is written on it. Consequently, the data recovery time is increased because all the extra data from the other streams must be read and discarded while recovering a single stream.

Figure 10-16: Multiple streams on tape media

Many times, even the buffering and speed adjustment features of a tape drive fail to prevent the gaps, causing the "*shoe shining effect*" or "*backhitching*." *Shoe shining* is the repeated back and forth motion a tape drive makes when there is an interruption in the backup data stream. For example, if a storage node sends data slower than the tape drive writes it to the tape, the drive periodically stops and waits for the data to catch up. After the drive determines that there is enough data to start writing again, it rewinds to the exact place where the last write took place and continues. This repeated back-and-forth motion not only causes a degradation of service, but also excessive wear and tear to tapes.

When the tape operation finishes, the tape rewinds to the starting position and it is unmounted. The robotic arm is then instructed to move the unmounted tape back to the slot. *Rewind time* can range from several seconds to minutes.

When a *restore* is initiated, the backup software identifies which tapes are required. The robotic arm is instructed to move the tape from its slot to a tape drive. If the required tape is not found in the tape library, the backup software displays a message, instructing the operator to manually insert the required tape in the tape library. When a file or a group of files require restores, the tape must move to that file location sequentially before it can start reading. This process can take a significant amount of time, especially if the required files are recorded at the end of the tape.

Modern tape devices have an indexing mechanism that enables a tape to be fast forwarded to a location near the required data. The tape drive then fine-tunes the tape position to get to the data. However, before adopting a solution that uses this mechanism, one should consider the benefits of data streaming performance versus the cost of writing an index.

Limitations of Tape

Tapes are primarily used for long-term offsite storage because of their low cost. Tapes must be stored in locations with a controlled environment to ensure preservation of the media and to prevent data corruption. Data access in a tape is sequential, which can slow backup and recovery operations. Tapes are highly susceptible to wear and tear and usually have shorter shelf life. Physical transportation of the tapes to offsite locations also adds to management overhead and increases the possibility of loss of tapes during offsite shipment.

Backup to Disk

Because of increased availability, low cost disks have now replaced tapes as the primary device for storing backup data because of their performance advantages. Backup-to-disk systems offer ease of implementation, reduced TCO, and improved quality of service. Apart from performance benefits in terms of data transfer rates, disks also offer faster recovery when compared to tapes.

Backing up to disk storage systems offers clear advantages due to their inherent random access and RAID-protection capabilities. In most backup environments, backup to disk is used as a staging area where the data is copied temporarily before transferring or staging it to tapes. This enhances backup performance. Some backup products allow for backup images to remain on the disk for a period of time even after they have been staged. This enables a much faster restore. Figure 10-17 illustrates a recovery scenario comparing tape versus disk in a Microsoft Exchange environment that supports 800 users with a 75 MB mailbox size and a 60 GB database. As shown in the figure, a restore from the

disk took 24 minutes compared to the restore from a tape, which took 108 minutes for the same environment.

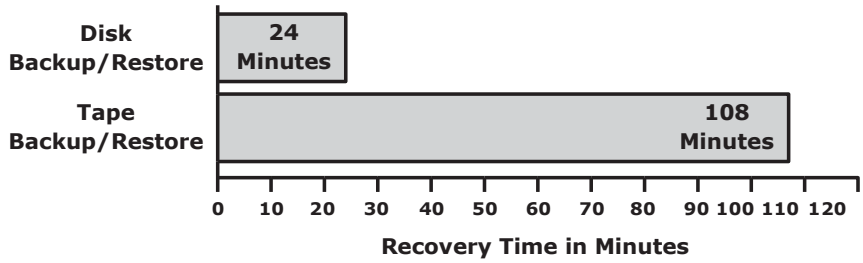


Figure 10-17: Tape versus disk restore

Recovering from a full backup copy stored on disk and kept onsite provides the fastest recovery solution. Using a disk enables the creation of full backups more frequently, which in turn improves RPO and RTO.

Backup to disk does not offer any inherent offsite capability and is dependent on other technologies, such as local and remote replication. In addition, some backup products require additional modules and licenses to support backup to disk, which may also require additional configuration steps, including creation of RAID groups and file system tuning. These activities are not usually performed by a backup administrator.

Backup to Virtual Tape

Virtual tapes are disk drives emulated and presented as tapes to the backup software. The key benefit of using a virtual tape is that it does not require any additional modules, configuration, or changes in the legacy backup software. This preserves the investment made in the backup software.

Virtual Tape Library

A *virtual tape library* (VTL) has the same components as that of a physical tape library, except that the majority of the components are presented as virtual resources. For the backup software, there is no difference between a physical tape library and a virtual tape library. Figure 10-18 shows a virtual tape library. Virtual tape libraries use disks as backup media. Emulation software has a database with a list of virtual tapes, and each virtual tape is assigned space on a LUN. A virtual tape can span multiple LUNs if required. File system awareness is not required while backing up because the virtual tape solution typically uses raw devices.

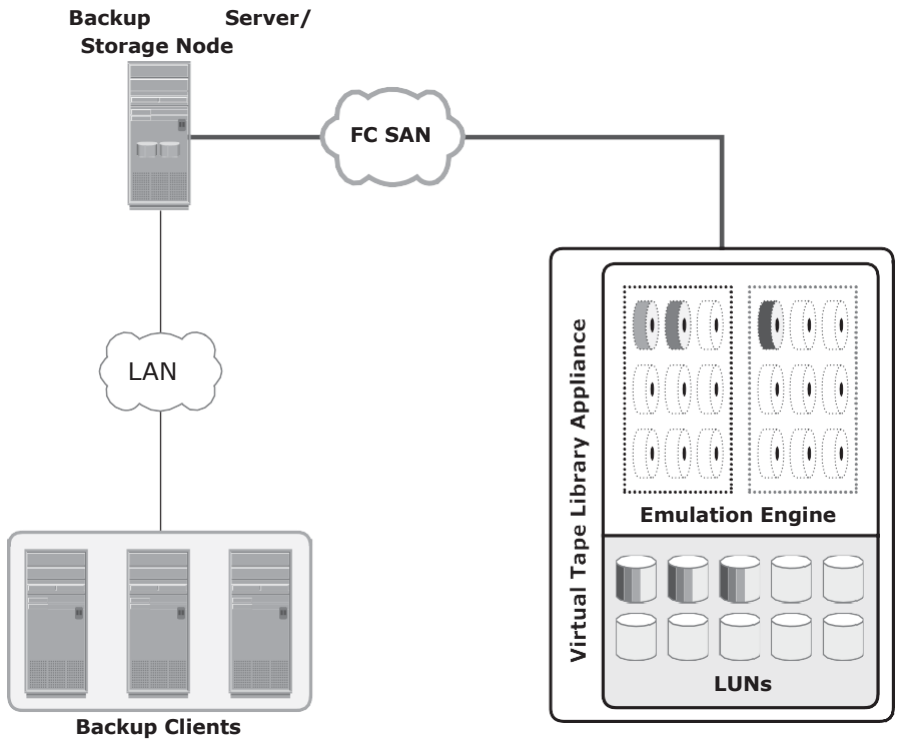


Figure 10-18: Virtual tape library

Similar to a physical tape library, a robot mount is virtually performed when a backup process starts in a virtual tape library. However, unlike a physical tape library, where this process involves some mechanical delays, in a virtual tape library it is almost instantaneous. Even the *load to ready* time is much less than in a physical tape library.

After the virtual tape is mounted and the virtual tape drive is positioned, the virtual tape is ready to be used, and backup data can be written to it. In most cases, data is written to the virtual tape immediately. Unlike a physical tape library, the virtual tape library is not constrained by the sequential access and shoe shining effect. When the operation is complete, the backup software issues a rewind command. This rewind is also instantaneous. The virtual tape is then unmounted, and the virtual robotic arm is instructed to move it back to a virtual slot.

The steps to restore data are similar to those in a physical tape library, but the restore operation is nearly instantaneous. Even though virtual tapes are based on disks, which provide random access, they still emulate the tape behavior.

A virtual tape library appliance offers a number of features that are not available with physical tape libraries. Some virtual tape libraries offer *multiple emulation engines* configured in an active cluster configuration. An engine is a dedicated server with a customized operating system that makes physical disks in the VTL appear as tapes to the backup application. With this feature, one engine can pick up the virtual resources from another engine in the event of any failure and enable the clients to continue using their assigned virtual resources transparently.

Data replication over IP is available with most of the virtual tape library appliances. This feature enables virtual tapes to be replicated over an inexpensive IP network to a remote site. As a result, organizations can comply with offsite requirements for backup data. Connecting the engines of a virtual tape library appliance to a physical tape library enables the virtual tapes to be copied onto the physical tapes, which can then be sent to a vault or shipped to an offsite location. Using virtual tapes offers several advantages over both physical tapes and disks. Compared to physical tapes, virtual tapes offer better single stream performance, better reliability, and random disk access characteristics. Backup and restore operations benefit from the disk's random access characteristics because they are always online and provide faster backup and recovery. A virtual tape drive does not require the usual maintenance tasks associated with a physical tape drive, such as periodic cleaning and drive calibration. Compared to backup-to-disk devices, a virtual tape library offers easy installation and administration because it is preconfigured by the manufacturer. However, a virtual tape library is generally used only for backup purposes. In a backup-to-disk environment, the disk systems are used for both production and backup data.

Table 10-1 shows a comparison between various backup targets.

Table 10-1: Backup Targets Comparison

FEATURES	TAPE	DISK	VIRTUAL TAPE
Offsite Replication Capabilities	No	Yes	Yes
Reliability	No inherent protection methods	Yes	Yes
Performance	Subject to mechanical operations, loading time	Faster single stream	Faster single stream
Use	Backup only	Multiple (backup, production)	Backup only

Data Deduplication for Backup

Traditional backup solutions do not provide any inherent capability to prevent duplicate data from being backed up. With the growth of information and 24x7 application availability requirements, backup windows are shrinking. Traditional backup processes back up a lot of duplicate data. Backing up duplicate data significantly increases the backup window size requirements and results in unnecessary consumption of resources, such as storage space and network bandwidth.

Data deduplication is the process of identifying and eliminating redundant data. When duplicate data is detected during backup, the data is discarded and only the pointer is created to refer the copy of the data that is already backed up. Data deduplication helps to reduce the storage requirement for backup, shorten the backup window, and remove the network burden. It also helps to store more backups on the disk and retain the data on the disk for a longer time.

Data Deduplication Methods

There are two methods of deduplication: file level and subfile level. Determining the uniqueness by implementing either method offers benefits; however, results can vary. The differences exist in the amount of data reduction each method produces and the time each approach takes to determine the unique content.

File-level deduplication (also called *single-instance storage*) detects and removes redundant copies of identical files. It enables storing only one copy of the file; the subsequent copies are replaced with a pointer that points to the original file. File-level deduplication is simple and fast but does not address the problem of duplicate content inside the files. For example, two 10-MB PowerPoint presentations with a difference in just the title page are not considered as duplicate files, and each file will be stored separately.

Subfile deduplication breaks the file into smaller chunks and then uses a specialized algorithm to detect redundant data within and across the file. As a result, subfile deduplication eliminates duplicate data across files. There are two forms of subfile deduplication: fixed-length block and variable-length segment. The *fixed-length block deduplication* divides the files into fixed length blocks and uses a hash algorithm to find the duplicate data. Although simple in design, fixed-length blocks might miss many opportunities to discover redundant data because the block boundary of similar data might be different. Consider the addition of a person's name to a document's title page. This shifts the whole document, and all the blocks appear to have changed, causing the failure of the deduplication method to detect equivalencies. In *variable-length segment deduplication*, if there is a change in the segment, the

boundary for only that segment is adjusted, leaving the remaining segments unchanged. This method vastly improves the ability to find duplicate data segments compared to fixed-block.

Data Deduplication Implementation

Deduplication for backup can happen at the data source or the backup target.

Source-Based Data Deduplication

Source-based data deduplication eliminates redundant data at the source before it transmits to the backup device. Source-based data deduplication can dramatically reduce the amount of backup data sent over the network during backup processes. It provides the benefits of a shorter backup window and requires less network bandwidth. There is also a substantial reduction in the capacity required to store the backup images. Figure 10-19 shows source-based data deduplication.

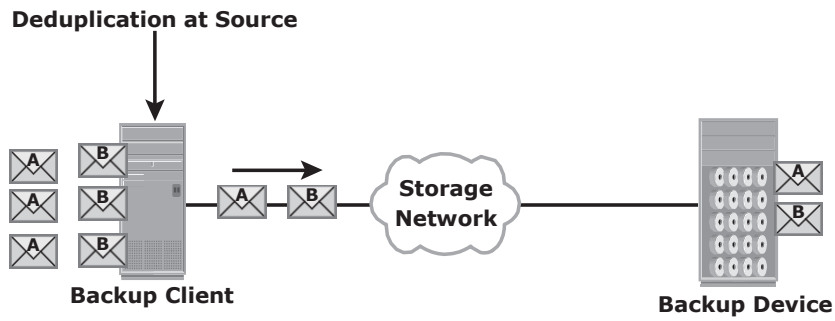


Figure 10-19: Source-based data deduplication

Source-based deduplication increases the overhead on the backup client, which impacts the performance of the backup and application running on the client. Source-based deduplication might also require a change of backup software if it is not supported by backup software.

Target-Based Data Deduplication

Target-based data deduplication is an alternative to source-based data deduplication. Target-based data deduplication occurs at the backup device, which offloads the backup client from the deduplication process. Figure 10-20 shows target-based data deduplication.

In this case, the backup client sends the data to the backup device and the data is deduplicated at the backup device, either immediately (inline) or at a scheduled time (post-process). Because deduplication occurs at the target, all the

backup data needs to be transferred over the network, which increases network bandwidth requirements. Target-based data deduplication does not require any changes in the existing backup software.

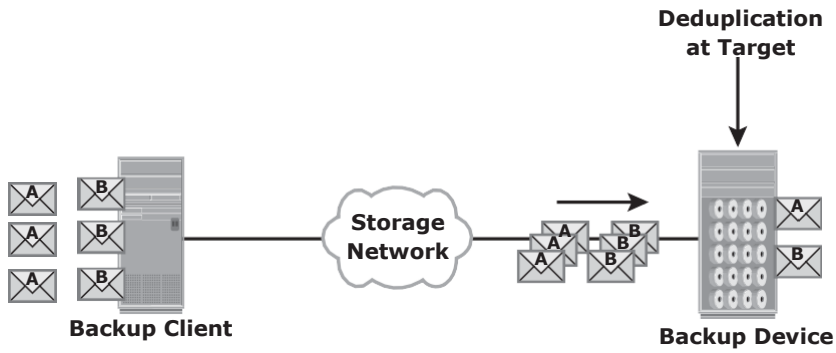


Figure 10-20: Target-based data deduplication

Inline deduplication performs deduplication on the backup data before it is stored on the backup device. Hence, this method reduces the storage capacity needed for the backup. Inline deduplication introduces overhead in the form of the time required to identify and remove duplication in the data. So, this method is best suited for an environment with a large backup window.

Post-process deduplication enables the backup data to be stored or written on the backup device first and then deduplicated later. This method is suitable for situations with tighter backup windows. However, post-process deduplication requires more storage capacity to store the backup images before they are deduplicated.

Backup in Virtualized Environments

In a virtualized environment, it is imperative to back up the virtual machine data (OS, application data, and configuration) to prevent its loss or corruption due to human or technical errors. There are two approaches for performing a backup in a virtualized environment: the traditional backup approach and the image-based backup approach.

In the *traditional backup approach*, a backup agent is installed either on the virtual machine (VM) or on the hypervisor. Figure 10-21 shows the traditional VM backup approach. If the backup agent is installed on a VM, the VM appears as a physical server to the agent. The backup agent installed on the VM backs up the VM data to the backup device. The agent does not capture VM files, such as the virtual BIOS file, VM swap file, logs, and configuration files. Therefore, for a VM restore, a user needs to manually re-create the VM and then restore data onto it.

If the backup agent is installed on the hypervisor, the VMs appear as a set of files to the agent. So, VM files can be backed up by performing a file system backup from a hypervisor. This approach is relatively simple because it requires having the agent just on the hypervisor instead of all the VMs. The traditional backup method can cause high CPU utilization on the server being backed up.

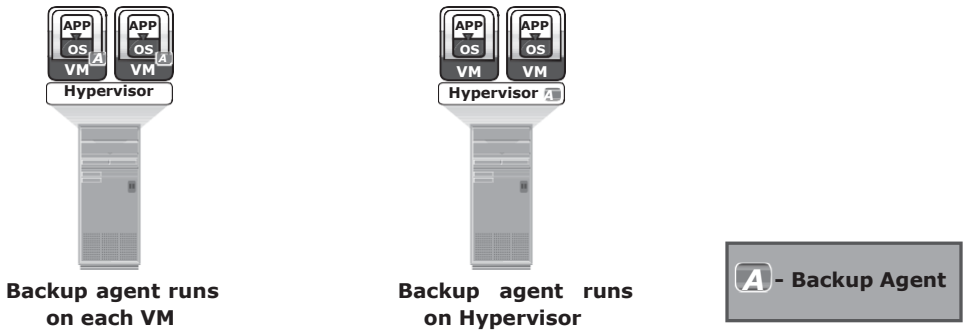


Figure 10-21: Traditional VM backup

In the traditional approach, the backup should be performed when the server resources are idle or during a low activity period on the network. Also consider allocating enough resources to manage the backup on each server when a large number of VMs are in the environment.

Image-based backup operates at the hypervisor level and essentially takes a snapshot of the VM. It creates a copy of the guest OS and all the data associated with it (snapshot of VM disk files), including the VM state and application configurations. The backup is saved as a single file called an "image," and this image is mounted on the separate physical machine–proxy server, which acts as a backup client. The backup software then backs up these image files normally. (see Figure 10-22). This effectively offloads the backup processing from the hypervisor and transfers the load on the proxy server, thereby reducing the impact to VMs running on the hypervisor. Image-based backup enables quick restoration of a VM.

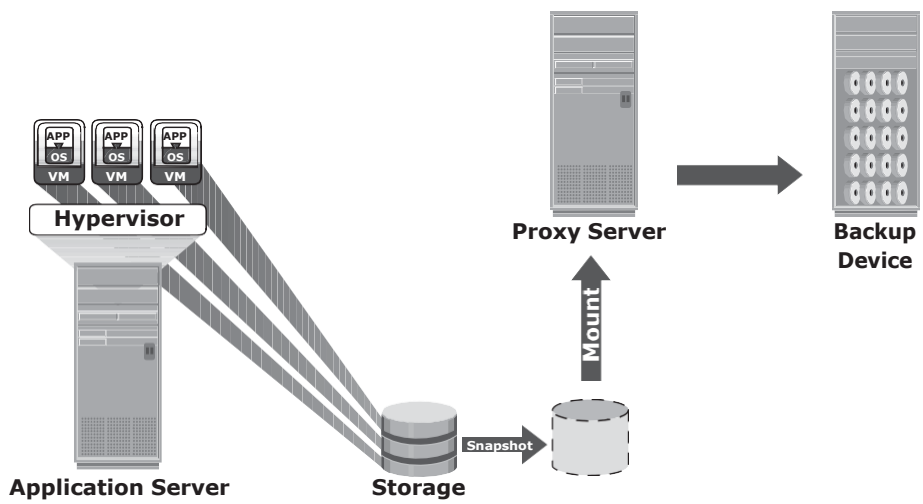


Figure 10-22: Image-based backup

The use of deduplication techniques significantly reduces the amount of data to be backed up in a virtualized environment. The effectiveness of deduplication is identified when VMs with similar configurations are deployed in a data center. The deduplication types and methods used in a virtualized environment are the same as in the physical environment.

Data Archive

In the life cycle of information, data is actively created, accessed, and changed. As data ages, it is less likely to be changed and eventually becomes "fixed" but continues to be accessed by applications and users. This data is called *fixed content*. X-rays, e-mails, and multimedia files are examples of fixed content. Figure 10-23 shows some examples of fixed content.

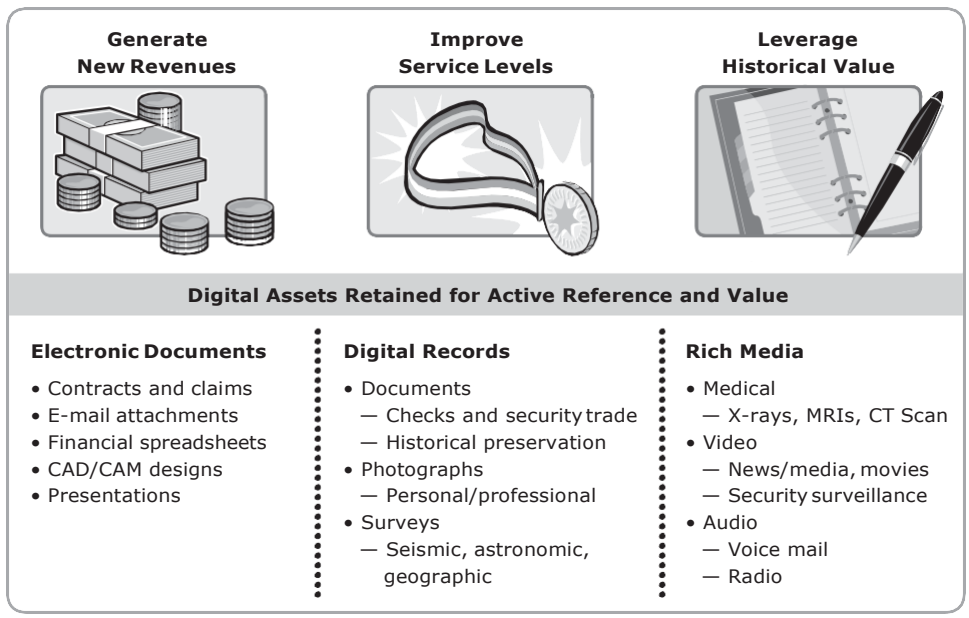


Figure 10-23: Examples of fixed content data

All organizations may require retention of their data for an extended period of time due to government regulations and legal/contractual obligations. Organizations also make use of this fixed content to generate new revenue strategies and improve service levels. A repository where fixed content is stored is known as an archive.

An archive can be implemented as an online, nearline, or offline solution:

- **Online archive:** A storage device directly connected to a host that makes the data immediately accessible.
- **Nearline archive:** A storage device connected to a host, but the device where the data is stored must be mounted or loaded to access the data.
- **Offline archive:** A storage device that is not ready to use. Manual intervention is required to connect, mount, or load the storage device before data can be accessed.

Traditionally, optical and tape media were used for archives. Optical media are typically *write once read many* (WORM) devices that protect the original file from being overwritten. Some tape devices also provide this functionality by implementing file-locking capabilities. Although these devices are inexpensive, they involve operational, management, and maintenance overhead. The traditional archival process using optical discs and tapes is not optimized to recognize the content, so the same content could be archived several times. Additional costs are involved in offsite storage of media and media management. Tapes and optical media are also susceptible to wear and tear. Frequent changes in these device technologies lead to the overhead of converting the media into new formats to enable access and retrieval. Government agencies and industry regulators are establishing new laws and regulations to enforce the protection of archives from unauthorized destruction and modification. These regulations and standards have established new requirements for preserving the integrity of information in the archives. These requirements have exposed the shortcomings of the traditional tape and optical media archive solutions.

Content addressed storage (CAS) is disk-based storage that has emerged as an alternative to tape and optical solutions. CAS meets the demand to improve data accessibility and to protect, dispose of, and ensure service-level agreements (SLAs) for archive data. CAS is detailed in Chapter 8.

Archiving Solution Architecture

Archiving solution architecture consists of three key components: archiving agent, archiving server, and archiving storage device (see Figure 10-24).

An archiving agent is software installed on the application server. The agent is responsible for scanning the data that can be archived based on the policy defined on the archiving server. After the data is identified for archiving, the agent sends the data to the archiving server. Then the original data on the application server is replaced with a stub file. The stub file contains the address of the archived data. The size of this file is small and significantly saves space on primary storage. This stub file is used to retrieve the file from the archive storage device.

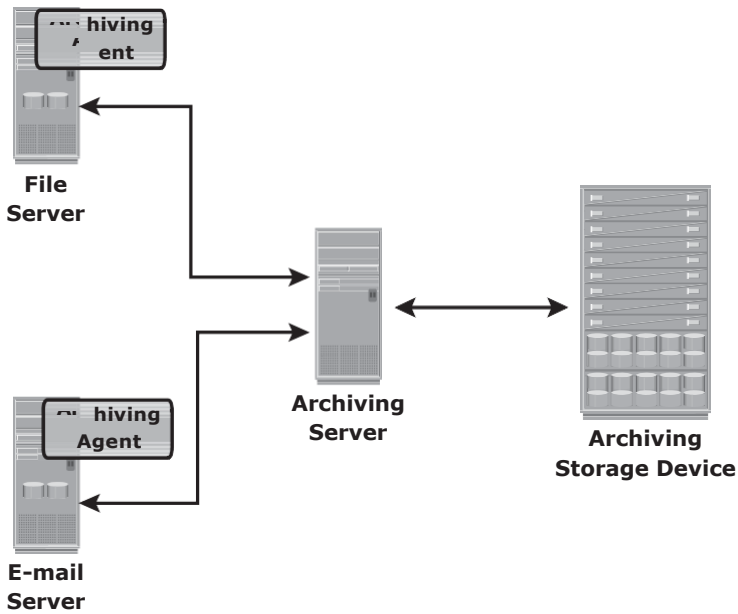


Figure 10-24: Archiving solution architecture

An archiving server is software installed on a host that enables administrators to configure the policies for archiving data. Policies can be defined based on file size, file type, or creation/modification/access time. The archiving server receives the data to be archived from the agent and sends it to the archive storage device.

An archiving storage device stores fixed content. Different types of storage media options such as optical, tapes, and low-cost disk drives are available for archiving.

Use Case: E-mail Archiving

E-mail is an example of an application that benefits most by an archival solution. Typically, a system administrator configures small mailboxes that store a limited number of e-mails. This is because large mailboxes with a large number of e-mails can make management difficult, increase primary storage cost, and degrade system performance. When an e-mail server is configured with a large number of mailboxes, the system administrator typically configures a quota on each mailbox to limit its size on that server. Configuring fixed quotas on mailboxes impacts end users. A fixed quota for a mailbox forces users to delete e-mails as they approach the quota size. End users often need to access e-mails that are weeks, months, or even years old.

E-mail archiving provides an excellent solution that overcomes the preceding challenges. Archiving solutions move e-mails that have been identified as candidates for archive from primary storage to the archive storage device based on a policy — for example, "e-mails that are 90 days old should be archived." After the e-mail is archived, it is retained for years based on the retention policy. This considerably saves space on primary storage and enables organizations to meet regulatory requirements. Implementation of an archiving solution gives end users virtually unlimited mailbox space.

Use Case: File Archiving

A file sharing environment is another environment that benefits from an archival solution. Typically, users store a large number of files in the shared location. Most of these files are old and rarely accessed. Administrators configure quotas on the file share that forces the users to delete these files. This impacts users because they may require access to files that may be months or even years old. In some cases the user may request an increase in the size of the file share. This in turn increases the cost of primary storage. A file archiving solution archives the files based on the policy such as age of files, size of files, and so on. This considerably reduces the primary storage requirement and also enables users to retain the files in the archive for longer periods.

Concepts in Practice: EMC NetWorker, EMC Avamar, and EMC Data Domain

The EMC backup, recovery, and deduplication portfolio consists of a broad range of products for an ever-increasing amount of backup data. This section provides a brief introduction to EMC NetWorker, EMC Avamar, and EMC Data Domain. For the latest information, visit `www.emc.com`.

EMC NetWorker

The EMC NetWorker backup and recovery software centralizes, automates, and accelerates data backup and recovery operations across the enterprise. Following are the features of EMC NetWorker:

- Supports heterogeneous platforms, such as Windows, UNIX, and Linux, and also supports virtual environments
- Supports clustering technologies and open-file backup
- Supports different backup targets: tapes, disks, and virtual tapes
- Supports Multiplexing (or multistreaming) of data
- Provides both source-based and target-based deduplication capabilities by integrating with EMC Avamar and EMC Data Domain respectively
- Uses 256-bit AES (advanced encryption standard) encryption to provide security for the backup data. NetWorker hosts are authenticated using strong authentication based on the Secure Sockets Layer (SSL) protocol.
- The cloud-backup option in NetWorker enables backing up data to both private and public cloud configurations.

NetWorker provides centralized management of the backup environment through a GUI, customizable reporting, and wizard-driven configuration. With the NetWorker Management Console (NMC), backup can be easily administered from any host with a supported web browser. NetWorker also provides many command-line utilities. To facilitate NetWorker administration, several reports are available through the NMC reporting feature. Data maintained in the NMC server database, gathered from any or all of the NetWorker servers, is used to prepare reports on backup statistics and status, events, hosts, users, and devices.

EMC Avamar

EMC Avamar is a disk-based backup and recovery solution that provides inherent source-based data deduplication. With its unique global data deduplication feature, Avamar differs from traditional backup and recovery solutions, by identifying and storing only unique subfile data objects. Redundant data is identified at the source, the amount of data that travels across the network is drastically reduced, and the backup storage requirement is also considerably reduced. The three major components of an Avamar system include Avamar server, Avamar backup clients, and Avamar administrator. Avamar server stores client backups and provides the essential processes and services required for client access and remote system administration. The Avamar client software runs on each computer or network server being backed up. Avamar administrator is

a user management console application used to remotely administer an Avamar system. Following are the three Avamar server editions:

- **Software only:** The Avamar Software edition is a software-only solution. The server software is installed on customer-supplied, Avamar-qualified hardware platforms.
- **Avamar Data Store:** The Avamar Data Store edition includes both hardware and Avamar server software from EMC.
- **Avamar Virtual Edition:** Avamar Virtual Edition for VMware is Avamar server software deployed as a virtual appliance.

The features of EMC Avamar follows:

- **Data deduplication:** Ensures that data is backed up only once across the backup environment.
- **Systematic fault tolerance:** Uses RAID, RAIN, checkpoints, and replica- tion, which provide data integrity and disaster recovery protection.
- **Standard IP network leveraging:** Optimizes the use of a network for backup; dedicated backup networks are not required. Daily full backups are possible using the existing networks and infrastructure.
- **Scalable server architecture:** Additional storage nodes can be added nondisruptively to an Avamar multinode server in Avamar Data Store to accommodate increased backup storage requirements.
- **Centralized management:** Enables remote management of Avamar servers from a centralized location and through the use of the Avamar Enterprise Manager and Avamar Administrator interfaces.

EMC Data Domain

The EMC Data Domain deduplication storage system is a target-based data deduplication solution. Using high-speed, inline deduplication technology, the Data Domain system provides a storage footprint that is signifiacantly smaller on an average than the original data set. It supports various backup and enterprise applications in database, e-mail, content management, and virtual environments. Data Domain systems can scale from small remote office appliances to large data-center systems. These systems are available as integrated appliances or as gateways that use external storage.

Data Domain deduplication storage systems provide the following unique advantages:

- **Data invulnerability architecture:** Provides unprecedented levels of data integrity, data verification, and self-healing capabilities, such as RAID 6

protection. Continuous fault detection, healing, and write verification ensure that the backup is accurately stored, available, and recoverable.

- **Data Domain SISL (Stream-Informed Segment Layout) scaling archi- tecture:** Enables scaling of CPUs to add a direct benefit to the system throughput scalability
- **Support native replication technology:** Enables automatic, secure transfer of compressed data over the wide area network (WAN) with minimum bandwidth requirement
- **Global compression:** Highly efficient deduplication and compression technology, which radically changes storage economics

EMC Data Domain Archiver is a solution for long-term retention of backup and archive data. It is designed with an internal tiering approach to enable cost-effective, long-term retention of data on disk by implementing deduplication technology.

Replication Terminology

The common terms used to represent various entities and operations in a replication environment are listed here:

- **Source:** A host accessing the production data from one or more LUNs on the storage array is called a *production host*, and these LUNs are known as source LUNs (devices/volumes), production LUNs, or simply the *source*.
- **Target:** A LUN (or LUNs) on which the production data is replicated, is called the target LUN or simply the *target* or replica.
- **Point-in-Time (PIT) and continuous replica:** Replicas can be either a PIT or a continuous copy. The PIT replica is an identical image of the source at some specific timestamp. For example, if a replica of a file system is created at 4:00 p.m. on Monday, this replica is the Monday 4:00 p.m. PIT copy. On the other hand, the continuous replica is in-sync with the production data at all times.
- **Recoverability and restartability:** Recoverability enables restoration of data from the replicas to the source if data loss or corruption occurs. Restartability enables restarting business operations using the replicas. The replica must be consistent with the source so that it is usable for both recovery and restart operations. Replica consistency is detailed in section "11.3 Replica Consistency."

Uses of Local Replicas

One or more local replicas of the source data may be created for various purposes, including the following:

- **Alternative source for backup:** Under normal backup operations, data is read from the production volumes (LUNs) and written to the backup device. This places an additional burden on the production infrastructure because production LUNs are simultaneously involved in production

operations and servicing data for backup operations. The local replica contains an exact point-in-time (PIT) copy of the source data, and therefore can be used as a source to perform backup operations. This alleviates the backup I/O workload on the production volumes. Another benefit of using local replicas for backup is that it reduces the *backup window* to zero.

- **Fast recovery:** If data loss or data corruption occurs on the source, a local replica might be used to recover the lost or corrupted data. If a complete failure of the source occurs, some replication solutions enable a replica to be used to restore data onto a different set of source devices, or production can be restarted on the replica. In either case, this method provides faster recovery and minimal RTO compared to traditional recovery from tape backups. In many instances, business operations can be started using the source device before the data is completely copied from the replica.
- **Decision-support activities, such as reporting or data warehousing:** Running the reports using the data on the replicas greatly reduces the I/O burden placed on the production device. Local replicas are also used for data-warehousing applications. The data-warehouse application may be populated by the data on the replica and thus avoid the impact on the production environment.
- **Testing platform:** Local replicas are also used for testing new applications or upgrades. For example, an organization may use the replica to test the production application upgrade; if the test is successful, the upgrade may be implemented on the production environment.
- **Data migration:** Another use for a local replica is data migration. Data migrations are performed for various reasons, such as migrating from a smaller capacity LUN to one of a larger capacity for newer versions of the application.

Replica Consistency

Most file systems and databases buffer the data in the host before writing it to the disk. A consistent replica ensures that the data buffered in the host is captured on the disk when the replica is created. The data staged in the cache and not yet committed to the disk should be flushed before taking the replica. The storage array operating environment takes care of flushing its cache before the replication operation is initiated. Consistency ensures the usability of a replica and is a primary requirement for all the replication technologies.

Consistency of a Replicated File System

File systems buffer the data in the host memory to improve the application response time. The buffered data is periodically written to the disk. In UNIX operating systems, *sync daemon* is the process that flushes the buffers to the disk

at set intervals. In some cases, the replica is created between the set intervals, which might result in the creation of an inconsistent replica. Therefore, host memory buffers must be flushed to ensure data consistency on the replica, prior to its creation. Figure 11-1 illustrates how the file system buffer is flushed to the source device before replication. If the host memory buffers are not flushed, the data on the replica will not contain the information that was buffered in the host. If the file system is unmounted before creating the replica, the buffers will be automatically flushed and the data will be consistent on the replica.

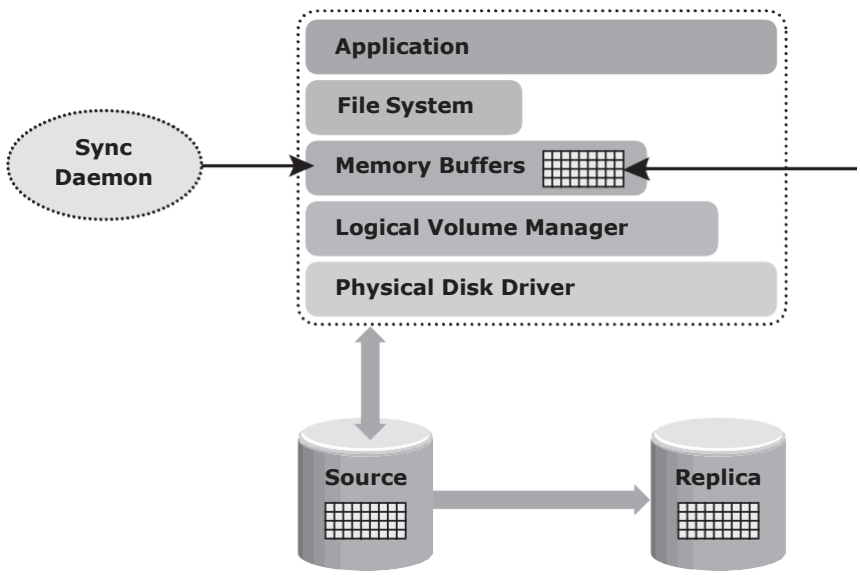


Figure 11-1: Flushing the file system buffer

If a mounted file system is replicated, some level of recovery, such as *fsck* or *log replay*, is required on the replicated file system. When the file system replication and check process are completed, the replica file system can be mounted for operational use.

Consistency of a Replicated Database

A database may be spread over numerous files, file systems, and devices. All of these must be replicated consistently to ensure that the replica is restorable and restartable. Replication is performed with the database offline or online. If the database is offline during the creation of the replica, it is not available for I/O operations. Because no updates occur on the source, the replica is consistent.

If the database is online, it is available for I/O operations, and transactions to the database update the data continuously. When a database is replicated while

it is online, changes made to the database at this time must be applied to the replica to make it consistent. A consistent replica of an online database is created by using the dependent write I/O principle or by holding I/Os momentarily to the source before creating the replica.

A *dependent write I/O* principle is inherent in many applications and database management systems (DBMS) to ensure consistency. According to this principle, a write I/O is not issued by an application until a prior related write I/O has completed. For example, a data write is dependent on the successful completion of the prior log write.

For a transaction to be deemed complete, databases require a series of writes to have occurred in a particular order. These writes will be recorded on the various devices or file systems. Figure 11-2, illustrates the process of flushing the buffer from the host to the source; I/Os 1 to 4 must complete for the transaction to be considered complete. I/O 4 is dependent on I/O 3 and occurs only if I/O 3 is complete. I/O 3 is dependent on I/O 2, which in turn depends on I/O 1. Each I/O completes only after completion of the previous I/O(s).

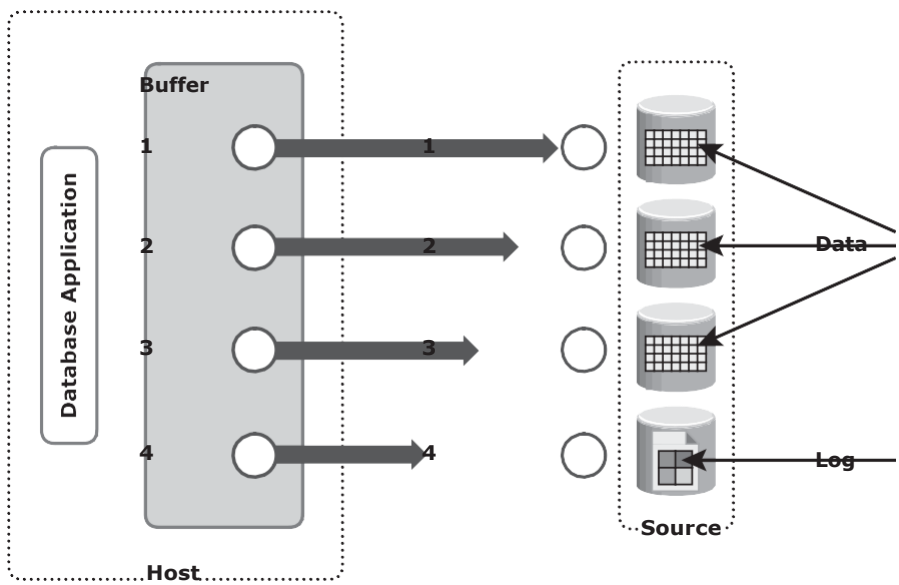


Figure 11-2: Dependent write consistency on sources

When the replica is created, all the writes to the source devices must be captured on the replica devices to ensure data consistency. Figure 11-3 illustrates the process of replication from the source to the replica. I/O transactions 1 to 4 must be carried out for the data to be consistent on the replica.

It is possible that I/O transactions 3 and 4 were copied to the replica devices, but I/O transactions 1 and 2 were not copied. Figure 11-4 shows this situation.

In this case, the data on the replica is inconsistent with the data on the source. If a restart were to be performed on the replica devices, I/O 4, which is available on the replica, might indicate that a particular transaction is complete, but all the data associated with the transaction will be unavailable on the replica, making the replica inconsistent.

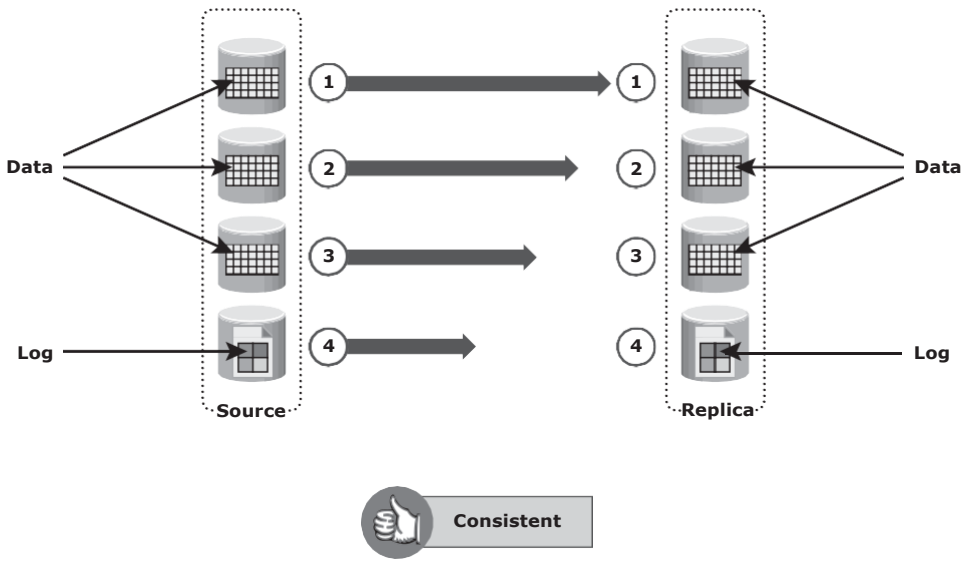


Figure 11-3: Dependent write consistency on replica

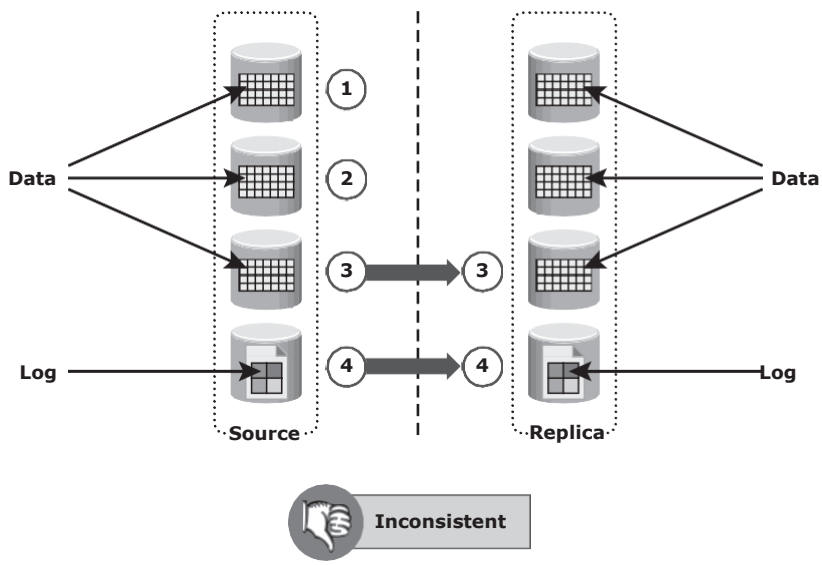


Figure 11-4: Inconsistent database replica

Another way to ensure consistency is to make sure that the write I/O to all source devices is held for the duration of creating the replica. This creates a consistent image on the replica. However, databases and applications might time out if the I/O is held for too long.

Local Replication Technologies

Host-based, storage array-based, and network-based replications are the major technologies used for local replication. File system replication and LVM-based replication are examples of host-based local replication. Storage array-based replication can be implemented with distinct solutions, namely, full-volume mirroring, pointer-based full-volume replication, and pointer-based virtual replication. Continuous data protection (CDP) (covered in section "11.4.3 Network-Based Local Replication") is an example of network-based replication.

Host-Based Local Replication

LVM-based replication and file system (FS) snapshot are two common methods of host-based local replication.

LVM-Based Replication

In *LVM-based replication*, the logical volume manager is responsible for creating and controlling the host-level logical volumes. An LVM has three components: physical volumes (physical disk), volume groups, and logical volumes. A *volume group* is created by grouping one or more physical volumes. *Logical volumes* are created within a given volume group. A volume group can have multiple logical volumes.

In LVM-based replication, each *logical block* in a logical volume is mapped to two physical blocks on two different physical volumes, as shown in Figure 11-5. An application write to a logical volume is written to the two physical volumes by the LVM device driver. This is also known as *LVM mirroring*. Mirrors can be split, and the data contained therein can be independently accessed.

Advantages of LVM-Based Replication

The LVM-based replication technology is not dependent on a vendor-specific storage system. Typically, LVM is part of the operating system, and no additional license is required to deploy LVM mirroring.

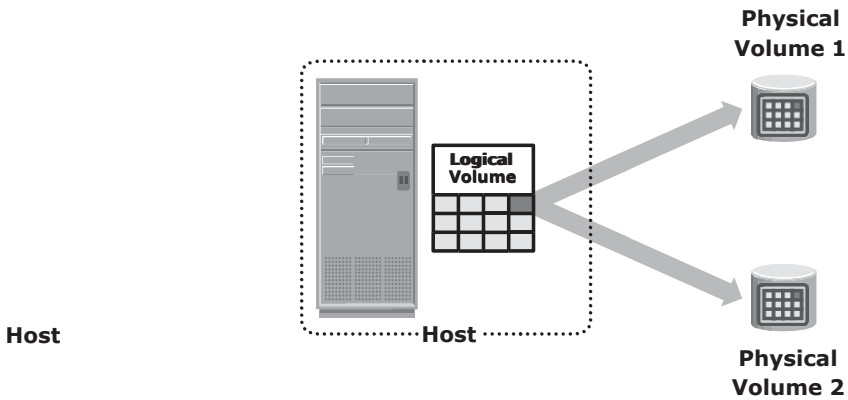


Figure 11-5: LVM-based mirroring

Limitations of LVM-Based Replication

Every write generated by an application translates into two writes on the disk, and thus, an additional burden is placed on the host CPU. This can degrade application performance. Presenting an LVM-based local replica to another host is usually not possible because the replica will still be part of the volume group, which is usually accessed by one host at any given time.

Tracking changes to the mirrors and performing incremental resynchronization operations is also a challenge because all LVMs do not support incremental resynchronization. If the devices are already protected by some level of RAID on the array, then the additional protection that the LVM mirroring provides is unnecessary. This solution does not scale to provide replicas of federated databases and applications. Both the replica and source are stored within the same volume group. Therefore, the replica might become unavailable if there is an error in the volume group. If the server fails, both the source and replica are unavailable until the server is brought back online.

File System Snapshot

A file system (FS) snapshot is a pointer-based replica that requires a fraction of the space used by the production FS. This snapshot can be implemented by either FS or by LVM. It uses the Copy on First Write (CoFW) principle to create snapshots.

When a snapshot is created, a bitmap and blockmap are created in the metadata of the Snap FS. The bitmap is used to keep track of blocks that are changed on the production FS after the snap creation. The blockmap is used to indicate the exact address from which the data is to be read when the data is accessed from the Snap FS. Immediately after the creation of the FS snapshot, all reads from the snapshot are actually served by reading the production FS. In a CoFW mechanism, if a write I/O is issued to the production FS for the first time after the creation of a snapshot, the I/O is held and the original data of production FS corresponding to that location is moved to the Snap FS. Then, the write is allowed to the production FS. The bitmap and blockmap are updated accordingly. Subsequent writes to the same location do not initiate the CoFW activity. To read from the Snap FS, the bitmap is consulted. If the bit is 0, then the read is directed to the production FS. If the bit is 1, then the block address is obtained from the blockmap, and the data is read from that address on the Snap FS. Read requests from the production FS work as normal.

Figure 11-6 illustrates the write operations to the production file system. For example, a write data "C" occurs on block 3 at the production FS, which currently holds data "c"' The snapshot application holds the I/O to the production FS and first copies the old data "c" to an available data block on the Snap FS. The bitmap and blockmap values for block 3 in the production FS are changed in the snap metadata. The bitmap of block 3 is changed to 1, indicating that this block has changed on the production FS. The block map of block 3 is changed and indicates the block number where the data is written in Snap FS, (in this case block 2). After this is done, the I/Os to the production FS are allowed to complete. Any subsequent writes to block 3 on the production FS occur as normal, and it does not initiate the CoFW operation. Similarly, if an I/O is issued to block 4 on the production FS to change the value of data "d" to "D," the snapshot application holds the I/O to the production FS and copies the old data to an available data block on the Snap FS. Then it changes the bitmap of block 4 to 1, indicating that the data block has changed on the production FS. The blockmap for block 4 indicates the block number where the data can be found on the Snap FS, in this case, data block 1 of the Snap FS. After this is done, the I/O to the production FS is allowed to complete.

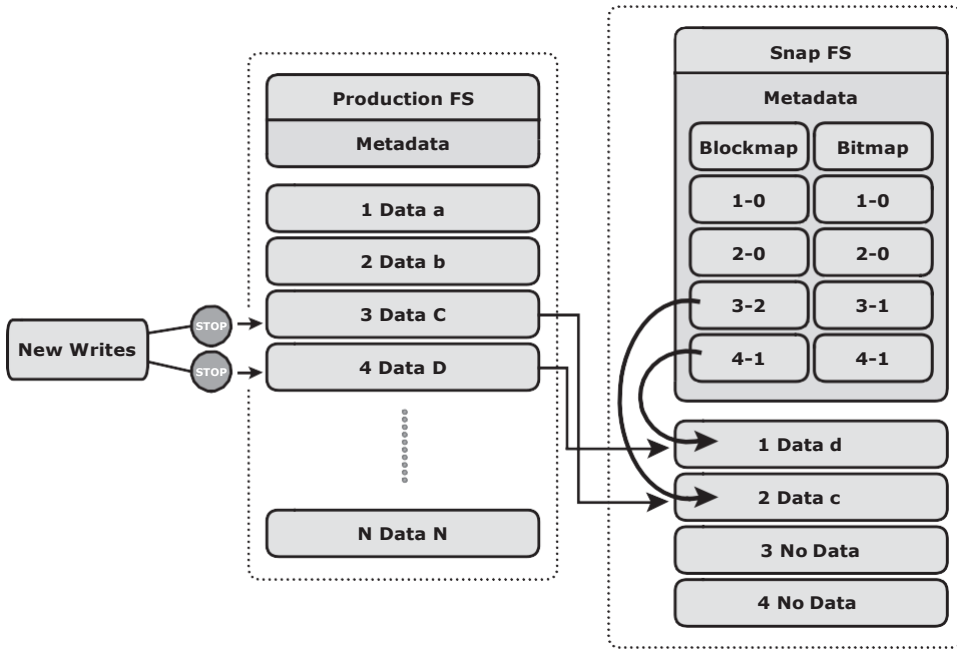


Figure 11-6: Write to production FS

Storage Array-Based Local Replication

In *storage array-based local replication*, the array-operating environment performs the local replication process. The host resources, such as the CPU and memory, are not used in the replication process. Consequently, the host is not burdened by the replication operations. The replica can be accessed by an alternative host for other business operations.

In this replication, the required number of replica devices should be selected on the same array and then data should be replicated between the source-replica pairs. Figure 11-7 shows a storage array-based local replication, where the source and target are in the same array and accessed by different hosts.

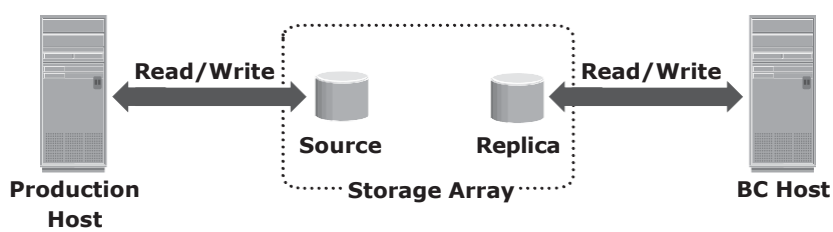
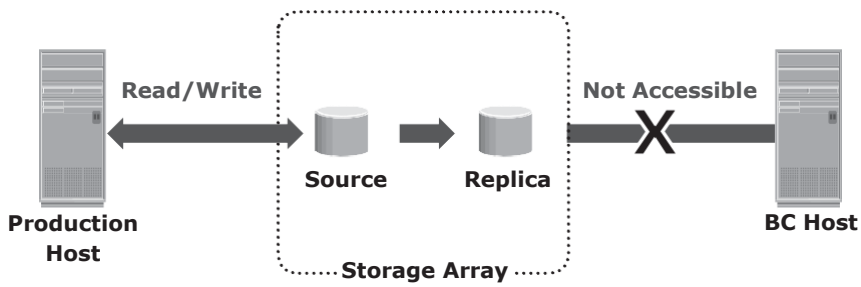


Figure 11-7: Storage array-based local replication

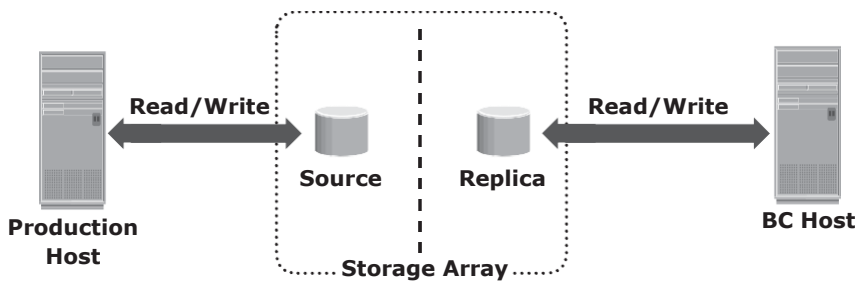
Storage array-based local replication is commonly implemented in three ways: full-volume mirroring, pointer-based full-volume replication, and pointer-based virtual replication. Replica devices are also referred as target devices, accessible by other hosts.

Full-Volume Mirroring

In *full-volume mirroring*, the target is attached to the source and established as a mirror of the source (Figure 11-8 [a]). The data on the source is copied to the target. New updates to the source are also updated on the target. After all the data is copied and both the source and the target contain identical data, the target can be considered as a mirror of the source.



(a) Full Volume Mirroring with Source Attached to Replica



(b) Full Volume Mirroring with Source Detached from Replica

Figure 11-8: Full-volume mirroring

While the target is attached to the source it remains unavailable to any other host. However, the production host continues to access the source.

After the synchronization is complete, the target can be detached from the source and made available for other business operations. Figure 11-8 (b) shows full-volume mirroring when the target is detached from the source. Both the source and the target can be accessed for read and write operations by the production and business continuity hosts respectively.

After detaching from the source, the target becomes a point-in-time (PIT) copy of the source. The PIT of a replica is determined by the time when the target is detached from the source. For example, if the time of detachment is 4:00 p.m., the PIT for the target is 4:00 p.m.

After detachment, changes made to both the source and replica can be tracked at some predefined granularity. This enables incremental resynchronization (source to target) or incremental restore (target to source). The granularity of the data change can range from 512 byte blocks to 64 KB blocks or higher.

Pointer-Based, Full-Volume Replication

Another method of array-based local replication is *pointer-based full-volume replication*. Similar to full-volume mirroring, this technology can provide full copies of the source data on the targets. Unlike full-volume mirroring, the target is immediately accessible by the BC host after the replication session is activated. Therefore, data synchronization and detachment of the target is not required to access it. Here, the time of replication session activation defines the PIT copy of the source.

Pointer-based, full-volume replication can be activated in either Copy on First Access (CoFA) mode or Full Copy mode. In either case, at the time of activation, a protection bitmap is created for all data on the source devices. The protection bitmap keeps track of the changes at the source device. The pointers on the target are initialized to map the corresponding data blocks on the source. The data is then copied from the source to the target based on the mode of activation.

In CoFA, after the replication session is initiated, the data is copied from the source to the target only when the following condition occurs:

- A write I/O is issued to a specific address on the source for the first time.
- A read or write I/O is issued to a specific address on the target for the first time.

When a write is issued to the source for the first time after replication session activation, the original data at that address is copied to the target. After this operation, the new data is updated on the source. This ensures that the original data at the point-in-time of activation is preserved on the target (see Figure 11-9). When a read is issued to the target for the first time after replication session activation, the original data is copied from the source to the target and is made available to the BC host (see Figure 11-10).

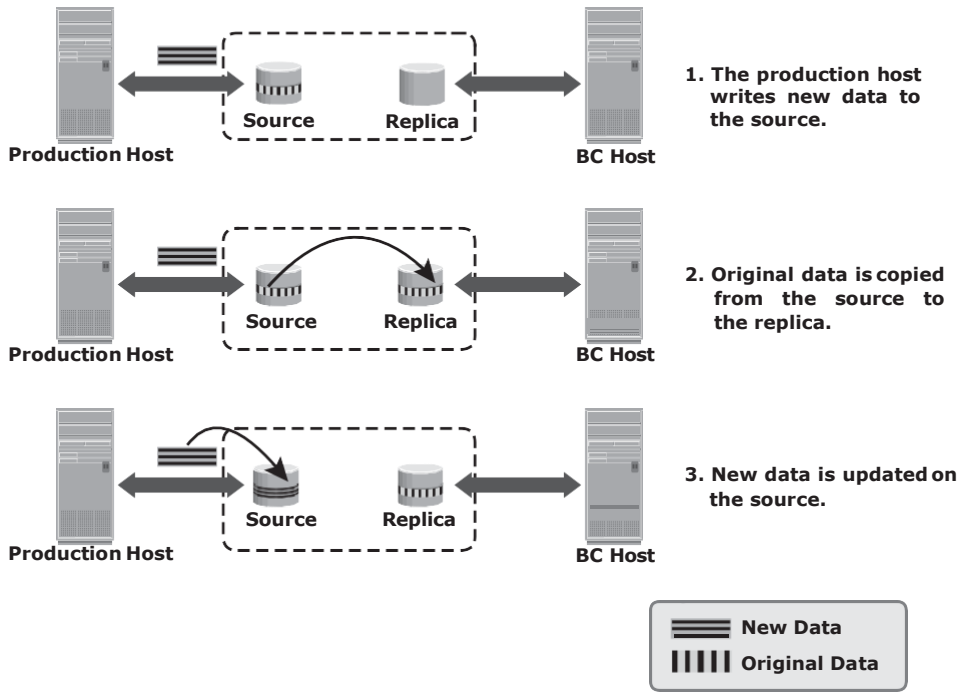


Figure 11-9: Copy on first access (CoFA) — write to source

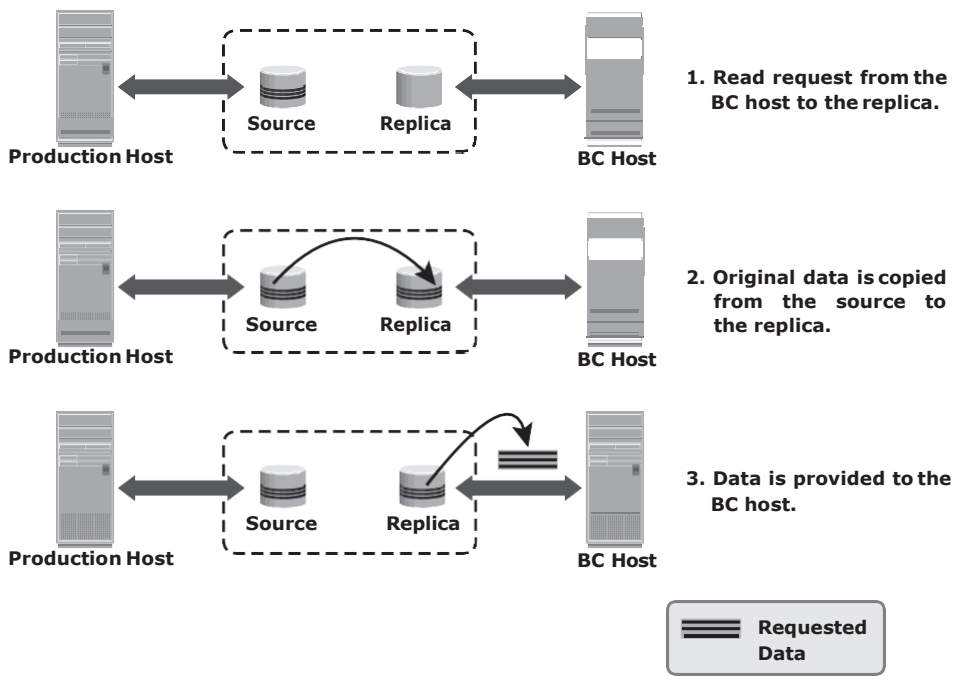


Figure 11-10: Copy on first access (CoFA) — read from target

When a write is issued to the target for the first time after the replication session activation, the original data is copied from the source to the target. After this, the new data is updated on the target (see Figure 11-11).

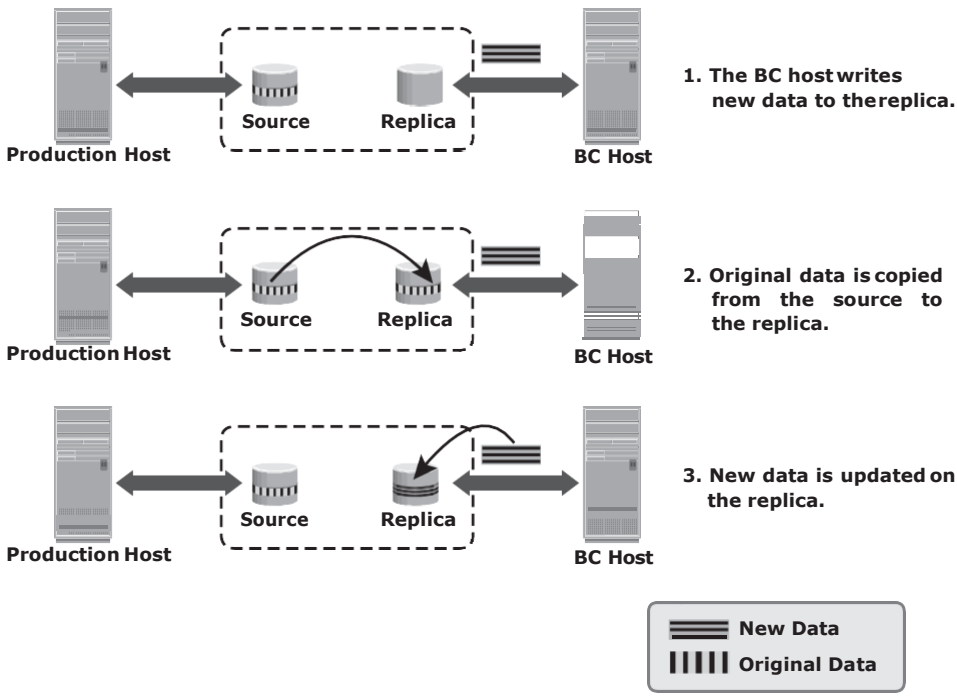


Figure 11-11: Copy on first access (CoFA) — write to target

In all cases, the protection bit for the data block on the source is reset to indicate that the original data has been copied over to the target. The pointer to the source data can now be discarded. Subsequent writes to the same data block on the source, and the reads or writes to the same data blocks on the target, do not trigger a copy operation, therefore this method is termed "Copy on First Access."

If the replication session is terminated, then the target device has only the data that was accessed until the termination, not the entire contents of the source at the point-in-time. In this case, the data on the target cannot be used for restore because it is not a full replica of the source.

In a Full Copy mode, all data from the source is copied to the target in the background. Data is copied regardless of access. If access to a block that has not yet been copied to the target is required, this block is preferentially copied to the target. In a complete cycle of the Full Copy mode, all data from the source is copied to the target. If the replication session is terminated now,

the target contains all the original data from the source at the point-in-time of activation. This makes the target a viable copy for restore or other business continuity operations.

The key difference between a pointer-based, Full Copy mode and full-volume mirroring is that the target is immediately accessible upon replication session activation in the Full Copy mode. Both the full-volume mirroring and pointer-based full-volume replication technologies require the target devices to be at least as large as the source devices. In addition, full-volume mirroring and pointer-based full-volume replication in the Full Copy mode can provide incremental resynchronization and restore capabilities.

Pointer-Based Virtual Replication

In *pointer-based virtual replication*, at the time of the replication session activation, the target contains pointers to the location of the data on the source. The target does not contain data at any time. Therefore, the target is known as a *virtual replica*. Similar to pointer-based full-volume replication, the target is immediately accessible after the replication session activation. A protection bitmap is created for all data blocks on the source device. Granularity of data blocks can range from 512 byte blocks to 64 KB blocks or greater.

Pointer-based virtual replication uses the CoFW technology. When a write is issued to the source for the first time after the replication session activation, the original data at that address is copied to a predefined area in the array. This area is generally known as the *save location*. The pointer in the target is updated to point to this data in the save location. After this, the new write is updated on the source. This process is illustrated in Figure 11-12.

When a write is issued to the target for the first time after replication session activation, the data is copied from the source to the save location, and the pointer is updated to the data in the save location. Another copy of the original data is created in the save location before the new write is updated on the save location. Subsequent writes to the same data block on the source or target do not trigger a copy operation. This process is illustrated in Figure 11-13.

When reads are issued to the target, unchanged data blocks since the session activation are read from the source, whereas data blocks that have changed are read from the save location.

Data on the target is a combined view of unchanged data on the source and data on the save location. Unavailability of the source device invalidates the data on the target. The target contains only pointers to the data, and therefore, the physical capacity required for the target is a fraction of the source device. The capacity required for the save location depends on the amount of the expected data change.

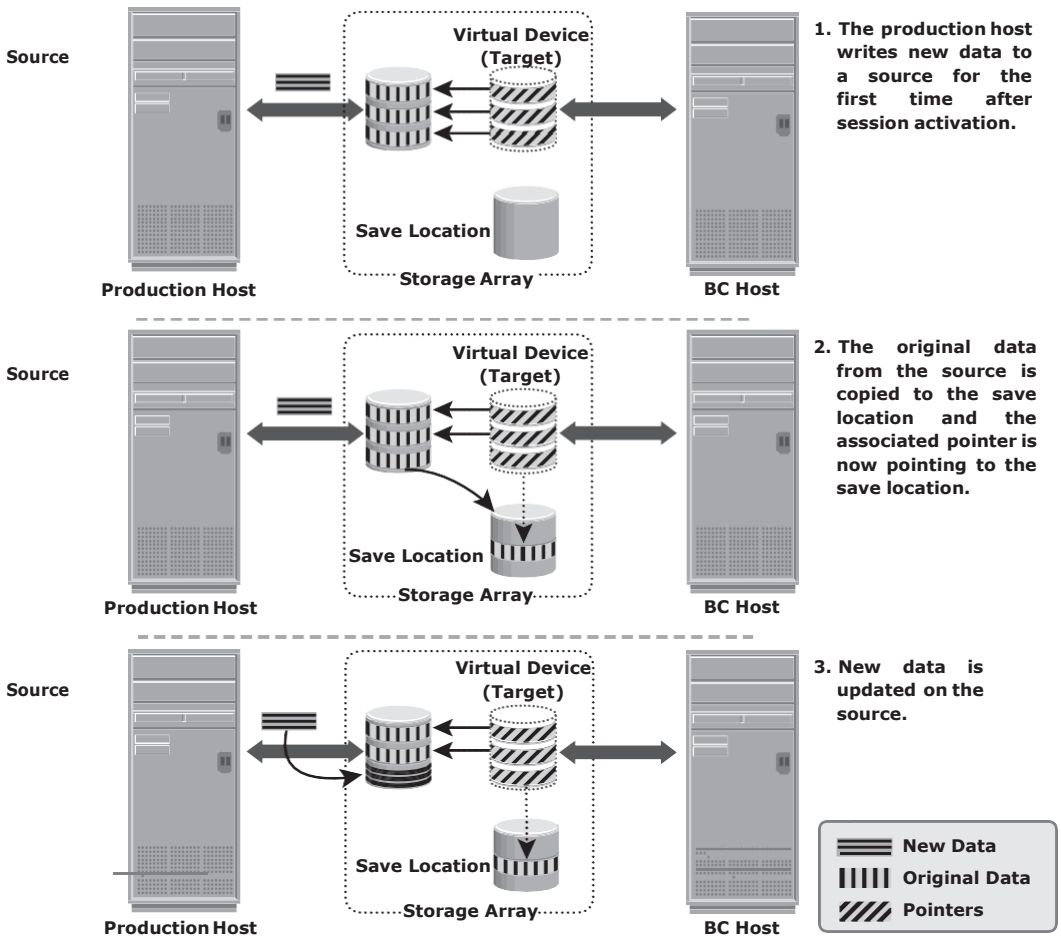


Figure 11-12: Pointer-based virtual replication — write to source

Network-Based Local Replication

In network-based replication, the replication occurs at the network layer between the hosts and storage arrays. Network-based replication combines the benefits of array-based and host-based replications. By offloading replication from servers and arrays, network-based replication can work across a large number of server platforms and storage arrays, making it ideal for highly heterogeneous environments. *Continuous data protection* (CDP) is a technology used for network-based local and remote replications. CDP for remote replication is detailed in Chapter 12.

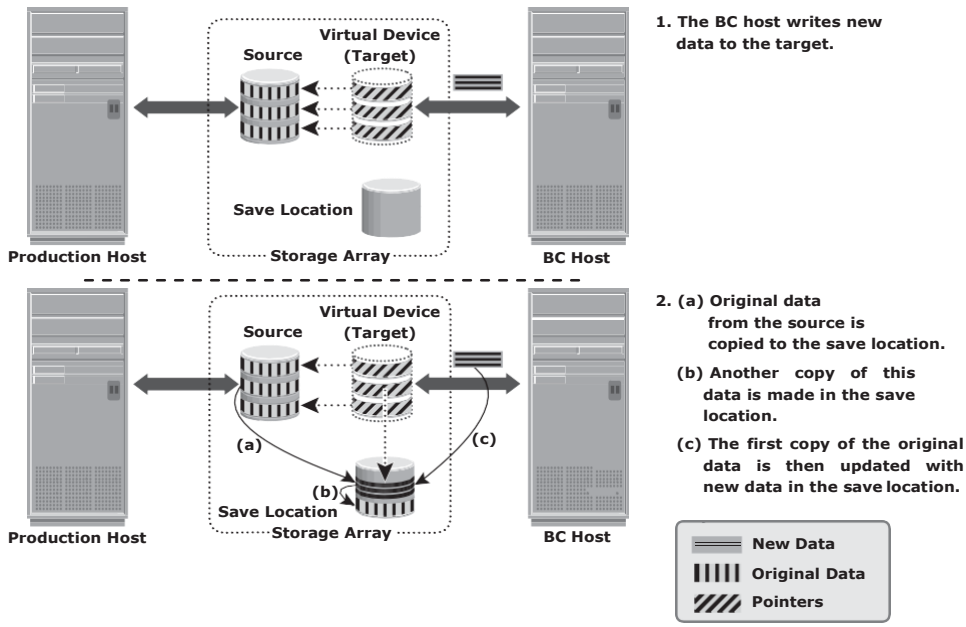


Figure 11-13: Pointer-based virtual replication — write to target

Continuous Data Protection

In a data center environment, mission-critical applications often require instant and unlimited data recovery points. Traditional data protection technologies offer limited recovery points. If data loss occurs, the system can be rolled back only to the last available recovery point. Mirroring offers continuous replication; however, if logical corruption occurs to the production data, the error might propagate to the mirror, which makes the replica unusable. In normal operation, CDP provides the ability to restore data to any previous PIT. It enables this capability by tracking all the changes to the production devices and maintaining consistent point-in-time images.

In CDP, data changes are continuously captured and stored in a separate location from the primary storage. Moreover, RPOs are random and do not need to be defined in advance. With CDP, recovery from data corruption poses no problem because it allows going back to a PIT image prior to the data corruption incident. CDP uses a *journal volume* to store all data changes on the primary storage. The journal volume contains all the data that has changed from the time the replication session started. The amount of space that is configured for the journal determines how far back the recovery points can go. CDP is

typically implemented using *CDP appliance* and *write splitters*. CDP implementation may also be host-based, in which CDP software is installed on a separate host machine.

CDP appliance is an intelligent hardware platform that runs the CDP software and manages local and remote data replications. Write splitters intercept writes to the production volume from the host and split each write into two copies. Write splitting can be performed at the host, fabric, or storage array.

CDP Local Replication Operation

Figure 11-14 describes CDP local replication. In this method, before the start of replication, the replica is synchronized with the source and then the replication process starts. After the replication starts, all the writes to the source are split into two copies. One of the copies is sent to the CDP appliance and the other to the production volume. When the CDP appliance receives a copy of a write, it is written to the journal volume along with its timestamp. As a next step, data from the journal volume is sent to the replica at predefined intervals.

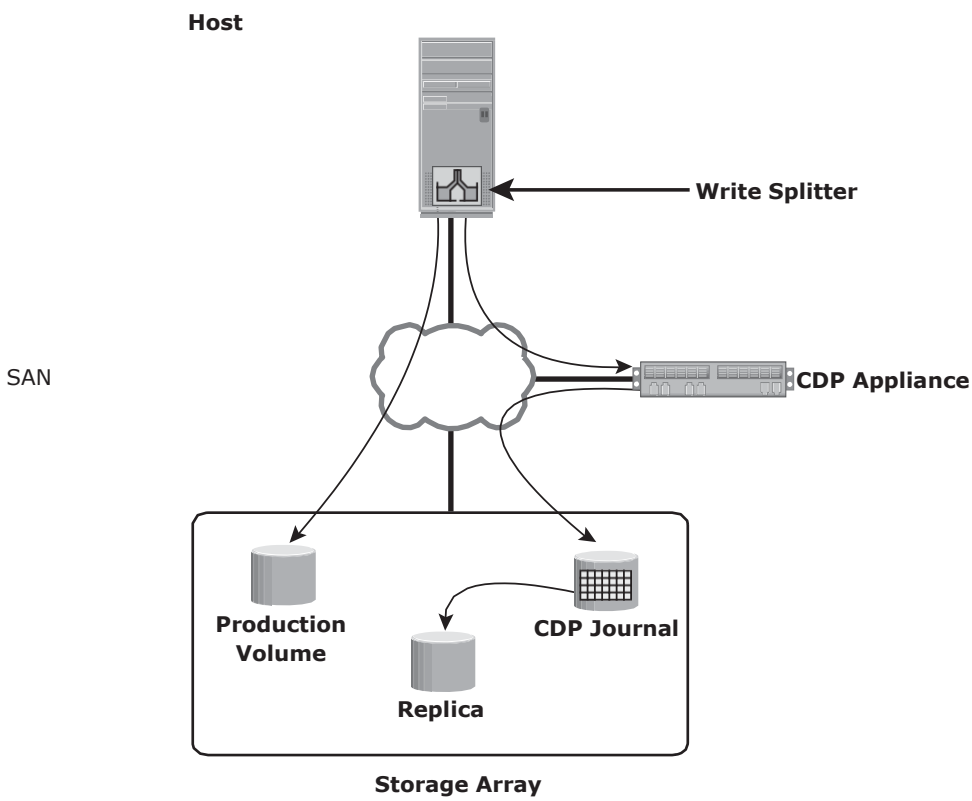


Figure 11-14: Continuous data protection — local replication

While recovering data to the source, the CDP appliance restores the data from the replica and applies journal entries up to the point in time chosen for recovery.

Tracking Changes to Source and Replica

Updates can occur on the source device after the creation of PIT local replicas. If the primary purpose of local replication is to have a viable PIT copy for data recovery or restore operations, then the replica devices should not be modified. Changes can occur on the replica device if it is used for other business operations. To enable incremental resynchronization or restore operations, changes to both the source and replica devices after the PIT should be tracked. This is typically done using bitmaps, where each bit represents a block of data. The data block sizes can range from 512 bytes to 64 KB or greater. For example, if the block size is 32 KB, then a 1-GB device would require 32,768 bits (1 GB divided by 32 KB). The size of the bitmap would be 4 KB. If the data in any 32 KB block is changed, the corresponding bit in the bitmap is flagged. If the block size is reduced for tracking purposes, then the bitmap size increases correspondingly.

The bits in the source and target bitmaps are all set to 0 (zero) when the replica is created. Any changes to the source or replica are then flagged by setting the appropriate bits to 1 in the bitmap. When resynchronization or restore is required, a *logical OR* operation between the source bitmap and the target bitmap is performed. The bitmap resulting from this operation references all blocks that have been modified in either the source or replica (see Figure 11-15). This enables an optimized resynchronization or a restore operation because it eliminates the need to copy all the blocks between the source and the replica. The direction of data movement depends on whether a resynchronization or a restore operation is performed.

If resynchronization is required, changes to the replica are overwritten with the corresponding blocks from the source. In this example, that would be blocks labeled 2, 3, and 7 on the replica.

If a restore is required, changes to the source are overwritten with the corresponding blocks from the replica. In this example, that would be blocks labeled 0, 3, and 5 on the source. In either case, changes to both the source and the target cannot be simultaneously preserved.

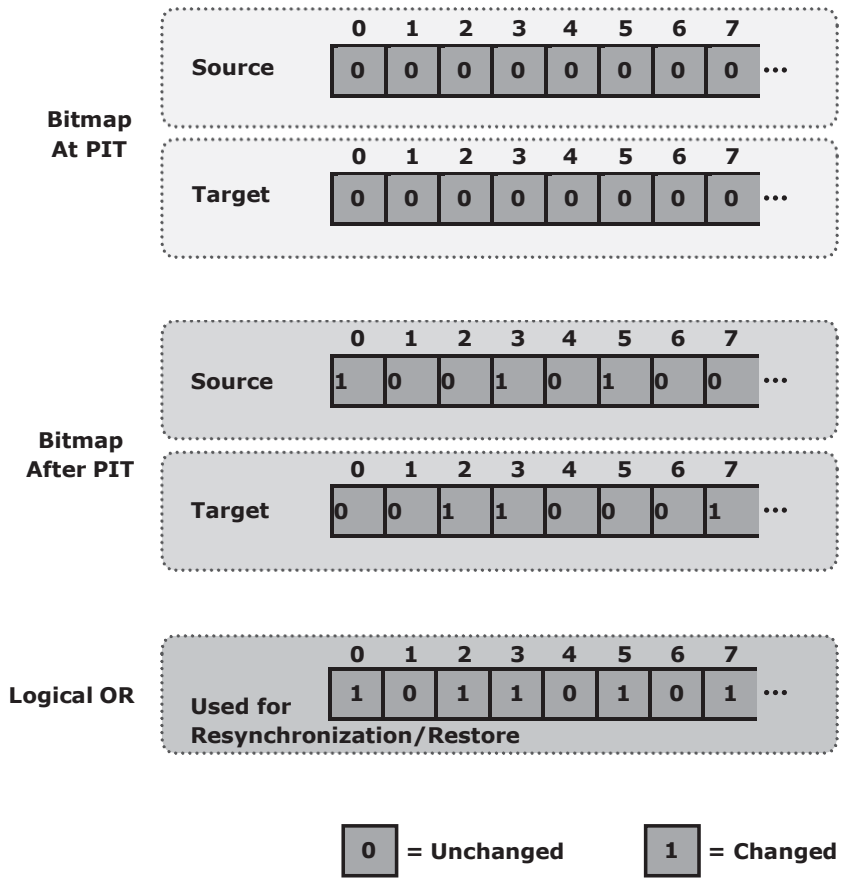


Figure 11-15: Tracking changes

Restore and Restart Considerations

Local replicas are used to restore data to production devices. Alternatively, applications can be restarted using the consistent PIT replicas.

Replicas are used to restore data to the production devices if logical corruption of data on production devices occurs — that is, the devices are available but the data on them is invalid. Examples of logical corruption include accidental deletion of data (tables or entries in a database), incorrect data entry, and incorrect data updates. Restore operations from a replica are incremental and provide a small RTO. In some instances, the applications can be resumed on the production devices prior to the completion of the data copy. Prior to the restore operation, access to production and replica devices should be stopped.

Production devices might also become unavailable due to physical failures, such as the production server or physical drive failure. In this case, applications

can be restarted using the data on the latest replica. As a protection against further failures, a Gold Copy (another copy of replica device) of the replica device should be created to preserve a copy of data in the event of failure or corruption of the replica devices. After the issue has been resolved, the data from the replica devices can be restored back to the production devices.

Full-volume replicas (both full-volume mirrors and pointer-based in Full Copy mode) can be restored to the original source devices or to a new set of source devices. Restores to the original source devices can be incremental, but restores to a new set of devices are full-volume copy operations.

In pointer-based virtual and pointer-based full-volume replication in CoFA mode, access to data on the replica is dependent on the health and accessibility of the source volumes. If the source volume is inaccessible for any reason, these replicas cannot be used for a restore or a restart operation.

Table 11-1 presents a comparative analysis of the various storage array-based replication technologies.

Table 11-1: Comparison of Local Replication Technologies

FACTOR	FULL-VOLUME MIRRORING	POINTER-BASED, FULL-VOLUME REPLICATION	POINTER-BASED VIRTUAL REPLICATION
Performance impact on source due to replica	No impact	CoFA mode — some impact Full copy mode — no impact	High impact
Size of target	At least the same as the source	At least the same as the source	Small fraction of the source
Availability of source for restoration	Not required	CoFA mode — required Full copy mode — not required	Required
Accessibility to target	Only after synchronization and detachment from the source	Immediately accessible	Immediately accessible

Creating Multiple Replicas

Most storage array-based replication technologies enable source devices to maintain replication relationships with multiple targets. Changes made to the source and each of the targets can be tracked. This enables incremental resynchronization of the targets. Each PIT copy can be used for different BC activities and as a restore point.

Figure 11-16 shows an example in which a copy is created every 6 hours from the same source.

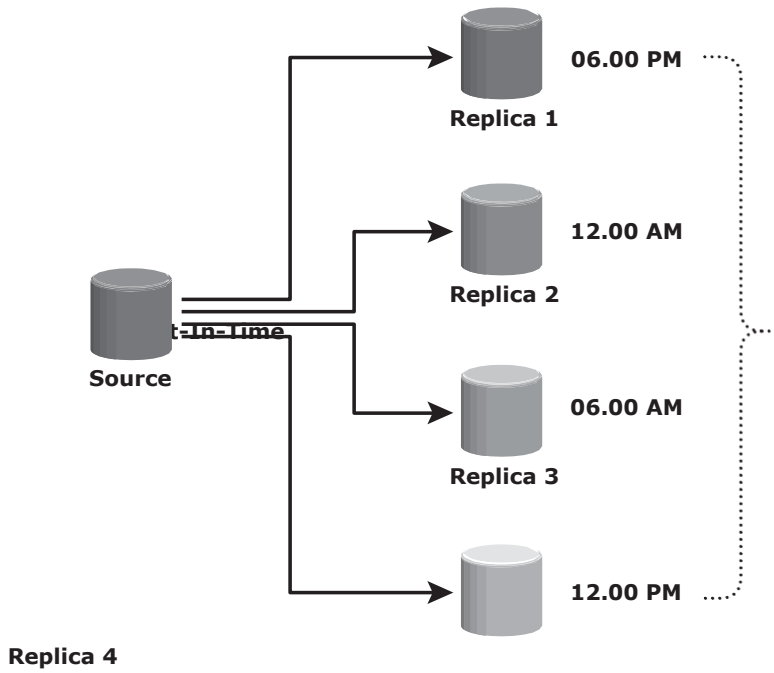


Figure 11-16: Multiple replicas created at different PIT

If the source is corrupted, the data can be restored from the latest PIT copy. The maximum RPO in the example shown in Figure 11-16 is 6 hours. More frequent replicas further reduce the RPO.

Array-based local replication technologies also enable the creation of multiple *concurrent* PIT replicas. In this case, all replicas contain identical data. One or more of the replicas can be set aside for restore operations. Decision support activities can be performed using the other replicas.

Local Replication in a Virtualized Environment

The discussion so far has focused on local replication in a physical infra-structure environment. In a virtualized environment, along with replicating storage volumes, virtual machine (VM) replication is also required. Typically, local replication of VMs is performed by the hypervisor at the compute level. However, it can also be performed at the storage level using array-based local replication, similar to the physical environment. In the array-based method,

the LUN on which the VMs reside is replicated to another LUN in the same array. For hypervisor-based local replication, two options are available: VM Snapshot and VM Clone.

VM Snapshot captures the state and data of a running virtual machine at a specific point in time. The VM state includes VM files, such as BIOS, network configuration, and its power state (powered-on, powered-off, or suspended). The VM data includes all the files that make up the VM, including virtual disks and memory. A VM Snapshot uses a separate delta file to record all the changes to the virtual disk since the snapshot session is activated. Snapshots are useful when a VM needs to be reverted to the previous state in the event of logical corruptions. Reverting a VM to a previous state causes all settings configured in the guest OS to be reverted to that PIT when that snapshot was created. There are some challenges associated with the VM Snapshot technology. It does not support data replication if a virtual machine accesses the data by using raw disks. Also, using the hypervisor to perform snapshots increases the load on the compute and impacts the compute performance.

VM Clone is another method that creates an identical copy of a virtual machine. When the cloning operation is complete, the clone becomes a separate VM from its parent VM. The clone has its own MAC address, and changes made to a clone do not affect the parent VM. Similarly, changes made to the parent VM do not appear in the clone. VM Clone is a useful method when there is a need to deploy many identical VMs. Installing guest OS and applications on multiple VMs is a time-consuming task; VM Clone helps to simplify this process.

Concepts in Practice: EMC TimeFinder, EMC SnapView, and EMC RecoverPoint

EMC offers a range of storage array-based local replication solutions for different storage arrays. For the Symmetrix array, the EMC TimeFinder family of products is used for full-volume and pointer-based local replication. EMC SnapView is the solution for EMC VNX storage arrays. EMC RecoverPoint is a network-based replication solution. Visit `www.emc.com` for the latest information.

EMC TimeFinder

The TimeFinder family of products consists of two base solutions and four add-on solutions. The base solutions are TimeFinder/Clone and TimeFinder/Snap. The add-on solutions are TimeFinder/Clone Emulation, TimeFinder/Consistency Groups, TimeFinder/Exchange Integration Module, and TimeFinder/SQL Integration Module.

TimeFinder is available for both open systems and mainframes. The base solutions support the different storage array-based local replication technologies discussed in this chapter. The add-on solutions are customizations of the replicas for specific application or database environments.

TimeFinder/Clone

TimeFinder/Clone creates a PIT copy of the source volume that can be used for backups, decision support, or any other process that requires parallel access to production data. TimeFinder/Clone uses pointer-based full-volume replication technology. TimeFinder/Clone allows creating up to 16 active clones from a single production device, and all the clones are available immediately for read and write access.

TimeFinder/Snap

TimeFinder/Snap creates space-saving, logical PIT images called snapshots. The snapshots are not full copies but contain pointers to the source data. The target device used by TimeFinder/Snap is called a virtual device (VDEV). It keeps pointers to the source device or SAVE devices. The SAVE devices keep the point-in-time data that has changed on the source after the start of the replication session. TimeFinder/Snap allows creating multiple snapshots, up to 128, from a single source device.

EMC SnapView

SnapView is an EMC VNX array-based local replication software that creates a pointer-based virtual copy and full-volume mirror of the source using SnapView snapshot and SnapView clone respectively.

SnapView Snapshot

A SnapView snapshot is not a full copy of the production volume; it is a logical view of the production volume based on the time at which the snapshot was created. Snapshots are created in seconds and can be retired when no longer needed. A snapshot rollback feature provides instant restore to the source volume. The key terminologies of SnapView snapshot are as follows:

- **SnapView session:** The SnapView snapshot mechanism is activated when a session starts and deactivated when a session stops. A snapshot appears "offline" until there is an active session. Multiple snapshots can be included in a session.

- **Reserved LUN pool:** This is a private area, also called a save area, used to contain Copy on First Write (CoFW) data. The "Reserved" part of the name refers to the fact that the LUNs are reserved and therefore cannot be assigned to a host.

SnapView Clone

SnapView Clones are full-volume copies that require the same disk space as the source. These PIT copies can be used for other business operations, such as backup and testing. SnapView Clone enables incremental resynchronization between the source and replica. Clone fracture is the process of breaking off a clone from its source. After the clone is fractured, it becomes a PIT copy and available for other business operations.

EMC RecoverPoint

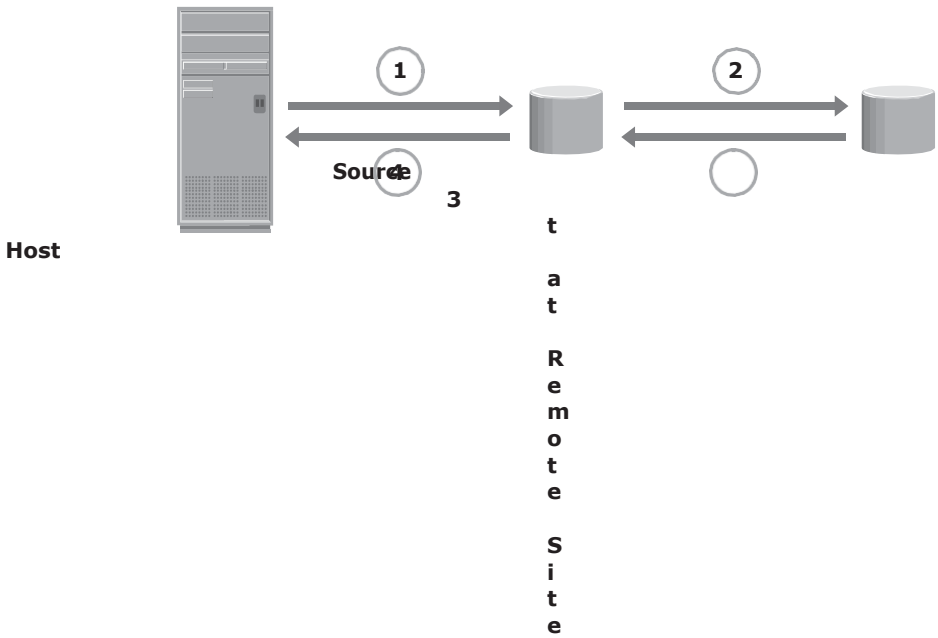
RecoverPoint is a high-performance, cost-effective, single product that provides local and remote data protection for both physical and virtual environments. It provides faster recovery and unlimited recovery points. RecoverPoint provides continuous data protection and performs replication between the LUNs that reside in one or more arrays at the same site. RecoverPoint uses lightweight splitting technology either at the application server, fabric, or arrays to mirror a write to a RecoverPoint appliance. The RecoverPoint family of products includes RecoverPoint/CL, RecoverPoint/EX, and RecoverPoint/SE.

RecoverPoint/CL is a replication product for a heterogeneous server and storage environment. It supports both EMC and non-EMC storage arrays. This product supports host-based, fabric-based, and array-based write splitters. RecoverPoint/EX supports replication between EMC storage arrays and enables only array-based write splitting. RecoverPoint/SE is a version of RecoverPoint targeted for VNX series arrays and enables only Windows-based host and array-based write splitting.

Modes of Remote Replication

The two basic modes of remote replication are synchronous and asynchronous. In *synchronous remote replication*, writes must be committed to the source and remote replica (or target), prior to acknowledging "write complete" to the host (see Figure 12-1). Additional writes on the source cannot occur until each preceding write has been completed and acknowledged. This ensures that data is identical on the source and replica at all times. Further, writes are transmitted to the remote site exactly in the order in which they are received

at the source. Therefore, write ordering is maintained. If a source-site failure occurs, synchronous remote replication provides zero or near-zero recovery-point objective (RPO).



1. The host writes data to the source.
2. Data from the source is replicated to the target at a remote site.
3. The target acknowledges back to the source.
4. The source acknowledges write complete to the host.

Figure 12-1: Synchronous replication

However, application response time is increased with synchronous remote replication because writes must be committed on both the source and target before sending the "write complete" acknowledgment to the host. The degree of impact on response time depends primarily on the distance between sites, bandwidth, and quality of service (QOS) of the network connectivity infrastructure. Figure 12-2 represents the network bandwidth requirement for synchronous replication. If the bandwidth provided for synchronous remote replication is less than the maximum write workload, there will be times during the day when the response time might be excessively elongated, causing applications to time out. The distances over which synchronous replication can be deployed depend on the application's capability to tolerate extensions in response time. Typically, it is deployed for distances less than 200 KM (125 miles) between

the two sites.

In *asynchronous remote replication*, a write is committed to the source and immediately acknowledged to the host. In this mode, data is buffered at the source and transmitted to the remote site later (see Figure 12-3).

Asynchronous replication eliminates the impact to the application's response time because the writes are acknowledged immediately to the source host. This enables deployment of asynchronous replication over distances ranging from

several hundred to several thousand kilometers between the primary and remote sites. Figure 12-4 shows the network bandwidth requirement for asynchronous replication. In this case, the required bandwidth can be provisioned equal to or greater than the average write workload. Data can be buffered during times when the bandwidth is not enough and moved later to the remote site. Therefore, sufficient buffer capacity should be provisioned.

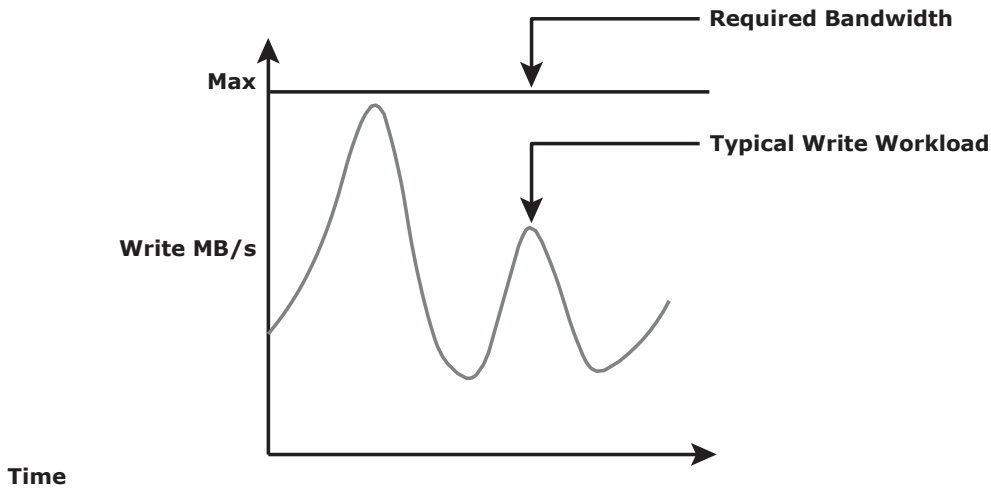
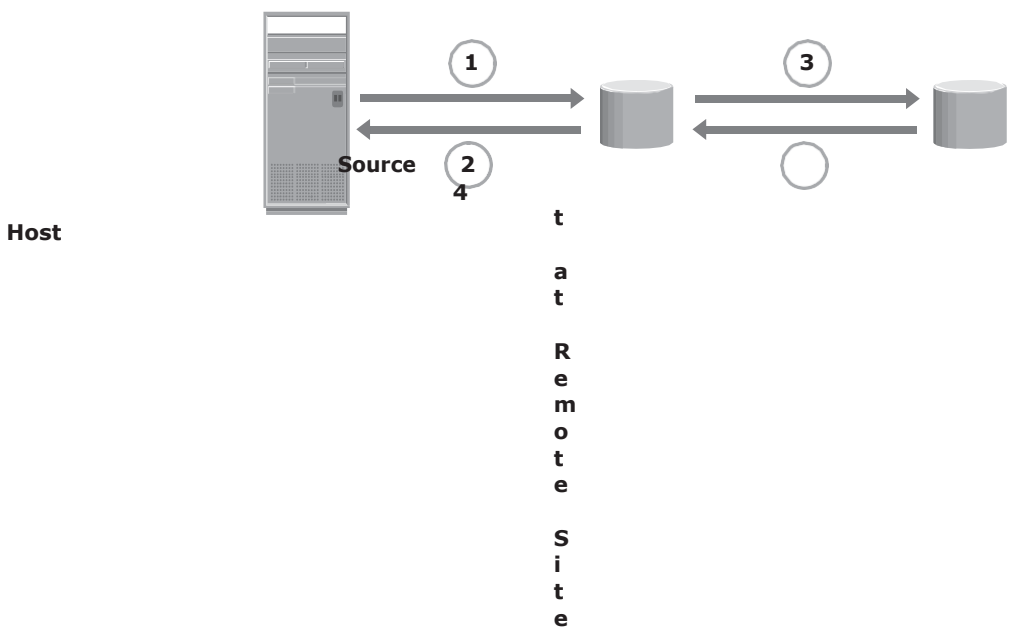


Figure 12-2: Bandwidth requirement for synchronous replication



1 **The host writes data to the source.**

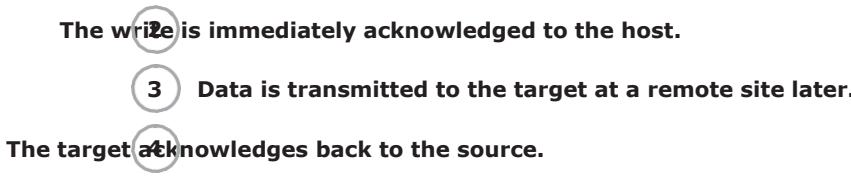


Figure 12-3: Asynchronous replication

In asynchronous replication, data at the remote site will be behind the source by at least the size of the buffer. Therefore, asynchronous remote replication provides a finite (nonzero) RPO disaster recovery solution. RPO depends on the size of the buffer, the available network bandwidth, and the write workload to the source.

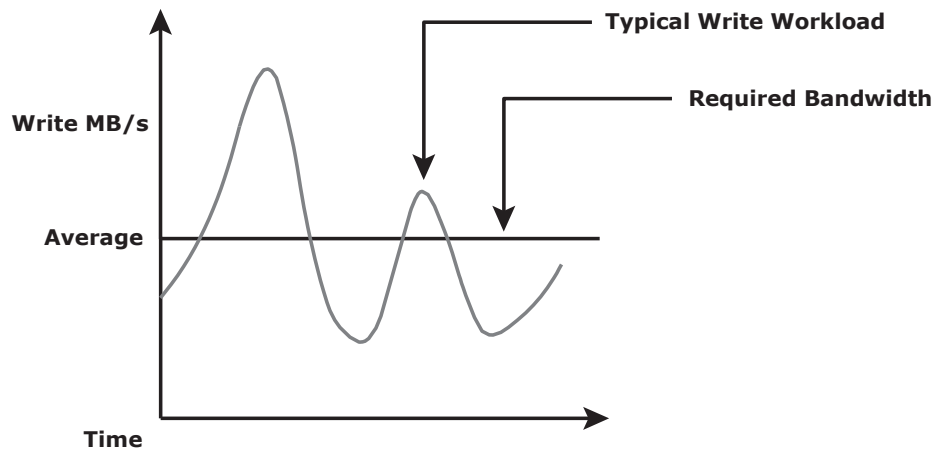


Figure 12-4: Bandwidth requirement for asynchronous replication

Asynchronous replication implementation can take advantage of *locality of reference* (repeated writes to the same location). If the same location is written multiple times in the buffer prior to transmission to the remote site, only the final version of the data is transmitted. This feature conserves link bandwidth.

In both synchronous and asynchronous modes of replication, only writes to the source are replicated; reads are still served from the source.

Remote Replication Technologies

Remote replication of data can be handled by the hosts or storage arrays. Other options include specialized network-based appliances to replicate data over the LAN or SAN. An advanced replication option such as three-site replication is discussed in section "12.3 Three-Site Replication."

Host-Based Remote Replication

Host-based remote replication uses the host resources to perform and manage the replication operation. There are two basic approaches to host-based remote replication: Logical volume manager (LVM) based replication and database replication via log shipping.

LVM-Based Remote Replication

LVM-based remote replication is performed and managed at the volume group level. Writes to the source volumes are transmitted to the remote host by the LVM. The LVM on the remote host receives the writes and commits them to the remote volume group.

Prior to the start of replication, identical volume groups, logical volumes, and file systems are created at the source and target sites. Initial synchronization of data between the source and replica is performed. One method to perform initial synchronization is to backup the source data and restore the data to the remote replica. Alternatively, it can be performed by replicating over the IP network. Until the completion of the initial synchronization, production work on the source volumes is typically halted. After the initial synchronization, production work can be started on the source volumes and replication of data can be performed over an existing standard IP network (see Figure 12-5).

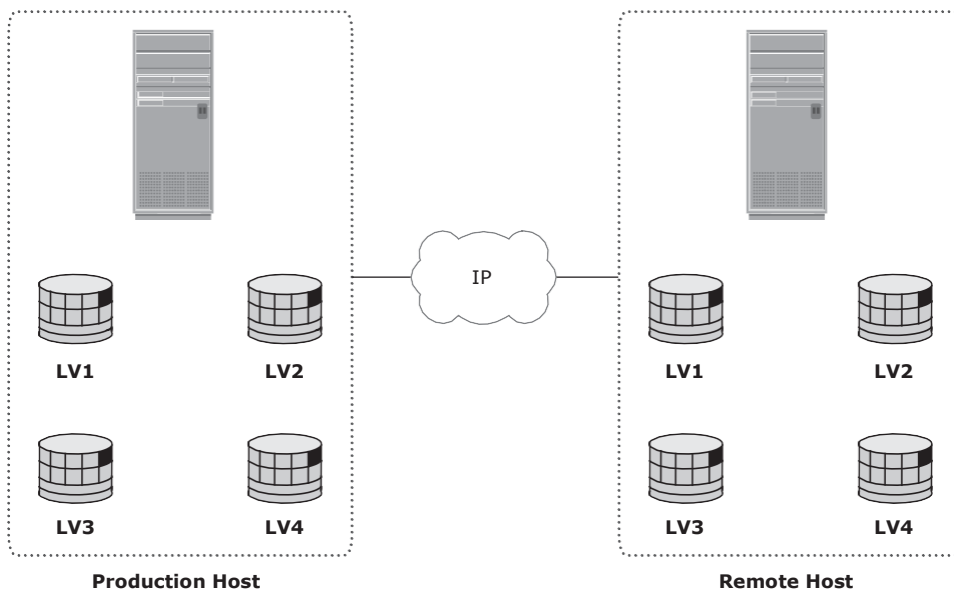


Figure 12-5: LVM-based remote replication

LVM-based remote replication supports both synchronous and asynchronous modes of replication. If a failure occurs at the source site, applications can be restarted on the remote host, using the data on the remote replicas.

LVM-based remote replication is independent of the storage arrays and therefore supports replication between heterogeneous storage arrays. Most operating

systems are shipped with LVMs, so additional licenses and specialized hardware are not typically required.

The replication process adds overhead on the host CPUs. CPU resources on the source host are shared between replication tasks and applications. This might cause performance degradation to the applications running on the host.

Because the remote host is also involved in the replication process, it must be continuously up and available.

Host-Based Log Shipping

Database replication via log shipping is a host-based replication technology supported by most databases. Transactions to the source database are captured in logs, which are periodically transmitted by the source host to the remote host (see Figure 12-6). The remote host receives the logs and applies them to the remote database.

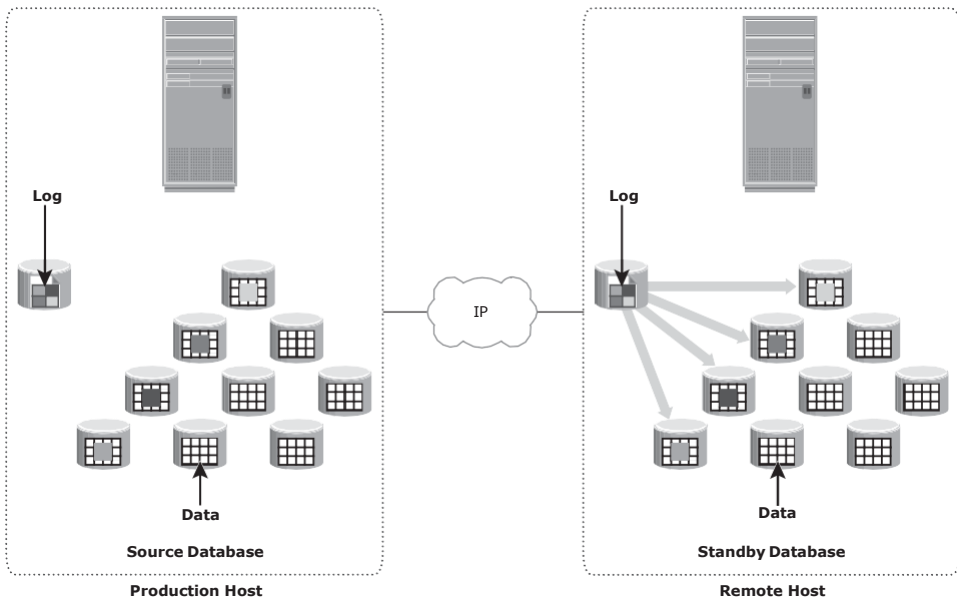


Figure 12-6: Host-based log shipping

Prior to starting production work and replication of log files, all relevant components of the source database are replicated to the remote site. This is done while the source database is shut down.

After this step, production work is started on the source database. The remote database is started in a standby mode. Typically, in standby mode, the database is not available for transactions.

All DBMSs switch log files at preconfigured time intervals or when a log file is full. The current log file is closed at the time of log switching, and a new log file is opened. When a log switch occurs, the closed log file is transmitted by the source host to the remote host. The remote host receives the log and updates the standby database.

This process ensures that the standby database is consistent up to the last committed log. RPO at the remote site is finite and depends on the size of the log and the frequency of log switching. Available network bandwidth, latency, rate of updates to the source database, and the frequency of log switching should be considered when determining the optimal size of the log file.

Similar to LVM-based remote replication, the existing standard IP network can be used for replicating log files. Host-based log shipping requires low network bandwidth because it transmits only the log files at regular intervals.

Storage Array-Based Remote Replication

In *storage array-based remote replication*, the array-operating environment and resources perform and manage data replication. This relieves the burden on the host CPUs, which can be better used for applications running on the host. A source and its replica device reside on different storage arrays. Data can be transmitted from the source storage array to the target storage array over a shared or a dedicated network.

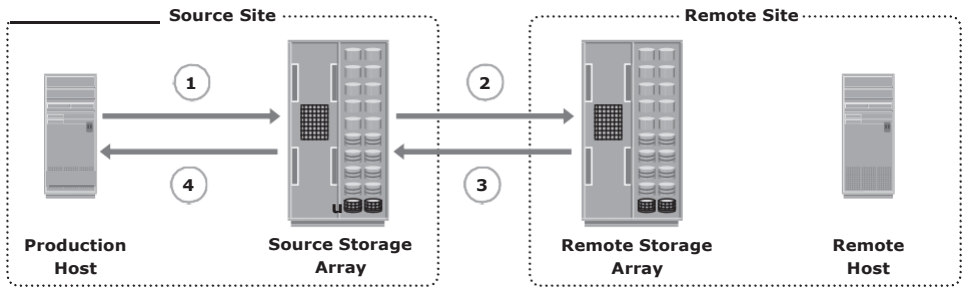
Replication between arrays may be performed in synchronous, asynchronous, or disk-buffered modes.

Synchronous Replication Mode

In array-based synchronous remote replication, writes must be committed to the source and the target prior to acknowledging "write complete" to the production host. Additional writes on that source cannot occur until each preceding write has been completed and acknowledged. Figure 12-7 shows the array-based synchronous remote replication process.

In the case of synchronous remote replication, to optimize the replication process and to minimize the impact on application response time, the write is placed on cache of the two arrays. The intelligent storage arrays destage these writes to the appropriate disks later.

If the network links fail, replication is suspended; however, production work can continue uninterrupted on the source storage array. The array operating environment keeps track of the writes that are not transmitted to the remote storage array. When the network links are restored, the accumulated data is transmitted to the remote storage array. During the time of network link outage, if there is a failure at the source site, some data will be lost, and the RPO at the target will not be zero.

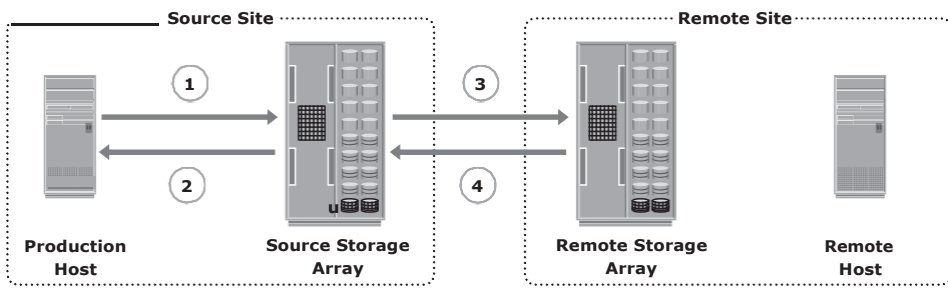


1. Write from the production host is received by the source storage array.
2. Write is then transmitted to the remote storage array.
3. Acknowledgment is sent to the source storage array by the remote storage array.
4. Source storage array signals write-completion to the production host.

Figure 12-7: Array-based synchronous remote replication

Asynchronous Replication Mode

In array-based *asynchronous remote replication mode*, as shown in Figure 12-8, a write is committed to the source and immediately acknowledged to the host. Data is buffered at the source and transmitted to the remote site later. The source and the target devices do not contain identical data at all times. The data on the target device is behind that of the source, so the RPO in this case is not zero.



1. The production host writes to the source storage array.
2. The source array immediately acknowledges the production host.
3. These writes are then transmitted to the target array.
4. After the writes are received by the target array, it sends an acknowledgment to the source array.

Figure 12-8: Array-based asynchronous remote replication

Similar to synchronous replication, asynchronous replication writes are placed in cache on the two arrays and are later destaged to the appropriate disks.

Some implementations of asynchronous remote replication maintain write ordering. A timestamp and sequence number are attached to each write when it is received by the source. Writes are then transmitted to the remote array, where they are committed to the remote replica in the exact order in which they were buffered at the source. This implicitly guarantees consistency of data on the remote replicas. Other implementations ensure consistency by leveraging the dependent write principle inherent in most DBMSs. In asynchronous remote replication, the writes are buffered for a predefined period of time. At the end of this duration, the buffer is closed, and a new buffer is opened for subsequent writes. All writes in the closed buffer are transmitted together and committed to the remote replica.

Asynchronous remote replication provides network bandwidth cost-savings because the required bandwidth is lower than the peak write workload. During times when the write workload exceeds the average bandwidth, sufficient buffer space must be configured on the source storage array to hold these writes.

Disk-Buffered Replication Mode

Disk-buffered replication is a combination of local and remote replication technologies. A consistent PIT local replica of the source device is first created. This is then replicated to a remote replica on the target array.

Figure 12-9 shows the sequence of operations in a disk-buffered remote replication. At the beginning of the cycle, the network links between the two arrays are suspended, and there is no transmission of data. While production application runs on the source device, a consistent PIT local replica of the source device is created. The network links are enabled, and data on the local replica in the source array transmits to its remote replica in the target array. After synchronization of this pair, the network link is suspended, and the next local replica of the source is created. Optionally, a local PIT replica of the remote device on the target array can be created. The frequency of this cycle of operations depends on the available link bandwidth and the data change rate on the source device. Because disk-buffered technology uses local replication, changes made to the source and its replica are possible to track. Therefore, all the resynchronization operations between the source and target can be done incrementally. When compared to synchronous and asynchronous replications, disk-buffered remote replication requires less bandwidth.

In disk-buffered remote replication, the RPO at the remote site is in the order of hours. For example, a local replica of the source device is created at 10:00 a.m., and this data transmits to the remote replica, which takes 1 hour to complete. Changes made to the source device after 10:00 a.m. are tracked. Another local replica of the source device is created at 11:00 a.m. by applying

track changes between the source and local replica (10:00 a.m. copy). During the next cycle of transmission (11:00 a.m. data), the source data has moved to 12:00 p.m. The local replica in the remote array has the 10:00 a.m. data until the 11:00 a.m. data is successfully transmitted to the remote replica. If there is a failure at the source site prior to the completion of transmission, then the worst-case RPO at the remote site would be 2 hours because the remote site has 10:00 a.m. data.

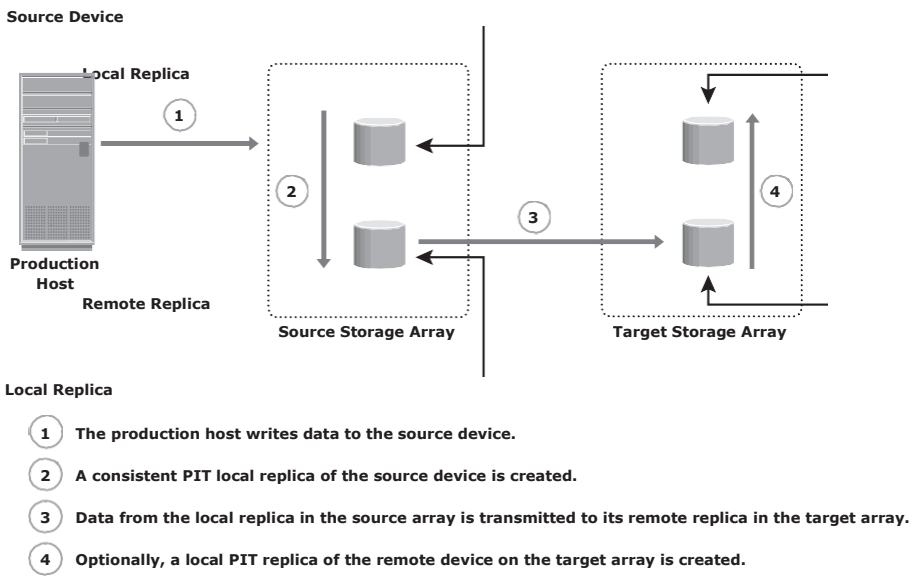


Figure 12-9: Disk-buffered remote replication

Network-Based Remote Replication

In network-based remote replication, the replication occurs at the network layer between the host and storage array. Continuous data protection technology, discussed in the previous chapter, also provides solutions for network-based remote replication.

CDP Remote Replication

In normal operation, CDP remote replication provides any-point-in-time recovery capability, which enables the target LUNs to be rolled back to any previous point in time. Similar to CDP local replication, CDP remote replication typically uses a *journal volume*, *CDP appliance*, or CDP software installed on a separate host (*host-based CDP*), and a *write splitter* to perform replication between sites. The CDP appliance is maintained at both source and remote sites.

Figure 12-10 describes CDP remote replication. In this method, the replica is synchronized with the source, and then the replication process starts. After the replication starts, all the writes from the host to the source are split into two copies. One of the copies is sent to the local CDP appliance at the source site, and the other copy is sent to the production volume. After receiving the write, the appliance at the source site sends it to the appliance at the remote site. Then, the write is applied to the journal volume at the remote site. For an asynchronous operation, writes at the source CDP appliance are accumulated, and redundant blocks are eliminated. Then, the writes are sequenced and stored with their corresponding timestamp. The data is then compressed, and a checksum is generated. It is then scheduled for delivery across the IP or FC network to the remote CDP appliance. After the data is received, the remote appliance verifies the checksum to ensure the integrity of the data. The data is then uncompressed and written to the remote journal volume. As a next step, data from the journal volume is sent to the replica at predefined intervals.

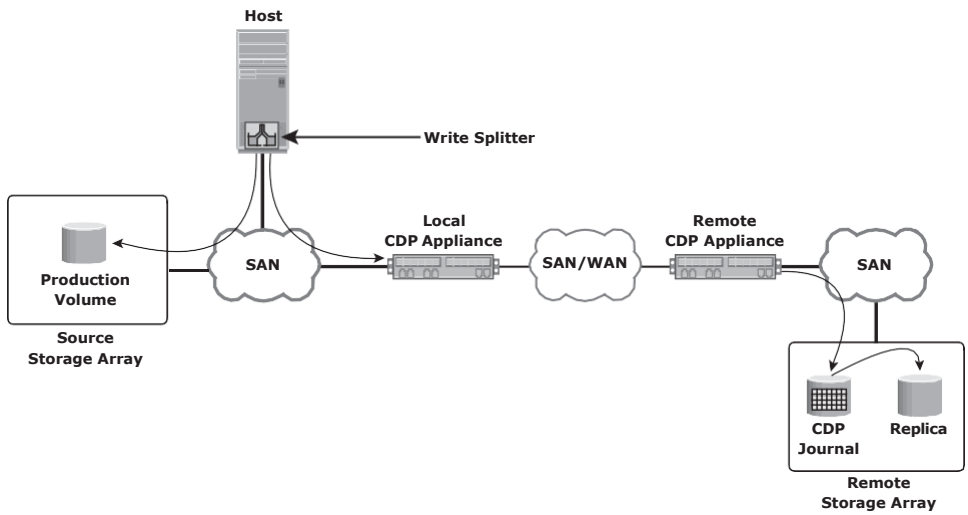


Figure 12-10: CDP remote replication

In the asynchronous mode, the local CDP appliance instantly acknowledges a write as soon as it is received. In the synchronous replication mode, the host application waits for an acknowledgment from the CDP appliance at the remote site before initiating the next write. The synchronous replication mode impacts the application's performance under heavy write loads.

For remote replication over extended distances, optical network technologies, such as dense wavelength division multiplexing (DWDM), coarse wavelength division multiplexing (CWDM), and synchronous optical network (SONET) are deployed. For more information about these technologies, refer to Appendix E.

Three-Site Replication

In synchronous replication, the source and target sites are usually within a short distance. Therefore, if a regional disaster occurs, both the source and the target sites might become unavailable. This can lead to extended RPO and RTO because the last known good copy of data would need to come from another source, such as an offsite tape library.

A regional disaster will not affect the target site in asynchronous replication because the sites are typically several hundred or several thousand kilometers apart. If the source site fails, production can be shifted to the target site, but there is no further remote protection of data until the failure is resolved.

Three-site replication mitigates the risks identified in two-site replication. In a three-site replication, data from the source site is replicated to two remote sites. Replication can be synchronous to one of the two sites, providing a near zero-RPO solution, and it can be asynchronous or disk buffered to the other remote site, providing a finite RPO. Three-site remote replication can be implemented as a cascade/multihop or a triangle/multitarget solution.

Three-Site Replication — Cascade/Multihop

In the *cascade/multihop* three-site replication, data flows from the source to the intermediate storage array, known as a *bunker*, in the first hop, and then from a bunker to a storage array at a remote site in the second hop. Replication between the source and the remote sites can be performed in two ways: synchronous + asynchronous or synchronous + disk buffered. Replication between the source and bunker occurs synchronously, but replication between the bunker and the remote site can be achieved either as disk-buffered mode or asynchronous mode.

Synchronous + Asynchronous

This method employs a combination of synchronous and asynchronous remote replication technologies. Synchronous replication occurs between the source and the bunker. Asynchronous replication occurs between the bunker and the remote site. The remote replica in the bunker acts as the source for asynchronous replication to create a remote replica at the remote site. Figure 12-11 (a) illustrates the synchronous + asynchronous method.

RPO at the remote site is usually in the order of minutes for this implementation. In this method, a minimum of three storage devices are required (including the source). The devices containing a synchronous replica at the bunker and the asynchronous replica at the remote are the other two devices.

If a disaster occurs at the source, production operations are failed over to the bunker site with zero or near-zero data loss. But unlike the synchronous two-site situation, there is still remote protection at the third site. The RPO between the bunker and third site could be in the order of minutes.

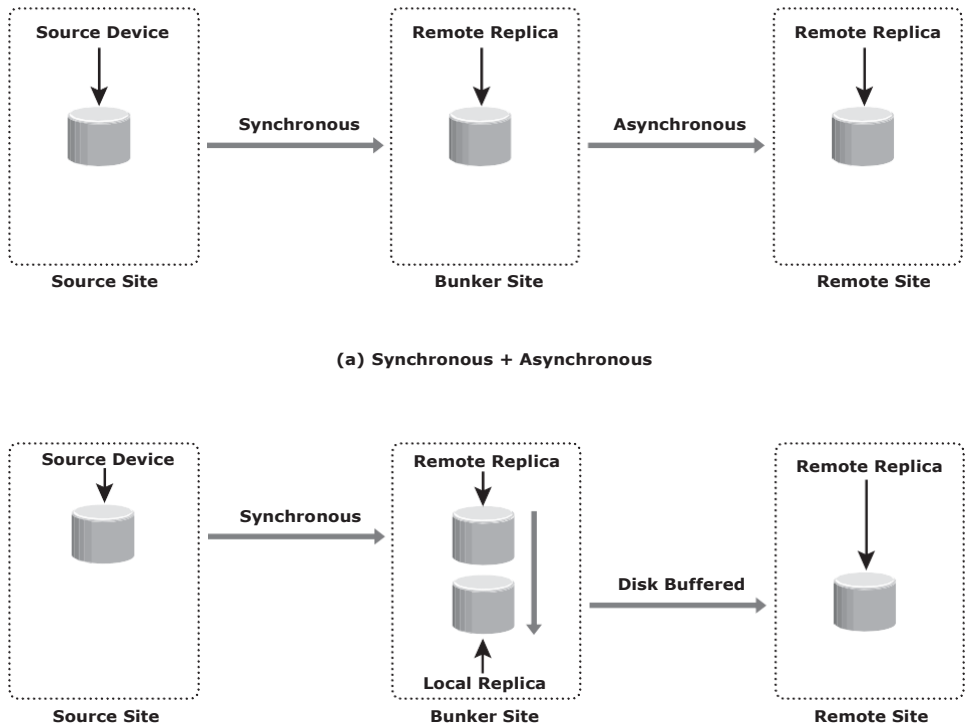


Figure 12-11: Three-site remote replication cascade/multihop

If there is a disaster at the bunker site or if there is a network link failure between the source and bunker sites, the source site continues to operate as normal but without any remote replication. This situation is similar to remote site failure in a two-site replication solution. The updates to the remote site cannot occur due to the failure in the bunker site. Therefore, the data at the remote site keeps falling behind, but the advantage here is that if the source fails during this time, operations can be resumed at the remote site. RPO at the remote site depends on the time difference between the bunker site failure and source site failure.

A *regional disaster* in three-site cascade/multihop replication is similar to a source site failure in two-site asynchronous replication. Operations are failover to the remote site with an RPO in the order of minutes. There is no remote protection until the regional disaster is resolved. Local replication technologies could be used at the remote site during this time.

If a disaster occurs at the remote site, or if the network links between the bunker and the remote site fail, the source site continues to work as normal with disaster recovery protection provided at the bunker site.

Synchronous + Disk Buffered

This method employs a combination of local and remote replication technologies. Synchronous replication occurs between the source and the bunker: a consistent PIT local replica is created at the bunker. Data is transmitted from the local replica at the bunker to the remote replica at the remote site. Optionally, a local replica can be created at the remote site after data is received from the bunker. Figure 12-11 (b) illustrates the synchronous + disk buffered method.

In this method, a minimum of four storage devices are required (including the source) to replicate one storage device. The other three devices are the synchronous remote replica at the bunker, a consistent PIT local replica at the bunker, and the replica at the remote site. RPO at the remote site is usually in the order of hours for this implementation.

The process to create the consistent PIT copy at the bunker and incrementally updating the remote replica occurs continuously in a cycle.

Three-Site Replication — Triangle/Multitarget

In *three-site triangle/multitarget replication*, data at the source storage array is concurrently replicated to two different arrays at two different sites, as shown in Figure 12-12. The source-to-bunker site (target 1) replication is synchronous with a near-zero RPO. The source-to-remote site (target 2) replication is asynchronous with an RPO in the order of minutes. The distance between the source and the remote sites could be thousands of miles. This implementation does not depend on the bunker site for updating data on the remote site because data is asynchronously copied to the remote site directly from the source. The triangle/multitarget configuration provides consistent RPO unlike cascade/multihop solutions in which the failure of the bunker site results in the remote site falling behind and the RPO increasing.

The key benefit of three-site triangle/multitarget replication is the ability to failover to either of the two remote sites in the case of source-site failure, with disaster recovery (asynchronous) protection between the bunker and remote sites. Resynchronization between the two surviving target sites is incremental. Disaster recovery protection is always available if any one-site failure occurs.

During normal operations, all three sites are available and the production workload is at the source site. At any given instant, the data at the bunker and the source is identical. The data at the remote site is behind the data at the source and the bunker. The replication network links between the bunker and remote sites will be in place but not in use. Thus, during normal operations, there is no data movement between the bunker and remote arrays. The difference in the data between the bunker and remote sites is tracked so that if a source site disaster occurs, operations can be resumed at the bunker or the remote sites with incremental resynchronization between these two sites.

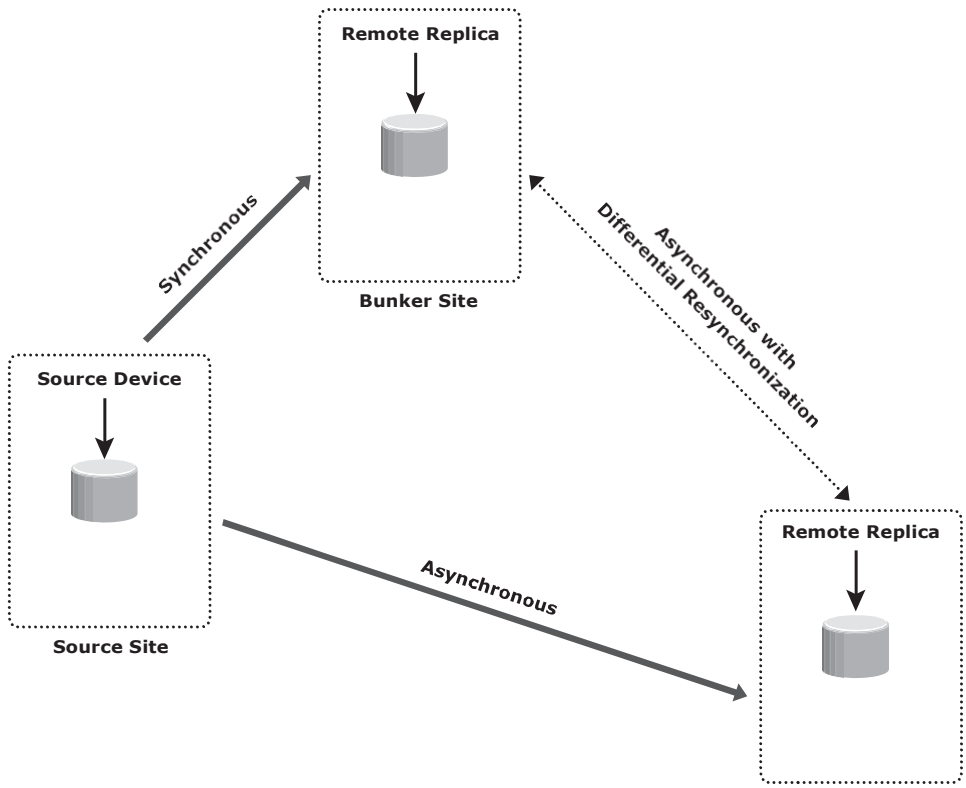


Figure 12-12: Three-site replication triangle/multitarget

A *regional disaster* in three-site triangle/multitarget replication is similar to a source site failure in two-site asynchronous replication. If failure occurs, operations failover to the remote site with an RPO within minutes. There is no remote protection until the regional disaster is resolved. Local replication technologies could be used at the remote site during this time.

A failure of the bunker or the remote site is not actually considered a disaster because the operation can continue uninterrupted at the source site while remote disaster recovery protection is still available. A network link failure to either the source-to-bunker or the source-to-remote site does not impact production at the source site while remote disaster recovery protection is still available with the site that can be reached.

Data Migration Solutions

A data migration and mobility solution is a specialized replication technique that enables creating remote point-in-time copies. These copies can be used for data mobility, migration, content distribution, and disaster recovery. This solution

moves data between heterogeneous storage arrays. Data is moved from one array to the other over the SAN or WAN. This technology is application- and server-operating-system independent because the replication operations are performed by one of the storage arrays.

Data mobility refers to moving data between heterogeneous storage arrays for cost, performance, or any other reason. It helps implement a tiered storage strategy. *Data migration* refers to moving data from one storage array to other heterogeneous storage arrays for technology refresh, consolidation, or any other reason. The array performing the replication operations is called the *control array*. Data can be moved from/to devices in the control array to/from a remote array. The devices in the control array that are part of the replication session are called *control devices*. For every control device, there is a counterpart, a *remote device*, on the *remote array*. The terms control or remote do not indicate the direction of data flow; they indicate only the array that is performing the replication operation. The direction of data movement is determined by the replication operation.

The front-end ports of the control array must be zoned to the front-end ports of the remote array. LUN masking should be performed on the remote array to allow access to the remote devices to the front-end port of the control array. In effect, the front-end ports of the control array act as an HBA, initiating data transfer to/from the remote array.

Data migration solutions perform push and pull operations for data movement. These terms are defined from the perspective of the control array. In the *push operation*, data is moved from the control array to the remote array. The control device, therefore, acts like the source, while the remote device is the target.

In the *pull operation*, data is moved from the remote array to the control array. The remote device is the source, and the control device is the target.

When a push or pull operation is initiated, the control array creates a protection bitmap to track the replication process. Each bit in the protection bitmap represents a data chunk on the control device. The chunk size varies with technology implementations. When the replication operation is initiated, all the bits are set to one, indicating that all the contents of the source device need to be copied to the target device. As the replication process copies data, the bits are changed to zero, indicating that a particular chunk has been copied. At the end of the replication process, all the bits become zero.

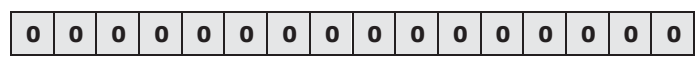
During the push and pull operations, host access to the remote device is not allowed because the control array has no control over the remote array and cannot track any change on the remote device. Data integrity cannot be guaranteed if changes are made to the remote device during the push and pull operations. The push and pull operations can be either hot or cold. These terms apply to the control devices only. In a *cold operation* the control device is inaccessible to the host during replication. Cold operations guarantee data consistency because

both the control and the remote devices are offline. In a *hot operation* the control device is online for host operations. During hot push and pull operations, changes can be made to the control device because the control array can keep track of all changes and thus ensure data integrity.

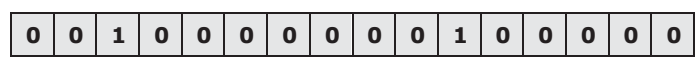
When the hot push operation is initiated, applications may be up-and-running on the control devices. I/O to the control devices is held while the protection bitmap is created. This ensures a consistent PIT image of the data. The protection bitmap is referred prior to any write to the control devices. If the bit is zero, the write is allowed. If the bit is one, the replication process holds the incoming write, copies the corresponding chunk to the remote device, and then allows the write to complete.

In the hot pull operation, the hosts can access control devices after starting the pull operation. The protection bitmap is referenced for every read or write operation. If the bit is zero, a read or write occurs. If the bit is one, the read or write is held, and the replication process copies the required chunk from the remote device. When the chunk is copied to the control device, the read or write is allowed to complete. The control devices are available for production soon after the pull operation is initiated and the protection bitmap is created. The control array can keep track of changes made to the control devices,

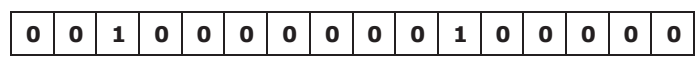
so incremental push operation is possible. A second bitmap, called a *resynchronization bitmap*, is created. All the bits in the resynchronization bitmap are set to zero when a push is initiated, as shown in Figure 12-13 (a). As changes are made to the control device, the bits are flipped from zero to one, indicating that changes have occurred, as shown in Figure 12-13 (b). When resynchronization is required, the push is reinitiated and the resynchronization bitmap becomes the new protection bitmap, as shown in Figure 12-13 (c), and only the modified chunks are transmitted to the remote devices. An incremental pull operation is not possible because tracking changes is not performed at the remote device.



(a) Resynchronization Bitmap When Push Is Initiated



(b) Resynchronization Bitmap When Data Chunks Are Updated



(c) Resynchronization Bitmap Becomes Protection Bitmap

Figure 12-13: Bitmap status during push operation

Remote Replication and Migration in a Virtualized Environment

In a virtualized environment, all VM data and VM configuration files residing on the storage array at the primary site are replicated to the storage array at the remote site. This process remains transparent to the VMs. The LUNs are replicated between the two sites using the storage array replication technology. This replication process can be either synchronous (limited distance, near zero RPO) or asynchronous (extended distance, nonzero RPO).

Virtual machine migration is another technique used to ensure business continuity in case of hypervisor failure or scheduled maintenance. VM migration is the process to move VMs from one hypervisor to another without powering off the virtual machines. VM migration also helps in load balancing when multiple virtual machines running on the same hypervisor contend for resources. Two commonly used techniques for VM migration are hypervisor-to-hypervisor and array-to-array migration.

In hypervisor-to-hypervisor VM migration, the entire active state of a VM is moved from one hypervisor to another. Figure 12-14 shows hypervisor-to-hypervisor VM migration. This method involves copying the contents of virtual machine memory from the source hypervisor to the target and then transferring the control of the VM's disk files to the target hypervisor. Because the virtual disks of the VMs are not migrated, this technique requires both source and target hypervisor access to the same storage.

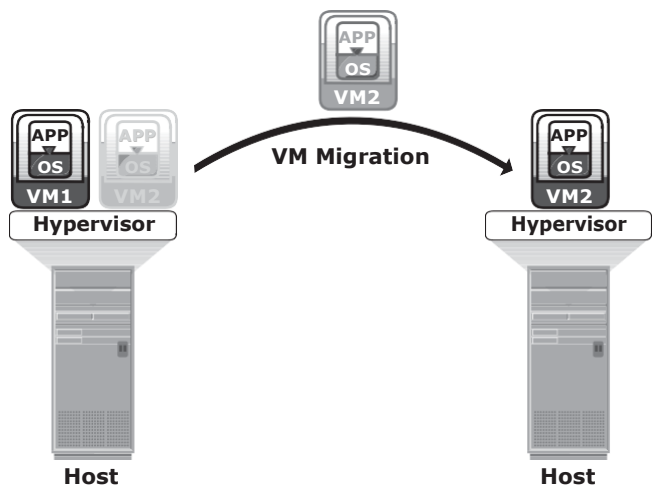


Figure 12-14: Hypervisor-to-hypervisor VM migration

In array-to-array VM migration, virtual disks are moved from the source array to the remote array. This approach enables the administrator to move

VMs across dissimilar storage arrays. Figure 12-15 shows array-to-array VM migration. Array-to-array migration starts by copying the metadata about the VM from the source array to the target. The metadata essentially consists of configuration, swap, and log files. After the metadata is copied, the VM disk file is replicated to the new location. During replication, there might be a chance that the source is updated; therefore, it is necessary to track the changes on the source to maintain data integrity. After the replication is complete, the blocks that have changed since the replication started are replicated to the new location. Array-to-array VM migration improves performance and balances the storage capacity by redistributing virtual disks to different storage devices.

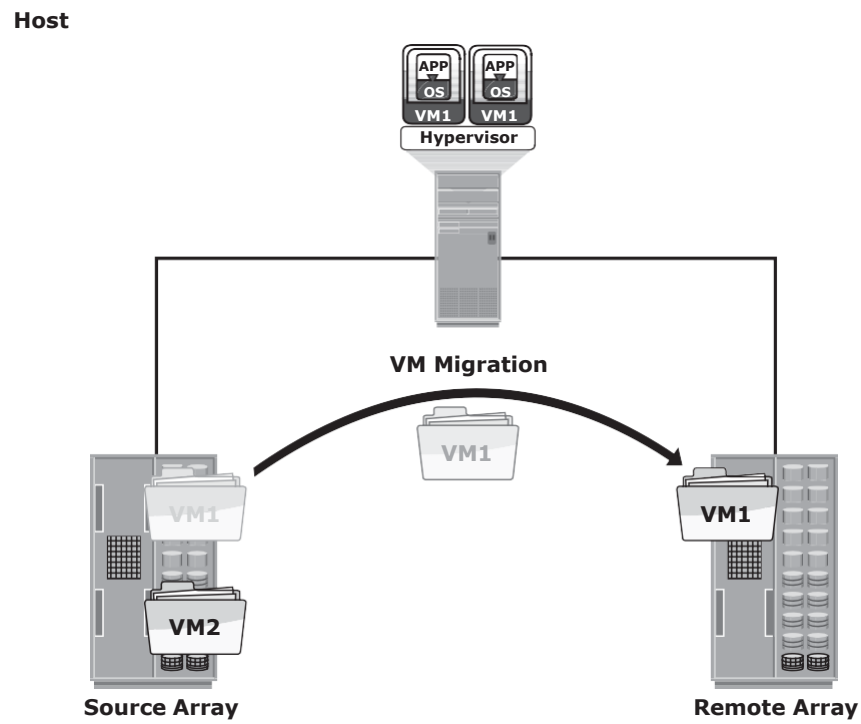


Figure 12-15: Array-to-array VM migration

Concepts in Practice: EMC SRDF, EMC MirrorView, and EMC RecoverPoint

This section discusses the EMC products for remote replication. EMC Symmetrix Remote Data Facility (SRDF) and EMC MirrorView are the storage array-based remote application software supported by EMC Symmetrix and VNX, respectively. EMC RecoverPoint is a network-based replication solution. For the latest information, visit `www.emc.com`.

EMC SRDF

SRDF offers a family of technology solutions to implement storage array-based remote replication. The SRDF family of software includes the following:

- **SRDF/Synchronous (SRDF/S):** A remote replication solution that creates a synchronous replica at one or more Symmetrix targets located within campus, metropolitan, or regional distances. SRDF/S provides a no-data-loss solution (near zero RPO) if a local disaster occurs.
- **SRDF/Asynchronous (SRDF/A):** A remote replication solution that enables the source to asynchronously replicate data. It incorporates delta set technology, which enables write ordering by employing a buffering mechanism. SRDF/A provides minimal data loss if a regional disaster occurs.
- **SRDF/DM:** A data migration solution that enables data migration from the source to the target volume over extended distances.
- **SRDF/Automated Replication (SRDF/AR):** A remote replication solution that uses both SRDF and TimeFinder/Mirror to implement disk-buffered replication technology. It is offered as SRDF/AR Single-hop for two-site replication and SRDF/AR Multihop for three-site cascade replication. SRDF/AR provides a long distance solution with RPO in the order of hours.
- **SRDF/Star:** Three-site multitarget remote replication solution that consists of primary (production), secondary (bunker), and tertiary (remote) sites. The replication between the primary and secondary sites is synchronous, whereas the replication between the primary and tertiary sites is asynchronous. If a primary site outage occurs, EMC's SRDF/Star solution enables organizations to quickly move operations and reestablish remote replication between the remaining two sites.

EMC MirrorView

The MirrorView software enables EMC VNX storage array-based remote replication. It replicates the contents of a primary volume to a secondary volume that resides on a different VNX storage system. The MirrorView family consists of MirrorView/Synchronous (MirrorView/S) and MirrorView/Asynchronous (MirrorView/A) solutions.

EMC RecoverPoint

EMC RecoverPoint Continuous Remote Replication (CRR) is a comprehensive data protection solution that provides bidirectional synchronous and asynchronous replication. In normal operations, RecoverPoint CRR enables users to

recover data remotely to any point in time. RecoverPoint dynamically switches between synchronous and asynchronous replication based on the policy for performance and latency.

Module – 4

Cloud Computing Characteristics and benefits

Cloud Enabling Technologies

Grid computing, utility computing, virtualization, and service-oriented architecture are enabling technologies of cloud computing.

- *Grid computing* is a form of distributed computing that enables the resources of numerous heterogeneous computers in a network to work together on a single task at the same time. Grid computing enables parallel computing and is best for large workloads.
- *Utility computing* is a service-provisioning model in which a service provider makes computing resources available to customers, as required, and charges them based on usage. This is analogous to other utility services, such as electricity, where charges are based on the consumption.
- *Virtualization* is a technique that abstracts the physical characteristics of IT resources from resource users. It enables the resources to be viewed and managed as a pool and lets users create virtual resources from the pool. Virtualization provides better flexibility for provisioning of IT resources compared to provisioning in a non-virtualized environment. It helps optimize resource utilization and delivering resources more efficiently.
- *Service Oriented Architecture* (SOA) provides a set of services that can communicate with each other. These services work together to perform some activity or simply pass data among services.

Characteristics of Cloud Computing

A computing infrastructure used for cloud services must have certain capabilities or characteristics. According to NIST, the cloud infrastructure should have five essential characteristics:

- **On-demand self-service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed, automatically without requiring human interaction with each service provider. A cloud service provider publishes a service catalogue, which contains information about all cloud services available to consumers. The service catalogue includes information about service attributes, prices, and request processes. Consumers view the service catalogue via a web-based user

interface and use it to request for a service. Consumers can either leverage the "ready-to-use" services or change a few service parameters to customize the services.

- **Broad network access:** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (for example, mobile phones, tablets, laptops, and workstations).
- **Resource pooling:** The provider's computing resources are pooled to serve multiple consumers using a multitenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (for example, country, state, or data center). Examples of resources include storage, processing, memory, and network bandwidth.
- **Rapid elasticity:** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

  Consumers can leverage rapid elasticity of the cloud when they have a fluctuation in their IT resource requirements. For example, an organization might require double the number of web and application servers for a specific duration to accomplish a specific task. For the remaining period, they might want to release idle server resources to cut down the expenses. The cloud enables consumers to grow and shrink the demand for resources dynamically.
- **Measured service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (for example, storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Benefits of Cloud Computing

Cloud computing offers the following key benefits:

- **Reduced IT cost:** Cloud services can be purchased based on pay-per-use or subscription pricing. This reduces or eliminates the consumer's IT capital expenditure (CAPEX).
- **Business agility:** Cloud computing provides the capability to allocate and scale computing capacity quickly. Cloud computing can reduce the time required to provision and deploy new applications and services from months to minutes. This enables businesses to respond more quickly to market changes and reduce time-to-market.
- **Flexible scaling:** Cloud computing enables consumers to scale up, scale down, scale out, or scale in the demand for computing resources easily. Consumers can unilaterally and automatically scale computing resources without any interaction with cloud service providers. The flexible service provisioning capability of cloud computing often provides a sense of unlimited scalability to the cloud service consumers.
- **High availability:** Cloud computing has the capability to ensure resource availability at varying levels depending on the consumer's policy and priority. Redundant infrastructure components (servers, network paths, and storage equipment, along with clustered software) enable fault tolerance for cloud deployments. These techniques can encompass multiple data centers located in different geographic regions, which prevents data unavailability due to regional failures.

Cloud Service Models

According to NIST, cloud service offerings are classified primarily into three models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

Infrastructure-as-a-Service

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems and deployed applications; and possibly limited control of select networking components (for example, host firewalls).

IaaS is the base layer of the cloud services stack (see Figure 13-1 [a]). It serves as the foundation for both the SaaS and PaaS layers.

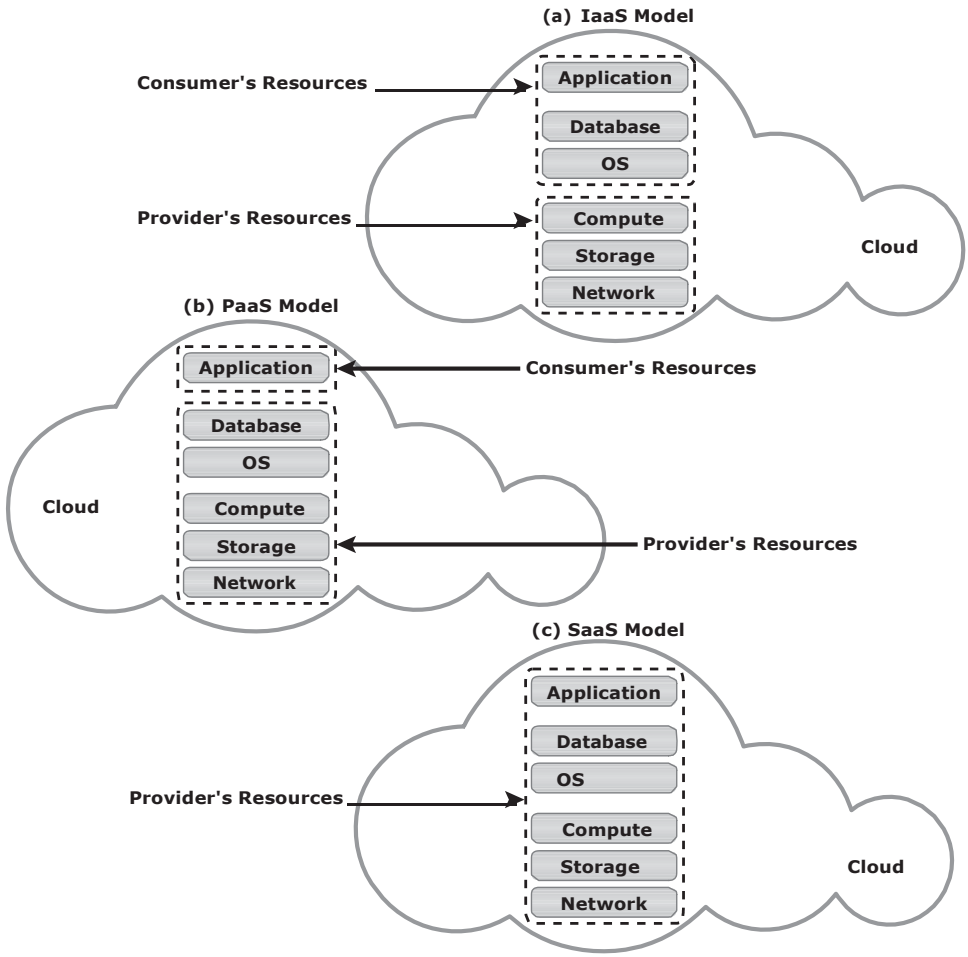


Figure 13-1: IaaS, PaaS, and SaaS models

Amazon Elastic Compute Cloud (Amazon EC2) is an example of IaaS that provides scalable compute capacity, on-demand, in the cloud. It enables consumers to leverage Amazon's massive computing infrastructure with no up-front capital investment.

Platform-as-a-Service

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not

manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. (See Figure 13-1 [b]).

PaaS is also used as an application development environment, offered as a service by the cloud service provider. The consumer may use these platforms to code their applications and then deploy the applications on the cloud. Because the workload to the deployed applications varies, the scalability of computing resources is usually guaranteed by the computing platform, transparently. Google App Engine and Microsoft Windows Azure Platform are examples of PaaS.

Software-as-a-Service

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (for example, web-based e-mail), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. (See Figure 13-1[c]).

In a SaaS model, applications, such as customer relationship management (CRM), e-mail, and instant messaging (IM), are offered as a service by the cloud service providers. The cloud service providers exclusively manage the required computing infrastructure and software to support these services. The consumers may be allowed to change a few application configuration settings to customize the applications.

EMC Mozy is an example of SaaS. Consumers can leverage the Mozy console to perform automatic, secured, online backup and recovery of their data with ease. Salesforce.com is a provider of SaaS-based CRM applications, such as Sales Cloud and Service Cloud.

Cloud Deployment Models

According to NIST, cloud computing is classified into four deployment models — public, private, community, and hybrid — which provide the basis for how cloud infrastructures are constructed and consumed.

Public Cloud

In a *public cloud* model, the cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Consumers use the cloud services offered by the providers via the Internet and pay metered usage charges or subscription fees. An advantage of the public cloud is its low capital cost with enormous scalability. However, for consumers, these benefits come with certain risks: no control over the resources in the cloud, the security of confidential data, network performance, and interoperability issues. Popular public cloud service providers are Amazon, Google, and Salesforce.com. Figure 13-2 shows a public cloud that provides cloud services to organizations and individuals.

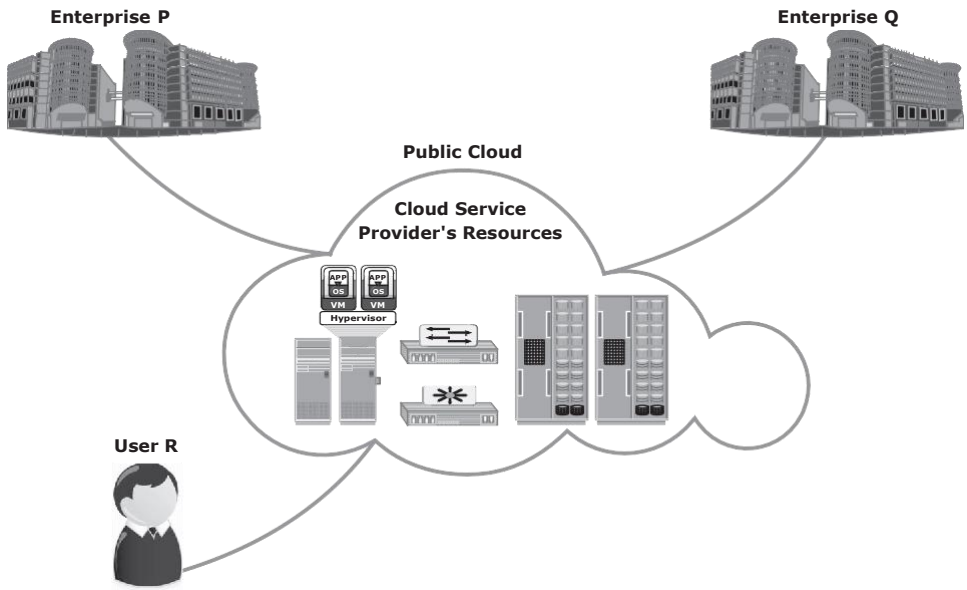
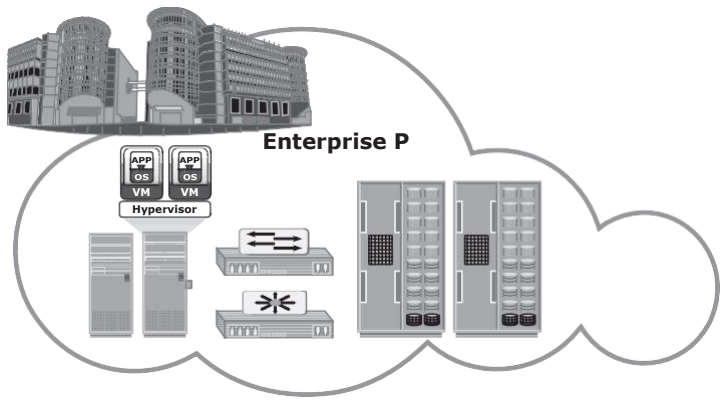


Figure 13-2: Public cloud

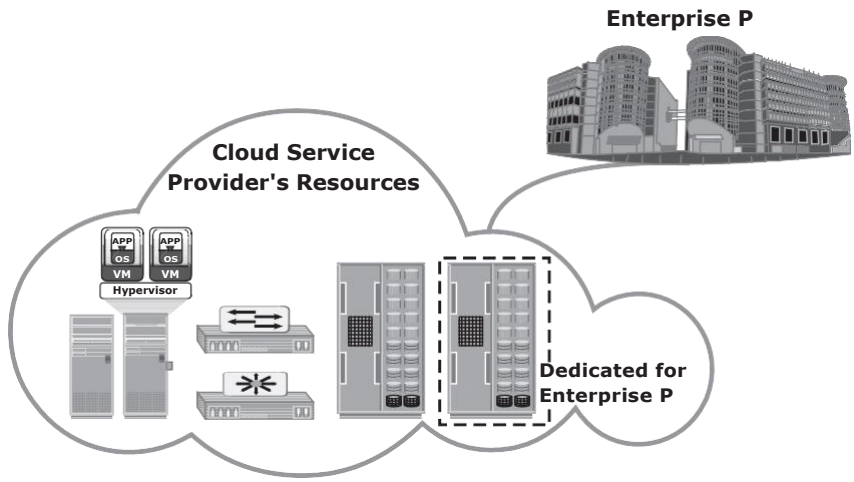
Private Cloud

In a *private cloud* model, the cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (for example, business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises. Following are two variations to the private cloud model:

- **On-premise private cloud:** The on-premise private cloud, also known as internal cloud, is hosted by an organization within its own data centers (see Figure 13-3 [a]). This model enables organizations to standardize their cloud service management processes and security, although this model has limitations in terms of size and resource scalability. Organizations would also need to incur the capital and operational costs for the physical resources. This is best suited for organizations that require complete control over their applications, infrastructure configurations, and security mechanisms.



(b) On-Premise Private Cloud



(c) Externally Hosted Private Cloud

Figure 13-3: On-premise and externally hosted private clouds

- **Externally hosted private cloud:** This type of private cloud is hosted external to an organization (see Figure 13-3 [b]) and is managed by a third-party organization. The third-party organization facilitates an exclusive cloud environment for a specific organization with full guarantee of privacy and confidentiality.

Community Cloud

In a *community cloud* model, the cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared

concerns (for example, mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises. (See Figure 13-4).

In a community cloud, the costs spread over to fewer consumers than a public cloud. Hence, this option is more expensive but might offer a higher level of privacy, security, and compliance. The community cloud also offers organizations access to a vast pool of resources compared to the private cloud. An example in which a community cloud could be useful is government agencies. If various agencies within the government operate under similar guidelines, they could all share the same infrastructure and lower their individual agency's investment.

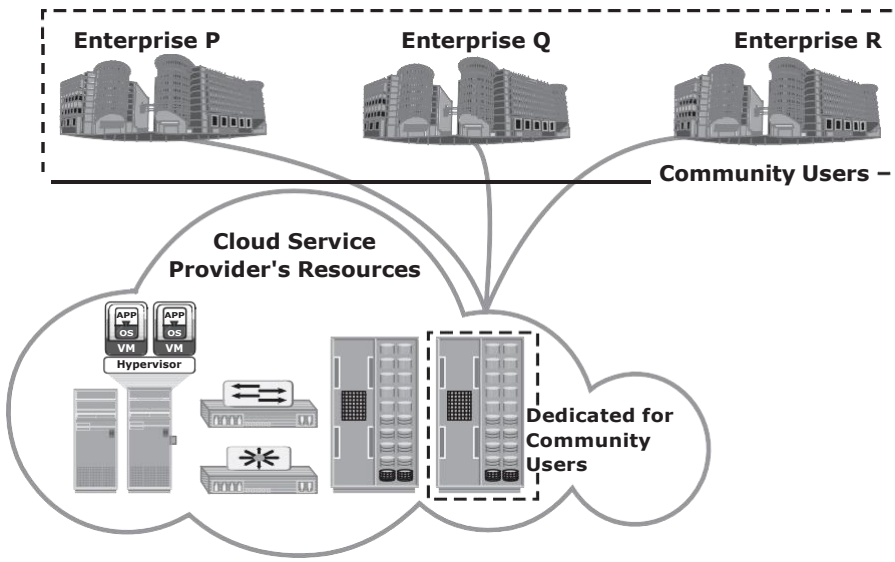


Figure 13-4: Community cloud

Hybrid Cloud

In a *hybrid cloud* model, the cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (for example, cloud bursting for load balancing between clouds).

The hybrid model allows an organization to deploy less critical applications and data to the public cloud, leveraging the scalability and cost-effectiveness of the public cloud. The organization's mission-critical applications and data remain on the private cloud that provides greater security. Figure 13-5 shows an example of a hybrid cloud.

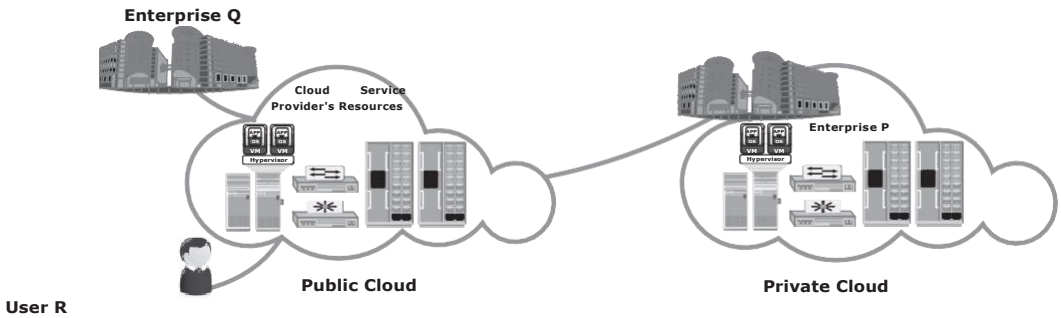


Figure 13-5: Hybrid cloud

Cloud Computing Infrastructure

A cloud computing infrastructure is the collection of hardware and software that enables the five essential characteristics of cloud computing. Cloud computing infrastructure usually consists of the following layers:

- Physical infrastructure
- Virtual infrastructure
- Applications and platform software
- Cloud management and service creation tools

The resources of these layers are aggregated and coordinated to provide cloud services to the consumers (see Figure 13-6).

Physical Infrastructure

The physical infrastructure consists of physical computing resources, which include physical servers, storage systems, and networks. Physical servers are connected to each other, to the storage systems, and to the clients via networks, such as IP, FC SAN, IP SAN, or FCoE networks.

Cloud service providers may use physical computing resources from one or more data centers to provide services. If the computing resources are distributed across multiple data centers, connectivity must be established among them. The connectivity enables the data centers in different locations to work as a single large data center. This enables migration of business applications and data across data centers and provisioning cloud services using the resources from multiple data centers.

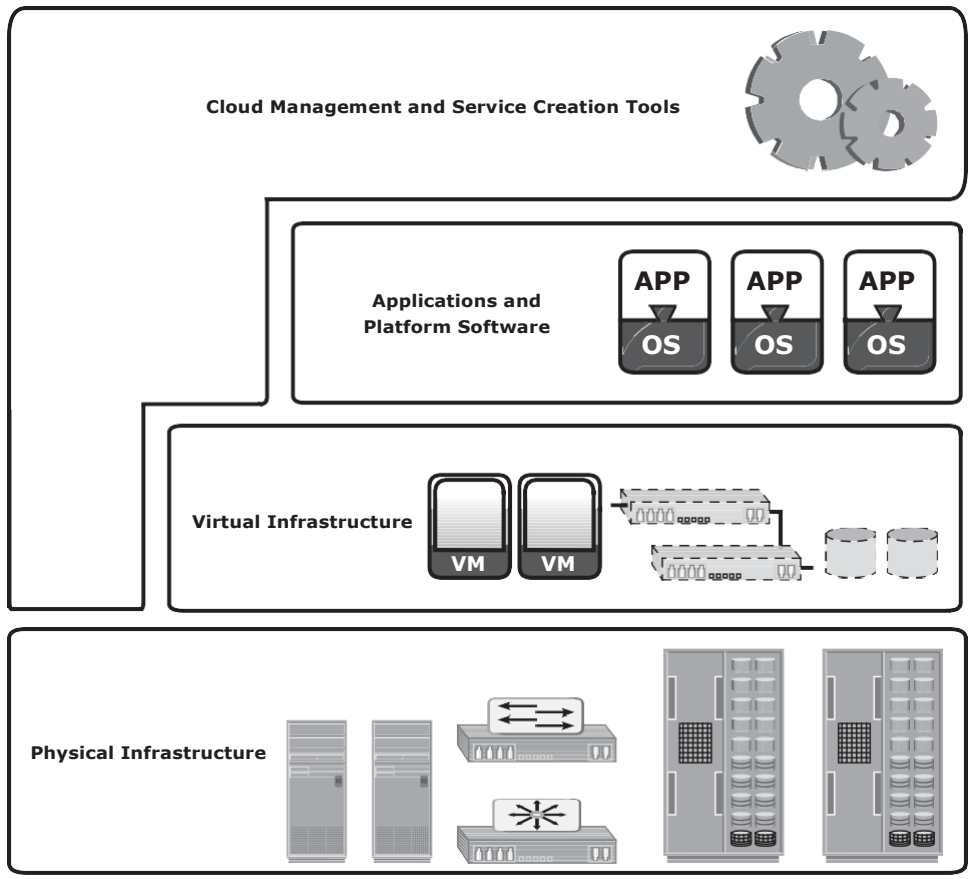


Figure 13-6: Cloud infrastructure layers

Virtual Infrastructure

Cloud service providers employ virtualization technologies to build a virtual infrastructure layer on the top of the physical infrastructure. Virtualization enables fulfilling some of the cloud characteristics, such as resource pooling and rapid elasticity. It also helps reduce the cost of providing the cloud services. Some cloud service providers may not have completely virtualized their physical infrastructure yet, but they are adopting virtualization for better efficiency and optimization.

Virtualization abstracts physical computing resources and provides a consolidated view of the resource capacity. The consolidated resources are managed as a single entity called a *resource pool*. For example, a resource pool might group CPUs of physical servers within a cluster. The capacity of the resource pool is

the sum of the power of all CPUs (for example, 10,000 megahertz) available in the cluster. In addition to the CPU pool, the virtual infrastructure includes other types of resource pools, such as memory pool, network pool, and storage pool. Apart from resource pools, the virtual infrastructure also includes *identity pools*, such as VLAN ID pools and VSAN ID pools. The number of each type of pool and the pool capacity depend on the cloud service provider's requirement to create different cloud services.

Virtual infrastructure also includes virtual computing resources, such as virtual machines, virtual storage volumes, and virtual networks. These resources obtain capacities, such as CPU power, memory, network bandwidth, and storage space from the resource pools. The capacity is allocated to the virtual computing resources easily and flexibly based on the service requirement. Virtual networks are created using network identifiers, such as VLAN IDs and VSAN IDs from the respective identity pools. Virtual computing resources are used for creating cloud infrastructure services.

Applications and Platform Software

This layer includes a suite of business applications and platform software, such as the OS and database. Platform software provides the environment on which business applications run. Applications and platform software are hosted on virtual machines to create SaaS and PaaS. For SaaS, both the application and platform software are provided by cloud service providers. In the case of PaaS, only the platform software is provided by cloud service providers; consumers export their applications to the cloud.

Cloud Management and Service Creation Tools

The cloud management and service creation tools layer includes three types of software:

- Physical and virtual infrastructure management software
- Unified management software
- User-access management software

This classification is based on the different functions performed by the software. This software interacts with each other to automate provisioning of cloud services. The physical and virtual infrastructure management software is offered by the vendors of various infrastructure resources and third-party organizations. For example, a storage array has its own management software. Similarly, network and physical servers are managed independently using network and compute management software respectively. This software provides interfaces to construct a virtual infrastructure from the underlying physical infrastructure.

Unified management software interacts with all standalone physical and virtual infrastructure management software. It collects information on the existing physical and virtual infrastructure configurations, connectivity, and utilization. Unified management software compiles this information and provides a consolidated view of infrastructure resources scattered across one or more data centers. It allows an administrator to monitor performance, capacity, and availability of physical and virtual resources centrally. Unified management software also provides a single management interface to configure physical and virtual infrastructure and integrate the compute (both CPU and memory), network, and storage pools. The integration allows a group of compute pools to use the storage and network pools for storing and transferring data respectively. The unified management software passes configuration commands to respective physical and virtual infrastructure management software, which executes the instructions. This eliminates the administration of compute, storage, and network resources separately using native management software.

The key function of the unified management software is to automate the creation of cloud services. It enables administrators to define service attributes such as CPU power, memory, network bandwidth, storage capacity, name and description of applications and platform software, resource location, and backup policy. When the unified management software receives consumer requests for cloud services, it creates the service based on predefined service attributes. The user-access management software provides a web-based user interface to consumers. Consumers can use the interface to browse the service catalogue and request cloud services. The user-access management software authenticates users before forwarding their request to the unified management software. It also monitors allocation or usage of resources associated to the cloud service instances. Based on the allocation or usage of resources, it generates a chargeback report. The chargeback report is visible to consumers and provides transparency between consumers and providers.

(Continued)

Cloud Challenges

Although there is growing acceptance of cloud computing, both the cloud service consumers and providers have been facing some challenges.

Challenges for Consumers

Business-critical data requires protection and continuous monitoring of its access. If the data moves to a cloud model other than an on-premise private cloud, consumers could lose absolute control of their sensitive data. Although most of the cloud service providers offer enhanced data security, consumers might not be willing to transfer control of their business-critical data to the cloud. Cloud service providers might use multiple data centers located in different countries to provide cloud services. They might replicate or move data across these data centers to ensure high availability and load distribution. Consumers may or may not know in which country their data is stored. Some cloud service providers allow consumers to select the location for storing their data. Data privacy concerns and regulatory compliance requirements, such as the EU Data Protection Directive and the U.S. Safe Harbor program, create challenges for the consumers in adopting cloud computing.

Cloud services can be accessed from anywhere via a network. However, network latency increases when the cloud infrastructure is not close to the access point. A high network latency can either increase the application

response time or cause the application to timeout. This can be addressed by implementing stringent Service Level Agreements (SLAs) with the cloud service providers.

Another challenge is that cloud platform services may not support consumers' desired applications. For example, a service provider might not be able to support highly specialized or proprietary environments, such as compatible OSs and preferred programming languages, required to develop and run the consumer's application. Also, a mismatch between hypervisors could impact migration of virtual machines into or between clouds.

Another challenge is vendor lock-in: the difficulty for consumers to change their cloud service provider. A lack of interoperability between the APIs of different cloud service providers could also create complexity and high migration costs when moving from one service provider to another.

Challenges for Providers

Cloud service providers usually publish a service-level agreement (SLA) so that their consumers know about the availability of service, quality of service, downtime compensation, and legal and regulatory clauses. Alternatively, customer-specific SLAs may be signed between a cloud service provider and a consumer. SLAs typically mention a penalty amount if cloud service providers fail to provide the service levels. Therefore, cloud service providers must ensure that they have adequate resources to provide the required levels of services. Because the cloud resources are distributed and service demands fluctuate, it is a challenge for cloud service providers to provision physical resources for peak demand of all consumers and estimate the actual cost of providing the services.

Many software vendors do not have a cloud-ready software licensing model. Some of the software vendors offer standardized cloud licenses at a higher price compared to traditional licensing models. The cloud software licensing complexity has been causing challenges in deploying vendor software in the cloud. This is also a challenge to the consumer.

Cloud service providers usually offer proprietary APIs to access their cloud. However, consumers might want open APIs or standard APIs to become the tenant of multiple clouds. This is a challenge for cloud service providers because this requires agreement among cloud service providers.

Cloud Adoption Considerations

Organizations that decide to adopt cloud computing always face this question: "How does the cloud fit the organization's environment?" Most organizations are not ready to abandon their existing IT investments to move all their business processes to the cloud at once. Instead, they need to consider various factors

before moving their business processes to the cloud. Even individuals seeking to use cloud services need to understand some cloud adoption considerations. Following are some key considerations for cloud adoption:

- **Selection of a deployment model:** Risk versus convenience is a key consideration for deciding on a cloud adoption strategy. This consideration also forms the basis for choosing the right cloud deployment model. A public cloud is usually preferred by individuals and start-up businesses. For them, the cost reduction offered by the public cloud outweighs the security or availability risks in the cloud. Small- and medium-sized businesses (SMBs) have a moderate customer base, and any anomaly in customer data and service levels might impact their business. Therefore, they may not be willing to deploy their tier 1 applications, such as Online Transaction Processing (OLTP), in the public cloud. A hybrid cloud model fits in this case. The tier 1applications should run on the private cloud, whereas less critical applications such as backup, archive, and testing can be deployed in the public cloud. Enterprises typically have a strong customer base worldwide. They usually enforce strict security policies to safeguard critical customer data. Because they are financially capable, they might prefer building their own private clouds.
- **Application suitability:** Not all applications are good candidates for a public cloud. This may be due to the incompatibility between the cloud platform software and the consumer applications, or maybe the organization plans to move a legacy application to the cloud. Proprietary and mission-critical applications are core and essential to the business. They are usually designed, developed, and maintained in-house. These applications often provide competitive advantages. Due to high security risk, organizations are unlikely to move these applications to the public cloud. These applications are good candidate for an on-premise private cloud. Nonproprietary and nonmission critical applications are suitable for deployment in the public cloud. If an application workload is network traffic-intensive, its performance might not be optimal if deployed in the public cloud. Also if the application communicates with other data center resources or applications, it might experience performance issues.
- **Financial advantage:** A careful analysis of financial benefits provides a clear picture about the cost-savings in adopting the cloud. The analysis should compare both the Total Cost of Ownership (TCO) and the Return on Investment (ROI) in the cloud and noncloud environment and identify the potential cost benefit. While calculating TCO and ROI, organizations and individuals should consider the expenditure to deploy and maintain their own infrastructure versus cloud-adoption costs. While calculating the expenditures for owning infrastructure resources, organizations should include both the capital expenditure (CAPEX) and operation expenditure

(OPEX). The CAPEX includes the cost of servers, storage, OS, application, network equipment, real estate, and so on. The OPEX includes the cost incurred for power and cooling, personnel, maintenance, backup, and so on. These expenditures should be compared with the operation cost incurred in adopting cloud computing. The cloud adoption cost includes the cost of migrating to the cloud, cost to ensure compliance and security, and usage or subscription fees. Moving applications to the cloud reduces CAPEX, except when the cloud is built on-premise.

- **Selection of a cloud service provider:** The selection of the provider is important for a public cloud. Consumers need to find out how long and how well the provider has been delivering the services. They also need to determine how easy it is to add or terminate cloud services with the service provider. The consumer should know how easy it is to move to another provider, when required. They must assess how the provider fulfills the security, legal, and privacy requirements. They should also check whether the provider offers good customer service support.
- **Service-level agreement (SLA):** Cloud service providers typically mention quality of service (QoS) attributes such as throughput and uptime, along with cloud services. The QoS attributes are generally part of an SLA, which is the service contract between the provider and the consumers. The SLA serves as the foundation for the expected level of service between the consumer and the provider. Before adopting the cloud services, consumers should check whether the QoS attributes meet their requirements.

Concepts in Practice: Vblock

Vblock is a completely integrated cloud infrastructure offering that includes compute, storage, network, and virtualization products. These products are provided by EMC, VMware, and Cisco, who have formed a coalition to deliver Vblocks.

Vblocks enable organizations to build virtualized data centers and cloud infrastructures. Vblocks are pre-architected, preconfigured, pretested and have defined performance and availability attributes. Rather than customers buying and assembling individual cloud infrastructure components, Vblock provides a validated cloud infrastructure solution and is factory-ready for deployment and production. This saves significant cost and deployment time.

EMC Unified Infrastructure Manager (UIM) is the unified management solution for Vblocks. UIM provides a single point of management for Vblocks and manages multiple Vblocks. With UIM, cloud infrastructure services can be provisioned automatically based on provisioning best practices.

Module – 5

Securing and Managing Storage Infrastructure

Information Security Framework

The basic information security framework is built to achieve four security goals: confidentiality, integrity, and availability (CIA), along with accountability. This framework incorporates all security standards, procedures, and controls, required to mitigate threats in the storage infrastructure environment.

- **Confidentiality:** Provides the required secrecy of information and ensures that only authorized users have access to data. This requires authentication of users who need to access information.

  Data in transit (data transmitted over cables) and data at rest (data residing on a primary storage, backup media, or in the archives) can be encrypted to maintain its confidentiality. In addition to restricting unauthorized users from accessing information, confidentiality also requires implementing traffic flow protection measures as part of the security protocol. These protection measures generally include hiding source and destination addresses, frequency of data being sent, and amount of data sent.

- **Integrity:** Ensures that the information is unaltered. Ensuring integrity requires detection of and protection against unauthorized alteration or deletion of information. Ensuring integrity stipulates measures such as error detection and correction for both data and systems.

- **Availability:** This ensures that authorized users have reliable and timely access to systems, data, and applications residing on these systems. Availability requires protection against unauthorized deletion of data and denial of service (discussed in section "14.2.2 Threats"). Availability also implies that sufficient resources are available to provide a service.

- **Accountability service:** Refers to accounting for all the events and operations that take place in the data center infrastructure. The accountability service maintains a log of events that can be audited or traced later for the purpose of security.

Risk Triad

Risk triad defines risk in terms of threats, assets, and vulnerabilities. Risk arises when a threat agent (an attacker) uses an existing vulnerability to compromise the security services of an asset, for example, if a sensitive document is transmitted without any protection over an insecure channel, an attacker might get unauthorized access to the document and may violate its confidentiality and integrity. This may, in turn, result in business loss for the organization. In this scenario potential business loss is the risk, which arises because an attacker

uses the vulnerability of the unprotected communication to access the document and tamper with it.

To manage risks, organizations primarily focus on vulnerabilities because they cannot eliminate threat agents that appear in various forms and sources to its assets. Organizations can enforce countermeasures to reduce the possibility of occurrence of attacks and the severity of their impact.

Risk assessment is the first step to determine the extent of potential threats and risks in an IT infrastructure. The process assesses risk and helps to identify appropriate controls to mitigate or eliminate risks. Based on the value of assets, risk assessment helps to prioritize investment in and provisioning of security measures. To determine the probability of an adverse event occurring, threats to an IT system must be analyzed with the potential vulnerabilities and the existing security controls.

The severity of an adverse event is estimated by the impact that it may have on critical business activities. Based on this analysis, a relative value of criticality and sensitivity can be assigned to IT assets and resources. For example, a particular IT system component may be assigned a high-criticality value if an attack on this particular component can cause a complete termination of mission-critical services.

The following sections examine the three key elements of the risk triad. Assets, threats, and vulnerabilities are considered from the perspective of risk identification and control analysis.

Assets

Information is one of the most important *assets* for any organization. Other assets include hardware, software, and other infrastructure components required to access the information. To protect these assets, organizations must develop a set of parameters to ensure the availability of the resources to authorized users and trusted networks. These parameters apply to storage resources, network infrastructure, and organizational policies.

Security methods have two objectives. The first objective is to ensure that the network is easily accessible to authorized users. It should also be reliable and stable under disparate environmental conditions and volumes of usage. The second objective is to make it difficult for potential attackers to access and compromise the system.

The security methods should provide adequate protection against unauthorized access, viruses, worms, trojans, and other malicious software programs. Security measures should also include options to encrypt critical data and disable unused services to minimize the number of potential security gaps. The security method must ensure that updates to the operating system and other software are installed regularly. At the same time, it must provide adequate redundancy in the form of replication and mirroring of the production data

to prevent catastrophic data loss if there is an unexpected data compromise. For the security system to function smoothly, all users are informed about the policies governing the use of the network.

The effectiveness of a storage security methodology can be measured by two key criteria. One, the cost of implementing the system should be a fraction of the value of the protected data. Two, it should cost heavily to a potential attacker, in terms of money, effort, and time.

Threats

Threats are the potential attacks that can be carried out on an IT infrastructure. These attacks can be classified as active or passive. *Passive attacks* are attempts to gain unauthorized access into the system. They pose threats to confidentiality of information. *Active attacks* include data modification, denial of service (DoS), and repudiation attacks. They pose threats to data integrity, availability, and accountability. In a data modification attack, the unauthorized user attempts to modify information for malicious purposes. A modification attack can target the data at rest or the data in transit. These attacks pose a threat to data integrity.

Denial of service (DoS) attacks prevent legitimate users from accessing resources and services. These attacks generally do not involve access to or modification of information. Instead, they pose a threat to data availability. The intentional flooding of a network or website to prevent legitimate access to authorized users is one example of a DoS attack.

Repudiation is an attack against the accountability of information. It attempts to provide false information by either impersonating someone or denying that an event or a transaction has taken place. For example, a repudiation attack may involve performing an action and eliminating any evidence that could prove the identity of the user (attacker) who performed that action. Repudiation attacks include circumventing the logging of security events or tampering with the security log to conceal the identity of the attacker.

Vulnerability

The paths that provide access to information are often vulnerable to potential attacks. Each of the paths may contain various access points, which provide different levels of access to the storage resources. It is important to implement adequate security controls at all the access points on an access path. Implementing security controls at each access point of every access path is known as *defense in depth*.

Defense in depth recommends using multiple security measures to reduce the risk of security threats if one component of the protection is compromised. It is also known as a "layered approach to security." Because there are multiple measures for security at different levels, defense in depth gives additional time to detect and respond to an attack. This can reduce the scope or impact of a security breach.

Attack surface, *attack vector*, and *work factor* are the three factors to consider when assessing the extent to which an environment is vulnerable to security threats. *Attack surface* refers to the various entry points that an attacker can use to launch an attack. Each component of a storage network is a source of potential vulnerability. An attacker can use all the external interfaces supported by that component, such as the hardware and the management interfaces, to execute various attacks. These interfaces form the attack surface for the attacker. Even unused network services, if enabled, can become a part of the attack surface.

An *attack vector* is a step or a series of steps necessary to complete an attack. For example, an attacker might exploit a bug in the management interface to execute a snoop attack whereby the attacker can modify the configuration of the storage device to allow the traffic to be accessed from one more host. This redirected traffic can be used to snoop the data in transit.

Work factor refers to the amount of time and effort required to exploit an attack vector. For example, if attackers attempt to retrieve sensitive information, they consider the time and effort that would be required for executing an attack on a database. This may include determining privileged accounts, determining the database schema, and writing SQL queries. Instead, based on the work factor, they may consider a less effort-intensive way to exploit the storage array by attaching to it directly and reading from the raw disk blocks.

Having assessed the vulnerability of the environment, organizations can deploy specific control measures. Any control measures should involve all the three aspects of infrastructure: people, process, and technology, and the relationships among them. To secure people, the first step is to establish and assure their identity. Based on their identity, selective controls can be implemented for their access to data and resources. The effectiveness of any security measure is primarily governed by processes and policies. The processes should be based on a thorough understanding of risks in the environment and should recognize the relative sensitivity of different types of data and the needs of various stakeholders to access the data. Without an effective process, the deployment

of technology is neither cost-effective nor aligned to organizations' priorities. Finally, the technologies or controls that are deployed should ensure compliance with the processes, policies, and people for its effectiveness. These security technologies are directed at reducing vulnerability by minimizing attack surfaces and maximizing the work factors. These controls can be technical or nontechnical. Technical controls are usually implemented through computer systems, whereas nontechnical controls are implemented through administrative and physical controls. Administrative controls include security and personnel policies or standard procedures to direct the safe execution of various operations. Physical controls include setting up physical barriers, such as security guards, fences, or locks.

Based on the roles they play, controls are categorized as preventive, detective, and corrective. The preventive control attempts to prevent an attack; the detective control detects whether an attack is in progress; and after an attack is discovered, the corrective controls are implemented. *Preventive controls* avert the vulnerabilities from being exploited and prevent an attack or reduce its impact. *Corrective controls* reduce the effect of an attack, whereas *detective controls* discover attacks and trigger preventive or corrective controls. For example, an Intrusion Detection/Intrusion Prevention System (IDS/IPS) is a detective control that determines whether an attack is underway and then attempts to stop it by terminating a network connection or invoking a firewall rule to block traffic.

Storage Security Domains

Storage devices connected to a network raise the risk level and are more exposed to security threats via networks. However, with increasing use of networking in storage environments, storage devices are becoming highly exposed to security threats from a variety of sources. Specific controls must be implemented to secure a storage networking environment. This requires a closer look at storage networking security and a clear understanding of the access paths leading to storage resources. If a particular path is unauthorized and needs to be prohibited by technical controls, ensure that these controls are not compromised. If each component within the storage network is considered a potential access point, the attack surface of all these access points must be analyzed to identify the associated vulnerabilities.

To identify the threats that apply to a storage network, access paths to data storage can be categorized into three security domains: *application access*, *management access*, and *backup, replication, and archive*. Figure 14-1 depicts the three security domains of a storage system environment.

The first security domain involves application access to the stored data through the storage network. The second security domain includes management access to storage and interconnect devices and to the data residing on those devices.

This domain is primarily accessed by storage administrators who configure and manage the environment. The third domain consists of backup, replication, and archive access. Along with the access points in this domain, the backup media also needs to be secured.

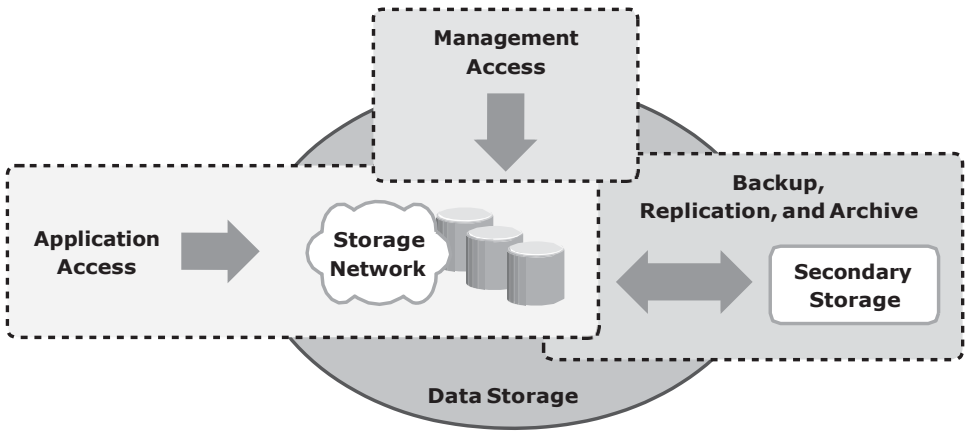


Figure 14-1: Storage security domains

To secure the storage networking environment, identify the existing threats within each of the security domains and classify the threats based on the type of security services — availability, confidentiality, integrity, and accountability. The next step is to select and implement various controls as countermeasures to the threats.

Securing the Application Access Domain

The *application access domain* may include only those applications that access the data through the file system or a database interface.

An important step to secure the application access domain is to identify the threats in the environment and appropriate controls that should be applied. Implementing physical security is also an important consideration to prevent media theft.

Figure 14-2 shows application access in a storage networking environment. Host A can access all V1 volumes; host B can access all V2 volumes. These volumes are classified according to the access level, such as confidential, restricted, and public. Some of the possible threats in this scenario could be host A spoofing the identity or elevating to the privileges of host B to gain access to host B's resources. Another threat could be that an unauthorized host gains access to the network; the attacker on this host may try to spoof the identity of another host and tamper with the data, snoop the network, or execute a DoS attack. Also any form of media theft could also compromise security. These threats can pose several serious challenges to the network security; therefore, they need to be addressed.

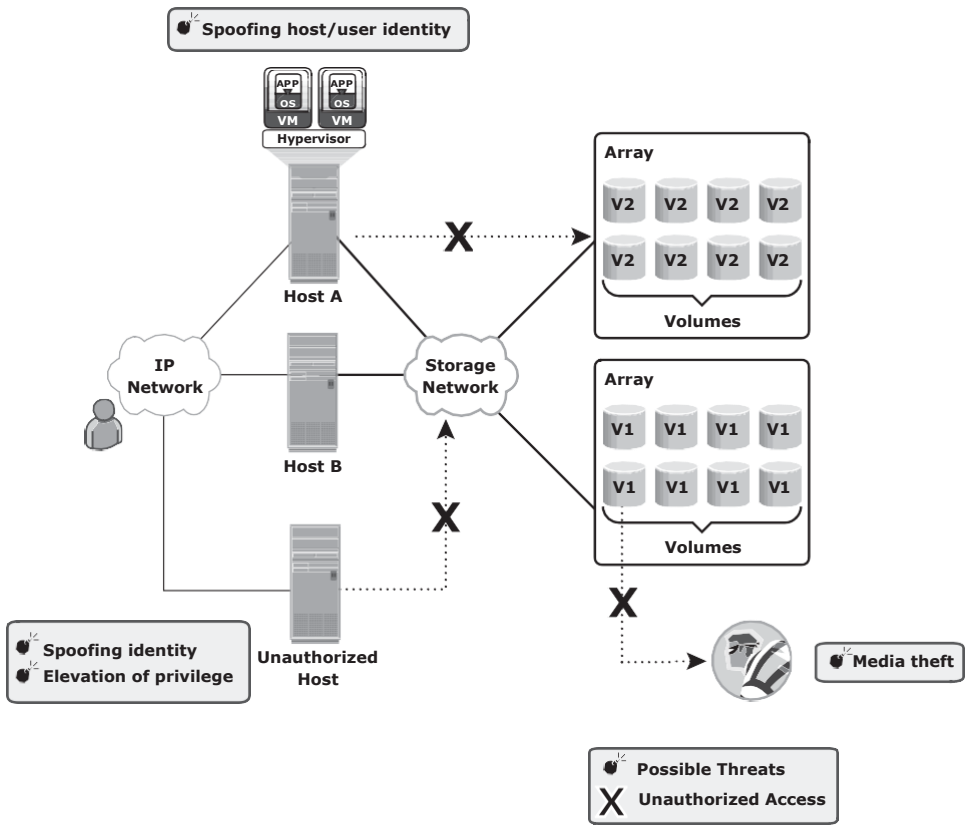


Figure 14-2: Security threats in an application access domain

Controlling User Access to Data

Access control services regulate user access to data. These services mitigate the threats of spoofing host identity and elevating host privileges. Both these threats affect data integrity and confidentiality.

Access control mechanisms used in the application access domain are user and host authentication (technical control) and authorization (administrative control). These mechanisms may lie outside the boundaries of the storage network and require various systems to interconnect with other enterprise identity management and authentication systems, for example, systems that provide strong authentication and authorization to secure user identities against spoofing. NAS devices support the creation of *access control lists* that regulate user access to specific files. The Enterprise Content Management application enforces access to data by using Information Rights Management (IRM) that specifies which users have what rights to a document. Restricting access at the host level starts with authenticating a node when it tries to connect to a network.

Different storage networking technologies, such as iSCSI, FC, and IP-based storage, use various authentication mechanisms, such as Challenge-Handshake Authentication Protocol (CHAP), Fibre Channel Security Protocol (FC-SP), and IPSec, respectively, to authenticate host access.

After a host has been authenticated, the next step is to specify security controls for the storage resources, such as ports, volumes, or storage pools, that the host is authorized to access. *Zoning* is a control mechanism on the switches that segments the network into specific paths to be used for data traffic; *LUN masking* determines which hosts can access which storage devices. Some devices support mapping of a host's WWN to a particular FC port and from there to a particular LUN. This binding of the WWN to a physical port is the most secure. Finally, it is important to ensure that administrative controls, such as defined security policies and standards, are implemented. Regular auditing is required to ensure proper functioning of administrative controls. This is enabled by logging significant events on all participating devices. Event logs should also be protected from unauthorized access because they may fail to achieve their goals if the logged content is exposed to unauthorized modifications by an attacker.

Protecting the Storage Infrastructure

Securing the storage infrastructure from unauthorized access involves protecting all the elements of the infrastructure. Security controls for protecting the storage infrastructure address the threats of unauthorized tampering of data in transit that leads to a loss of data integrity, denial of service that compromises availability, and network snooping that may result in loss of confidentiality.

The security controls for protecting the network fall into two general categories: *network infrastructure integrity* and *storage network encryption*. Controls for ensuring the infrastructure integrity include a fabric switch function that ensures fabric integrity. This is achieved by preventing a host from being added to the SAN fabric without proper authorization. Storage network encryption methods include the use of IPSec for protecting IP-based storage networks, and FC-SP for protecting FC networks.

In secure storage environments, root or administrator privileges for a specific device are not granted to every user. Instead, *role-based access control* (RBAC) is deployed to assign necessary privileges to users, enabling them to perform their roles. A role may represent a job function, for example, an administrator. Privileges are associated with the roles and users acquire these privileges based upon their roles.

It is also advisable to consider administrative controls, such as "separation of duties," when defining data center procedures. Clear separation of duties ensures that no single individual can both specify an action and carry it out. For example, the person who authorizes the creation of administrative accounts

should not be the person who uses those accounts. Securing management access is covered in detail in the next section.

Management networks for storage systems should be logically separate from other enterprise networks. This segmentation is critical to facilitate ease of management and increase security by allowing access only to the components existing within the same segment. For example, IP network segmentation is enforced with the deployment of filters at Layer 3 by using routers and firewalls, and at Layer 2 by using VLANs and port-level security on Ethernet switches.

Finally, physical access to the device console and the cabling of FC switches must be controlled to ensure protection of the storage infrastructure. All other established security measures fail if a device is physically accessed by an unauthorized user; this access may render the device unreliable.

Data Encryption

The most important aspect of securing data is protecting data held inside the storage arrays. Threats at this level include tampering with data, which violates data integrity, and media theft, which compromises data availability and confidentiality. To protect against these threats, encrypt the data held on the storage media or encrypt the data prior to being transferred to the disk. It is also critical to decide upon a method for ensuring that data deleted at the end of its life cycle has been completely erased from the disks and cannot be reconstructed for malicious purposes.

Data should be encrypted as close to its origin as possible. If it is not possible to perform encryption on the host device, an encryption appliance can be used for encrypting data at the point of entry into the storage network. Encryption devices can be implemented on the fabric that encrypts data between the host and the storage media. These mechanisms can protect both the data at rest on the destination device and data in transit.

On NAS devices, adding antivirus checks and file extension controls can further enhance data integrity. In the case of CAS, use of MD5 or SHA-256 cryptographic algorithms guarantees data integrity by detecting any change in content bit patterns. In addition, the data erasure service ensures that the data has been completely overwritten by bit sequence before the disk is discarded. An organization's data classification policy determines whether the disk should actually be scrubbed prior to discarding it and the level of erasure needed based on regulatory requirements.

Securing the Management Access Domain

Management access, whether monitoring, provisioning, or managing storage resources, is associated with every device within the storage network. Most management software supports some form of CLI, system management console,

or a web-based interface. Implementing appropriate controls for securing storage management applications is important because the damage that can be caused by using these applications can be far more extensive.

Figure 14-3 depicts a storage networking environment in which production hosts are connected to a SAN fabric and are accessing production storage array A, which is connected to remote storage array B for replication purposes. Further, this configuration has a storage management platform on Host A. A possible threat in this environment is an unauthorized host spoofing the user or host identity to manage the storage arrays or network. For example, an unauthorized host may gain management access to remote array B.

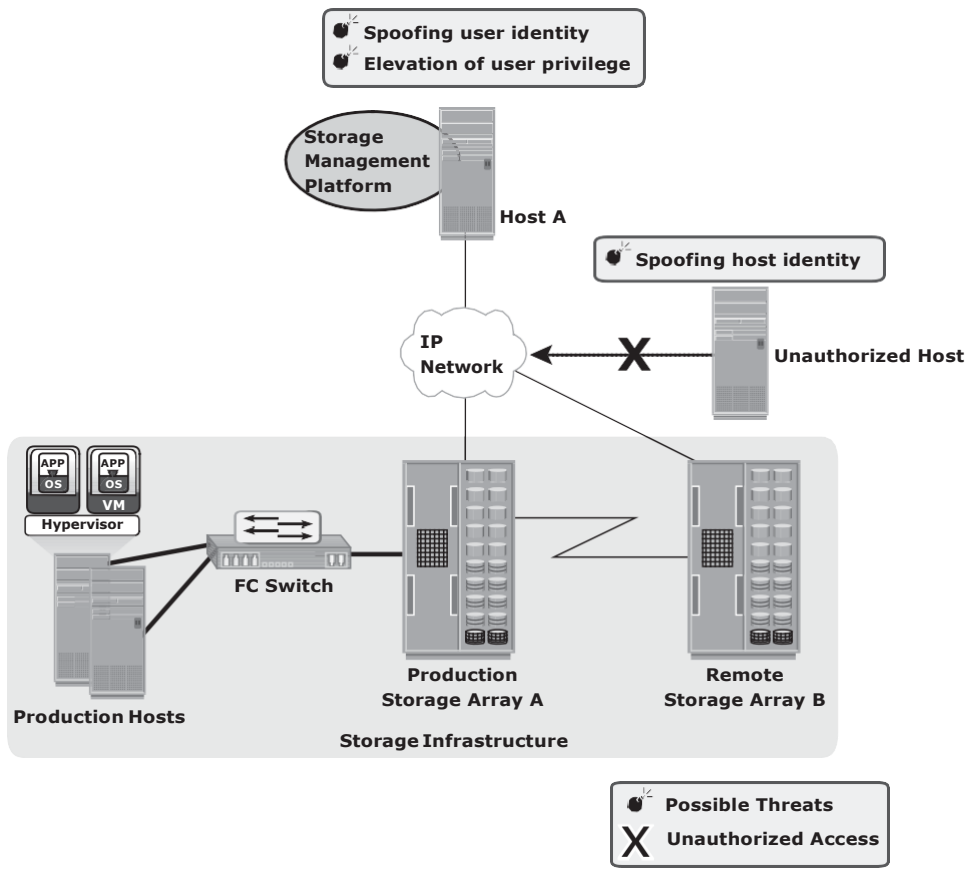


Figure 14-3: Security threats in a management access domain

Providing management access through an external network increases the potential for an unauthorized host or switch to connect to that network. In such circumstances, implementing appropriate security measures prevents certain types of remote communication from occurring. Using secure communication

channels, such as Secure Shell (SSH) or Secure Sockets Layer (SSL)/Transport Layer Security (TLS), provides effective protection against these threats. Event log monitoring helps to identify unauthorized access and unauthorized changes to the infrastructure. Event logs should be placed outside the shared storage systems where they can be reviewed if the storage is compromised.

The storage management platform must be validated for available security controls and ensures that these controls are adequate to secure the overall storage environment. The administrator's identity and role should be secured against any spoofing attempts so that an attacker cannot manipulate the entire storage array and cause intolerable data loss by reformatting storage media or making data resources unavailable.

Controlling Administrative Access

Controlling administrative access to storage aims to safeguard against the threats of an attacker spoofing an administrator's identity or elevating privileges to gain administrative access. Both of these threats affect the integrity of data and devices. To protect against these threats, administrative access regulation and various auditing techniques are used to enforce accountability of users and processes. Access control should be enforced for each storage component. In some storage environments, it may be necessary to integrate storage devices with third-party authentication directories, such as Lightweight Directory Access Protocol (LDAP) or Active Directory.

Security best practices stipulate that no single user should have ultimate control over all aspects of the system. If an administrative user is a necessity, the number of activities requiring administrative privileges should be minimized. Instead, it is better to assign various administrative functions by using RBAC. Auditing logged events is a critical control measure to track the activities of an administrator. However, access to administrative log files and their content must be protected. Deploying a reliable Network Time Protocol on each system that can be synchronized to a common time is another important requirement to ensure that activities across systems can be consistently tracked. In addition, having a Security Information Management (SIM) solution supports effective analysis of the event log files.

Protecting the Management Infrastructure

Mechanisms to protect the management network infrastructure include encrypting management traffic, enforcing management access controls, and applying IP network security best practices. These best practices include the use of IP routers and Ethernet switches to restrict the traffic to certain devices. Restricting network activity and access to a limited set of hosts minimizes the threat of an unauthorized device attaching to the network and gaining access to the

management interfaces. Access controls need to be enforced at the storage-array level to specify which host has management access to which array. Some storage devices and switches can restrict management access to particular hosts and limit the commands that can be issued from each host.

A separate private management network is highly recommended for management traffic. If possible, management traffic should not be mixed with either production data traffic or other LAN traffic used in the enterprise. Unused network services must be disabled on every device within the storage network. This decreases the attack surface for that device by minimizing the number of interfaces through which the device can be accessed.

To summarize, security enforcement must focus on the management communication between devices, confidentiality and integrity of management data, and availability of management networks and devices.

Securing Backup, Replication, and Archive

Backup, replication, and archive is the third domain that needs to be secured against an attack. As explained in Chapter 10, a backup involves copying the data from a storage array to backup media, such as tapes or disks. Securing backup is complex and is based on the backup software that accesses the storage arrays. It also depends on the configuration of the storage environments at the primary and secondary sites, especially with remote backup solutions performed directly on a remote tape device or using array-based remote replication.

Organizations must ensure that the disaster recovery (DR) site maintains the same level of security for the backed up data. Protecting the backup, replication, and archive infrastructure requires addressing several threats, including spoofing the legitimate identity of a DR site, tampering with data, network snooping, DoS attacks, and media theft. Such threats represent potential violations of integrity, confidentiality, and availability. Figure 14-4 illustrates a generic remote backup design whereby data on a storage array is replicated over a DR network to a secondary storage at the DR site. In a remote backup solution where the storage components are separated by a network, the threats at the transmission layer need to be countered. Otherwise, an attacker can spoof the identity of the backup server and request the host to send its data. The unauthorized host claiming to be the backup server may lead to a remote backup being performed to an unauthorized and unknown site. In addition, attackers can use the DR network connection to tamper with data, snoop the network, and create a DoS attack against the storage devices.

The physical threat of a backup tape being lost, stolen, or misplaced, especially if the tapes contain highly confidential information, is another type of threat. Backup-to-tape applications are vulnerable to severe security implications if they do not encrypt data while backing it up.

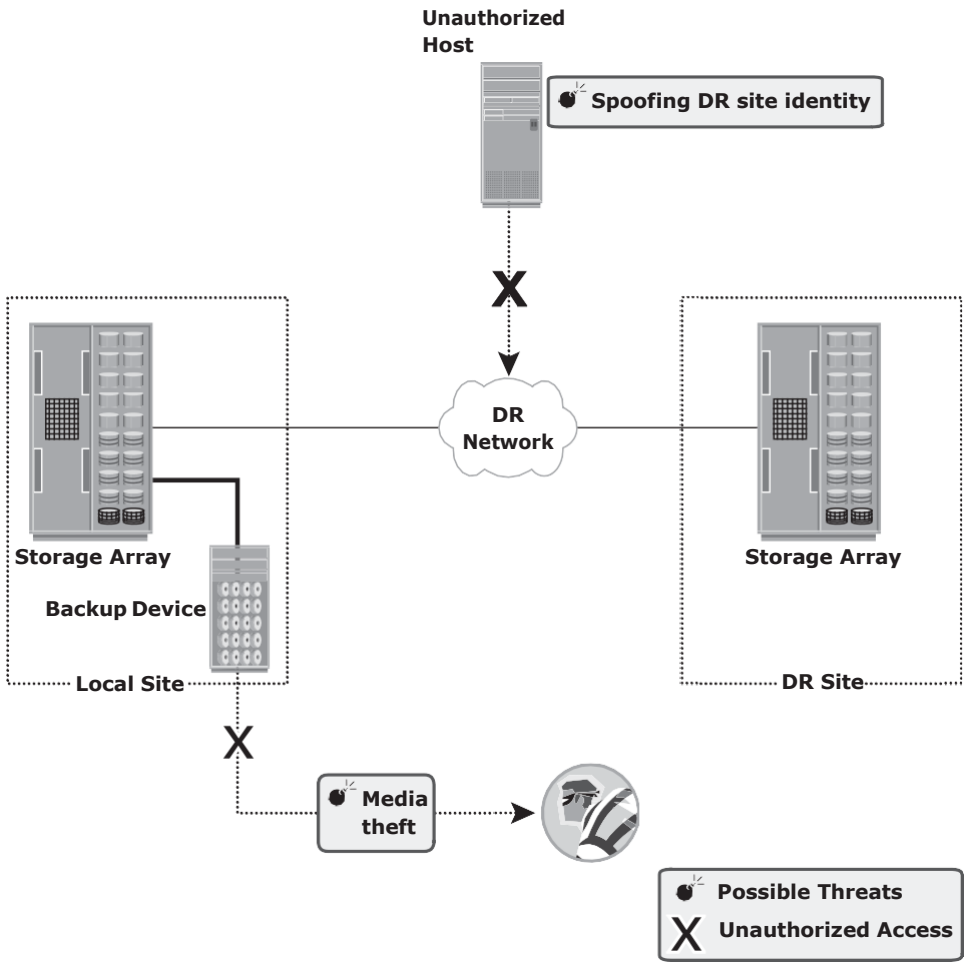


Figure 14-4: Security threats in a backup, replication, and archive environment

Security Implementations in Storage Networking

The following discussion details some of the basic security implementations in FC SAN, NAS, and IP-SAN environments.

FC SAN

Traditional FC SANs enjoy an inherent security advantage over IP-based networks. An FC SAN is configured as an isolated private environment with fewer nodes than an IP network. Consequently, FC SANs impose fewer security

threats. However, this scenario has changed with converged networks and storage consolidation, driving rapid growth and necessitating designs for large, complex SANs that span multiple sites across the enterprise. Today, no single comprehensive security solution is available for FC SANs. Many FC SAN security mechanisms have evolved from their counterpart in IP networking, thereby bringing in matured security solutions.

Fibre Channel Security Protocol (FC-SP) standards (T11 standards), published in 2006, align security mechanisms and algorithms between IP and FC interconnects. These standards describe protocols to implement security measures in a FC fabric, among fabric elements and N_Ports within the fabric. They also include guidelines for authenticating FC entities, setting up session keys, negotiating the parameters required to ensure frame-by-frame integrity and confidentiality, and establishing and distributing policies across an FC fabric.

FC SAN Security Architecture

Storage networking environments are a potential target for unauthorized access, theft, and misuse because of the vastness and complexity of these environments. Therefore, security strategies are based on the *defense in depth* concept, which recommends multiple integrated layers of security. This ensures that the failure of one security control will not compromise the assets under protection. Figure 14-5 illustrates various levels (zones) of a storage networking environment that must be secured and the security measures that can be deployed.

FC SANs not only suffer from certain risks and vulnerabilities that are unique, but also share common security problems associated with physical security and remote administrative access. In addition to implementing SAN-specific security measures, organizations must simultaneously leverage other security implementations in the enterprise. Table 14-1 provides a comprehensive list of protection strategies that must be implemented in various security zones. Some of the security mechanisms listed in Table 14-1 are not specific to SAN but are commonly used data center techniques. For example, two-factor authentication is implemented widely; in a simple implementation it requires the use of a username/password and an additional security component such as a smart card for authentication.

Basic SAN Security Mechanisms

LUN masking and zoning, switch-wide and fabric-wide access control, RBAC, and logical partitioning of a fabric (Virtual SAN) are the most commonly used SAN security methods.

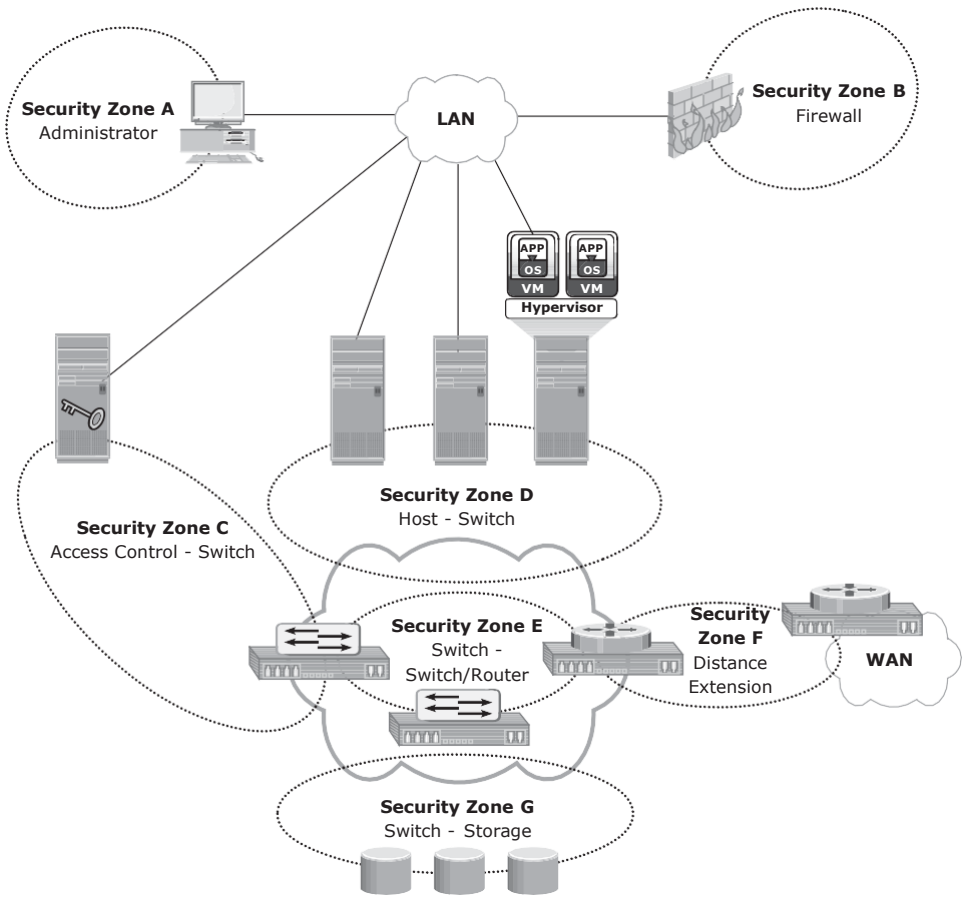


Figure 14-5: FC SAN security architecture

Table 14-1: Security Zones and Protection Strategies

SECURITY ZONES	PROTECTION STRATEGIES
Zone A (Authentication at the Management Console)	(a) Restrict management LAN access to authorized users (lock down MAC addresses); (b) implement VPN tunneling for secure remote access to the management LAN; and (c) use two-factor authentication for network access.
Zone B (Firewall)	Block inappropriate traffic by (a) filtering out addresses that should not be allowed on your LAN; and (b) screening for allowable protocols, block ports that are not in use.
Zone C (Access Control-Switch)	Authenticate users/administrators of FC switches using Remote Authentication Dial In User Service (RADIUS), DH-CHAP (Diffie-Hellman Challenge Handshake Authentication Protocol), and so on.

SECURITY ZONES	PROTECTION STRATEGIES
Zone D (Host to switch)	Restrict Fabric access to legitimate hosts by (a) implementing ACLs: Known HBAs can connect on specific switch ports only; and (b) implementing a secure zoning method, such as port zoning (also known as hard zoning).
Zone E (Switch to Switch/Switch to Router)	Protect traffic on fabric by (a) using E_Port authentication; (b) encrypting the traffic in transit; and (c) implementing FC switch controls and port controls.
Zone F (Distance Extension)	Implement encryption for in-flight data (a) FC-SP for long-distance FC extension; and (b) IPSec for SAN extension via FCIP.
Zone G (Switch to Storage)	Protect the storage arrays on your SAN via (a) WWPN-based LUN masking; and (b) S_ID locking: masking based on source FC address.

LUN Masking and Zoning

LUN masking and zoning are the basic SAN security mechanisms used to protect against unauthorized access to storage. LUN masking and zoning are detailed in Chapter 4 and Chapter 5, respectively. The standard implementations of LUN masking on storage arrays mask the LUNs presented to a front-end storage port based on the WWPNs of the source HBAs. A stronger variant of LUN masking may sometimes be offered whereby masking can be done on the basis of source FC addresses. It offers a mechanism to lock down the FC address of a given node port to its WWN. *WWPN zoning* is the preferred choice in security-conscious environments.

Securing Switch Ports

Apart from zoning and LUN masking, additional security mechanisms, such as port binding, port lockdown, port lockout, and persistent port disable, can be implemented on switch ports. *Port binding* limits the number of devices that can attach to a particular switch port and allows only the corresponding switch port to connect to a node for fabric access. Port binding mitigates but does not eliminate WWPN spoofing. *Port lockdown* and *port lockout* restrict a switch port's type of initialization. Typical variants of port lockout ensure that the switch port cannot function as an E_Port and cannot be used to create an ISL, such as a rogue switch. Some variants ensure that the port role is restricted to only FL_Port, F_Port, E_Port, or a combination of these. *Persistent port disable* prevents a switch port from being enabled even after a switch reboot.

Switch-Wide and Fabric-Wide Access Control

As organizations grow their SANs locally or over longer distances, there is a greater need to effectively manage SAN security. Network security can be configured on the FC switch by using *access control lists* (ACLs) and on the fabric by using fabric binding.

ACLs incorporate the device connection control and switch connection control policies. The device connection control policy specifies which HBAs and storage ports can be a part of the fabric, preventing unauthorized devices from accessing it. Similarly, the switch connection control policy specifies which switches are allowed to be part of the fabric, preventing unauthorized switches from joining it. *Fabric binding* prevents an unauthorized switch from joining any existing switch in the fabric. It ensures that authorized membership data exists on every switch and any attempt to connect any switch in the fabric by using an ISL causes the fabric to segment.

Role-based access control provides additional security to a SAN by preventing unauthorized activity on the fabric for management operations. It enables the security administrator to assign roles to users that explicitly specify privileges or access rights after logging into the fabric. For example, the *zone admin* role can modify the zones on the fabric, whereas a basic user may view only fabric-related information, such as port types and logged-in nodes.

Logical Partitioning of a Fabric: Virtual SAN

VSANs enable the creation of multiple logical SANs over a common physical SAN. They provide the capability to build larger consolidated fabrics and still maintain the required security and isolation between them. Figure 14-6 depicts logical partitioning in a VSAN.

The SAN administrator can create distinct VSANs by populating each of them with switch ports. In the example, the switch ports are distributed over two VSANs: 10 and 20 — for the Engineering and HR divisions, respectively. Although they share physical switching gear with other divisions, they can be managed individually as standalone fabrics. Zoning should be done for each VSAN to secure the entire physical SAN. Each managed VSAN can have only one active zone set at a time. VSANs minimize the impact of fabricwide disruptive events because management and control traffic on the SAN — which may include RSCNs, zone set activation events, and more — does not traverse VSAN boundaries. Therefore, VSANs are a cost-effective alternative for building isolated physical fabrics. They contribute to information availability and security by isolating fabric events and providing authorization control within a single fabric.

NAS

NAS is open to multiple exploits, including viruses, worms, unauthorized access, snooping, and data tampering. Various security mechanisms are implemented in NAS to secure data and the storage networking infrastructure.

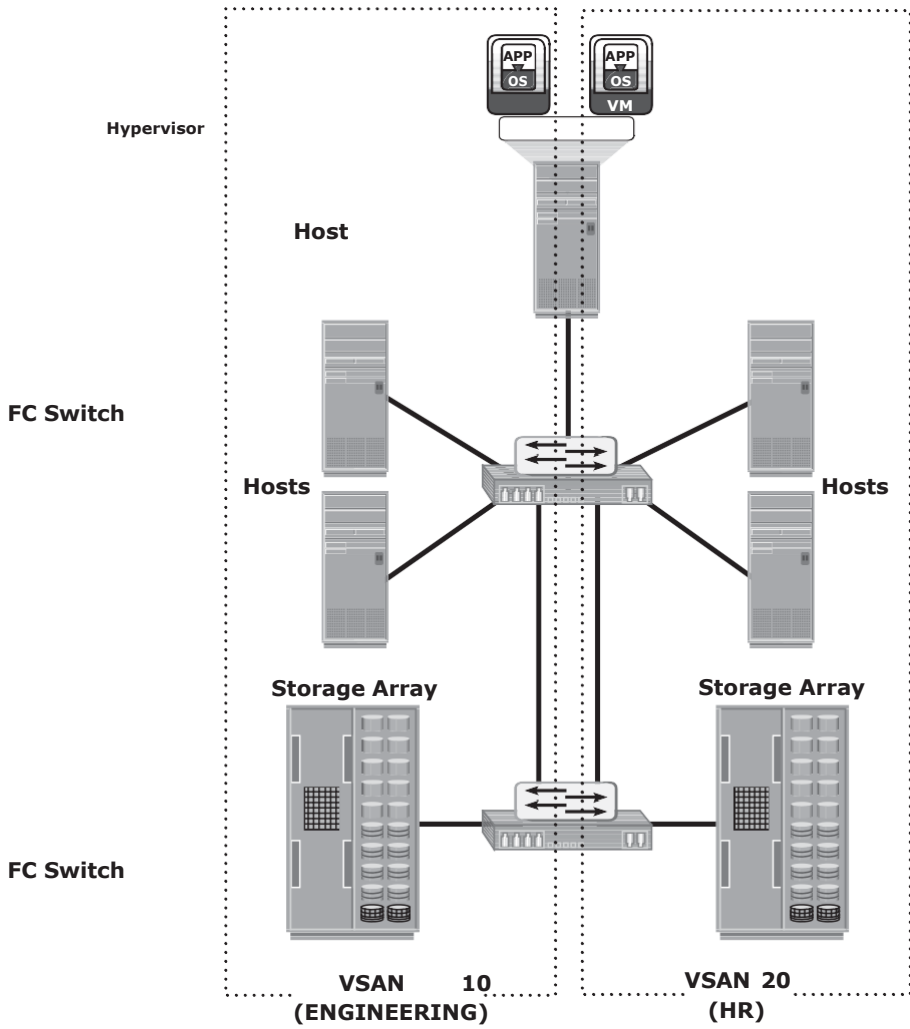


Figure 14-6: Securing SAN with VSAN

Permissions and ACLs form the first level of protection to NAS resources by restricting accessibility and sharing. These permissions are deployed over and above the default behaviors and attributes associated with files and folders. In addition, various other authentication and authorization mechanisms, such as Kerberos and directory services, are implemented to verify the identity of network users and define their privileges. Similarly, firewalls protect the storage infrastructure from unauthorized access and malicious attacks.

NAS File Sharing: Windows ACLs

Windows supports two types of ACLs: *discretionary access control lists* (DACLs) and *system access control lists* (SACLs). The DACL, commonly referred to as the

ACL, that determines access control. The SACL determines what accesses need to be audited if auditing is enabled.

In addition to these ACLs, Windows also supports the concept of object ownership. The owner of an object has hard-coded rights to that object, and these rights do not need to be explicitly granted in the SACL. The owner, SACL, and DACL are all statically held as attributes of each object. Windows also offers the functionality to inherit permissions, which allows the child objects existing within a parent object to automatically inherit the ACLs of the parent object.

ACLs are also applied to directory objects known as security identifiers (SIDs). These are automatically generated by a Windows server or domain when a user or group is created, and they are abstracted from the user. In this way, though a user may identify his login ID as "User1," it is simply a textual representation of the true SID, which is used by the underlying operating system. Internal processes in Windows refer to an account's SID rather than the account's username or group name while granting access to an object. ACLs are set by using the standard Windows Explorer GUI but can also be configured with CLI commands or other third-party tools.

NAS File Sharing: UNIX Permissions

For the UNIX operating system, a *user* is an abstraction that denotes a logical entity for assignment of ownership and operation privileges for the system. A user can be either a person or a system operation. A UNIX system is only aware of the privileges of the user to perform specific operations on the system and identifies each user by a user ID (UID) and a username, regardless of whether it is a person, a system operation, or a device.

In UNIX, users can be organized into one or more groups. The concept of group serves the purpose to assign sets of privileges for a given resource and sharing them among many users that need them. For example, a group of people working on one project may need the same permissions for a set of files.

UNIX permissions specify the operations that can be performed by any ownership relation with respect to a file. In simpler terms, these permissions specify what the owner can do, what the owner group can do, and what everyone else can do with the file. For any given ownership relation, three bits are used to specify access permissions. The first bit denotes read (r) access, the second bit denotes write (w) access, and the third bit denotes execute (x) access. Because UNIX defines three ownership relations (Owner, Group, and All), a triplet (defining the access permission) is required for each ownership relationship, resulting in nine bits. Each bit can be either set or clear. When displayed, a set bit is marked by its corresponding operation letter (r, w, or x), a clear bit is denoted by a dash (-), and all are put in a row, such as rwxr-xr-x. In this example, the owner can do anything with the file, but group owners and the rest of the world can read or execute only. When displayed, a character denoting the mode of the file may

precede this nine-bit pattern. For example, if the file is a directory, it is denoted as "d"; and if it is a link, it is denoted as "l."

NAS File Sharing: Authentication and Authorization

In a file-sharing environment, NAS devices use standard file-sharing protocols, NFS and CIFS. Therefore, authentication and authorization are implemented and supported on NAS devices in the same way as in a UNIX or Windows file-sharing environment.

Authentication requires verifying the identity of a network user and therefore involves a login credential lookup on a Network Information System (NIS) server in a UNIX environment. Similarly, a Windows client is authenticated by a Windows domain controller that houses the Active Directory. The Active Directory uses LDAP to access information about network objects in the directory and Kerberos for network security. NAS devices use the same authentication techniques to validate network user credentials. Figure 14-7 depicts the authentication process in a NAS environment.

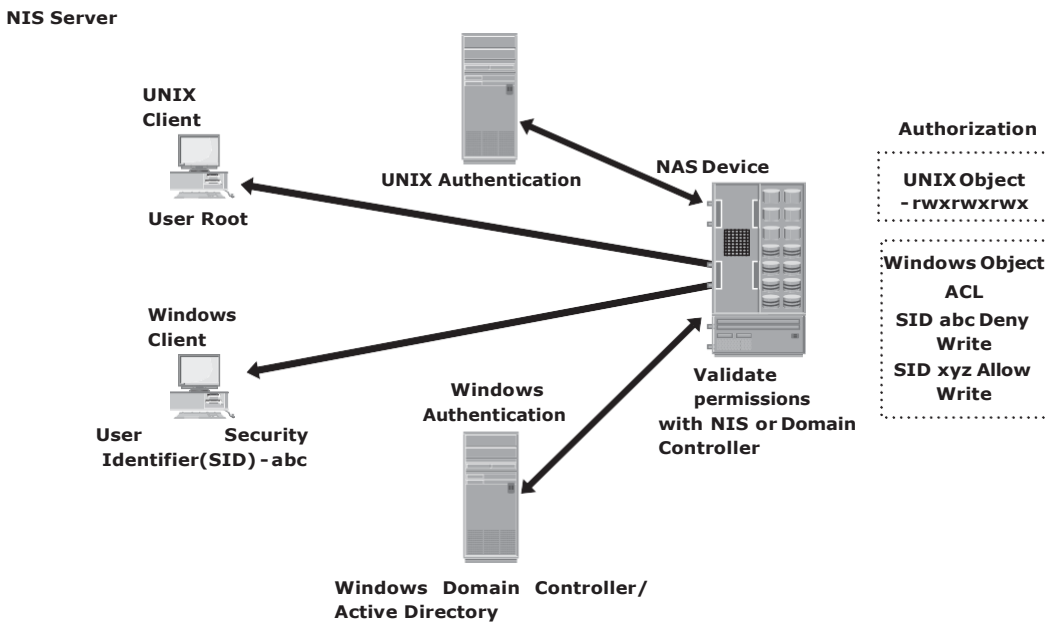


Figure 14-7: Securing user access in a NAS environment

Authorization defines user privileges in a network. The authorization techniques for UNIX users and Windows users are quite different. UNIX files use mode bits to define access rights granted to owners, groups, and other users, whereas Windows uses an ACL to allow or deny specific rights to a particular user for a particular file.

Although NAS devices support both of these methodologies for UNIX and Windows users, complexities arise when UNIX and Windows users access and share the same data. If the NAS device supports multiple protocols, the integrity of both permission methodologies must be maintained. NAS device vendors provide a method of mapping UNIX permissions to Windows and vice versa, so a multiprotocol environment can be supported. However, consider these complexities of multiprotocol support when designing a NAS solution. At the same time, validate the domain controller and NIS server connectivity and bandwidth. If multiprotocol access is required, specific vendor access policy implementations need to be considered.

Kerberos

Kerberos is a network authentication protocol, which is designed to provide strong authentication for client/server applications by using secret-key cryptography. It uses cryptography so that a client and server can prove their identity to each other across an insecure network connection. After the client and server have proven their identities, they can choose to encrypt all their communications to ensure privacy and data integrity.

In Kerberos, authentications occur between clients and servers. The client gets a ticket for a service and the server decrypts this ticket by using its secret key. Any entity, user, or host that gets a service ticket for a Kerberos service is called a *Kerberos client*. The term *Kerberos server* generally refers to the Key Distribution Center (KDC). The KDC implements the Authentication Service (AS) and the Ticket Granting Service (TGS). The KDC has a copy of every password associated with every principal, so it is absolutely vital that the KDC remain secure. In Kerberos, users and servers for which a secret key is stored in the KDC database are known as *principals*.

In a NAS environment, Kerberos is primarily used when authenticating against a Microsoft Active Directory domain, although it can be used to execute security functions in UNIX environments. The Kerberos authentication process shown in Figure 14-8 includes the following steps:

1. The user logs on to the workstation in the Active Directory domain (or forest) using an ID and a password. The client computer sends a request to the AS running on the KDC for a Kerberos ticket. The KDC verifies the user's login information from Active Directory. (This step is not explicitly shown in Figure 14-8.)
2. The KDC responds with an encrypted Ticket Granting Ticket (TGT) and an encrypted session key. TGT has a limited validity period. TGT can be decrypted only by the KDC, and the client can decrypt only the session key.
3. When the client requests a service from a server, it sends a request, consisting of the previously generated TGT, encrypted with the session key and the resource information to the KDC.

4. The KDC checks the permissions in Active Directory and ensures that the user is authorized to use that service.
5. The KDC returns a service ticket to the client. This service ticket contains fields addressed to the client and to the server hosting the service.
6. The client then sends the service ticket to the server that houses the required resources.
7. The server, in this case the NAS device, decrypts the server portion of the ticket and stores the information in a keytab file. As long as the client's Kerberos ticket is valid, this authorization process does not need to be repeated. The server automatically allows the client to access the appropriate resources.
8. A client-server session is now established. The server returns a session ID to the client, which tracks the client activity, such as file locking, as long as the session is active.

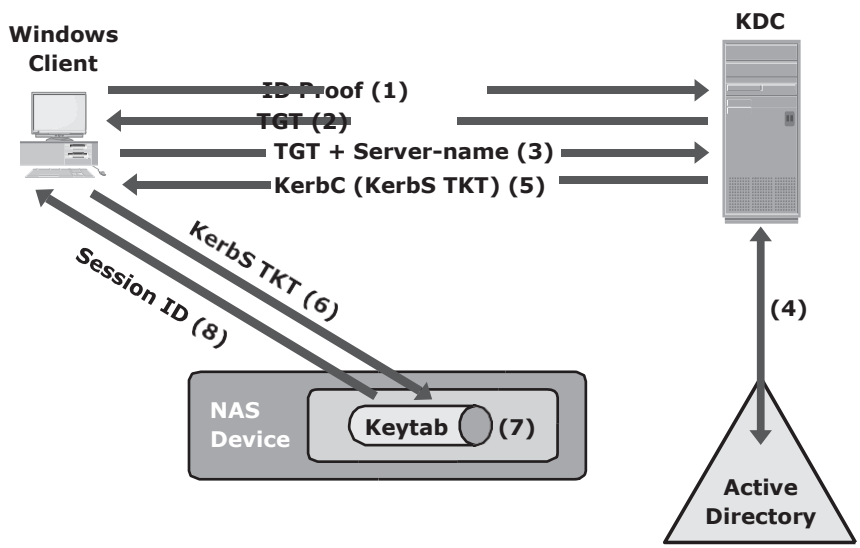


Figure 14-8: Kerberos authorization

Network-Layer Firewalls

Because NAS devices utilize the IP protocol stack, they are vulnerable to various attacks initiated through the public IP network. Network layer firewalls are implemented in NAS environments to protect the NAS devices from these security threats. These network-layer firewalls can examine network packets and compare them to a set of configured security rules. Packets that are not authorized by a security rule are dropped and not allowed to continue to the destination. Rules can be established based on a source address (network or host), a destination address (network or host), a port, or a combination of those

factors (source IP, destination IP, and port number). The effectiveness of a firewall depends on how robust and extensive the security rules are. A loosely defined rule set can increase the probability of a security breach.

Figure 14-9 depicts a typical firewall implementation. A demilitarized zone (DMZ) is commonly used in networking environments. A DMZ provides a means to secure internal assets while allowing Internet-based access to various resources. In a DMZ environment, servers that need to be accessed through the Internet are placed between two sets of firewalls. Application-specific ports, such as HTTP or FTP, are allowed through the firewall to the DMZ servers. However, no Internet-based traffic is allowed to penetrate the second set of firewalls and gain access to the internal network.

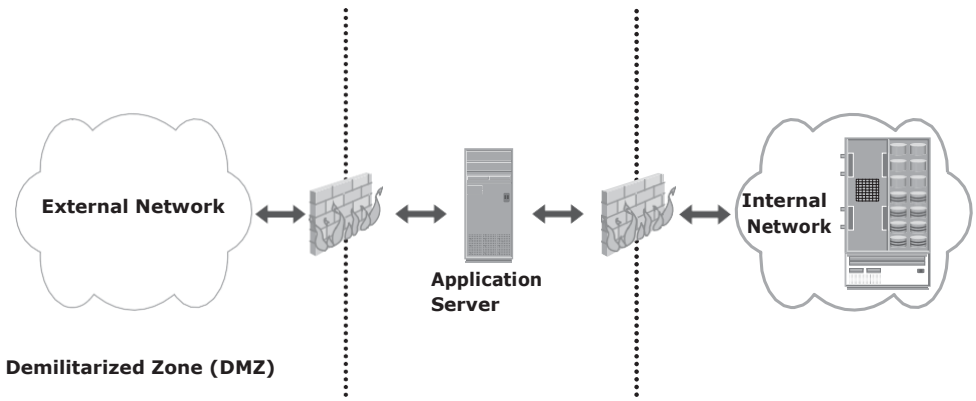


Figure 14-9: Securing a NAS environment with a network-layer firewall

The servers in the DMZ may or may not be allowed to communicate with internal resources. In such a setup, the server in the DMZ is an Internet-facing web application accessing data stored on a NAS device, which may be located on the internal private network. A secure design would serve only data to internal and external applications through the DMZ.

IP SAN

This section describes some of the basic security mechanisms used in IP SAN environments. The *Challenge-Handshake Authentication Protocol* (CHAP) is a basic authentication mechanism that has been widely adopted by network devices and hosts. CHAP provides a method for initiators and targets to authenticate each other by utilizing a secret code or password. CHAP secrets are usually random secrets of 12 to 128 characters. The secret is never exchanged directly over the communication channel; rather, a one-way hash function converts it into a hash value, which is then exchanged. A hash function, using the MD5 algorithm, transforms data in such a way that the result is unique and cannot be changed back to its original form. Figure 14-10 depicts the CHAP authentication process.

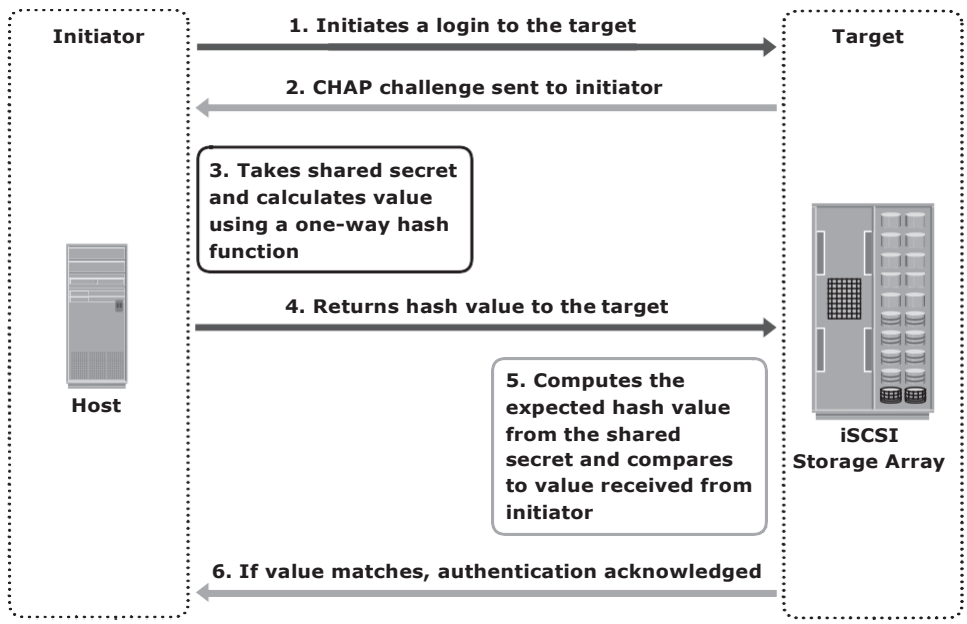


Figure 14-10: Securing IPSAN with CHAP authentication

If the initiator requires reverse CHAP authentication, the initiator authenticates the target by using the same procedure. The CHAP secret must be configured on the initiator and the target. A CHAP entry, composed of the name of a node and the secret associated with the node, is maintained by the target and the initiator. The same steps are executed in a two-way CHAP authentication scenario. After these steps are completed, the initiator authenticates the target. If both authentication steps succeed, then data access is allowed. CHAP is often used because it is a fairly simple protocol to implement and can be implemented across a number of disparate systems.

iSNS discovery domains function in the same way as FC zones. Discovery domains provide functional groupings of devices in an IP-SAN. For devices to

communicate with one another, they must be configured in the same discovery domain. State change notifications (SCNs) inform the iSNS server when devices are added to or removed from a discovery domain. Figure 14-11 depicts the discovery domains in iSNS.

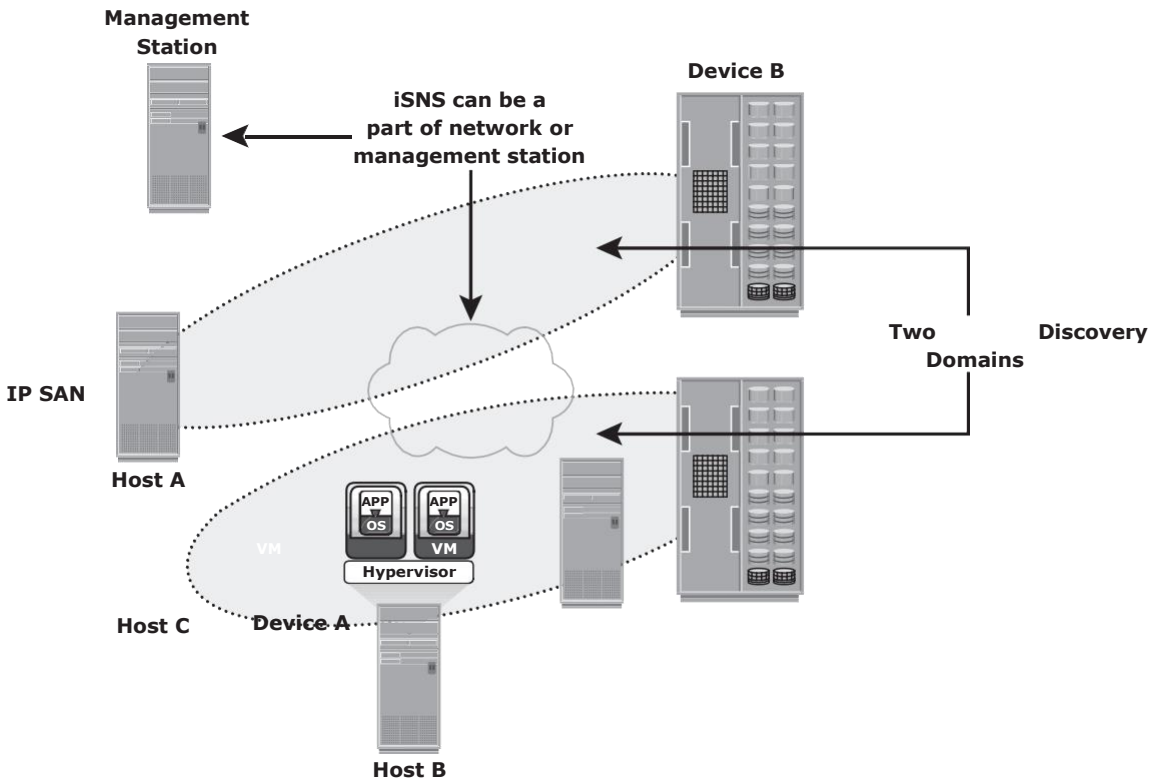


Figure 14-11: Securing IPSAN with iSNS discovery domains

Securing Storage Infrastructure in Virtualized and Cloud Environments

This chapter, so far, focused only on the security threats and measures in a traditional data center. These threats and measures are also applicable to information storage in virtualized and cloud environments. However, virtualized and cloud computing environments pose additional threats to an organization's data due to multitenancy and lack of control over the cloud resources. A public cloud has more security concerns compared to a private cloud and demands additional counter measures. This is because in a public cloud, cloud users (consumers) usually have limited control over resources, and therefore, enforcement of security mechanisms by consumers is comparatively difficult.

From a security perspective, both consumers and cloud service providers (CSP) have several security concerns and face multiple threats. Security concerns and security measures are detailed next.

Security Concerns

Organizations are rapidly adopting virtualization and cloud computing, however they have some security concerns. These key security concerns are multitenancy, velocity of attack, information assurance, and data privacy.

Multitenancy, by virtue of virtualization, enables multiple independent tenants to be serviced using the same set of storage resources. In spite of the benefits offered by multitenancy, it is still a key security concern for users and service providers. Colocation of multiple VMs in a single server and sharing the same resources increase the attack surface. It may happen that business critical data of one tenant is accessed by other competing tenants who run applications using the same resources.

Velocity-of-attack refers to a situation in which any existing security threat in the cloud spreads more rapidly and has a larger impact than that in the traditional data center environments. *Information assurance* for users ensures confidentiality, integrity, and availability of data in the cloud. Also the cloud user needs assurance that all the users operating on the cloud are genuine and access the data only with legitimate rights and scope.

Data privacy is also a major concern in a virtualized and cloud environment. A CSP needs to ensure that Personally Identifiable Information (PII) about its clients is legally protected from any unauthorized disclosure.

Security Measures

Security measures can be implemented at the compute, network, and storage levels. These security measures implemented at three layers mitigate the risks in virtualized and cloud environments.

Security at the Compute Level

Securing a compute infrastructure includes enforcing the security of the physical server, hypervisor, VM, and guest OS (OS running within a virtual machine). *Physical server security* involves implementing user authentication and authorization mechanisms. These mechanisms identify users and provide access privileges on the server. To minimize the attack surface on the server, unused hardware components, such as NICs, USB ports, or drives, should be removed or disabled. A *hypervisor* is a single point of security failure for all the VMs running on it. Rootkits and malware installed on a hypervisor make detection difficult for the antivirus software installed on the guest OS. To protect against attacks,

security-critical hypervisor updates should be installed regularly. Further, the hypervisor management system must also be protected. Malicious attacks and infiltration to the management system can impact all the existing VMs and allow attackers to create new VMs. Access to the management system should be restricted to authorized administrators. Furthermore, there must be a separate firewall installed between the management system and the rest of the network. *VM isolation* and *hardening* are some of the common security mechanisms to effectively safeguard a VM from an attack. VM isolation helps to prevent a compromised guest OS from impacting other guest OSs. VM isolation is implemented at the hypervisor level. Apart from isolation, VMs should be hardened against security threats. Hardening is a process to change the default configuration to achieve greater security.

Apart from the measures to secure a hypervisor and VMs, virtualized and cloud environments also require further measures on the guest OS and application levels.

Security at the Network Level

The key security measures that minimize vulnerabilities at the network layer are firewall, intrusion detection, demilitarized zone (DMZ), and encryption of data-in-flight.

A *firewall* protects networks from unauthorized access while permitting only legitimate communications. In a virtualized and cloud environment, a firewall can also protect hypervisors and VMs. For example, if remote administration is enabled on a hypervisor, access to all the remote administration interfaces should be restricted by a firewall. A firewall also secures VM-to-VM traffic. This firewall service can be provided using a *Virtual Firewall* (VF). A VF is a firewall service running entirely on the hypervisor. A VF provides packet filtering and monitoring of the VM-to-VM traffic. A VF gives visibility and control over the VM traffic and enforces policies at the VM level.

Intrusion Detection (ID) is the process to detect events that can compromise the confidentiality, integrity, or availability of a resource. An ID System (IDS)

automatically analyzes events to check whether an event or a sequence of events match a known pattern for anomalous activity, or whether it is (statistically) different from most of the other events in the system. It generates an alert if an irregularity is detected. DMZ and data encryption are also deployed as security measures in the virtualized and cloud environments. However, these deployments work in the same way as in the traditional data center.

Security at the Storage Level

Major threats to storage systems in virtualized and cloud environments arise due to compromises at compute, network, and physical security levels. This is because access to storage systems is through compute and network infrastructure. Therefore, adequate security measures should be in place at the compute and network levels to ensure storage security. Common security mechanisms that protect storage include the following:

- Access control methods to regulate which users and processes access the data on the storage systems
- Zoning and LUN masking
- Encryption of data-at-rest (on the storage system) and data-in-transit. Data encryption should also include encrypting backups and storing encryption keys separately from the data.
- Data shredding that removes the traces of the deleted data

Apart from these mechanisms, isolation of different types of traffic using VSANs further enhances the security of storage systems. In the case of storage utilized by hypervisors, additional security steps are required to protect the storage. Storage for hypervisors using clustered file systems supporting multiple VMs may require separate LUNs for VM components and VM data.

Concepts in Practice: RSA and VMware Security Products

RSA, the security division of EMC, is the premier provider of security, risk, and compliance solutions, helping organizations to solve their most complex and sensitive security challenges.

VMware offers secure and robust virtualization solutions for virtualized and cloud environments. This section provides a brief introduction to RSA SecureID, RSA Identity and Access Management, RSA Data Protection Manager, and VMware vShield.

RSA SecureID

RSA SecurID two-factor authentication provides an added layer of security to ensure that only valid users have access to systems and data. RSA SecurID is based on something a user knows (a password or PIN) and something a user has (an authenticator device). It provides a much more reliable level of user authentication than reusable passwords. It generates a new one-time password code every 60 seconds, making it difficult for anyone other than the genuine user to input the correct token code at any given time. To access their resources, users combine their secret Personal Identification Number (PIN) with the token code that appears on their SecurID authenticator display at that given time. The result is a unique, one-time password to assure a user's identity.

RSA Identity and Access Management

The RSA Identity and Access Management product provides identity, security, and access-controls management for physical, virtual, and cloud-based environments through access management. It enables trusted identities to freely and securely interact with systems and access. The RSA Identity and Access Management family has two products: *RSA Access Manager* and *RSA Federated Identity Manager*. RSA Access Manager enables organizations to centrally manage authentication and authorization policies for a large number of users, online web portals, and application resources. Access Manager provides seamless user access with single sign-on (SSO) and preserves identity context for greater security. RSA Federated Identity Manager enables end users to collaborate with business partners, outsourced service providers, and supply-chain partners or across multiple offices or agencies all with a single identity and logon.

RSA Data Protection Manager

RSA Data Protection Manager enables deployment of encryption, tokeniza- tion, and enterprise key management simply and affordably. The RSA Data Protection Manager family is composed of two products: *Application Encryption and Tokenization* and *Enterprise Key Management*.

- Application Encryption and Tokenization with RSA Data Protection Manager helps to achieve compliance with regulations related to PII by quickly embedding the encryption and tokenization of sensitive data and helping to prevent data loss. It works at the point of creation, ensuring that the data stays encrypted as it is transmitted and stored.
- Enterprise key management is an easy-to-use management tool for encrypting keys at the database, file server, and storage layers. It is designed to simplify the deployment of encryption throughout the enterprise. It also helps to ensure that information is properly secured and fully accessible when needed at any point in its life cycle.

VMware vShield

The VMware vShield family includes three products: *vShield App*, *vShield Edge*, and *vShield Endpoint*.

VMware vShield App is a hypervisor-based application-aware firewall solution. It protects applications in a virtualized environment from network-based threats by providing visibility into network communications and enforcing granular policies with security groups. VMware vShield App observes network activity between virtual machines to define and refine firewall policies and secure business processes through detailed reporting of application traffic. VMware vShield Edge provides comprehensive perimeter network security for a virtualized environment. It is deployed as a virtual appliance and serves as a network security gateway for all the hosts within the virtualized environment. It provides many services including firewall, VPN, and Dynamic Host Configuration Protocol (DHCP) services.

VMware vShield Endpoint consists of a hardened special security VM with a third party antivirus software. VMware vShield Endpoint streamlines and accelerates antivirus and antimalware deployment because antivirus engine and signature files are updated only within the special security VM. VMware vShield Endpoint improves VM performance by offloading file scanning and other tasks from VMs to the security VM. It prevents antivirus storms and bottlenecks associated with multiple simultaneous antivirus and antimalware scans and updates. It also satisfies audit requirements with detailed logging of antivirus and antimalware activities.

Monitoring the Storage Infrastructure

Monitoring is one of the most important aspects that forms the basis for managing storage infrastructure resources. Monitoring provides the performance and accessibility status of various components. It also enables administrators to perform essential management activities. Monitoring also helps to analyze the utilization and consumption of various storage infrastructure resources. This analysis facilitates capacity planning, forecasting, and optimal use of these resources. Storage infrastructure environment parameters such as heating and power supplies are also monitored.

Monitoring Parameters

Storage infrastructure components should be monitored for accessibility, capacity, performance, and security. *Accessibility* refers to the availability of a component to perform its desired operation during a specified time period. Monitoring the accessibility of hardware components (for example, a port, an HBA, or a disk drive) or software component (for example, a database) involves checking their availability status by reviewing the alerts generated from the system. For example, a port failure might result in a chain of availability alerts.

A storage infrastructure uses redundant components to avoid a single point

of failure. Failure of a component might cause an outage that affects application availability, or it might cause performance degradation even though accessibility is not compromised. Continuously monitoring for expected accessibility of each component and reporting any deviation helps the administrator to identify failing components and plan corrective action to maintain SLA requirements.

Capacity refers to the amount of storage infrastructure resources available. Examples of capacity monitoring include examining the free space available on a file system or a RAID group, the mailbox quota allocated to users, or the numbers of ports available on a switch. Inadequate capacity leads to degraded performance or even application/service unavailability. *Capacity monitoring* ensures uninterrupted data availability and scalability by averting outages before they occur. For example, if 90 percent of the ports are utilized in a particular

SAN fabric, this could indicate that a new switch might be required if more arrays and servers need to be installed on the same fabric. Capacity monitoring usually leverages analytical tools to perform trend analysis. These trends help to understand future resource requirements and provide an estimation on the time line to deploy them.

Performance monitoring evaluates how efficiently different storage infrastructure components are performing and helps to identify bottlenecks. Performance monitoring measures and analyzes behavior in terms of response time or the ability to perform at a certain predefined level. It also deals with the utilization of resources, which affects the way resources behave and respond. Performance measurement is a complex task that involves assessing various components on several interrelated parameters. The number of I/Os performed by a disk, application response time, network utilization, and server-CPU utilization are examples of performance parameters that are monitored.

Monitoring a storage infrastructure for security helps to track and prevent unauthorized access, whether accidental or malicious. *Security monitoring* helps to track unauthorized configuration changes to storage infrastructure resources. For example, security monitoring tracks and reports the initial zoning configuration performed and all the subsequent changes. Security monitoring also detects unavailability of information to authorized users due to a security breach. Physical security of a storage infrastructure can also be continuously monitored using badge readers, biometric scans, or video cameras.

Components Monitored

Hosts, networks, and storage are the components within the storage environment that should be monitored for accessibility, capacity, performance, and security. These components can be physical or virtualized.

Hosts

The accessibility of a host depends on the availability status of the hardware components and the software processes running on it. For example, a host's NIC failure might cause inaccessibility of the host to its user. Server clustering is a mechanism that provides high availability if a server failure occurs.

Monitoring the file system capacity utilization of a host is important to ensure that sufficient storage capacity is available to the applications. Running out of file system space disrupts application availability. Monitoring helps estimate the file system's growth rate and predict when it will reach 100 percent. Accordingly, the administrator can extend (manually or automatically) the file system's space proactively to prevent application outage. Use of virtual provisioning technology

enables efficient management of storage capacity requirements but is highly dependent on capacity monitoring.

Host performance monitoring mainly involves a status check on the utilization of various server resources, such as CPU and memory. For example, if a server running an application is experiencing 80 percent of CPU utilization continuously, it indicates that the server may be running out of processing power, which can lead to degraded performance and slower response time. Administrators can take several actions to correct the problem, such as upgrading or adding more processors and shifting the workload to different servers. In a virtualized environment, additional CPU and memory may be allocated to VMs dynamically from the pool, if available, to meet performance requirements.

Security monitoring on servers involves tracking of login failures and execution of unauthorized applications or software processes. Proactive measures against unauthorized access to the servers are based on the threat identified. For example, an administrator can block user access if multiple login failures are logged.

Storage Network

Storage networks need to be monitored to ensure uninterrupted communication between the server and the storage array. Uninterrupted access to data over the storage network depends on the accessibility of the physical and logical components of the storage network. The physical components of a storage network include switches, ports, and cables. The logical components include constructs, such as zones. Any failure in the physical or logical components causes data unavailability. For example, errors in zoning, such as specifying the wrong WWN of a port, result in failure to access that port, which potentially prevents access from a host to its storage.

Capacity monitoring in a storage network involves monitoring the number of available ports in the fabric, the utilization of the interswitch links, or individual ports, and each interconnect device in the fabric. Capacity monitoring provides all the required inputs for future planning and optimization of fabric resources.

Monitoring the performance of the storage network enables assessing individual component performance and helps to identify network bottlenecks. For example, monitoring port performance involves measuring the receive or transmit link utilization metrics, which indicates how busy the switch port is. Heavily used ports can cause queuing of I/Os on the server, which results in poor performance.

For IP networks, monitoring the performance includes monitoring network latency, packet loss, bandwidth utilization for I/O, network errors, packet retransmission rates, and collisions.

Storage network security monitoring provides information about any unauthorized change to the configuration of the fabric — for example, changes to the zone policies that can affect data security. Login failures and unauthorized access to switches for performing administrative changes should be logged and monitored continuously.

Storage

The accessibility of the storage array should be monitored for its hardware components and various processes. Storage arrays are typically configured with redundant components, and therefore individual component failure does not usually affect their accessibility. However, failure of any process in the storage array might disrupt or compromise business operations. For example, the failure of a replication task affects disaster recovery capabilities. Some storage arrays provide the capability to send messages to the vendor's support center if hardware or process failures occur, referred to as a *call home*.

Capacity monitoring of a storage array enables the administrator to respond to storage needs preemptively based on capacity utilization and consumption trends. Information about unconfigured and unallocated storage space enables the administrator to decide whether a new server can be allocated storage capacity from the storage array.

A storage array can be monitored using a number of performance metrics, such as utilization rates of the various storage array components, I/O response time, and cache utilization. For example, an over utilized storage array component might lead to performance degradation.

A storage array is usually a shared resource, which may be exposed to security threats. Monitoring security helps to track unauthorized configuration of the storage array and ensures that only authorized users are allowed to access it.

Monitoring Examples

A storage infrastructure requires implementation of an end-to-end solution to actively monitor all the parameters of its components. Early detection and preemptive alerting ensure uninterrupted services from critical assets. In addition, the monitoring tool should analyze the impact of a failure and deduce the root cause of symptoms.

Accessibility Monitoring

Failure of any component might affect the accessibility of one or more components due to their interconnections and dependencies. Consider an implementation in a storage infrastructure with three servers: H1, H2, and H3. All the servers

are configured with two HBAs, each connected to the production storage array through two switches, SW1 and SW2, as shown in Figure 15-1. All the servers share two storage ports on the storage array and multipathing software is installed on all the servers.

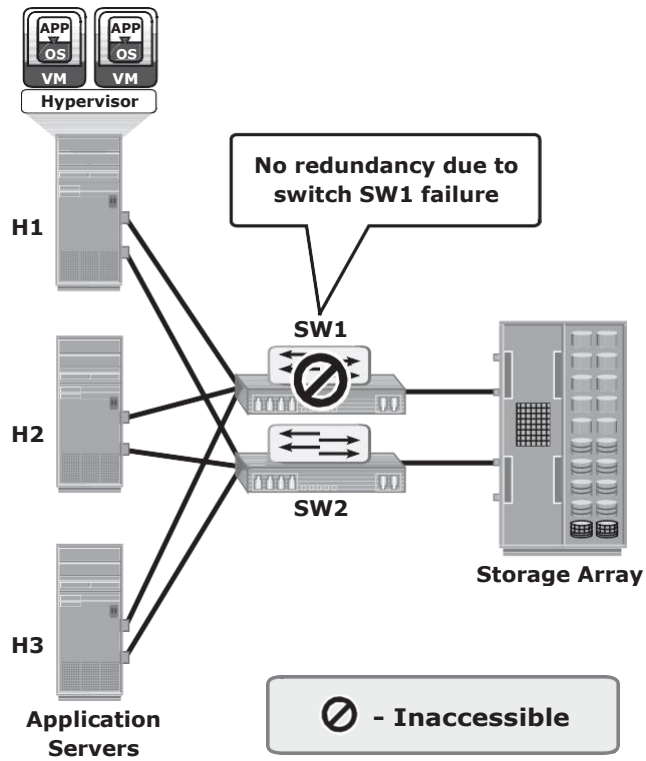


Figure 15-1: Switch failure in a storage infrastructure

If one of the switches (SW1) fails, the multipathing software initiates a path failover, and all the servers continue to access data through the other switch, SW2. However, due to the absence of a redundant switch, a second switch failure could result in inaccessibility of the array. Monitoring for accessibility enables detecting the switch failure and helps an administrator to take corrective action before another failure occurs.

In most cases, the administrator receives symptom alerts for a failing component and can initiate actions before the component fails.

Capacity Monitoring

In the scenario shown in Figure 15-2, servers H1, H2, and H3 are connected to the production array through two switches, SW1 and SW2. Each of the servers

is allocated storage on the storage array. When a new server is deployed in this configuration, the applications on the new server need to be given storage capacity from the production storage array. Monitoring the available capacity (configurable and unallocated) on the array helps to proactively decide whether the array can provide the required storage to the new server. Also, monitoring the available number of ports on SW1 and SW2 helps to decide whether the new server can be connected to the switches.

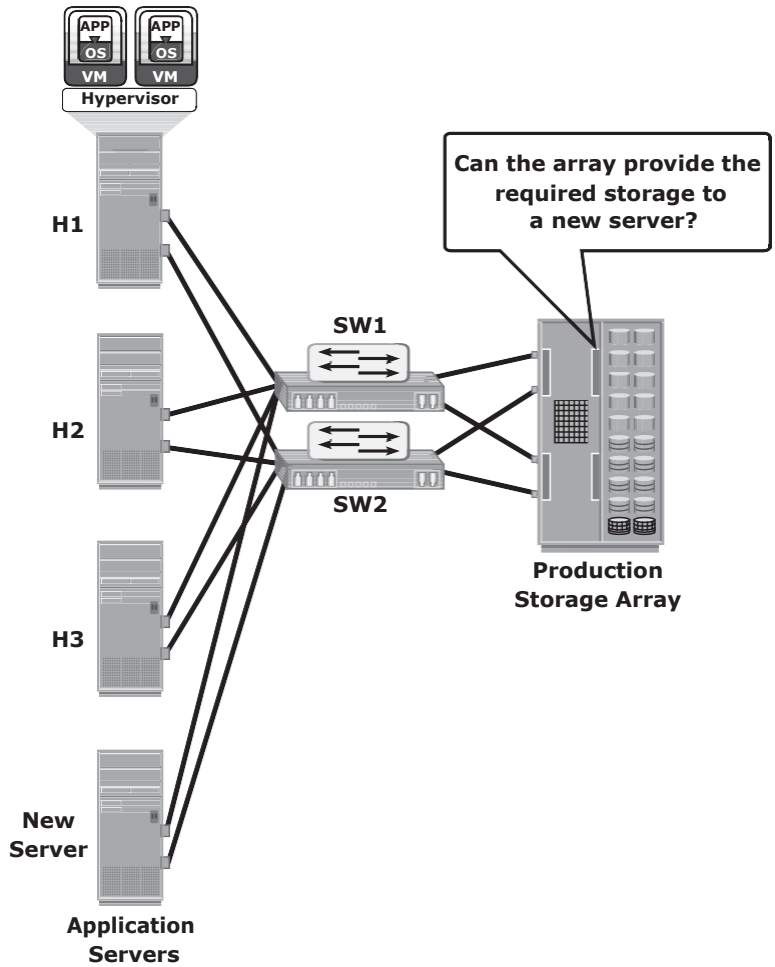


Figure 15-2: Monitoring storage array capacity

The following example illustrates the importance of monitoring the file system capacity on file servers. Figure 15-3 (a) illustrates the environment of a file system when full and that results in application outage when no capacity

monitoring is implemented. Monitoring can be configured to issue a message when thresholds are reached on the file system capacity. For example, when the file system reaches 66 percent of its capacity, a warning message is issued, and a critical message is issued when the file system reaches 80 percent of its capacity (see Figure 15-3 [b]). This enables the administrator to take action to extend the file system before it runs out of capacity. Proactively monitoring the file system can prevent application outages caused due to lack of file system space.

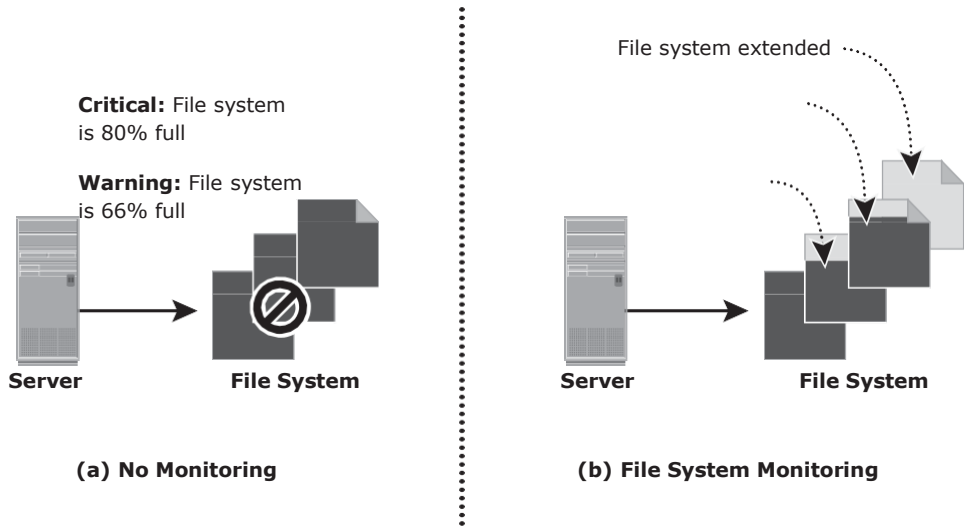


Figure 15-3: Monitoring server file system space

Performance Monitoring

The example shown in Figure 15-4 illustrates the importance of monitoring performance on storage arrays. In this example, servers H1, H2, and H3 (with two HBAs each) are connected to the storage array through switch SW1 and SW2. The three servers share the same storage ports on the storage array to access LUNs. A new server running an application with a high work load must be deployed to share the same storage port as H1, H2, and H3.

Monitoring array port utilization ensures that the new server does not adversely affect the performance of the other servers. In this example, utilization of the shared storage port is shown by the solid and dotted lines in the graph. If the port utilization prior to deploying the new server is close to 100 percent, then deploying the new server is not recommended because it might impact the

performance of the other servers. However, if the utilization of the port prior to deploying the new server is closer to the dotted line, then there is room to add a new server.

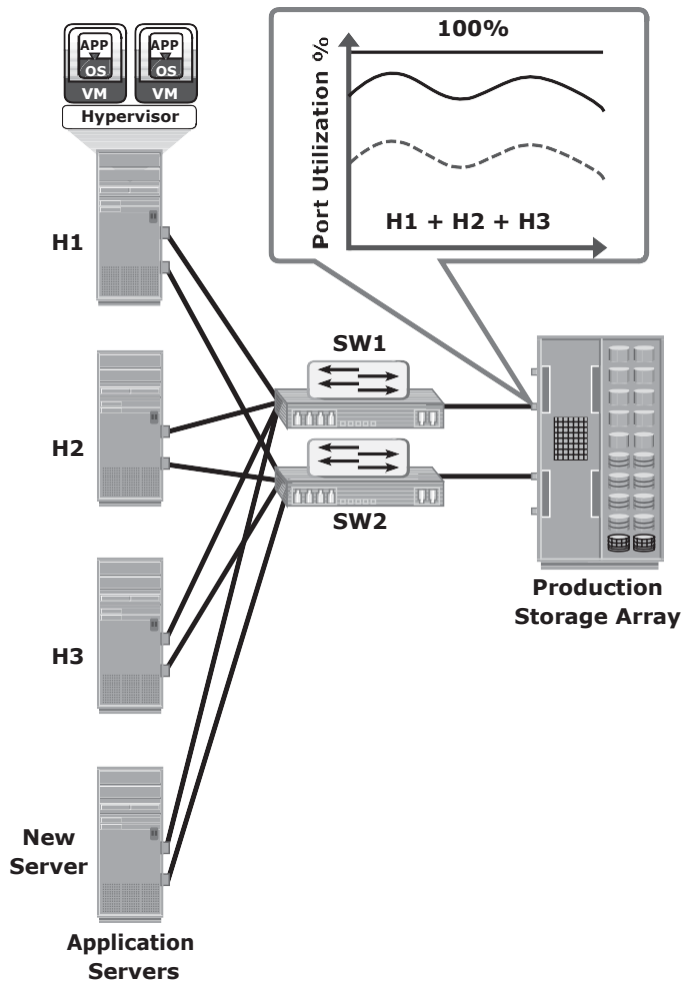


Figure 15-4: Monitoring array port utilization

Most servers offer tools that enable monitoring of server CPU usage. For example, Windows Task Manager displays CPU and memory usage, as shown in Figure 15-5. However, these tools are inefficient at monitoring hundreds of servers running in a data-center environment. A data-center environment requires intelligent performance monitoring tools that are capable of monitoring many servers simultaneously.

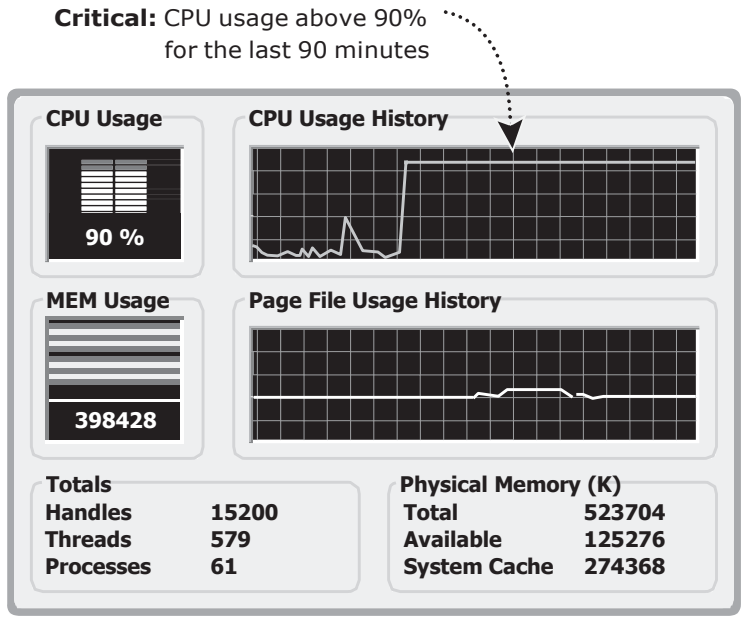


Figure 15-5: Monitoring the CPU and memory usage of a server

Security Monitoring

The example shown in Figure 15-6 illustrates the importance of monitoring security in a storage array.

In this example, the storage array is shared between two workgroups, WG1 and WG2. The data of WG1 should not be accessible to WG2 and vice versa. A user from WG1 might try to make a local replica of the data that belongs to WG2. If this action is not monitored or recorded, it is difficult to track such a violation of information security. Conversely, if this action is monitored, a warning message can be sent to prompt a corrective action or at least enable discovery as part of regular auditing operations.

An example of host security monitoring is tracking of login attempts at the host. The login is authorized if the login ID and password entered are correct; or the login attempt fails. These login failures might be accidental (mistyping) or a deliberate attempt to access a server. Many servers usually allow a fixed number of successive login failures, prohibiting any additional attempts after these login failures. In a monitored environment, the login information is recorded in a system log file, and three successive login failures trigger a message, warning of a possible security threat.

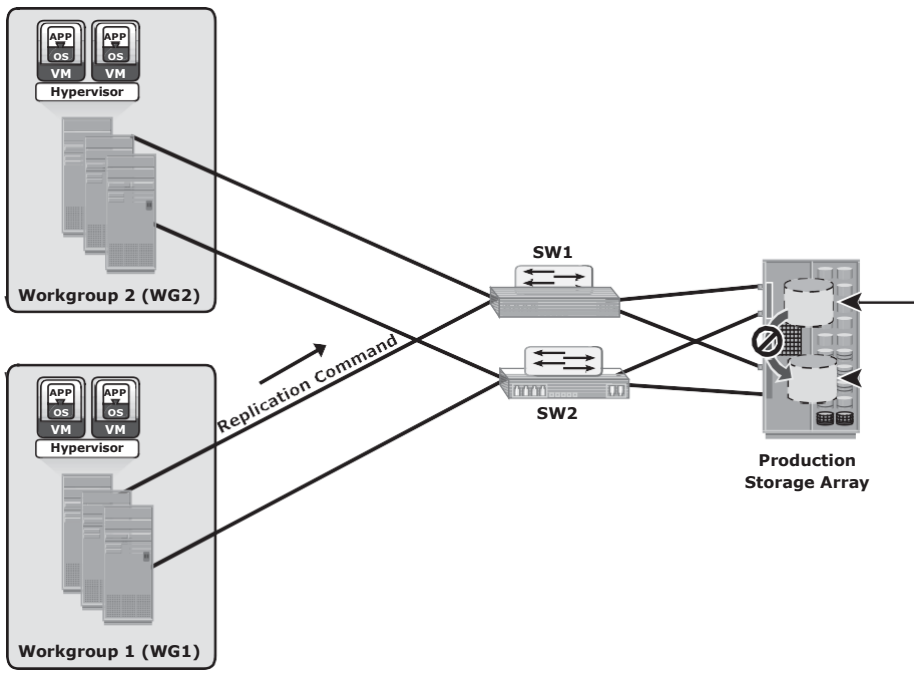


Figure 15-6: Monitoring security in a storage array

Alerts

Alerting of events is an integral part of monitoring. Alerting keeps administrators informed about the status of various components and processes — for example, conditions such as failure of power, disks, memory, or switches, which can impact the availability of services and require immediate administrative attention. Other conditions, such as a file system reaching a capacity threshold or a soft media error on disks, are considered warning signs and may also require administrative attention.

Monitoring tools enable administrators to assign different severity levels based on the impact of the alerted condition. Whenever a condition with a particular severity level occurs, an alert is sent to the administrator, a script is triggered, or an incident ticket is opened to initiate a corrective action. Alert classifications can range from information alerts to fatal alerts. *Information alerts* provide useful information but do not require any intervention by the administrator. The creation of a zone or LUN is an example of an information alert. *Warning alerts* require administrative attention so that the alerted condition is contained and

does not affect accessibility. For example, if an alert indicates that the number of soft media errors on a disk is approaching a predefined threshold value, the administrator can decide whether the disk needs to be replaced. *Fatal alerts* require immediate attention because the condition might affect overall performance, security, or availability. For example, if a disk fails, the administrator must ensure that it is replaced quickly.

Continuous monitoring, with automated alerting, enables administrators to respond to failures quickly and proactively. Alerting provides information that helps administrators prioritize their response to events.

Storage Infrastructure Management Activities

The pace of information growth, proliferation of applications, heterogeneous infrastructure, and stringent service-level requirements have resulted in increased complexity of managing storage infrastructures. However, the emergence of storage virtualization and other technologies, such as data deduplication and compression, virtual provisioning, federated storage access, and storage tiering, have enabled administrators to efficiently manage storage resources.

The key storage infrastructure management activities performed in a data center can be broadly categorized into availability management, capacity management, performance management, security management, and reporting.

Availability Management

A critical task in availability management is establishing a proper guideline based on defined service levels to ensure availability. *Availability management* involves all availability-related issues for components or services to ensure that service levels are met. A key activity in availability management is to provision redundancy at all levels, including components, data, or even sites. For example, when a server is deployed to support a critical business function, it requires high availability. This is generally accomplished by deploying two or more HBAs, multipathing software, and server clustering. The server must be connected to the storage array using at least two independent fabrics and switches that have built-in redundancy. In addition, the storage arrays should have built-in redundancy for various components and should support local and remote replication.

Capacity Management

The goal of *capacity management* is to ensure adequate availability of resources based on their service level requirements. Capacity management also involves optimization of capacity based on the cost and future needs. Capacity management

provides capacity analysis that compares allocated storage to forecasted storage on a regular basis. It also provides trend analysis based on the rate of consumption, which must be rationalized against storage acquisition and deployment timetables. Storage provisioning is an example of capacity management. It involves activities, such as creating RAID sets and LUNs, and allocating them to the host. Enforcing capacity quotas for users is another example of capacity management. Provisioning a fixed amount of user quotas restricts users from exceeding the allocated capacity.

Technologies, such as data deduplication and compression, have reduced the amount of data to be backed up and thereby reduced the amount of storage capacity to be managed.

Performance Management

Performance management ensures the optimal operational efficiency of all components. Performance analysis is an important activity that helps to identify the performance of storage infrastructure components. This analysis provides information on whether a component meets expected performance levels.

Several performance management activities need to be performed when deploying a new application or server in the existing storage infrastructure. Every component must be validated for adequate performance capabilities as defined by the service levels. For example, to optimize the expected performance levels, activities on the server, such as the volume configuration, database design or application layout, configuration of multiple HBAs, and intelligent multipathing software, must be fine-tuned. The performance management tasks on a SAN include designing and implementing sufficient ISLs in a multiswitch fabric with adequate bandwidth to support the required performance levels. The storage array configuration tasks include selecting the appropriate RAID type, LUN layout, front-end ports, back-end ports, and cache configuration, when considering the end-to-end performance.

Security Management

The key objective of the *security management* activity is to ensure confidentiality, integrity, and availability of information in both virtualized and nonvirtualized environments. Security management prevents unauthorized access and configuration of storage infrastructure components. For example, while deploying an application or a server, the security management tasks include managing the user accounts and access policies that authorize users to perform role-based activities. The security management tasks in a SAN environment include configuration of zoning to restrict an unauthorized HBA from accessing specific storage array ports. Similarly, the security management task on a storage array includes LUN masking that restricts a host's access to intended LUNs only.

Reporting

Reporting on a storage infrastructure involves keeping track and gathering information from various components and processes. This information is compiled to generate reports for trend analysis, capacity planning, chargeback, and performance. Capacity planning reports contain current and historic information about the utilization of storage, file systems, database tablespace, ports, and so on. Configuration and asset management reports include details about device allocation, local or remote replicas, and fabric configuration. This report also lists all the equipment, with details, such as their purchase date, lease status, and maintenance records. Chargeback reports contain information about the allocation or utilization of storage infrastructure components by various departments or user groups. Performance reports provide details about the performance of various storage infrastructure components.

Storage Infrastructure Management in a Virtualized Environment

Virtualization technology has dramatically changed the complexity of storage infrastructure management. In fact, flexibility and ease of management are key drivers for wide adoption of virtualization at all layers of the IT infrastructure.

Storage virtualization has enabled dynamic migration of data and extension of storage volumes. Due to dynamic extension, storage volumes can be expanded nondisruptively to meet both capacity and performance requirements. Because virtualization breaks the bond between the storage volumes presented to the host and its physical storage, data can be migrated both within and across data centers without any downtime. This has made the administrator's tasks easier while reconfiguring the physical environment.

Virtual storage provisioning is another tool that has changed the infrastructure management cost and complexity scenario. In conventional provisioning, storage capacity is provisioned upfront in anticipation of future growth. Because growth is uneven, some users or applications find themselves running out of capacity, whereas others have excess capacity that remains underutilized. Use of virtual provisioning can address this challenge and make capacity management less challenging. In virtual provisioning, storage is allocated from the shared pool to hosts on-demand. This improves the storage capacity utilization, and thereby reduces capacity management complexities.

Virtualization has also contributed to network management efficiency. VSANs and VLANs made the administrator's job easier by isolating different

networks logically using management tools rather than physically separating them. Disparate virtual networks can be created on a single physical network, and reconfiguration of nodes can be done quickly without any physical changes. It has also addressed some of the security issues that might exist in a conventional environment.

On the host side, compute virtualization has made host deployment, reconfiguration, and migration easier than physical environment. Compute, application, and memory virtualization have not only improved provisioning, but also contributed to the high availability of resources.

Storage Management Examples

The following section provides examples of various storage management activities.

Example 1: Storage Allocation to a New Server/Host

Consider the deployment of a new RDBMS server to the existing nonvirtualized storage infrastructure. As a part of storage management activities, first, the administrator needs to install and configure the HBAs and device drivers on the server before it is physically connected to the SAN. Optionally, multipathing software can be installed on the server, which might require additional configuration. Further, storage array ports should be connected to the SAN.

As the next step, the administrator needs to perform zoning on the SAN switches to allow the new server access to the storage array ports via its HBAs. To ensure redundant paths between the server and the storage array, the HBAs of the new server should be connected to different switches and zoned with different array ports.

Further, the administrator needs to configure LUNs on the array and assign these LUNs to the storage array front-end ports. In addition, LUN masking configuration is performed on the storage array, which restricts access to LUNs by a specific server.

The server then discovers the LUNs assigned to it by either a *bus rescan* process or sometimes through a server reboot, depending upon the operating system installed. A volume manager may be used to configure the logical volumes and file systems on the host. The number of logical volumes or file systems to be created depends on how a database or an application is expected to use the storage.The administrator's task also includes installation of a database or an application on the logical volumes or file systems that were created.

The last step is to make the database or application capable of using the new file system space. Figure 15-7 illustrates the activities performed on a server, a SAN, and a storage array for the allocation of storage to a new server.

In a virtualized environment, provisioning storage to a VM that runs an RDBMS requires different administrative tasks.

Similar to a nonvirtualized environment, a physical connection must be established between the physical server, which hosts the VMs, and the storage array through the SAN. At the SAN level, a VSAN can be configured to transfer data between the physical server and the storage array. The VSAN isolates this storage traffic from any other traffic in the SAN. Further, the administrator can configure zoning within the VSAN.

At the storage side, administrators need to create thin LUNs from the shared storage pool and assign these thin LUNs to the storage array front-end ports. Similar to a physical environment, LUN masking needs to be carried out on the storage array.

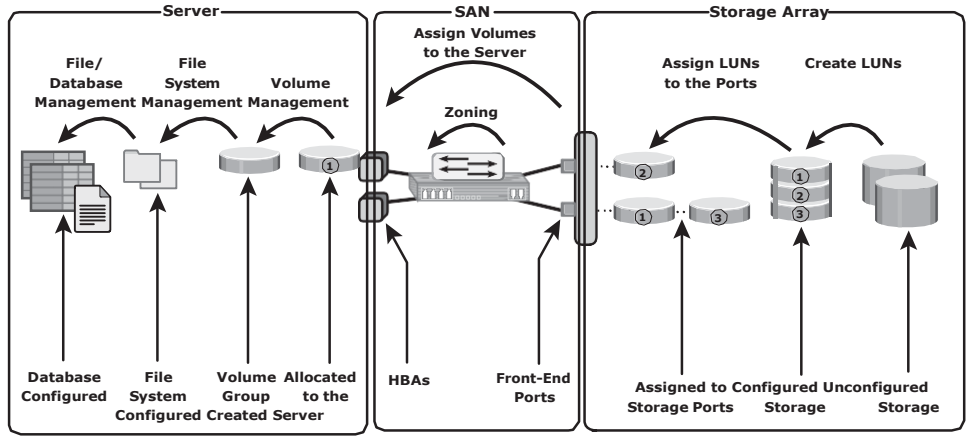


Figure 15-7: Storage allocation tasks

At the physical server side, the hypervisor discovers the assigned LUNs. The hypervisor creates a logical volume and file system to store and manage VM files. Then, the administrator creates a VM and installs the OS and RDBMS on the VM. While creating the VM, the hypervisor creates a virtual disk file and other VM files in the hypervisor file system. The virtual disk file appears to the VM as a SCSI disk and is used to store the RDBMS data. Alternatively, the hypervisor enables virtual provisioning to create a thin virtual disk and assigns it to the VM. Hypervisors usually have native multipathing capabilities. Optionally, a third-party multipathing software may be installed on the hypervisor.

Example 2: File System Space Management

To prevent a file system from running out of space, administrators need to perform tasks to offload data from the existing file system. This includes deleting unwanted files or archiving data that is not accessed for a long time.

Alternatively, an administrator can extend the file system to increase its size and avoid an application outage. The dynamic extension of file systems or a logical volume depends on the operating system or the logical volume manager (LVM) in use. Figure 15-8 shows the steps and considerations for the extension of file systems in the flow chart.

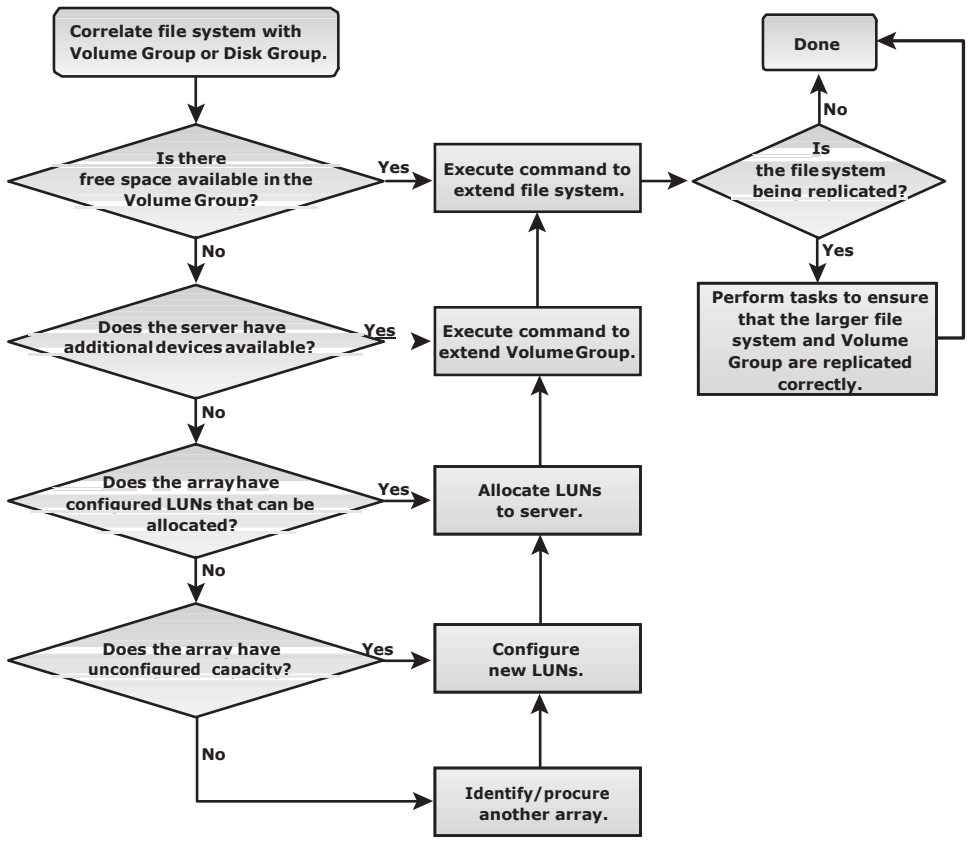


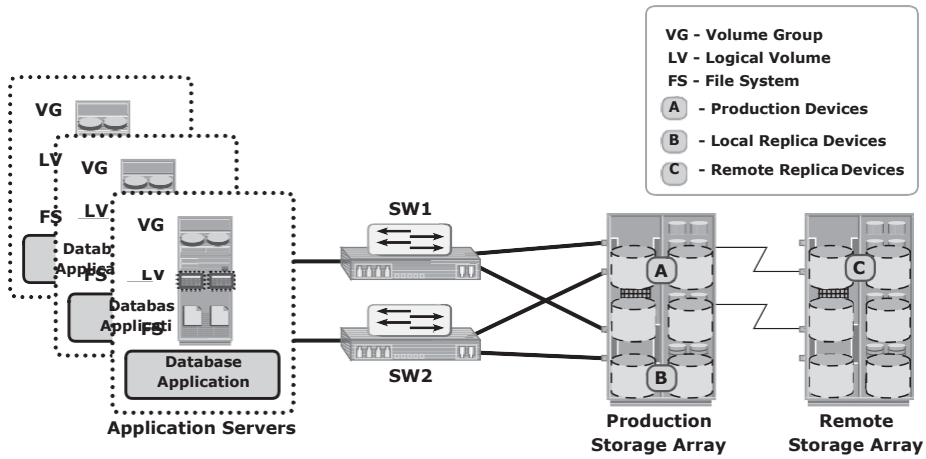
Figure 15-8: Extending a file system

Example 3: Chargeback Report

This example explores the storage infrastructure management tasks necessary to create a chargeback report.

Figure 15-9 shows a configuration deployed in a storage infrastructure. Three servers with two HBAs each connect to a storage array via two switches, SW1 and SW2. Individual departmental applications run on each of the servers. Array replication technology is used to create local and remote replicas. The production device is represented as A, the local replica device as B, and the remote replica device as C.

A report documenting the exact amount of storage resources used by each application is created using a chargeback analysis for each department. If the unit for billing is based on the amount of raw storage (usable capacity plus protection provided) configured for an application used by a department, the exact amount of raw space configured must be reported for each application. Figure 15-9 shows a sample report. The report shows the information for two applications, Payroll_1 and Engineering_1.



Application	Storage (GB)	Production Storage Raw (GB)	Local Replica Storage Raw (GB)	Remote Replica Storage Raw (GB)	Total Storage Raw (GB)	Chargeback Cost $ 5/Raw (GB)
Payroll_1	100	200	100	125	425	$ 2125
Engineering_1	200	250	200	250	700	$ 3500

Figure 15-9: Chargeback report

The first step to determine chargeback costs is to correlate the application with the exact amount of raw storage configured for that application.

As indicated in Figure 15-10, the Payroll_1 application storage space is traced from file systems to logical volumes to volume groups and to the LUNs on the array. When the applications are replicated, the storage space used for local replication and remote replication is also identified. In the example shown, the application is using Source Vol 1 and Vol 2 (in the production array). The replication volumes are Local Replica Vol 1 and Vol 2 (in the production array) and Remote Replica Vol 1 and Vol 2 (in the remote array).

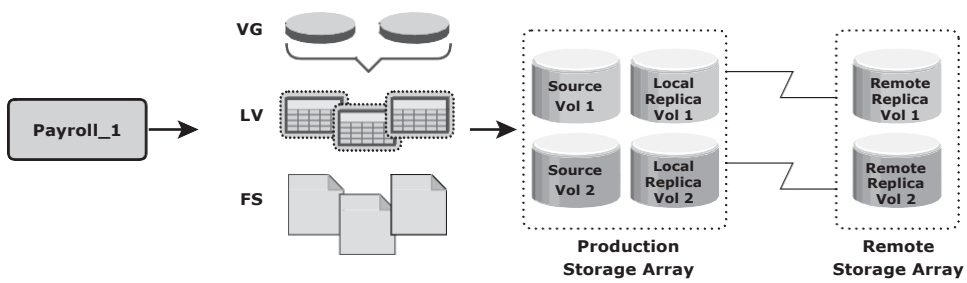


Figure 15-10: Correlation of capacity configured for an application

The amount of storage allocated to the application can be easily computed after the array devices are identified. In this example, consider that Source

Vol 1 and Vol 2 are each 50 GB in size, the storage allocated to the application is 100 GB (50 + 50). The allocated storage for replication is 100 GB for local replication and 100 GB for remote replication. From the allocated storage, the raw storage configured for the application is determined based on the RAID protection that is used for various array devices.

If the Payroll_1 application's production volumes are RAID 1-protected, the raw space used by the production volumes is 200 GB. Assume that the local replicas are on unprotected volumes, and the remote replicas are protected with a RAID 5 configuration, then 100 GB of raw space is used by the local replica and 125 GB by the remote replica. Therefore, the total raw capacity used by the Payroll_1 application is 425 GB. The total cost of storage provisioned for Payroll_1 application will be $2,125 (assume cost per GB of storage is $5). This exercise must be repeated for each application in the enterprise to generate the chargeback report.

Chargeback reports can be extended to include a pre-established cost of other resources, such as the number of switch ports, HBAs, and array ports in the configuration. Chargeback reports are used by data center administrators to ensure that storage consumers are well aware of the costs of the services that they have requested.

Storage Infrastructure Management Challenges

Monitoring and managing today's complex storage infrastructure is challenging. This is due to the heterogeneity of storage arrays, networks, servers, databases, and applications in the environment. For example, heterogeneous storage arrays vary in their capacity, performance, protection, and architectures.

Each of the components in a data center typically comes with vendor-specific tools for management. An environment with multiple tools makes understanding the overall status of the environment challenging because the tools may not be interoperable. Ideally, management tools should correlate information from all components in one place. Such tools provide an end-to-end view of the environment, and a quicker root cause analysis for faster resolution to alerts.

Developing an Ideal Solution

An ideal solution should offer meaningful insight into the status of the overall infrastructure and provide root cause analysis for each failure. This solution should also provide central monitoring and management in a multivendor storage environment and create an end-to-end view of the storage infrastructure.

The benefit of end-to-end monitoring is the ability to correlate one component's behavior with the other. In many cases, looking at each component individually may not be sufficient to reveal the actual cause of the problem. The central monitoring and management system should gather information from all the components and manage them through a single-user interface. In addition, it must provide a mechanism to notify administrators about various events using methods, such as e-mail and Simple Network Management Protocol (SNMP) traps. It should also have the capability to generate monitoring reports and run automated scripts for task automation.

The ideal solution must be based on industry standards, by leveraging common APIs, data model terminology, and taxonomy. This enables the implementation of policy-based management across heterogeneous devices, services, applications, and deployed topologies.

Traditionally, SNMP protocol was the standard used to manage multivendor SAN environments. However, SNMP was inadequate for providing the detailed information required to manage the SAN environment. The unavailability of automatic discovery functions and weak modeling constructs are some inadequacies of SNMP in a SAN environment. Even with these limitations, SNMP still holds a predominant role in SAN management, although newer open storage SAN management standards have emerged to monitor and manage storage environments more effectively.

Storage Management Initiative

The Storage Networking Industry Association (SNIA) has been engaged in an initiative to develop a common storage management interface. SNIA has developed a specification called Storage Management Initiative-Specification (SMI-S). This specification is based on the Web-Based Enterprise Management (WBEM) technology, and Distributed Management Task Force's (DMTF) Common Information Model (CIM). The initiative was formally created to enable broad interoperability and management among heterogeneous storage and SAN components. For more information, see `www.snia.org`.

SMI-S offers substantial benefits to users and vendors. It forms a normalized, abstracted model to which a storage infrastructure's physical and logical components can be mapped. This model is used by management applications, such as storage resource management, device management, and data management, for standardized, end-to-end control of storage resources.

Using SMI-S, device software developers have a unified object model with details about managing the breadth of storage and SAN components. SMI-S-compliant products lead to easier, faster deployment and accelerated adoption of policy-based storage management frameworks. Moreover, SMI-S eliminates the need for the development of vendor-proprietary management interfaces and enables vendors to focus on value-added features.

Enterprise Management Platform

An enterprise management platform (EMP) is a suite of applications that provides an integrated solution for managing and monitoring an enterprise storage infrastructure. These applications have powerful, flexible, unified frameworks that provide end-to-end management of both physical and virtual resources. EMP provides a centrally managed, single point of control for resources throughout the storage environment.

These applications can proactively monitor storage infrastructure components and alert users about events. These alerts are either shown on a console depicting the faulty component in a different color, or they can be configured to send the alert by e-mail. In addition to monitoring, an EMP provides the necessary management functionality, which can be natively implemented into the EMP or can launch the proprietary management utility supplied by the component manufacturer.

An EMP also enables easy scheduling of operations that must be performed regularly, such as the provisioning of resources, configuration management, and fault investigation. These platforms also provide extensive analytical, remedial, and reporting capabilities to ease storage infrastructure management. EMC ControlCenter and EMC Prosphere, described in section 15.7 "Concepts in Practice," are examples of an EMP.

Information Lifecycle Management

In both traditional data center and virtualized environments, managing information can be expensive if not managed appropriately. Along with the tools, an effective management strategy is also required to manage information efficiently. This strategy should address the following key challenges that exist in today's data centers:

- **Exploding digital universe:** The rate of information growth is increasing exponentially. Creating copies of data to ensure high availability and repurposing has contributed to the multifold increase of information growth.
- **Increasing dependency on information:** The strategic use of informa- tion plays an important role in determining the success of a business and provides competitive advantages in the marketplace.
- **Changing value of information**: Information that is valuable today might become less important tomorrow. The value of information often changes over time.

Framing a strategy to meet these challenges involves understanding the value of information over its life cycle. When information is first created, it often has the highest value and is accessed frequently. As the information ages, it is accessed less frequently and is of less value to the organization. Understanding the value

of information helps to deploy the appropriate infrastructure according to the changing value of information.

For example, in a sales order application, the value of the information (customer data) changes from the time the order is placed until the time that the warranty becomes void (see Figure 15-11). The value of the information is highest when a company receives a new sales order and processes it to deliver the product. After the order fulfillment, the customer data does not need to be available for real-time access. The company can transfer this data to less expensive secondary storage with lower performance until a warranty claim or another event triggers its need. After the warranty becomes void, the company can dispose of the information.

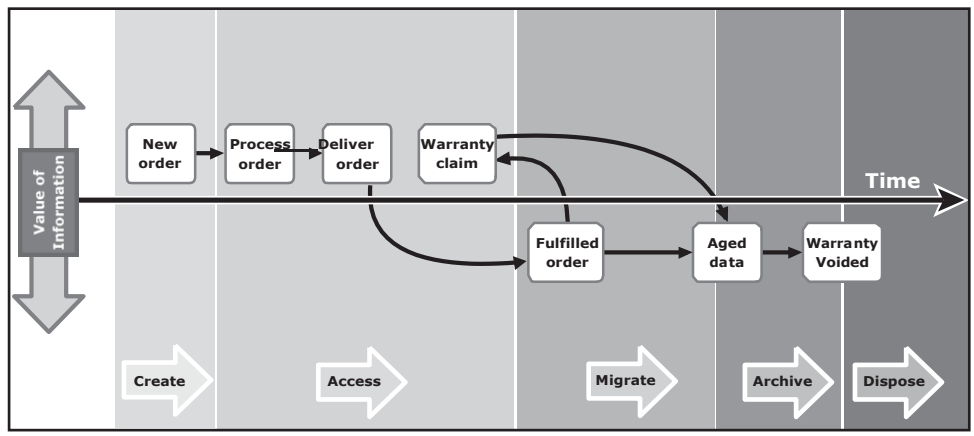


Figure 15-11: Changing value of sales order information

Information Lifecycle Management (ILM) is a proactive strategy that enables an IT organization to effectively manage information throughout its life cycle based on predefined business policies. From data creation to data deletion, ILM aligns the business requirements and processes with service levels in an automated fashion. This allows an IT organization to optimize the storage infrastructure for maximum return on investment. Implementing an ILM strategy has the following key benefits that directly address the challenges of information management:

- **Lower Total Cost of Ownership (TCO):** By aligning the infrastructure and management costs with information value. As a result, resources are not wasted, and complexity is not introduced by managing low-value data at the expense of high-value data.
- **Simplified management:** By integrating process steps and interfaces with individual tools and by increasing automation
- **Maintaining compliance:** By knowing what data needs to be protected for what length of time
- **Optimized utilization:** By deploying storage tiering

Storage Tiering

Storage tiering is a technique of establishing a hierarchy of different storage types (tiers). This enables storing the right data to the right tier, based on service level requirements, at a minimal cost. Each tier has different levels of protection, performance, and cost. For example, high performance solid-state drives (SSDs) or FC drives can be configured as tier 1 storage to keep frequently accessed data, and low cost SATA drives as tier 2 storage to keep the less frequently accessed data. Keeping frequently used data in SSD or FC improves application performance. Moving less-frequently accessed data to SATA can free up storage capacity in high performance drives and reduce the cost of storage. This movement of data happens based on defined tiering policies. The tiering policy might be based on parameters, such as file type, size, frequency of access, and so on. For example, if a policy states "Move the files that are not accessed for the last 30 days to the lower tier," then all the files matching this condition are moved to the lower tier.

Storage tiering can be implemented as a manual or an automated process. *Manual storage tiering* is the traditional method where the storage administrator monitors the storage workloads periodically and moves the data between the tiers. Manual storage tiering is complex and time-consuming. *Automated storage tiering* automates the storage tiering process, in which data movement between the tiers is performed nondisruptively. In automated storage tiering, the application workload is proactively monitored; the active data is automatically moved to a higher performance tier and the inactive data to a higher capacity, lower performance tier. Data movements between various tiers can happen within (intra-array) or between (inter-array) storage arrays.

Intra-Array Storage Tiering

The process of storage tiering within a storage array is called *intra-array storage tiering*. It enables the efficient use of SSD, FC, and SATA drives within an array and provides performance and cost optimization. The goal is to keep the SSDs busy by storing the most frequently accessed data on them, while moving out the less frequently accessed data to the SATA drives. Data movements executed between tiers can be performed at the LUN level or at the sub-LUN level. The performance can be further improved by implementing tiered cache. LUN tiering, sub-LUN tiering, and cache tiering are detailed next.

Traditionally, storage tiering is operated at the LUN level that moves an entire LUN from one tier of storage to another (see Figure 15-12 [a]). This movement includes both active and inactive data in that LUN. This method does not give effective cost and performance benefits. Today, storage tiering

can be implemented at the sub-LUN level (see Figure 15-12 [b]). In sub-LUN level tiering, a LUN is broken down into smaller segments and tiered at that level. Movement of data with much finer granularity, for example 8 MB, greatly enhances the value proposition of automated storage tiering. Tiering at the sub-LUN level effectively moves active data to faster drives and less active data to slower drives.

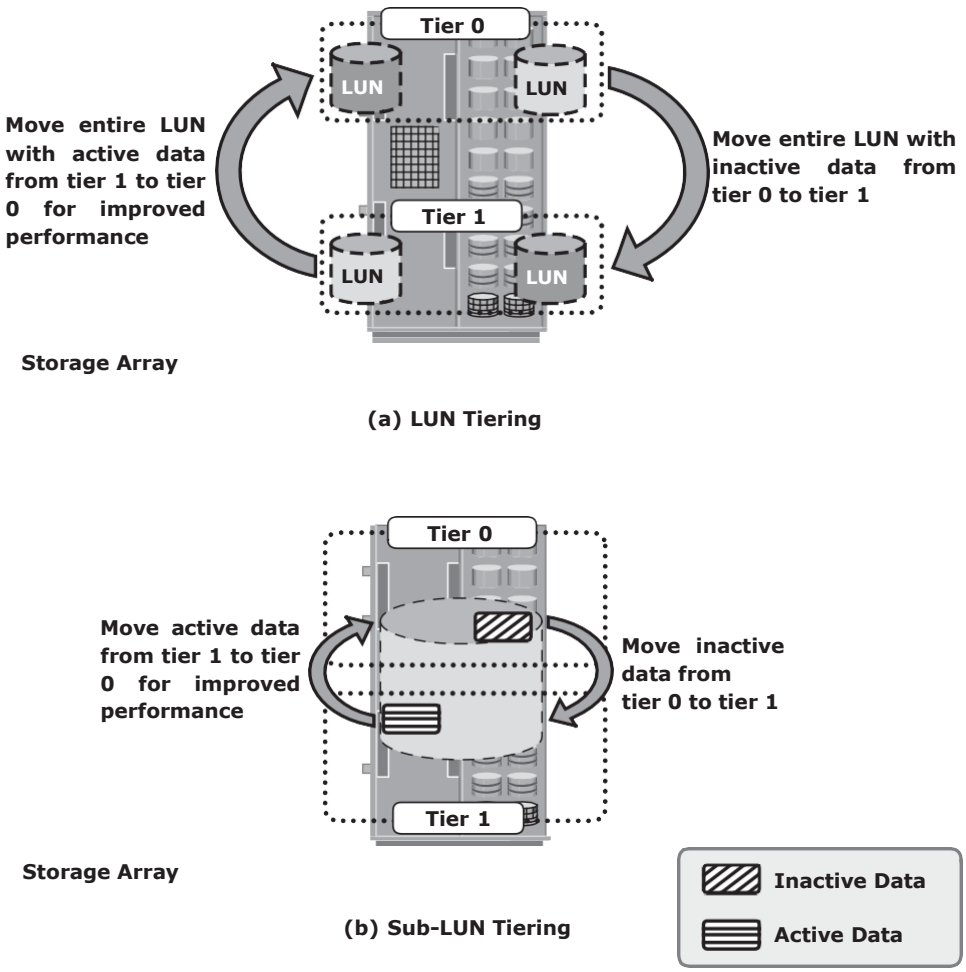


Figure 15-12: Implementation of intra-array storage tiering

Tiering is also be implemented at the cache level, as shown in Figure 15-13. A large cache in a storage array improves performance by retaining a large amount of frequently accessed data in a cache, so most reads are served directly from the

cache. However, configuring a large cache in the storage array involves more cost. An alternative way to increase the size of the cache is by utilizing the SSDs on the storage array. In cache tiering, SSDs are used as a large capacity secondary cache to enable tiering between DRAM (primary cache) and SSDs (secondary cache). Server flash-caching is another tier of cache in which a flash-cache card is installed in the server to further enhance the application's performance.

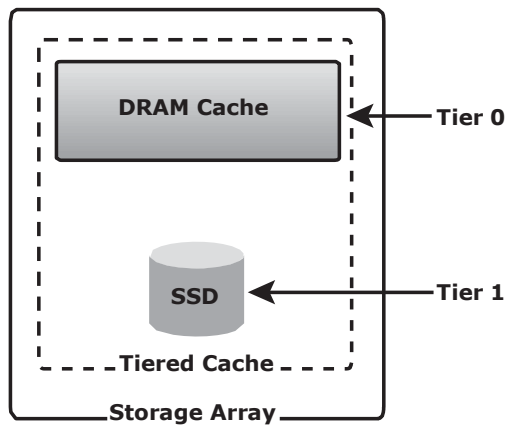


Figure 15-13: Cache tiering

Inter-Array Storage Tiering

The process of storage tiering between storage arrays is called *inter-array storage tiering*. Inter-array storage tiering automates the identification of active or inactive data to relocate them to different performance or capacity tiers between the arrays. Figure 15-14 illustrates an example of a two-tiered storage environment. This environment optimizes the primary storage for performance and the secondary storage for capacity and cost. The policy engine, which can be software or hardware where policies are configured, facilitates moving inactive or infrequently accessed data from the primary to the secondary storage. Some prevalent reasons to tier data across arrays is archival or to meet compliance requirements. As an example, the policy engine might be configured to relocate all the files in the primary storage that have not been accessed in one month and archive those files to the secondary storage. For each archived file, the policy engine creates a small space-saving stub file in the primary storage that points to the data on the secondary storage. When a user tries to access the file at its original location on the primary storage, the user is transparently provided with the actual file from the secondary storage.

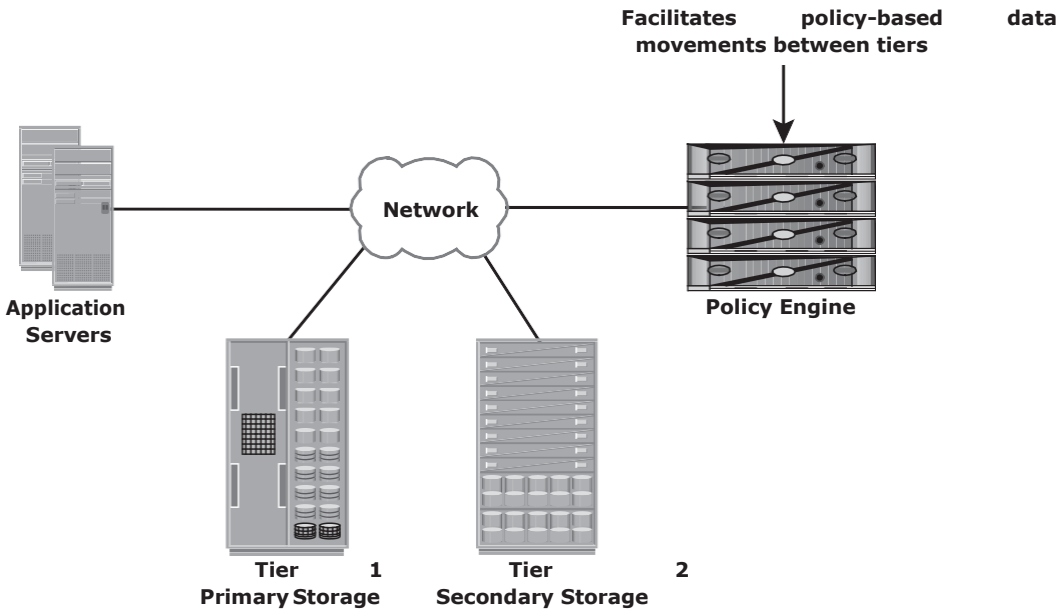


Figure 15-14: Implementation of inter-array storage tiering

Concepts in Practice: EMC Infrastructure Management Tools

Businesses today face challenges in managing their IT infrastructure due to the large number of heterogeneous resources in their environment. These resources may be physical resources, virtualized resources, or cloud resources. EMC offers different tools that satisfy different requirements of the business. EMC ControlCenter and ProSphere are suites of software that can perform end-to-end management of storage infrastructure, while EMC Unisphere is software that manages EMC storage arrays, such as VNX and VNXe. EMC Unified Infrastructure Manager (UIM) is software that manages the Vblock infrastructure (cloud resources). For more information, visit `www.emc.com/`.

EMC ControlCenter and Prosphere

EMC ControlCenter is a family of storage resource management (SRM) applications that provide a unified solution to manage a multivendor storage infrastructure. It helps address the challenges to manage a large, complex storage environment that includes hosts, storage networks, storage, and virtualization across all the layers. ControlCenter provides capabilities, such as storage planning,

provisioning, monitoring, and reporting. It enables implementing an ILM strategy by providing comprehensive management of tiered storage infrastructure. It also provides an end-to-end view of the entire networked storage infrastructure that includes SAN, NAS, and host storage resources, including a virtualized environment. It provides a central administrative console, discovery of new components, quota management, event management, root cause analysis, and chargeback. ControlCenter comes with built-in security features that provide access control, data confidentiality, data integrity, logging, and auditing. It offers an intuitive, easy-to-use interface that provides insight into the complex relationships of the environment. ControlCenter uses an agent to discover the components in the environment.

EMC ProSphere is also storage resource management software built to meet the demands of the new cloud computing era. EMC ProSphere improves productivity and service levels in the virtualized and cloud environment. ProSphere includes the following key capabilities:

- **End-to-end visibility:** It offers an intuitive, easy-to-use interface that provides insight into the complex relationships between objects in large, virtualized environments.
- **Multi-site management:** From a single console, ProSphere's federated architecture aggregates information from across sites and simplifies information management between data centers. ProSphere is managed from a web browser to allow easy access over the Internet for remote management.
- **Improved productivity in growing virtualized environments:** ProSphere introduces an innovative technology called Smart Groups, which groups objects with similar characteristics into a user-defined group for performing management tasks. This enables IT to take a policy-based approach to manage objects or to set data collection policies.
- **Fast, easy, and efficient deployment:** Agent-less discovery eliminates the burden of deploying and managing host agents. ProSphere is packaged as a virtual appliance that can be installed in a short time.
- **Delivery of IT as a service:** With ProSphere, service levels can now be monitored from host-to-storage layers. This allows organizations to maintain consistent service levels at an optimal price-performance ratio to meet business objectives to delivering IT-as-a-service.

EMC Unisphere

EMC Unisphere is a unified storage management platform that provides intuitive user interfaces for managing EMC VNX and EMC VNXe storage arrays. Unisphere

is web-enabled and supports remote management of storage arrays. Some of the key capabilities offered by Unisphere follow:

- Provides unified management for file, block, and object storage
- Provides single sign-on for all devices in a management domain
- Supports automated storage tiering and ensures that data is stored in the correct tier to meet performance and cost
- Provides management of both physical and virtual components

EMC Unified Infrastructure Manager (UIM)

EMC Unified Infrastructure Manager is a unified management solution for Vblocks. (Vblock is covered in Chapter 13.) It enables configuring the Vblock infrastructure resources and activating cloud services. It provides a single user interface to manage multiple Vblocks and eliminates the need for configuring compute, network, and storage separately using different virtual infrastructure management tools. UIM provides a dashboard that shows how the Vblock infrastructure is configured and how the resources are used. This enables an administrator to monitor the configuration and utilization of the Vblock infrastructure resources and to plan for capacity requirements. UIM also provides a topology or a map view of the Vblock infrastructure, which enables an administrator to quickly locate and understand the interconnections of the Vblock infrastructure components and services. It provides an alerts console, which allows an administrator to see the alerts against the Vblock infrastructure resources and the associated services affected by problems. UIM performs a compliance check during resource configuration. It validates compliance with configuration best practices. It also prevents conflicting resource identity assignments, for example, accidentally assigning a MAC address to more than one virtual NIC.